AMSAA

TECHNICAL REPORT NO. 97

LEGAL MIX: A FORTRAN MODEL FOR EVALUATING ARTILLERY SYSTEMS

EDWARD G. STAUCH

JULY 1974

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

U.S. ARMY MATERIEL SYSTEMS ANALYSIS ACTIVITY
Aberdeen Proving Ground, Maryland

DISPOSITION

Destroy this report when no longer needed. Do not return it to the originator.

DISCLAIMER

The findings in this report are not to be construed as an official Department of the Army position.

WARNING

Information and data contained in this document are based on the input available at the time of preparation. The results may be subject to change and should not be construed as representing the AMC position unless so specified.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER TECHNICAL REPORT NO. 97	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) LEGAL MIX: A FORTRAN MODEL FOR EVALUATING ARTILLERY SYSTEMS		5. TYPE OF REPORT & PERIOD COVERED 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Edward G. Stauch		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS US Army Materiel Systems Analysis Activity Aberdeen Proving Ground, Md. 21005		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS RDT&E Project No. 1T765706M541
11. CONTROLLING OFFICE NAME AND ADDRESS US Army Materiel Command 5001 Eisenhower Avenue Alexandria, VA 22333		12. REPORT DATE July 1974 13. NUMBER OF PAGES 195
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)

Artillery Simulation Model
Legal Mix IV
Artillery Force Performance
Artillery Effectiveness
Target Military Worth

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

A FORTRAN version of the "Legal Mix" Artillery Simulation Model has been developed at the U.S. Army Materiel Systems Analysis Activity (AMSAA). The model is similar to versions that were originally used for conduct of both "Legal Mix III" and "IV" studies. The model is designed to measure the relative performance of a series of "division-slice" families of artillery weapons by simulating the basic demands placed on an artillery force during the typical day of combat. This report furnishes documentation of the model to include

20. ABSTRACT (CONTINUED):

basic model description, model flow charts, and listing of sample input and output data.

ACKNOWLEDGEMENT

The author wishes to thank Mr. Eddie Morrow of AMSAA for his substantial effort in coding the FORTRAN program and for his assistance in outlining the input data requirement. Also, a word of thanks is due Mrs. Cinda J. Roberts for her suggestions in the preparation of the program flow charts and for the typing of the flow charts.

The following page is blank.

CONTENTS

The following page is blank.

LEGAL MIX: A FORTRAN MODEL FOR EVALUATING ARTILLERY SYSTEMS

1. INTRODUCTION

A FORTRAN II and IV version of the Legal Mix Artillery simulation model has been developed for evaluating artillery systems. An earlier version of the model was prepared in 1966 in the FORAST coding for use on the BRLESC I computer of the Ballistic Research Laboratories at Aberdeen Proving Ground, Maryland (Reference 1).

The rationale, assumptions and methodology used in preparing the model are found in References 2 and 3. The sole intent of this report is to provide a "users manual" for those who desire to employ the Legal Mix model in the evaluation of artillery systems. Therefore, a minimum of supporting methodology is included.

2. METHODOLOGY

The methodologies employed in the program are based upon those found in References 2 and 3. In essence, the model employs five components, which are briefly outlined in the following paragraphs.

2.1 Target List.

The target list, derived from a war game and subsequent target acquisition analysis, represents the threat and demands on the friendly artillery force. The result of the derivation is a time-ordered list of both acquired target missions and pre-planned support missions such as illumination, smoke and harassment fires. Each target on the list is described by a number of data elements, including location, time of acquisition, estimated target duration time, number of tactical elements (personnel, tanks and armored personnel carriers), and other estimated and actual data. (See paragraph 3.2 for a complete listing of target parameters). Several parameters require a brief explanation:

- Target Frequency - The program allows four levels of battle intensity: Low, Mid, Base and High. The target list itself represents

---

[1] Odom, C.T., Kramar, J. W., and Thomas, A.S., An Improved Model for Evaluating Artillery Weapons, BRL Report No. 1321, September 1966, APG, MD. UNCLASSIFIED

[2] Report, Optimum Mix of Artillery Units, 1971-1975 (U), Phase III, USACDC Field Artillery Agency, July 1967, Fort Sill, OK, SECRET

[3] Report, Optimum Mix of Artillery Units, 1975-1980 (U) (Short Title: Legal Mix IV), USACDC Field Artillery Agency, August 1970, SECRET

Base-intensity, where each listed target acquisition signifies a single (frequency = 1) demand on the artillery force. For other intensities, however, the target frequency may be increased (or eliminated) as a demand on the force. Therefore, based on an analysis of the war game which generated the target list, each target acquisition is assigned a frequency for each of the allowed intensities.

- Military Worth - Based upon questionnaires administered to a group of field grade officers representing various combat arms, a scale of relative military worth values has been developed for the various type tactical elements on the target list. (References 3 & 4). This military worth value is used for various purposes in the program. First, it provides for a priority ordering of targets for attack, whereby the acquisitions with highest military worth are attacked first in each game time increment. Secondly, it allows for a segmenting of targets into categories which control the level of attack and allowable ammunition weight expenditure against a target. Table 2.1 lists the categories utilized in the program. Lastly, Military Worth provides for a measure of force performance, by suming up the military worth points of damaged target elements.

- Target Posture Mix - Past efforts have identified typical postures for the elements (personnel, tanks and APC's) which make up each target (Reference 2). These postures indicate the percentage of personnel standing, prone and crouching (in foxholes) as well as the status of materiel elements (static or moving) and proximity to the Forward Edge of the Battle Area (FEBA) for both warned and unwarned conditions. The Legal Mix studies have defined 12 "posture mixes" accounting for various combinations of these postures. (See the typical data inputs in paragraph 3.1).

2.2 Artillery Force.

The second component of the model accounts for the artillery resources available to the friendly force. The allowable fire units and their movement schedules throughout the battlefield day are generated from the tactical situation developed in the war game. Associated with each fire unit are weapon system and ammunition parameters which define the capabilities of the artillery force. (See paragraph 3.1 for a detailed listing of the various parameters.)

---

[3] Loc. Cit.

[4] Wood, William J., and Tice, Jerry, A Proposed Method for Determining Target Worth as an Input to Weapon Systems Analyses, AMSAA TM 15, Oct 68, Aberdeen Proving Ground, MD, UNCLASSIFIED.

[2] Loc. Cit.

TABLE 2.1 MILITARY WORTH

Target Category	Military Worth Points
I	M.W. $\geq$ 10 (Maximum value = 21.)
II	3 $\leq$ M.W. < 10
III	M.W. < 3

2.3 Effectiveness Computation.

The model employs the same basic effectiveness computation routine as outlined in Reference 5. This routine determines the number of rounds and fire units required to reach specified attack levels against estimated data for each target, and calculates the amount of target damage inflicted, in terms of fractional survivors, against actual target data. The program examines each target in priority order and identifies the possible attack solutions available at the time the target is presented for consideration. Three attack level thresholds are used in the model as criteria for engagement (Reference 3):

- Threshold A, representing a defeat level of 50% damage

- Threshold B, representing a fixed level of damage required to disrupt unit integrity. For materiel targets, this level is specified at 30% damage, while for personnel targets the level is set at 1/2 (100-F)% where F represents the percentage of personnel in foxholes, and where the quantity 1/2 (100-F) is restrained between 25% to 50% damage levels.

- Threshold C is restricted to Category I and II targets (Military Worth $\geq$ 3.00) and is defined as a minimum acceptable damage level equal to (.9/Target Military Worth). This attack level is used only when Threshold B damage cannot be achieved against Category I and II targets and the target's duration is such that the target will not be considered for attack again.

2.4 Allocation Process.

The allocation process in the model controls the massing of fire units and the tactical method of attack, in determining the optimum solution against a target. Two attack methods are examined:

- One-volley method - Fire units are added as necessary to reach the specified attack levels when constrained to fire only one volley per unit.

- Multi-volley method - Fire units firing all available (within specified constraints) ammunition are added in turn in order of effectiveness until the specified attack level is reached.

---

[5] Odom, C.T., Kramar, J.W., Michels, H.W., Thomas, A.S., and Thomas, C.M., Reoptimization of a Multiple Artillery Rocket System - MARS II (U), BRL Report No. 1736, September 1966, Aberdeen Proving Ground, MD, SECRET.

[3] Loc. Cit.

The order in which units at the various tactical echelons are examined and massed depends on the echelon which acquired the target, as determined in the war game. The order of massing fire units is shown in Table 2.2.

TABLE 2.2 FIRE UNIT MASSING ORDER

Attempted Solution	Acquiring Echelon		
	DS	GS	CORPS
1st	Closest DS alone	GS alone	CORPS alone
2nd	GS alone	GS & DS	CORPS & GS
3rd	GS & DS	CORPS alone	All
4th	CORPS alone	CORPS & GS	
5th	CORPS & GS	All	
6th	All		

2.5 Model Outputs

The final component of the model provides for an hourly game output of effort and effectiveness measures. The principal measures of effort are the cost and weight of ammunition expended against the target list. Effectiveness is measured in the amount of personnel and materiel damage inflicted; the number of targets fired upon, defeated, and not engaged; and a summary of military worth points scored. A sample output is shown in Appendix E. The user, of course, may desire to print out additional data by adding the appropriate computer statements to the OUTPUT Subroutine.

3. INPUT FORMATS

All data inputs, except for the target list, are entered by standard 80-column cards. The target list is entered from 120-column tape. The following paragraphs outline the various input requirements to include columns, variable names and units associated with each input variable.

3.1 Card Input

For ease in setting up the input card deck, the cards are segmented into 16 card sets which are read into the computer in five subroutines, as indicated in Figure Figure 3.1. Tables 3.1 through 3.5 outline the specific sets, cards, columns and units for the subroutines. Appendix D contains a tabulation of sample input data.

3.2 Target Tape Input

Each target on the target list is described by 33 data points on two lines of tape input. For special (or "OTHER") type targets (H&I, Illumination & Smoke Missions) there is an additional line of tape input with 8 data points. Format for first 2 lines is: FORMAT (F7.1, 16F7.2,/ 16F8.2). Format for 3rd line (when used) is: FORMAT (8F8.2).

Targets are listed on the tape in time sequence, according to their estimated time of arrival in the battle. As a target is input, the data points are assigned to a TNI(I) list, from which they are then transferred to the TN(I,J) array according to the priority order in which they will be attacked. Table 3.6 outlines the specific data points required for each target.

Tapes currently available for use with the Legal Mix models which were generated for the Legal Mix Studies are:

- European Target Array (Reference 3)
- Korean Target Array (Reference 3)

Efforts are now in progress to conduct additional war games to furnish updated European target lists.

4. SUMMARY

The basic input formats representing friendly capabilities and the enemy threat have been described. Detailed rationale and methodologies may be found in the cited literature.

The computer program as written (Appendix A) requires approximately 60,000 words of computer memory. This memory requirement may be lessened by reducing the present number of rounds (50) and fire units (100) allowed in the model. A typical computer run of the model requires from 30 to 60 minutes, depending on the number of systems, units, and rounds considered in a given "mix".

Additions and modifications to the model are planned in the near future to further expand the usefulness and applicability of the Legal Mix methodology. Suggestions and comments are welcomed by the author.

---

[3] Loc. Cit.

Figure 3.1 Card Inputs

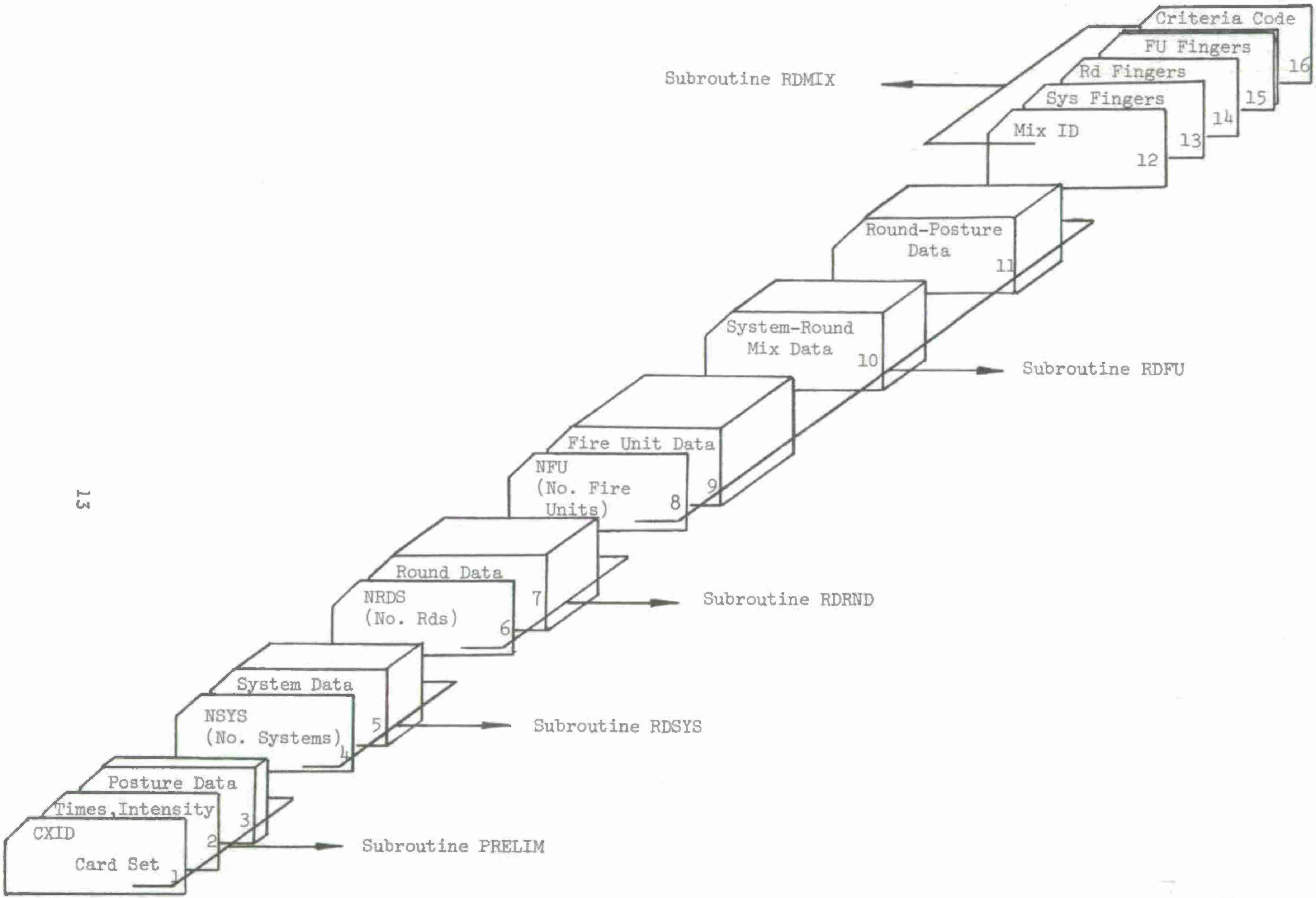


TABLE 3.1 SUBROUTINE PRELIM

Card Set	No. of Cards	Card Cols.	Variable Name	Input Units	Comments
1	1	1-80	CXID	Alpha-Numeric	Force Identifier - Used for info only.
2	1	1-7	TZRO	Dec. Hours	Game Start Time (First printout is at Start + 1 Hour)
		8-14	TMX	Dec. Hours	Game End Time (Last printout at Game End Hour)
		15-21	FACT	Real	Game Intensity Level Key: 1-Low; 2-Mid; 3-Base; 4-High
3	12		POST(I,J)		Lists Posture Mix for each of 12 allowable posture mixes.
	(1st)	1-7	POST(1,1)	0	Posture ID Number for 1st posture
		8-14	POST(1,2)	Real	% of UNWARNED Pers. Standing for 1st posture
		15-21	POST(1,3)	Real	% of UNWARNED Pers. Prone for 1st posture
		22-28	POST(1,4)	Real	% of UNWARNED Pers. in Foxholes for 1st posture
		29-35	POST(1,5)	Real	Key for UNWARNED Tanks for 1st posture 0 = No tanks 1 = Tanks
		36-42	POST(1,6)	Real	Key for UNWARNED APC's for 1st posture 0 = No APC's 1 = APC's
		43-49	POST(1,7)	Real	% of WARNED Pers. Standing for 1st posture

TABLE 3.1 SUBROUTINE PRELIM (CONT'D)

Card Set	No. of Cards	Card Cols.	Variable Name	Input Units	Comments
3 Cont'd		50-56	POST(1,8)	Real	% of WARNED Pers. Prone for 1st posture
		57-63	POST(1,9)	Real	% of WARNED Pers. in Foxholes for 1st posture
		64-70	POST(1,10)	Real	Key for WARNED Tanks for 1st posture 0 = No tanks 1 = Tanks
		71-77	POST(1,11)	Real	Key for WARNED APC's for 1st posture 0 = No APC's 1 = APC's
	(2nd) thru (12th)				Same 11 data points for the 2nd thru 12th postures. (Posture numbers are 0 thru 11 for the 12 allowable postures.

TABLE 3.2 SUBROUTINE RDSYS

Card Set	No. of Cards	Card Cols.	Variable Name	Input Units	Comments
4	1	1-5	NSYS	Systems	Number of weapon systems in force (NSYS≤10)
5	2* NSYS				These 2*NSYS cards list 12 data points for each system.
	(1st)	1-7	SYSID(1)	Real	System ID number for 1st system (e.g. 1200.1)
		8-14	FRWM(1)	Real	Fraction of 1st system units remaining in place during unit moves.
		15-21	TPFU(1)	Tubes/Unit	Tubes (or launchers) per FU for 1st system.
		22-28	SROF(1)	Rd/Min/Tube	Maximum rate of fire vs static targets for 1st system.
		29-35	DROF(1)	Rd/Min/Tube	Maximum rate of fire vs moving targets for 1st system.
		36-42	TBM(1)	Minutes	Time between missions - the time to set up and fire 1 volley for 1st system.
		43-49	BLD(1)	Rd/Unit	Ammunition Basic Load for units of 1st system.
		50-56	RSPY(1)	Rd/Unit/Hr	Ammunition Resupply Rate for units of 1st system.
		57-63	SNMX(1)	Rd/Tube/Mission	Maximum rounds allowed per mission vs static target for 1st system units.

TABLE 3.2 SUBROUTINE RDSYS (CONT'D)

Card Set	No. of Cards	Card Cols.	Variable Name	Input Units	Comments
5 Cont'd		64-70	DNMX(1)	Rd/Tube/Mission	Maximum rounds allowed per mission vs moving target for 1st system units.
	(2nd)	1-7	HNMX(1)	Rd/Tube/Hr	Maximum rounds allowed in 1 hour vs all targets for 1st system units.
		8-14	STYP(1)	Real	Key to 1st system type: 1 = Cannon; 2 = Missile
	(3rd) thru (2*NSYS)				Same 12 data points for the other systems.

TABLE 3.3 SUBROUTINE RDRND

Card Set	No. of Cards	Card Cols.	Variable Name	Input Units	Comments
6	1	1-5	NRDS	Rounds	Number of different rounds in force (NRDS≤50)
7	6 or 14 per each of NRDS groups				There are NRDS card groups in card set 7. Depending on the round type (RTP(I)), there are either 6 or 14 cards per group. If RTP(I)=1 (ICM type round) there are 6 cards; if RTP(I)=2 (HE type round) there are 14 cards in the group. The first 4 cards of all groups contain data for the same variables.
	(1st)	1-7	RNDID(I)	Real	Round ID number for Ith Rd. (e.g. 1203.1)
		8-14	WGT(I)	Metric/Tons/Rd	Crated weight of Ith Rd.
		15-21	CST(I)	Kilo $/Rd	Cost per round of Ith Rd.
		22-28	RMX(I)	KM	Maximum range of Ith Rd.
		29-35	REL(I)	Real	In-flight reliability of Ith Rd.
		36-42	DEP(I)	Real	% of recoverable misfires for Ith Rd. (Not used in program).
		43-49	RTP(I)	Real	Key to rd. type for Ith Rd: 1=ICM; 2=HE
		50-56	WARN(I)	Seconds	Signature or Warning for Ith Rd.

TABLE 3.3 SUBROUTINE RDRND (CONT'D)

Card Set	No. of Cards	Card Cols.	Variable Name	Input Units	Comments
7 Cont'd	(2nd)	1-7	RNG(I,1)	KM	1st range value in list of up to 10 ranges for I^{th} Rd.
		8-14	RNG(I,2)	KM	2nd range value in list of up to 10 ranges for I^{th} Rd.
		15-70	RNG(I,3) thru RNG(I,10)	KM	3rd thru 10th range value in list of up to 10 ranges for I^{th} Rd.
	(3rd)	1-7	CPR(I,1)	Meters	CPE (Random) at 1st range value for I^{th} Rd.
		8-14	CPR(I,2)	Meters	CPE (Random) at 2nd range value for I^{th} Rd.
		15-70	CPR(I,3) thru(I,10)	Meters	CPE (Random) at 3rd thru 10th range value for I^{th} Rd.
	(4th)	1-7	CPS(I,1)	Meters	CPE (Total) at 1st range value for I^{th} Rd.
		8-14	CPS(I,2)	Meters	CPE (Total) at 2nd range value for I^{th} Rd.
		15-70	CPS(I,3) thru CPS(I,10)	Meters	CPE (Total) at 3rd thru 10th range value for I^{th} Rd.

TABLE 3.3 SUBROUTINE RDRND (CONT'D)

Card Set	No. of Cards	Card Cols.	Variable Name	Input Units	Comments
7 Cont'd			If RTP(I)=1 (ICM), the next 2 cards complete the 6 card group. (Subroutine ACMINP)		
	(5th)	1-7	SRE	Real	Slope of Radius of Effects vs Range plot for I^{th} Rd.
		8-14	REZ	Meters	"Y" - intercept of Radius of Effects vs Range plot for I^{th} Rd.
		15-21	SRO	Real	Submissile reliability in open environment for I^{th} Rd.
		22-28	SRW	Real	Submissile reliability in wooded environment for I^{th} Rd.
		29-35	EN	Submissiles	Number of submissiles in I^{th} Rd.
	(6th)	1-7	AL(1)	M^2	Lethal Area of 1 submissile vs standing target in open for I^{th} Rd.
		8-14	AL(2)	M^2	Lethal Area of 1 submissile vs prone target in open for I^{th} Rd.
		15-21	AL(3)	M^2	Lethal Area of 1 submissile vs foxhole target in open for I^{th} Rd.
		22-28	AL(4)	M^2	Lethal Area of 1 submissile vs tank target in open for I^{th} Rd.

TABLE 3.3 SUBROUTINE RDRND (CONT'D)

Card Set	No. of Cards	Card Cols.	Variable Name	Input Units	Comments
7 Cont'd		29-35	AL(5)	M^2	Lethal Area of 1 submissile vs APC target in open for I^{th} Rd.
		36-42	AL(6)	M^2	Lethal Area of 1 submissile vs standing target in woods for I^{th} Rd.
		43-49	AL(7)	M^2	Lethal Area of 1 submissile vs prone target in woods for I^{th} Rd.
		50-56	AL(8)	M^2	Lethal Area of 1 submissile vs foxhole target in woods for I^{th} Rd.
		57-63	AL(9)	M^2	Lethal Area of 1 submissile vs tank target in woods for I^{th} Rd.
		64-70	AL(10)	M^2	Lethal Area of 1 submissile vs APC target in woods for I^{th} Rd.
			If RTP(I)=2 (HE), the next 10 cards complete the 14 card group. (Subroutine HEINP)		
	(5th)	1-7	AL(1)	M^2	Rd. Lethal Area vs standing target in open at 1st range for I^{th} Rd.
		8-14	AL(2)	M^2	Rd. Lethal Area vs standing target in open at 2nd range for I^{th} Rd.
		15-70	AL(3)-AL(10)	M^2	Rd. Lethal Area vs standing target in open at 3rd thru 10th ranges for I^{th} Rd.

TABLE 3.3 SUBROUTINE RDRND (CONT'D)

Card Set	No. of Cards	Card Cols.	Variable Name	Input Units	Comments
7 Cont'd	(6th)	1-70	AL(11)-AL(20)	M^2	Rd. Lethal Area vs prone target in open at 10 ranges for I^{th} Rd.
	(7th)	1-70	AL(21)-AL(30)	M^2	Rd. Lethal Area vs foxhole target in open at 10 ranges for I^{th} Rd.
	(8th)	1-70	AL(31)-AL(40)	M^2	Rd. Lethal Area vs tank target in open at 10 ranges for I^{th} Rd.
	(9th)	1-70	AL(41)-AL(50)	M^2	Rd. Lethal Area vs APC target in open at 10 ranges for I^{th} Rd.
	(10th)	1-70	AL(51)-AL(60)	M^2	Rd. Lethal Area vs standing target in woods at 10 ranges for I^{th} Rd.
	(11th)	1-70	AL(61)-AL(70)	M^2	Rd. Lethal Area vs prone target in woods at 10 ranges for I^{th} Rd.
	(12th)	1-70	AL(71)-AL(80)	M^2	Rd. Lethal Area vs foxhole target in woods at 10 ranges for I^{th} Rd.
	(13th)	1-70	AL(81)-AL(90)	M^2	Rd. Lethal Area vs tank target in woods at 10 ranges for I^{th} Rd.
	(14th)	1-70	AL(91)-AL(100)	M^2	Rd. Lethal Area vs APC target in woods at 10 ranges for I^{th} Rd.

TABLE 3.4 SUBROUTINE RDFU

Card Set	No. of Cards	Card Cols.	Variable Name	Input Units	Comments
8	1	1-5	NFU	Fire Units	Number of Fire Units in Force ($NFU \le 100$).
9	NFU Groups				There are NFU card groups in card set 9. The number of cards in each group depends on the number of firing sites the FU will occupy during the game (NSITE(I)). Each group has 2+NSITE(I) cards. ($NSITE(I) \le 8$).
	(1st)	1-5	NSITE(I)	SITES	Number of sites for the I^{th} fire unit.
	(2nd)	1-8	FSID(I)	Real	Identifies which of the NSYS systems this FU is. (e.g. 1200.1)
	(3rd) thru (2+NSITE(I))	1-8	TA(1,I)	HR.MIN	Arrival Time of I^{th} FU at its 1st site. (Example: 9 hr & 15 min is input as 9.15)
		9-16	TD(1,I)	HR.MIN	Departure Time of I^{th} FU from its 1st site.
		17-24	XS(1,I)	KM	X-coordinate of I^{th} FU's 1st site (Easting)
		25-32	YS(1,I)	KM	Y-coordinate of I^{th} FU's 1st site (Northing)
There is a card with the above 4 data points for each of I^{th} FU's sites.					
10	NSYS				This card set contains one card per system to identify the rounds allowed for each system.
	(1st) thru NSYS	1-7	SYSRD(I,1)	Real	Weapon system ID number for I^{th} system (e.g. 1200.1)

TABLE 3.4 SUBROUTINE RDFU (CONT'D)

Card Set	No. of Cards	Card Cols.	Variable Name	Input Units	Comments
10 Cont'd	(1st) thru (NSYS)	8-14	SYSRD(I,2)	Real	Rd. ID of 1st rd allowed for I^{th} system (e.g. 1203.1)
		15-21	SYSRD(I,3)	Real	Rd. ID of 2nd rd allowed for I^{th} system.
		22-70	SYSRD(I,4) thru (I,10)	Real	Rd. ID of 3rd thru max of 9th rd for I^{th} system.
11					This card set contains 12 groups (one for each allowable posture) which define which rounds are allowed to be fired vs each posture. The number of cards in each group depends on the number of rounds allowed vs each posture in each of the two environments - open and woods.
	(1st)	1-5	NP	ID No.	Posture ID Number of first posture (=0)
		6-10	NRO(I)	Rds	Number of Rds allowed vs 1st posture in open ($NRO(I) \leq 12$).
		11-15	NRW(I)	Rds	Number of Rds allowed vs 1st posture in woods ($NRW(I) \leq 12$).
	(2nd)	1-7	ORVP(I,1)	Real	Rd. ID of 1st allowable round vs 1st posture in open.
		8-14	ORVP(I,2)	Real	Rd. ID of 2nd allowable round vs 1st posture in open.

TABLE 3.4 SUBROUTINE RDFU (CONT'D)

Card Set	No. of Cards	Card Cols.	Variable Name	Input Units	Comments
11 Cont'd		15-70	ORVP(I,3) thru ORVP(I,10)	Real	Rd. ID of 3rd thru 10th allowable round vs 1st posture in open.
	(3rd)	1-7	ORVP(I,11)	Real	Rd. ID of 11th allowable round vs 1st posture in open.
		8-14	ORVP(I,12)	Real	Rd. ID of 12th allowable round vs 1st posture in open.
					This 3rd card is needed only if 11 or 12 rds are allowed vs the 1st posture in open.
	(4th)	1-7	WRVP(I,1)	Real	Rd. ID of 1st allowable round vs 1st posture in woods.
		8-14	WRVP(I,2)	Real	Rd. ID of 2nd allowable round vs 1st posture in woods.
		15-70	WRVP(I,3) thru WRVP(I,10)	Real	Rd. ID of 3rd thru 10th allowable round vs 1st posture in woods.
	(5th)	1-7	WRVP(I,11)	Real	Rd. ID of 11th allowable round vs 1st posture in woods.

TABLE 3.4 SUBROUTINE RDFU (CONT'D)

Card Set	No. of Cards	Card Cols.	Variable Name	Input Units	Comments
11 Cont'd		8-14	WRVP(I,12)	Real	Rd. ID of 12th allowable round vs 1st posture in woods.
					This 5th card is needed only if 11 or 12 rds. are allowed vs the 1st posture in woods. These 5 cards are repeated for the other 11 postures.

TABLE 3.5 SUBROUTINE RDMIX

Card Set	No. of Cards	Card Cols.	Variable Name	Input Units	Comments
12	1	1-80	MIXID	Alpha-Numeric	System-Rd-FU Mix Identifer. Used for Info only.
13	1	1	KSIG(1)	0 or 1	This card keys those systems (of up to 10 defined in RDSYS) which are allowed in this mix. A "1" signifies the system is in the mix; "0" = not in mix.
		2	KSIG(2)	0 or 1	
		3 thru 10	KSIG(3)-(10)	0 or 1	
14	1	1	KRIG(1)	0 or 1	This card keys those rounds (of up to 50 defined in RDRND) which are allowed in this mix. A "1" signifies the rd is in the mix; a "0" = not in mix.
		2	KRIG(2)	0 or 1	
		3 thru 50	KRIG(3)-(50)	0 or 1	
15	1st	1	KFIG(1)	0,1,2, or 3	These cards key those fire units (of up to 100 identified in RDFU) which are allowed in this mix. 0 = not in mix; 1 = at DS echelon; 2 = at GS echelon; 3 = at CORPS echelon. The second card is not used if 80 or less FU's have been identified in RDFU.
		2 thru 80	KFIG(2)-(80)	0,1,2, or 3	
	2nd	1 thru 20	KFIG(81)-(100)	0,1,2 or 3	
16	1	1-7	CRTERA	1.or 2.	Keys which criterion is to be considered in choosing rounds for employment against targets. 1 = cost is critical; 2 = weight is critical.

A series of mixes may be considered on a given computer run by stacking a series of these (12 thru 16) card sets at the end of the input deck.

TABLE 3.6 TARGET INPUT VARIABLES

Tape Line	Tape Columns	Variable Name	Input Units	Comments
1st	1-7	TNI(1)	Real	Target ID Number (e.g. 9016.0)
	8-14	(2)	0, 1, 2 or 3	Acquisition Key: 0 = Single Acq. 1 = 1st of Several Acq. 2 = Intermediate Acq. 3 = Last of Several Acq.
	15-21	(3)	1	Target frequency at Base Intensity (1 for all targets)
	22-28	(4)	0, 1 thru 11	Estimated posture mix for target main element.
	29-35	(5)	Real	Estimated fraction of target in wooded environment.
	36-42	(6)	Real	Estimated fraction of target in open environment.
	43-49	(7)	Meters	Estimated target radius.
	50-56	(8)	Decimal Hours	Estimated target Arrival Time.
	57-63	(9)	Decimal Hours	Estimated target Departure Time.
	64-70	(10)	Meters	Target Location Error
	71-77	(11)	KM	Estimated Target Easting (Location)
	78-84	(12)	KM	Estimated Target Northing (Location)
	85-91	(13)	KM	Target Distance from FEBA.
	92-98	(14)	0 thru 21	Estimated Military Worth of Target.
	99-105	(15)	1, 2, or 3	Defines echelon which acquired target 1=DS; 2=GS; 3=CORPS
	106-112	(16)	1 thru 70	Target type identifier: 1 - Artillery 2 - Mortars 3 - Anti-air 4 - Antitank 5 - Msl/Rocket Launchers

TABLE 3.6 TARGET INPUT VARIABLES (CONT'D)

Tape Line	Tape Columns	Variable Name	Input Units	Comments
1st Cont'd	106-112 Cont'd	TNI (16) Cont'd	1 thru 70	Target type identifier: 6 - APC's
				7 - Tanks
				8 - Command Post
				9 - Observation Post
				10 - Assembly Area
				11 - Engineer Units
				12 - Service Units
				13 - Aviation Units
				20 - H&I Mission
				30 - Illumination Mission
				40 - Preparation Fires
				50 - Counter-prep Fires
				60 - Smoke Mission
				70 - Final Protection Fires
	113-119	(17)	.5	Threshold "A" Attack Level (=.5 for all targets)
2nd	1-8	(18)	.5	Target Defeat Level (=.5 for all targets)
	9-16	(19)	0, 1, thru 11	Actual posture mix for target main element.
	17-24	(20)	Real	Actual fraction of target in wooded environment.
	25-32	(21)	Real	Actual fraction of target in open environment.
	33-40	(22)	Meters	Actual target radius.
	41-48	(23)	Decimal Hours	Actual target Arrival Time.
	49-56	(24)	Decimal Hours	Actual target Departure Time.
	57-64	(25)	Personnel	Number of Personnel in Target (including those inside vehicles)
	65-72	(26)	Tanks	Number of Tanks in Target.
	73-80	(27)	APC	Number of APC's in Target.
	81-88	(28)	Real	Initial fraction of personnel survivors.

TABLE 3.6 TARGET INPUT VARIABLES (CONT'D)

Tape Line	Tape Columns	Variable Name	Input Units	Comments
2nd Cont'd	89-96	(29)	Real	Initial fraction of tank survivors.
	97-104	(30)	Real	Initial fraction of APC survivors.
	105-112	(31)	0, 1, 2, 3	Target frequency at Low Intensity.
	113-120	(32)	0, 1, 2, 3	Target frequency at Mid Intensity.
	121-128	(33)	0, 1, 2, 3	Target frequency at High Intensity.
				For "Other" Type Missions (TNI(16)=20, 30, or 60) only:
3rd	1-8	AMSN(I,1)	Real	Target ID Number of I^{th} "other" mission.
	9-16	AMSN(I,2)	Real	Number of rounds 105mm system needs to fire I^{th} mission.
	17-24	AMSN(I,3)	Real	Number of rounds 155mm system needs to fire I^{th} mission.
	25-32	AMSN(I,4)	Real	Number of rounds 175mm system needs to fire I^{th} mission.
	33-40	AMSN(I,5)	Real	Number of rounds 203mm system needs to fire I^{th} mission.
	41-48	AMSN(I,6)	Real	Number of rounds MARS system needs to fire I^{th} mission.
	49-56	AMSN(I,7)	Real	Number of rounds LANCE system needs to fire I^{th} mission.
	57-64	AMSN(I,8)	Real	Number of rounds HJ system needs to fire I^{th} mission.

REFERENCES

1. Odom, C. T., Kramar, J. W., and Thomas, A.S., An Improved Model for Evaluating Artillery Weapons, BRL Report No. 1321, September 1966, Aberdeen Proving Ground, MD. UNCLASSIFIED.

2. Report, Optimum Mix of Artillery Units, 1971-1975 (U), Phase III, USACDC Field Artillery Agency, July 1967, Fort Sill, OK. SECRET

3. Report, Optimum Mix of Artillery Units, 1975-1980 (U), (Short Title: Legal Mix IV), USACDC Field Artillery Agency, August 1970, Fort Sill, OK. SECRET

4. Wood, William J., and Tice, Jerry, A Proposed Method for Determining Target Worth as an Input to Weapon Systems Analyses, AMSAA Technical Memorandum 15, October 1968, Aberdeen Proving Ground, MD. UNCLASSIFIED.

5. Odom, C. T., and Kramar, J. W., Michels, H. W., Thomas, A. S., and Thomas, C. M., Reoptimization of a Multiple Artillery Rocket System - MARS II (U), BRL Report No. 1736, September 1966, Aberdeen Proving Ground, MD. SECRET

6. Report, An Operational and Cost-Effectiveness Study of the LANCE Missile System (U), Vol III, USACDC/USAMC Study, April 1965. SECRET

The following page is blank.

APPENDIX A

PROGRAM LISTING

The following page is blank.

```
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     MAIN PROGRAM
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      COMMON W,W1,PII,CPET,RT,REF,EC1
      COMMON CXID(16),TZRO,TMX,FACT,DELT,POST(12,11),WKS
      COMMONSYSID(10),FRWM(10),TPFU(10),SROF(10),DROF(10),TBM(10),BLD(10
     1),RSPY(10),SNMX(10),DNMX(10),HNMX(10),STYP(10),HBLD(10),QBLD(10)
      COMMON RNDID(50),WGT(50),CST(50),RMX(50),REL(50),DEP(50),RTP(50),
     1WARN(50),WGTI(50),CSTI(50),R2MX(50),RELI(50),DEPI(50),
     2RNG(50,10),CPR(50,10),CPS(50,10),RE(50,10),AE(50,10),SOP(50,10),
     3POP(50,10),COP(50,10),TOP(50,10),AOP(50,10),SWP(50,10),
     4PWP(50,10),CWP(50,10),TWP(50,10),AWP(50,10)
      COMMON NSITE(100),FSID(100),TA(8,100),TD(8,100),XS(8,100),
     1YS(8,100),SYSRD(10,10),NRO(12),NRW(12),ORVP(12,12),WRVP(12,12)
      COMMON MIXID(16),KSIG(10),KRIG(50),KFIG(100),CRTERA,CRT(50)
      COMMON TN(30,300),TLOST(5,1000),TDFT(1000),A(27,100),FT(6,100),
     1S(10,5),RDCNT(50,30),TUBFU(100),NACQ,SPERS,STANK,SAPC,SMW,
     2NQ,NTGT,NOMF,NQLP,NRPD,NRW2,NQOM,NQD,NFM,NFMD
      COMMON TNI(33),TN28T,TN29T,TN30T
      COMMON REFIRE,WAIT,WAIT2,WAIT3,NA,NBA1,NBA2,NBA3,NEA1,NEA2,NEA3,
     1TR,EXCES1,EXCES2,AMSN(550,8)
      COMMON QUV(100),CPOST(5),CPK(10),CRE(10),CCOV(10),CSURV(10)
      COMMON AMWS,APC,ATLVL,CPER,CRIT,CRITP,CTI,DEFSP,DSFLAG,DVFLAG,F
      COMMON FH,FM,IFM,IFONE,ITC,KOUNT,LOSS,MATCH,MFEAT,MORG,NA1,NB,NE
      COMMON NFU,NOM,NP,NRDS,NSYS,ONCRT,PERO,PERW,PO,QN,RPV,SMCRT,SQN
      COMMON ST, SURV,SURVP,T,TK,TOEP,XVN,SAVE1,SAVE2,SAVE3
      CALL PRELIM
      CALL RDSYS
      CALL RDRND
      CALL RDFU
   77 CALL RDMIX
      CALL TZERO
    1 DN=0.0
      CALL RTAPE
      FR=TNI(3)
    5 IF(TNI(8).LT.T+DELT-.00001)GOTO 2
      IF(KOUNT.EQ.0)GOTO 40
      GOTO 3
    2 IF(TNI(16).EQ.20.0.OR.TNI(16).EQ.30.0.OR.TNI(16).EQ.60.) NOM=NOM+1
   78 NACQ=NACQ + 1
      IF(TNI(2) .GT. 1.) GOTO 20
      NTGT=NTGT + 1
      SPERS=SPERS+TNI(25)*TNI(28)
      STANK=STANK+TNI(26)*TNI(29)
      SAPC=SAPC+TNI(27)*TNI(30)
      AMWS=AMWS+TNI(14)
   20 CALL COMPAR
      IF(FR.EQ.1.0) GOTO 1
      FR=FR-1.0
       DN= DN+1.0
      TNI(1)=TNI(1)+100000.0* DN
      IF(TNI(16).NE.20.0.AND.TNI(16).NE.30.0.AND.TNI(16).NE.60.0)GOTO 78
      ITC=ITC+1
      AMSN(ITC,1)=AMSN(ITC-1,1)+100000.
      DO 31 KJ=2,8
   31 AMSN(ITC,KJ)=AMSN(ITC-1,KJ)
      GOTO 2
    3 KT=KOUNT
      DO 10 IT=1,KT
      IF(TN(17,IT).LE..5)GOTO 18
```

```
      IF(TN(9,IT) .GT. T+DELT-.00001) GOTO 10
   18 IF(TN(1,IT).EQ.0.0)GOTO 10
      IF(MFEAT.EQ.0)GOTO 17
      DO 8 JK=1,MFEAT
      IF(TN(1,IT).EQ.TDFT(JK)) GOTO 22
    8 CONTINUE
   17 CTI=1000000.0
      REFIRE=0.0
      WAIT=0.0
      WAIT2=0.0
      WAIT3=0.0
      NA=0
      DEFSP=0.0
      NBA1=0
      NBA2=0
      NBA3=0
      NEA1=0
      NEA2=0
      NEA3=0
      TR=0.0
      DSFLAG=0.0
      DVFLAG=0.0
      EXCES1=0.0
      EXCES2=0.0
      IF(TN(16,IT).EQ.20.0)GOTO 19
      IF(TN(16,IT).EQ.30.0)GOTO 19
      IF(TN(16,IT).EQ.60.0)GOTO 19
      NP = TN(4,IT) + 1.0
      DO 11 J=1,27
      DO 11 K=1,NFU
   11 A(J,K) = 0.0
      IF(TN(9,IT).LT.T) GOTO 23
      IF(LOSS.EQ.0)GOTO  7
      DO 9 JL=1,LOSS
      IF(TN(1,IT).EQ.TLOST(1,JL))GOTO 16
    9 CONTINUE
      GOTO  7
   16 TN(28,IT)=TLOST(3,JL)
      TN(29,IT)=TLOST(4,JL)
      TN(30,IT)=TLOST(5,JL)
      LOSS=LOSS-1
      IF(LOSS + 1 .EQ. JL) GOTO 7
      DO 32 J=1,5
      DO 32 K=JL,LOSS
   32 TLOST(J,K)=TLOST(J,K+1)
      GOTO 7
   19 CALL SPECIL(IT)
      IF(DEFSP.EQ.0.0)GOTO 6
      NOMF=NOMF + 1
   21 CALL REMOVE(IT)
      NFM=NFM + 1
      GOTO 18
   22 CALL REMOVE(IT)
      NRPD=NRPD + 1
      GOTO 18
    6 CALL REMOVE(IT)
      NQ=NQ + 1
      NQOM=NQOM + 1
      GOTO 18
   23 CALL REMOVE(IT)
```

```
      NQD=NQD + 1
      NQ=NQ + 1
      GOTO 18
    7 L=TN(15,IT)
      GOTO(13,14,15),L
   13 CALL DIRSUP(IT)
      IF(REFIRE.EQ.1.0)GOTO 18
      IF(TR.EQ.1.0)GOTO 21
      GOTO 10
   14 CALL DIVISN(IT)
      IF(REFIRE.EQ.1.0)GOTO 18
      IF(TR.EQ.1.0)GOTO 21
      GOTO 10
   15 CALL CORP(IT)
      IF(REFIRE.EQ.1.0)GOTO 18
      IF(TR.EQ.1.0)GOTO 21
   10 CONTINUE
   40 T=T+DELT
      DO 28 J=1,NFU
   28 IF(TUBFU(J).LT.T) TUBFU(J)=T
      DO 29 J=1,NFU
      DO 29 I=1,5
   29 FT(I,J)=FT(I+1,J)
      TOUT=TOUT+1.0
      IF(TOUT.NE.4.0)GOTO 4
      TOUT=0.0
      CALL OUTPUT
    4 IF(T.GE.TMX)GOTO 30
      GOTO 5
   30 REWIND 3
      GOTO 77
      END
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      SUBROUTINE PRELIM
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      COMMON (USE MAIN)
      READ(5,101) (CXID(I), I=1,16)
      WRITE(6,101)CXID
      READ(5,102)TZRO,TMX,FACT
      DELT=0.25
      DO 10 I=1,12
   10 READ (5,103)(POST(I,J),J=1,11)
      W=ALOG(2.0)
      PI=3.14159
      PII=1./PI
      WKS=-PII/ALOG(.7)
      W1=2.0*W
  999 RETURN
101   FORMAT(16A5)
102   FORMAT(10F7.4,10X)
  103 FORMAT(11F7.4)
      END
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      SUBROUTINE RDSYS
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      COMMON (USE MAIN)
      READ(5,101)NSYS
      WRITE(6,101)NSYS
      DO 1 I=1,NSYS
      READ(5,102)SYSID(I),FRWM(I),TPFU(I),SROF(I),DROF(I),TBM(I),
```

```
     1BLD(I),RSPY(I),SNMX(I),DNMX(I),HNMX(I),STYP(I)
      HBLD(I)=.5*BLD(I)
      QBLD(I)=.25*BLD(I)
      TBM(I)=TBM(I)/60.
      SROF(I)=SROF(I)*60.0
      DROF(I)=DROF(I)*60.0
      SNMX(I)=TPFU(I)*SNMX(I)
      DNMX(I)=TPFU(I)*DNMX(I)
    1 HNMX(I)=TPFU(I)*HNMX(I)
999   RETURN
101   FORMAT(16I5)
102   FORMAT(10F7.4,10X)
      END
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      SUBROUTINE RDRND
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      COMMON (USE MAIN)
      READ(5,101)NRDS
      WRITE(6,101)NRDS
      DO 4 I=1,NRDS
      READ(5,102)RNDID(I),WGT(I),CST(I),RMX(I),REL(I),DEP(I),RTP(I),WARN
     1(I)
      WGTI(I)=1./WGT(I)
      CSTI(I)=1./CST(I)
      R2MX(I)=RMX(I)**2
      RELI(I)=1./REL(I)
      DEPI(I)=1./DEP(I)
      READ(5,102)(RNG(I,J),J=1,10),(CPR(I,J),J=1,10),(CPS(I,J),J=1,10)
      IF(RTP(I).EQ.1.)GOTO 1
      CALL HEINP(I)
      GOTO 2
    1 CALL ACMINP(I)
    2 DO 3 J=1,10
    3 RNG(I,J)=RNG(I,J)**2
4     CONTINUE
999   RETURN
101   FORMAT(16I5)
102   FORMAT(10F7.4,10X)
      END
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      SUBROUTINE ACMINP(I)
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      DIMENSION AL(10)
      COMMON(USE MAIN)
      READ(5,101)SRE,REZ,SRO,SRW,EN
      SRO=SRO*EN
      SRW=SRW*EN
      DO 1 J=1,10
      RE(I,J)=REZ+RNG(I,J)*SRE
    1 AE(I,J) = PII / (RE(I,J) ** 2 )
      READ(5,101)(AL(K),K=1,10)
      DO 2 J=1,10
      PRO=-SRO * AE(I,J)
```

```
      PRW=-SRW * AE(I,J)
      SOP(I,J)=1.-EXP(PRO*AL(1))
      POP(I,J)=1.-EXP(PRO*AL(2))
      COP(I,J)=1.-EXP(PRO*AL(3))
      TOP(I,J)=1.-EXP(PRO*AL(4))
      AOP(I,J)=1.-EXP(PRO*AL(5))
      SWP(I,J)=1.-EXP(PRW*AL(6))
      PWP(I,J)=1.-EXP(PRW*AL(7))
      CWP(I,J)=1.-EXP(PRW*AL(8))
      TWP(I,J)=1.-EXP(PRW*AL(9))
    2 AWP(I,J)=1.-EXP(PRW*AL(10))
  999 RETURN
  101 FORMAT(10F7.4,10X)
      END
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      SUBROUTINE HEINP(I)
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      DIMENSION AL(100)
      COMMON(USE MAIN)
      READ(5,101)(AL(J),J=1,100)
      DO 1 J=1,10
      RE(I,J)=.3
      AE(I,J)=0.
      SOP(I,J)=SQRT(AL(J)*WKS )
      POP(I,J)=SQRT(AL(J+10)*WKS )
      COP(I,J)=SQRT(AL(J+20)*WKS )
      TOP(I,J)=SQRT(AL(J+30)*WKS )
      AOP(I,J)=SQRT(AL(J+40)*WKS )
      SWP(I,J)=SQRT(AL(J+50)*WKS )
      PWP(I,J)=SQRT(AL(J+60)*WKS )
      CWP(I,J)=SQRT(AL(J+70)*WKS )
      TWP(I,J)=SQRT(AL(J+80)*WKS )
    1 AWP(I,J)=SQRT(AL(J+90)*WKS)
  999 RETURN
  101 FORMAT(10F7.4,10X)
      END
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      SUBROUTINE RDFU
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      COMMON (USE MAIN)
      READ(5,101)NFU
      WRITE(6,101)NFU
      DO 1 I= 1, NFU
      READ(5,101)NSITE(I)
      NS=NSITE(I)
      READ(5,102)FSID(I)
      DO 1 J=1,NS
      READ(5,102)TA(J,I),TD(J,I),XS(J,I),YS(J,I)
      LTA = TA(J,I)
      XTA = LTA
      TA(J,I) = XTA + (TA(J,I) - XTA) / .6
      LTD = TD(J,I)
      XTD = LTD
    1 TD(J,I) = XTD + (TD(J,I) - XTD) / .6
      DO 3 I=1,NSYS
    3 READ(5 ,103)(SYSRD(I,J),J=1,10)
      DO 2 I= 1,12
      READ(5,101) NP,NRO(I),NRW(I)
      LK1=NRO(I)
      LK2=NRW(I)
```

```
      READ(5,103) (ORVP(I,J),J=1,LK1)                                   FU 025
    2 READ(5,103)(WRVP(I,J),J=1,LK2)                                    FU 026
999   RETURN                                                            FU 027
101   FORMAT(16I5)                                                      FU 028
102   FORMAT(4F8.4,48X)                                                 FU 029
103   FORMAT(11F7.4)                                                    FU 030
      END                                                               FU 031
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *  MIX001
      SUBROUTINE RDMIX                                                  MIX002
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *  MIX003
      COMMON (USE MAIN)                                                 MIX004
      WRITE(6,102)                                                      MIX005
      READ(5,101)(MIXID(I),I=1,16)                                      MIX006
      WRITE(6,101)MIXID                                                 MIX007
      READ(5,104) (KSIG(I),I=1,NSYS)                                    MIX008
      READ(5,104) (KRIG(I),I=1,NRDS)                                    MIX009
      READ(5,104) (KFIG(I),I=1,NFU)                                     MIX010
      READ(5,103)CRTERA                                                 MIX011
      ICRT=CRTERA                                                       MIX012
      WRITE(6,106) KSIG                                                 MIX013
      WRITE(6,109) KRIG                                                 MIX014
      WRITE(6,107) KFIG                                                 MIX015
      WRITE(6,108) ICRT                                                 MIX016
      GOTO(3,4),ICRT                                                    MIX017
    3 DO 5 I=1,NRDS                                                     MIX018
    5 CRT(I)=CST(I)                                                     MIX019
      GOTO 999                                                          MIX020
    4 DO 6 I=1,NRDS                                                     MIX021
    6 CRT(I)=WGT(I)                                                     MIX022
999   RETURN                                                            MIX023
101   FORMAT(16A5)                                                      MIX024
  102 FORMAT(1H1)                                                       MIX025
103   FORMAT(10F7.4,10X)                                                MIX026
104   FORMAT(80I1)                                                      MIX027
  106 FORMAT(22HOSYSTEMS IN THIS MIX =,10I4)                            MIX028
  107 FORMAT(/ 25HOFIRE UNITS IN THIS MIX =,40I2/, 55I2/,55I2)          MIX029
  108 FORMAT(/22HOALLOCATION CRITERIA =,I4/38H 1 IS LEAST COST *** 2 IS MIX030
     1LEAST WEIGHT)                                                     MIX031
  109 FORMAT(21HOROUNDS IN THIS MIX=, 50I2)                             MIX032
      END                                                               MIX033
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *  TZR001
      SUBROUTINE TZERO                                                  TZR002
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *  TZR003
      COMMON(USE MAIN)                                                  TZR004
      KOUNT=0$ MATCH=0$ MFEAT=0$ LOSS=0$ ITC=0                          TZR005
      DO 1 I=1,30                                                       TZR006
      DO 1 J=1,300                                                      TZR007
    1 TN(I,J)=0.0                                                       TZR008
      DO 3 I=1,5                                                        TZR009
      DO 3 J=1,1000                                                     TZR010
    3 TLOST(I,J)=0.0                                                    TZR011
      DO 5 I=1,1000                                                     TZR012
    5 TDFT(I)=0.0                                                       TZR013
      DO 8 I=1,6                                                        TZR014
      DO 8 J=1,NFU                                                      TZR015
    8 FT(I,J)=0.0                                                       TZR016
      DO 11 I=1,NSYS                                                    TZR017
      DO 11 J=1,5                                                       TZR018
   11 S(I,J)=0.0                                                        TZR019
      DO 12 I=1,NRDS                                                    TZR020
```

```
      DO 12 J=1,30                                                      TZR021
   12 RDCNT(I,J) = 0.                                                   TZR022
      AMWS=0.                                                           TZR023
      NACQ=0 $ SPERS=0.$ STANK=0. $ SAPC=0. $ NQ=0 $ SMW=0. $  NTGT=0   TZR024
      NOMF=0 $ NQLP=0 $ NRPD=0 $ NRW2=0 $ NQOM=0 $ NQD=0 $ NFM=0        TZR025
      NFMD=0 $ NOM=0                                                    TZR026
      T=TZRO                                                            TZR027
      DO  2 I=1,NFU                                                     TZR028
    2 TUBFU(I)=T                                                        TZR029
  999 RETURN                                                            TZR030
      END                                                               TZR031
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * TPE001
      SUBROUTINE RTAPE                                                  TPE002
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * TPE003
      COMMON(USE MAIN)                                                  TPE004
    1 READ(3,102) TNI                                                   TPE005
      IF( TNI(10) .GT. 3. * TNI(7) ) GOTO 1                             TPE006
      KFACT=FACT                                                        TPE007
      GOTO(20,21,22,23),KFACT                                           TPE008
   20 TNI(3)=TNI(31)                                                    TPE009
      GO TO 50                                                          TPE010
   21 TNI(3)=TNI(32)                                                    TPE011
      GO TO 50                                                          TPE012
   22 TNI(3)=1.0                                                        TPE013
      GO TO 50                                                          TPE014
   23 TNI(3)=TNI(33)                                                    TPE015
   50 IF(TNI(3).EQ.0.) GOTO 51                                          TPE016
      IF(TNI(16).EQ.20.0.OR.TNI(16).EQ.30.0.OR.TNI(16).EQ.60.0)GOTO 52  TPE017
      GOTO 999                                                          TPE018
   51 IF(TNI(16).EQ.20.0.OR.TNI(16).EQ.30.0.OR.TNI(16).EQ.60.0)GOTO 53  TPE019
      GOTO 1                                                            TPE020
   52 ITC=ITC+1                                                         TPE021
      READ(3,105)(AMSN(ITC,J),J=1,8)                                    TPE022
      GOTO 999                                                          TPE023
   53 ITC=ITC+1                                                         TPE024
      READ(3,105)(AMSN(ITC,J),J=1,8)                                    TPE025
      ITC=ITC-1                                                         TPE026
      GOTO 1                                                            TPE027
  999 RETURN                                                            TPE028
  102 FORMAT(F7.1,16F7.2,/16F8.2)                                       TPE029
  105 FORMAT(8F8.2)                                                     TPE030
      END                                                               TPE031
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * RMV001
      SUBROUTINE REMOVE(K)                                              RMV002
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * RMV003
      COMMON(USE MAIN)                                                  RMV004
      IKZ=KOUNT-1                                                       RMV005
      IF(IKZ + 1 .EQ. K) GOTO 2                                         RMV006
      DO 1 J=K,IKZ                                                      RMV007
      DO 1 I=1,30                                                       RMV008
    1 TN(I,J)=TN(I,J+1)                                                 RMV009
    2 TN(1,KOUNT)=0.0                                                   RMV010
      KOUNT=IKZ                                                         RMV011
  999 RETURN                                                            RMV012
      END                                                               RMV013
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * CMP001
      SUBROUTINE COMPAR                                                 CMP002
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * CMP003
      COMMON(USE MAIN)                                                  CMP004
      IF(TNI(2).LT.2.0) GOTO 10                                         CMP005
```

```
      IF(MFEAT.EQ.0) GOTO 11                                            CMP006
      DO 1 I=1,MFEAT                                                    CMP007
      IF(TNI(1).EQ.TDFT(I))GOTO 12                                      CMP008
    1 CONTINUE                                                          CMP009
      GOTO 11                                                           CMP010
   12 IF(TNI(2).EQ.2.0) GOTO 6                                          CMP011
      MFEAT=MFEAT-1                                                     CMP012
      IF(MFEAT + 1 .EQ. I) GOTO 6                                       CMP013
      DO 5 J=I,MFEAT                                                    CMP014
    5 TDFT(J)=TDFT(J+1)                                                 CMP015
    6 NRPD=NRPD + 1                                                     CMP016
      GOTO 999                                                          CMP017
   11 IF(LOSS.EQ.0) GOTO 15                                             CMP018
      DO 2 I=1,LOSS                                                     CMP019
      IF(TNI(1).EQ.TLOST(1,I)) GOTO 14                                  CMP020
    2 CONTINUE                                                          CMP021
      GOTO 15                                                           CMP022
   14 TNI(28)=TLOST(3,I)                                                CMP023
      TNI(29)=TLOST(4,I)                                                CMP024
      TNI(30)=TLOST(5,I)                                                CMP025
      LOSS=LOSS-1                                                       CMP026
      IF(LOSS + 1 .EQ. I) GOTO 15                                       CMP027
      DO 7 J=1,5                                                        CMP028
      DO 7 K=I,LOSS                                                     CMP029
    7 TLOST(J,K)=TLOST(J,K+1)                                           CMP030
   15 M=1                                                               CMP031
      IF(KOUNT.EQ.0) GOTO 16                                            CMP032
   21 DO 3 J=M,KOUNT                                                    CMP033
      IF(TNI(1).EQ.TN(1,J)) GOTO 17                                     CMP034
    3 CONTINUE                                                          CMP035
      GOTO 10                                                           CMP036
   19 M=J+1                                                             CMP037
      IF(M.GT.KOUNT)GOTO 10                                             CMP038
      GOTO 21                                                           CMP039
   17 TNI(28)=TN(28,J)                                                  CMP040
      TNI(29)=TN(29,J)                                                  CMP041
      TNI(30)=TN(30,J)                                                  CMP042
      D=(TNI(11)-TN(11,J))**2+(TNI(12)-TN(12,J))**2                     CMP043
      IF(D.GE.0.04)GOTO 19                                              CMP044
      NRW2=NRW2 + 1                                                     CMP045
      CALL REMOVE(J)                                                    CMP046
   10 IF(KOUNT.EQ.0)GOTO 16                                             CMP047
      DO 4 J=1,KOUNT                                                    CMP048
      IF(TNI(14).LT.TN(14,J)) GOTO 4                                    CMP049
      IF(TNI(14).GT.TN(14,J)) GOTO 18                                   CMP050
      IF(TNI(9).GE.TN(9,J)) GOTO 4                                      CMP051
   18 MATCH=J                                                           CMP052
      GOTO 20                                                           CMP053
    4 CONTINUE                                                          CMP054
   16 MATCH=KOUNT+1                                                     CMP055
   20 MAX=300                                                           CMP056
      IF(MATCH .LE. MAX) GOTO 8                                         CMP057
      IF(TNI(16).NE.20..AND.TNI(16).NE.30..AND.TNI(16).NE.60.) GOTO 25  CMP058
      NQ=NQ + 1                                                         CMP059
      NQOM=NQOM + 1                                                     CMP060
      GOTO999                                                           CMP061
   25 NQ=NQ + 1                                                         CMP062
      NQLP=NQLP + 1                                                     CMP063
      GOTO 999                                                          CMP064
    8 IF(KOUNT .LT. MAX) GOTO 27                                        CMP065
```

```
      IF(TN(16,KOUNT).NE.20..AND.TN(16,KOUNT).NE.30..AND.TN(16,KOUNT).
     1NE.60.)GOTO 26
      NQ=NQ + 1
      NQOM=NQOM + 1
      GOTO 13
   26 NQ=NQ + 1
      NQLP=NQLP + 1
   13 KOUNT = KOUNT - 1
   27 K = KOUNT-MATCH+1
      IF(K .EQ. 0) GOTO 9
      KI=KOUNT
      DO 22 J=1,K
      DO 23 I=1,30
   23 TN(I,KI+1)=TN(I,KI)
   22 KI=KI-1
    9 DO 24 I=1,30
   24 TN(I,MATCH)=TNI(I)
      KOUNT=KOUNT+1
  999 RETURN
      END
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      SUBROUTINE DIRSUP(IT)
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      COMMON(USE MAIN)
      IDS1=0
      MORG=1
      CALL AMASS(IT)
      IF(NA.EQ.0) GOTO 1
      RANGE=1000000.
      DO 50 IDS=1,NA
      IF(A(9,IDS).LE.RANGE)GOTO 3
      GOTO 50
    3 RANGE=A(9,IDS)
      IDS1=IDS
   50 CONTINUE
      IF(IDS1.EQ.0)GOTO 1
      IR=A(24,IDS1)
      IF(A(13,IDS1).GT.A(4,IDS1))GOTO 1
      XVN=A(13,IDS1)
      IS=A(1,IDS1)
      CONSTR=15.0
      IF(TN(14,IT).GE.10.0)CONSTR=30.0
      WAIT3=XVN*WGT(IR)
      IF(WAIT3.GT.CONSTR)GOTO 1
      IF(TN(24,IT).LT.T) GOTO 4
      IF(TN(4,IT).EQ.TN(19,IT).AND.TN(5,IT).EQ.TN(20,IT).AND.TN(6,IT).EQ
     1.TN(21,IT).AND.TN(7,IT).EQ.TN(22,IT))GOTO 4
      NP =  TN(19,IT)  + 1.0
      RT=TN(22,IT)
      PERW=TN(20,IT)
      PERO=TN(21,IT)
      RPV=TPFU(IS) * A(5,IDS1)
      ATLVL=0.0
      CPER=0.0
      CPET=0.0
      CALL INTERP(IR,IDS1)
      CALL EFFECT(IDS1,IR,IT)
    4 IXZ=A(27,IDS1)
      FT(6,IXZ )=FT(6,IXZ )+A(13,IDS1)
      A9=SQRT(A(9,IDS1))
```

```
      IRNG=A9
      RNGINT=IRNG
      RNGINT=RNGINT+.5
      IF(A9        .GE.RNGINT)IRNG=IRNG+1
      IF(A9        .GE.30.0)IRNG=30
      RDCNT(IR,IRNG)=RDCNT(IR,IRNG)+A(13,IDS1)
      TUBFU(IXZ)=TUBFU(IXZ)+TBM(IS)*A(5,IDS1)
      IF(STYP(IS).NE.2.0)GOTO 6
      GOTO 7
    6 B=(A(13,IDS1)-A(5,IDS1)*TPFU(IS))/(A(5,IDS1)*TPFU(IS)*A(3,IDS1))
      IF(TN(10,IT) .EQ. 0.) B=B+.067
      B=AMAX1(0.0,B)
      TUBFU(IXZ)=TUBFU(IXZ)+B
    7 IF(TN(24,IT).GE.T) GOTO 9
      A(19,IDS1)= 0.
      A(20,IDS1)= 0.
      A(21,IDS1)= 0.
      A(22,IDS1)= 1.
      A(23,IDS1)= 1.
    9 ASUM=A(19,IDS1)+A(20,IDS1)+A(21,IDS1)
      TOEP=TN(25,IT)
      IF(NP.LT.6.OR.NP.GT.11)GOTO 10
      TOE=TN(26,IT)*4.0
      IF(NP.EQ.6.OR.NP.EQ.7)TOE=TN(27,IT)*15.0
      TOEP=AMAX1(0.0,TN(25,IT)-TOE)
      IF(ASUM.EQ.0.0)ASUM=1.0
   10 S(IS,1)=S(IS,1)+A(13,IDS1)*WGT(IR)
      S(IS,2)=S(IS,2)+A(13,IDS1)*CST(IR)
      S(IS,3)=S(IS,3)+(1.0-ASUM)*TN(28,IT)*TOEP
      S(IS,4)=S(IS,4)+(1.0-A(22,IDS1))*TN(29,IT)*TN(26,IT)
      S(IS,5)=S(IS,5)+(1.0-A(23,IDS1))*TN(30,IT)*TN(27,IT)
      SAVE1=TN(28,IT)
      SAVE2=TN(29,IT)
      SAVE3=TN(30,IT)
      TN(28,IT)=TN(28,IT)*(ASUM)
      TN(29,IT)=TN(29,IT)*A(22,IDS1)
      TN(30,IT)=TN(30,IT)*A(23,IDS1)
      GOTO(11,11,11,11,11,12,12,13,13,13,13,11),NP
   11 SMW=SMW + (SAVE1 - TN(28,IT)) * 2.0 * TN(14,IT)
      IF(TN(28,IT).LE.TN(18,IT))GOTO 8
      GOTO 14
   12 SMW=SMW + (SAVE3 - TN(30,IT)) * 2.0 * TN(14,IT)
      IF(TN(30,IT).LE.TN(18,IT))GOTO 8
      GOTO 14
   13 SMW=SMW + (SAVE2 - TN(29,IT)) * 2.0 * TN(14,IT)
      IF(TN(29,IT).LE.TN(18,IT))GOTO 8
   14 LOSS = LOSS + 1
      TLOST(1,LOSS)=TN(1,IT)
      TLOST(2,LOSS)=TN(3,IT)
      TLOST(3,LOSS)=TN(28,IT)
      TLOST(4,LOSS)=TN(29,IT)
      TLOST(5,LOSS)=TN(30,IT)
      TR=1.0
      GOTO 999
    8 MFEAT= MFEAT + 1
      TDFT(MFEAT)  =TN(1,IT)
      NFMD=NFMD + 1
      TR=1.0
      GOTO 999
    1 NEA1=NA
```

```
      NBA1=MIN0(1,NA)                                                   DS 101
      DSFLAG=1.0                                                        DS 102
      CALL DIVISN(IT)                                                   DS 103
  999 RETURN                                                            DS 104
      END                                                               DS 105
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * INT001
      SUBROUTINE INTERP(IR,IA)                                          INT002
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * INT003
      COMMON(USE MAIN)                                                  INT004
      RG=A(9,IA)                                                        INT005
      DO 10 K=1,10                                                      INT006
      IF(RG-RNG(IR,K))1,2,10                                            INT007
   10 CONTINUE                                                          INT008
    1 KS= K - 1                                                         INT009
      D1 = RG - RNG(IR,KS)                                              INT010
      D2=RNG(IR,K)-RNG(IR,KS)                                           INT011
      RA=D1/D2                                                          INT012
      CPER=RA*(CPR(IR,K)-CPR(IR,KS))+CPR(IR,KS)                         INT013
      CPET=RA*(CPS(IR,K)-CPS(IR,KS))+CPS(IR,KS)                         INT014
      CPK(1)=RA*(SOP(IR,K)-SOP(IR,KS))+SOP(IR,KS)                       INT015
      CPK(2)=RA*(POP(IR,K)-POP(IR,KS))+POP(IR,KS)                       INT016
      CPK(3)=RA*(COP(IR,K)-COP(IR,KS))+COP(IR,KS)                       INT017
      CPK(4)=RA*(TOP(IR,K)-TOP(IR,KS))+TOP(IR,KS)                       INT018
      CPK(5)=RA*(AOP(IR,K)-AOP(IR,KS))+AOP(IR,KS)                       INT019
      CPK(6)=RA*(SWP(IR,K)-SWP(IR,KS))+SWP(IR,KS)                       INT020
      CPK(7)=RA*(PWP(IR,K)-PWP(IR,KS))+PWP(IR,KS)                       INT021
      CPK(8)=RA*(CWP(IR,K)-CWP(IR,KS))+CWP(IR,KS)                       INT022
      CPK(9)=RA*(TWP(IR,K)-TWP(IR,KS))+TWP(IR,KS)                       INT023
      CPK(10)=RA*(AWP(IR,K)-AWP(IR,KS))+AWP(IR,KS)                      INT024
      REI=RA*(RE(IR,K)-RE(IR,KS))+RE(IR,KS)                             INT025
      DO 14 L= 1,10                                                     INT026
   14 CRE(L)=REI                                                        INT027
      GOTO 15                                                           INT028
    2 CPER=CPR(IR,K)                                                    INT029
      CPET=CPS(IR,K)                                                    INT030
      CPK(1)=SOP(IR,K)                                                  INT031
      CPK(2)=POP(IR,K)                                                  INT032
      CPK(3)=COP(IR,K)                                                  INT033
      CPK(4)=TOP(IR,K)                                                  INT034
      CPK(5)=AOP(IR,K)                                                  INT035
      CPK(6)=SWP(IR,K)                                                  INT036
      CPK(7)=PWP(IR,K)                                                  INT037
      CPK(8)=CWP(IR,K)                                                  INT038
      CPK(9)=TWP(IR,K)                                                  INT039
      CPK(10)= AWP(IR,K)                                                INT040
      DO 12 L=1,10                                                      INT041
   12 CRE(L)=RE(IR,K)                                                   INT042
   15 IF(RTP(IR).EQ.1.0)GOTO 999                                        INT043
      DO 13 L=1,10                                                      INT044
      CRE(L)=CPK(L)                                                     INT045
   13 CPK(L)=.3                                                         INT046
  999 RETURN                                                            INT047
      END                                                               INT048
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * AMS001
      SUBROUTINE AMASS(IT)                                              AMS002
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * AMS003
      DIMENSION TEMPA(12)                                               AMS004
      COMMON (USE MAIN)                                                 AMS005
      DO 10 JF=1,NFU                                                    AMS006
      IF( KFIG(JF).NE.MORG)GOTO 10                                      AMS007
```

```
      IF(TUBFU(JF).GE.(T+DELT-.00001)) GOTO 10
      DO 11 IS=1,NSYS
      IF(KSIG(IS).EQ.0) GOTO 11
      IF(FSID(JF).NE.SYSID(IS))GOTO 11
      GOTO 1
   11 CONTINUE
      WRITE(6,101)FSID(JF),MORG
  101 FORMAT(12H FIRE SYSTEM,F8.2,25HIS NOT IN LIST OF SYSTEMS,9HECHELON
     1 =,I5)
      STOP
    1 NA=NA+1
      A(27,NA)=JF
      A(1,NA)=IS
      A(2,NA)=TUBFU(JF)
      R=HBLD(IS)+RSPY(IS)*(T-TZRO)
      Z=R-FT(6,JF)
      IF(TN(14,IT).GE.10.0)Z=AMIN1(R+QBLD(IS),24.0*RSPY(IS)+HBLD(IS))-
     1FT(6,JF)
      R=HNMX(IS)-FT(6,JF)+FT(2,JF)
      R=AMIN1(R,Z)
      NS=NSITE(JF)
      DO 19 INS=1,NS
      IF(T.LT.TA(INS,JF))GOTO 2
      IF(TA(INS,JF).LE.T.AND.T.LE.TD(INS,JF))GOTO 3
   19 CONTINUE
      WRITE(6,100)JF
  100 FORMAT(16H ERROR FIRE UNIT,I3,39HDOES NOT HAVE A TIME TO MATCH GAM
     1E TIME)
      STOP
    2 A(5,NA)=FRWM(IS)
      GOTO 4
    3 A(5,NA)=1.0
    4 A(6,NA)=XS(INS,JF)
      A(7,NA)=YS(INS,JF)
      IF(TN(8,IT).EQ.TN(9,IT))GOTO 5
      Z=SNMX(IS)*A(5,NA)
      A(3,NA)=SROF(IS)
      GOTO 6
    5 Z=DNMX(IS)*A(5,NA)
      A(3,NA)=DROF(IS)
    6 R=AMIN1(R,Z)
      A(4,NA)=R
      A(8,NA)=MORG
      A(9,NA)=(A(6,NA)-TN(11,IT))**2+(A(7,NA)-TN(12,IT))**2
      IF(A(4,NA).LE.0.0)GOTO 24
      CTMIN=CTI
      DO 13 LSYS=1,NSYS
      IF(SYSID(IS).EQ.SYSRD(LSYS,1))GOTO 7
   13 CONTINUE
      WRITE(6,102)SYSID(IS)
  102 FORMAT(29H ERROR, UNDEFINED SYSTEM. ID=,F10.2)
      STOP
    7 DO 12 IR=1,NRDS
      IF(KRIG(IR).EQ.0)GOTO 12
      DO 14 M=2,10
      IF(RNDID(IR).EQ.SYSRD(LSYS,M))GOTO 8
   14 CONTINUE
      GOTO 12
    8 IF(A(9,NA).GT.R2MX(IR))GOTO 12
      IF(TN(6,IT).EQ.0.0)GOTO 9
```

```
      LK1=NRO(NP)                                                       AMS068
      DO 15 L=1,LK1                                                     AMS069
      IF(RNDID(IR).EQ.ORVP(NP,L))GOTO 20                                AMS070
   15 CONTINUE                                                          AMS071
    9 IF(TN(5,IT).EQ.0.0)GOTO 12                                        AMS072
      LK2=NRW(NP)                                                       AMS073
      DO 16 L=1,LK2                                                     AMS074
      IF(RNDID(IR).EQ.WRVP(NP,L))GOTO 20                                AMS075
   16 CONTINUE                                                          AMS076
      GOTO 12                                                           AMS077
   20 CPER=0.0                                                          AMS078
      CPET=0.0                                                          AMS079
      CALL INTERP(IR,NA)                                                AMS080
      A(10,NA)=CPER                                                     AMS081
      A(11,NA)=CPET                                                     AMS082
      XVN=0.0                                                           AMS083
      ATLVL=TN(17,IT)                                                   AMS084
      RT=TN(7,IT)                                                       AMS085
      PERW=TN(5,IT)                                                     AMS086
      PERO=TN(6,IT)                                                     AMS087
      RPV=TPFU(IS)*A(5,NA)                                              AMS088
      CALL EFFECT(NA,IR,IT)                                             AMS089
      IF(CRTERA.EQ.2.0)GOTO 22                                          AMS090
      CT=A(13,NA)*CST(IR)                                               AMS091
      CT1V=A(5,NA)*TPFU(IS)*CST(IR)                                     AMS092
      GOTO 23                                                           AMS093
   22 CT=A(13,NA)*WGT(IR)                                               AMS094
      CT1V=A(5,NA)*TPFU(IS)*WGT(IR)                                     AMS095
   23 IF(CT.GE.CTMIN)GOTO 12                                            AMS096
      CTMIN=CT                                                          AMS097
      A(24,NA)=IR                                                       AMS098
      A(25,NA)=CT                                                       AMS099
      A(26,NA)=CT1V                                                     AMS100
      DO 25 I=1,12                                                      AMS101
   25 TEMPA(I)=A(I+11,NA)                                               AMS102
   12 CONTINUE                                                          AMS103
      IF(CTMIN.GE.CTI)GOTO 24                                           AMS104
      DO 26 I=1,12                                                      AMS105
   26 A(I+11,NA)=TEMPA(I)                                               AMS106
      GOTO 10                                                           AMS107
   24 NA = NA - 1                                                       AMS108
   10 CONTINUE                                                          AMS109
  999 RETURN                                                            AMS110
      END                                                               AMS111
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * DIV001
      SUBROUTINE DIVISN(IT)                                             DIV002
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * DIV003
      COMMON(USE MAIN)                                                  DIV004
      SAVE1=TN(28,IT)                                                   DIV005
      SAVE2=TN(29,IT)                                                   DIV006
      SAVE3=TN(30,IT)                                                   DIV007
      MORG=2                                                            DIV008
      CALL AMASS(IT)                                                    DIV009
      NEA2=NA                                                           DIV010
      NBA2= MIN0(NA,NEA1+1)                                             DIV011
      NB=NBA2$NE=NEA2$ST=1.$PO=1.$FH=1.$TK=1.$APC=1.$SURVP=1.$SURV=1.$  DIV012
      CRITP=0.$ CRIT= 0.                                                DIV013
      TN28T=TN(28,IT)*TN(25,IT)                                         DIV014
      IF(NP.LT.6.OR.NP.GT.11)GOTO 19                                    DIV015
      TOE=TN(26,IT)*4.0                                                 DIV016
```

```
      IF(NP.EQ.6.OR.NP.EQ.7)TOE=TN(27,IT)*15.0                          DIV017
      TN28T=(AMAX1(0.0,TN(25,IT)-TOE))*TN(28,IT)                        DIV018
   19 TN29T=TN(29,IT)*TN(26,IT)                                         DIV019
      TN30T=TN(30,IT)*TN(27,IT)                                         DIV020
      DO 310 IQ=1,NFU                                                   DIV021
  310 QUV(IQ)=0.0                                                       DIV022
      SQN=0. $ SMCRT= 0.                                                DIV023
      QN=1000000.0  $ ONCRT=0.0                                         DIV024
      JF = 0                                                            DIV025
      IF(NBA2.LE.NEA1) GOTO 35                                          DIV026
      CALL ONEVOL(IT)                                                   DIV027
      QN = A(13,NB)                                                     DIV028
      NA1 = NB                                                          DIV029
      JF = NA1                                                          DIV030
      CALL MULVOL(IT)                                                   DIV031
      IF(SURV.GT.TN(17,IT))GOTO 32                                      DIV032
      IF(SQN.LT.QN) GOTO 33                                             DIV033
      IF(ONCRT.LT.SMCRT) GOTO 33                                        DIV034
      GOTO 34                                                           DIV035
   32 IF (SQN.LT.QN) GOTO 35                                            DIV036
   34 CALL SHMUVL(IT,JF)                                                DIV037
      GOTO 999                                                          DIV038
   33 ST=1.0 $ PO=1.0 $ FH=1.0 $ TK=1.0 $ APC=1.0                       DIV039
      CALL SHONVL(IT)                                                   DIV040
      GOTO 999                                                          DIV041
   35 IF(EXCES1.EQ.1. .AND. EXCES2.EQ.1.) GOTO 3                        DIV042
      IF(DSFLAG.EQ.1.0) GOTO 1                                          DIV043
      MORG=1                                                            DIV044
      CALL AMASS(IT)                                                    DIV045
      NEA1=NA                                                           DIV046
      NBA1=MIN0(NA,NEA2+1)                                              DIV047
      IF(NBA1.LE.NEA2)GOTO 3                                            DIV048
      GOTO 2                                                            DIV049
    1 IF(NEA1.EQ.0) GOTO 3                                              DIV050
    2 NB=NBA1                                                           DIV051
      NE=NEA1                                                           DIV052
      CALL ONEVOL(IT)                                                   DIV053
      IF(QN.EQ.1000000.0.OR.NBA2.EQ.0) GOTO 23                          DIV054
      IF(DSFLAG.EQ.1.0.AND.NBA2.EQ.NEA1) GOTO 23                        DIV055
      GOTO 8                                                            DIV056
   23 QN = A(13,NB)                                                     DIV057
    8 NA1=NB                                                            DIV058
      JF = NA1                                                          DIV059
      CALL MULVOL(IT)                                                   DIV060
      IF(SURV.GT.TN(17,IT)) GOTO 4                                      DIV061
      IF(SQN.LT.QN) GOTO 5                                              DIV062
      IF(ONCRT.LT.SMCRT) GOTO 5                                         DIV063
      GOTO 6                                                            DIV064
    4 IF(SQN.LT.QN) GOTO 3                                              DIV065
    6 IR=A(24,JF)                                                       DIV066
      NP= TN(19,IT) + 1.0                                               DIV067
      IF(TN(24,IT).LT.T) GOTO 15                                        DIV068
      IF(TN(4,IT).EQ.TN(19,IT).AND.TN(5,IT).EQ.TN(20,IT).AND.TN(6,IT)   DIV069
     1.EQ.TN(21,IT).AND.TN(7,IT).EQ.TN(22,IT)) GOTO 15                  DIV070
      CALL INTERP(IR,JF)                                                DIV071
      RT=TN(22,IT)                                                      DIV072
      PERW=TN(20,IT)                                                    DIV073
      PERO=TN(21,IT)                                                    DIV074
      ATLVL=0.0                                                         DIV075
      XVN=A(13,JF)                                                      DIV076
```

```
      IK=A(1,JF)
      RPV=TPFU(IK)*A(5,JF)
      CALL EFFECT(JF,IR,IT)
   15 TN28=TN(28,IT)
      TN29=TN(29,IT)
      TN30=TN(30,IT)
      IF(TN(24,IT).GE.1) GOTO 13
      A(19,JF)= 0.
      A(20,JF)= 0.
      A(21,JF)= 0.
      A(22,JF)= 1.
      A(23,JF)= 1.
   13 ASUM=A(19,JF)+A(20,JF)+A(21,JF)
      TOEP=TN(25,IT)
      IF(NP.LT.6.OR.NP.GT.11)GOTO 18
      TOE=TN(26,IT)*4.0
      IF(NP.EQ.6.OR.NP.EQ.7)TOE=TN(27,IT)*15.0
      TOEP=AMAX1(0.0,TN(25,IT)-TOE)
      IF(ASUM.EQ.0.0)ASUM=1.0
   18 TN(28,IT)=TN(28,IT)*ASUM
      TN(29,IT)=TN(29,IT)*A(22,JF)
      TN(30,IT)=TN(30,IT)*A(23,JF)
      IF(NBA2.EQ.0)GOTO 9
      IF(DSFLAG.EQ.1.0.AND.NBA2.EQ.NEA1) GOTO 9
      DO 10 IF3=NBA2,NEA2
      IF(A(4,IF3).LE.0.0)GOTO 10
      IS=A(1,IF3)
      IR=A(24,IF3)
      IXZ=A(27,IF3)
      FT(6,IXZ)=FT(6,IXZ)+A(4,IF3)
      A9=SQRT(A(9,IF3))
      IRNG=A9
      RNGINT=IRNG
      RNGINT=RNGINT+.5
      IF(A9       .GE.RNGINT)IRNG=IRNG+1
      IF(A9       .GE.30.0)IRNG=30
      RDCNT(IR,IRNG)=RDCNT(IR,IRNG)+A(4,IF3)
      S(IS,1)=S(IS,1)+A(4,IF3)*WGT(IR)
      S(IS,2)=S(IS,2)+A(4,IF3)*CST(IR)
      S(IS,3)=S(IS,3)+(1.0-ASUM)*TN28*TOEP*QUV(IF3)*A(4,IF3)/A(13,JF)
      S(IS,4)=S(IS,4)+(1.0-A(22,JF))*TN29*TN(26,IT)*QUV(IF3)*A(4,IF3)/A(
     113,JF)
      S(IS,5)=S(IS,5)+(1.0-A(23,JF))*TN30*TN(27,IT)*QUV(IF3)*A(4,IF3)/A(
     113,JF)
      TUBFU(IXZ)=TUBFU(IXZ)+TBM(IS)*A(5,IF3)
      IF(STYP(IS).EQ.2.0)GOTO 10
      B=(A(4,IF3)-A(5,IF3)*TPFU(IS))/(A(5,IF3)*TPFU(IS)*A(3,IF3))
      IF(TN(10,IT) .EQ. 0.) B=B+.067
      B=AMAX1(0.0,B)
      TUBFU(IXZ)=TUBFU(IXZ)+B
   10 CONTINUE
    9 IF(IFM.LT.NB) GOTO 14
      DO 11 IF3=NB,IFM
      IF(A(4,IF3).LE.0.0)GOTO 11
      IS=A(1,IF3)
      IR=A(24,IF3)
      IXZ=A(27,IF3)
      FT(6,IXZ)=FT(6,IXZ)+A(4,IF3)
      A9=SQRT(A(9,IF3))
      IRNG=A9
```

```
      RNGINT=IRNG
      RNGINT=RNGINT+.5
      IF(A9       .GE.RNGINT)IRNG=IRNG+1
      IF(A9       .GE.30.0)IRNG=30
      RDCNT(IR,IRNG)=RDCNT(IR,IRNG)+A(4,IF3)
      S(IS,1)=S(IS,1)+A(4,IF3)*WGT(IR)
      S(IS,2)=S(IS,2)+A(4,IF3)*CST(IR)
      S(IS,3)=S(IS,3)+(1.0-ASUM)*TN28*TOEP*QUV(IF3)*A(4,IF3)/A(13,JF)
      S(IS,4)=S(IS,4)+(1.0-A(22,JF))*TN29*TN(26,IT)*QUV(IF3)*A(4,IF3)/A(
     113,JF)
      S(IS,5)=S(IS,5)+(1.0-A(23,JF))*TN30*TN(27,IT)*QUV(IF3)*A(4,IF3)/A(
     113,JF)
      TUBFU(IXZ)=TUBFU(IXZ)+TBM(IS)*A(5,IF3)
      IF(STYP(IS).EQ.2.0)GOTO 11
      B=(A(4,IF3)-A(5,IF3)*TPFU(IS))/(A(5,IF3)*TPFU(IS)*A(3,IF3))
      IF(TN(10,IT) .EQ. 0.) B=B+.067
      B=AMAX1(0.0,B)
      TUBFU(IXZ)=TUBFU(IXZ)+B
   11 CONTINUE
   14 IF3=IFM+1
      IS=A(1,IF3)
      IR=A(24,IF3)
      IXZ=A(27,IF3)
      FT(6,IXZ)=FT(6,IXZ)+A(4,IF3)*FM
      A9=SQRT(A(9,IF3))
      IRNG=A9
      RNGINT=IRNG
      RNGINT=RNGINT+.5
      IF(A9       .GE.RNGINT)IRNG=IRNG+1
      IF(A9       .GE.30.0)IRNG=30
      RDCNT(IR,IRNG)=RDCNT(IR,IRNG)+A(4,IF3)*FM
      S(IS,1)=S(IS,1)+A(4,IF3)*WGT(IR)*FM
      S(IS,2)=S(IS,2)+A(4,IF3)*CST(IR)*FM
      S(IS,3)=S(IS,3)+(1.0-ASUM)*TN28*TOEP*QUV(IF3)*A(4,IF3)*FM/A(13,JF)
      S(IS,4)=S(IS,4)+(1.0-A(22,JF))*TN29*TN(26,IT)*QUV(IF3)*A(4,IF3)*FM
     1/A(13,JF)
      S(IS,5)=S(IS,5)+(1.0-A(23,JF))*TN30*TN(27,IT)*QUV(IF3)*A(4,IF3)*FM
     1/A(13,JF)
      TUBFU(IXZ)=TUBFU(IXZ)+TBM(IS)*A(5,IF3)*FM
      IF(STYP(IS).EQ.2.0)GOTO 17
      B=(A(4,IF3)*FM-A(5,IF3)*TPFU(IS))/(A(5,IF3)*TPFU(IS)*A(3,IF3))
      IF(TN(10,IT) .EQ. 0.) B=B+.067
      B=AMAX1(0.0,B)
      TUBFU(IXZ)=TUBFU(IXZ)+B
   17 GOTO(38,38,38,38,38,39,39,40,40,40,40,38),NP
   38 SMW=SMW + (SAVE1 - TN(28,IT)) * 2.0 * TN(14,IT)
      IF(TN(28,IT).LE.TN(18,IT))GOTO 7
      GOTO 41
   39 SMW=SMW + (SAVE3 - TN(30,IT)) * 2.0 * TN(14,IT)
      IF(TN(30,IT).LE.TN(18,IT))GOTO 7
      GOTO 41
   40 SMW=SMW + (SAVE2 - TN(29,IT)) * 2.0 * TN(14,IT)
      IF(TN(29,IT).LE.TN(18,IT))GOTO 7
   41 LOSS = LOSS + 1
      TLOST(1,LOSS)=TN(1,IT)
      TLOST(2,LOSS)=TN(3,IT)
      TLOST(3,LOSS)=TN(28,IT)
      TLOST(4,LOSS)=TN(29,IT)
      TLOST(5,LOSS)=TN(30,IT)
      TR=1.0
```

```
      GOTO 999
    7 MFEAT= MFEAT + 1
      TDFT(MFEAT)  =TN(1,IT)
      NFMD=NFMD + 1
      TR=1.0
      GOTO 999
    5 ST=1.0 $ PO=1.0 $ FH=1.0 $ TK=1.0 $ APC=1.0
      RT=TN(22,IT)
      NP= TN(19,IT) + 1.0
      PERW=TN(20,IT)
      PERO=TN(21,IT)
      IF(NBA2.EQ.0)GOTO 26
      IF(DSFLAG.EQ.1.0.AND.NBA2.EQ.NEA1) GOTO 26
      DO 12 IF4=NBA2,NEA2
      IS=A(1,IF4)
      TEMP=A(5,IF4)*TPFU(IS)
      IF(A(4,IF4).LT.TEMP)GOTO 12
      IR=A(24,IF4)
      IF(TN(24,IT).LT.T) GOTO 16
      IF(TN(4,IT).EQ.TN(19,IT).AND.TN(5,IT).EQ.TN(20,IT).AND.TN(6,IT)
     1.EQ.TN(21,IT).AND.TN(7,IT).EQ.TN(22,IT)) GOTO 16
      CALL INTERP(IR,IF4)
      ATLVL=0.0
      XVN=TEMP
      RPV=TEMP
      CALL EFFECT(IF4,IR,IT)
   16 ST=ST*A(14,IF4)
      PO=PO*A(15,IF4)
      FH=FH*A(16,IF4)
      TK=TK*A(17,IF4)
      APC=APC*A(18,IF4)
      IF(TN(24,IT).GE.T) GOTO 20
      A(19,IF4)= 0.
      A(20,IF4)= 0.
      A(21,IF4)= 0.
      A(22,IF4)= 1.
      A(23,IF4)= 1.
   20 ASUM=A(19,IF4)+A(20,IF4)+A(21,IF4)
      IF(ASUM.EQ.0.0)ASUM=1.0
      A9=SQRT(A(9,IF4))
      IRNG=A9
      RNGINT=IRNG
      RNGINT=RNGINT+.5
      IF(A9       .GE.RNGINT)IRNG=IRNG+1
      IF(A9       .GE.30.0)IRNG=30
      RDCNT(IR,IRNG)=RDCNT(IR,IRNG)+TEMP
      IXZ=A(27,IF4)
      FT(6,IXZ)=FT(6,IXZ)+TEMP
      S(IS,1)=S(IS,1)+TEMP*WGT(IR)
      S(IS,2)=S(IS,2)+TEMP*CST(IR)
      S(IS,3)=S(IS,3)+(1.0-ASUM)*TN28T
      S(IS,4)=S(IS,4)+(1.0-A(22,IF4))*TN29T
      S(IS,5)=S(IS,5)+(1.0-A(23,IF4))*TN30T
      TN28T=TN28T-TN28T*(1.0-ASUM)
      TN29T=TN29T-TN29T*(1.0-A(22,IF4))
      TN30T=TN30T-TN30T*(1.0-A(23,IF4))
      TUBFU(IXZ)=TUBFU(IXZ)+TBM(IS)*A(5,IF4)
      IF(TN(10,IT).EQ.0..AND.STYP(IS).EQ.1.)TUBFU(IXZ)=TUBFU(IXZ)+.067
   12 CONTINUE
   26 CALL SHONVL(IT)
```

```
      GOTO 999
    3 DVFLAG=1.0
      CALL CORP(IT)
  999 RETURN
      END
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      SUBROUTINE SHMUVL(IT,JF)
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      COMMON(USE MAIN)
      IR=A(24,JF)
      NP =  TN(19,IT)  + 1.0
      IF(TN(24,IT).LT.T) GOTO 4
      IF(TN(4,IT).EQ.TN(19,IT).AND.TN(5,IT).EQ.TN(20,IT).AND.TN(6,IT)
     1.EQ.TN(21,IT).AND.TN(7,IT).EQ.TN(22,IT)) GOTO 4
      CALL INTERP(IR,JF)
      RT=TN(22,IT)
      PERW=TN(20,IT)
      PERO=TN(21,IT)
      ATLVL=0.0
      IS=A(1,JF)
      RPV=A(5,JF)*TPFU(IS)
      XVN=A(13,JF)
      CALL EFFECT(JF,IR,IT)
    4 TN28=TN(28,IT)
      TN29=TN(29,IT)
      TN30=TN(30,IT)
      IF(TN(24,IT).GE.T) GOTO 3
      A(19,JF)= 0.
      A(20,JF)= 0.
      A(21,JF)= 0.
      A(22,JF)= 1.
      A(23,JF)= 1.
    3 ASUM=A(19,JF)+A(20,JF)+A(21,JF)
      TOEP=TN(25,IT)
      IF(NP.LT.6.OR.NP.GT.11)GOTO 6
      TOE=TN(26,IT)*4.0
      IF(NP.EQ.6.OR.NP.EQ.7)TOE=TN(27,IT)*15.0
      TOEP=AMAX1(0.0,TN(25,IT)-TOE)
      IF(ASUM.EQ.0.0)ASUM=1.0
    6 TN(28,IT)=TN(28,IT)*ASUM
      TN(29,IT)=TN(29,IT)*A(22,JF)
      TN(30,IT)=TN(30,IT)*A(23,JF)
      IF(IFM.LT.NB)GOTO 2
      DO 10 IF3=NB,IFM
      IF(A(4,IF3).LE.0.0)GOTO 10
      IXZ=A(27,IF3)
      FT(6,IXZ)=FT(6,IXZ)+A(4,IF3)
      IR=A(24,IF3)
      A9=SQRT(A(9,IF3))
      IRNG=A9
      RNGINT=IRNG
      RNGINT=RNGINT+.5
      IF(A9      .GE.RNGINT)IRNG=IRNG+1
      IF(A9      .GE.30.0)IRNG=30
      RDCNT(IR,IRNG)=RDCNT(IR,IRNG)+A( 4,IF3)
      IS=A(1,IF3)
      S(IS,1)=S(IS,1)+A(4,IF3)*WGT(IR)
      S(IS,2)=S(IS,2)+A(4,IF3)*CST(IR)
      S(IS,3)=S(IS,3)+(1.0-ASUM)*TN28*TOEP*QUV(IF3)*A(4,IF3)/A(13,JF)
      S(IS,4)=S(IS,4)+(1.0-A(22,JF))*TN29*TN(26,IT)*QUV(IF3)*A(4,IF3)/A(
```

```
     113,JF)                                                              SMV056
      S(IS,5)=S(IS,5)+(1.0-A(23,JF))*TN30*TN(27,IT)*QUV(IF3)*A(4,IF3)/A(  SMV057
     113,JF)                                                              SMV058
      TUBFU(IXZ)=TUBFU(IXZ)+TBM(IS)*A(5,IF3)                              SMV059
      IF(STYP(IS).EQ.2.0)GOTO 10                                          SMV060
      B=(A(4,IF3)-A(5,IF3)*TPFU(IS))/(A(5,IF3)*TPFU(IS)*A(3,IF3))         SMV061
      IF(TN(10,IT) .EQ. 0.) B=B+.067                                      SMV062
      B=AMAX1(0.0,B)                                                      SMV063
      TUBFU(IXZ)=TUBFU(IXZ)+B                                             SMV064
   10 CONTINUE                                                            SMV065
    2 IF3=IFM+1                                                           SMV066
      IS=A(1,IF3)                                                         SMV067
      IR=A(24,IF3)                                                        SMV068
      IXZ=A(27,IF3)                                                       SMV069
      FT(6,IXZ)=FT(6,IXZ)+A(4,IF3)*FM                                     SMV070
      A9=SQRT(A(9,IF3))                                                   SMV071
      IRNG=A9                                                             SMV072
      RNGINT=IRNG                                                         SMV073
      RNGINT=RNGINT+.5                                                    SMV074
      IF(A9       .GE.RNGINT)IRNG=IRNG+1                                  SMV075
      IF(A9       .GE.30.0)IRNG=30                                        SMV076
      RDCNT(IR,IRNG)=RDCNT(IR,IRNG)+A( 4,IF3)*FM                          SMV077
      S(IS,1)=S(IS,1)+A(4,IF3)*WGT(IR)*FM                                 SMV078
      S(IS,2)=S(IS,2)+A(4,IF3)*CST(IR)*FM                                 SMV079
      S(IS,3)=S(IS,3)+(1.0-ASUM)*TN28*TOEP*QUV(IF3)*A(4,IF3)*FM/A(13,JF)  SMV080
      S(IS,4)=S(IS,4)+(1.0-A(22,JF))*TN29*TN(26,IT)*QUV(IF3)*A(4,IF3)*FM  SMV081
     1/A(13,JF)                                                           SMV082
      S(IS,5)=S(IS,5)+(1.0-A(23,JF))*TN30*TN(27,IT)*QUV(IF3)*A(4,IF3)*FM  SMV083
     1/A(13,JF)                                                           SMV084
      TUBFU(IXZ)=TUBFU(IXZ)+TBM(IS)*A(5,IF3)*FM                           SMV085
      IF(STYP(IS).EQ.2.0)GOTO 5                                           SMV086
      B=(A(4,IF3)*FM-A(5,IF3)*TPFU(IS))/(A(5,IF3)*TPFU(IS)*A(3,IF3))      SMV087
      IF(TN(10,IT) .EQ. 0.) B=B+.067                                      SMV088
      B=AMAX1(0.0,B)                                                      SMV089
      TUBFU(IXZ)=TUBFU(IXZ)+B                                             SMV090
    5 GOTO(7,7,7,7,7,8,8,9,9,9,9,7),NP                                    SMV091
    7 SMW=SMW + (SAVE1 - TN(28,IT)) * 2.0 * TN(14,IT)                     SMV092
      IF(TN(28,IT).LE.TN(18,IT))GOTO 1                                    SMV093
      GOTO 11                                                             SMV094
    8 SMW=SMW + (SAVE3 - TN(30,IT)) * 2.0 * TN(14,IT)                     SMV095
      IF(TN(30,IT).LE.TN(18,IT))GOTO 1                                    SMV096
      GOTO 11                                                             SMV097
    9 SMW=SMW + (SAVE2 - TN(29,IT)) * 2.0 * TN(14,IT)                     SMV098
      IF(TN(29,IT).LE.TN(18,IT))GOTO 1                                    SMV099
   11 LOSS = LOSS + 1                                                     SMV100
      TLOST(1,LOSS)=TN(1,IT)                                              SMV101
      TLOST(2,LOSS)=TN(3,IT)                                              SMV102
      TLOST(3,LOSS)=TN(28,IT)                                             SMV103
      TLOST(4,LOSS)=TN(29,IT)                                             SMV104
      TLOST(5,LOSS)=TN(30,IT)                                             SMV105
      TR=1.0                                                              SMV106
      GOTO 999                                                            SMV107
    1 MFEAT= MFEAT + 1                                                    SMV108
      TDFT(MFEAT)  =TN(1,IT)                                              SMV109
      NFMD=NFMD + 1                                                       SMV110
      TR=1.0                                                              SMV111
  999 RETURN                                                              SMV112
      END                                                                 SMV113
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *   SOV001
      SUBROUTINE SHONVL(IT)                                               SOV002
```

```
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      COMMON (USE MAIN)
      NP =  TN(19,IT)  + 1.0
      RT=TN(22,IT)
      PERW=TN(20,IT)
      PERO=TN(21,IT)
      IF(IFONE.LT.NB)GOTO 3
      DO 10 IF2=NB,IFONE
      IS=A(1,IF2)
      TEMP=A(5,IF2)*TPFU(IS)
      IF(A(4,IF2).LT.TEMP)GOTO 10
      IR=A(24,IF2)
      IF(TN(24,IT).LT.T) GOTO 4
      IF(TN(4,IT).EQ.TN(19,IT).AND.TN(5,IT).EQ.TN(20,IT).AND.TN(6,IT)
     1.EQ.TN(21,IT).AND.TN(7,IT).EQ.TN(22,IT)) GOTO 4
      CALL INTERP(IR,IF2)
      ATLVL=0.0
      XVN=TEMP
      RPV=TEMP
      CALL EFFECT(IF2,IR,IT)
    4 ST=ST*A(14,IF2)
      PO=PO*A(15,IF2)
      FH=FH*A(16,IF2)
      TK=TK*A(17,IF2)
      APC=APC*A(18,IF2)
      IF(TN(24,IT).GE.T) GOTO 2
      A(19,IF2)= 0.
      A(20,IF2)= 0.
      A(21,IF2)= 0.
      A(22,IF2)= 1.
      A(23,IF2)= 1.
    2 ASUM=A(19,IF2)+A(20,IF2)+A(21,IF2)
      IF(ASUM.EQ.0.0)ASUM=1.0
      A9=SQRT(A(9,IF2))
      IRNG=A9
      RNGINT=IRNG
      RNGINT=RNGINT+.5
      IF(A9      .GE.RNGINT)IRNG=IRNG+1
      IF(A9      .GE.30.0)IRNG=30
      RDCNT(IR,IRNG)=RDCNT(IR,IRNG)+TEMP
      IXZ=A(27,IF2)
      FT(6,IXZ)=FT(6,IXZ)+TEMP
      S(IS,1)=S(IS,1)+TEMP*WGT(IR)
      S(IS,2)=S(IS,2)+TEMP*CST(IR)
      S(IS,3)=S(IS,3)+(1.0-ASUM)*TN28T
      S(IS,4)=S(IS,4)+(1.0-A(22,IF2))*TN29T
      S(IS,5)=S(IS,5)+(1.0-A(23,IF2))*TN30T
      TN28T=TN28T-TN28T*(1.0-ASUM)
      TN29T=TN29T-TN29T*(1.0-A(22,IF2))
      TN30T=TN30T-TN30T*(1.0-A(23,IF2))
      TUBFU(IXZ)=TUBFU(IXZ)+TBM(IS)*A(5,IF2)
      IF(TN(10,IT).EQ.0..AND.STYP(IS).EQ.1.)TUBFU(IXZ)=TUBFU(IXZ)+.067
   10 CONTINUE
    3 IF2=IFONE+1
      IS=A(1,IF2)
      TEMP=A(5,IF2)*TPFU(IS)*F
      IR=A(24,IF2)
      CALL INTERP(IR,IF2)
      ATLVL=0.0
      XVN=TEMP
```

```
      RPV=A(5,IF2)*TPFU(IS)
      CALL EFFECT(IF2,IR,IT)
      ST=ST*A(14,IF2)
      PO=PO*A(15,IF2)
      FH=FH*A(16,IF2)
      TK=TK*A(17,IF2)
      APC=APC*A(18,IF2)
      IF(TN(24,IT).GE.T) GOTO 5
      A(19,IF2)= 0.
      A(20,IF2)= 0.
      A(21,IF2)= 0.
      A(22,IF2)= 1.
      A(23,IF2)= 1.
      TK=0.
      APC=0.
    5 ASUM=A(19,IF2)+A(20,IF2)+A(21,IF2)
      IF(ASUM.EQ.0.0)ASUM=1.0
      IXZ=A(27,IF2)
      FT(6,IXZ)=FT(6,IXZ)+TEMP
      A9=SQRT(A(9,IF2))
      IRNG=A9
      RNGINT=IRNG
      RNGINT=RNGINT+.5
      IF(A9       .GE.RNGINT)IRNG=IRNG+1
      IF(A9       .GE.30.0)IRNG=30
      RDCNT(IR,IRNG)=RDCNT(IR,IRNG)+TEMP
      S(IS,1)=S(IS,1)+TEMP*WGT(IR)
      S(IS,2)=S(IS,2)+TEMP*CST(IR)
      S(IS,3)=S(IS,3)+(1.0-ASUM)*TN28T
      S(IS,4)=S(IS,4)+(1.0-A(22,IF2))*TN29T
      S(IS,5)=S(IS,5)+(1.0-A(23,IF2))*TN30T
      TN28T=TN28T-TN28T*(1.0-ASUM)
      TN29T=TN29T-TN29T*(1.0-A(22,IF2))
      TN30T=TN30T-TN30T*(1.0-A(23,IF2))
      TUBFU(IXZ)=TUBFU(IXZ)+TBM(IS)*A(5,IF2)*F
      IF(TN(10,IT).EQ.0..AND.STYP(IS).EQ.1.)TUBFU(IXZ)=TUBFU(IXZ)+.067
      TOEP=TN(25,IT)
      IF(NP.LT.6.OR.NP.GT.11)GOTO6
      TOE=TN(26,IT)*4.0
      IF(NP.EQ.6.OR.NP.EQ.7)TOE=TN(27,IT)*15.0
      TOEP=AMAX1(0.0,TN(25,IT)-TOE)
    6 IF(TOEP.EQ.0.0)GOTO 7
      TN(28,IT)=TN28T/TOEP
    7 TKPO=POST(NP,5)*TK
      IF(TKPO.EQ.0.0)TKPO=1.0
      APCPO=POST(NP,6)*APC
      IF(APCPO.EQ.0.0)APCPO=1.0
      TN(29,IT)=TN(29,IT)*TKPO
      TN(30,IT)=TN(30,IT)*APCPO
      GOTO(8,8,8,8,8,9,9,11,11,11,11,8),NP
    8 SMW=SMW + (SAVE1 - TN(28,IT)) * 2.0 * TN(14,IT)
      IF(TN(28,IT).LE.TN(18,IT))GOTO 1
      GOTO 12
    9 SMW=SMW + (SAVE3 - TN(30,IT)) * 2.0 * TN(14,IT)
      IF(TN(30,IT).LE.TN(18,IT))GOTO 1
      GOTO 12
   11 SMW=SMW + (SAVE2 - TN(29,IT)) * 2.0 * TN(14,IT)
      IF(TN(29,IT).LE.TN(18,IT))GOTO 1
   12 LOSS = LOSS + 1
      TLOST(1,LOSS)=TN(1,IT)
```

```
      TLOST(2,LOSS)=TN(3,IT)
      TLOST(3,LOSS)=TN(28,IT)
      TLOST(4,LOSS)=TN(29,IT)
      TLOST(5,LOSS)=TN(30,IT)
      TR=1.0
      GOTO 999
    1 MFEAT= MFEAT + 1
      TDFT(MFEAT)  =TN(1,IT)
      NFMD=NFMD + 1
      TR=1.0
  999 RETURN
      END
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      SUBROUTINE ONEVOL(IT)
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      DIMENSION A1(2,50),B(27)
      COMMON(USE MAIN)
      NEA= NE - NB + 1
      IAC=0
      K=0
      DO 12 J=1,NEA
      CRITMI=1000000.0
      DO 13 M=NB,NE
      IF(A(25,M) .LT. CRITMI) GOTO 2
      GOTO 13
    2 CRITMI=A(25,M)
      MI=M
   13 CONTINUE
      IAC=IAC+1
      A1(1,IAC)=MI
      A1(2,IAC)=A(25,MI)
      A(25,MI)=1000000.0
   12 CONTINUE
      DO 14 M=NB,NE
      K=K+1
      MI=A1(1,K)
      DO 15 L=1,27
      B(L)  =A(L,M)
      A(L,M)=A(L,MI)
      A(L,MI)=B(L)
   15 CONTINUE
      A(25,M)=A1(2,K)
      DO 16 L=K,NEA
      IM=A1(1,L)
      IF(IM .EQ. M) GOTO 3
   16 CONTINUE
    3 A1(1,L)=MI
   14 CONTINUE
      CONSTR=15.0
      IF(TN(14,IT).GE.10.0)CONSTR=30.0
      DO 10 JF=NB,NE
      IRS=A(24,JF)
      IS=A(1,JF)
      ROUNDS=A(5,JF)*TPFU(IS)
      IF(A(4,JF).LT.ROUNDS)GOTO 10
      CRITP=CRIT
      CRIT=CRIT+A(26,JF)
      ST=ST*A(14,JF)
      PO=PO*A(15,JF)
      FH=FH*A(16,JF)
```

```
      TK=TK*A(17,JF)                                                    ONE049
      APC=APC*A(18,JF)                                                  ONE050
      TEMP1=ROUNDS*WGT(IRS)                                             ONE051
      SURVP=SURV                                                        ONE052
      SURV=POST(NP,7)*ST + POST(NP,8)*PO + POST(NP,9)*FH +              ONE053
     1      POST(NP,10)*TK + POST(NP,11)*APC                            ONE054
      IF(RTP(IRS).EQ.2.0.AND.TN(10,IT).NE.0.)SURV=POST(NP,2)*ST+POST    ONE055
     1(NP,3)*PO+POST(NP,4)*FH+POST(NP,5)*TK+POST(NP,6)*APC              ONE056
      IF(SURV.LE.TN(17,IT))GOTO 11                                      ONE057
      WAIT=WAIT+TEMP1                                                   ONE058
      IF(WAIT.GT.CONSTR)GOTO 1                                          ONE059
   10 CONTINUE                                                          ONE060
      GOTO 999                                                          ONE061
   11 IFONE=JF-1                                                        ONE062
      F=(SURVP-TN(17,IT))/(SURVP-SURV)                                  ONE063
      WAIT=WAIT+TEMP1*F                                                 ONE064
      IF(WAIT.GT.CONSTR)GOTO 1                                          ONE065
      ONCRT =CRITP+F*(CRIT-CRITP)                                       ONE066
      GOTO 999                                                          ONE067
    1 SURV=1.0                                                          ONE068
      EXCES1=1.0                                                        ONE069
  999 RETURN                                                            ONE070
      END                                                               ONE071
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * MUL001
      SUBROUTINE MULVOL(IT)                                             MUL002
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * MUL003
      COMMON(USE MAIN)                                                  MUL004
      CONSTR=15.0                                                       MUL005
      IF(TN(14,IT).GE.10.0)CONSTR=30.0                                  MUL006
      DO 10 IF1=NA1,NE                                                  MUL007
      IRS=A(24,IF1)                                                     MUL008
      QUV(IF1)=QN/A(13,IF1)                                             MUL009
      SQNP=SQN                                                          MUL010
      SQN=SQN+AMAX1(A(4,IF1),0.0)*QUV(IF1)                              MUL011
      TEMP=A(4,IF1)*CRT(IRS)                                            MUL012
      TEMP2=A(4,IF1)*WGT(IRS)                                           MUL013
      IF(SQN.GE.QN)GOTO 1                                               MUL014
      WAIT2=WAIT2+TEMP2                                                 MUL015
      IF(WAIT2.GT.CONSTR)GOTO 2                                         MUL016
      SMCRT=SMCRT+TEMP                                                  MUL017
   10 CONTINUE                                                          MUL018
      GOTO 999                                                          MUL019
    1 FM=((QN-SQNP)/(SQN-SQNP))                                         MUL020
      IFM=IF1-1                                                         MUL021
      SMCRT=SMCRT+FM*TEMP                                               MUL022
      WAIT2=WAIT2+TEMP2*FM                                              MUL023
      IF(WAIT2.GT.CONSTR)GOTO 2                                         MUL024
      GOTO 999                                                          MUL025
    2 SQN=0.0                                                           MUL026
      EXCES2=1.0                                                        MUL027
  999 RETURN                                                            MUL028
      END                                                               MUL029
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * COR001
      SUBROUTINE CORP(IT)                                               COR002
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * COR003
      COMMON(USE MAIN)                                                  COR004
      SAVE1=TN(28,IT)                                                   COR005
      SAVE2=TN(29,IT)                                                   COR006
      SAVE3=TN(30,IT)                                                   COR007
      KEY=0                                                             COR008
```

```
      MORG=3                                                            COR009
      CALL AMASS(IT)                                                    COR010
      NEA3=NA                                                           COR011
      NEA=MAX0(NEA2,NEA1)                                               COR012
      NBA3=MIN0(NA,NEA +1)                                              COR013
      NB=NBA3                                                           COR014
      NE=NEA3                                                           COR015
      OFLAG1=0.0                                                        COR016
      OFLAG2=0.0                                                        COR017
      QN= 1000000.0 $ QNCRT=0.0                                         COR018
      JF = 0                                                            COR019
      WAIT=0. $ WAIT2=0. $ EXCES1=0. $ EXCES2=0.                        COR020
      ST=1. $ PO=1. $ FH=1. $ TK=1. $ APC=1.                            COR021
      SURVP=1. $ SURV=1. $ CRITP=0. $ CRIT=0. $ SQN=0. $ SMCRT=0.       COR022
      TN28T=TN(28,IT)*TN(25,IT)                                         COR023
      IF(NP.LT.6.OR.NP.GT.11)GOTO 38                                    COR024
      TOE=TN(26,IT)*4.0                                                 COR025
      IF(NP.EQ.6.OR.NP.EQ.7)TOE=TN(27,IT)*15.0                          COR026
      TN28T=(AMAX1(0.0,TN(25,IT)-TOE))*TN(28,IT)                        COR027
   38 TN29T=TN(29,IT)*TN(26,IT)                                         COR028
      TN30T=TN(30,IT)*TN(27,IT)                                         COR029
      DO 50 IQ=1,NFU                                                    COR030
   50 QUV(IQ)=0.0                                                       COR031
      IF(NBA3.LE.NEA2.OR.NBA3.LE.NEA1)GOTO 1                            COR032
      CALL ONEVOL(IT)                                                   COR033
      QN = A(13,NB)                                                     COR034
      NA1 = NB                                                          COR035
      JF = NA1                                                          COR036
      CALL MULVOL(IT)                                                   COR037
      IF(SURV .GT. TN(17,IT)) GOTO 4                                    COR038
      IF(SQN.LT.QN)GOTO 5                                               COR039
      IF(QNCRT.LT.SMCRT)GOTO 5                                          COR040
    6 CALL SHMUVL(IT,JF)                                                COR041
      GOTO 999                                                          COR042
    5 ST=1.0 $ PO=1.0 $ FH=1.0 $ TK=1.0 $ APC=1.0                       COR043
      CALL SHONVL(IT)                                                   COR044
      GOTO 999                                                          COR045
    1 OFLAG1=5.0                                                        COR046
      GOTO 30                                                           COR047
    4 IF(SQN.GE.QN)GOTO 6                                               COR048
      IF(EXCES1.EQ.1.0.AND.EXCES2.EQ.1.0)GOTO 60                        COR049
   30 IF(DVFLAG.EQ.1.0) GOTO 8                                          COR050
      MORG=2                                                            COR051
      CALL AMASS(IT)                                                    COR052
      NEA2=NA                                                           COR053
      NBA2=MIN0(NA,NEA3+1)                                              COR054
      IF(NBA2.LE.NEA3)GOTO 31                                           COR055
      GOTO 35                                                           COR056
    8 IF(NEA2.EQ.0) GOTO 7                                              COR057
      IF(NEA1.EQ.NBA2.AND.NBA2.EQ.NEA2.AND.DSFLAG.EQ.1.0) GOTO 7        COR058
      GOTO 35                                                           COR059
    7 OFLAG2=5.0                                                        COR060
      IF(NEA1.EQ.0) GOTO 60                                             COR061
      GOTO 36                                                           COR062
   35 NB=NBA2                                                           COR063
      NE=NEA2                                                           COR064
      CALL ONEVOL(IT)                                                   COR065
      IF(QN.EQ.1000000.0)GOTO 41                                        COR066
      IF(OFLAG1.NE.5.0)GOTO 9                                           COR067
   41 QN = A(13,NB)                                                     COR068
```

```
 9 NA1=NB
   JF = NA1
   CALL MULVOL(IT)
   IF( SURV .GT. TN(17,IT) ) GOTO 13
   IF(SQN.LT.QN)GOTO 14
   IF(QNCRT.LT.SMCRT)GOTO 14
   GOTO 15
13 IF(SQN.LT.QN)GOTO 16
15 TN28=TN(28,IT)
   TN29=TN(29,IT)
   TN30=TN(30,IT)
   CALL SHMUVL(IT,JF)
   ASUM=A(19,JF)+A(20,JF)+A(21,JF)
   IF(ASUM.EQ.0.0)ASUM=1.0
   IF(OFLAG1.EQ.5.0) GOTO 999
   DO 53 IF2=NBA3,NEA3
   IF(A(4,IF2).LE.0.0)GOTO 53
   IR=A(24,IF2)
   IXZ=A(27,IF2)
   FT(6,IXZ)=FT(6,IXZ)+A(4,IF2)
   A9=SQRT(A(9,IF2))
   IRNG=A9
   RNGINT=IRNG
   RNGINT=RNGINT+.5
   IF(A9       .GE.RNGINT)IRNG=IRNG+1
   IF(A9       .GE.30.0)IRNG=30
   RDCNT(IR,IRNG)=RDCNT(IR,IRNG)+A(4,IF2)
   IS=A(1,IF2)
   S(IS,1)=S(IS,1)+A(4,IF2)*WGT(IR)
   S(IS,2)=S(IS,2)+A(4,IF2)*CST(IR)
   S(IS,3)=S(IS,3)+(1.0-ASUM)*TN28*TOEP*QUV(IF2)*A(4,IF2)/A(13,JF)
   S(IS,4)=S(IS,4)+(1.0-A(22,JF))*TN29*TN(26,IT)*QUV(IF2)*A(4,IF2)/A(
  113,JF)
   S(IS,5)=S(IS,5)+(1.0-A(23,JF))*TN30*TN(27,IT)*QUV(IF2)*A(4,IF2)/A(
  113,JF)
   TUBFU(IXZ)=TUBFU(IXZ)+TBM(IS)*A(5,IF2)
   IF(STYP(IS).EQ.2.0)GOTO 53
   B=(A(4,IF2)-A(5,IF2)*TPFU(IS))/(A(5,IF2)*TPFU(IS)*A(3,IF2))
   IF(TN(10,IT) .EQ. 0.) B=B+.067
   B= AMAX1(0.0,B)
   TUBFU(IXZ)=TUBFU(IXZ)+B
53 CONTINUE
   GOTO 999
14 ST=1.0 $ PO=1.0 $ FH=1.0 $ TK=1.0 $ APC=1.0
   RT=TN(22,IT)
   NP= TN(19,IT) + 1.0
   PERW=TN(20,IT)
   PERO=TN(21,IT)
   IF(OFLAG1.EQ.5.0)GOTO 18
   K1= NBA3  $  K2= NEA3  $ KEY= 1
   GOTO 59
18 CALL SHONVL(IT)
   GOTO 999
16 IF(EXCES1.EQ.1.0.AND.EXCES2.EQ.1.0)GOTO 60
   IF(DVFLAG.EQ.1.0) GOTO 19
   GOTO 11
31 OFLAG2=5.0
11 MORG=1
   CALL AMASS(IT)
   NEA1=NA
```

```
      NBA1=MIN0(NA,NEA2+1)
      IF(NBA1.LE.NEA2)GOTO 60
      GOTO 36
   19 IF(NBA1.EQ.0) GOTO 60
      IF(NBA1.EQ.NEA2.AND.DSFLAG.NE.1.0) GOTO 60
   36 NB=NBA1
      NE=NEA1
      CALL ONEVOL(IT)
      IF(QN.EQ.1000000.0)GOTO 42
      IF(OFLAG1.NE.5.0.OR.OFLAG2.NE.5.0)GOTO 40
   42 QN = A(13,NB)
   40 NA1=NB
      JF = NA1
      CALL MULVOL(IT)
      IF( SURV .GT. TN(17,IT)) GOTO 23
      IF(SQN.LT.QN)GOTO 24
      IF(QNCRT.LT.SMCRT)GOTO 24
      GOTO 25
   23 IF(SQN.LT.QN)GOTO 60
   25 TN28=TN(28,IT)
      TN29=TN(29,IT)
      TN30=TN(30,IT)
      CALL SHMUVL(IT,JF)
      ASUM=A(19,JF)+A(20,JF)+A(21,JF)
      IF(ASUM.EQ.0.0)ASUM=1.0
      IF(OFLAG1.EQ.5.0)GOTO 26
      DO 56 IF4=NBA3,NEA3
      IF(A(4,IF4).LE.0.0)GOTO 56
      IR=A(24,IF4)
      IXZ=A(27,IF4)
      FT(6,IXZ)=FT(6,IXZ)+A(4,IF4)
      A9=SQRT(A(9,IF4))
      IRNG=A9
      RNGINT=IRNG
      RNGINT=RNGINT+.5
      IF(A9       .GE.RNGINT)IRNG=IRNG+1
      IF(A9       .GE.30.0)IRNG=30
      RDCNT(IR,IRNG)=RDCNT(IR,IRNG)+A(4,IF4)
      IS=A(1,IF4)
      S(IS,1)=S(IS,1)+A(4,IF4)*WGT(IR)
      S(IS,2)=S(IS,2)+A(4,IF4)*CST(IR)
      S(IS,3)=S(IS,3)+(1.0-ASUM)*TN28*TOEP*QUV(IF4)*A(4,IF4)/A(13,JF)
      S(IS,4)=S(IS,4)+(1.0-A(22,JF))*TN29*TN(26,IT)*QUV(IF4)*A(4,IF4)/A(
     113,JF)
      S(IS,5)=S(IS,5)+(1.0-A(23,JF))*TN30*TN(27,IT)*QUV(IF4)*A(4,IF4)/A(
     113,JF)
      TUBFU(IXZ)=TUBFU(IXZ)+TBM(IS)*A(5,IF4)
      IF(STYP(IS).EQ.2.0)GOTO 56
      B=(A(4,IF4)-A(5,IF4)*TPFU(IS))/(A(5,IF4)*TPFU(IS)*A(3,IF4))
      IF(TN(10,IT) .EQ. 0.) B=B+.067
      B=AMAX1(0.0,B)
      TUBFU(IXZ)=TUBFU(IXZ)+B
   56 CONTINUE
   26 IF(OFLAG2.EQ.5.0) GOTO 999
      DO 57 IF4=NBA2,NEA2
      IF(A(4,IF4).LE.0.0)GOTO 57
      IR=A(24,IF4)
      IXZ=A(27,IF4)
      FT(6,IXZ)=FT(6,IXZ)+A(4,IF4)
      A9=SQRT(A(9,IF4))
```

```
      IRNG=A9
      RNGINT=IRNG
      RNGINT=RNGINT+.5
      IF(A9       .GE.RNGINT)IRNG=IRNG+1
      IF(A9       .GE.30.0)IRNG=30
      RDCNT(IR,IRNG)=RDCNT(IR,IRNG)+A(4,IF4)
      IS=A(1,IF4)
      S(IS,1)=S(IS,1)+A(4,IF4)*WGT(IR)
      S(IS,2)=S(IS,2)+A(4,IF4)*CST(IR)
      S(IS,3)=S(IS,3)+(1.0-ASUM)*TN28*TOEP*QUV(IF4)*A(4,IF4)/A(13,JF)
      S(IS,4)=S(IS,4)+(1.0-A(22,JF))*TN29*TN(26,IT)*QUV(IF4)*A(4,IF4)/A(
     113,JF)
      S(IS,5)=S(IS,5)+(1.0-A(23,JF))*TN30*TN(27,IT)*QUV(IF4)*A(4,IF4)/A(
     113,JF)
      TUBFU(IXZ)=TUBFU(IXZ)+TBM(IS)*A(5,IF4)
      IF(STYP(IS).EQ.2.0)GOTO 57
      B=(A(4,IF4)-A(5,IF4)*TPFU(IS))/(A(5,IF4)*TPFU(IS)*A(3,IF4))
      IF(TN(10,IT) .EQ. 0.) B=B+.067
      B=AMAX1(0.0,B)
      TUBFU(IXZ)=TUBFU(IXZ)+B
   57 CONTINUE
      GOTO 999
   24 ST=1.0 $ PO=1.0 $ FH=1.0 $ TK=1.0 $ APC=1.0
      RT=TN(22,IT)
      NP= TN(19,IT) + 1.0
      PERW=TN(20,IT)
      PERO=TN(21,IT)
      IF(OFLAG1.EQ.5.0)GOTO 28
      K1= NBA3          $  K2= NEA3
      GOTO 59
   28 IF(OFLAG2.EQ.5.0)GOTO 29
      K1= NBA2  $ K2= NEA2
   59 DO 58 IF3= K1,K2
      IS=A(1,IF3)
      TEMP=A(5,IF3)*TPFU(IS)
      IF(A(4,IF3).LT.TEMP)GOTO 58
      IR=A(24,IF3)
      IF(TN(24,IT).LT.T) GOTO 34
      IF(TN(4,IT).EQ.TN(19,IT).AND.TN(5,IT).EQ.TN(20,IT).AND.TN(6,IT)
     1.EQ.TN(21,IT).AND.TN(7,IT).EQ.TN(22,IT)) GOTO 34
      CALL INTERP(IR,IF3)
      RPV=TEMP
      ATLVL=0.0
      XVN=TEMP
      CALL EFFECT(IF3,IR,IT)
   34 ST=ST*A(14,IF3)
      PO=PO*A(15,IF3)
      FH=FH*A(16,IF3)
      TK=TK*A(17,IF3)
      APC= APC * A(18,IF3)
      IF(TN(24,IT).GE.T) GOTO 32
      A(19,IF3)= 0.
      A(20,IF3)= 0.
      A(21,IF3)= 0.
      A(22,IF3)= 1.
      A(23,IF3)= 1.
   32 ASUM=A(19,IF3)+A(20,IF3)+A(21,IF3)
      IF(ASUM.EQ.0.0)ASUM=1.0
      IXZ=A(27,IF3)
      FT(6,IXZ)=FT(6,IXZ)+TEMP
```

```
      A9=SQRT(A(9,IF3))
      IRNG=A9
      RNGINT=IRNG
      RNGINT=RNGINT+.5
      IF(A9      .GE.RNGINT)IRNG=IRNG+1
      IF(A9      .GE.30.0)IRNG=30
      RDCNT(IR,IRNG)=RDCNT(IR,IRNG)+TEMP
      S(IS,1)=S(IS,1)+TEMP*WGT(IR)
      S(IS,2)=S(IS,2)+TEMP*CST(IR)
      S(IS,3)=S(IS,3)+(1.0-ASUM)*TN28T
      S(IS,4)=S(IS,4)+(1.0-A(22,IF3))*TN29T
      S(IS,5)=S(IS,5)+(1.0-A(23,IF3))*TN30T
      TN28T=TN28T-TN28T*(1.0-ASUM)
      TN29T=TN29T-TN29T*(1.0-A(22,IF3))
      TN30T=TN30T-TN30T*(1.0-A(23,IF3))
      TUBFU(IXZ)=TUBFU(IXZ)+TBM(IS)*A(5,IF3)
      IF(TN(10,IT).EQ.0..AND.STYP(IS).EQ.1.)TUBFU(IXZ)=TUBFU(IXZ)+.067
   58 CONTINUE
      IF( KEY .EQ. 1 ) GOTO 18
      IF(K1 .EQ. NBA3) GOTO 28
   29 CALL SHONVL(IT)
      GOTO 999
   60 IF(TN(17,IT).EQ..5)GOTO 61
   64 IF(TN(14,IT).LT.3.0) GOTO 999
      ALPHA=1.0-.9/TN(14,IT)
      IF(ALPHA.LE.TN(17,IT)) GOTO 999
      IF(TN(9,IT).GE.(T+DELT+.00001)) GOTO 999
      TN(17,IT)=ALPHA
   63 REFIRE=1.0
      GOTO 999
   61 NP= TN(4,IT) + 1.
      IF( NP .LT. 6 .OR. NP .GT. 11) GOTO 62
      TN(17,IT)=.7
      GOTO 63
   62 TN(17,IT)=AMIN1(.75,1.0-.5*(1.0 - POST(NP,9)))
      IF(TN(17,IT).EQ.0.5) GOTO 64
      GOTO 63
  999 RETURN
      END
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      SUBROUTINE EFFECT(IA,KR,IT)
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      COMMON(USE MAIN)
      REAL LVT $ RND=RTP(KR) $ IST=A(1,IA) $ CN=XVN*REL(KR) $ NPSET=NP
      ART=TN(8,IT)$ LVT=TN(9,IT)
      CRSQ = CPER * CPER
      CTSQ = CPET * CPET
      TNSQ = TN(10,IT) * TN(10,IT)
      IF(LVT .EQ. ART) GOTO 1
      IF(STYP(IST) .EQ. 2.0) GOTO 2
      IF(TN(10,IT) .EQ. 0.) GOTO 3
      CPET = CTSQ + TNSQ
      SHEAF= .36 * (CPET - CRSQ )
      CPER  = SQRT(CRSQ + SHEAF)
      CPET=SQRT(CPET+SHEAF)
      GOTO 4
    3 CPET= CTSQ + 2500.
      SHEAF= .36 * (CPET - CRSQ )
      CPER = SQRT(CRSQ + SHEAF)
      CPET=SQRT(CPET+SHEAF)
```

```
      GOTO 4
    2 IF(TN(10,IT) .EQ. 0.) GOTO 5
      CPET = CTSQ + TNSQ
      SHEAF= .36 * (CPET - CRSQ )
      CPER = SQRT(CRSQ + SHEAF)
      CPET=SQRT(CPET+SHEAF)
      GOTO 4
    5 SHEAF= .36 * (CTSQ - CRSQ)
      CPER = SQRT(CRSQ + SHEAF)
      CPET = SQRT(CTSQ + SHEAF)
      GOTO 4
    1 IF(STYP(IST) .EQ. 2.0) GOTO 6
      CPET= CTSQ + 3125.
      SHEAF= .36 * (CPET - CRSQ )
      CPER = SQRT(CRSQ + SHEAF)
      CPET=SQRT(CPET+SHEAF)
      GOTO 4
    6 CPET= CTSQ + 10000.
      SHEAF= .36 * (CPET - CRSQ )
      CPER = SQRT(CRSQ + SHEAF)
      CPET=SQRT(CPET+SHEAF)
    4 RASR= SQRT(CPER/CPET)
      MP=2
      IF(NP .LT. 6  .OR. NP  .GT.  11 ) MP= 1
      IF(ATLVL .NE. 0. .AND. MP .EQ. 2) GOTO 7
      NCODE= 1
      GOTO 8
    9 IF(ATLVL .NE. 0.) GOTO 10
    7 NCODE= 1
      GOTO 11
   12 IF(ATLVL .NE. 0.) GOTO 10
      RPV= RPV * REL(KR)
      IF(CN .LE. RPV) GOTO 13
   10 OVN= 0.
      IF(ATLVL .NE. 0. .AND. MP .EQ. 2) GOTO 14
      DO 60 I=1,8
      IF(I .EQ. 4 .OR. I .EQ. 5) GOTO 60
      ECZ=CCOV(I) * (1.-(1.-CPK(I)) ** RPV)
      ECO= 1. - (1.-CCOV(I) * CPK(I))**RPV
      CSURV(I)= 1. - (ECZ + RASR * (ECO - ECZ))
   60 CONTINUE
      IF( ATLVL .EQ. 0.0 .AND. MP .EQ. 2 ) NP=1
      DO 61 I= 1,3
      IB=I+1
      IF(RND .EQ. 1.0) IB= I + 6
      IF(RND .EQ. 2. .AND. TN(10,IT).EQ. 0.) IB=I+6
      A(I+13,IA)=CSURV(I)*PERO+CSURV(I+5)*PERW
   61 OVN= OVN + POST(NP,IB) * A(I+13,IA)
      IF(RND .EQ. 2. .AND. TN(10,IT).NE. 0.) GOTO 15
      CPOST(1)=A(14,IA)*POST(NP,7)
      CPOST(2)=A(15,IA)*POST(NP,8)
      CPOST(3) = A(16,IA) * POST(NP,9)
      NP=NPSET
      IF(ATLVL .EQ. 0. .OR. MP .EQ. 2) GOTO 14
      GOTO 16
   15 FSP= AMAX1(0.0,POST(NP,8)-POST(NP,3))
      FSF=POST(NP,2)-POST(NP,7)-FSP
      FPF=POST(NP,3)-POST(NP,8)+FSP
      CPOST(1)=A(14,IA)*(POST(NP,2)-FSP-FSF)
      CPOST(2)= A(15,IA) * POST(NP,3) + A(14,IA) * FSP - A(15,IA) * FPF
```

```
      CPOST(3)= A(16,IA) * POST(NP,4) + A(14,IA) * FSF + A(15,IA) * FPF
      NP=NPSET
      IF(ATLVL .NE. 0. .OR. MP .EQ. 1) GOTO 16
   14 DO 62 I=4,10
      IF(I .EQ. 6 .OR. I .EQ. 7 .OR. I .EQ. 8) GOTO 62
      ECZ=CCOV(I) * (1.-(1.-CPK(I)) ** RPV)
      ECO= 1. - (1.-CCOV(I) * CPK(I))**RPV
      CSURV(I)= 1. - (ECZ + RASR * (ECO - ECZ))
   62 CONTINUE
      DO 63 I=4,5
      A(I+13,IA)=CSURV(I)*PERO+CSURV(I+5)*PERW
      OVN=OVN+POST(NP,I+1)*A(I+13,IA)
   63 CPOST(I)= A(I+13,IA) *POST(NP,I+6)
   16 A(12,IA)= OVN
      IF(ATLVL .EQ. 0.) GOTO 17
      IF(OVN .LE. ATLVL) GOTO 18
      IF(LVT .EQ. ART .AND. STYP(IST) .EQ. 2.) GOTO 19
      IF(LVT .NE. ART) GOTO 20
      CPET= CTSQ + 19400.
      SHEAF= .36 * (CPET - CRSQ )
      CPER=SQRT(CRSQ+SHEAF)
      CPET=SQRT(CPET+SHEAF)
      RASR=SQRT(CPER/CPET)
      GOTO (21,22), MP
   21 NCODE= 2
    8 K=1
      REF=CRE(K)
      CALL COV
      CCOV(K)=EC1
      KB=K
      DO 64 K=2,8
      IF(K .EQ. 4 .OR. K .EQ. 5) GOTO 64
      IF(CRE(K) .EQ. CRE(KB)) GOTO 23
      REF=CRE(K)
      CALL COV
      CCOV(K)=EC1
      GOTO 64
   23 CCOV(K)= CCOV(KB)
      KB=K
   64 CONTINUE
      GOTO (9,20), NCODE
   22 NCODE= 2
   11 I= 4
      REF=CRE(I)
      CALL COV
      CCOV(I)=EC1
      IB= I
      DO 65 I=5,10
      IF(I .EQ. 6 .OR. I .EQ. 7 .OR. I .EQ. 8) GOTO 65
      IF(CRE(I) .EQ. CRE(IB)) GOTO 24
      REF=CRE(I)
      CALL COV
      CCOV(I)=EC1
      GOTO 65
   24 CCOV(I)= CCOV(IB)
      IB= I
   65 CONTINUE
      GOTO (12,20), NCODE
   20 UP= 1000.
      DOWN= 0.
```

```
      CN= 500.
      GOTO (25,26), MP
   25 NCODE= 1
      GOTO 27
   28 SRV= 0.
      DO 66 I=1,3
   66 SRV=(CSURV(I)*PERO + CSURV(I+5)*PERW)*CPOST(I) + SRV
      GOTO 29
   26 NCODE=1
      GOTO 30
   39 SRV= 0.
      DO 67 I=4,5
   67 SRV=(CSURV(I)*PERO + CSURV(I+5)*PERW)*CPOST(I) + SRV
   29 IF(SRV .GT. .95 .OR. CN .GT. 980.) GOTO 19
      IF(ATLVL - .005.LE. SRV .AND. SRV .LE. ATLVL + .005) GOTO 31
      IF(SRV .GT. ATLVL) GOTO 32
      UP=CN
      CN=.5*(UP+DOWN)
      GOTO (25,26), MP
   32 DOWN= CN
      CN=.5*(UP+DOWN)
      GOTO (25,26), MP
   31 A(13,IA)= (CN+RPV) * RELI(KR)
      IF(LVT .EQ. ART .AND. CN .GT. RPV) GOTO 19
      GOTO (33,34), MP
   33 DO 68 I=1,3
   68 A(I+18,IA)=(CSURV(I)*PERO + CSURV(I+5)*PERW) * CPOST(I)
      GOTO 35
   34 DO 36 I=4,5
   36 A(I+18,IA)=(CSURV(I)*PERO + CSURV(I+5)*PERW) * CPOST(I)
   35 IF(A(22,IA) .EQ. 0.) A(22,IA)= 1.
      IF(A(23,IA).EQ.0.0)A(23,IA)=1.0
      GOTO 999
   17 CN=CN - RPV * RELI(KR)
      NCODE= 2.
      GOTO 27
   37 DO 69 I=1,3
      A(I+18,IA)= (CSURV(I) * PERO + CSURV(I+5) * PERW) * CPOST(I)
   69 ATLVL= ATLVL + A(I+18,IA)
      IF(MP .EQ. 1) GOTO 38
      NCODE= 2
      GOTO 30
   40 DO 70 I=4,5
      A(I+18,IA)= (CSURV(I) * PERO + CSURV(I+5) * PERW) * CPOST(I)
   70 ATLVL= ATLVL + A(I+18,IA)
   38 IF(A(22,IA) .EQ. 0.) A(22,IA)= 1.
      IF(A(23,IA).EQ.0.0)A(23,IA)=1.0
      GOTO 999
   13 OVN= 0.
      NCODE= 3
   27 DO 71 I=1,8
      IF(I .EQ. 4 .OR. I .EQ. 5) GOTO 71
      ECZ= CCOV(I) * (1.-(1.-CPK(I)) ** CN)
      ECO= 1. -(1. - CCOV(I) * CPK(I)) ** CN
      CSURV(I)= 1. -(ECZ + RASR * (ECO-ECZ))
   71 CONTINUE
      GOTO (28,37,41), NCODE
   41 IF(MP .EQ. 2) NP=1
      DO 72 I=1,3
      IC=I+1
```

```
      IF(RND .EQ. 1.0 .OR. MP  .EQ. 2) IC= I + 6
      IF(RND .EQ. 2. .AND. TN(10,IT).EQ. 0.) IC=I+6
      A(I+13,IA)=CSURV(I)*PERO+CSURV(I+5)*PERW
      A1=A(I+13,IA)*POST(NP,IC)
      A(I+18,IA)=A1
   72 OVN= OVN + A1
      NP=NPSET
      IF(MP .EQ. 1) GOTO 42
      NCODE= 3
   30 DO 73 I=4,10
      IF(I .EQ. 6 .OR. I .EQ. 7 .OR. I .EQ. 8) GOTO 73
      ECZ= CCOV(I) * (1.-(1.-CPK(I)) ** CN)
      ECO= 1. -(1. - CCOV(I) * CPK(I)) ** CN
      CSURV(I)= 1. -(ECZ + RASR * (ECO-ECZ))
   73 CONTINUE
      GOTO (39,40,43), NCODE
   43 DO 74 I=4,5
      A1=(CSURV(I)*PERO+CSURV(I+5)*PERW)*POST(NP,I+1)
      A(I+18,IA)=A1
   74 OVN= OVN +A1
   42 IF(A(22,IA) .EQ. 0.) A(22,IA)= 1.
      IF(A(23,IA).EQ.0.0)A(23,IA)=1.0
      A(12,IA)=OVN
      GOTO 999
   18 A(13,IA)= RPV * RELI(KR)
      GOTO (44,45), MP
   44 DO 75 I=1,3
      IE = I + 1
      IF( RND .EQ. 1.0) IE = I + 6
      IF(RND .EQ. 2. .AND. TN(10,IT).EQ. 0.) IE=I+6
   75 A(I+18,IA)= A(I+13,IA) * POST(NP,IE)
      GOTO 46
   45 DO 76 I=4,5
   76 A(I+18,IA)= A(I+13,IA) * POST(NP,I+1)
   46 IF(A(22,IA) .EQ. 0.) A(22,IA)= 1.
      IF(A(23,IA).EQ.0.0)A(23,IA)=1.0
      GOTO 999
   19 A(13,IA)= 5000000C.
  999 RETURN
      END
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      SUBROUTINE COV
C     COMPUTES A CIRCLE ON CIRCLE ONE ROUND COVERAGE BY 16 INTERVALS
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      COMMON W,W1,PII,CPET,RT,REF,EC1
      T1=REF/RT
      T2=T1*T1
      FR=0.
      IF(CPET) 1,2,1
    1 RMAX=REF + RT
      TEMP=3.61 * CPET
      CPSI=1./(CPET * CPET)
      IF(RMAX-TEMP)4,4,3
 4    RMAX1=RMAX
      GOTO 5
    2 IF(REF - RT) 6,7,7
 6    EC1=T2
      GOTO 999
 7    EC1=1.
      GOTO 999
```

```
 3     RMAX=TEMP                                                        COV021
       RMAX1=TEMP                                                       COV022
 5     DR1=RMAX*.0625                                                   COV023
       R=DR1                                                            COV024
       DR=R                                                             COV025
       DELTA=DR1 * .25                                                  COV026
   61 IF(REF - R - RT) 71,8,8                                           COV027
    8 F1R= 1.                                                           COV028
      GOTO 9                                                            COV029
   71 IF(RT - R - REF) 10,11,11                                         COV030
   11 F1R= T2                                                           COV031
      GOTO 9                                                            COV032
   10 TX= R / RT                                                        COV033
      CB=(TX*TX +T2 -1.)/(2.* TX * T1)                                  COV034
      IF(CB-1.0)12,13,13                                                COV035
 12   IF(CB+1.)14,14,15                                                 COV036
   15 B= ARCCOS(CB)                                                     COV037
      GO TO 16                                                          COV038
 13   B=0.                                                              COV039
      GO TO 16                                                          COV040
 14   B=3.14159                                                         COV041
   16 CD=(TX*TX -T2 +1.)/(2.* TX)                                       COV042
      IF(CD-1.)17,18,18                                                 COV043
 17   IF(CD+1.)19,19,20                                                 COV044
   20 D=ARCCOS(CD)                                                      COV045
      SD=(1.-CD*CD)**.5                                                 COV046
      GO TO 21                                                          COV047
 18   D=0.                                                              COV048
      SD=0.                                                             COV049
      GO TO 21                                                          COV050
 19   D=3.14159                                                         COV051
      SD=0.                                                             COV052
   21 F1R=(D+B *T2-TX*SD) * PII                                         COV053
 9    RHOR=W1*R*EXP(-W*R*R*CPSI)*CPSI                                   COV054
      T6=F1R*RHOR                                                       COV055
      FR=FR+T6                                                          COV056
      IF(ABS(R - RMAX) - DELTA) 22,22,23                                COV057
 23   R=R+DR                                                            COV058
      GO TO 61                                                          COV059
 22   EC1=FR*DR1                                                        COV060
  999 RETURN                                                            COV061
      END                                                               COV062
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *   OUT001
      SUBROUTINE OUTPUT                                                 OUT002
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *   OUT003
      DIMENSION ICOUNT(30),IRDCNT(50,30),HT(10),HA(10)                  OUT004
      COMMON(USE MAIN)                                                  OUT005
      WG=0. $ CT=0. $ CAS=0. $ TKS=0. $ APCS=0.  $  SQ=NQ               OUT006
      SACQ=NACQ                                                         OUT007
      WRITE(6,134)T,MIXID                                               OUT008
      WRITE(6,100)SACQ,SPERS,STANK,SAPC,AMWS                            OUT009
      WRITE(6,101)                                                      OUT010
      SACQ=NACQ - NRPD - NRW2                                           OUT011
      DO 10 I=1,NSYS                                                    OUT012
      IF(KSIG(I).EQ.0) GOTO 10                                          OUT013
      S(I,3)=S(I,3)+(S(I,4)-HT(I))*4.0+(S(I,5)-HA(I))*15.0              OUT014
      WRITE(6,102)SYSID(I),S(I,2),S(I,1),S(I,3),S(I,4),S(I,5)           OUT015
      HT(I)=S(I,4)                                                      OUT016
      HA(I)=S(I,5)                                                      OUT017
      WG=WG+S(I,1)                                                      OUT018
```

```
      CT=CT+S(I,2)
      CAS=CAS+S(I,3)
      TKS=TKS + S(I,4)
      APCS=APCS + S(I,5)
   10 CONTINUE
      PCTQ=(SQ/(SACQ + .000001)) * 100.
      WRITE(6,103)CT,WG,CAS,TKS,APCS,PCTQ,SMW
      WRITE(6,104)
      DO 3 I=1,30
    3 ICOUNT(I)=I
      WRITE(6,106)(ICOUNT(I),I=1,30)
      DO  2 I=1,NRDS
      DO 2 J=1,30
    2 IRDCNT(I,J)=RDCNT(I,J)
      DO  1 I=1,NRDS
      RSUM= 0.0
      IF(KRIG(I).EQ.0) GOTO 1
      WRITE(6,105)RNDID(I),(IRDCNT(I,J),J=1,30)
      DO 20 K=1,30
   20 RSUM = RSUM + RDCNT(I,K)
      WRITE(6,116) RSUM
    1 CONTINUE
      NRACQ= NACQ - NOM
      NRTGT= NTGT -NOM
      NRQ=NQ - NQOM
      NQAL= NQ + KOUNT
      NRQL=NQAL - NQOM
      NRF=NFM - NOMF
      NTD= NFMD + NOMF
      NRFL=NRF - NFMD
      NREAC= NACQ - NTGT
      WRITE(6,107)
      WRITE(6,108) NOM,NRACQ,NACQ
      WRITE(6,109) NOM,NRTGT,NTGT
      WRITE(6,113) NQOM,NRQ,NQ
      WRITE(6,114) NQOM,NRQL,NQAL
      WRITE(6,110)NOMF,NRF,NFM
      WRITE(6,111) NOMF,NFMD,NTD
      WRITE(6,112) NRFL,NRFL
      WRITE(6,115) NQLP,NQD,NQOM,KOUNT,NRPD,NRW2,NREAC
      IF(T.LT.TMX) GOTO 999
      DO  5 I=1,NSYS
      HT(I)=0.
    5 HA(I)=0.
  999 RETURN
  100 FORMAT(5H ACQ=,F10.2,13X,5HPERS=,F10.2,3X,5HTANK=,F10.2,3X,4HAPC=,
     1F10.2,3X,10HMIL WORTH=,F10.2/)
  101 FORMAT(11H     SYSTEM, 8X,4HCOST,10X,6HWEIGHT,8X,9HPERSONNEL,8X,5H
     1TANKS,10X,4HAPCS)
  102 FORMAT(2X,F8.2,5X,F10.4,5X,F10.4,5X,F10.4,5X,F10.4,5X,F10.4)
  103 FORMAT(/12H   TOTALS    ,4X,F10.4, 5X,F10.4, 5X,F10.4, 5X,F10.4, 5X
     1,F10.4,5X,6HPCTQ =,F10.4,2X,3HMW=,F8.2)
  104 FORMAT(//,56X,19HRANGE IN KILOMETERS/)
  105 FORMAT(1H ,F8.2,30I4)
  106 FORMAT(9H ROUND ID,1X,I2,29(2X,I2))
  107 FORMAT(//,35X,6H OTHER,9X,7HREGULAR,9X,6HTOTALS,/,34X,8HMISSIONS,
     18X,7HTARGETS,/)
  108 FORMAT(16H   ACQUISITIONS=,19X,3(I5,11X),/)
  109 FORMAT(18H   NO. OF TARGETS=,17X,3(I5,11X),/)
  110 FORMAT(22H   NO. MSN/TGTS FIRED=,13X,3(I5,11X),/)
```

```
  111 FORMAT(25H   NO. MSN/TGTS DEFEATED=,10X,3(I5,11X),/)
  112 FORMAT(22H   TGT FIRED BUT LOST=,17X,1H0,2(11X,I5),/)
  113 FORMAT(23H   NO. MSN/TGTS QUEUED=,12X,3(I5,11X),/)
  114 FORMAT(30H   SUM QUEUED + STILL ON LIST=,5X,3(I5,11X),//)
  115 FORMAT(67H   QUEUED MISSION TOTAL INCLUDES THOSE DROPPED DUE TO LO
     1W PRIORITY(,I5,35H), THOSE DEPARTED BEFORE ATTEMPT TO,/,6H FIRE(,
     2I5,45H), AND OTHER-TYPE MISSIONS TRIED BUT CANT DO(,I5,2H).,//,
     331H NO. OF TGTS STILL ON TGT LIST=,I5,1H.,//,54H NO. OF PREVIOUSLY
     4 DEFEATED TGTS WHICH ARE REACQUIRED=,I5,1H.,//,45H NO. OF TARGETS
     5COMBINED (WITHIN 200 METERS)=,I5,1H.,//,29H TOTAL OF ALL REACQUISI
     6TIONS=,I5,1H.,//)
  116 FORMAT(8X,9H RNDSUM =,F12.4)
  134 FORMAT(6H1TIME=,F8.4,6X,16A5,/)
      END
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      SUBROUTINE SPECIL(IT)
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      COMMON (USE MAIN)
      MORG=TN(15,IT) $ MORGT1=0 $ MORGT2=0 $ MORGT3=0
   20 IKR=0 $ NAK=0 $ CSTM=1000000.0
      DO 1 IFS=1,NFU
      IF(KFIG(IFS).NE.MORG)GOTO 1
      IF(TUBFU(IFS).GE.(T+DELT-.00001)) GOTO 1
      DO 2 ISS=1,NSYS
      IF(KSIG(ISS).EQ.0)GOTO 2
      IF(FSID(IFS).EQ.SYSID(ISS))GOTO 21
    2 CONTINUE
      WRITE(6,104)FSID(IFS),MORG
  104 FORMAT(12H FIRE SYSTEM,F8.2,25HIS NOT IN LIST OF SYSTEMS,9HECHELON
     1 =,I5)
      STOP
   21 NA= NA + 1
      A(27,NA)=IFS
      A(1,NA)=ISS
      A(2,NA)=TUBFU(IFS)
      R=HBLD(ISS)+RSPY(ISS)*(T-TZRO)
      Z=R-FT(6,IFS)
      IF(TN(14,IT).GE.10.0)Z=AMIN1(R+QBLD(ISS),24.0*RSPY(ISS)+HBLD(ISS))
     1-FT(6,IFS)
      R=HNMX(ISS)-FT(6,IFS)+FT(2,IFS)
      R=AMIN1(R,Z)
      NS=NSITE(IFS)
      DO 3 INSS=1,NS
      IF(T.LT.TA(INSS,IFS))GOTO 22
      IF(TA(INSS,IFS).LE.T.AND.T.LE.TD(INSS,IFS))GOTO 23
    3 CONTINUE
      WRITE(6,100)IFS
  100 FORMAT(16H ERROR FIRE UNIT,I5,39HDOES NOT HAVE A TIME TO MATCH GAM
     1E TIME)
      STOP
   22 A(5,NA)=FRWM(ISS)
      GOTO 24
   23 A(5,NA)=1.0
   24 A(6,NA)=XS(INSS,IFS)
      A(7,NA)=YS(INSS,IFS)
      IF(TN(8,IT).EQ.TN(9,IT))GOTO 25
      Z=SNMX(ISS)*A(5,NA)
      A(3,NA)=SROF(ISS)
      GOTO 26
   25 Z=DNMX(ISS)*A(5,NA)
```

```
      A(3,NA)=DROF(ISS)                                                 SPC047
   26 R=AMIN1(R,Z)                                                      SPC048
      A(4,NA)=R                                                         SPC049
      A(8,NA)=MORG                                                      SPC050
      A(9,NA)=(A(6,NA)-TN(11,IT))**2+(A(7,NA)-TN(12,IT))**2             SPC051
      IF(A (4,NA) .LE. 0.) GOTO 1                                       SPC052
      K=ISS                                                             SPC053
      GOTO(40,40,41,41,41,42,43,44,45,46),K                             SPC054
   40 K=2                                                               SPC055
      GOTO 19                                                           SPC056
   41 K=3                                                               SPC057
      GOTO 19                                                           SPC058
   42 K=4                                                               SPC059
      GOTO 19                                                           SPC060
   43 K=5                                                               SPC061
      GOTO 19                                                           SPC062
   44 K=6                                                               SPC063
      GOTO 19                                                           SPC064
   45 K=7                                                               SPC065
      GOTO 19                                                           SPC066
   46 K=8                                                               SPC067
   19 DO 10 IST=1,ITC                                                   SPC068
      IF(TN(1,IT).NE.AMSN(IST,1))GOTO 10                                SPC069
      RDNO=AMSN(IST,K)                                                  SPC070
      A(13,NA)=RDNO                                                     SPC071
      GOTO 11                                                           SPC072
   10 CONTINUE                                                          SPC073
      WRITE(6,103)TN(1,IT)                                              SPC074
  103 FORMAT(7H TARGET,F8.1,46HIS A SPECIAL MISSION BUT NO RDS ASSIGNED SPC075
     1TO IT)                                                            SPC076
      STOP                                                              SPC077
   11 IF(A(4,NA).LT.RDNO)GOTO 1                                         SPC078
      DO 4 IRS=1,NRDS                                                   SPC079
      IF(KRIG(IRS).EQ.0)GOTO 4                                          SPC080
      DO 5 LS=1,NSYS                                                    SPC081
      IF(SYSID(ISS).EQ.SYSRD(LS,1))GOTO 27                              SPC082
    5 CONTINUE                                                          SPC083
      WRITE(6,102)SYSID(ISS)                                            SPC084
  102 FORMAT(29H ERROR, UNDEFINED SYSTEM. ID=,F10.2)                    SPC085
      STOP                                                              SPC086
   27 DO 6 MS=2,10                                                      SPC087
      IF(RNDID(IRS).EQ.SYSRD(LS,MS))GOTO 28                             SPC088
    6 CONTINUE                                                          SPC089
      GOTO 4                                                            SPC090
   28 IF( A(9,NA).GT.R2MX(IRS))GOTO 4                                   SPC091
      TCST=CST(IRS)*RDNO                                                SPC092
      IF(TCST.GE.CSTM) GOTO 4                                           SPC093
      CSTM=TCST                                                         SPC094
      A(24,NA )=IRS                                                     SPC095
      JFS=NA                                                            SPC096
    4 CONTINUE                                                          SPC097
      IF(CSTM.GE.1000000.0)GOTO 1                                       SPC098
      NAK=NAK+1                                                         SPC099
    1 CONTINUE                                                          SPC100
      IF(NAK.EQ.0)GOTO 29                                               SPC101
      IXZ=A(27,JFS)                                                     SPC102
      IKR=A(24,JFS)                                                     SPC103
      RDNO=A(13,JFS)                                                    SPC104
      FT(6,IXZ)=FT(6,IXZ)+RDNO                                          SPC105
      ISS=A(1,JFS)                                                      SPC106
```

```
      A9=SQRT(A(9,JFS))
      IRNG=A9
      RNGINT=IRNG
      RNGINT=RNGINT+.5
      IF(A9.GE.RNGINT)IRNG=IRNG+1
      IF(A9.GE.30.0)IRNG=30
      RDCNT(IKR,IRNG)=RDCNT(IKR,IRNG)+RDNO
      S(ISS,1)=S(ISS,1)+WGT(IKR)*RDNO
      S(ISS,2)=S(ISS,2)+CST(IKR)*RDNO
      TUBFU(IXZ)=TUBFU(IXZ)+TBM(ISS)*A(5,JFS)
      IF(STYP(ISS).EQ.2.0)GOTO 9
      B=(RDNO-A(5,JFS)*TPFU(ISS))/(A(5,JFS)*TPFU(ISS)*A(3,JFS))
      B=AMAX1(0.0,B)
      TUBFU(IXZ)=TUBFU(IXZ)+B
    9 SMW=SMW + TN(14,IT)
      DEFSP=1.
      GOTO 999
   29 IF(MORG.EQ.1)GOTO 30
      IF(MORG.EQ.2)GOTO 31
      IF(MORG.EQ.3)GOTO 32
   30 IF(MORGT2.EQ.1)GOTO 33
      MORGT2=1
      MORGT1=1
      MORG=2
      GOTO 34
   33 IF(MORGT3.EQ.1)GOTO 999
      MORGT1=1
      MORGT3=1
      MORG=3
      GOTO 34
   31 IF(MORGT1.EQ.1)GOTO 35
      MORGT2=1
      MORGT1=1
      MORG=1
      GOTO 34
   35 IF(MORGT3.EQ.1)GOTO 999
      MORGT2=1
      MORGT3=1
      MORG=3
      GOTO 34
   32 IF(MORGT2.EQ.1)GOTO 36
      MORGT3=1
      MORGT2=1
      MORG=2
      GOTO 34
   36 IF(MORGT1.EQ.1)GOTO 999
      MORGT3=1
      MORGT1=1
      MORG=1
   34 DO 8 IC=1,NA
      DO 8 ICR=1,27
      A(ICR,IC)=0.0
    8 CONTINUE
      NA=0
      GOTO 20
  999 RETURN
      END
```


The following page is blank.

APPENDIX B

PROGRAM FLOW CHARTS

The following page is blank.

This appendix provides a flow chart of the Legal Mix main program and each of the sub-routines contained in the model. In addition, a verbal description of the program accompanies the flow charts, to assist the reader in following the program logic.

Program Element: Main Program

Symbolic Name: Main Program

Arguments in Call Statement: Not Applicable

Subroutines which call Main Program: Not Applicable

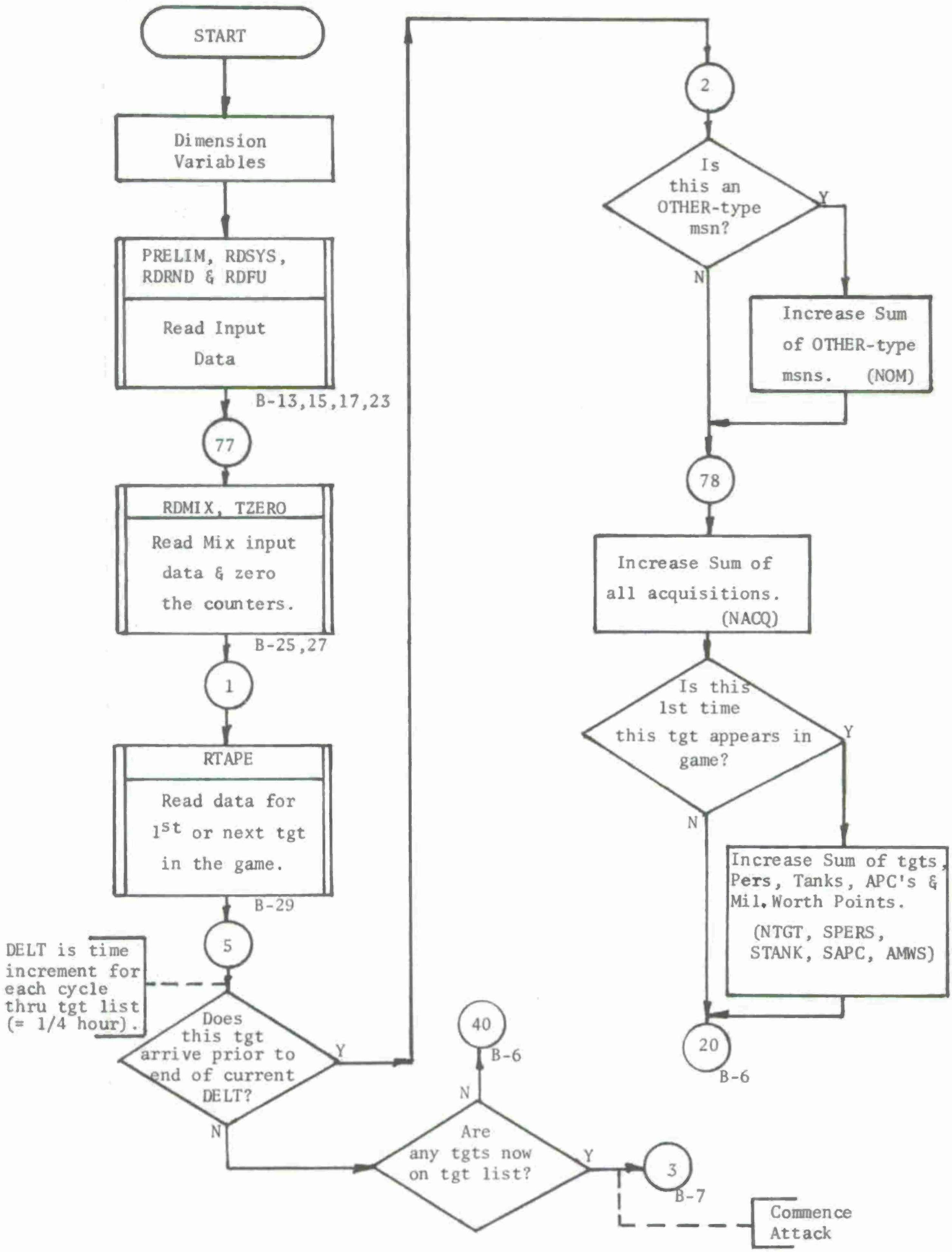
Subroutines called by Main Program:

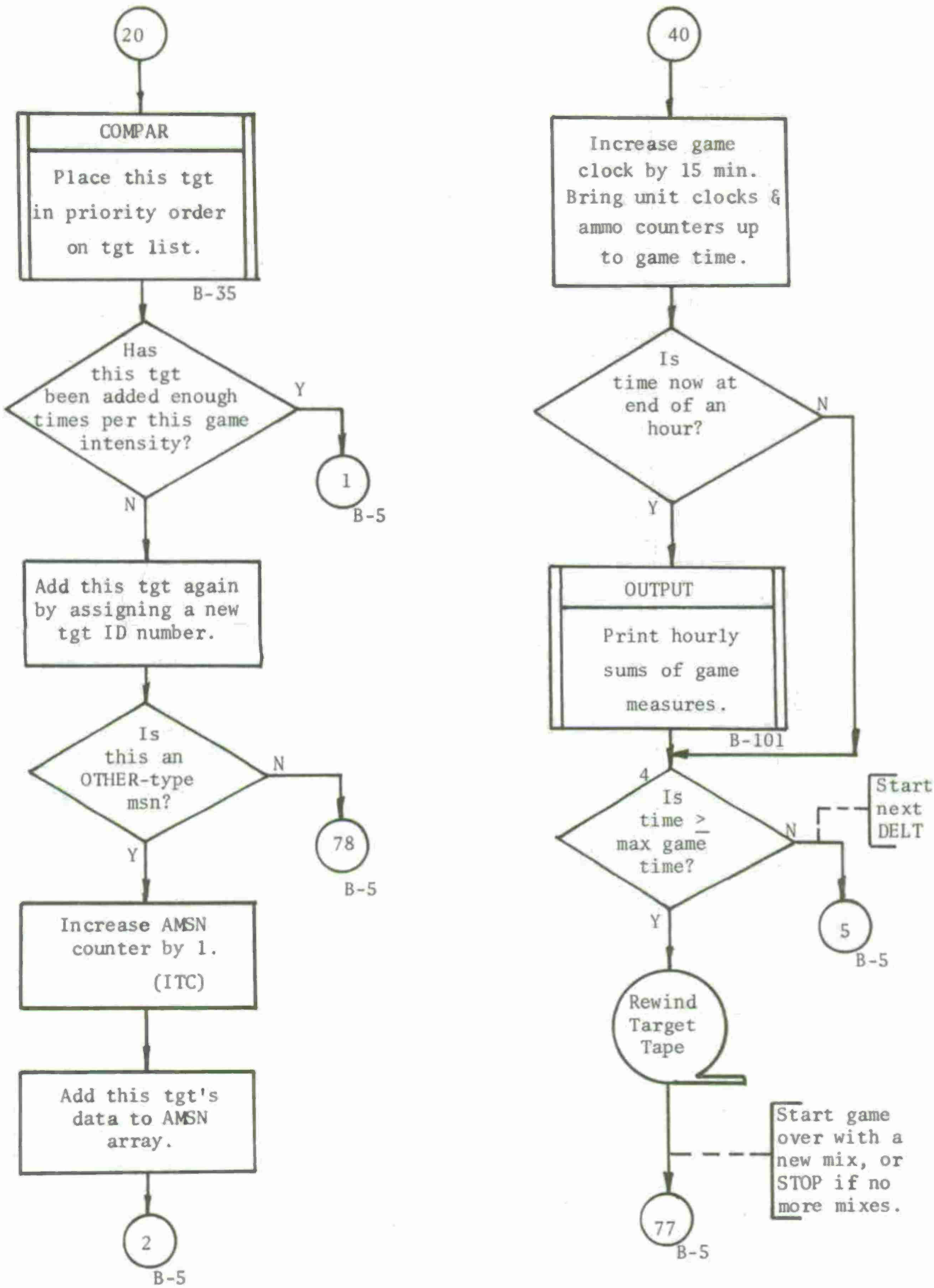
PRELIM, RDSYS, RDRND, RDFU, RDMIX, TZERO, RTAPE, COMPAR, SPECIL, REMOVE, DIRSUP, DIVISN, CORP, OUTPUT

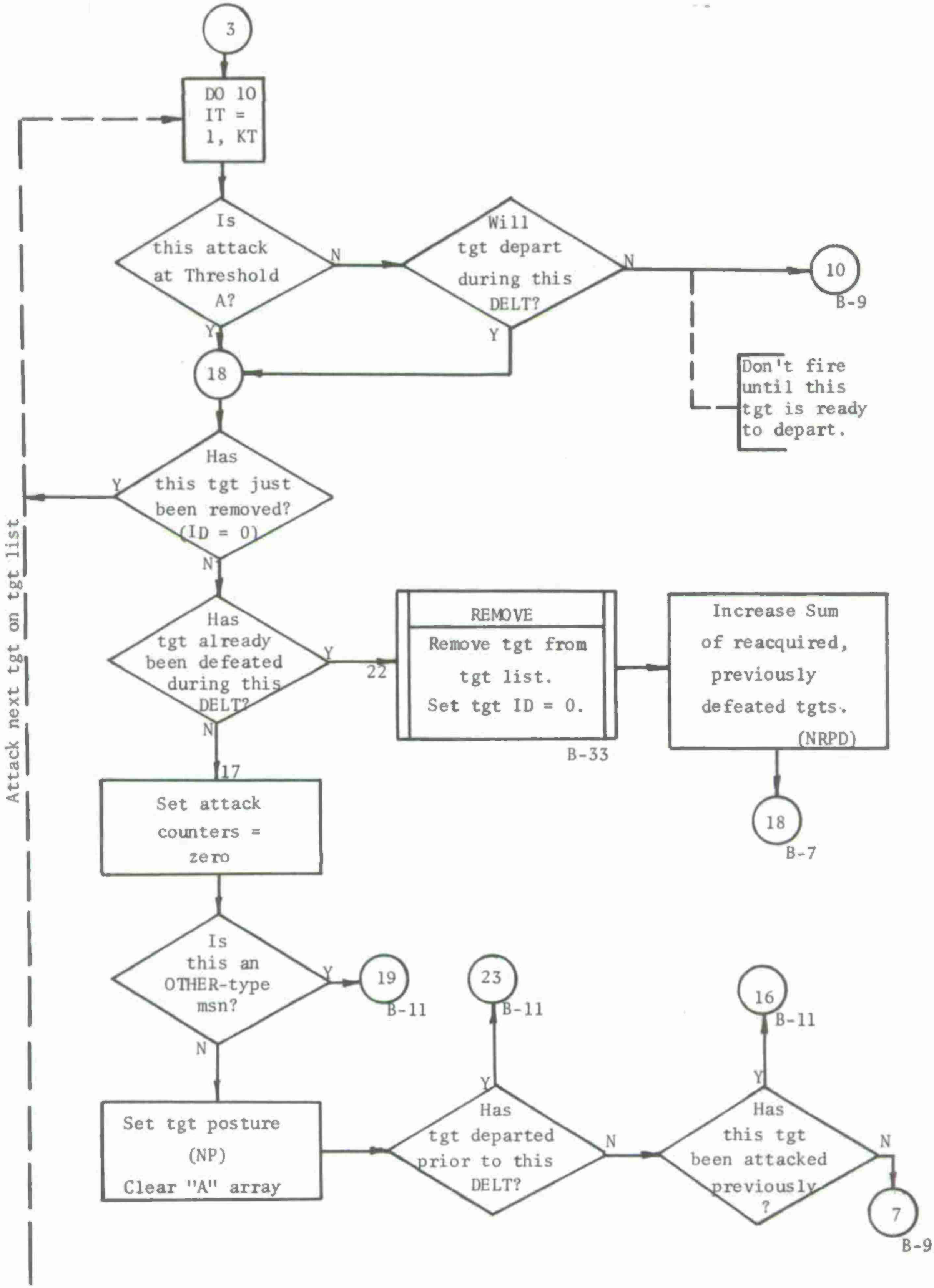
The main program initially provides for the reading of preliminary data (PRELIM) and weapon system, round and fire unit data (RDSYS, RDRND, RDFU) from input cards. The game "mix" of systems, rounds and fire units is then input (RDMIX) and various counters, arrays and clocks are initialized (TZERO). For each 15-minute game interval, the program then reads in from tape the target data for all targets arriving during the 15-minute interval (RTAPE), and as each target is input it is placed on the target array TN(I,J) in priority order (COMPAR).

Each target is added the appropriate number of times (according to game intensity) and counters for target acquisitions, target types, and target composition (personnel, tanks and APC's) are increased. When all targets for a given 15-minute interval have been input, the program then attempts to attack each target on the priority-ordered target array. If a given target has been previously defeated, it is removed from the target list (REMOVE); and if a target has been previously attacked but not defeated, previous damage inflicted is charged to the target.

If the target is a special (or "other") mission type, i.e. a smoke, illumination or Harassment & Interdiction (H&I) mission, subroutine SPECIL is then called to attempt accomplishment of the mission.


START
Dimension Variables
PRELIM, RDSYS, RDRND & RDFU
Read Input Data
B-13,15,17,23
77
RDMIX, TZERO
Read Mix input data & zero the counters.
B-25,27
1
RTAPE
Read data for 1st or next tgt in the game.
B-29
5
DELT is time increment for each cycle thru tgt list (= 1/4 hour).
Does this tgt arrive prior to end of current DELT?
Y
N
Are any tgts now on tgt list?
Y
N
40
B-6
3
B-7
Commence Attack
2
Is this an OTHER-type msn?
Y
N
Increase Sum of OTHER-type msns. (NOM)
78
Increase Sum of all acquisitions. (NACQ)
Is this 1st time this tgt appears in game?
Y
N
Increase Sum of tgts, Pers, Tanks, APC's & Mil. Worth Points.
(NTGT, SPERS, STANK, SAPC, AMWS)
20
B-6


20
COMPAR
Place this tgt in priority order on tgt list.
B-35
Has this tgt been added enough times per this game intensity?
Y
1
B-5
N
Add this tgt again by assigning a new tgt ID number.
Is this an OTHER-type msn?
N
78
B-5
Y
Increase AMSN counter by 1. (ITC)
Add this tgt's data to AMSN array.
2
B-5
40
Increase game clock by 15 min. Bring unit clocks & ammo counters up to game time.
Is time now at end of an hour?
N
Y
OUTPUT
Print hourly sums of game measures.
B-101
4
Is time ≥ max game time?
N
Start next DELT
5
B-5
Y
Rewind Target Tape
Start game over with a new mix, or STOP if no more mixes.
77
B-5

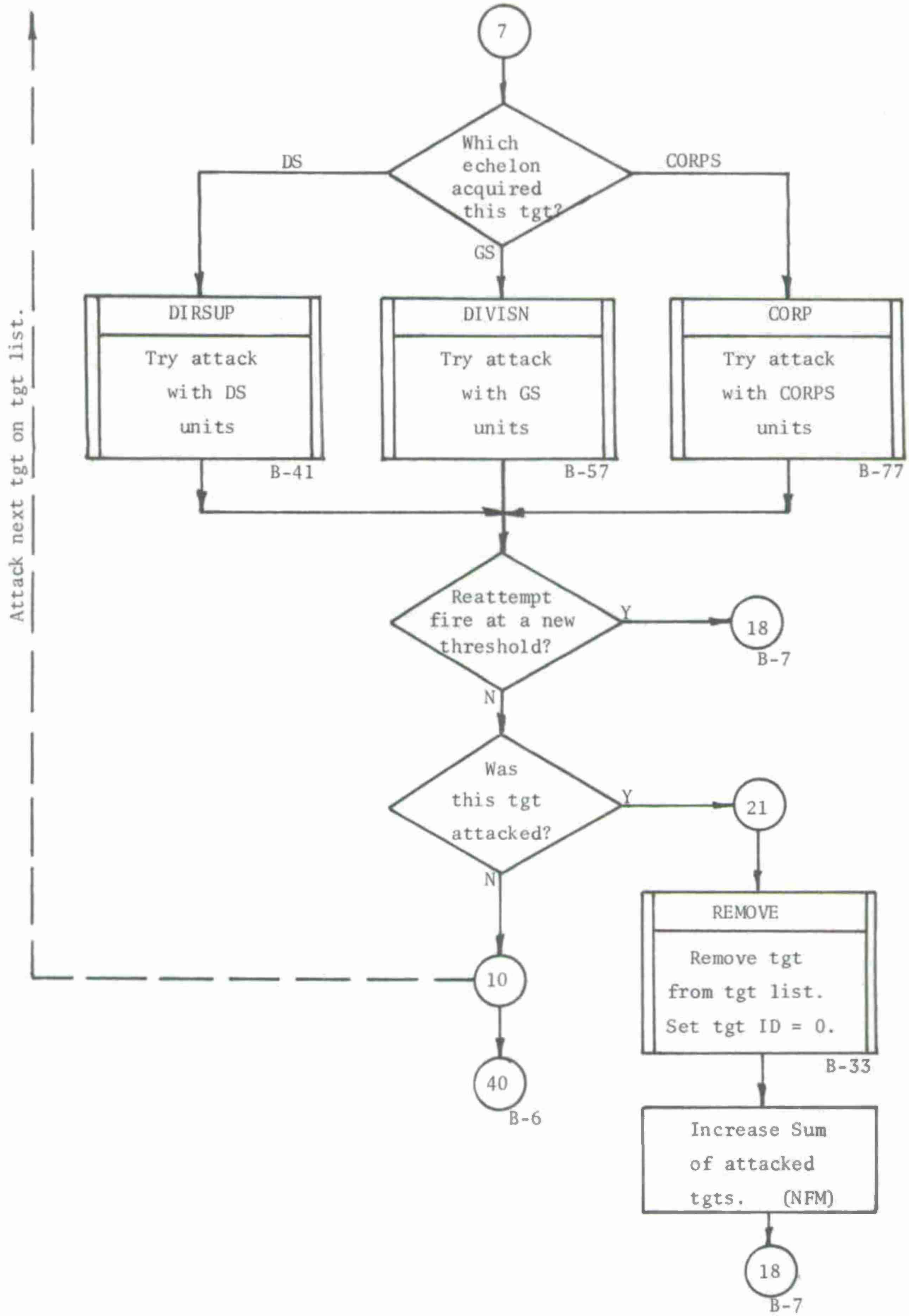

3
DO 10
IT =
1, KT
Is
this attack
at Threshold
A?
N
Will
tgt depart
during this
DELT?
N
10
B-9
Y
Y
18
Don't fire
until this
tgt is ready
to depart.
Has
this tgt just
been removed?
(ID = 0)
Y
N
Attack next tgt on tgt list
Has
tgt already
been defeated
during this
DELT?
Y
22
REMOVE
Remove tgt from
tgt list.
Set tgt ID = 0.
B-33
Increase Sum
of reacquired,
previously
defeated tgts.
(NRPD)
18
B-7
N
17
Set attack
counters =
zero
Is
this an
OTHER-type
msn?
Y
19
B-11
23
B-11
16
B-11
N
Set tgt posture
(NP)
Clear "A" array
Y
Has
tgt departed
prior to this
DELT?
N
Y
Has
this tgt
been attacked
previously
?
N
7
B-9

For regular targets, either Direct Support (DIRSUP), General Support (DIVISN) or Corps (CORP) subroutines are called to attempt engagement, depending upon the echelon (DS, GS or CORPS) which acquired the target. If a regular target cannot be attacked at Threshold A attack level, engagement is re-attempted at a lower attack level (Threshold B).

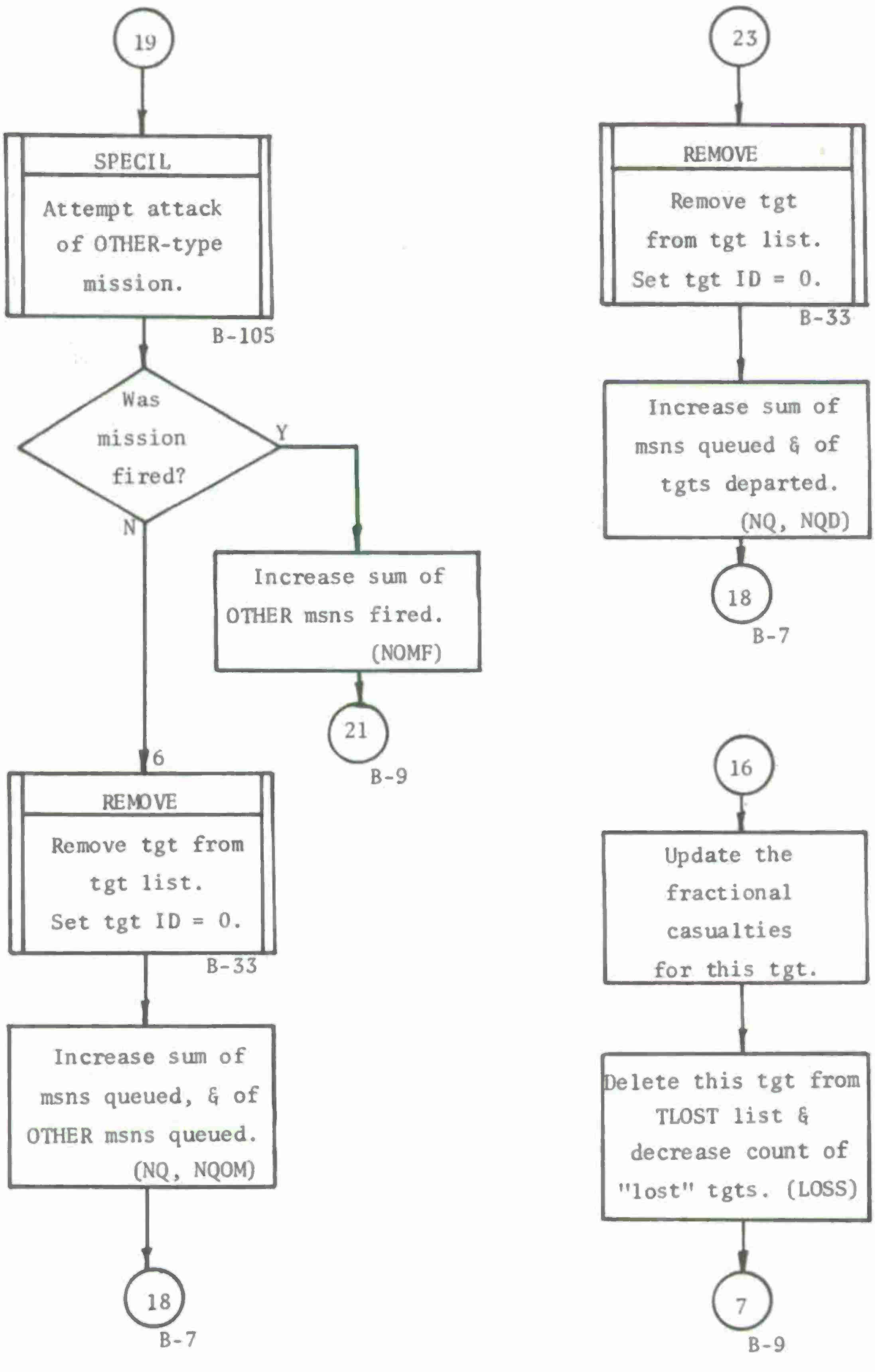
If a regular target is attacked, it is removed from the target array and the attacked-target counter is increased. For special missions, the target is removed whether fired or not, with appropriate counter (s) being increased.

After all targets on the list for a given 15-minute game increment have been considered, appropriate game and fire unit (FU) clocks are then increased by 15 minutes, and FU ammo counters are incremented. The program then begins the next 15-minute game interval by reading in the targets for that interval. At the end of four such cycles (every game hour) subroutine OUTPUT is called to provide detailed data for analysis.

The program continues this cycle until the game clock exceeds the input maximum game time, at which point the target tape is re-wound and subroutine RDMIX is called to specify the system-round-fire unit mix for the next play of the game. If no additional mixes are to be played, the program stops.



The following page is blank.


19
SPECIL
Attempt attack of OTHER-type mission.
B-105
Was mission fired?
Y
N
Increase sum of OTHER msns fired. (NOMF)
21
B-9
6
REMOVE
Remove tgt from tgt list. Set tgt ID = 0.
B-33
Increase sum of msns queued, & of OTHER msns queued. (NQ, NQOM)
18
B-7
23
REMOVE
Remove tgt from tgt list. Set tgt ID = 0.
B-33
Increase sum of msns queued & of tgts departed. (NQ, NQD)
18
B-7
16
Update the fractional casualties for this tgt.
Delete this tgt from TLOST list & decrease count of "lost" tgts. (LOSS)
7
B-9

Program Element: Preliminary Initialization

Symbolic Name: PRELIM

Arguments in Call Statement: None

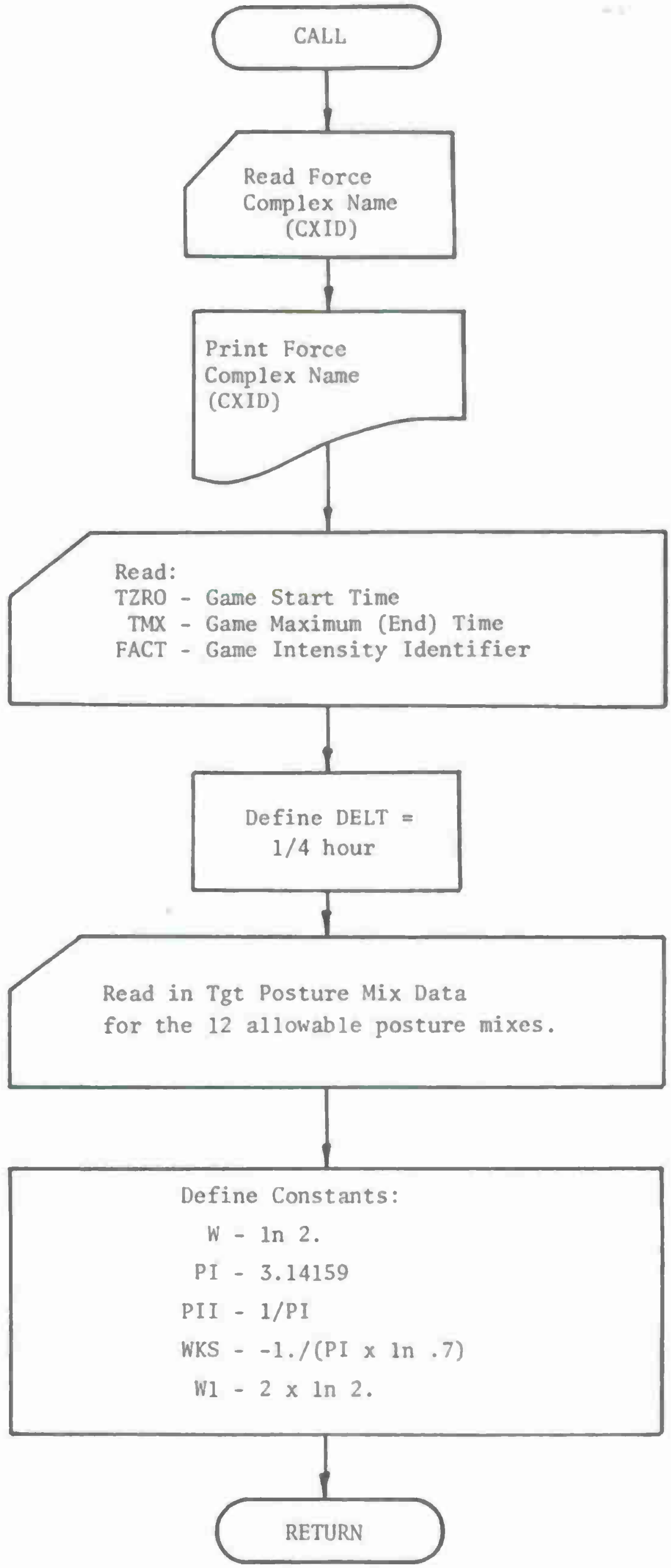
Subroutines which call PRELIM: Main Program

Subroutines called by PRELIM: None

This subroutine is called at the start of the main program and provides for card input of a game force identifier code, game time parameters, definition of game intensity, and definition of allowable target posture mixes. See Card Sets 1, 2 and 3 for explanation of inputs. (Table 3.1)

PRELIM also defines constants used elsewhere in the program.

Subroutine PRELIM



Program Element: Read Input for Systems

Symbolic Name: RDSYS

Arguments in Call Statement: None

Subroutines which call RDSYS: Main Program

Subroutines called by RDSYS: None

This subroutine provides for the card input of 12 parameters which define each weapon system allowed in the game. In addition, it converts the input units for six of these parameters to units needed in the program and also calculates 1/4 and 1/2 ammunition basic loads from the basic load (which is input). See card sets 4 and 5, Table 3.2.

Subroutine RDSYS

CALL

Read the number of systems in the force: (NSYS)

Print NSYS

Do 1 I = 1,NSYS

Read for this system:

NAME	MEANING	INPUT UNITS
SYSID	System ID Number (e.g., 1200.1)	--
FRWM	Fraction of Unit in Place During Move	decimal
TPFU	Tubes/Launchers per Fire Unit	tubes/unit
*SROF	Max Rate of Fire Vs Static Tgt	rd/min/tube
*DROF	Max Rate of Fire Vs Moving Tgt	rd/min/tube
*TBM	Time Between Missions	minutes
BLD	Basic Load of Ammo per Fire Unit	rd/unit
RSPY	Resupply Rate for Ammo	rd/unit/hr
*SNMX	Max Rd Allowed per Msn Vs Static Tgt	rd/tube/msn
*DNMX	Max Rd Allowed per Msn Vs Moving Tgt	rd/tube/msn
*HNMX	Max Rd per Hour Vs all Tgts	rd/tube/hr
STYP	System Type (1. = Cannon; 2. = Missile)	--

*Units Converted Below

Calculate for this system:

HBLD	1/2 Basic Load	rd/unit
QBLD	1/4 Basic Load	rd/unit
TBM	Time Between Missions	decimal hrs
SROF	Max Rate of Fire Vs Static Tgt	rd/hour/tube
DROF	Max Rate of Fire Vs Moving Tgt	rd/hour/tube
SNMX	Max Rd Allowed per Msn Vs Static Tgt	rd/unit/msn
DNMX	Max Rd Allowed per Msn Vs Moving Tgt	rd/unit/msn
HNMX	Max Rd per Hour Vs all Tgts	rd/unit/hr

1

Read data for next system

RETURN

Program Element: Read Input for Rounds

Symbolic Name: RDRND

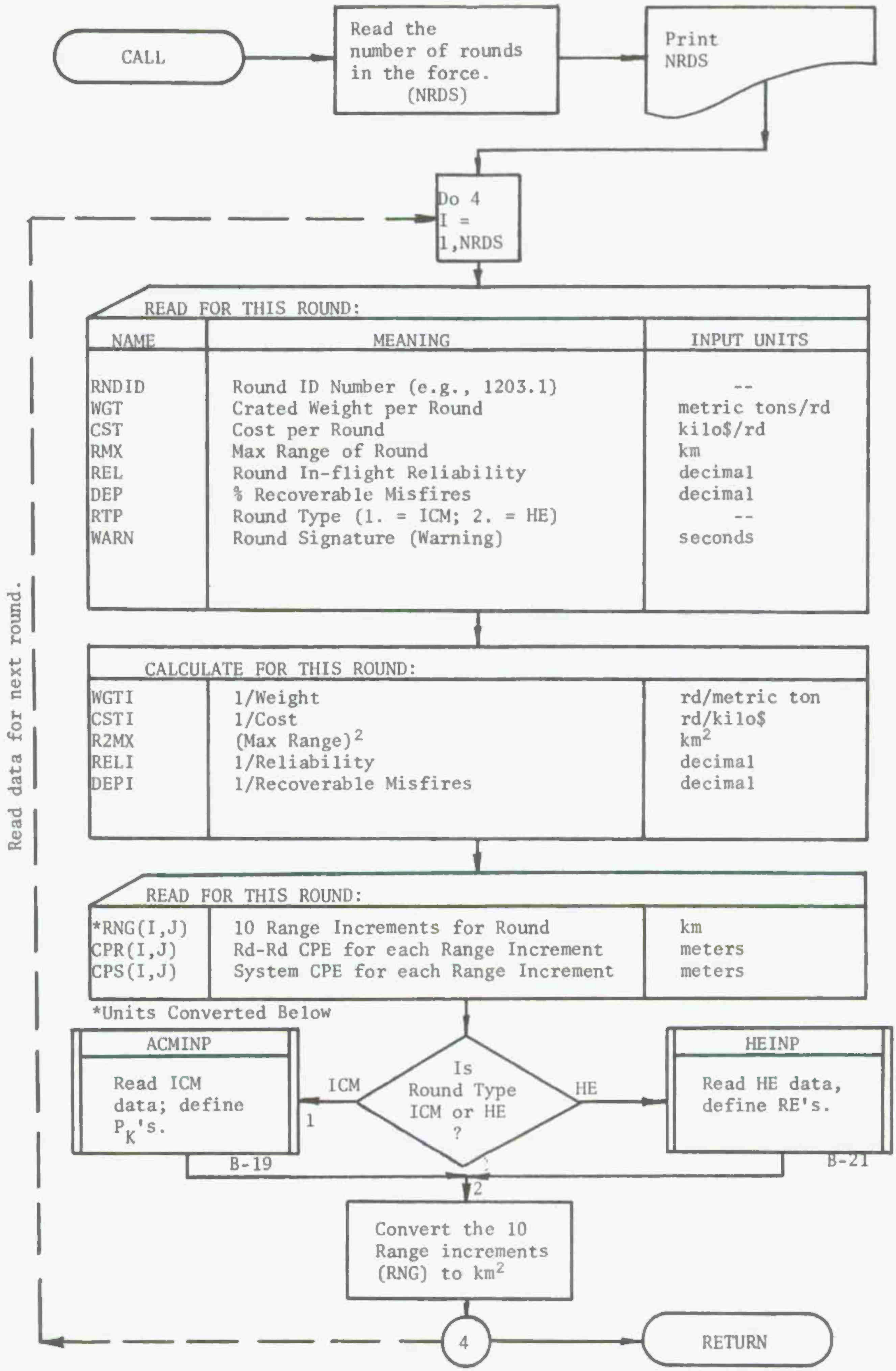
Arguments in Call Statement: None

Subroutines which call RDRND: Main Program

Subroutines called by RDRND: HEINP, ACMINP

This subroutine provides for the card input of parameters which define each round of ammunition allowed in the game. The number of parameters to be input depends on the type of round: Improved Conventional Munitions (ICM) or High Explosive (HE). RDRND calculates 5 parameters used in the program and calls either ACMINP (for ICM) or HEINP (for HE) subroutine to complete the read-in of round data. As a final step for each round, RDRND converts input ranges in kilometers to ranges-squared (KM^2) for use in the program. See card sets 6 and 7, Table 3.3.

Subroutine RDRND



Program Element: Read ICM Input

Symbolic Name: ACMINP

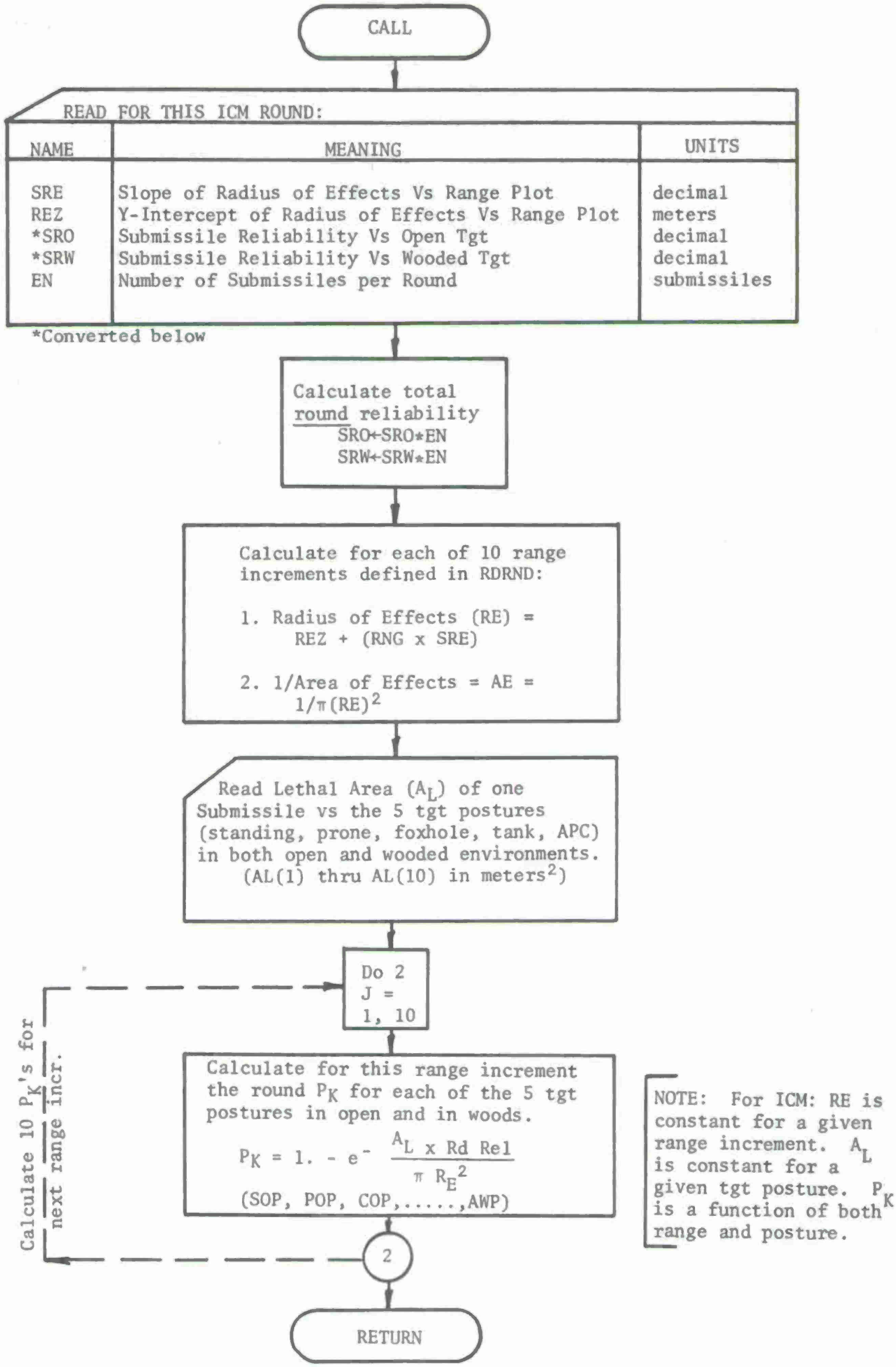
Argument in Call Statement: (I) - Identifies the position (in the list of rounds) of the round being input.

Subroutines which call ACMINP: RDRND

Subroutines called by ACMINP: None

This subroutine provides for the card input of 15 parameters which define each Improved Conventional Munition (ICM) allowed in the game. Based on these inputs, the subroutine then calculates radii of effects and 1/area of effects at input range increments, and then calculates the Probability of Kill (P_K) at each range increment for each of five target postures in both the open and wooded environments, i.e. 10 P_K's at each range increment.

Subroutine ACMINP



Program Element: Read HE Input

Symbolic Name: HEINP

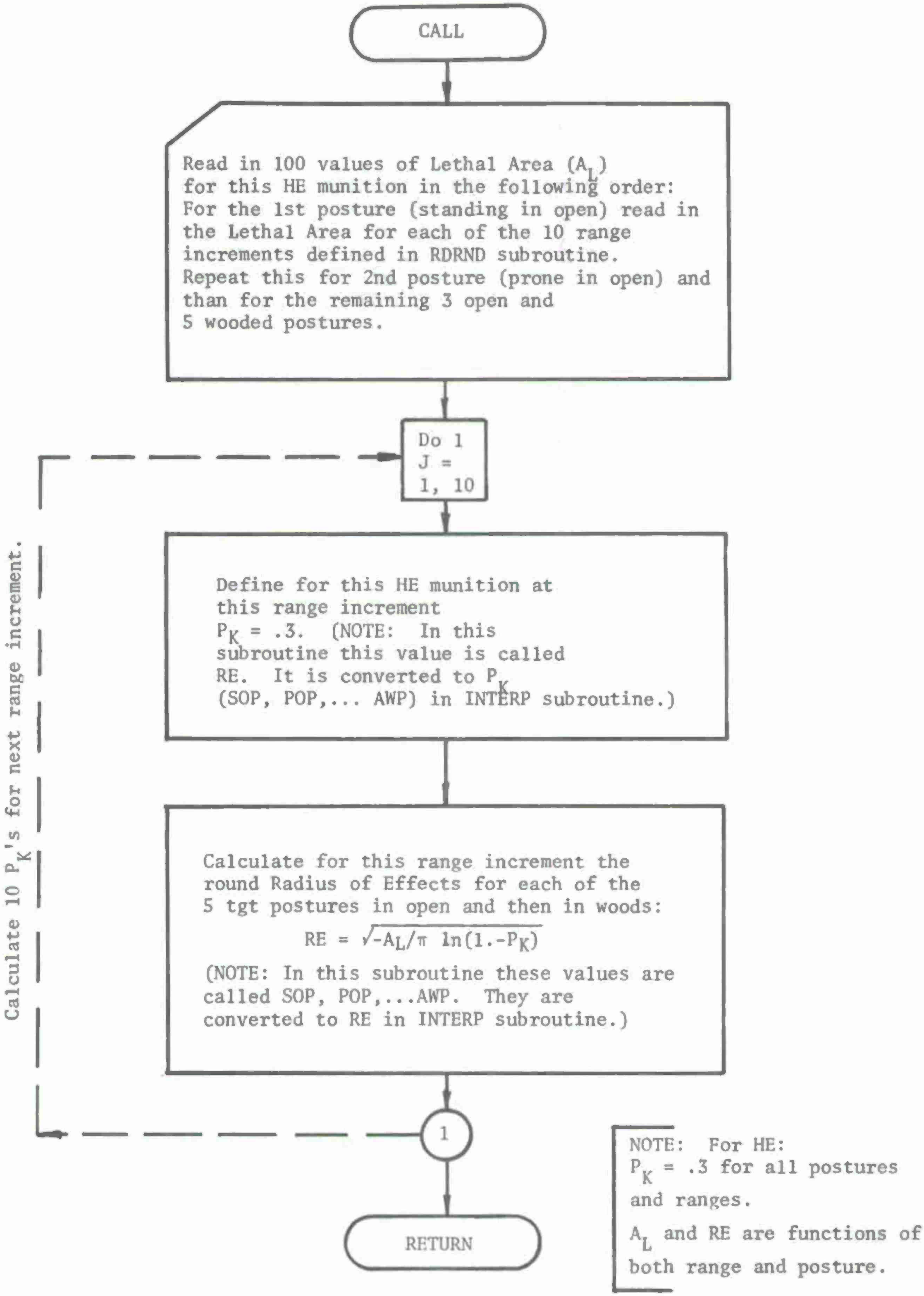
Argument in Call Statement: (I) - Identifies the position (in the list of rounds) of the round being input.

Subroutines which call HEINP: RDRND

Subroutines called by HEINP: None

This subroutine provides for the card input of 100 lethal areas (for 5 postures in both open and wooded environments at each of 10 range increments) for each High Explosive (HE) round allowed in the game. Based on these inputs, the subroutine then calculates the radius of effects associated with each lethal area input.

Subroutine HEINP



Program Element: Read Input for Fire Units

Symbolic Name: RDFU

Arguments in Call Statement: None

Subroutines which call RDFU: Main Program

Subroutines called by RDFU: None

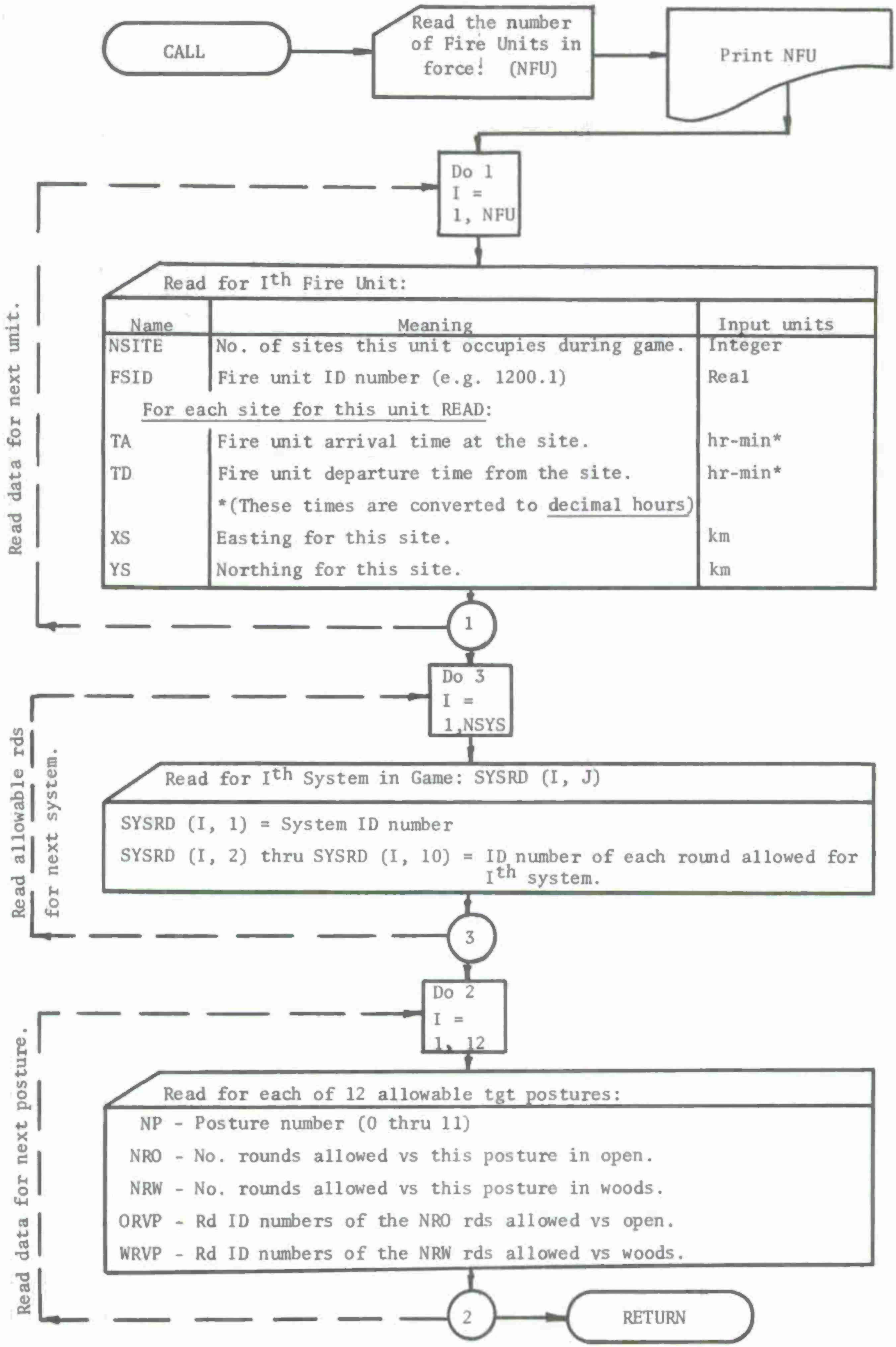
This subroutine provides for the card input of fire unit parameters for each fire unit allowed in the game, to include the fire unit's weapon system and times and locations of each site occupied by the fire unit during the game.

Then for each weapon system in the game this subroutine reads which of the input rounds is allowed for use by each system.

Finally, for each posture mix defined in PRELIM, the rounds allowed against each posture mix are read in.

See card sets 8, 9, 10 and 11, Table 3.4.

Subroutine RDFU



Program Element: Read Input for Mix of Systems, Rounds and Fire Units

Symbolic Name: RDMIX

Arguments in Call Statement: None

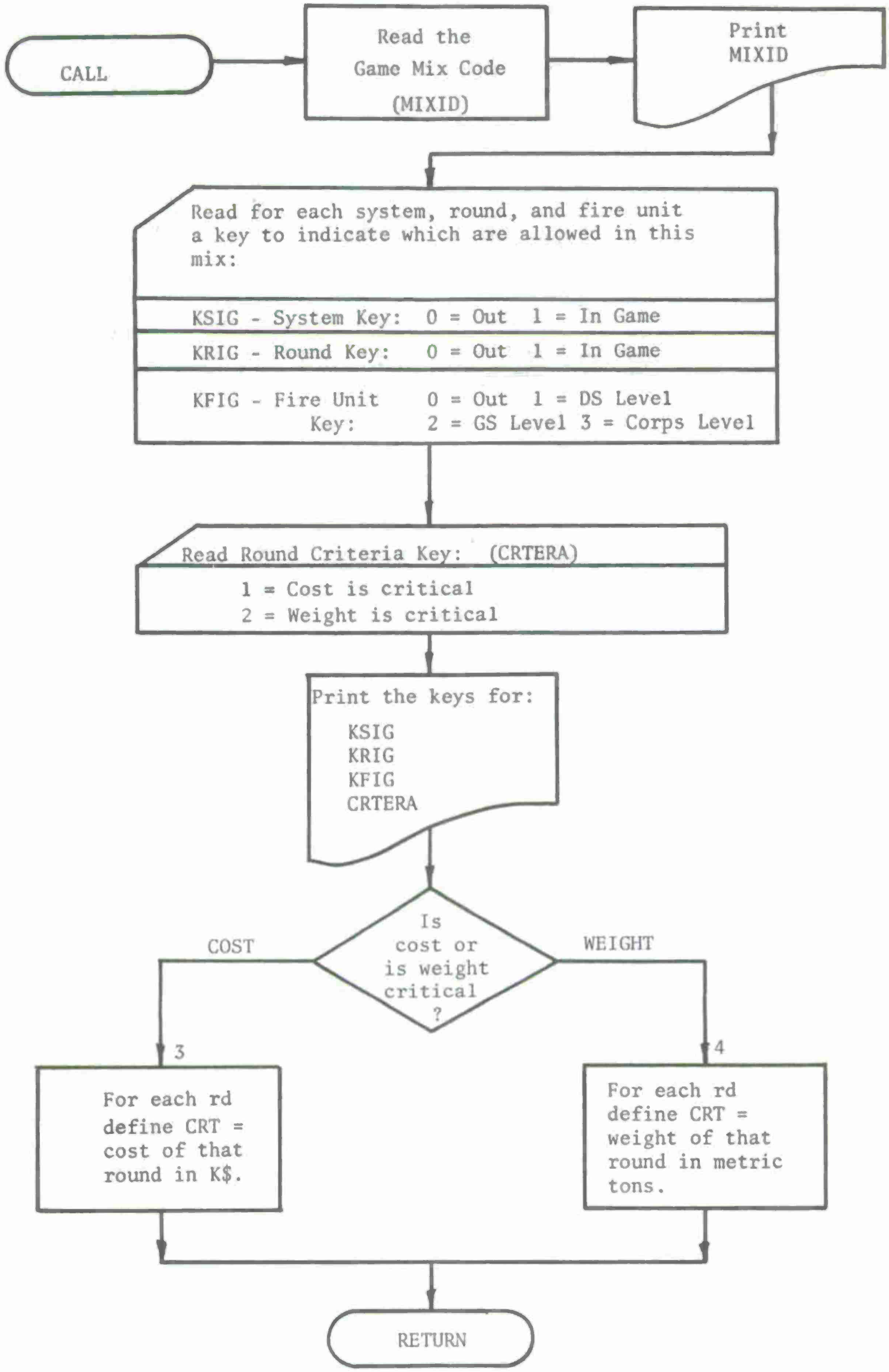
Subroutines which call RDMIX: Main Program

Subroutines called by RDMIX: None

This subroutine provides for the card input of a given system-round-fire unit mix for each cycle thru the game. The inputs include an alpha-numeric Mix Code and a key to define each system, round, and fire unit (from among all those input) which are allowed for a specific computer cycle through the target list. In addition, a criteria key is also read in to define whether least cost or least weight of ammunition is to be minimized in the selection of a "best round" to be fired against each target. See card sets 12 through 16, Table 3.5.

This subroutine completes the card input cycle, and return to the main program allows the computer game to begin.

Subroutine RDMIX



Program Element: Initialize Time, Counters and Arrays

Symbolic Name: TZERO

Arguments in Call Statement: None

Subroutines which call TZERO: Main Program

Subroutines Called by TZERO: None

This subroutine zeros various counters, lists, and arrays at the start of each computer cycle through the target list. It also sets the game clock and fire unit clocks to the input game-start-time.

Subroutine TZERO

CALL

Set following equal to zero to start the game:

KOUNT	No. of tgts in TN array at any time.
MATCH	Position of a given tgt on TN array.
MFEAT	No. of tgts on Defeated List at any time.
LOSS	No. of tgts on Lost Tgt list at any time (tgts attacked but not defeated).
ITC	No. of "OTHER" missions acquired till now.
TN Array	Priority listing of tgts (w/tgt data) for a given time period.
TLOST Array	Listing of tgt ID and data for tgts attacked but not defeated.
TDFT List	Listing of defeated tgt ID's.
FT Array	Lists ammo expended by ea unit for the current & last 5 quarter hours.
S Array	Sums ammo COST & WGT, No. of Pers, Tanks & APC's defeated by ea system.
RDCNT Array	Sums no. of rds fired at ea GUN-TGT range from 1 to 30 (or more) KM.
AMWS	Sums Military Worth Points scored in game.
NACQ	Sums no. of tgt acquisitions in game.
SPERS	Sums no. of Personnel acquired in game.
STANK	Sums no. of tanks acquired in game.
SAPC	Sums no. of APC's acquired in game.
NQ	Sums no. of tgts/missions not undertaken for any reason (queued).
SMW	Sums military worth of damaged targets.
NTGT	Sums individual tgts acquired in game (counts only 1st acquisition).
NOMF	Sums no. of "OTHER" missions fired, i.e. defeated.
NQLP	Sums tgts "queued" (not fired) due to low priority.
NRPD	Sums reacquisitions of previously defeated tgts.
NRW2	Sums number of "combined" tgts (w/i 200 meters of each other).
NQOM	Sums OTHER missions tried but can't do, i.e. queued.
NQD	Sums tgts queued when tgt departs before it can be fired.
NFM	Sums all fire missions/targets attacked each time attacked.
NFMD	Sums all OTHER missions & regular tgts (fire missions) which are defeated.
NOM	Sums OTHER type missions in game.

Set T = TZRO (Sets game clock = game start time.)

Set TUBFU List = T (Sets clock for each unit = game start time.)

RETURN

Program Element: Read Input from Target Tape

Symbolic Name: RTAPE

Arguments in Call Statement: None

Subroutines which call RTAPE: Main Program

Subroutines called by RTAPE: None

This subroutine reads from tape 33 target parameters for each target on the target tape. If the target location error is greater than three times its estimated radius, or if the frequency of this target is zero for the defined game intensity the target is disregarded.

If the target is a special ("other") type mission (Smoke, Illumination or H&I), an additional 8 parameters are read from the tape to define the precalculated number of rounds required by each weapon system to accomplish the mission. See Target Input Variables, Table 3.6.

Subroutine RTAPE

CALL

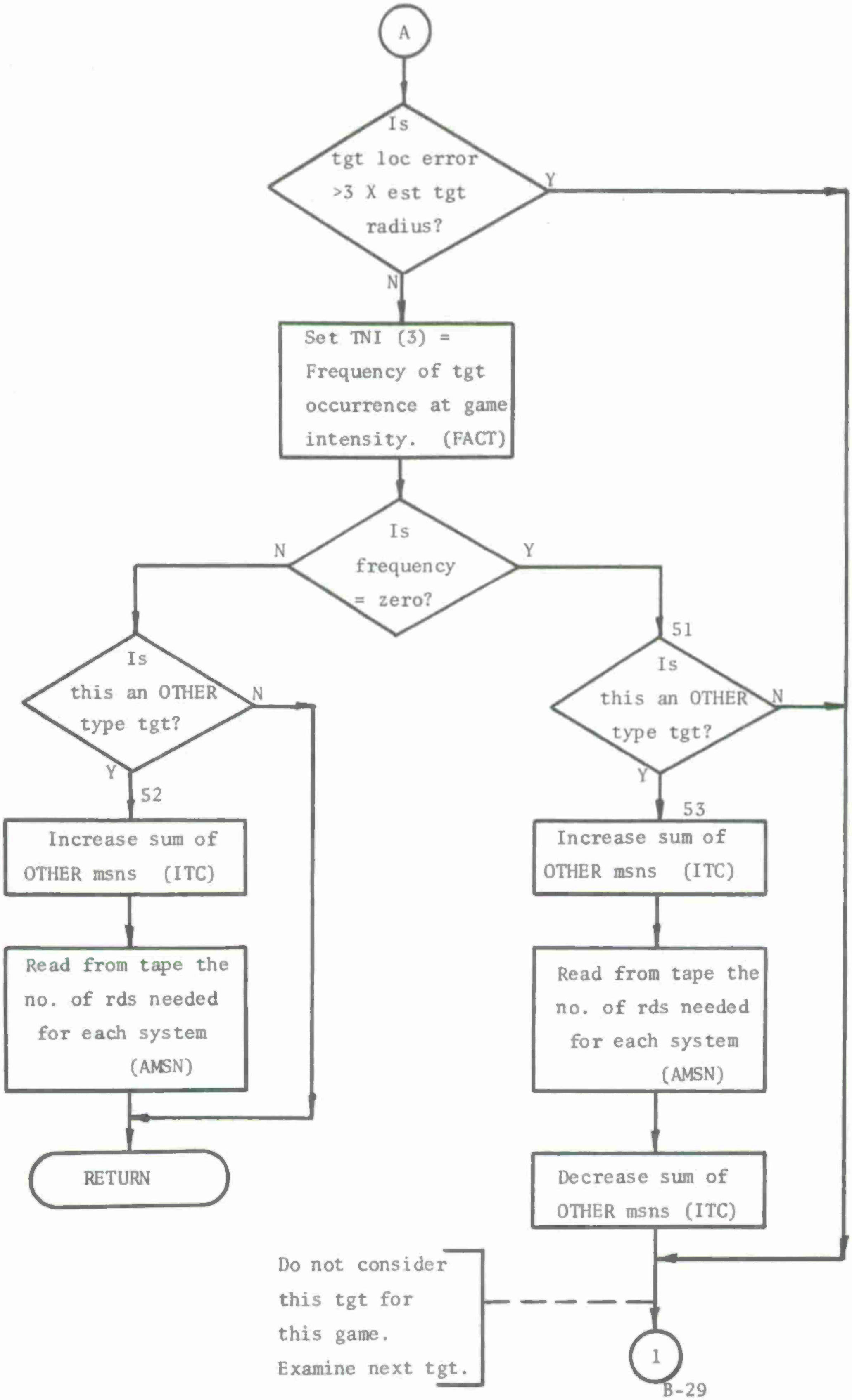
(1)

READ FROM TAPE, TGT DATA:

VARIABLE	MEANING	UNITS
TNI (1)	Tgt ID Number (e.g., 9016.0)	real
(2)	Acquisition No. (0.,1.,2., or 3.)	real
(3)	Frequency of Tgt at BASE Intensity	1.
(4)	Estimated Tgt Posture Mix	0 thru 11
(5)	Estimated Fraction of Tgt in Woods	real
(6)	Estimated Fraction of Tgt in Open	real
(7)	Estimated Tgt Radius	meters
(8)	Estimated Arrival Time of Tgt	decimal hours
(9)	Estimated Departure Time of Tgt	decimal hours
(10)	Tgt Location Error	meters
(11)	Estimated Easting of Tgt	km
(12)	Estimated Northing of Tgt	km
(13)	Tgt Distance from FEBA	km
(14)	Est. Military Worth of Tgt	0 thru 21
(15)	Tgt Acquisition Code	1,2, or 3
(16)	Target Type Code No.	1 thru 70
(17)	Threshold "A" Attack Level	.5
(18)	Defeat Level	.5
(19)	Actual Tgt Posture Mix	0 thru 11
(20)	Actual Fraction of Tgt in Woods	real
(21)	Actual Fraction of Tgt in Open	real
(22)	Actual Tgt Radius	meters
(23)	Actual Arrival Time of Tgt	decimal hours
(24)	Actual Departure Time of Tgt	decimal hours
(25)	No. Personnel in Tgt	personnel
(26)	No. Tanks in Tgt	tanks
(27)	No. APC's in Tgt	APC
(28)	Initial Frac. of Pers. Survivors	real
(29)	Initial Frac. of Tank Survivors	real
(30)	Initial Frac. of APC Survivors	real
(31)	Frequency of Tgt at LOW Intensity	1,2,3...
(32)	Frequency of Tgt at MID Intensity	1,2,3...
(33)	Frequency of Tgt at HIGH Intensity	1,2,3...

(A)

The following page is blank.


A
Is tgt loc error >3 X est tgt radius?
Y
N
Set TNI (3) = Frequency of tgt occurrence at game intensity. (FACT)
Is frequency = zero?
N
Y
51
Is this an OTHER type tgt?
N
Y
52
Increase sum of OTHER msns (ITC)
Read from tape the no. of rds needed for each system (AMSN)
RETURN
Is this an OTHER type tgt?
N
Y
53
Increase sum of OTHER msns (ITC)
Read from tape the no. of rds needed for each system (AMSN)
Decrease sum of OTHER msns (ITC)
Do not consider this tgt for this game. Examine next tgt.
1
B-29

Program Element: Remove Target from Target List

Symbolic Name: REMOVE

Arguments in Call Statement: (K) - Identifies the position on target list of target to be removed.

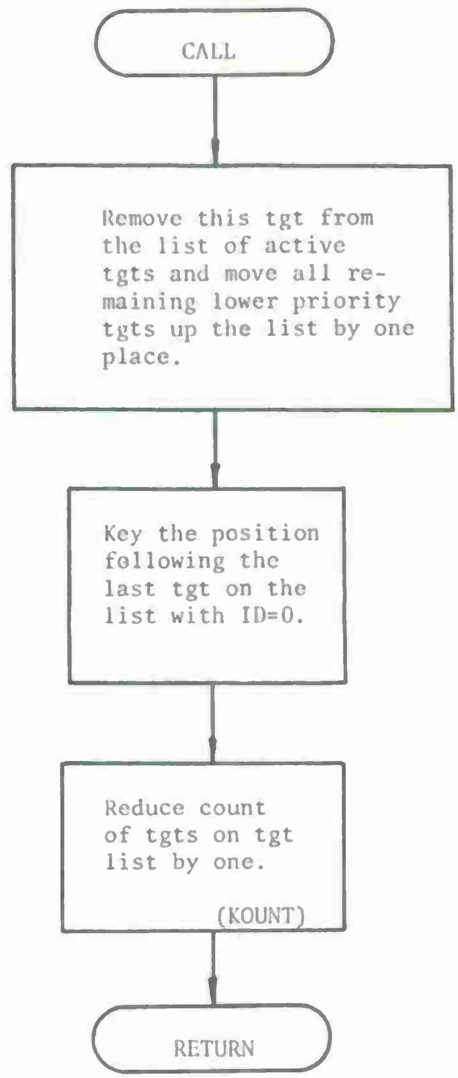
Subroutines which call REMOVE: Main Program, COMPAR

Subroutines called by REMOVE: None

This subroutine removes a target from the priority ordered list of targets and moves all other targets beneath the removed target up the list by one position.

For program control, the subroutine also sets the target ID equal to zero at the position following the last remaining target on the list and reduces the target list counter by one.

Subroutine REMOVE



Program Element: Compare Targets by Priority

Symbolic Name: COMPAR

Arguments in Call Statement: None

Subroutines which call COMPAR: Main Program

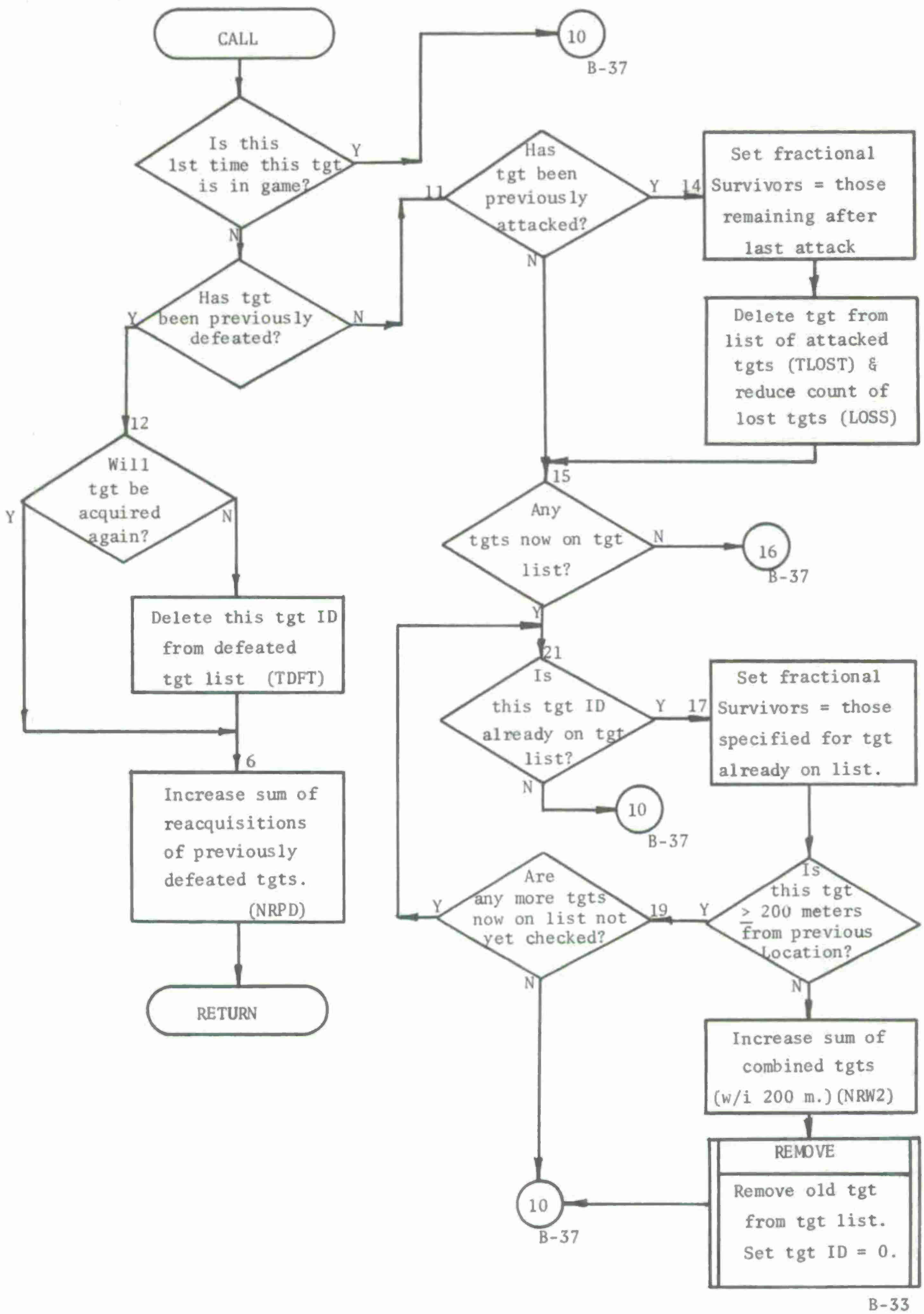
Subroutines called by COMPAR: REMOVE

This subroutine places a given target in its proper position on the target list according to its priority (Military Worth) and its estimated departure time.

If the target had been previously acquired, a check is made to see if it was attacked and/or defeated. If it had been defeated (determined by checking against the defeated target list) it is not added to the target list, but is only counted as a reacquisition of a previously defeated target. If it had been attacked-but not defeated-(determined by checking against the list of attacked targets) the damage previously inflicted is charged against the new acquisition and the old acquisition is removed from the attacked target list.

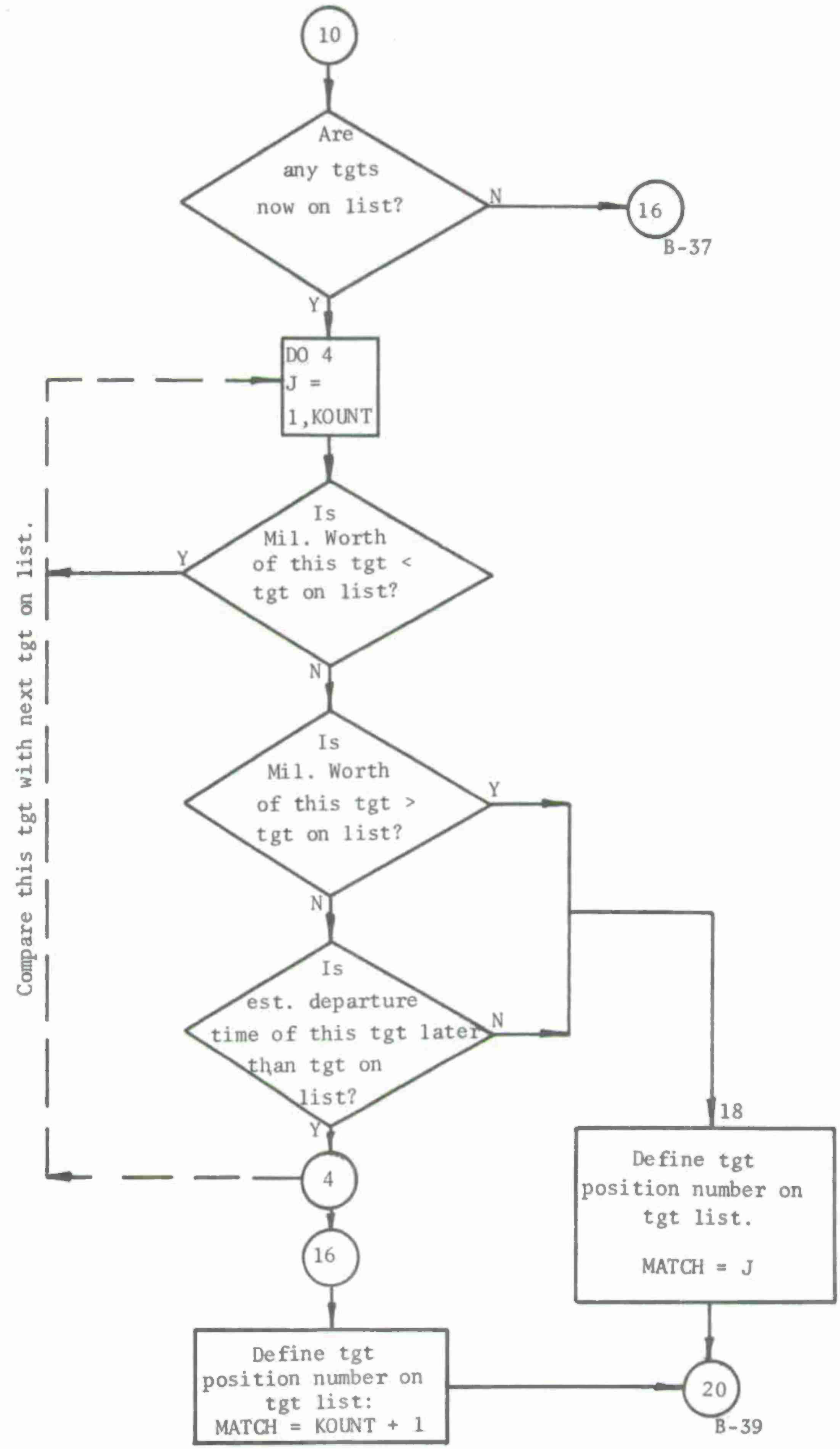
A check is then made to determine if previous acquisitions of this target are still on the target list (i.e. not yet attacked). If so, the damages for the last previous acquisition are assigned to the current acquisitions and each acquisition remains on the target list. However, if the current target is located within 200 meters of any of the previous acquisitions still on the list, that "old" acquisition is removed from the target list and the sum of "combined targets" is increased by one.

Subroutine COMPAR

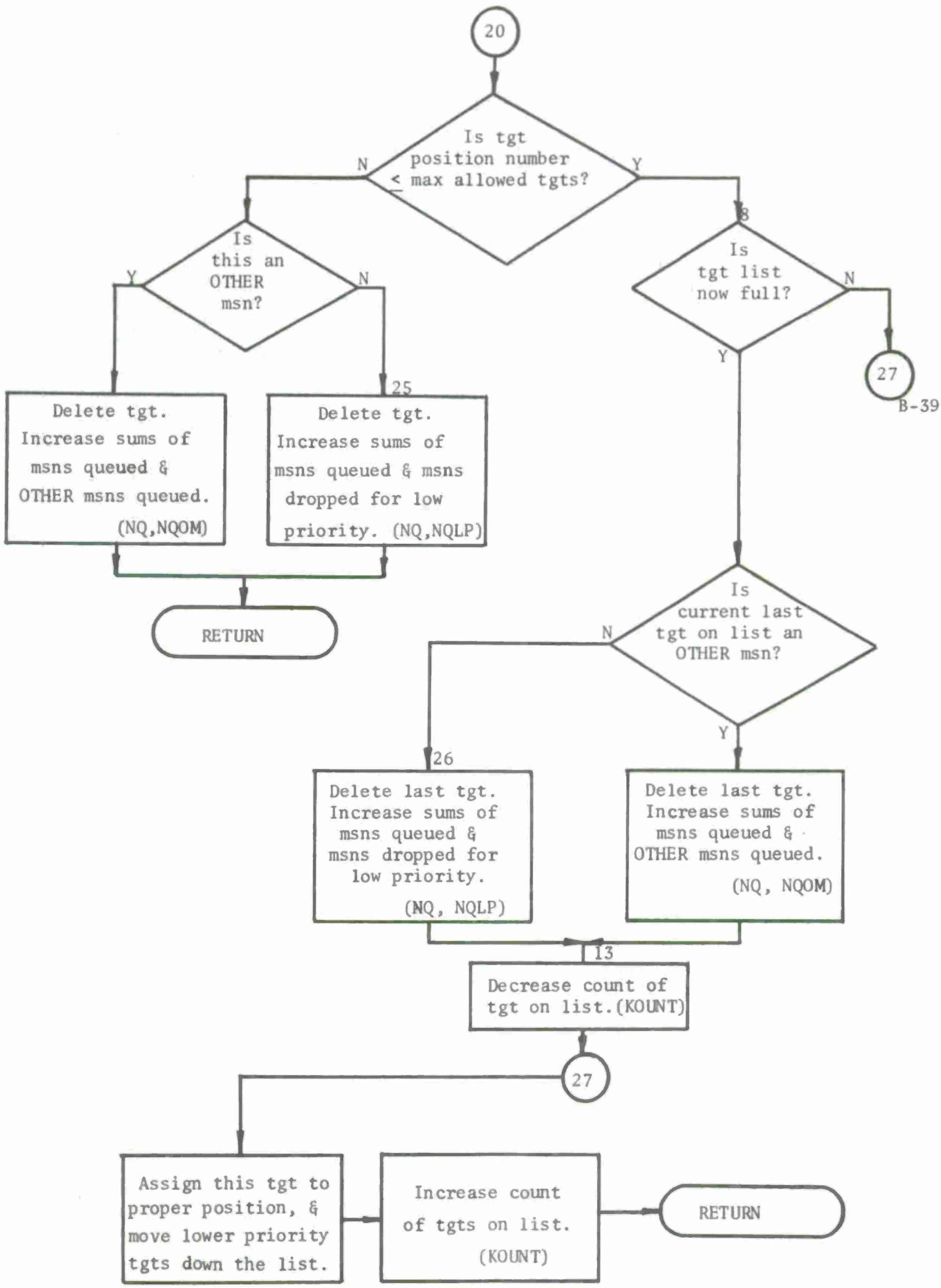


The position of the current target on the target list is then determined. The higher a target's military worth, the higher will be its position on the list. For targets of equal military worth, the one departing soonest will be given higher priority. Should both military worth and departure time be equal, the earlier arriving target receives higher priority.

Finally, the target (and its data) are placed on the target list in its assigned position. If the addition of the target causes the list capacity to be exceeded, the lowest priority target is dropped and the appropriate counters are increased.



The following page is blank.


20
Is tgt position number ≤ max allowed tgts?
N
Y
8
Is this an OTHER msn?
Y
N
Is tgt list now full?
N
Y
27
B-39
25
Delete tgt. Increase sums of msns queued & OTHER msns queued. (NQ,NQOM)
Delete tgt. Increase sums of msns queued & msns dropped for low priority. (NQ,NQLP)
RETURN
Is current last tgt on list an OTHER msn?
N
Y
26
Delete last tgt. Increase sums of msns queued & msns dropped for low priority. (NQ, NQLP)
Delete last tgt. Increase sums of msns queued & OTHER msns queued. (NQ, NQOM)
13
Decrease count of tgt on list.(KOUNT)
27
Assign this tgt to proper position, & move lower priority tgts down the list.
Increase count of tgts on list. (KOUNT)
RETURN

Program Element: Direct Support Echelon

Symbolic Name: DIRSUP

Arguments in Call Statement: (IT) - Identifies position on the target list of the target which is being considered.

Subroutines which call DIRSUP: Main Program

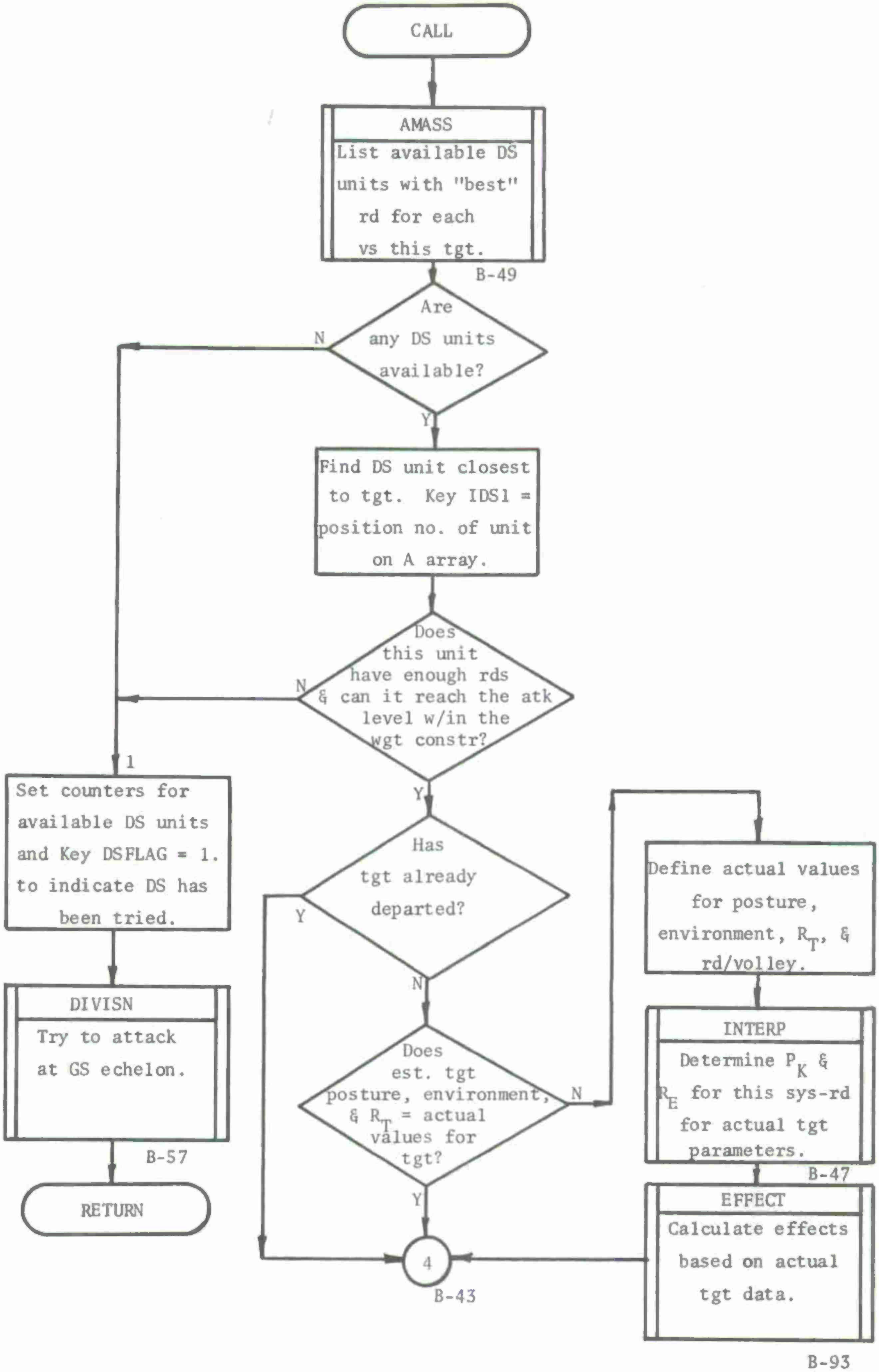
Subroutines Called by DIRSUP: AMASS, INTERP, EFFECT, DIVISN

This subroutine provides the initial attempt to fire upon targets acquired at the direct support (DS) echelon. After keys are set to indicate DS level, subroutine AMASS is called to provide a list of all available DS units, including the most effective round, number of rounds available and number of rounds required by each available DS unit to reach the specified attack level. If there are no available DS units, subroutine DIVISN is called, to attempt attack with GS echelon units.

If DS units are available, the unit closest to the target is identified and checked to determine if its required number of rounds is available and if those rounds are within the overall ammunition weight constraint (30 ton limit for Category I targets, 15 ton limit for Category II and III targets). If this closest unit does not have sufficient rounds available or if it cannot reach the specified attack level within the weight constraint, the DS echelon is deemed unable to attack the target alone, so DIVISN subroutine is called to attempt GS level attack.

Should the closest DS unit have sufficient rounds and be within the target weight constraint, the target is considered to be attacked.

Subroutine DIRSUP

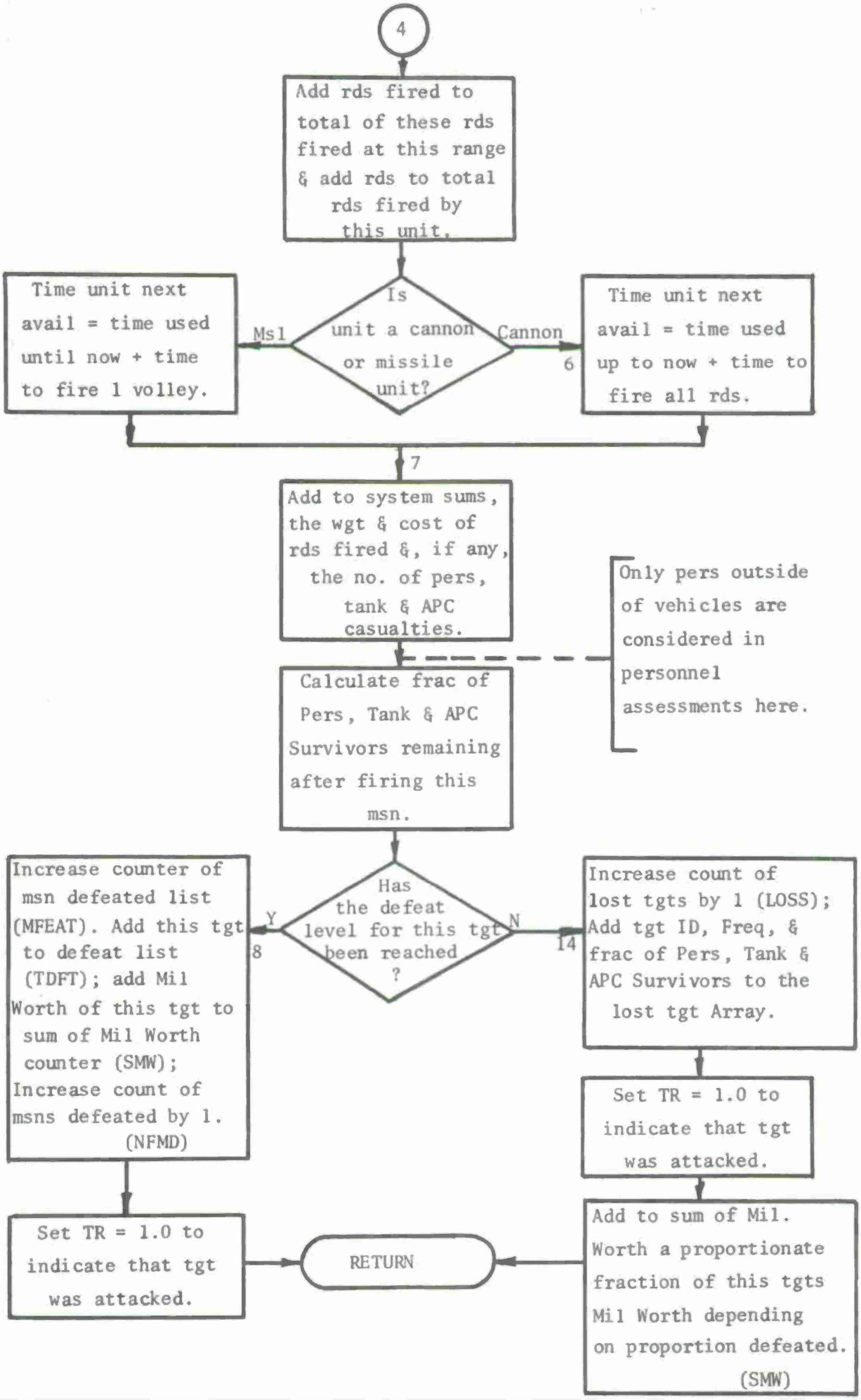


A check of actual target data (posture, environment and radius) is then made. If actual data do not agree with estimated target data, subroutines INTERP and EFFECT are called to determine the effects of the previously determined required number of rounds against the actual target data. (If actual target data equal estimated data then the effects which were calculated when AMASS was called at the beginning of this DIRSUP subroutine are still valid). This subroutine then credits the rounds fired, damage accomplished and time used by the fire unit to the appropriate counters as explained below.*

The number of rounds fired are credited by round ID number and gun-target range (to nearest kilometer) in a round-counter matrix. See subroutine OUTPUT for the format of this matrix. The "Time used by fire-unit" clock is then increased to account for the time used by the fire unit to fire the required number of rounds against the target. Weapon system counters are then increased for the specific weapon system which defines the fire unit. The weight and cost of ammunition fired and the number of personnel, tanks and APC's defeated by the fire unit are credited to the appropriate weapon system. If actual target departure time indicates that the target has departed its location prior to the time of engagement, then no damage is credited to this mission, although the rounds fired and time used are still charged to the fire unit.

---

*NOTE: These processes are repeated in other subroutines where fire units are credited with the accomplishment of a mission. The process is detailed only for this subroutine and reference is made to this explanation where appropriate in other subroutines: DIVISN, SHMUVL, SHONVL & CORP.



The following page is blank.

After calculating the remaining personnel, tank and APC survivors in the target after firing this fire mission, a check is made to determine if the total damage inflicted on the target at this time meets the defeat level (50% damage). If so, the target is added to the defeated target list (TDFT) and the military worth of the target is added to the counter of military worth points scored. If the target was not defeated (<50% damage inflicted) the target is added to the "attacked-but-not-defeated" (TLOST) list. (This list contains the target ID number and the fractional personnel, tank and APC survivors remaining in the target. Should this target be reacquired later in the game, these survivor values will then be assigned to the target, so as to account for the previous damage inflicted). A proportionate amount of the targets' Military Worth is then added to the counter of Military Worth points scored.

The final step when a target is attacked, whether defeated or not, is to set a key to indicate that it has been attacked and control is returned to the main program.

Program Element: Interpolation

Symbolic Name: INTERP

Arguments in Call Statement:

(IR) - Identifies which of the input rounds is being considered.

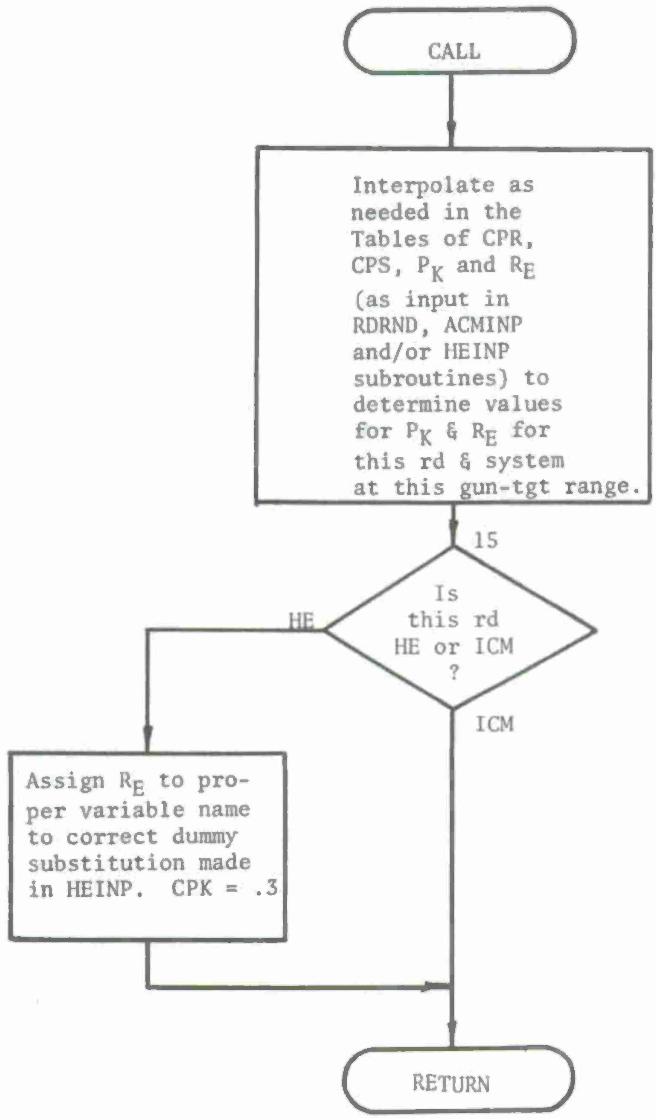
(IA) - Identifies which fire unit on "A" array is being considered.

Subroutines which call INTERP: DIRSUP, AMASS, DIVISN, SHMUVL, SHONVL, CORP

Subroutines called by INTERP: None

This subroutine provides interpolation of the input data for the various rounds (as input in RDRND, ACMINP and HEINP subroutines). It determines the probability of kill (P_K) and Radius of Effects (R_E) for a given round at a specific gun-target range against the five postures (personnel standing, prone and crouching and tanks and APC's) in both open and wooded environments.

NOTE: If the round under consideration is of the High Explosive (HE) type, this subroutine also assigns the P_K's and R_E's to the proper variable names. This corrects the "dummy" substitutions made in subroutine HEINP.


CALL
Interpolate as needed in the Tables of CPR, CPS, P_K and R_E (as input in RDRND, ACMINP and/or HEINP subroutines) to determine values for P_K & R_E for this rd & system at this gun-tgt range.
15
Is this rd HE or ICM ?
HE
ICM
Assign R_E to proper variable name to correct dummy substitution made in HEINP. CPK = .3
RETURN

Program Element: Mass Fire Units

Symbolic Name: AMASS

Arguments in Call Statement: (IT) - Identifies position on target list of the target which is being considered.

Subroutines which call AMASS: DIRSUP, DIVISN, CORP

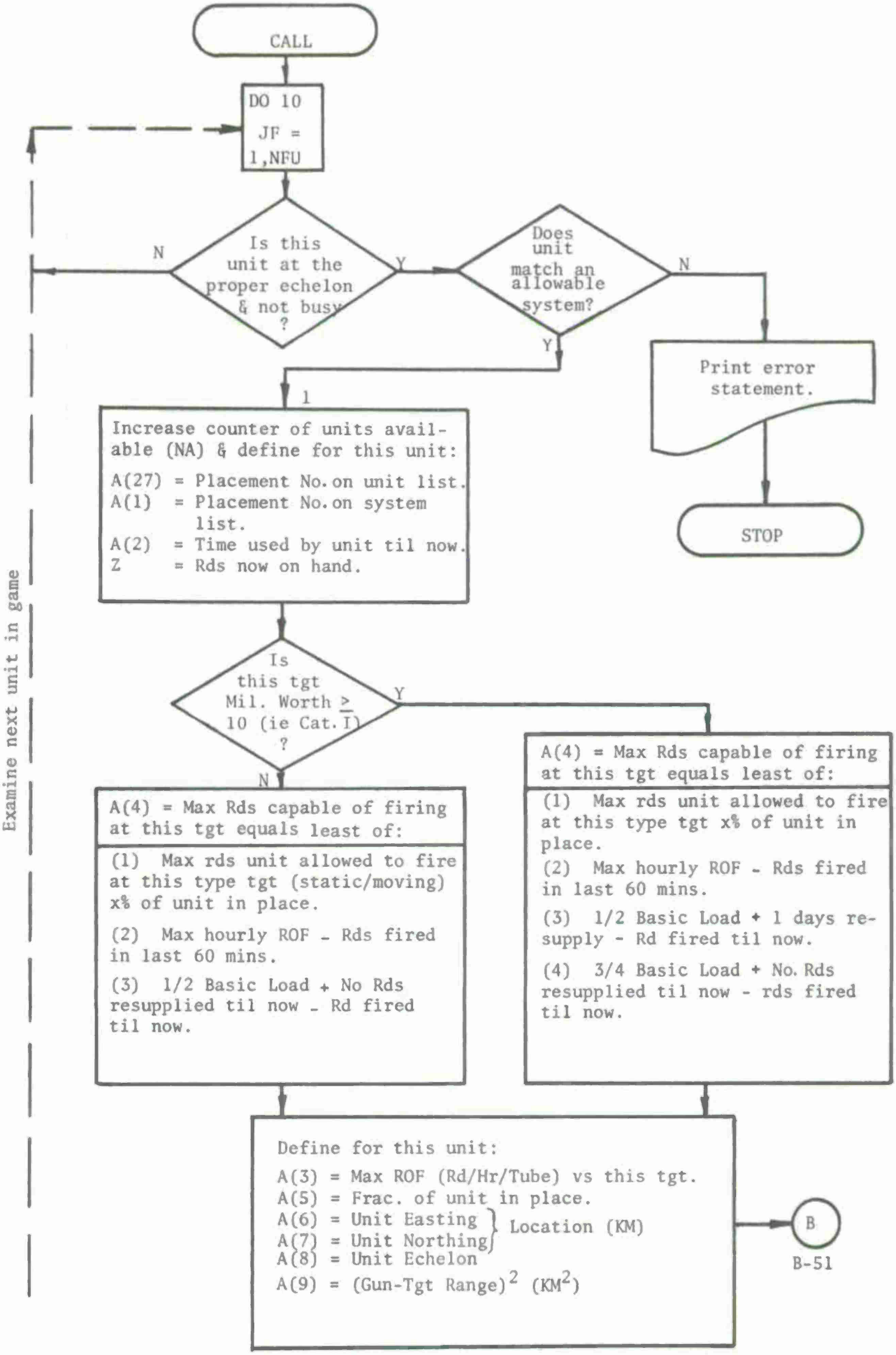
Subroutines called by AMASS: INTERP, EFFECT

This subroutine examines all fire units at a given echelon and builds an array ("A" array) of units capable of contributing fire against the target under consideration.

The subroutine searches through the list of fire units in the mix, immediately rejecting those units which are not at the echelon under consideration and those which are busy (whose clocks indicate that they are already committed beyond the current 15-minute game interval). (An error print is made if a unit is in the game whose weapon system type is not allowed in the mix for the game being run).

If a fire unit passes these initial checks, the counter of available fire units (NA) is increased by one and the subroutine begins to fill in the 27-element list of the "A" array for the fire unit. (See a compilation of this array on page B-54).

The initial elements of the array define fire unit parameters such as weapon system type, time already used, maximum allowable rate of fire, unit location, unit echelon and fire unit-to-target range. The maximum number of rounds the unit is allowed to fire is also calculated based upon the target's military worth and various unit

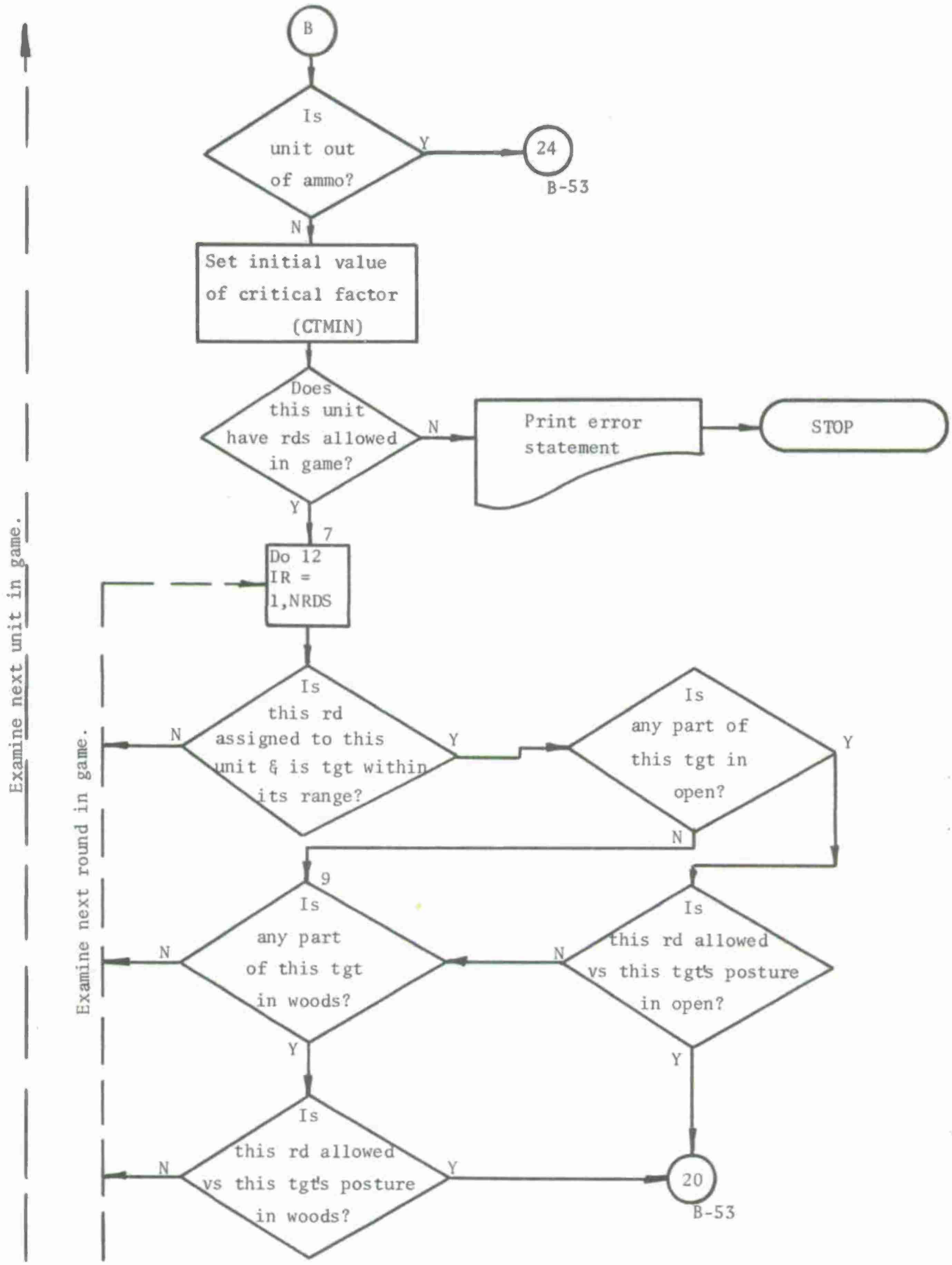

CALL
DO 10 JF = 1,NFU
Is this unit at the proper echelon & not busy ?
N
Y
Does unit match an allowable system?
N
Y
Print error statement.
STOP
1
Increase counter of units available (NA) & define for this unit:
A(27) = Placement No. on unit list.
A(1) = Placement No. on system list.
A(2) = Time used by unit til now.
Z = Rds now on hand.
Examine next unit in game
Is this tgt Mil. Worth ≥ 10 (ie Cat. I) ?
Y
N
A(4) = Max Rds capable of firing at this tgt equals least of:
(1) Max rds unit allowed to fire at this type tgt (static/moving) x% of unit in place.
(2) Max hourly ROF - Rds fired in last 60 mins.
(3) 1/2 Basic Load + No Rds resupplied til now - Rd fired til now.
A(4) = Max Rds capable of firing at this tgt equals least of:
(1) Max rds unit allowed to fire at this type tgt x% of unit in place.
(2) Max hourly ROF - Rds fired in last 60 mins.
(3) 1/2 Basic Load + 1 days resupply - Rd fired til now.
(4) 3/4 Basic Load + No. Rds resupplied til now - rds fired til now.
Define for this unit:
A(3) = Max ROF (Rd/Hr/Tube) vs this tgt.
A(5) = Frac. of unit in place.
A(6) = Unit Easting
A(7) = Unit Northing
Location (KM)
A(8) = Unit Echelon
A(9) = (Gun-Tgt Range)^2 (KM^2)
B
B-51

parameters. If the unit has no ammunition on hand, it is deleted from consideration and the next unit in the game is considered.

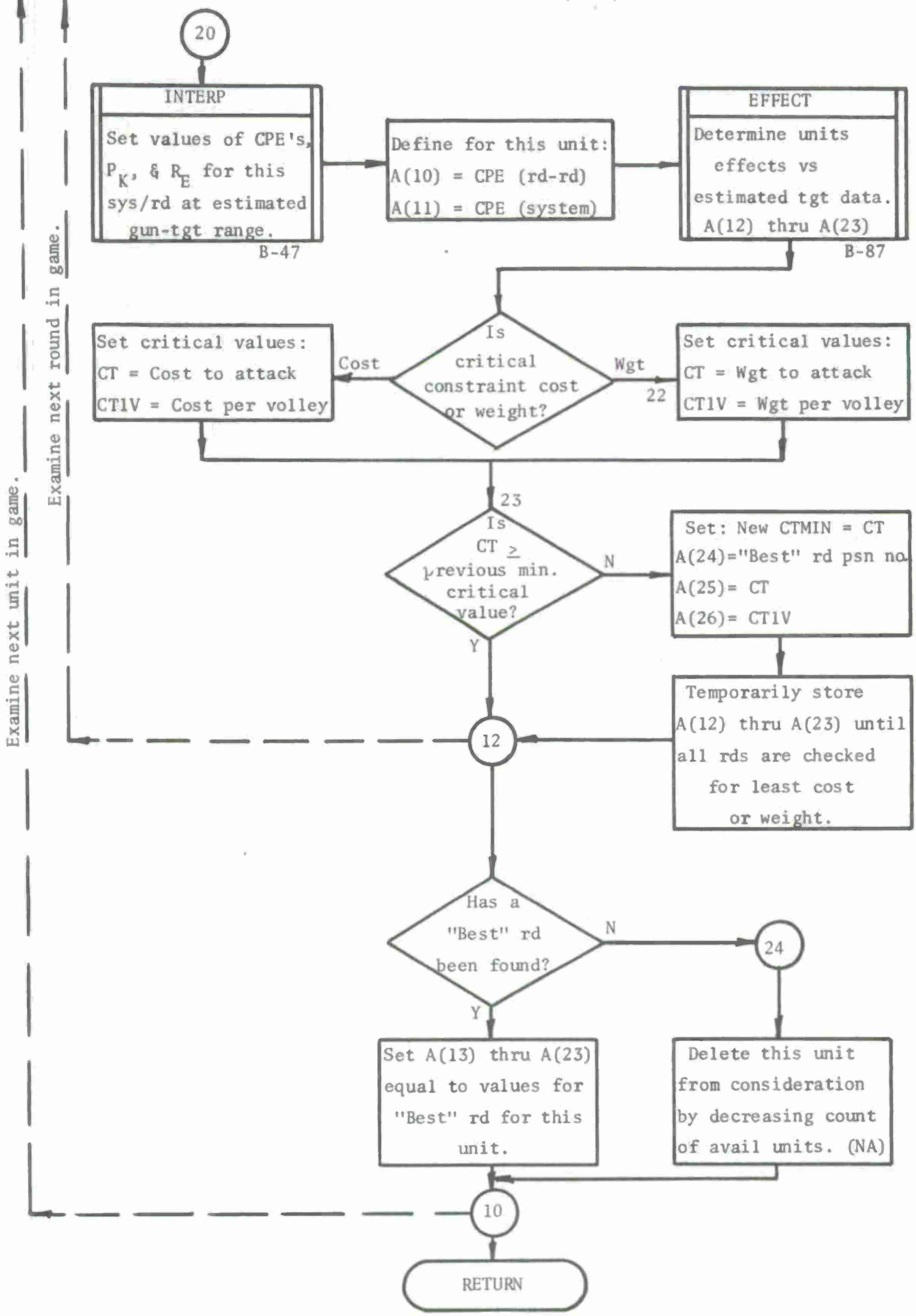
After checking to insure that at least one round has been defined (in RDFU) for use by this unit, all the rounds so defined are examined in order to choose a "best round" for use against this target. (If no rounds are so defined, an error print results).

As each round is considered, initial checks are made to insure that a round's maximum range is within gun-target range and that the round is allowed against the particular target posture mix. Subroutines INTERP and EFFECT are then called to determine the number of those rounds needed by the unit, when firing alone, to reach the specified attack level. As these calculations are completed for each round, total required ammunition cost or weight (depending upon the specified allocation constraint defined in RDMIX) is compared with other rounds. After all rounds have been considered, the data for the "best round" (i.e. least cost or least weight) is assigned to the A array, and the next fire unit is examined.

If the total cost or weight of the "best round" exceeds a specified key value (CTI = 1,000,000), the fire unit is deleted from consideration. After all units at the specified echelon have been examined program control returns to the calling subroutine.



The following page is blank.


20
INTERP
Set values of CPE's, P_K, & R_E for this sys/rd at estimated gun-tgt range.
B-47
Define for this unit:
A(10) = CPE (rd-rd)
A(11) = CPE (system)
EFFECT
Determine units effects vs estimated tgt data. A(12) thru A(23)
B-87
Is critical constraint cost or weight?
Cost
Wgt
22
Set critical values:
CT = Cost to attack
CT1V = Cost per volley
Set critical values:
CT = Wgt to attack
CT1V = Wgt per volley
23
Is CT ≥ previous min. critical value?
N
Y
Set: New CTMIN = CT
A(24)="Best" rd psn no.
A(25)= CT
A(26)= CT1V
Temporarily store A(12) thru A(23) until all rds are checked for least cost or weight.
12
Examine next round in game.
Examine next unit in game.
Has a "Best" rd been found?
N
Y
24
Set A(13) thru A(23) equal to values for "Best" rd for this unit.
Delete this unit from consideration by decreasing count of avail units. (NA)
10
RETURN

ELEMENTS OF THE "A" ARRAY

(Defined for each capable FU)

Array Elements	Definition	Units	Where Calculated	Variable Name
A(1,NA)	Which of NSYS systems describes this FU.	Int.	AMASS	IS
2	Time already used by this FU.	Dec. Hrs.	AMASS	TUBFU(JF)
3	Max. rate-of-fire vs this target.	Rd/Hr/Tube	AMASS	DROF(IS) or SROF(IS)
4	No. rounds FU now capable of firing at tgt.	Rds	AMASS	R
5	Fraction of FU now emplaced.	Decimal	AMASS	FRWM(IS) or 1
6	Current FU Easting	KM	AMASS	XS(INS,JF)
7	Current FU Northing	KM	AMASS	YS(INS,JF)
8	FU Echelon	Int.	AMASS	MORG
9	(Unit-to-target Range)^2	KM^2	AMASS	-
10	CPE (Random) at this unit-target range	Decimal	INTERP	CPER
11	CPE (Total) at this unit-target range	Decimal	INTERP	CPET
12	% of target survivors from 1 volley fire.	Decimal	EFFECT	OVN
13	Rds. needed by FU alone to meet attack level	Rds	EFFECT	-
14	% of standing personnel survivors from 1 volley	Decimal	EFFECT	-
15	% of prone personnel survivors from 1 volley	Decimal	EFFECT	-
16	% of crouching personnel survivors from 1 volley	Decimal	EFFECT	-
17	% of tank survivors from 1 volley	Decimal	EFFECT	-
18	% of APC survivors from 1 volley	Decimal	EFFECT	-
19	% of standing personnel survivors from all req. rds.	Decimal	EFFECT	-
20	% of prone personnel survivors from all req. rds.	Decimal	EFFECT	-
21	% of crouching personnel survivors from all req. rds.	Decimal	EFFECT	-

ELEMENTS OF THE "A" ARRAY (CONT'D)

(Defined for each capable FU)

Array Elements	Definition	Units	Where Calculated	Variable Name
22	% of tank survivors from all req. rds.	Decimal	EFFECT	-
23	% of APC survivors from all req. rds.	Decimal	EFFECT	-
24	Which of NRDS is "best round"	Int.	AMASS	IR
25	Total Weight or Cost of all required rounds	K$ or Metric Tons	AMASS	CT
26	Weight or Cost of 1 volley	K$ or Metric Tons	AMASS	CT1V
27	Which of the NFU units is being considered.	Int.	AMASS	JF

Program Element: General Support Echelon

Symbolic Name: DIVISN

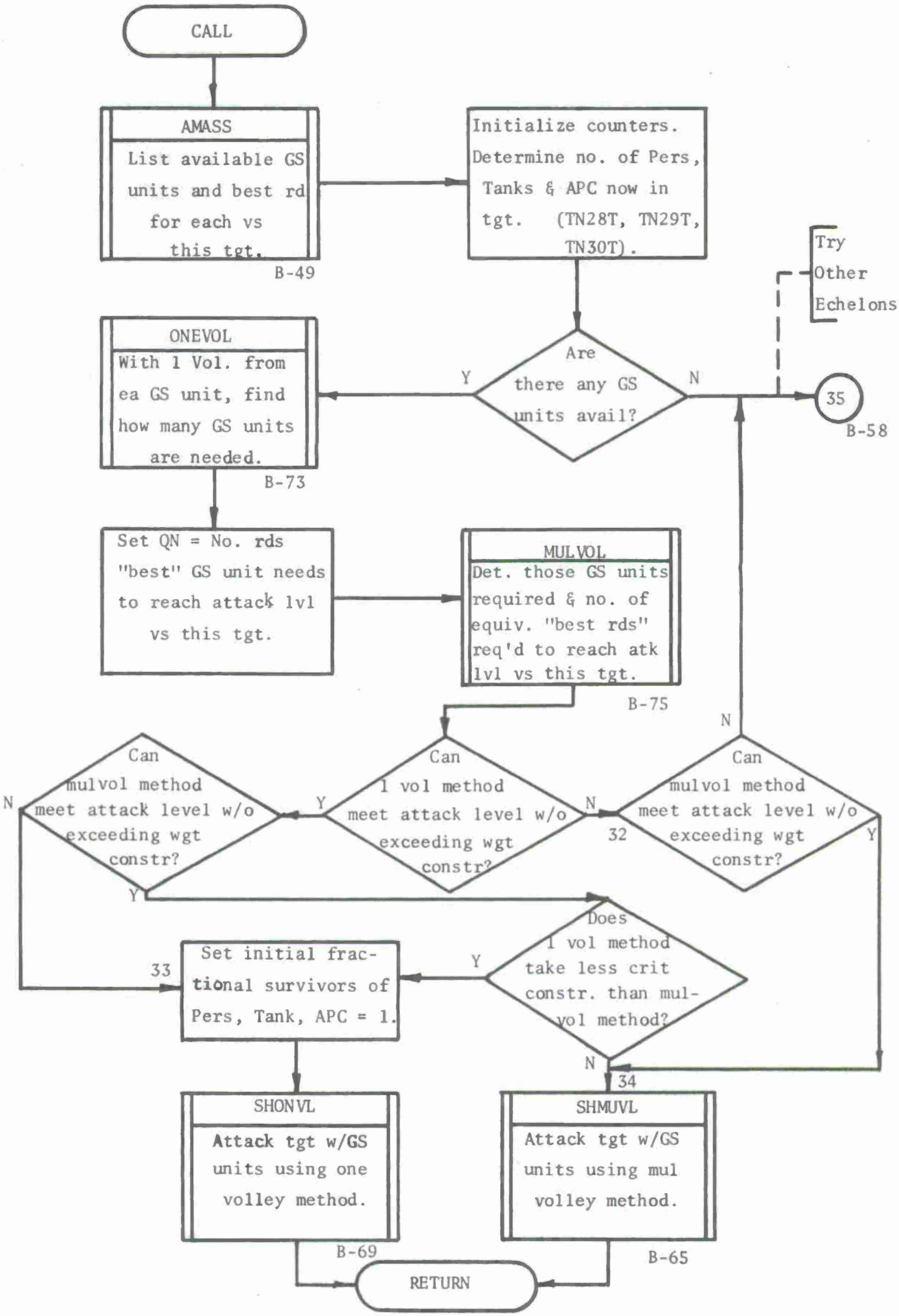
Arguments in Call Statement: (IT) - Identifies position on target list of the target which is being considered.

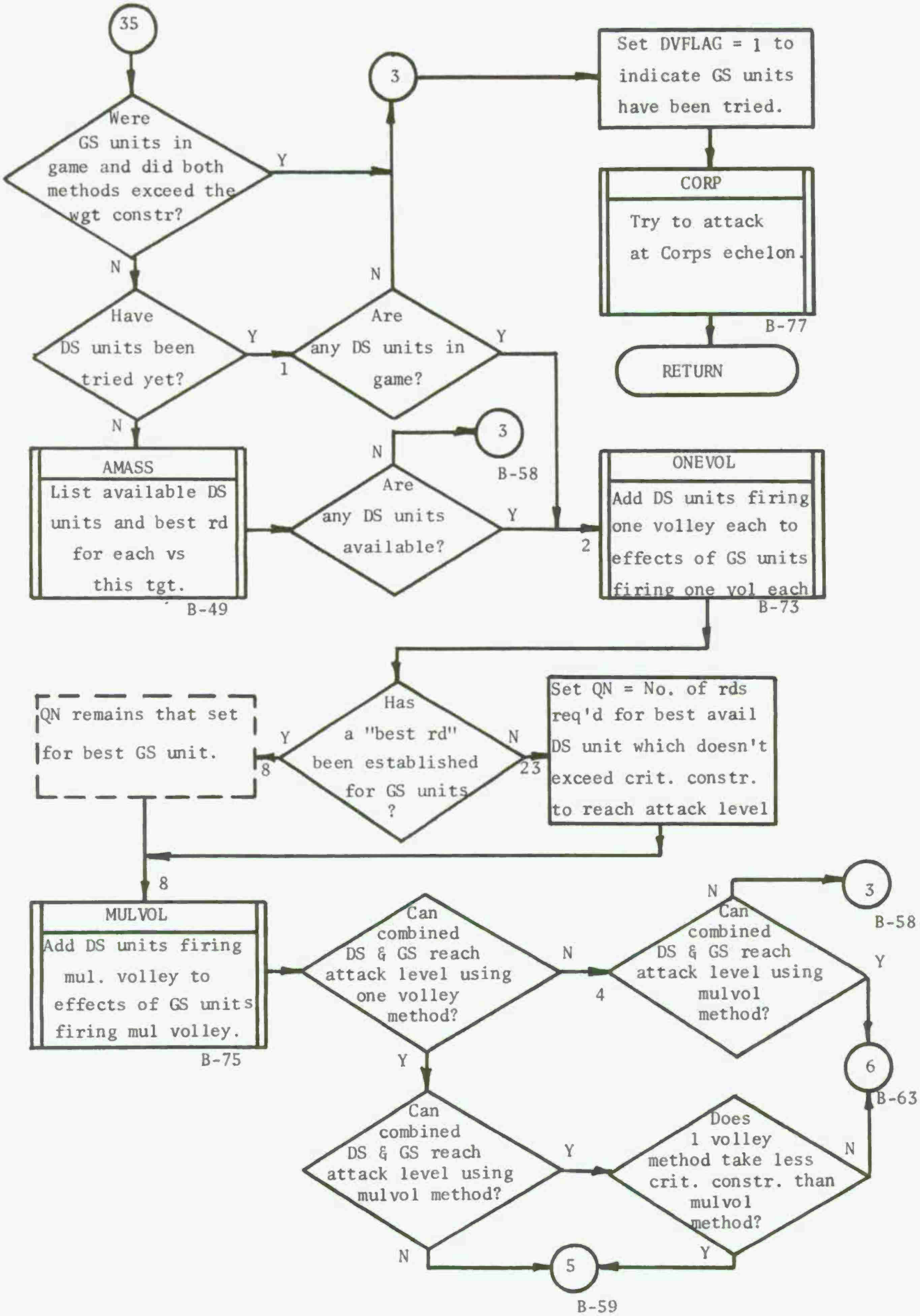
Subroutines which call DIVISN: Main Program, DIRSUP

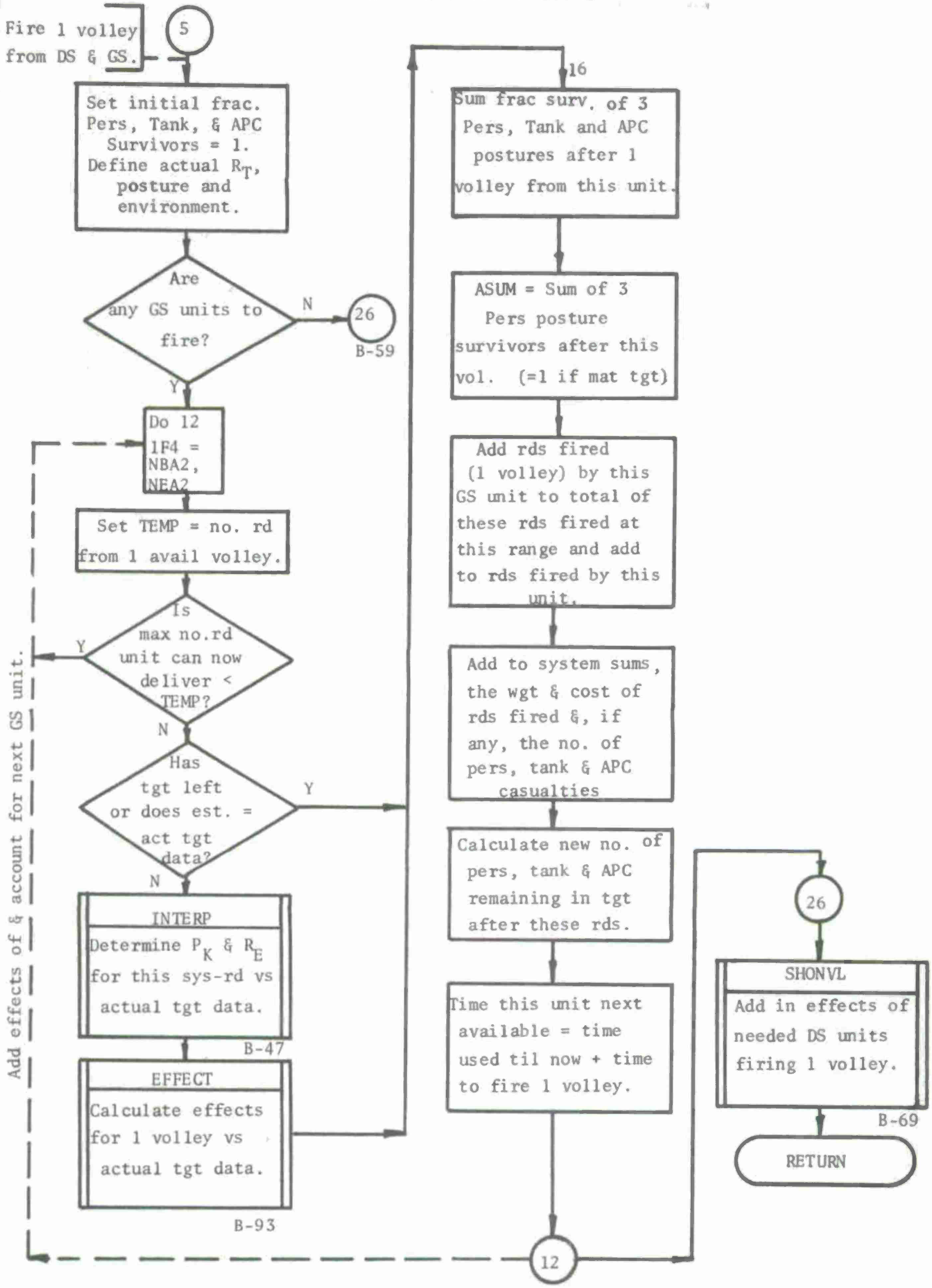
Subroutines called by DIVISN: AMASS, ONEVOL, MULVOL, SHMUVL, SHONVL, INTERP, EFFECT, CORP

This subroutine, when called from the main program, provides the initial attempt to fire upon General Support-acquired targets; and, when called from DIRSUP, it attempts to fire upon DS acquired targets that cannot be attacked by the closest DS fire unit to the target. After keys are set to indicate the GS level, subroutine AMASS is called to provide a list of available DIVISN(GS) units. Assuming GS units are available, subroutines ONEVOL and MULVOL are then called to determine how many GS units must be massed to reach the attack level when firing only one volley per unit (ONEVOL) and then when firing all rounds allowed from each unit (MULVOL).

After checking to insure that the overall ammunition weight constraint (15 or 30 tons, depending on target category) is not exceeded, the subroutine then calls SHONVL if the one-volley method requires less cost or weight of ammo; or SHMUVL is called if the multi-volley method requires less cost or weight. These two subroutines consider the target to be attacked and appropriate counters are increased to account for rounds fired and time used (as explained in DIRSUP). Control is then returned to the main program.

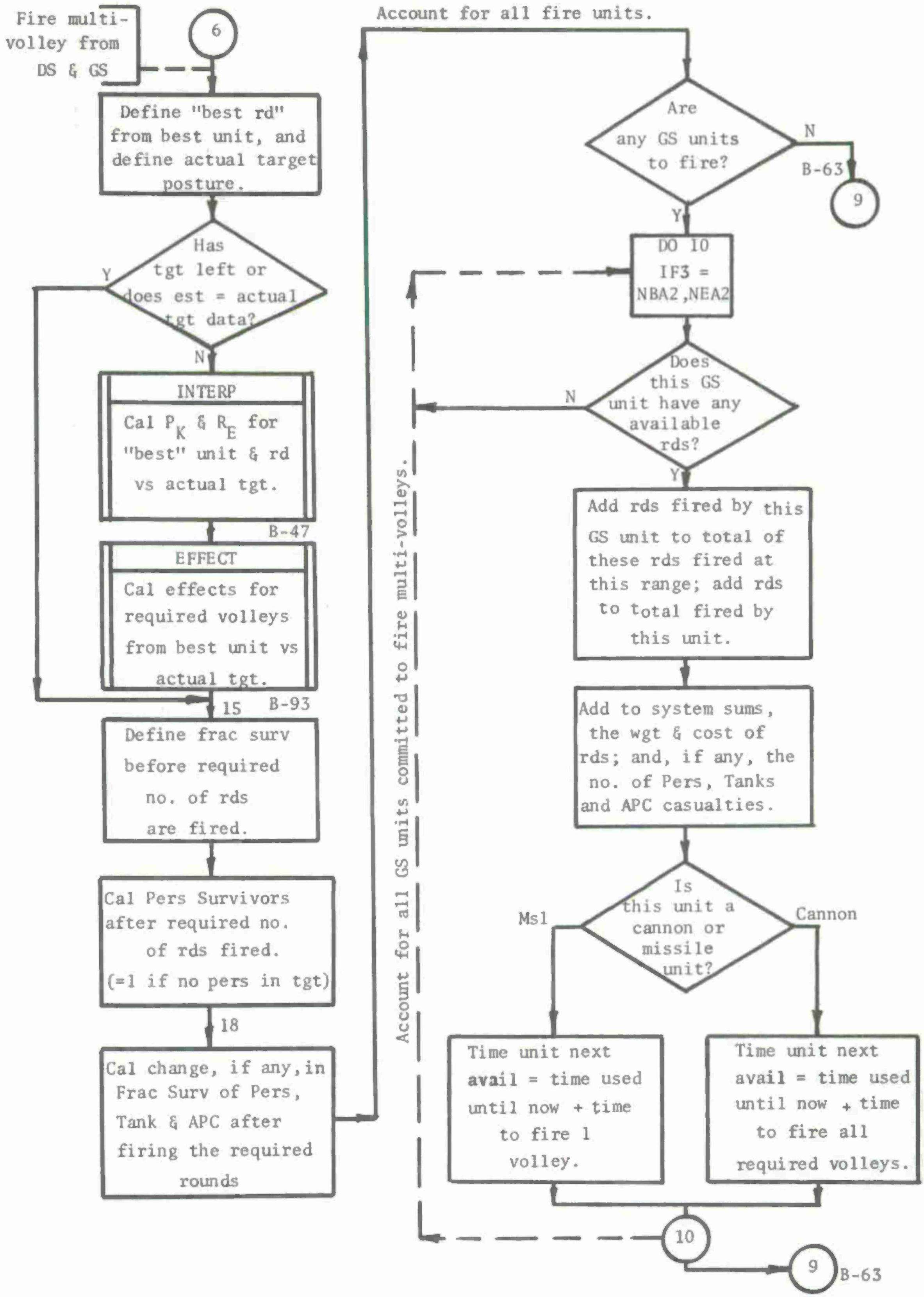

CALL
AMASS
List available GS units and best rd for each vs this tgt.
B-49
Initialize counters. Determine no. of Pers, Tanks & APC now in tgt. (TN28T, TN29T, TN30T).
Try Other Echelons
Are there any GS units avail?
Y
N
35
B-58
ONEVOL
With 1 Vol. from ea GS unit, find how many GS units are needed.
B-73
Set QN = No. rds "best" GS unit needs to reach attack lvl vs this tgt.
MULVOL
Det. those GS units required & no. of equiv. "best rds" req'd to reach atk lvl vs this tgt.
B-75
Can mulvol method meet attack level w/o exceeding wgt constr?
Can 1 vol method meet attack level w/o exceeding wgt constr?
Can mulvol method meet attack level w/o exceeding wgt constr?
32
Does 1 vol method take less crit constr. than mul-vol method?
33
Set initial frac-tional survivors of Pers, Tank, APC = 1.
34
SHONVL
Attack tgt w/GS units using one volley method.
B-69
SHMUVL
Attack tgt w/GS units using mul volley method.
B-65
RETURN


35
Were GS units in game and did both methods exceed the wgt constr?
Y
N
3
Set DVFLAG = 1 to indicate GS units have been tried.
CORP
Try to attack at Corps echelon.
B-77
RETURN
Have DS units been tried yet?
Y
1
N
Are any DS units in game?
Y
N
AMASS
List available DS units and best rd for each vs this tgt.
B-49
Are any DS units available?
N
3
B-58
Y
2
ONEVOL
Add DS units firing one volley each to effects of GS units firing one vol each
B-73
Has a "best rd" been established for GS units ?
Y
8
N
23
QN remains that set for best GS unit.
Set QN = No. of rds req'd for best avail DS unit which doesn't exceed crit. constr. to reach attack level
8
MULVOL
Add DS units firing mul. volley to effects of GS units firing mul volley.
B-75
Can combined DS & GS reach attack level using one volley method?
N
4
Y
Can combined DS & GS reach attack level using mulvol method?
N
3
B-58
Y
6
B-63
Can combined DS & GS reach attack level using mulvol method?
Y
N
Does 1 volley method take less crit. constr. than mulvol method?
N
Y
5
B-59

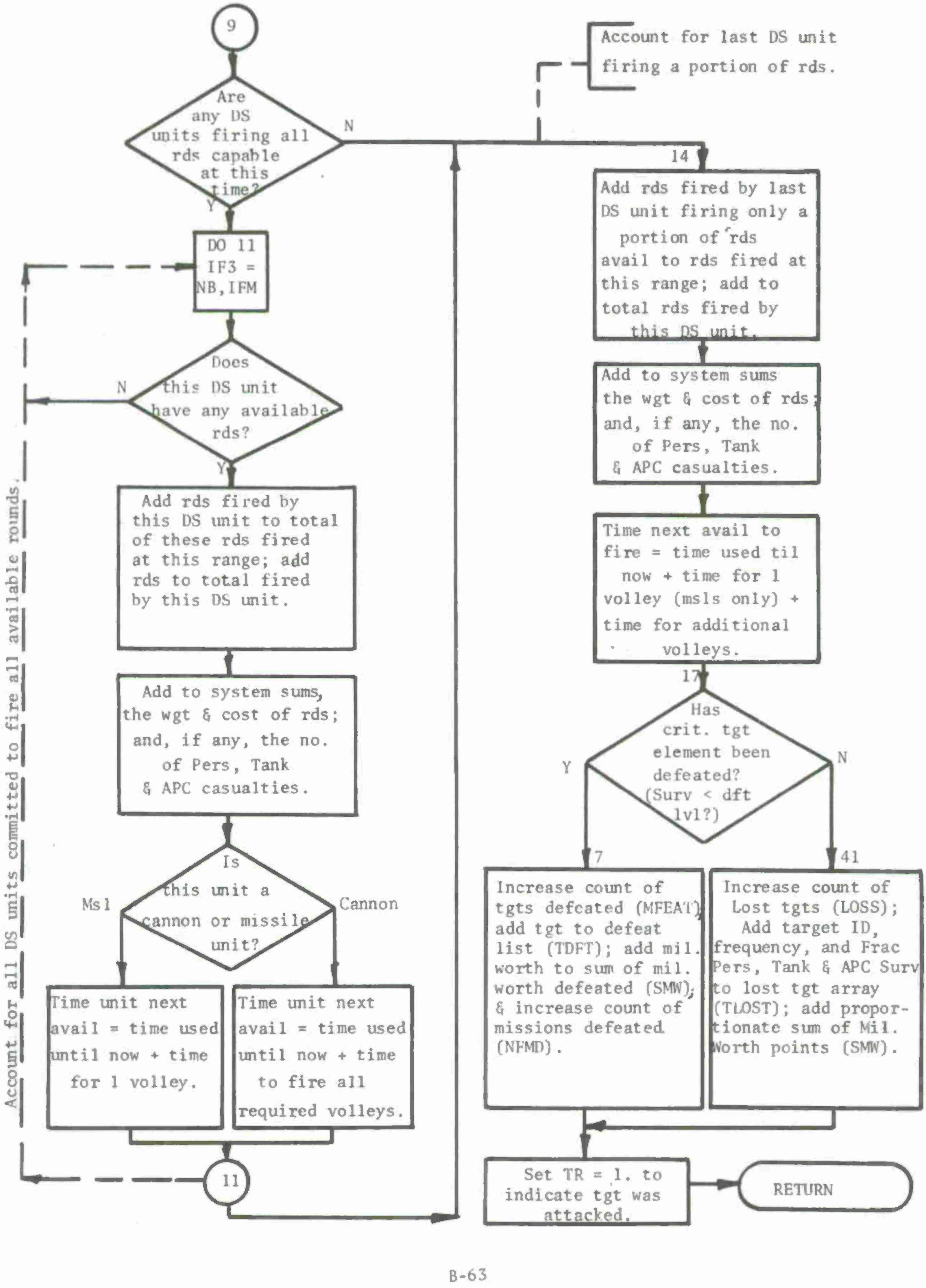

Fire 1 volley from DS & GS.
5
Set initial frac. Pers, Tank, & APC Survivors = 1. Define actual R_T, posture and environment.
Are any GS units to fire?
N
26
B-59
Y
Do 12 IF4 = NBA2, NEA2
Set TEMP = no. rd from 1 avail volley.
Is max no.rd unit can now deliver < TEMP?
Y
N
Has tgt left or does est. = act tgt data?
Y
N
INTERP
Determine P_K & R_E for this sys-rd vs actual tgt data.
B-47
EFFECT
Calculate effects for 1 volley vs actual tgt data.
B-93
Add effects of & account for next GS unit.
16
Sum frac surv. of 3 Pers, Tank and APC postures after 1 volley from this unit.
ASUM = Sum of 3 Pers posture survivors after this vol. (=1 if mat tgt)
Add rds fired (1 volley) by this GS unit to total of these rds fired at this range and add to rds fired by this unit.
Add to system sums, the wgt & cost of rds fired &, if any, the no. of pers, tank & APC casualties
Calculate new no. of pers, tank & APC remaining in tgt after these rds.
Time this unit next available = time used til now + time to fire 1 volley.
12
26
SHONVL
Add in effects of needed DS units firing 1 volley.
B-69
RETURN

If neither the one-volley nor multi-volley methods can accumulate enough rounds at the GS level to reach the specified attack level (and provided the ammunition overall weight constraint has not been exceeded) DS units are then examined in an attempt to supplement the GS units. If GS and DS units together can reach the attack level with either one-volley or multi-volley attack (without exceeding the weight constraints) then the appropriate subroutine is called to account for the GS and DS combined attack and appropriate counters are increased as above.

If neither attack method is successful for GS and DS units combined (whether for insufficient rounds available or for excessive ammunition weight) the CORP subroutine is called to attempt CORPS echelon attack.



The following page is blank.


9
Are any DS units firing all rds capable at this time?
N
Y
Account for last DS unit firing a portion of rds.
14
Add rds fired by last DS unit firing only a portion of rds avail to rds fired at this range; add to total rds fired by this DS unit.
Add to system sums the wgt & cost of rds; and, if any, the no. of Pers, Tank & APC casualties.
Time next avail to fire = time used til now + time for 1 volley (msls only) + time for additional volleys.
17
Has crit. tgt element been defeated? (Surv < dft lvl?)
Y
N
7
41
Increase count of tgts defeated (MFEAT); add tgt to defeat list (TDFT); add mil. worth to sum of mil. worth defeated (SMW); & increase count of missions defeated (NFMD).
Increase count of Lost tgts (LOSS); Add target ID, frequency, and Frac Pers, Tank & APC Surv to lost tgt array (TLOST); add proportionate sum of Mil. Worth points (SMW).
Set TR = 1. to indicate tgt was attacked.
RETURN
DO 11 IF3 = NB,IFM
Does this DS unit have any available rds?
N
Y
Add rds fired by this DS unit to total of these rds fired at this range; add rds to total fired by this DS unit.
Add to system sums, the wgt & cost of rds; and, if any, the no. of Pers, Tank & APC casualties.
Is this unit a cannon or missile unit?
Msl
Cannon
Time unit next avail = time used until now + time for 1 volley.
Time unit next avail = time used until now + time to fire all required volleys.
11
Account for all DS units committed to fire all available rounds.

Program Element: Shoot Multi-Volley

Symbolic Name: SHMUVL

Arguments in Call Statement:

(IT) - Identifies position on target list of target being attacked.

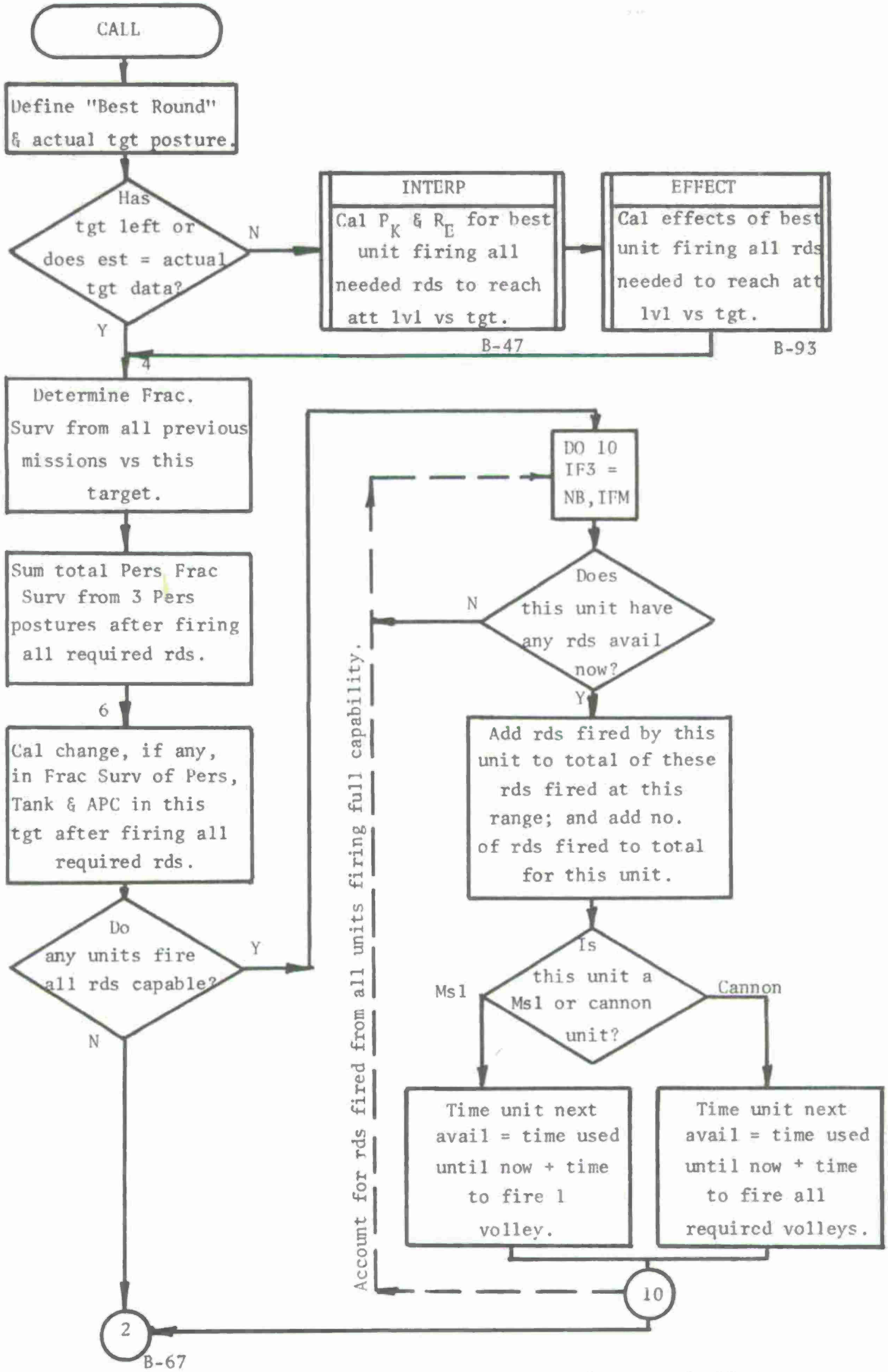
(JF) - Identifies first (best) Fire Unit to fire on this target.

Subroutines which call SHMUVL: DIVISN, CORP

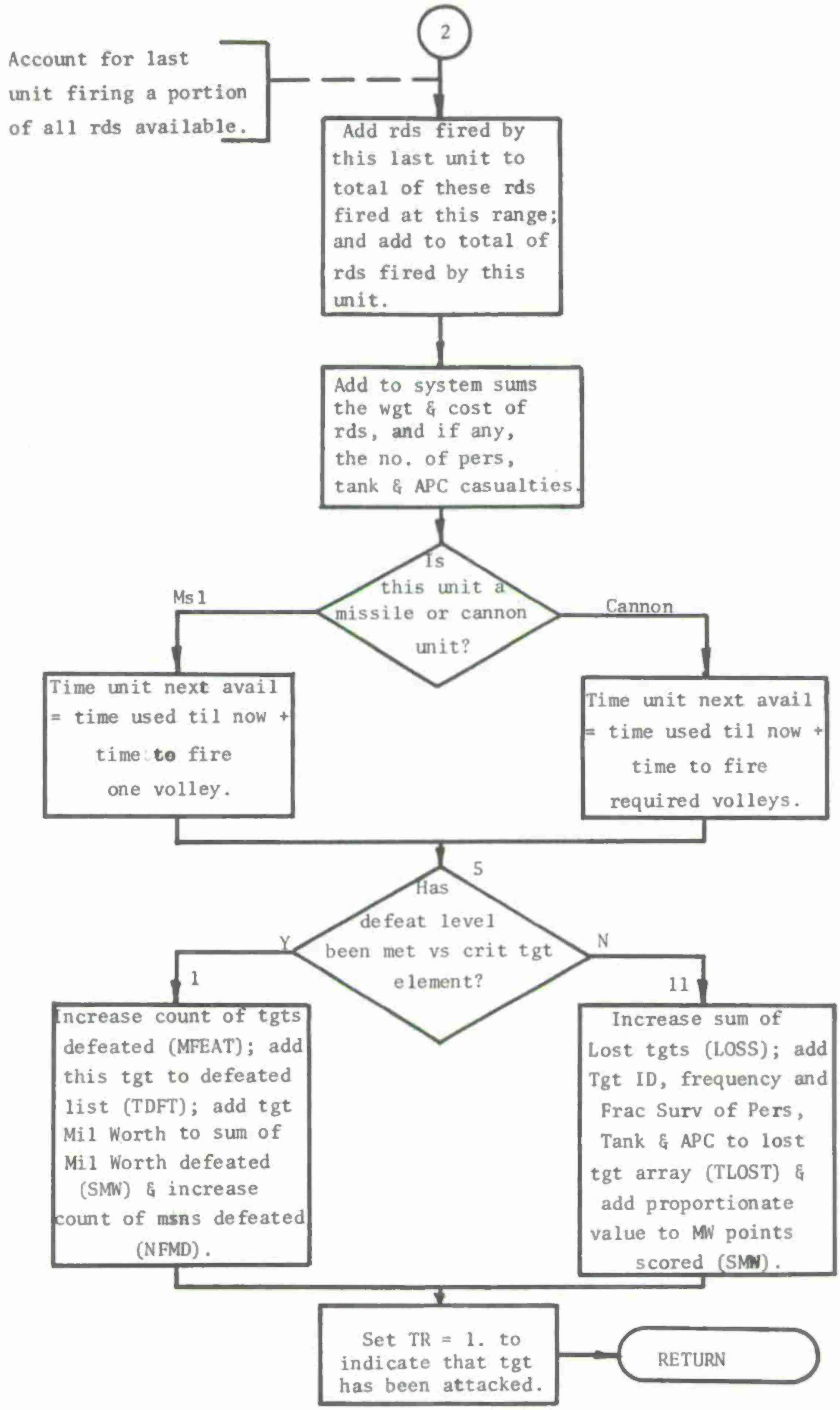
Subroutines called by SHMUVL: INTERP, EFFECT

This subroutine is called when DIVISN or CORP subroutines have determined that a target is to be attacked using the multi-volley method of attack. As explained in detail in the DIRSUP subroutine, an accounting is made of the damage inflicted on the target by each unit firing; also, the number of rounds fired and time used by each of the fire units at a particular echelon are credited to the appropriate units. (The particular echelon treated in this subroutine is the lowest ranking echelon which contributes to the attack, and therefore, that echelon whose last unit to contribute may fire only a portion of its available rounds.)

This SHMUVL subroutine also increases the "defeated" or "lost" list as appropriate, before returning control to the calling subroutine.



The following page is blank.


2
Account for last unit firing a portion of all rds available.
Add rds fired by this last unit to total of these rds fired at this range; and add to total of rds fired by this unit.
Add to system sums the wgt & cost of rds, and if any, the no. of pers, tank & APC casualties.
Is this unit a missile or cannon unit?
Msl
Cannon
Time unit next avail = time used til now + time to fire one volley.
Time unit next avail = time used til now + time to fire required volleys.
5
Has defeat level been met vs crit tgt element?
Y
N
1
11
Increase count of tgts defeated (MFEAT); add this tgt to defeated list (TDFT); add tgt Mil Worth to sum of Mil Worth defeated (SMW) & increase count of msns defeated (NFMD).
Increase sum of Lost tgts (LOSS); add Tgt ID, frequency and Frac Surv of Pers, Tank & APC to lost tgt array (TLOST) & add proportionate value to MW points scored (SMW).
Set TR = 1. to indicate that tgt has been attacked.
RETURN

Program Element: Shoot One-Volley

Symbolic Name: SHONVL

Arguments in Call Statement: (IT) - Identifies position on target list of the target being attacked.

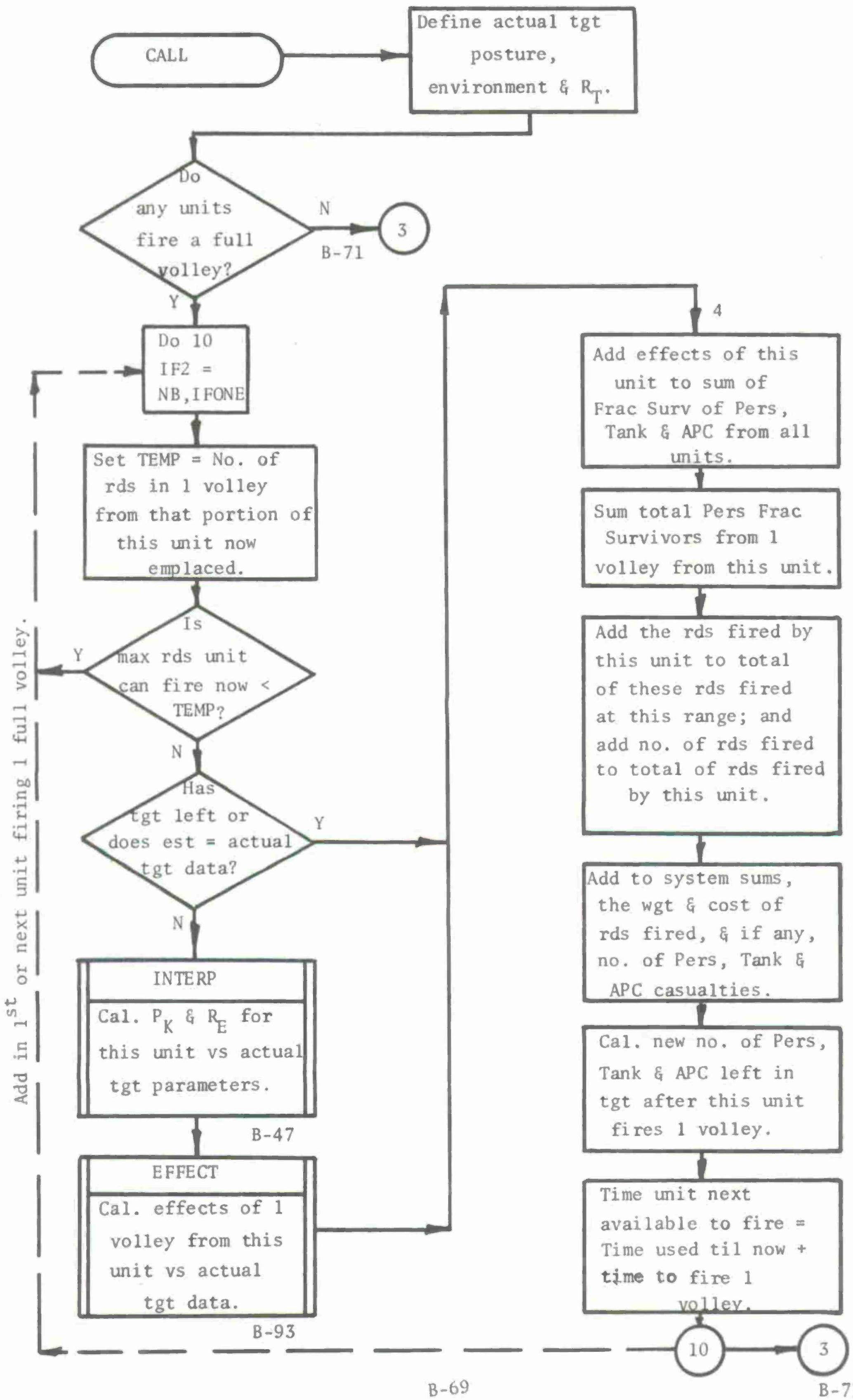
Subroutines which call SHONVL: DIVISN, CORP

Subroutines called by SHONVL: INTERP, EFFECT

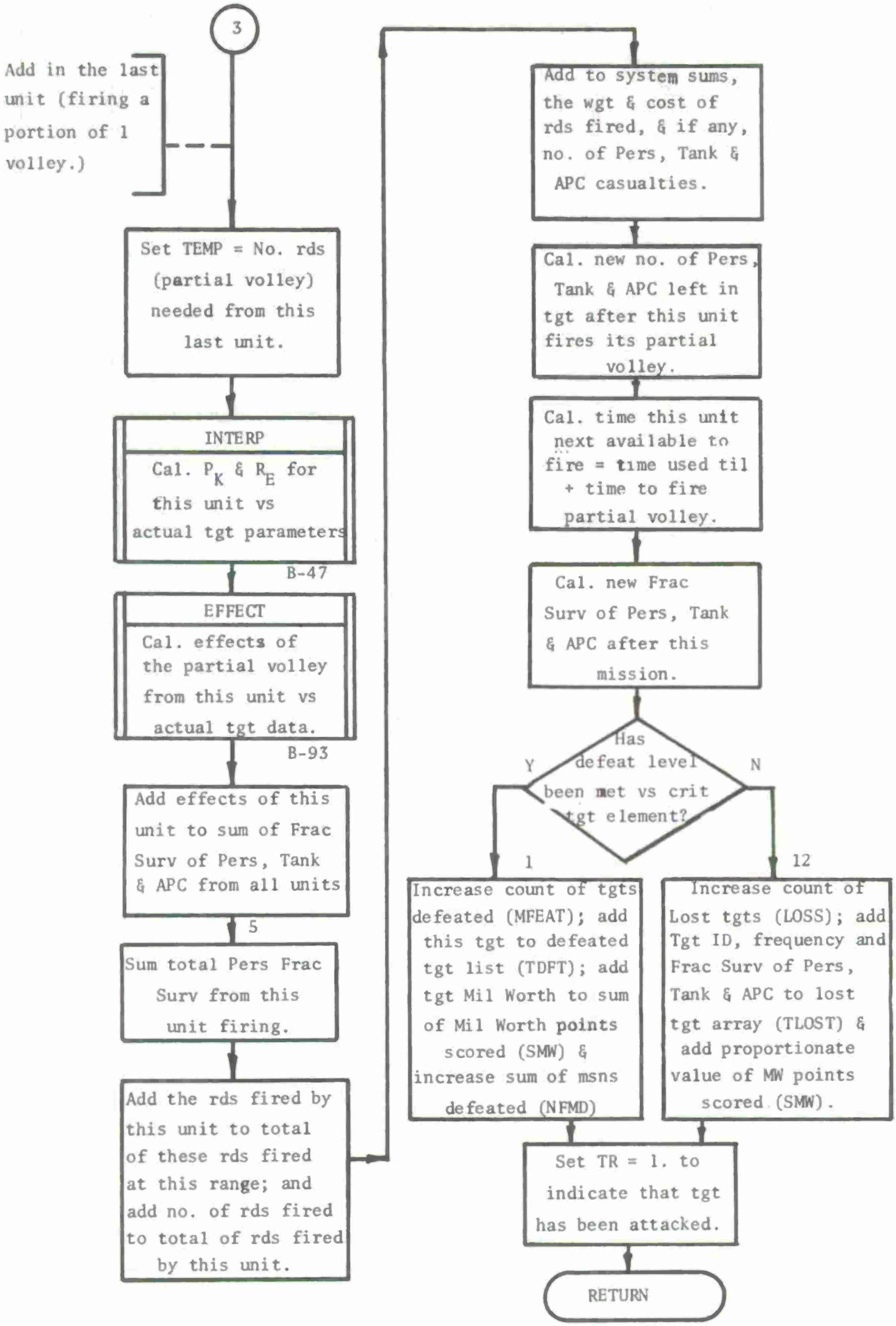
This subroutine is called when DIVISN or CORP subroutines have determined that a target is to be attacked using the one-volley method of attack. As explained in detail in the DIRSUP subroutine, an accounting is made of rounds fired, damage inflicted, and time used by each fire unit at a particular echelon. (The echelon considered is the lowest echelon which contributes to the attack, i.e., that echelon whose last unit to contribute may fire only a portion of a volley.)

The subroutine also increases the "defeated" or "lost" list as appropriate, before returning control to the calling subroutine.

Subroutine SHONVL



The following page is blank.



3
Add in the last unit (firing a portion of 1 volley.)
Set TEMP = No. rds (partial volley) needed from this last unit.
INTERP
Cal. P_K & R_E for this unit vs actual tgt parameters
B-47
EFFECT
Cal. effects of the partial volley from this unit vs actual tgt data.
B-93
Add effects of this unit to sum of Frac Surv of Pers, Tank & APC from all units
5
Sum total Pers Frac Surv from this unit firing.
Add the rds fired by this unit to total of these rds fired at this range; and add no. of rds fired to total of rds fired by this unit.
Add to system sums, the wgt & cost of rds fired, & if any, no. of Pers, Tank & APC casualties.
Cal. new no. of Pers, Tank & APC left in tgt after this unit fires its partial volley.
Cal. time this unit next available to fire = time used til + time to fire partial volley.
Cal. new Frac Surv of Pers, Tank & APC after this mission.
Has defeat level been met vs crit tgt element?
Y
N
1
12
Increase count of tgts defeated (MFEAT); add this tgt to defeated tgt list (TDFT); add tgt Mil Worth to sum of Mil Worth points scored (SMW) & increase sum of msns defeated (NFMD)
Increase count of Lost tgts (LOSS); add Tgt ID, frequency and Frac Surv of Pers, Tank & APC to lost tgt array (TLOST) & add proportionate value of MW points scored (SMW).
Set TR = 1. to indicate that tgt has been attacked.
RETURN

Program Element: Attempt One-Volley Method

Symbolic Name: ONEVOL

Arguments in Call Statement: (IT) - Identifies position on target list of the target being considered.

Subroutines which call ONEVOL: DIVISN, CORP

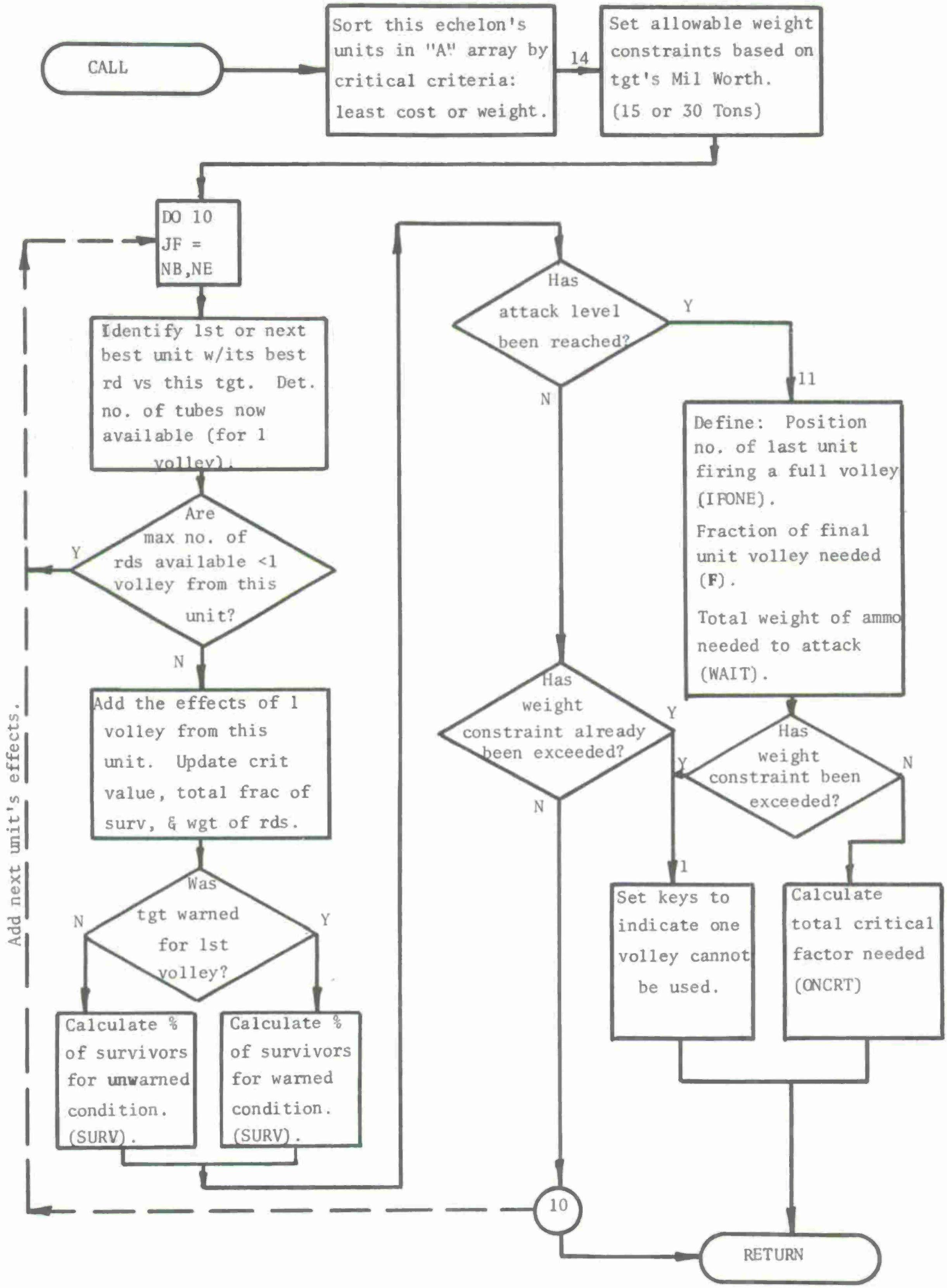
Subroutines called by ONEVOL: None

This subroutine is called by DIVISN and CORP subroutines to determine whether one-volley fired from each available fire unit (at specified echelon(s)) can reach the attack level needed to engage the target.

The subroutine begins by sorting the available fire units (as determined in AMASS) using the critical factor of least weight or least cost (as specified in RDMIX) as the sorting criteria. Units are then added in the sorted order until enough rounds are made available to reach the specified attack level. When the attack level is finally reached or exceeded by adding in the next available unit, a calculation is made to determine what fraction of that unit's rounds are required to just meet the attack level. Should the addition of a unit's rounds cause the overall weight constraint to be exceeded, keys are set to indicate that the one-volley method cannot be used against this target. If a unit does not have enough rounds on hand to fire one round from each of its guns, that unit is not permitted to add its rounds to the mission.

At the completion of the calculation, program control is returned to the calling subroutine.

Subroutine ONEVOL



Program Element: Attempt Multi-Volley Method

Symbolic Name: MULVOL

Arguments in Call Statement: (IT) - Identifies position on target list of the target being considered.

Subroutines which call MULVOL: DIVISN, CORP

Subroutines called by MULVOL: None

This subroutine is called by DIVISN and CORP subroutines to determine whether multi-volleys (all allowable, available rounds) from all available fire units (at specified echelon(s)) can reach the attack level needed to engage the target.

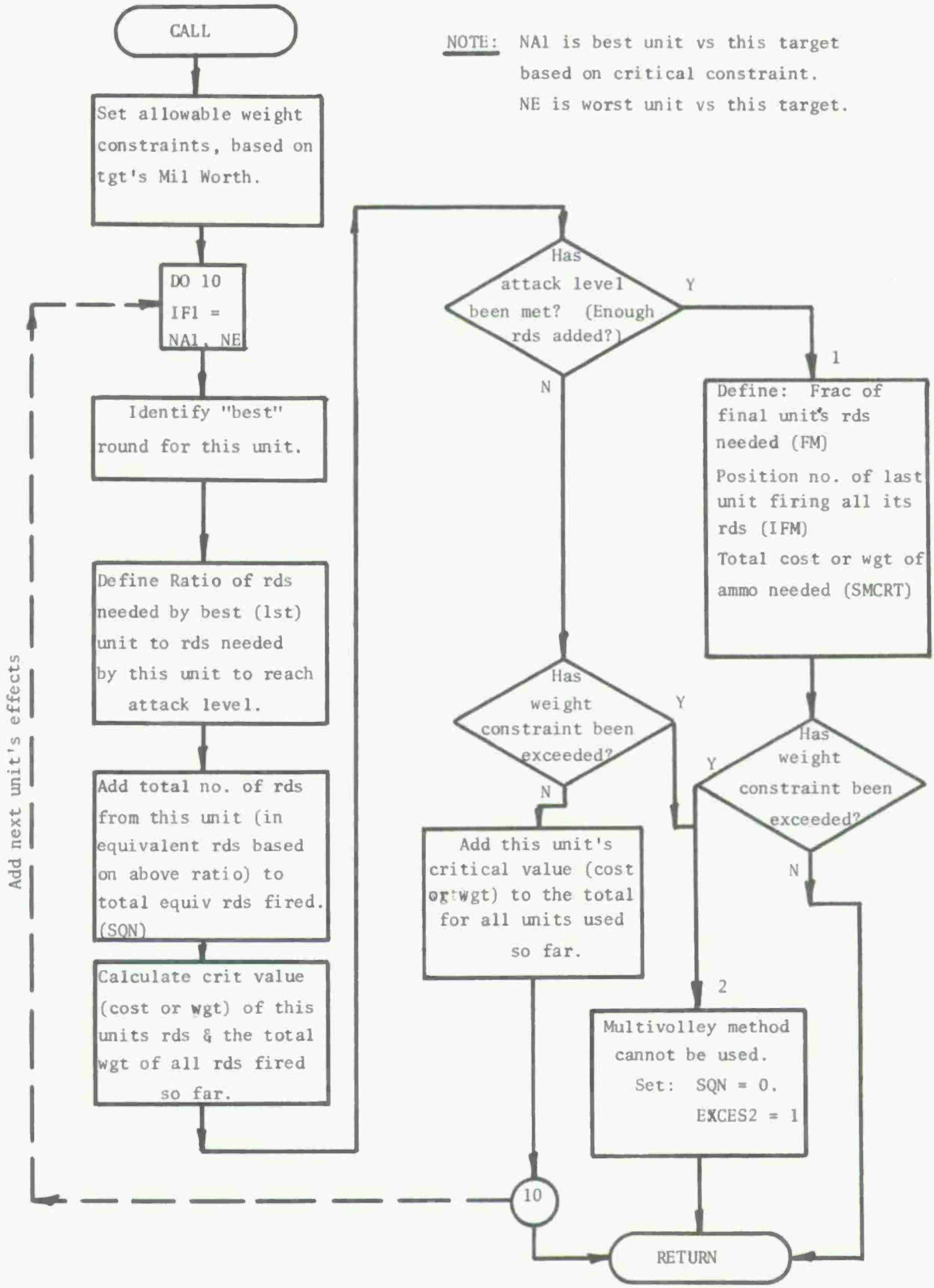
Units are added in the sorted order determined in ONEVOL, with each unit adding in all its available rounds (as determined by the fire unit and ammunition constraints in AMASS). The round for the first ("best") fire unit is set as the base round, against which the effects of other unit's rounds are compared, to establish their equivalent effects in terms of the "best" round.

As with the ONEVOL subroutine, units are added until sufficient "equivalent best rounds" are available to reach the specified attack level. The fractional part of the last unit's rounds needed to just meet the attack level is also calculated.

If the overall weight constraint is exceeded by the addition of any unit's rounds, keys are set to indicate that the multi-volley method cannot be used against this target.

At the completion of the calculation, control is returned to the calling subroutine.

Subroutine MULVOL



Program Element: CORPS Echelon

Symbolic Name: CORP

Arguments in Call Statement: (IT) - Identifies position on target list of the target being considered.

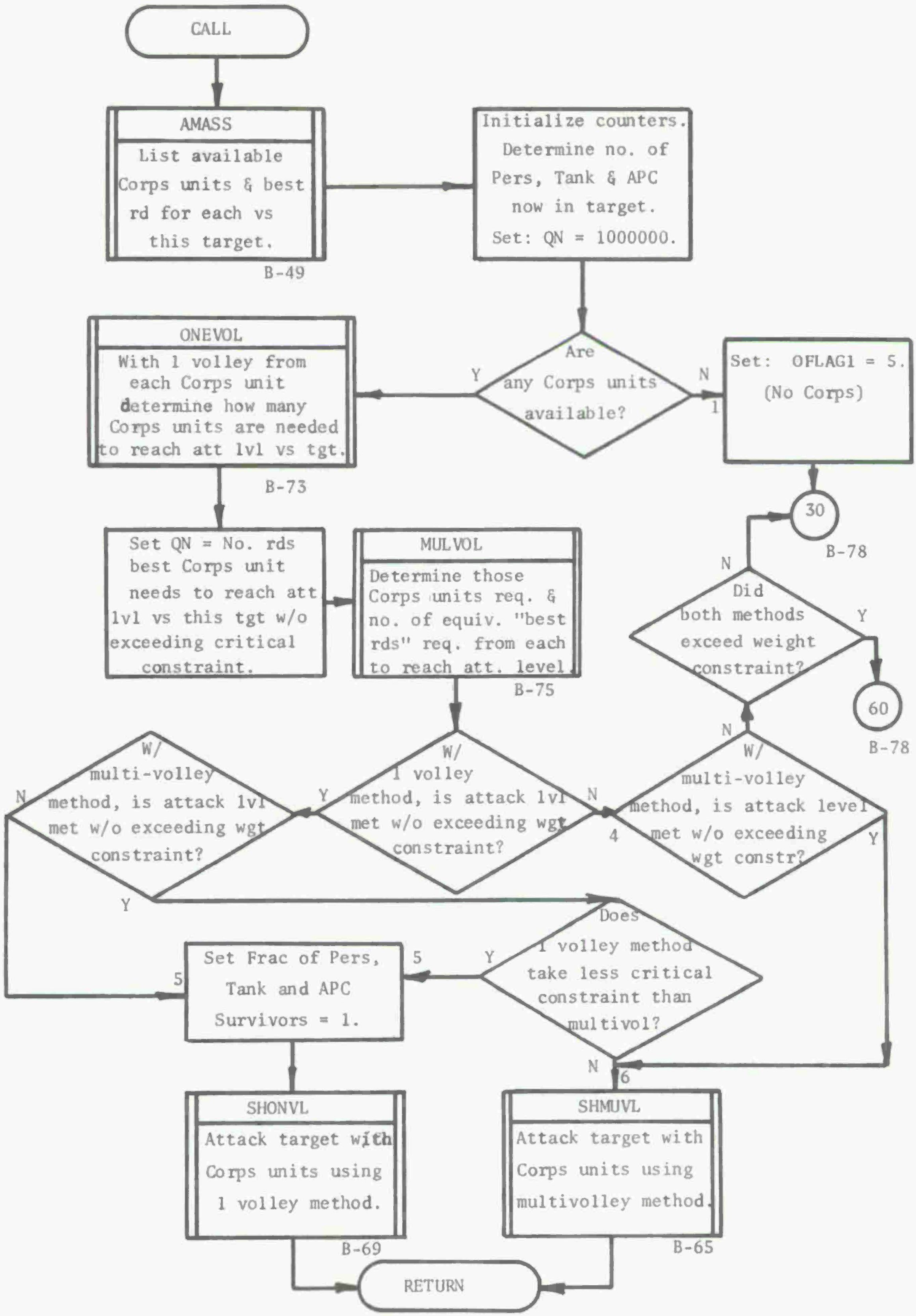
Subroutines which call CORP: Main Program, DIVISN

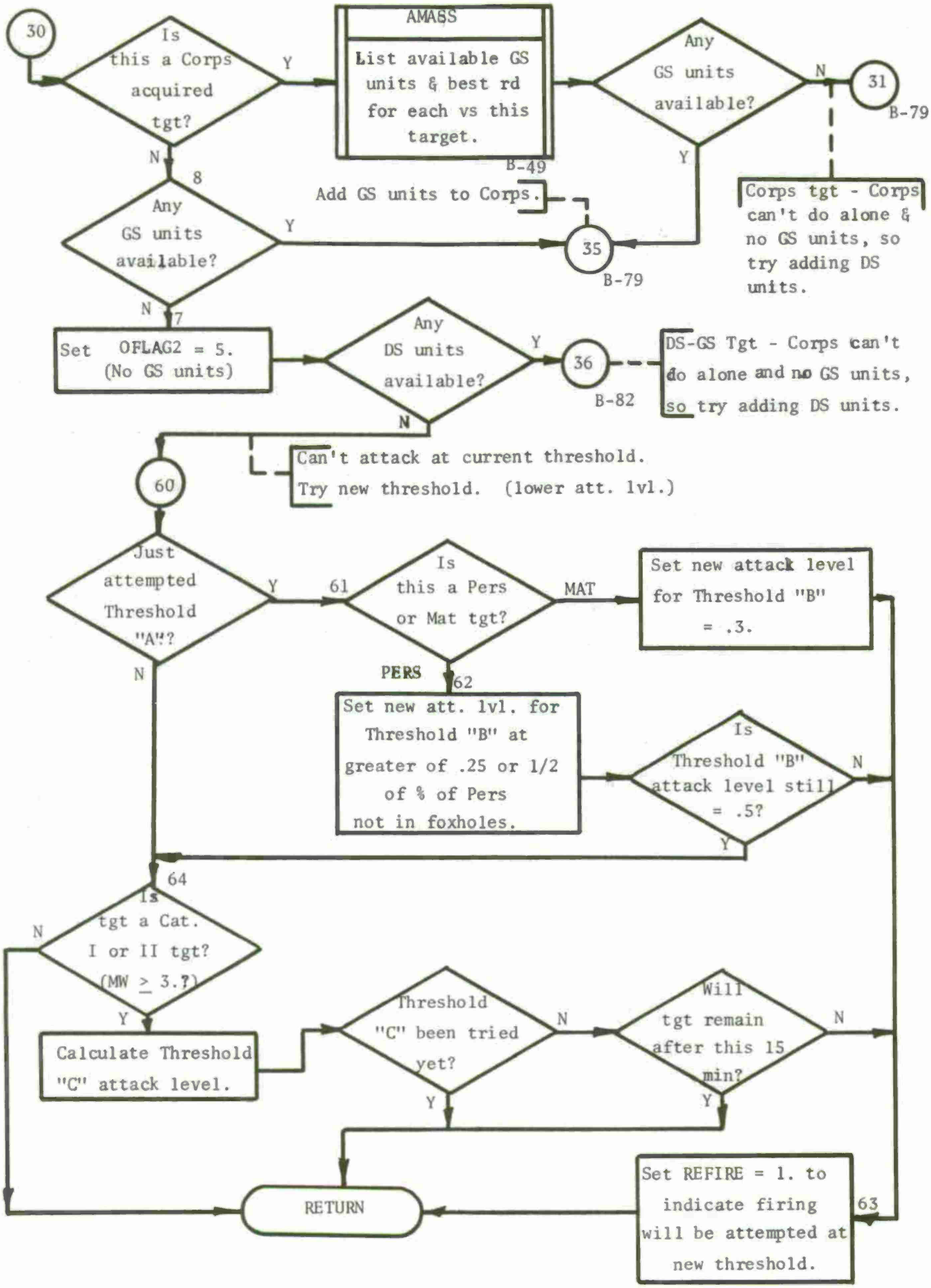
Subroutines called by CORP: AMASS, ONEVOL, MULVOL, SHMUVL, SHONVL, INTERP, EFFECT

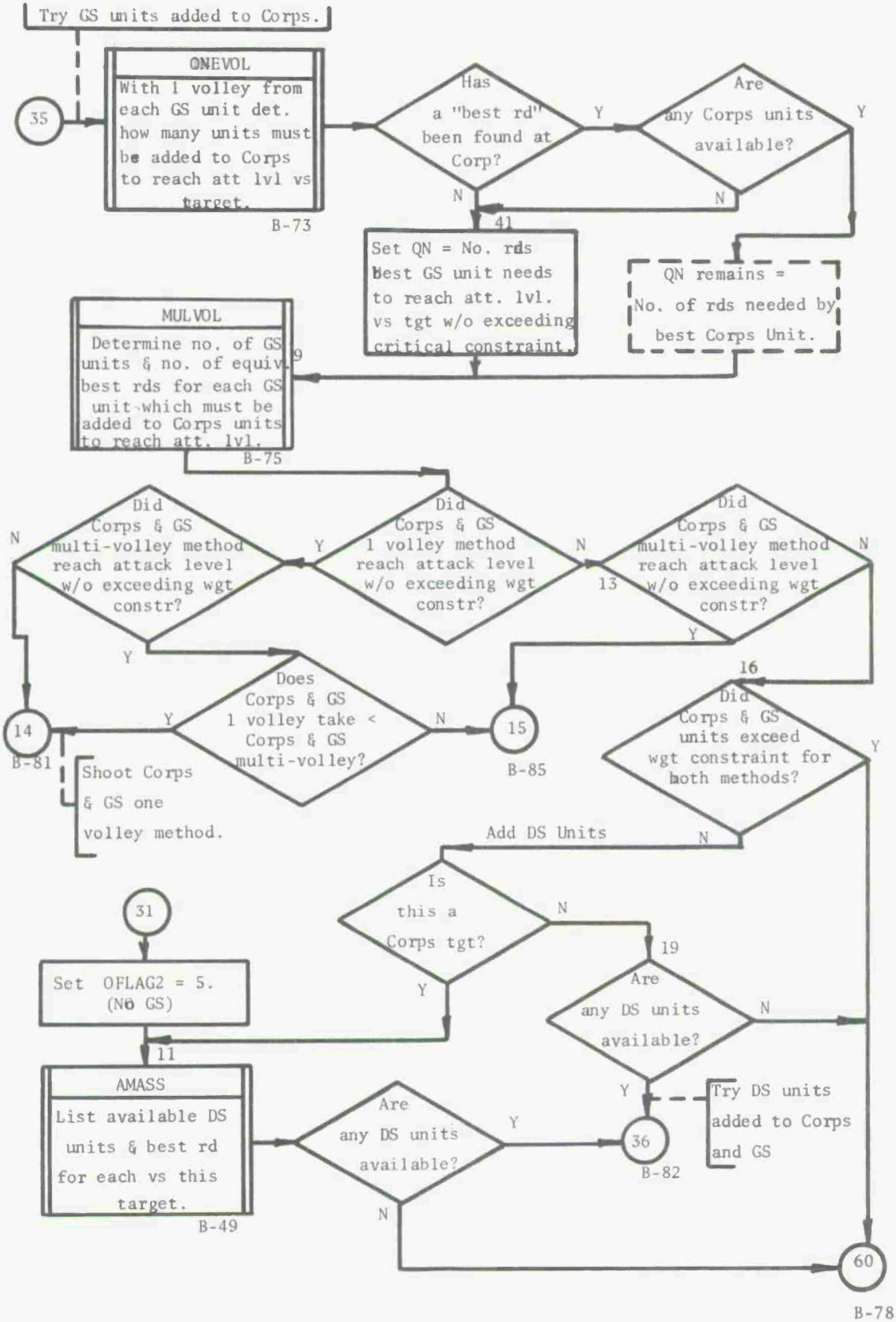
This subroutine, when called from the main program, provides the initial attempt to fire upon Corps-acquired targets; and, when called from DIVISN, it attempts to fire upon DS-and GS-acquired targets that cannot be attacked at the DS and GS levels.

After AMASS is called to provide a list of available Corps units, one-volley (ONEVOL) and multi-volley (MULVOL) methods of attack are first examined for Corps echelon units only. If attack levels cannot be reached by the Corps units, then available GS and finally DS units are added in turn. Whenever the attack level can be met within the overall weight constraint by either method of attack, appropriate calculations are made to credit the damage inflicted, rounds fired and time used by each fire unit participating. If both attack methods meet the attack level, then the one using least ammunition cost or weight, depending on the criteria, is chosen. (Appropriate subroutines (AMASS, ONEVOL, MULVOL, SHMUVL and SHONVL) are called as necessary to provide the needed calculations.)

If a search through all units (Corps, GS and DS) cannot mass sufficient rounds, or if the overall weight constraint is exceeded at


CALL
AMASS
List available Corps units & best rd for each vs this target.
B-49
Initialize counters. Determine no. of Pers, Tank & APC now in target. Set: QN = 1000000.
ONEVOL
With 1 volley from each Corps unit determine how many Corps units are needed to reach att lvl vs tgt.
B-73
Are any Corps units available?
Y
N
1
Set: OFLAG1 = 5. (No Corps)
30
B-78
Set QN = No. rds best Corps unit needs to reach att lvl vs this tgt w/o exceeding critical constraint.
MULVOL
Determine those Corps units req. & no. of equiv. "best rds" req. from each to reach att. level.
B-75
Did both methods exceed weight constraint?
N
Y
60
B-78
W/ multi-volley method, is attack lvl met w/o exceeding wgt constraint?
N
Y
W/ 1 volley method, is attack lvl met w/o exceeding wgt constraint?
Y
N
4
W/ multi-volley method, is attack level met w/o exceeding wgt constr?
N
Y
Does 1 volley method take less critical constraint than multivol?
Y
N
5
Set Frac of Pers, Tank and APC Survivors = 1.
5
6
SHONVL
Attack target with Corps units using 1 volley method.
B-69
SHMUVL
Attack target with Corps units using multivolley method.
B-65
RETURN

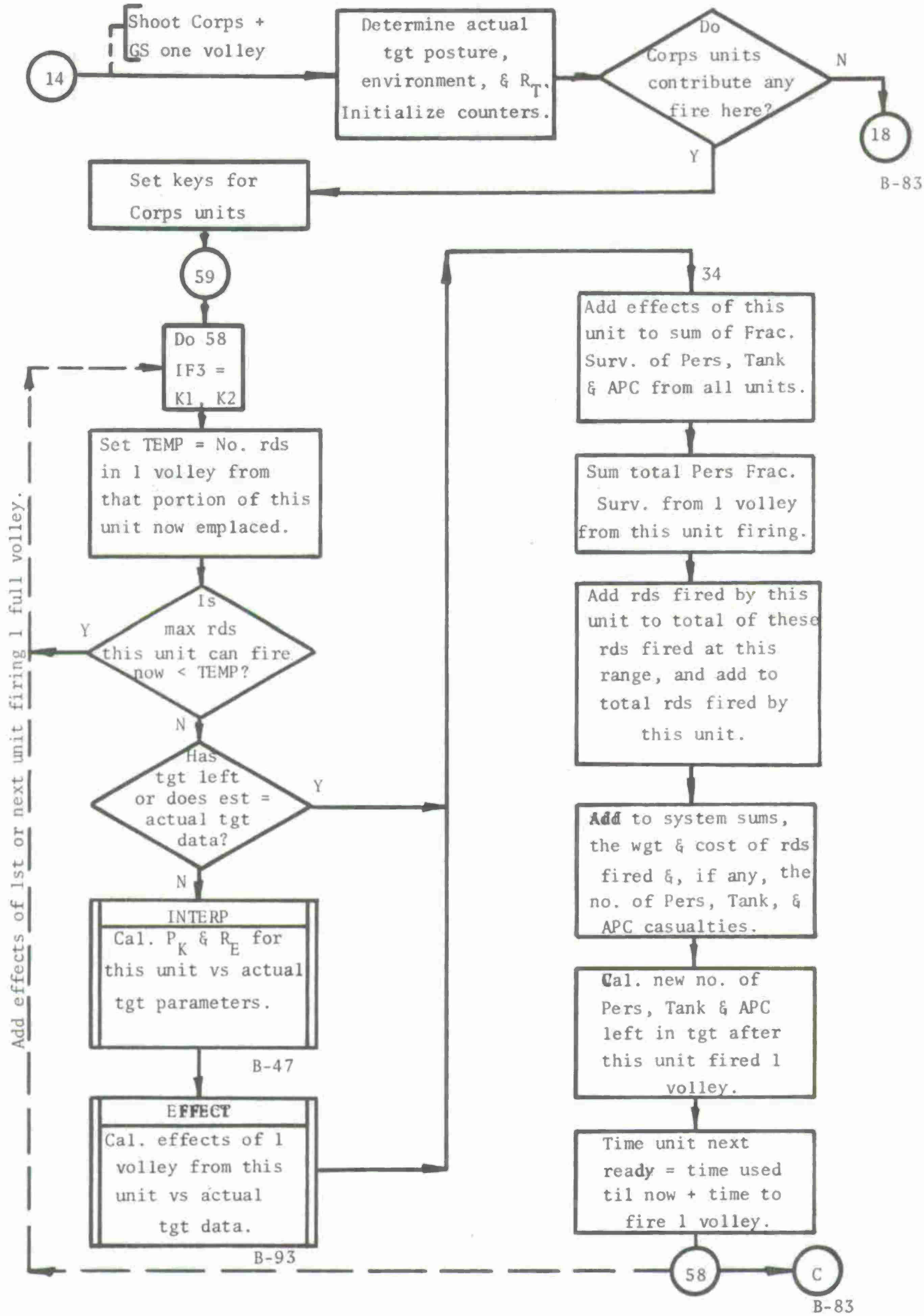

30
Is this a Corps acquired tgt?
Y
N
AMASS
List available GS units & best rd for each vs this target.
B-49
Any GS units available?
N
31
B-79
Y
Corps tgt - Corps can't do alone & no GS units, so try adding DS units.
8
Add GS units to Corps.
Any GS units available?
Y
35
B-79
N
7
Set OFLAG2 = 5. (No GS units)
Any DS units available?
Y
36
B-82
DS-GS Tgt - Corps can't do alone and no GS units, so try adding DS units.
N
Can't attack at current threshold. Try new threshold. (lower att. lvl.)
60
Just attempted Threshold "A"?
Y
61
Is this a Pers or Mat tgt?
MAT
Set new attack level for Threshold "B" = .3.
N
PERS
62
Set new att. lvl. for Threshold "B" at greater of .25 or 1/2 of % of Pers not in foxholes.
Is Threshold "B" attack level still = .5?
N
Y
64
Is tgt a Cat. I or II tgt? (MW ≥ 3.7)
N
Y
Calculate Threshold "C" attack level.
Threshold "C" been tried yet?
N
Y
Will tgt remain after this 15 min?
N
Y
RETURN
Set REFIRE = 1. to indicate firing will be attempted at new threshold.
63

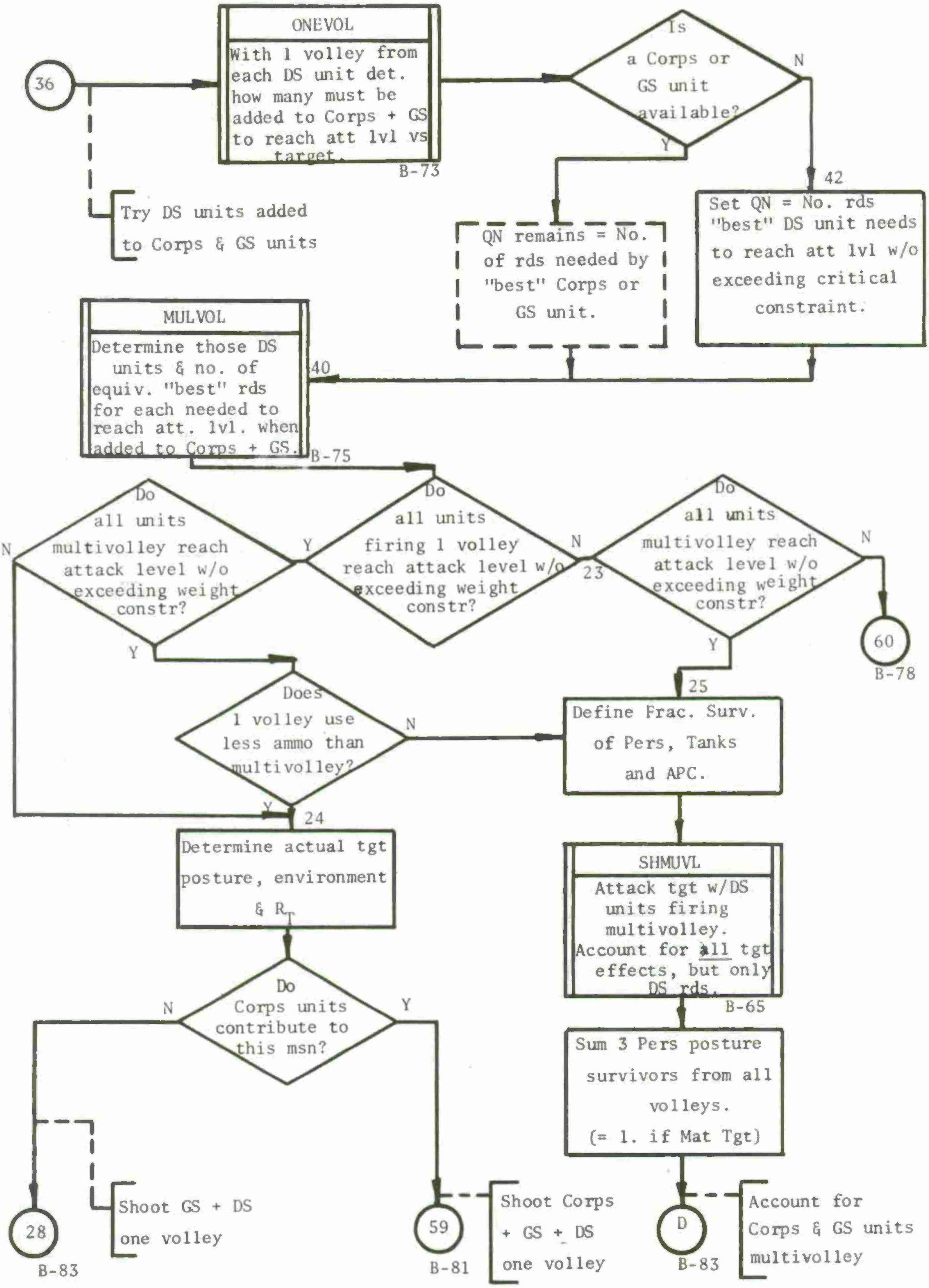

Try GS units added to Corps.
35
ONEVOL
With 1 volley from each GS unit det. how many units must be added to Corps to reach att lvl vs target.
B-73
Has a "best rd" been found at Corp?
Y
N
Are any Corps units available?
Y
N
41
Set QN = No. rds best GS unit needs to reach att. lvl. vs tgt w/o exceeding critical constraint.
QN remains = No. of rds needed by best Corps Unit.
MULVOL
Determine no. of GS units & no. of equiv. best rds for each GS unit which must be added to Corps units to reach att. lvl.
9
B-75
Did Corps & GS multi-volley method reach attack level w/o exceeding wgt constr?
N
Y
Did Corps & GS 1 volley method reach attack level w/o exceeding wgt constr?
Y
N
13
Did Corps & GS multi-volley method reach attack level w/o exceeding wgt constr?
N
Y
16
Does Corps & GS 1 volley take < Corps & GS multi-volley?
Y
N
14
B-81
15
B-85
Shoot Corps & GS one volley method.
Did Corps & GS units exceed wgt constraint for both methods?
Y
N
Add DS Units
Is this a Corps tgt?
N
Y
19
Are any DS units available?
N
Y
31
Set OFLAG2 = 5. (NO GS)
11
AMASS
List available DS units & best rd for each vs this target.
B-49
Are any DS units available?
Y
N
36
B-82
Try DS units added to Corps and GS
60
B-78

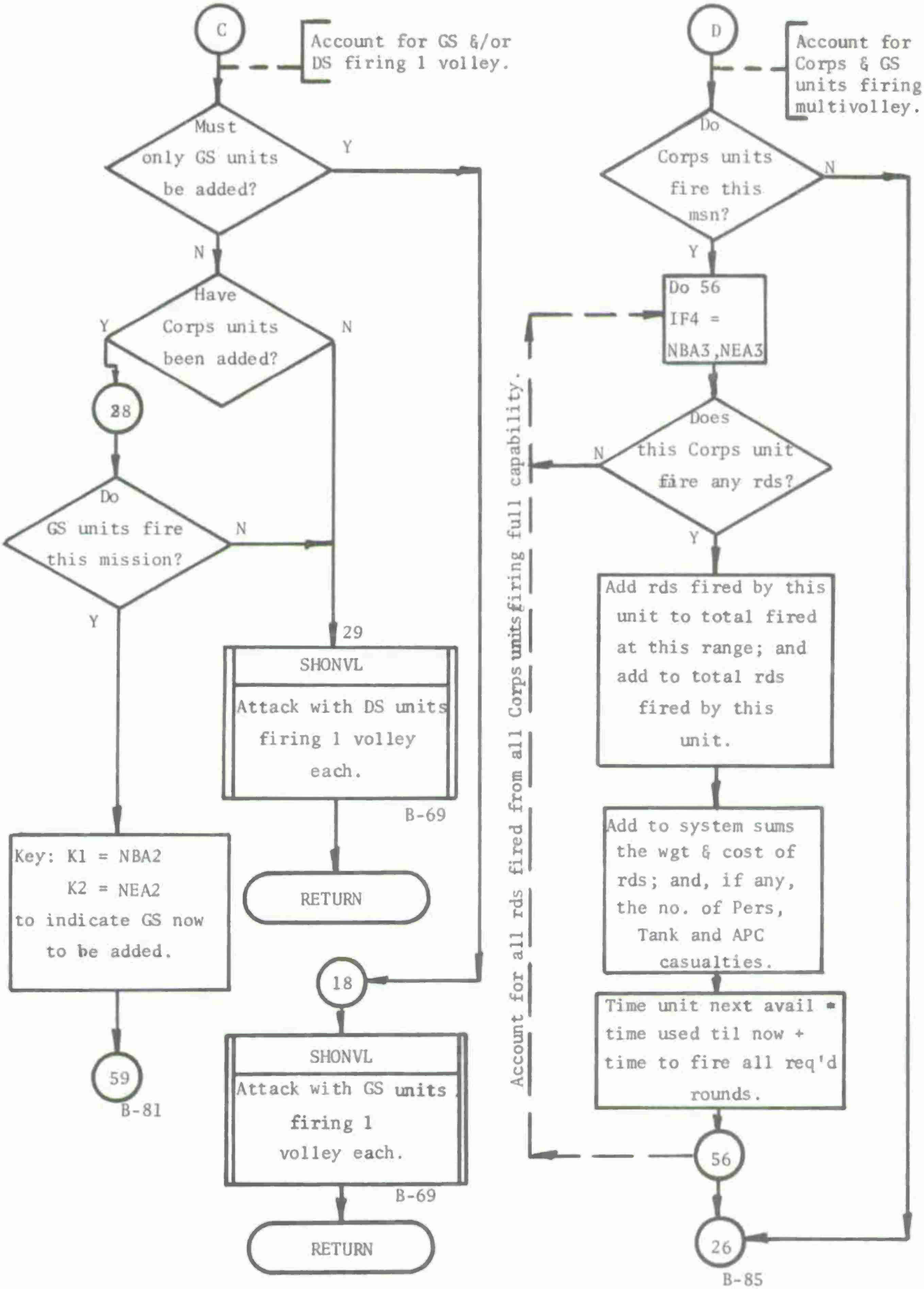
any time, the target cannot be attacked at the specified attack level.

If the initial attempt (at Threshold A-50% attack level) fails, CORP subroutine then calculates a new (lower) attack level (Threshold B) and returns program control to the main program to immediately reattempt attack at the lower level. If attempts to fire at this lower attack level also fail, then the target is not attacked at this time and it will remain on the target list for consideration during subsequent 15-minute game intervals. However, for Category I and II targets only, and only if the target will depart its location during the current 15-minute game interval, a third attempt to fire is made after the first two attempts fail. A Threshold C attack level is calculated and a final attempt at attack is made.

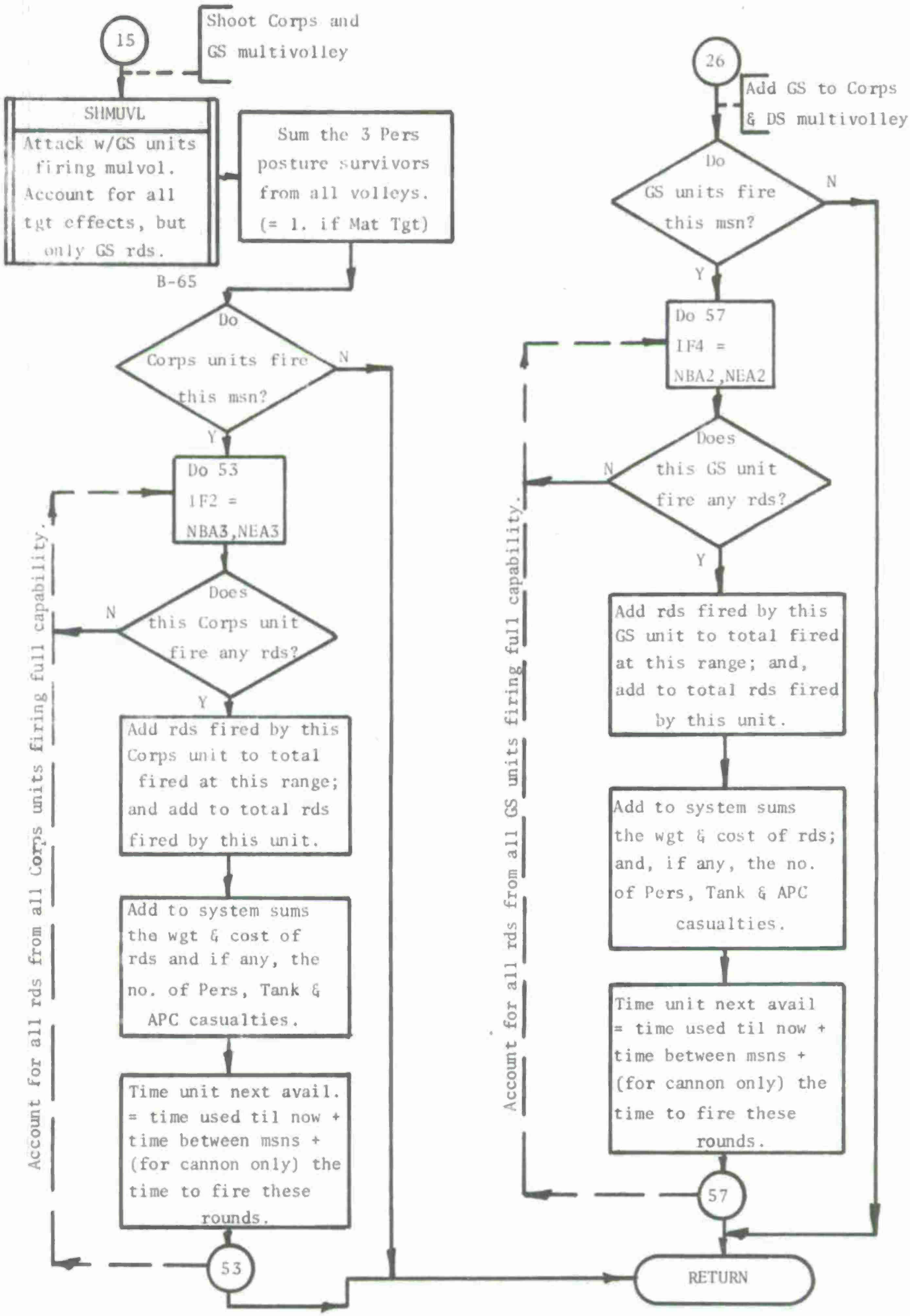
At the completion of CORP subroutine, control returns to the main program.


14
Shoot Corps + GS one volley
Determine actual tgt posture, environment, & R_T. Initialize counters.
Do Corps units contribute any fire here?
N
18
B-83
Y
Set keys for Corps units
59
Do 58 IF3 = K1, K2
Set TEMP = No. rds in 1 volley from that portion of this unit now emplaced.
Is max rds this unit can fire now < TEMP?
Y
N
Has tgt left or does est = actual tgt data?
Y
N
INTERP
Cal. P_K & R_E for this unit vs actual tgt parameters.
B-47
EFFECT
Cal. effects of 1 volley from this unit vs actual tgt data.
B-93
Add effects of 1st or next unit firing 1 full volley.
34
Add effects of this unit to sum of Frac. Surv. of Pers, Tank & APC from all units.
Sum total Pers Frac. Surv. from 1 volley from this unit firing.
Add rds fired by this unit to total of these rds fired at this range, and add to total rds fired by this unit.
Add to system sums, the wgt & cost of rds fired &, if any, the no. of Pers, Tank, & APC casualties.
Cal. new no. of Pers, Tank & APC left in tgt after this unit fired 1 volley.
Time unit next ready = time used til now + time to fire 1 volley.
58
C
B-83


36
ONEVOL
With 1 volley from each DS unit det. how many must be added to Corps + GS to reach att lvl vs target.
B-73
Is a Corps or GS unit available?
N
Y
Try DS units added to Corps & GS units
QN remains = No. of rds needed by "best" Corps or GS unit.
42
Set QN = No. rds "best" DS unit needs to reach att lvl w/o exceeding critical constraint.
MULVOL
Determine those DS units & no. of equiv. "best" rds for each needed to reach att. lvl. when added to Corps + GS.
40
B-75
Do all units multivolley reach attack level w/o exceeding weight constr?
N
Y
Do all units firing 1 volley reach attack level w/o exceeding weight constr?
Y
N
23
Do all units multivolley reach attack level w/o exceeding weight constr?
N
Y
60
B-78
Does 1 volley use less ammo than multivolley?
N
Y
25
Define Frac. Surv. of Pers, Tanks and APC.
24
Determine actual tgt posture, environment & R_T
SHMUVL
Attack tgt w/DS units firing multivolley. Account for all tgt effects, but only DS rds.
B-65
Do Corps units contribute to this msn?
N
Y
Sum 3 Pers posture survivors from all volleys. (= 1. if Mat Tgt)
28
B-83
Shoot GS + DS one volley
59
B-81
Shoot Corps + GS + DS one volley
D
B-83
Account for Corps & GS units multivolley



The following page is blank.


15
Shoot Corps and
GS multivolley
SHMUVL
Attack w/GS units
firing mulvol.
Account for all
tgt effects, but
only GS rds.
B-65
Sum the 3 Pers
posture survivors
from all volleys.
(= 1. if Mat Tgt)
Do
Corps units fire
this msn?
N
Y
Do 53
IF2 =
NBA3,NEA3
Does
this Corps unit
fire any rds?
N
Y
Add rds fired by this
Corps unit to total
fired at this range;
and add to total rds
fired by this unit.
Add to system sums
the wgt & cost of
rds and if any, the
no. of Pers, Tank &
APC casualties.
Time unit next avail.
= time used til now +
time between msns +
(for cannon only) the
time to fire these
rounds.
53
Account for all rds from all Corps units firing full capability.
26
Add GS to Corps
& DS multivolley
Do
GS units fire
this msn?
N
Y
Do 57
IF4 =
NBA2,NEA2
Does
this GS unit
fire any rds?
N
Y
Add rds fired by this
GS unit to total fired
at this range; and,
add to total rds fired
by this unit.
Add to system sums
the wgt & cost of rds;
and, if any, the no.
of Pers, Tank & APC
casualties.
Time unit next avail
= time used til now +
time between msns +
(for cannon only) the
time to fire these
rounds.
57
Account for all rds from all GS units firing full capability.
RETURN

Program Element: Calculate Effects

Symbolic Name: EFFECT

Arguments in Call Statement:

(IA) - Identifies which fire unit on "A" array is being considered.

(KR) - Identifies which round is being considered.

(IT) - Identifies position on target list of the target being considered.

Subroutines which call EFFECT: DIRSUP, AMASS, DIVISN, SHMUVL, SHONVL, CORP

Subroutines called by EFFECT: COV

This subroutine performs effects calculations for two situations, depending upon the calling subroutine. When called from AMASS, this subroutine determines, for the particular fire unit, round and target under consideration, 12 elements of datum for the "A" array associated with the fire unit. Estimated target data are used in these calculations. When called from the other listed subroutines, however, EFFECT calculates, based upon actual target data, the actual effects on the target of a previously calculated number of rounds the unit will fire.

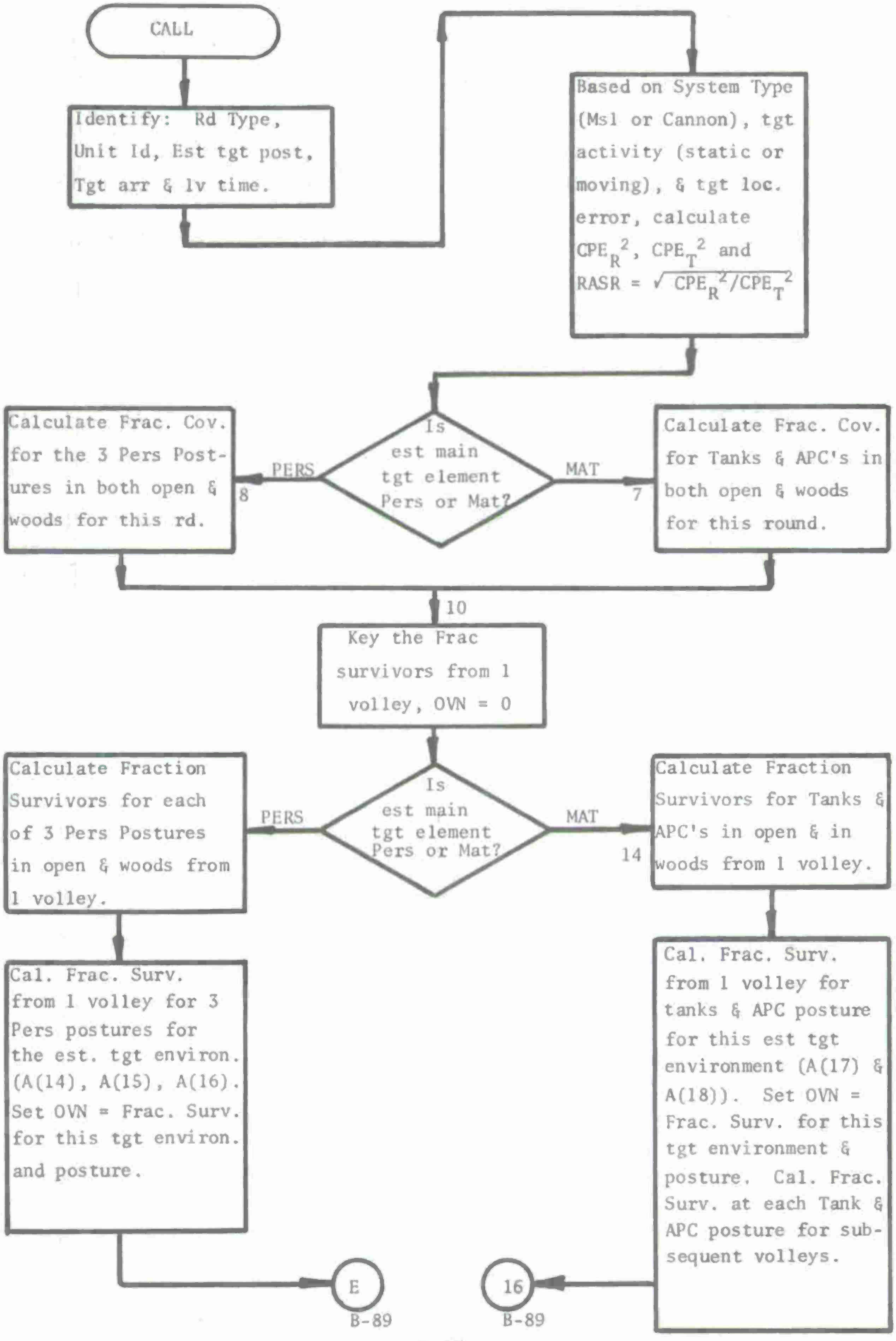
The two processes are outlined below:

A. When called from AMASS

In this situation EFFECT is used to calculate the number of rounds required by a unit to reach the attack level, if that unit were firing by itself. (In essence, the potential of the unit to contribute to a mission is being evaluated, since these results will

Subroutine EFFECT

(Called from AMASS - Uses estimated tgt data)

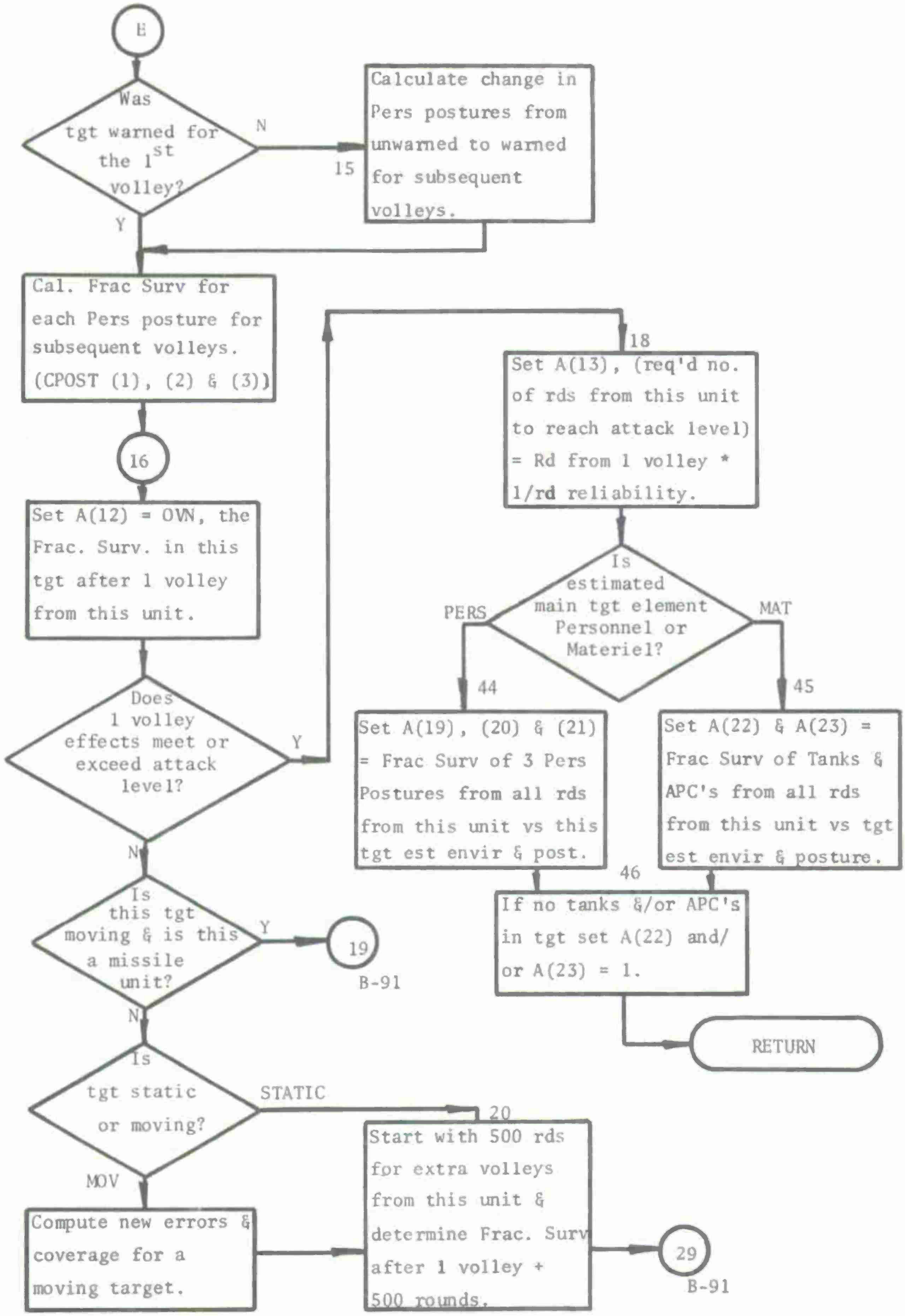


be compared with other units' results to choose an optimum "mix" of firing units.)

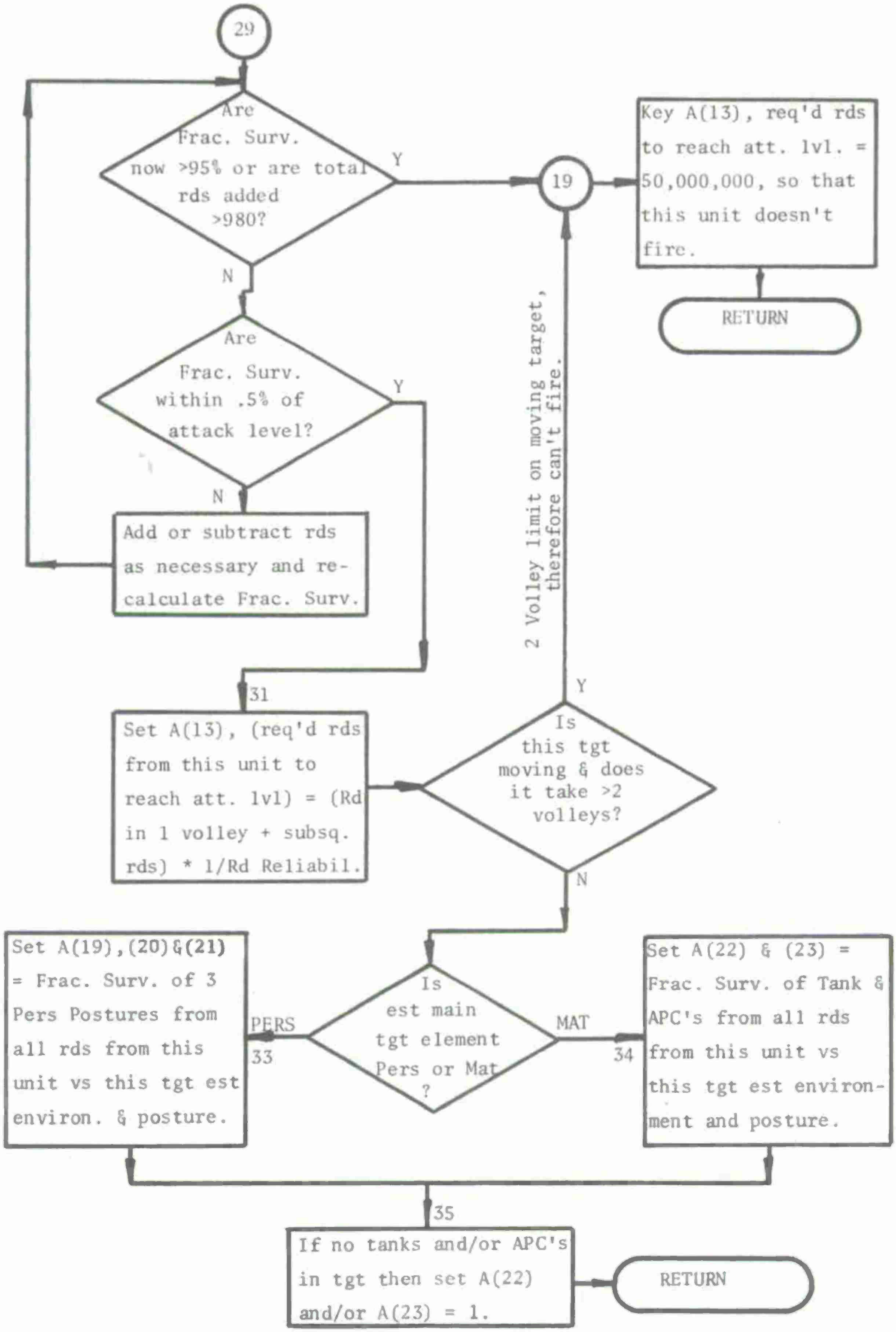
After defining the round, unit, estimated target posture, target arrival and departure times and delivery errors, the subroutine calculates for the given unit and target parameters, the fractional coverages (COV) which one round will provide against general target elements (personnel or materiel) in both the open and wooded environments. Then the fractional survivors in the specific target are calculated for this unit firing one-volley of this round.

For non-observed fire missions, if the round is the HE-type, the change in personnel posture from unwarned to warned is then assigned, before calculating effects of subsequent volleys. Likewise, new errors are assigned if the target is moving. If the one-volley target damage (effects) meets the attack level then appropriate elements of the "A" array are calculated to indicate the fractional survivors of target elements (personnel or materiel) and the required number of rounds is set at one-volley. Otherwise the subroutine begins an iteration process to determine how many rounds must be added to one-volley in order to reach the specified attack level. (A missile unit is dropped from consideration if it requires more than one volley and if the target is moving, since the time to prepare for subsequent volleys is excessive for missile units.)

When the iteration process has determined the number of rounds needed to reach the attack level, appropriate elements of the "A"


E
Was tgt warned for the 1st volley?
N
15
Calculate change in Pers postures from unwarned to warned for subsequent volleys.
Y
Cal. Frac Surv for each Pers posture for subsequent volleys. (CPOST (1), (2) & (3))
16
Set A(12) = OVN, the Frac. Surv. in this tgt after 1 volley from this unit.
Does 1 volley effects meet or exceed attack level?
Y
18
Set A(13), (req'd no. of rds from this unit to reach attack level) = Rd from 1 volley * 1/rd reliability.
Is estimated main tgt element Personnel or Materiel?
PERS
MAT
44
45
Set A(19), (20) & (21) = Frac Surv of 3 Pers Postures from all rds from this unit vs this tgt est envir & post.
Set A(22) & A(23) = Frac Surv of Tanks & APC's from all rds from this unit vs tgt est envir & posture.
46
If no tanks &/or APC's in tgt set A(22) and/or A(23) = 1.
RETURN
N
Is this tgt moving & is this a missile unit?
Y
19
B-91
N
Is tgt static or moving?
STATIC
20
Start with 500 rds for extra volleys from this unit & determine Frac. Surv after 1 volley + 500 rounds.
MOV
Compute new errors & coverage for a moving target.
29
B-91

array are calculated to indicate fractional survivors and total rounds required, as done above for the one-volley solution. A unit is dropped from consideration (because of inefficiency) if it requires over 980 rounds to reach the attack level, if an additional 500 rounds cannot achieve at least 5% damage or if more than two volleys are required against a moving target. In all cases control is returned to the AMASS subroutine.


29
Are Frac. Surv. now >95% or are total rds added >980?
Y
N
19
Key A(13), req'd rds to reach att. lvl. = 50,000,000, so that this unit doesn't fire.
RETURN
Are Frac. Surv. within .5% of attack level?
Y
N
Add or subtract rds as necessary and re-calculate Frac. Surv.
2 Volley limit on moving target, therefore can't fire.
31
Set A(13), (req'd rds from this unit to reach att. lvl) = (Rd in 1 volley + subsq. rds) * 1/Rd Reliabil.
Is this tgt moving & does it take >2 volleys?
Y
N
Set A(19),(20)&(21) = Frac. Surv. of 3 Pers Postures from all rds from this unit vs this tgt est environ. & posture.
PERS
33
Is est main tgt element Pers or Mat ?
MAT
34
Set A(22) & (23) = Frac. Surv. of Tank & APC's from all rds from this unit vs this tgt est environment and posture.
35
If no tanks and/or APC's in tgt then set A(22) and/or A(23) = 1.
RETURN

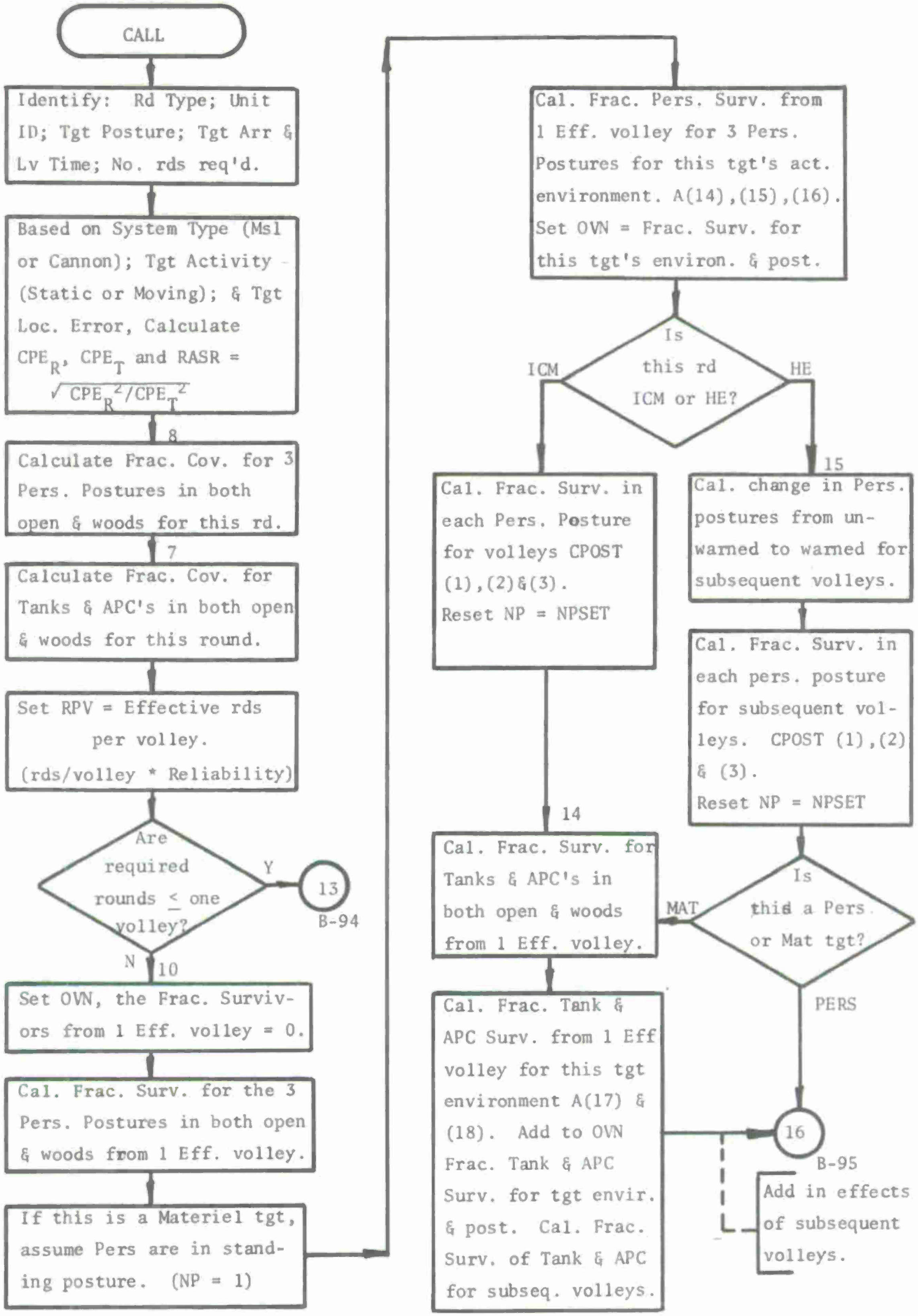
B. When called to determine actual effects

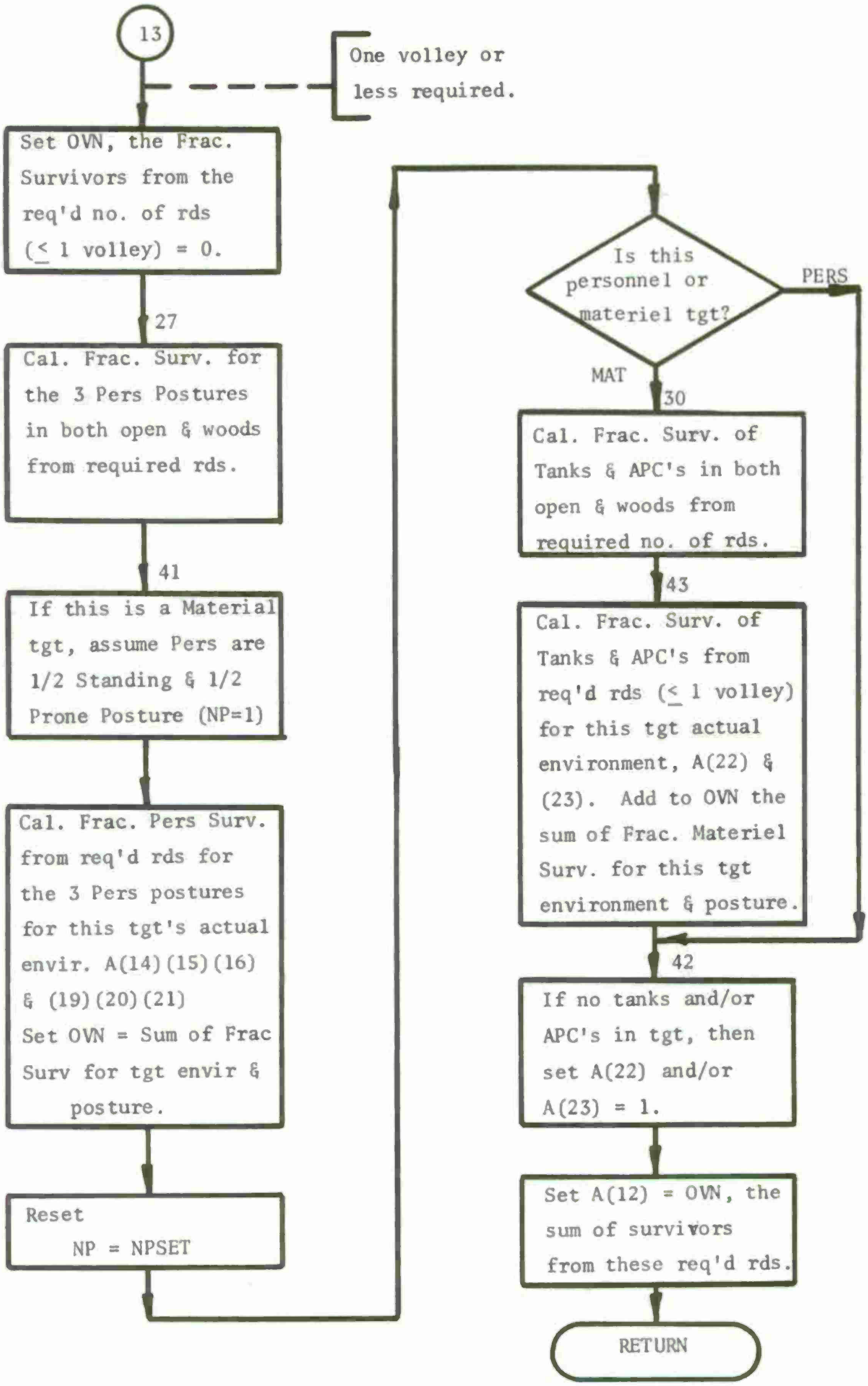
In this situation EFFECT is used to calculate actual effects of a pre-calculated number of rounds, given that a fire unit has been selected to fire those rounds on a target. This is done only when actual target data for posture, environment, and target radius differs from the estimated values. Otherwise, data generated in AMASS is still valid.

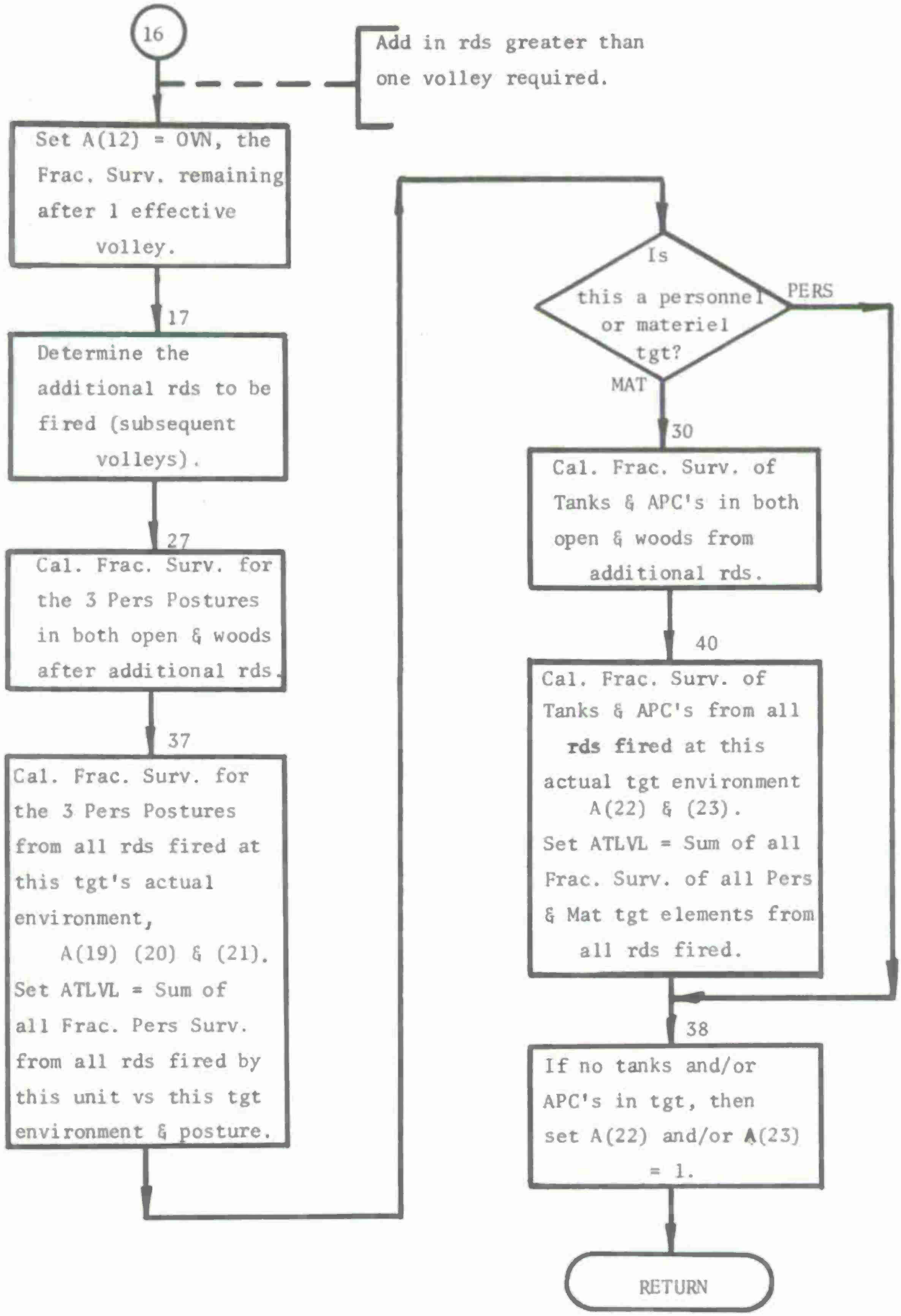
After defining the round, unit, actual posture, arrival and departure time, number of rounds required and delivery errors, the fractional casualties against target elements (personnel and materiel) are calculated (COV). Then the fractional survivors from one volley are calculated for the specific target, and, if necessary, these values are also calculated for the effects of more than one volley. As before, changes in personnel posture are made (unwarned to warned) for HE ammunition used in non-observed missions. Program control is then passed to the calling subroutine.

Subroutine EFFECT (cont)

(When called to determine effects vs actual tgt data.)




13
One volley or less required.
Set OVN, the Frac. Survivors from the req'd no. of rds (≤ 1 volley) = 0.
27
Cal. Frac. Surv. for the 3 Pers Postures in both open & woods from required rds.
41
If this is a Material tgt, assume Pers are 1/2 Standing & 1/2 Prone Posture (NP=1)
Cal. Frac. Pers Surv. from req'd rds for the 3 Pers postures for this tgt's actual envir. A(14)(15)(16) & (19)(20)(21) Set OVN = Sum of Frac Surv for tgt envir & posture.
Reset NP = NPSET
Is this personnel or materiel tgt?
PERS
MAT
30
Cal. Frac. Surv. of Tanks & APC's in both open & woods from required no. of rds.
43
Cal. Frac. Surv. of Tanks & APC's from req'd rds (≤ 1 volley) for this tgt actual environment, A(22) & (23). Add to OVN the sum of Frac. Materiel Surv. for this tgt environment & posture.
42
If no tanks and/or APC's in tgt, then set A(22) and/or A(23) = 1.
Set A(12) = OVN, the sum of survivors from these req'd rds.
RETURN


16
Add in rds greater than one volley required.
Set A(12) = OVN, the Frac. Surv. remaining after 1 effective volley.
17
Determine the additional rds to be fired (subsequent volleys).
27
Cal. Frac. Surv. for the 3 Pers Postures in both open & woods after additional rds.
37
Cal. Frac. Surv. for the 3 Pers Postures from all rds fired at this tgt's actual environment, A(19) (20) & (21). Set ATLVL = Sum of all Frac. Pers Surv. from all rds fired by this unit vs this tgt environment & posture.
Is this a personnel or materiel tgt?
PERS
MAT
30
Cal. Frac. Surv. of Tanks & APC's in both open & woods from additional rds.
40
Cal. Frac. Surv. of Tanks & APC's from all rds fired at this actual tgt environment A(22) & (23). Set ATLVL = Sum of all Frac. Surv. of all Pers & Mat tgt elements from all rds fired.
38
If no tanks and/or APC's in tgt, then set A(22) and/or A(23) = 1.
RETURN

Program Element: Calculate Fractional Coverage

Symbolic Name: COV

Arguments in Call Statement: None

Subroutines which call COV: EFFECT

Subroutines called by COV: None

This subroutine is an approximation model which calculates a circle-on-circle coverage of one round (defined by a radius of effects) on a target (defined by a target radius), with known delivery errors. The expected coverage formula is

$$FC_1 = \frac{1}{\pi R_T^2} \int_0^{3.5\ CPE_T} A_T(R)\,\phi(R)\,dR$$

where

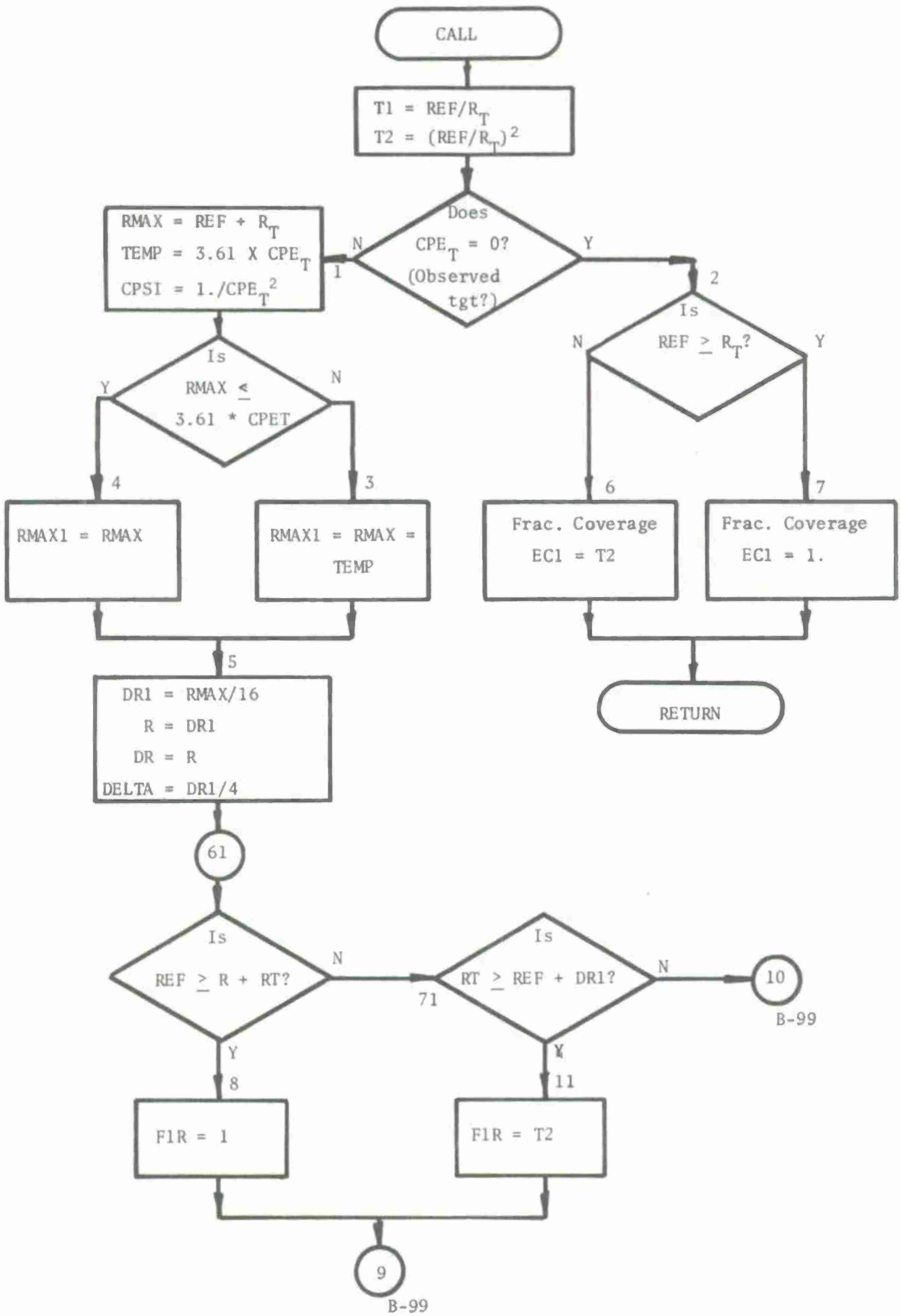
FC_1 = Expected area of target covered by one round

R = Distance between target center and center of effects circle

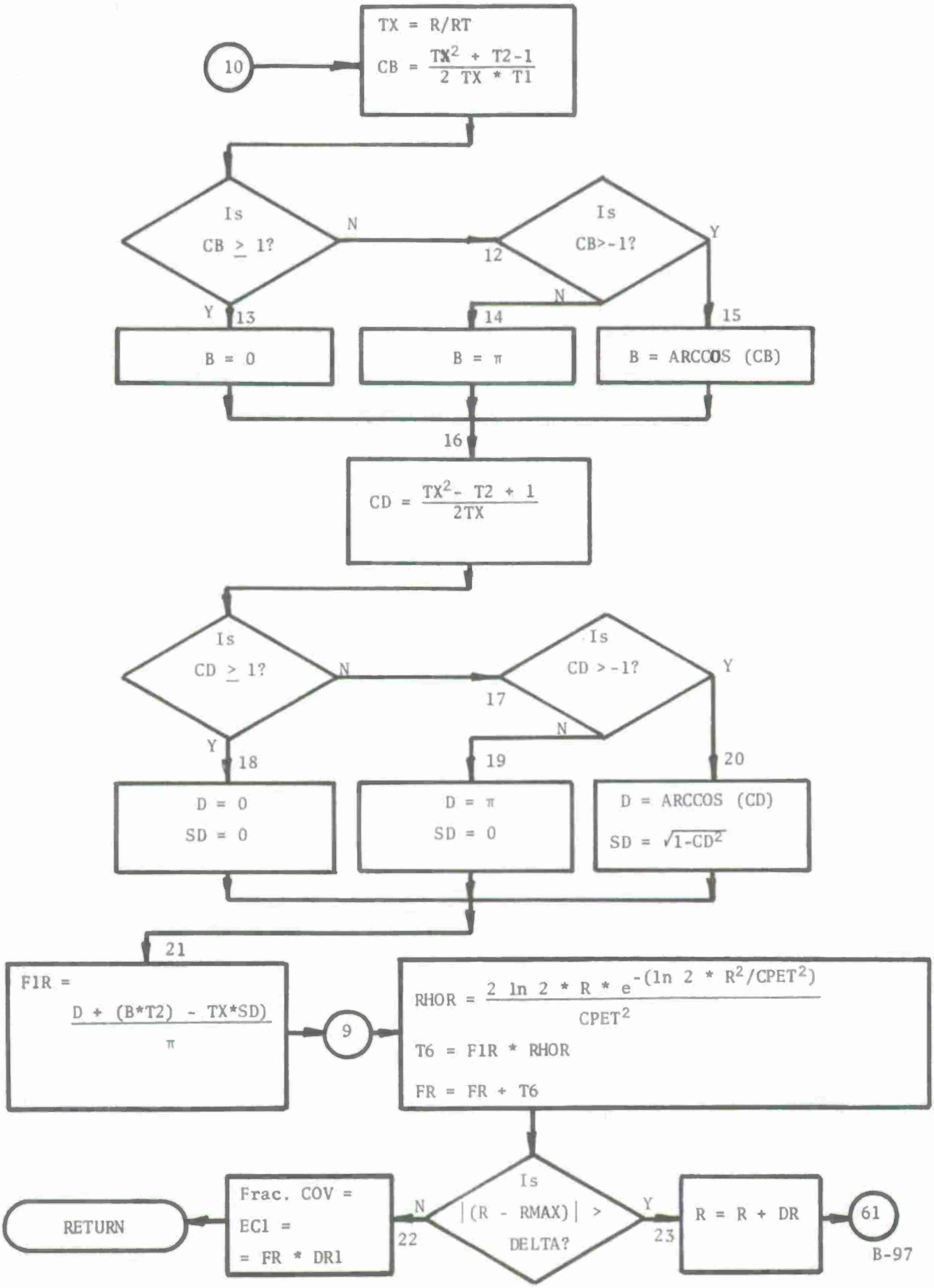
$A_T(R)$ = Area of target covered by the effects circle whose center is R units from the center of the target.

$\phi(R)$ = Gaussian delivery error density function evaluated at R. See Reference 6.

[6] Report, An Operational and Cost-Effectiveness Study of the LANCE Missile System (U), Vol III, USACDC/USAMC Study, April 1965. SECRET



The following page is blank.


10
TX = R/RT
CB = TX² + T2-1 / 2 TX * T1
Is CB ≥ 1?
N
Y
13
12
Is CB>-1?
Y
N
14
15
B = 0
B = π
B = ARCCOS (CB)
16
CD = TX²- T2 + 1 / 2TX
Is CD ≥ 1?
N
Y
18
17
Is CD > -1?
Y
N
19
20
D = 0
SD = 0
D = π
SD = 0
D = ARCCOS (CD)
SD = √1-CD²
21
F1R =
D + (B*T2) - TX*SD / π
9
RHOR = 2 ln 2 * R * e^-(ln 2 * R²/CPET²) / CPET²
T6 = F1R * RHOR
FR = FR + T6
Is |(R - RMAX)| > DELTA?
N
22
Y
23
Frac. COV =
EC1 =
= FR * DR1
RETURN
R = R + DR
61
B-97

Program Element: Print Data Outputs

Symbolic Name: OUTPUT

Arguments in Call Statement: None

Subroutines which call OUTPUT: Main Program

Subroutines called by OUTPUT: None

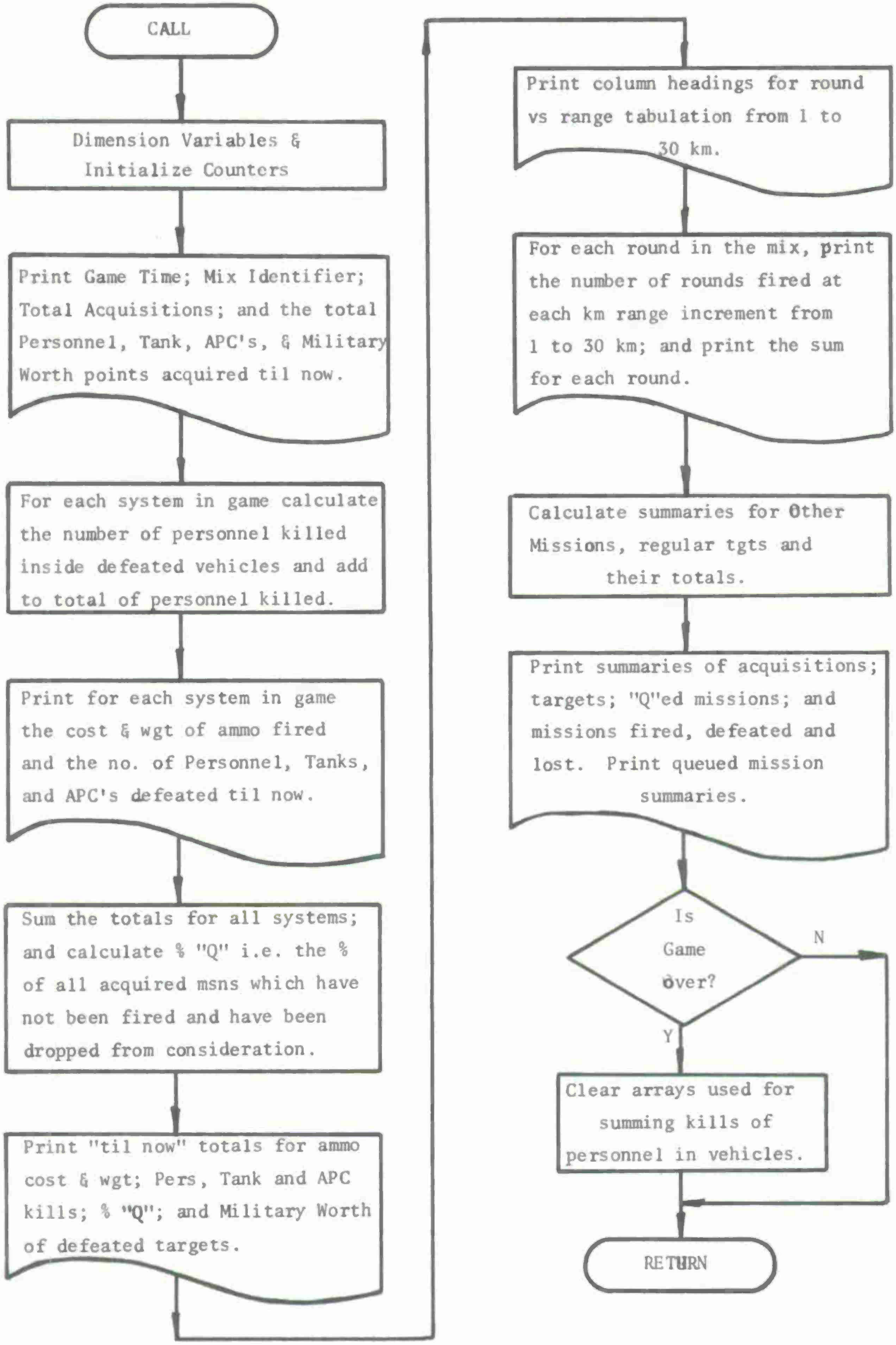
This subroutine provides the hourly data outputs throughout the course of the computer game.

The initial prints include game time, game "mix" identifier, and the cumulative number of acquisitions, personnel, tanks, APC's and Military Worth points acquired up to the printed game time.

The subroutine then credits 4 personnel per tank killed and 15 personnel per APC killed to the total of personnel killed. (Calculations made during the course of the computer game assume these personnel are inside vehicles, and are therefore, not included in personnel damage assessments.) The subroutine then prints for each system allowed in RDMIX, the total cost and weight of ammunition expended and the number of personnel, tanks and APC's defeated, along with a grand total of all systems. Additionally, the percent of "queued" missions (missions not fired and dropped from the target list) and the sum of Military Worth points scored are also printed.

Then a table is printed showing round expenditures by one-kilometer range increments (up to 30 kilometers) for each round allowed in RDMIX, along with a cumulative total for each round.

Subroutine OUTPUT



The following page is blank.

Next a table is printed showing number of acquisitions, targets, queued missions, queued missions plus missions still on the target list, missions fired, missions defeated, and missions fired but not defeated (lost), broken down by "other" missions, regular targets and a sum of both.

Finally, a queued mission breakdown is printed, along with target list, reacquired mission and combined target totals.

After clearing summation counters if the game is over, control is returned to the main program. See Sample Output in Appendix E.

Program Element: Special Missions

Symbolic Name: SPECIL

Arguments in Call Statement: (IT) - Identifies position on target list of the target being considered.

Subroutines which call SPECIL: Main Program

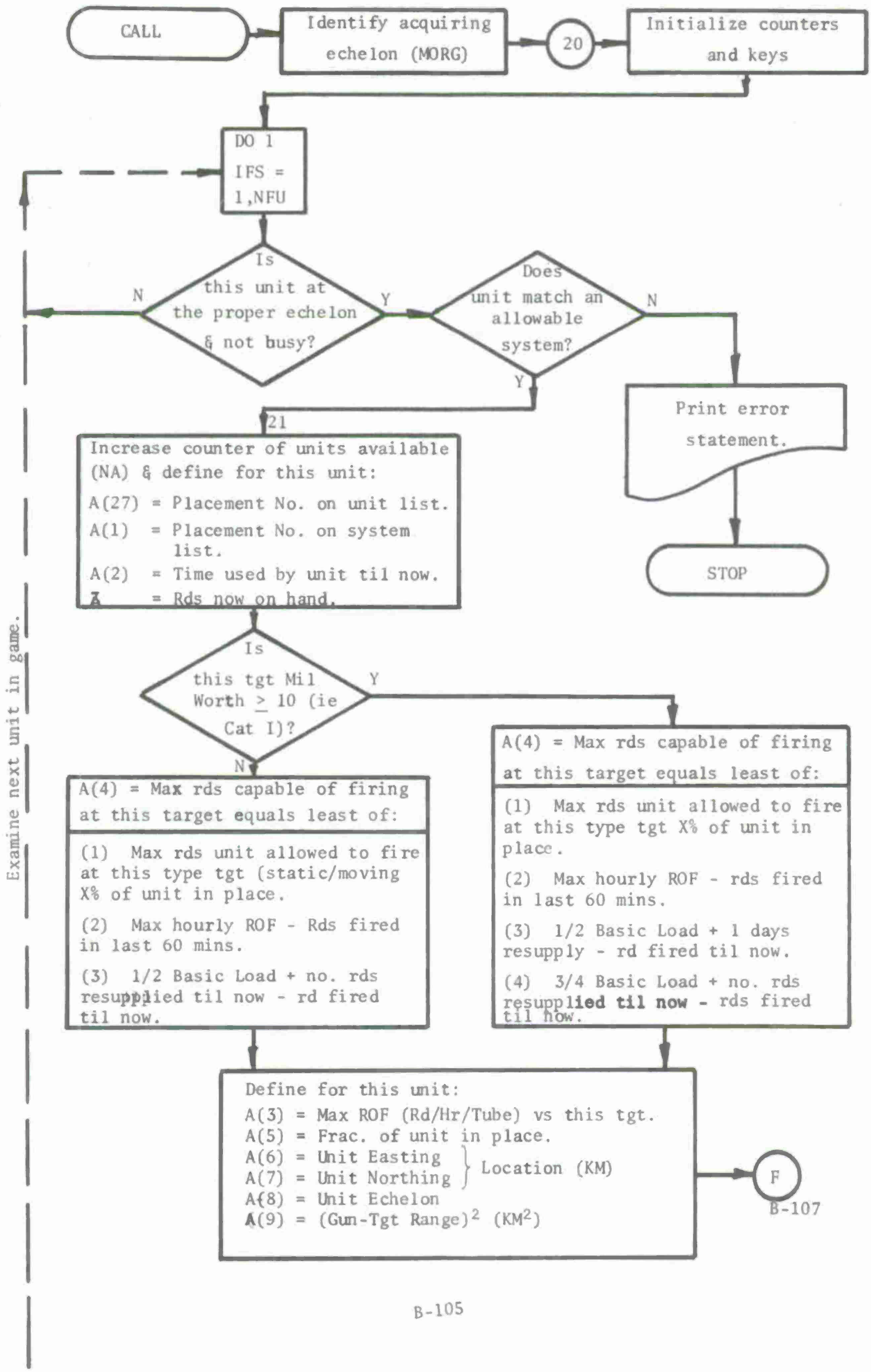
Subroutines called by SPECIL: None

This subroutine provides for attempts to fire the special or "other" type missions, i.e. Smoke, Illumination or H&I. After setting keys to indicate the appropriate acquiring echelon, this subroutine examines the list of fire units at that echelon to determine which are not busy and have enough rounds available to fire the pre-calculated number of rounds required. (The number of rounds required for each type system are read from tape (RTAPE) when the target list was input.)

Since smoke and illumination rounds are not included in the list of rounds input; and since H&I missions usually fire HE type ammunition, this subroutine searches through the allowed rounds for each available fire unit to find the minimum cost HE round. Rounds fired for these type missions will then be credited to the minimum cost HE round for the fire unit.

The subroutine then determines which single available unit can accomplish the mission with the least total cost of ammunition and proceeds to credit the rounds fired to that unit, determines the time used by the unit and increases the military worth point counters.

Subroutine SPECIL

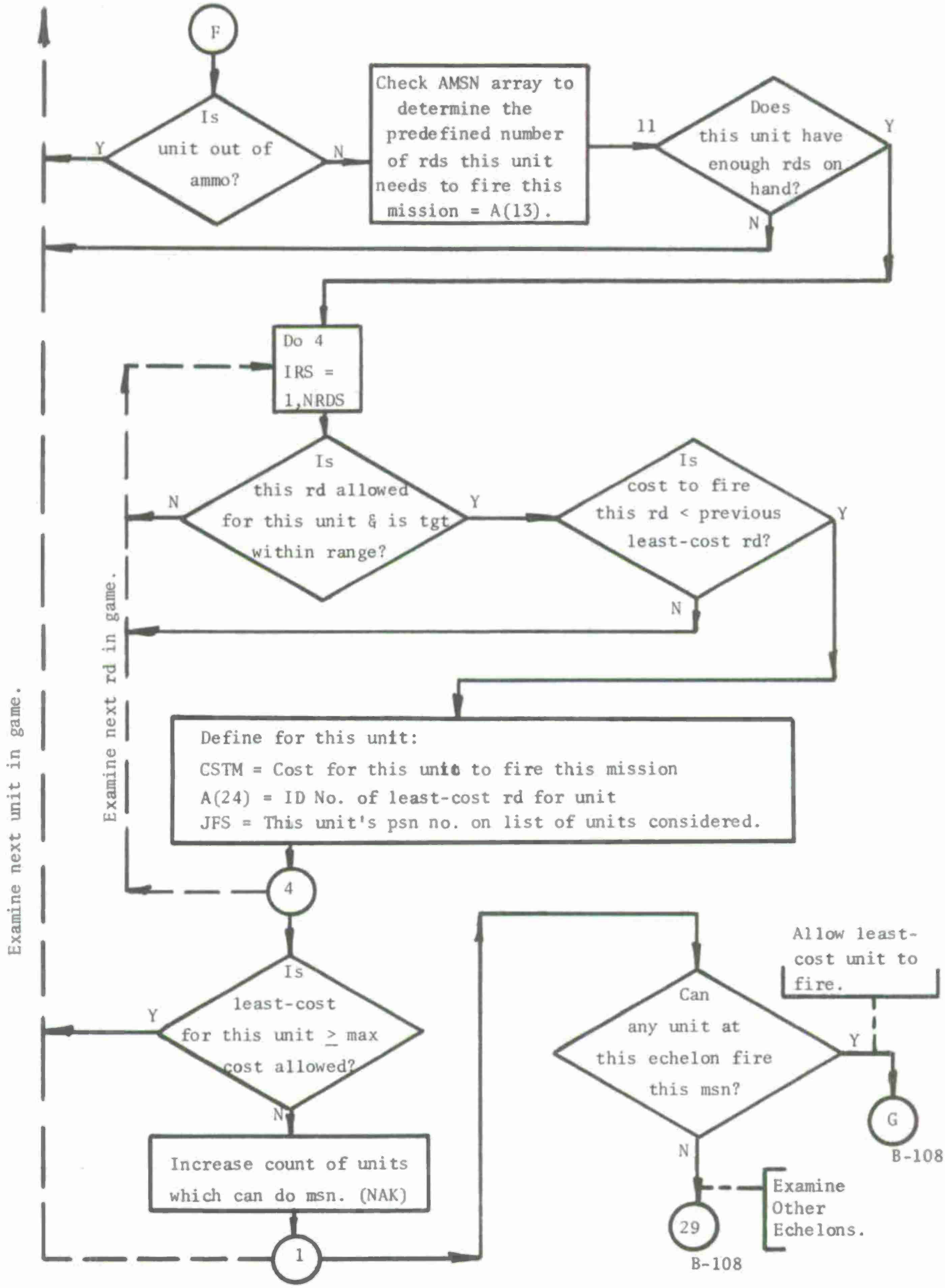


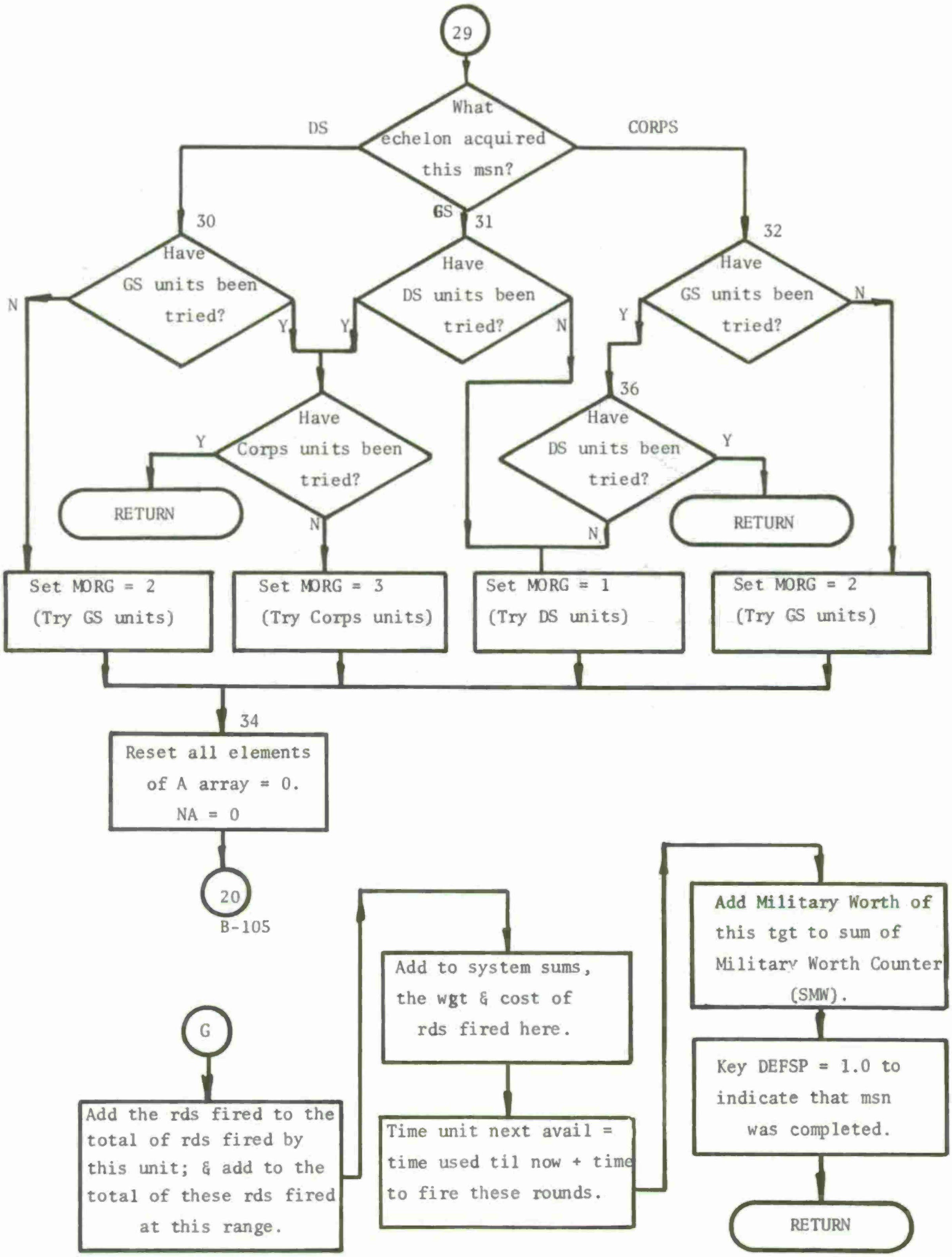
After setting a key to indicate that the mission was accomplished (=defeated) control returns to the main program.

If the acquiring echelon does not have a unit which can accomplish the mission other echelons are then searched in the following order:

Acquiring Echelon	Next Attempt	Final Attempt
DS	GS	CORPS
GS	DS	CORPS
CORPS	GS	DS

If no unit can be found to accomplish the mission from among all echelons, control returns to the main program where this mission will be removed from the target list and counted towards the sum of queued missions.


F
Is unit out of ammo?
Y
N
Check AMSN array to determine the predefined number of rds this unit needs to fire this mission = A(13).
11
Does this unit have enough rds on hand?
Y
N
Do 4 IRS = 1,NRDS
Is this rd allowed for this unit & is tgt within range?
N
Y
Is cost to fire this rd < previous least-cost rd?
Y
N
Examine next rd in game.
Examine next unit in game.
Define for this unit:
CSTM = Cost for this unit to fire this mission
A(24) = ID No. of least-cost rd for unit
JFS = This unit's psn no. on list of units considered.
4
Is least-cost for this unit ≥ max cost allowed?
Y
N
Increase count of units which can do msn. (NAK)
1
Can any unit at this echelon fire this msn?
Y
N
Allow least-cost unit to fire.
G
B-108
Examine Other Echelons.
29
B-108



The following page is blank.

APPENDIX C

PROGRAM VARIABLE LISTING

The following page is blank.

COMMON TERMS

Name	Size	Input Units	Meaning	Remarks
A	27 x 100	Real	Array of data for each unit vs a given target.	See "A" Array Data
AE	50 x 10	$1/m^2$	1/Rds Area of Effects (For HE munitions, this is set = 0)	Sub. HEINP & ACMINP
AMSN	385 x 8	Real	List of target ID and number rounds each system needs to fire a given special (OTHER) mission.	Sub. RTAPE & SPECIL See Table 3.6
AMWS	-	Real	Accumulates total Military Worth points scored on attacked targets.	
AOP	50 x 10	Real	P_K at the 10 range increments for each round vs APC's in open.	See ACMINP & HEINP
APC	-	Real	Accumulates fractional survivors of APC as fire units are added to attack.	Used only in one-volley attack method.
ATLVL	-	Real	Attack level - the maximum % of survivors allowed after firing.	Initially set at .5. If exceeded, cannot fire.
AWP	50 x 10	Real	P_K at the 10 range increments for each round vs APC's in woods.	See ACMINP & HEINP
BLD	10	Rd/Unit	Fire Unit Basic Ammo Load	Sub. RDSYS
CCOV	10	Real	Fractional coverages for given condition for 5 posture elements in open & woods.	See EFFECT
COP	50 x 10	Real	P_K at 10 range increments for each round vs crouching personnel in open.	See ACMINP & HEINP
CPER	-	Meters	Rd-Rd CPE at specific Gun-Target Range	See INTERP
CPET	-	Meters	Total CPE at specific G-T Range	See INTERP
CPK	10	Real	P_K at specific Gun-Target Range for the 5 posture elements in open & woods.	See INTERP
CPOST	5	Real	The fractional survivors in the five posture elements after the 1st volley.	See EFFECT
CPR	50 x 10	Meters	Rd-Rd CPE at each of 10 Range Increments.	Sub. RDRND
CPS	50 x 10	Meters	Total CPE at each of 10 Range Increments.	Sub. RDRND
CRE	10	Meters	Rd. Radius of Effects at specific Gun-Target Range for the 5 postures in open & woods.	See INTERP
CRIT	-	Real	Accumulates total cost or weight of rounds as fire units are added to attack.	Used only in one-volley attack method.

Name	Size	Input Units	Meaning	Remarks
CRITF	-	Real	Intermediate sum of cost or weight of rounds as fire units are added to attack.	Used only in one-volley attack method.
CRT	50	Real	Critical Factor: Set = Weight or cost for each round, per CRTERA.	Sub. RDMIX
CRTERA	-	Int.	Keys ammo cost or weight as critical.	Sub. RDMIX
CST	50	Kilo$/Rd	Rd. Cost	Sub. RDRND
CSTI	50	Rd/Kilo$	1/Rd. Cost	Sub. RDRND
CSURV	10	Real	Fractional survivors from one volley for 5 posture elements in open & woods.	See EFFECT
CTI	-	Real	Absolute maximum value for round cost or weight.	Initialized as 10^6 in Main Program.
CWP	50 x 10	Real	P_K at 10 range increments for each round vs crouching personnel in woods.	See ACMINP & HEINP
CXID	16	Alpha-Numeric	Force Identifier Code	Sub. PRELIM
DEFSP	-	Real	Key to indicate that "OTHER MSN" has been undertaken.	See SPECIL
DELT	-	Decimal Hours	Time Increment = .25 Game cycles thru target list each DELT.	Sub. PRELIM
DEP	50	Real	% of round recoverable misfires.	Sub. RDRND
DEPI	50	Real	1/% of round recoverable misfires.	Sub. RDRND
DNMX	10	Rd/Tube/Msn	Maximum rounds allowed vs moving target.	Sub. RDSYS (Converted to Rd/FU/Msn)
DROF	10	Rd/Min/Tube	System maximum rate of fire vs moving target.	Sub. RDSYS (Converted to Rd/Hr/Tube)
DSFLAG	-	Real	Key to indicate that DS units have been tried.	
DVFLAG	-	Real	Key to indicate that GS units have been tried.	
EC1	-	Real	Fractional coverage for a given posture & environment.	See COV
EXCES1	-	Real	Key to indicate that one-volley method needs excessive weight.	Used only in one-volley attack method.

Name	Size	Input Units	Meaning	Remarks
EXCES2	-	Real	Key to indicate that multi-volley method needs excessive weight.	Used only in multi-volley attack method.
F	-	Real	Fractional part of last volley needed to reach attack level.	Used only in one-volley attack method.
FACT	-	Real	Game Intensity Key	Sub. PRELIM
FH	-	Real	Accumulates fractional survivors of personnel in foxholes as fire units are added to attack.	Used only in one-volley attack method.
FM	-	Real	Fractional part of last fire unit's available rounds needed to reach attack level.	Used only in multi-volley attack method.
FRWM	10	Real	Fraction of fire units in place during moves.	Sub. RDSYS
FSID	100	Real	Identifies each fire unit by system ID.	Sub. RDFU
FT	6 x 100	Rds.	Lists total ammo fired for each fire unit for current & last 5 quarter hours.	
HBLD	10	Rds.	1/2 Fire Unit Basic Ammo Load	See RDSYS
HNMX	10	Rd/Tube/Hr	Maximum rounds allowed to fire in 1 hour.	Sub. RDSYS (Converted to Rd/FU/Hr)
IFM	-	Int.	Identifies last fire unit to fire all available rounds, as fire units are added.	Used only in multi-volley attack method.
IFONE	-	Int.	Identifies last fire unit which fires a full volley, as fire units are added.	Used only in one-volley attack method.
ITC	-	Msns.	No. of "OTHER" Msns. acquired.	
KFIG	100	Int.	Keys those fire units, by echelon, which are allowed in the game mix.	Sub. RDMIX
KOUNT	-	Tgts.	No. of targets in target (TN) array.	
KRIG	50	Int.	Keys the rounds allowed in the mix.	Sub. RDMIX
KSIG	10	Int.	Keys the systems allowed in the mix.	Sub. RDMIX
LOSS	-	Tgts.	No. of targets on Partially Defeated List (TLOST).	
MATCH	-	Int.	A given target position in TN Array.	Used to place targets in priority position.

COMMON TERMS (CONT)

Name	Size	Input Units	Meaning	Remarks
MFEAT	-	Tgts.	No. of targets on Defeated Target List (TDFT).	
MIXID	16	Alpha Numeric	System-Rd-Fire Unit Game Mix Identifier.	Sub. RDMIX
MORG	-	Int.	Key to which echelon is being massed in AMASS.	1 - DS, 2 - GS, 3 - CORPS
NA	-	Int.	Counts available Fire Units during massing process.	
NA1	-	Int.	Indicates position in "A" array of best fire unit.	Used only in multi-volley attack method.
NACQ	-	Acquisitions	Sum of target acquisitions.	
NB	-	Int.	Indicates position in "A" array of 1st fire unit in a given echelon.	
NBA1	-	Int.	Indicates position of 1st DS fire unit in "A" array.	
NBA2	-	Int.	Indicates position of 1st GS fire unit in "A" array.	
NBA3	-	Int.	Indicates position of 1st CORPS fire unit in "A" array.	
NE	-	Int.	Indicates position in "A" array of last fire unit in a given echelon.	
NEA1	-	Int.	Indicates position of last DS fire unit in "A" array.	
NEA2	-	Int.	Indicates position of last GS fire unit in "A" array.	
NEA3	-	Int.	Indicates position of last CORPS fire unit in "A" array.	
NFM	-	Fire Msns	Sum of all missions fired.	
NFMD	-	Msns.	Sum of all targets/msns defeated.	
NFU	-	Fire Units	Number of Fire Units	Sub. RDFU
NOM	-	Msns.	Sum of "OTHER MSNS" acquired.	
NOMF	-	Msns.	Sum of "OTHER MSNS" fired.	
NP	-	Int.	Posture Mix Identifier for main target element.	Sub. RDFU
NQ	-	Msns.	Total of all msns not undertaken.	
NQD	-	Tgts.	Sum of targets which depart prior to firing.	
NQLP	-	Tgts.	Sum of targets not fired due to low priority.	

Name	Size	Input Units	Meaning	Remarks
NQOM	-	Msns.	Sum of "OTHER MSNS" tried but can't do.	
NRDS	-	Rds.	Number of rounds in force.	Sub. RDRND
NRO	12	Rds.	The number of round types allowed to fire vs a given target posture in open.	Sub. RDFU
NRFD	-	Reacq's	Sum of reacqs of defeated targets.	
NRW	12	Rd.	The number of round types allowed to fire vs a given target posture in woods.	Sub. RDFU
NRW2	-	Tgts.	Sum of targets which are combined.	Targets within 200m of each other.
NSITE	100	Sites	Number of sites for each fire unit.	Sub. RDFU
NSYS	-	Systems	Number of Wpn System Types	Sub. RDSYS
NTGT	-	Tgts.	Sum of targets acquired.	Counts only 1st acquisition.
ONCRT	-	Real	Total cost or weight (whichever is critical) to reach attack level.	Used only in one-volley attack method.
ORVP	12 x 12	Real	Rd. ID Number of the rounds allowed to fire vs a given target posture in open.	Sub. RDFU
PERO	-	Real	% of target element in open.	
PERW	-	Real	% of target element in woods.	
PII	-	Real	$1/\pi$	Constant term.
PO	-	Real	Accumulates fractional survivors of prone personnel as fire units are added to attack.	Used only in one-volley attack method.
POP	50 x 10	Real	P_K at the 10 range increments for each round vs prone personnel in open.	See ACMINP & HEINP
POST	12 x 11	Real	Table of 12 possible posture mixes.	Sub. PRELIM
PWP	50 x 10	Real	P_K at the 10 range increments for each round vs prone personnel in woods.	See ACMINP & HEINP
QBLD	10	Rds.	1/4 Fire Unit Basic Ammo Load.	See RDSYS
QN	-	Rds.	Number rounds required by "best" available fire unit to reach attack level.	Used only in multi-volley attack method.
QUV	100	Real	Ratio of rounds needed by a fire unit compared to rounds needed by "best" fire unit.	Used only in multi-volley attack method.

Name	Size	Input Units	Meaning	Remarks
R2MX	50	KM^2	Rd. Max $Range^2$	Sub. RDRND
RDCNT	50 x 30	Rds.	Sums for each round type the no. of rounds fired at 1, 2, thru 30 or more KM.	
RE	50 x 10	Meters	Rd. radius of effects (For HE munitions, P_K values are set).	Sub. HEINP & ACMINP
REF	-	Meters	Set equal to round radius of effects for given condition in calculating coverage.	See EFFECT
REFIRE	-	Real	A key to indicate reattack at a lower attack level is to be tried.	
REL	50	Real	Rd. In-flight reliability	Sub. RDRND
RELI	50	Real	1/Rd. In-flight reliability	Sub. RDRND
RMX	50	KM	Rd. Max. Range	Sub. RDRND
RNDID	50	Real	Rd. ID Number	Sub. RDRND
RNG	50 x 10	KM	10 Range Increments per rd.	Sub. RDRND (Converted to KM^2)
RPV	-	Rd/volley	No. of rds/volley for a given fire unit.	
RSPY	10	Rd/FU/Hr	Fire Unit Ammo Resupply Rate	Sub. RDSYS
RT	-	Meters	Target Radius	
RTP	50	Real	Key to round type (ICM or HE)	Sub. RDRND
S	10 x 5	Real	Sum of ammo cost & weight and number of personnel, tank & APC defeated.	Sums for each system.
SAPC	-	APC's	Sum of number of APC acquired.	
SAVE1	-	Real	Temporary storage of fractional personnel survivors from previous attacks.	
SAVE2	-	Real	Temporary storage of fractional tank survivors from previous attacks.	
SAVE3	-	Real	Temporary storage of fractional APC survivors from previous attacks.	
SMCRT	-	Real	Total cost or weight (whichever is critical) needed to reach attack level.	Used only in multi-volley attack method.
SMW	-	Real	Sum of Mil Worth of damaged tgts.	
SNMX	10	Rd/Tube/Msn	Max. rds. allowed vs static tgt.	Sub. RDSYS (Converted to Rd/FU/Msn)

Name	Size	Input Units	Meaning	Remarks
SOP	50 x 10	Real	P_K at the 10 range increments for each round vs standing personnel in open.	See ACMINP & HEINP
SPERS	-	Pers.	Sum of number of personnel acquired.	
SQN	-	Rds.	Accumulates total number of effective rounds needed as fire units are added.	Used only in multi-volley attack method.
SROF	10	Rd/Min/Tube	System maximum rate of fire vs static tgt.	Sub. RDSYS (Converted to Rd/Hr/Tube)
ST	-	Real	Accumulates fractional survivors of standing personnel as fire units are added to attack.	Used only in one-volley attack method.
STANK	-	Real	Sum of number of tanks acquired.	
STYP	10	Real	Key to system type (Cannon or Missile)	Sub. RDSYS
SURV	-	Real	Accumulates total fractional survivors of all target elements as fire units are added.	Used only in one-volley attack method.
SURVP	-	Real	Intermediate sum of fractional survivors of all target elements.	Used only in one-volley attack method.
SWP	50 x 10	Real	P_K at the 10 range increments for each round vs standing personnel in woods.	See ACMINP & HEINP.
SYSID	10	Real	Wpn system ID Numbers	Sub. RDSYS
SYSRD	10 x 10	Real	List of Rd ID's allowed for each system.	Sub. RDFU
T	-	Decimal Hours	Game Clock Time	
TA	8 x 100	Hr. min	Arrival Time for fire units at each site.	Sub. RDFU (Converted to Decimal Hours)
TBM	10	Minutes	Time Between Missions - Time for fire units to prepare for and fire one volley.	Sub. RDSYS (Converted to Hours)
TD	8 x 100	Hr. min	Departure Time for fire units from each site.	Sub. RDFU (Converted to Decimal Hours)
TDFT	1000	Real	List of defeated tgt. ID numbers.	
TK	-	Real	Accumulates fractional survivors of tanks as fire units are added to attack.	Used only in one-volley attack method.

Name	Size	Input Units	Meaning	Remarks
TLOST	5 x 1000	Real	Array of partially defeated tgts.	Includes Tgt. ID, % Pers, Tank & APC Survivors.
TMX	-	Decimal Hours	Game End Time	Sub. PRELIM
TN	30 x 300	Real	A priority listing of targets to be considered for attack.	See Table 3.6
TN28T	-	Pers.	Temporary storage of number of personnel survivors in target at start of attack by a given fire unit.	
TN29T	-	Tanks	Temporary storage of number of tanks in target at start of attack by a given fire unit.	
TN30T	-	APC's	Temporary storage of number of APC's in target at start of attack by a given fire unit.	
TNI	33	Real	Temporary storage of target data.	See RTAPE
TOEP	-	Pers.	No. of personnel in target outside of tank & APC.	
TOP	50 x 10	Real	P_K at the 10 range increments for each round vs tanks in open.	See ACMINP & HEINP
TPFU	10	Tubes/Unit	No. of tubes (launchers) per fire unit.	Sub. RDSYS
TR	-	Real	Key to indicate target was attacked.	
TUBFU	100	Decimal Hours	Clocks of time used by each fire unit.	
TWP	50 x 10	Real	P_K at the 10 range increments for each round vs tanks in woods.	See ACMINP & HEINP
TZRO	-	Decimal Hours	Game Start Time	Sub. PRELIM
W	-	Real	ln 2.	Constant term.
W1	-	Real	2 ln 2.	Constant term.
WAIT	-	Metric Tons	Total weight of rounds needed for one-volley attack.	Used only in one-volley attack method.
WAIT2	-	Metric Tons	Total weight of rounds needed for multi-volley attack.	Used only in multi-volley attack method.
WAIT3	-	Metric Tons	Total weight of rounds needed for a single DS fire unit to attack.	Used only in DIRSUP.

COMMON TERMS (CONT)

Name	Size	Input Units	Meaning	Remarks
WARN	50	Seconds	Rd. Signiture (Warning)	Sub. RDRND
WGT	50	Metric Tons/Rd	Crated Weight per round.	Sub. RDRND
WGTI	50	Rd/Metric Ton	1/Crated Weight per round.	Sub. RDRND
WKS	-	Real	-1./Π x ln. 7	Constant term
WRVF	12 x 12	Real	Rd. ID Number of the rounds allowed to fire vs a given target posture in woods.	Sub. RDFU
XS	8 x 100	KM	X-coord (Northing) of fire unit at each site.	Sub. RDFU
XVN	-	Rds.	No. rounds a given fire unit needs to reach attack level.	Also used as a key when calling EFFECT from AMASS.
YS	8 x 100	KM	Y-coord (Easting) of fire unit at each site.	Sub. RDFU

The following page is blank.

MAIN PROGRAM VARIABLES

Name	Units	Meaning
DN	Real	Counter for placing target in game FR times.
FR	Real	Target frequency at this game's intensity.
I	-	Subscript to identify elements in ammo counter array FT.
IT	-	Subscript to identify targets in target array TN.
J	-	Subscript to identify fire units, "A" array elements and TLOST elements.
JK	-	Subscript to identify targets on list of defeated targets TDFT.
JL	-	Subscript to identify targets on array of partially defeated targets TLOST.
K	-	Subscript to identify fire units in "A" array & elements in TLOST array.
KJ	-	Subscript to identify elements in AMSN array.
KT	-	Number of targets on TN array at start of 15 min game cycle.
L	-	Key for the echelon which acquired a given target.
TOUT	-	Counter to control OUTPUT frequency.

The following page is blank.

SUBROUTINE VARIABLES

Name	Units	Meaning
		PRELIM
I	Int.	Subscript used to identify each of 12 postures in POST array.
J	Int.	Subscript used to identify data points for each posture.
PI	Real	π = 3.14159 (A constant term)
		RDSYS
I	Int.	Subscript used to identify inputs for each system.
		RDRND
I	Int.	Subscript used to identify data inputs for each round.
J	Int.	Subscript used to identify inputs at each of 10 ranges for each round.
		ACMINP
AL(10)	$Meters^2$	List of Lethal Areas at 10 range increments.
EN	Submissiles	Number of submissiles in a given round.
I	Int.	Subscript to identify a given round.
J	Int.	Subscript to identify the 10 RE's and P_K's for given round.
K	Int.	Subscript to identify 10 Lethal Area inputs for given round.
PRO	Real	Intermediate value in calculating P_K's vs open targets.
PRW	Real	Intermediate value in calculating P_K's vs wooded targets.
REZ	Meters	Y-intercept of Radius of Effects vs Range Plot for given round.
SRE	Real	Slope of Radius of Effects vs Range Plot for given round.
SRO	Real	Submissile reliability in open environment for given round.
SRW	Real	Submissile reliability in wooded environment for given round.
		HEINP
AL(100)	Meters	List of 100 Lethal Areas at 10 ranges for 10 postures.
I	Int.	Subscript to identify a given round.
J	Int.	Subscript to identify 100 Lethal Areas.
		RDFU
I	Int.	Subscript to identify fire units, systems, and postures.
J	Int.	Subscript to identify fire unit sites and rounds allowed for each system & posture.
LK1	Rounds	Number of rounds allowed vs open targets with a given posture.
LK2	Rounds	Number of rounds allowed vs wooded targets with a given posture.
LTA	Int.	Used to convert fire unit arrive times from hr-min to decimal hours.
LTD	Int.	Used to convert fire unit departure times from hr-min to decimal hours.
NS	Sites	Number of firing sites occupied by a given fire unit.
XTA	Real	Used to convert fire unit arrive times from hr-min to decimal hours.
XTD	Real	Used to convert fire unit departure times from hr-min to decimal hours.

Name	Units	Meaning
		RDMIX
I	Int.	Subscript to identify systems, rounds, fire units and ammo critical parameter.
ICRT	Int.	Defines critical ammo parameter (either cost (1) or weight (2).)
		TZERO
I	Int.	Subscripts to identify systems, rounds, targets and associated
J	Int.	data during initialization of arrays & lists.
		RTAPE
J	Int.	Subscript to identify various systems for "OTHER" msns.
KFACT	Int.	Game Intensity Key (1=Low; 2=Mid, 3=Base, 4=High)
		REMOVE
I	Int.	Subscript to identify 30 target parameters per target on target list.
IKZ	Targets	Number of targets remaining on target list after a given target is removed.
J	Int.	Subscript to identify a target's position on target list.
K	Int.	Position on target list (TN) of target to be removed.
		COMPAR
D	KM^2	Distance squared between 2 acquired locations of same target.
I	Int.	Subscript to identify target on defeated & partially defeated lists and data pointson target list.
J	Int.	Subscript to identify target on target & defeated target lists and data points on TLOST array.
K	Int.	Subscript to identify target on TLOST array; specifies target position on target list.
KI	Int.	Subscript to identify target position on target list.
M	Int.	Defines position of a given target on target list.
MAX	Int.	Defines maximum number of targets carried on target list.
		DIRSUP
A9	KM	Gun-Target range for the fire unit firing the DS mission.
ASUM	Real	Sum of fractional personnel survivors (standing, crouching and prone).
B	Decimal Hours	Time used by cannon fire unit to fire more than 1 volley.
CONSTR	Kilometric Tons	Maximum total weight of ammo allowed vs target.
IDS	Int.	Subscript to identify a DS fire unit on "A" array (array of available units).
IDS1	Int.	Subscript to identify closest DS fire unit to target.
IR	Int.	Subscript to identify best round of closest DS fire unit to target.
IRNG	KM	Subscript to identify gun-target range rounded off to nearest KM.
IS	Int.	Subscript to identify the system of the closest fire unit to target.

Name	Units	Meaning
		DIRSUP (Cont'd)
IT	Int.	Subscript to identify which target on target list is being attacked.
IXZ	Int.	Subscript to identify which fire unit of all those input is being used.
RANGE	KM^2	$(\text{Gun-target range})^2$ for the fire unit firing mission.
RNGINT	KM	Used to round off the gun-target range to nearest KM.
TOE	Personnel	Number of personnel (if any) in tanks & APC's.
		INTERP
D1	KM^2	Proportional part of interval between 2 entries in RNG array.
D2	KM^2	Interval between 2 entries in RNG array.
IA	Int.	Subscript to identify a fire unit in "A" array.
IR	Int.	Subscript to identify a round in list of input rounds.
K	Int.	Subscript to identify smallest entry in RNG array which is $\geq (\text{Gun-Target Range})^2$
KS	Int.	Subscript to identify largest entry in RNG array which is $< (\text{Gun-Target Range})^2$
L	Int.	Subscript to identify entries in CRE & CPK lists.
RA	Real	Fractional Interval of $(\text{Gun-Target Range})^2$ between 2 entries in RNG array.
REI	Real	Round radius of effects (ICM) or P_K (HE) at Gun-Target Range.
RG	KM^2	$(\text{Gun-Target Range})^2$ for a given fire unit.
		AMASS
CT	Kilo $ or Metric Tons	Final (minimum) value (cost or weight) for the "best" round used with a given fire unit.
CT1V	Kilo $ or Metric Tons	Final (minimum) value (cost or weight) for 1 volley of best round used with fire unit.
CTMIN	Kilo $ or Metric Tons	Intermediate minimum value of critical factor (cost or weight) for fire unit.
I	Int.	Subscript used as a counter in setting up TEMPA list.
INS	Int.	Subscript to identify the site at which a fire unit is now located.
IR	Int.	Subscript to identify which of the input rounds is being considered.
IS	Int.	Subscript to identify the system for the fire unit being considered.
IT	Int.	Subscript to identify which target on the target list is being considered.
JF	Int.	Subscript to identify which fire unit of input fire units is being considered.
L	Int.	Subscript to check which rounds are allowed vs a given posture.
LK1	Int.	Number of round types allowed vs this target's posture (open environment).
LK2	Int.	Number of round types allowed vs this target's posture (wooded environment).
LSYS	Int.	Subscript to identify the system for the fire unit being considered.
M	Int.	Subscript to identify the rounds allowed for the fire unit being considered.
NS	Sites	The number of sites a given fire unit occupies during the game.

Name	Units	Meaning
		AMASS (Cont'd)
R	Rounds	Intermediate, then final number of rounds a given fire unit can fire vs target.
TEMPA (12)	Real	Intermediate storage for fire unit data while determining "best" round.
Z	Rounds	Intermediate value of rounds a given fire unit can fire vs target.
		DIVISN
A9	KM	Gun-Target Range for a given fire unit firing the mission.
ASUM	Real	Sum of fractional personnel survivors (standing, prone, and crouching)
B	Decimal Hours	Time used by a cannon fire unit to fire more than 1 volley.
IF3	Int.	Subscript to identify psn on "A" array of fire units firing multi-volley method.
IF4	Int.	Subscript to identify psn on "A" array of fire units firing one-volley method.
IK	Int.	Subscript to identify the system for the "best" fire unit firing multi-volley method.
IQ	Int.	Counter used to initialize QUV list.
IR	Int.	Subscript to identify "best" round of fire unit being considered.
IRNG	KM	Subscript to identify Gun-Target Range rounded off to nearest kilometer.
IS	Int.	Subscript to identify the system for the fire unit being considered.
IT	Int.	Subscript to identify which target on target list is being attacked.
IXZ	Int.	Subscript to identify which fire unit of all those input is being considered.
JF	Int.	Subscript to identify which fire unit on "A" array is being considered.
RNGINT	KM	Used to round-off the Gun-Target Range to nearest kilometer.
TEMP	Rounds	Defines the number of rounds a given fire unit fires in one volley.
TN28	Real	Fractional personnel survivors in target prior to attack (by multi-volley).
TN29	Real	Fractional tank survivors in target prior to attack (by multi-volley).
TN30	Real	Fractional APC survivors in target prior to attack (by multi-volley).
TOE	Personnel	Number of personnel (if any) in tanks and APC's.
		SHMUVL
A9	KM	Gun-Target Range for a given fire unit firing the mission.
ASUM	Real	Sum of fractional personnel survivors (standing, prone and crouching).
B	Decimal Hours	Time used by a cannon fire unit to fire more than one volley.
IF3	Int.	Subscript to identify psn on "A" array of fire unit being considered.
IR	Int.	Subscript to identify "best" round of fire unit being considered.
IRNG	KM	Subscript to identify Gun-Target Range rounded to nearest kilometer.

Name	Units	Meaning
		SHMUVL (Cont'd)
IS	Int.	Subscript to identify the system for the fire unit being considered.
IT	Int.	Subscript to identify which target on target list is being attacked.
IXZ	Int.	Subscript to identify which fire unit of all those input is being considered.
JF	Int.	Subscript to identify which fire unit on "A" array is being considered.
RNGINT	KM	Used to round-off the Gun-Target Range to the nearest kilometer.
TN28	Real	Fractional personnel survivors in target prior to multi-volley attack.
TN29	Real	Fractional tank survivors in target prior to multi-volley attack.
TN30	Real	Fractional APC survivors in target prior to multi-volley attack.
TOE	Personnel	Number of personnel (if any) in tanks & APC's.
		SHONVL
A9	KM	Gun-Target Range for a given fire unit firing the mission.
APCPO	Real	Fractional APC survivors from last fire unit firing (1 volley or less).
ASUM	Real	Sum of fractional personnel survivors (standing, prone and crouching).
IF2	Int.	Subscript to identify psn on "A" array of the fire unit being considered.
IR	Int.	Subscript to identify "best" round of fire unit being considered.
IRNG	KM	Subscript to identify Gun-Target Range rounded off to nearest kilometer.
IS	Int.	Subscript to identify the system for the fire unit being considered.
IT	Int.	Subscript to identify which target on target list is being attacked.
IXZ	Int.	Subscript to identify which fire unit of all those input is being considered.
RNGINT	KM	Used to round off the gun-target range to nearest kilometer.
TEMP	Rounds	Defines the number of rounds a given fire unit fires in 1 volley.
TKPO	Real	Fractional tank survivors from last fire unit firing (1 volley or less).
TOE	Personnel	Number of personnel (if any) in tanks & APC's.
		ONEVOL
A1(2,50)	Real	Temporary storage of fire unit data during sort of fire units into critical order.
B(27)	Real	Temporary storage of fire unit data during sort of fire units into critical order.
CONSTR	Metric Tons	Defines (according to target priority) maximum weight of ammo allowed vs target.

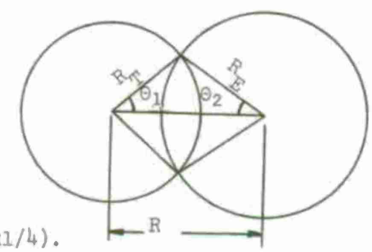
Name	Units	Meaning
		ONEVOL (Cont'd)
CRITMI	Real	Temporary storage for lowest value of critical factor (cost or weight).
IAC	Int.	Counter used in sorting thru "A" array.
IM	Int.	Counter used in sorting thru "A" array.
IRS	Int.	Subscript to identify which of input rounds is "best round" for fire unit considered.
IS	Int.	Subscript to identify the system for the fire unit being considered.
IT	Int.	Subscript to identify which target on target list is being attacked.
J	Int.	Counter used to sort thru "A" array.
JF	Int.	Subscript to identify psn of fire unit on "A" array.
K	Int.	Counter used in sorting fire units.
L	Int.	Counter used to transfer 27 fire unit data points from"B"list to "A" array.
M	Int.	Subscript to identify psn on "A" array of fire units being sorted.
MI	Int.	Subscript to identify psn on "A" array of "best" fire unit (with least critical factor).
NEA	Int.	Number of fire units on "A" array to be sorted.
ROUNDS	Rounds	Defines number of rounds a given fire unit can fire in 1 volley.
TEMP1	Metric Tons	Weight of rounds from a given fire unit (to be added to total weight).
		MULVOL
CONSTR	Metric Tons	Defines (according to target priority) maximum weight of ammo allowed vs target.
IF1	Int.	Subscript to identify psn on "A" array of fire unit being considered.
IRS	Int.	Subscript to identify "best" round for the fire unit being considered.
IT	Int.	Subscript to identify which target on target list is being attacked.
SQNP	Rounds	Intermediate value of number of effective rounds used, as fire units are added.
TEMP	Real	Critical amount (cost or weight) of ammo used by a given fire unit.
TEMP2	Metric Tons	Weight of rounds from a given fire unit (to be added to total weight).
		CORP
A9	KM	Gun-Target Range for a given fire unit firing the mission.
ALPHA	Real	Defines attack level for attack at Threshold "C".
ASUM	Real	Sum of fractional personnel survivors (standing, prone and crouching).
B	Decimal Hours	Time used by a cannon fire unit to fire more than 1 volley.
IF2	Int.	Subscript to identify psn on "A" array of CORPS fire units firing multi-volley method.
IF3	Int.	Subscript to identify psn on "A" array of given fire unit firing 1 volley method.
IF4	Int.	Subscript to identify psn on "A" array of GS or CORPS fire unit firing multi-volley method.
IQ	Int.	Counter used to initialize QUV list.
IR	Int.	Subscript to identify "best" round of a fire unit being considered.
IRNG	KM	Subscript to identify Gun-Target Range rounded off to nearest kilometer.

Name	Units	Meaning
		CORP (Cont'd)
IS	Int.	Subscript to identify the system for the fire unit being considered.
IT	Int.	Subscript to identify which target on target list is being attacked.
IXZ	Int.	Subscript to identify which fire unit of all those input is being considered.
JF	Int.	Subscript to identify which fire unit on "A" array is being considered.
K1	Int.	Key used to identify first fire unit at a given echelon to shoot 1 volley method.
K2	Int.	Key used to identify last fire unit at a given echelon to shoot 1 volley method.
KEY	Int.	Key to control program flow during use of 1 volley attack method.
NEA	Int.	Used to determine psn on "A" array of 1st CORPS fire unit.
OFLAG1	Real	Key used to indicate that no CORPS echelon fire units are in game.
OFLAG2	Real	Key used to indicate that no GS echelon fire units are in game.
RNGINT	KM	Used to round-off the Gun-Target Range to nearest kilometer.
TEMP	Rounds	Defines the number of rounds a given fire unit fires in 1 volley.
TN28	Real	Fractional personnel survivors in target prior to attack (by multi-volley).
TN29	Real	Fractional tank survivors in target prior to attack (by multi-volley).
TN30	Real	Fractional APC survivors in target prior to attack (by multi-volley).
TOE	Personnel	Number of personnel (if any) in tanks & APC's.
		EFFECT
A1	Real	Intermediate value of fractional survivors of personnel and material.
ART	Decimal Hours	Target acquisition (arrival) Time at its current location.
CN	Rounds	Used as a key (CN=0) if EFFECT is called from AMASS to determine number of rounds needed. CN is then calculated as number of rounds in addition to 1 volley which is needed to reach attack level.
CRSQ	$Meters^2$	$(Rd\text{-}Rd\ CPE)^2$
CTSQ	$Meters^2$	$(Total\ CPE)^2$
DOWN	Rounds	Used in iteration to determine number of rounds needed in addition to 1 volley.
ECO	Real	An intermediate value in the calculation of fractional survivors.
ECZ	Real	An intermediate value in the calculation of fractional survivors.
FPF	Real	% of personnel going from prone to crouching posture (now warned) after 1st volley.
FSF	Real	% of personnel going from standing to crouching posture (now warned) after 1st volley.
FSP	Real	% of personnel going from standing to prone posture (now warned) after 1st volley.
I	Int.	Subscript to identify personnel and material postures.
IA	Int.	Subscript to identify which fire unit on the "A" array is being considered.
IB	Int.	Subscript to identify personnel and material postures.

EFFECT (Cont'd)

Name	Units	Meaning
IC	Int.	Subscript to identify personnel and material postures.
IE	Int.	Subscript to identify personnel and material postures.
IST	Int.	Subscript to identify the system for the fire unit being considered.
IT	Int.	Subscript to identify which target on the target list is being considered.
K	Int.	Subscript to identify personnel and material postures.
KB	Int.	Subscript to identify personnel and material postures.
KR	Int.	Subscript to identify which round of those input is being considered.
LVT	Decimal Hours	Target leave time from its current location.
MP	Int.	Used to define main target element as Personnel (MP=1) or Material (MP=2).
NCODE	Int.	A variable used to control program flow.
NPSET	Int.	Used to define target posture mix.
OVN	Real	Total value of fractional survivors in the target.
RASR	Meters	Square root of ratio of Rd-Rd CPE/Total CPE.
RND	Real	Defines type of round as ICM (RND=1) or HE(RND=2).
SHEAF	$Meters^2$	Defines sheafing error, depending on system type, TLE and target activity.
SRV	Real	Fractional survivors of main target element.
TNSQ	$Meters^2$	$(Target Location Error)^2$
UP	Rounds	Used in iteration to determine number of rounds needed in addition to 1 volley.

COV



Name	Units	Meaning
B	Radians	Angle θ_2
CB	Real	Cos θ_2
CD	Real	Cos θ_1
CPSI	Real	$1./(\text{Total CPE})^2$
D	Radians	Angle θ_1
DELTA	Real	Step size for integration (DR1/4).
DR	Real	Incremented miss distance.
DR1	Real	Used to determine step size of integration (RMAX/16).
F1R	Real	Coverage for a given value of R.
FR	Real	Cumulative sum of coverage (as function of R).
R	Real	Current miss distance for which coverage is computed.
RHOR	Real	Probability of occurence of miss distance R.
RMAX	Real	Linear sum of Radius of Effects & Radius of Target.
RMAX1	Real	Limit of integration (Maximum miss distance).
SD	Real	Sin θ_1
T1	Real	Radius of Effects/Radius of Target.
T2	Real	$(\text{Radius of Effects/Radius of Target})^2 = (T1)^2$
T6	Real	Coverage weighted by probability of occurence.
TEMP	Real	3.61 * Total CPE
TX	Real	Ratio of miss distance to target radius.

Name	Units	Meaning
		OUTPUT
APCS	APCs	Total number of APCs defeated by all systems up to current time.
CAS	Personnel	Total number of personnel defeated by all systems up to current time.
CT	Kilo $	Total cost of ammo fired by all systems up to current time.
HA(10)	APCs	Number APCs defeated by each system at end of previous hour.
HT(10)	Tanks	Number tanks defeated by each system at end of previous hour.
I	Int.	Subscript to identify system, range & rounds.
ICOUNT(30)	KM	A count of Gun-Target ranges from 1 to 30 km.
IRDCNT(50, 30)	Rounds	Total number of each round in game fired at ranges from 1 km to 30 km.
J	Int.	Subscript to identify ranges.
K	Int.	Subscript to identify ranges.
NQAL	Msns.	Number of all missions not undertaken plus those still on target list.
NRACQ	Targets	Number of regular targets acquired (excluding "OTHER" msns).
NREAC	Targets	Number of regular targets reacquired.
NRF	Targets	Number of regular targets fired upon, (excluding "OTHER" msns).
NRFL	Targets	Number of regular targets fired upon but not defeated.
NRQ	Targets	Number of regular targets not undertaken (excluding "OTHER" msns).
NRQL	Targets	Number of regular targets not undertaken plus those on target list (less "OTHER" msns).
NRTGT	Targets	Number of regular targets in game (excluding "OTHER" msns).
NTD	Targets	Number of regular targets defeated plus "OTHER" msns fired.
PCTQ	Real	(Sum of msns not undertaken/total acquisitions) * 100.
RSUM	Rounds	Total rounds fired up to current time for each round in game.
SACQ	Acquisitions	Sum of target acquisitions.
SQ	Msns.	Total of all missions not undertaken.
TKS	Tanks	Total number of tanks defeated by all systems up to current time.
WG	Metric Tons	Total weight of ammo fired by all systems up to current time.
		SPECIL
A9	KM	Gun-Target range for a given fire unit firing the mission.
B	Decimal Hours	Time used by a cannon fire unit to fire more than 1 volley.
CSTM	Kilo $	Temporary storage used to determine cheapest round.
IC	Int.	Subscript to clear "A" array when next echelon is to be tried.
ICR	Int.	Subscript to clear "A" array when next echelon is to be tried.
IFS	Int.	Subscript to identify which fire unit of all those input is being considered.
IKR	Int.	Subscript to identify which round of those input is being used.
INSS	Int.	Subscript to identify the site at which a given fire unit is now located.
IRNG	KM	Subscript to identify Gun-Target range rounded off to nearest km.
IRS	Int.	Subscript to identify which round of all input rounds is being considered.
ISS	Int.	Subscript to identify the system for the fire unit being considered.
IST	Int.	Subscript to identify the mission on the AMSN array.
IT	Int.	Subscript to identify the target ("OTHER" msn) on target list which is being fired.
IXZ	Int.	Subscript to identify which fire unit of all those input is being considered.
JFS	Int.	Subscript to identify which fire unit on "A" array is being considered.

Name	Units	Meaning
		SPECIL (Cont'd)
K	Int.	Subscript to identify the system for the fire unit being considered.
LS	Int.	Subscript to check which rounds of those input can be used by a given fire unit.
MORGT1	Int.	Key to identify that DS echelon has been considered.
MORGT2	Int.	Key to identify that GS echelon has been considered.
MORGT3	Int.	Key to identify that CORPS echelon has been considered.
MS	Int.	Subscript to identify the rounds allowed for the fire unit being considered.
NAK	Int.	Counter of number of fire units available for this mission.
NS	Sites	The number of sites a given fire unit occupies during the game.
R	Rounds	Intermediate, then final number of rounds a given fire unit can fire vs target.
RDNO	Rounds	Number of rounds the fire unit which will fire the mission will expend.
RNGINT	KM	Used to round-off Gun-Target range to nearest km.
TCST	Kilo $	Temporary value used to determine the round with the cheapest total cost.
Z	Rounds	Number of rounds a given fire unit has on hand at current time.

The following page is blank.

APPENDIX D

SAMPLE INPUT LISTING

The following page is blank.

SAMPLE INPUT FOR AMSAA TECHNICAL REPORT NO. 97

```
 -6.00    24.0    3.0
0.0     .6      0.0     .4      0.0     0.0     0.0     .25     .75     0.0     0.0
1.0     .5      .5      0.0     0.0     0.0     .5      .5      0.0     0.0     0.0
2.0     0.0     .75     .25     0.0     0.0     0.0     .75     .25     0.0     0.0
3.0     0.0     0.0     1.0     0.0     0.0     0.0     0.0     1.0     0.0     0.0
4.0     .25     0.0     .75     0.0     0.0     0.0     .25     .75     0.0     0.0
5.0     0.0     0.0     0.0     0.0     1.0     0.0     0.0     0.0     0.0     1.0
6.0     0.0     0.0     0.0     0.0     1.0     0.0     0.0     0.0     0.0     1.0
7.0     0.0     0.0     0.0     1.0     0.0     0.0     0.0     0.0     1.0     0.0
8.0     0.0     0.0     0.0     1.0     0.0     0.0     0.0     0.0     1.0     0.0
9.0     0.0     0.0     0.0     1.0     0.0     0.0     0.0     0.0     1.0     0.0
10.0    0.0     0.0     0.0     1.0     0.0     0.0     0.0     0.0     1.0     0.0
11.0    .00     .75     0.25    0.0     0.0     .00     .75     0.25    0.0     0.0
  2
1200.3    .667     18.      4.      3.      2.   4500.    400.     12.     10.
   60.      1.
1400.1      0.      6.      3.      2.      2.   1000.    120.      9.      7.
   60.      1.
  5
1203.3    .080    .200     16.     .95      1.      1.      5.
    0.      3.      6.      9.     12.     15.     16.
   15.     15.     20.     23.     35.     63.     90.
   55.     55.    100.    135.    165.    260.    300.
    2.     30.    .910     .81     42.
  110.     67.    15.2      0.      0.     25.     15.     2.6      0.      0.
1204.3    .086    .270     16.     .92      1.      1.      5.
    0.      2.      4.      6.      8.     10.     12.     14.     16.
   10.     10.     14.     20.     27.     37.     50.     82.    110.
   55.     55.     72.    100.    119.    142.    165.    236.    315.
   2.6     21.    .920     .85     60.
   90.     37.     12.     5.0     7.6     23.     10.     1.2     1.4     2.1
1205.3    .067    .312     24.     .95      1.      2.      0.
    0.      4.      8.     12.     16.     20.     24.
   12.     12.     17.     23.     35.     62.     90.
   45.     45.     87.    136.    170.    240.    400.
 1110.   1110.   1280.   1325.   1572.   2470.   3050.
  908.    908.    930.    995.   1123.   1680.   2720.
  200.    200.    215.    225.    230.    235.    235.
   50.     50.     50.     50.     50.     50.     50.
   73.     73.     73.     73.     73.     73.     73.
  375.    375.    422.    456.    582.    870.   1105.
  280.    280.    305.    327.    384.    620.    890.
   57.     57.     53.     49.     41.     30.     23.
   25.     25.     25.     25.     25.     25.     25.
   67.     67.     67.     67.     67.     67.     67.
1401.1    .200    .612     22.     .93      1.      1.      5.
    0.      4.      8.     12.     16.     20.     24.
    0.     12.     18.     26.     40.     61.     70.
    0.     34.     67.    113.    170.    246.    277.
   1.3     66.    .947     .72    512.
  210.     98.     12.      0.      0.    160.     74.      8.      0.      0.
1402.1    .157    .478     22.     .93      1.      2.      0.
    0.      8.     16.     24.
   30.     30.     40.     80.
   72.     72.    120.    290.
 1110.   1450.   2000.   2325.
  800.    975.   1540.   1850.
```

170.	207.	280.	300.
60.	60.	60.	60.
90.	90.	90.	90.
810.	820.	950.	1060.
490.	490.	510.	520.
60.	60.	65.	70.
35.	35.	35.	35.
65.	65.	65.	65.
10			
4			
1200.3			
- 6.	06.30	465.0	593.3
08.57	15.00	468.0	583.3
17.27	18.30	479.2	584.4
20.00	30.00	481.5	577.5
4			
1200.3			
- 6.	05.00	468.5	596.6
07.06	08.00	471.0	598.0
10.27	12.00	470.3	582.5
13.21	30.00	471.7	578.0
3			
1200.3			
- 6.	06.20	483.6	596.0
08.14	10.00	488.4	590.2
12.06	30.00	489.5	582.1
3			
1200.3			
- 6.	05.15	462.3	595.9
06.57	09.30	462.0	589.4
11.24	30.00	462.5	582.5
4			
1200.3			
- 6.	06.30	480.9	593.2
08.24	09.30	479.2	585.5
10.45	18.00	478.3	574.7
18.57	30.00	479.5	579.4
4			
1200.3			
- 6.00	06.00	457.5	597.4
08.06	11.00	458.5	591.5
12.54	21.00	459.8	584.5
22.09	30.00	460.3	580.2
4			
1400.1			
- 6.	05.48	480.6	597.2
06.36	09.33	484.4	590.9
10.06	13.07	485.0	583.7
13.44	30.00	483.5	577.4
4			
1400.1			
- 6.	05.00	480.6	597.2
05.48	09.00	484.4	590.9
09.33	12.30	485.0	583.7
13.07	30.00	483.5	577.4
4			
1400.1			
- 6.	06.36	480.6	597.2
07.24	10.06	484.4	590.9
10.39	13.44	485.0	583.7

```
14.21    30.00    483.5    577.4
   3
1400.1
- 6.     06.00    465.7    593.5
06.45    09.30    468.5    588.5
10.25    30.00    468.5    578.2
1200.3 1203.3 1204.3 1205.3
1400.1 1401.1 1402.1
   0     5     5
1203.3 1204.3 1205.3 1401.1 1402.1
1203.3 1204.3 1205.3 1401.1 1402.1
   1     5     5
1203.3 1204.3 1205.3 1401.1 1402.1
1203.3 1204.3 1205.3 1401.1 1402.1
   2     5     5
1203.3 1204.3 1205.3 1401.1 1402.1
1203.3 1204.3 1205.3 1401.1 1402.1
   3     5     5
1203.3 1204.3 1205.3 1401.1 1402.1
1203.3 1204.3 1205.3 1401.1 1402.1
   4     5     5
1203.3 1204.3 1205.3 1401.1 1402.1
1203.3 1204.3 1205.3 1401.1 1402.1
   5     3     3
1204.3 1205.3 1402.1
1204.3 1205.3 1402.1
   6     3     3
1204.3 1205.3 1402.1
1204.3 1205.3 1402.1
   7     3     3
1204.3 1205.3 1402.1
1204.3 1205.3 1402.1
   8     3     3
1204.3 1205.3 1402.1
1204.3 1205.3 1402.1
   9     3     3
1204.3 1205.3 1402.1
1204.3 1205.3 1402.1
  10     3     3
1204.3 1205.3 1402.1
1204.3 1205.3 1402.1
  11     5     5
1203.3 1204.3 1205.3 1401.1 1402.1
1203.3 1204.3 1205.3 1401.1 1402.1
SAMPLE MIX FOR AMSAA TECHNICAL REPORT NO. 97
11
11111
1112233332
2.
```


The following page is blank.

<<

APPENDIX E

SAMPLE OUTPUT LISTING

The following page is blank.

TIME= 24.0000 SAMPLE MIX FOR AMSAA TECHNICAL REPORT NO. 97

ACQ= 3284.00 PERS= 48511.42 TANK= 1396.03 APC= 2251.60 MIL WORTH= 7774.30

SYSTEM	COST	WEIGHT	PERSONNEL	TANKS	APCS		
1200.30	10116.0342	3263.4341	4017.7347	7.6976	34.1575		
1400.10	5890.2011	1924.9070	7968.4797	-0.0000	-0.0000		
TOTALS	16006.2353	5188.3411	11986.2144	7.6976	34.1575	PCTQ = 56.8921	MW= 4575.03

RANGE IN KILOMETERS

ROUND ID	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1203.30	189	0	303	696	1015	736	579	1750	1346	1768	1982	1398	1243	2863	2924
1204.30	264	101	201	100	888	805	1411	2401	4409	1876	423	782	359	0	0
1205.30	0	0	0	183	0	0	0	0	0	26	0	0	0	6	198
1401.10	5	21	5	27	93	44	93	223	220	393	724	552	840	788	807
1402.10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ROUND ID	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	RNDSUM =
1203.30	42	0	0	0	0	0	0	0	0	0	0	0	0	0	0	18840.8022
1204.30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	14026.2479
1205.30	1118	2780	804	559	36	1098	338	584	471	0	0	0	0	0	0	8207.6502
1401.10	906	744	885	706	681	570	282	0	0	0	0	0	0	0	0	9619.8252
1402.10	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6.0000

	OTHER MISSIONS	REGULAR TARGETS	TOTALS
ACQUISITIONS=	369	2915	3284
NO. OF TARGETS=	369	807	1176
NO. MSN/TGTS QUEUED=	68	1261	1329
SUM QUEUED + STILL ON LIST=	68	1352	1420
NO. MSN/TGTS FIRED=	301	615	916
NO. MSN/TGTS DEFEATED=	301	254	555
TGT FIRED BUT LOST=	0	361	361

QUEUED MISSION TOTAL INCLUDES THOSE DROPPED DUE TO LOW PRIORITY(0), THOSE DEPARTED BEFORE ATTEMPT TO FIRE(1261), AND OTHER-TYPE MISSIONS TRIED BUT CANT DO(68).

NO. OF TGTS STILL ON TGT LIST= 91.

NO. OF PREVIOUSLY DEFEATED TGTS WHICH ARE REACQUIRED= 657.

NO. OF TARGETS COMBINED (WITHIN 200 METERS)= 291.

TOTAL OF ALL REACQUISITIONS= 2108.

DISTRIBUTION LIST

No. of Copies	Organization
12	Commander Defense Documentation Center ATTN: TIPCR Cameron Station Alexandria, VA 22314
1	Commander US Army Materiel Command ATTN: AMCCP 5001 Eisenhower Ave Alexandria, VA 22333
1	Commander US Army Materiel Command ATTN: AMCDL 5001 Eisenhower Ave Alexandria, VA 22333
1	Commander US Army Materiel Command ATTN: AMCMA 5001 Eisenhower Ave Alexandria, VA 22333
1	Commander US Army Materiel Command ATTN: AMCGS-L 5001 Eisenhower Ave Alexandria, VA 22333
1	Commander US Army Materiel Command ATTN: AMCPA-S 5001 Eisenhower Ave Alexandria, VA 22333
1	Commander US Army Materiel Command ATTN: AMCQA 5001 Eisenhower Ave Alexandria, VA 22333
1	Commander US Army Materiel Command ATTN: AMCRD-E 5001 Eisenhower Ave Alexandria, VA 22333
1	Commander US Army Materiel Command ATTN: AMCRD-G 5001 Eisenhower Ave Alexandria, VA 22333
1	Commander US Army Materiel Command ATTN: AMCRD-M 5001 Eisenhower Ave Alexandria, VA 22333
1	Commander US Army Materiel Command ATTN: AMCRD-R 5001 Eisenhower Ave Alexandria, VA 22333
1	Commander US Army Materiel Command ATTN: AMCRD-P 5001 Eisenhower Ave Alexandria, VA 22333
1	Commander US Army Materiel Command ATTN: AMCRD-W 5001 Eisenhower Ave Alexandria, VA 22333
1	Commander US Army Materiel Command ATTN: AMCRD-T 5001 Eisenhower Ave Alexandria, VA 22333

DISTRIBUTION LIST (cont)

No. of Copies	Organization
1	Commander US Army Materiel Command ATTN: AMCPM-SA Picatinny Arsenal Dover, NJ 07801
1	Commander US Army Armament Command ATTN: AMSAR-SA Rock Island, IL 61201
1	Commander Harry Diamond Laboratories ATTN: Sys Anal Ofc Washington, D.C. 20438
1	Commander Frankford Arsenal Philadelphia, PA 19137
1	Chief, Analytical Sciences Office USA Biological Defense Research Laboratory ATTN: AMXBL-AS Dugway, UT 84022
1	Commander US Army Aviation Systems Command ATTN: AMSAV-ERW PO Box 209 St. Louis, MO 63166
1	Commander US Army Electronics Command ATTN: AMSEL-SA Fort Monmouth, NJ 07703
1	Commander US Army Missile Command ATTN: AMSMI-DA AMSMI-C Redstone Arsenal, AL 35809
1	Commander US Army Troop Support Command ATTN: Sys Anal Ofc 4300 Goodfellow Blvd St. Louis, MO 63120
1	Commander Rock Island Arsenal ATTN: SARRI-LR-S Rock Island, IL 61201
1	Commander US Army Armament Command ATTN: Sys Anal Ofc Rock Island, IL 61201
1	Manager Defense Logistics Studies Information Exchange US Army Logistics Management Center ATTN: Mr. Leon T. Scarbrough Fort Lee, VA 23801

DISTRIBUTION LIST (cont)

No. of Copies	Organization
1	Commander Picatinny Arsenal ATTN: SARPA-AD-C (Dr. S. Einbinder) Dover, NJ 07801
1	Project Manager, ARTADS ATTN: AMCPM-TDS Fort Monmouth 07703
1	Commander USA Field Artillery School ATTN: ATSF-CTD Fort Sill, OK 73503
1	Commander USA Combined Arms Combat Developments Activity ATTN: ATCACA-A Fort Leavenworth, KS 66027

Aberdeen Proving Ground

Cdr, USATECOM
ATTN: AMSTE-TS

Ch, Tech Lib
Bldg. 305

Dir, BRL
Bldg. 328

No. of Copies	Organization
1	Commander US Army Safeguard Evaluation Agency ATTN: SSEA-T White Sands Missile Range, NM 88002