

AD-A061 363

SRI INTERNATIONAL MENLO PARK CA NAVAL WARFARE RESEAR--ETC F/G 15/7
STRIKE OUTCOME CALCULATOR (SOC)--DESCRIPTION, AND OPERATING INS--ETC(U)
MAR 78 R S GARNERO, J C BOBICK, D AYERS N00014-75-C-0742

UNCLASSIFIED

1 OF 2
AD
A061363

NWRC-TR-15

NL



DDC FILE COPY AD A061363

NAVAL WARFARE RESEARCH CENTER

333 Ravenswood Avenue
Menlo Park, California 94025 U.S.A.
(415) 326-6200
Cable: STANRES, Menlo Park
TWX: 910-373-1246



Prepared for:
NAVAL ANALYSIS PROGRAMS (Code 431)
OPERATIONAL DECISION AIDS PROJECT
OFFICE OF NAVAL RESEARCH
DEPARTMENT OF THE NAVY
ARLINGTON, VIRGINIA 22217

CONTRACT N00014-75-C-0742

15

This document has been approved
for public release and sale; its
distribution is unlimited.

410 521

78 10 23 135

LB

**STRIKE OUTCOME CALCULATOR (SOC)--
DESCRIPTION, AND OPERATING INSTRUCTIONS.**

By: ROBERT S. GARNERO, JOHN C. BOBICK, DAVID AYERS

9

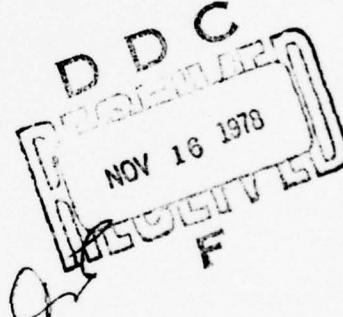
Technical Report,
NWRC-TR-15

14

11 March 1978

12

12 15% p.





sri
International

Technical Report
NWRC-TR-15

March 1978

STRIKE OUTCOME CALCULATOR (SOC)-- DESCRIPTION, AND OPERATING INSTRUCTIONS

By: ROBERT S. GARNERO JOHN C. BOBICK DAVID AYERS

Prepared for:

NAVAL ANALYSIS PROGRAMS (Code 431)
OPERATIONAL DECISION AIDS PROJECT
OFFICE OF NAVAL RESEARCH
DEPARTMENT OF THE NAVY
ARLINGTON, VIRGINIA 22217

CONTRACT N00014-75-C-0742

SRI Project 4028

Approved by:

A. BIEN, *Director*
Naval Warfare Research Center

D. D. ELLIOTT, *Executive Director*
Systems Research and Analysis Division

333 Ravenswood Avenue • Menlo Park, California 94025 • U.S.A.

7810 23 135

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) STRIKE OUTCOME CALCULATOR (SOC), DESCRIPTION AND OPERATING INSTRUCTIONS		5. TYPE OF REPORT & PERIOD COVERED Technical Report NWRC-TR-15
7. AUTHOR(s) Robert S. Garner, John C. Bobick, David Ayers		6. PERFORMING ORG. REPORT NUMBER SRI Project 4028
9. PERFORMING ORGANIZATION NAME AND ADDRESS SRI International 333 Ravenswood Avenue Menlo Park, California 95014		8. CONTRACT OR GRANT NUMBER(s) N00014-75-C-0742
11. CONTROLLING OFFICE NAME AND ADDRESS Operational Decision Aids Project (Code 431) Office of Naval Research Department of the Navy Arlington, Virginia 22217		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS Task NR 274-199
14. MONITORING AGENCY NAME & ADDRESS (if diff. from Controlling Office)		12. REPORT DATE 13. NO. OF PAGES March 1978 1
16. DISTRIBUTION STATEMENT (of this report)		15. SECURITY CLASS. (of this report) Unclassified
		16a. DECLASSIFICATION/DOWNGRADING SCHEDULE
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from report)		
18. SUPPLEMENTARY NOTES Reproduction in whole or in part is permitted for any purpose of the United States government.		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Naval Air Strike Man-Machine Interaction Decision Aid Outcome Calculation Naval Combat Computer Simulation		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) This report describes an innovative and efficient outcome calculator to assist a task force commander in obtaining estimates for possible sequences of decisions and events related to air strike planning. The computational algorithm, input requirements, outputs, and man-machine interactions are described.		

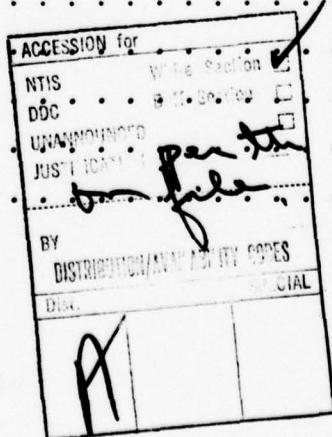
DD FORM 1 JAN 73 1473
EDITION OF 1 NOV 65 IS OBSOLETE

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

CONTENTS

LIST OF ILLUSTRATIONS	vii
LIST OF TABLES	vii
LIST OF EXHIBITS	ix
I INTRODUCTION	1
A. Project Background	1
B. SOC Concept	2
C. Document Purpose and Organization	6
D. Computer Hardware and Software Requirements	6
II SOC Data	9
A. Introduction	9
B. Input	10
1. Background Data	10
2. Scenario Data	22
3. Course of Action Data	26
C. Output	31
III SOC Battle Logic	35
A. Description	35
B. NEWRUN	35
C. NEWDAY	37
D. SCHEDULE	37
E. RECOVER	39
F. LAUNCH	39
G. ENGAGE	40



IV	SOC MAN-MACHINE INTERACTION	43
A.	Interactive Commands	43
1.	LOAD INPUT.	43
2.	SAVE INPUT.	43
3.	DISPLAY/INPUT TABLE	43
4.	PRINT TABLE	43
5.	RUN	43
6.	STOP.	44
7.	LIST TABLES	44
8.	EXPLAIN ABOVE FUNCTIONS	44
B.	Command Use.	44
APPENDICES		
A	ONRODA PROBLEM SCENARIO.	47
B	SOC PROGRAM LISTINGS	55

ILLUSTRATIONS

1 SOC Concept	3
2 Macro Flowchart of SOC Computational Algorithm	36
A-1 ONRODA Scenario Area Map.	50

TABLES

1 Strike Outcome Calculator Computer Tables	9
2 Sample Engagement Data Base	18
A-1 ONRODA Scenario Blue Task Force Major Combatants	52
A-2 ONRODA Scenario Enemy Major Combatants.	53

EXHIBITS

1	Blue and Red Force Elements	11
2	Blue Force Units.	13
3	Red Force Units	13
4	Engagement Statistics, Blue Attacking Red	16
5	Engagement Statistics, Red Attacking Blue	16
6	Weapon Platform Availability.	21
7	Operations and Damage Repair Capabilities	21
8	Blue Force Complexes.	23
9	Red Force Complexes	23
10	Miscellaneous Inputs	25
11	Blue Operations Plans	27
12	Red Operations Plans.	27
13	Relative Force Positions.	30
14	Weather Days.	30
15	Blue Mission Accomplishment Results	32
16	Red Mission Accomplishment Results.	32
17	Blue Battle Attrition	33
18	Red Battle Attrition Results	33
19	Aircraft Expenditure Summary.	34

I INTRODUCTION

A. Project Background

The Office of Naval Research (ONR) is currently pursuing a 5-year multicontractor program to promote the development of decision aids and procedures in support of fleet operations. The objective of the work is to improve tactical decisionmaking by blending a number of technologies such as decision-analysis, computer-driven graphic displays, advanced data management systems, information feedback, mathematical prediction, tactical models, and organizational analysis into a practical system for shipboard use. By concentrating on the needs and activities of the task force commander (TFC) and his staff, the project emphasizes decision aids that rely on the judgments of senior officers experienced in operational situations, rather than on the predictions of system designers. These aids will provide guidelines and tools to structure decision problems, elicit judgments of probabilities and outcome preferences, furnish stored data and models requested by the decision maker, make statistical inferences, and display and/or print the implications of trial tactics before their execution. All of these objectives are compatible with ongoing command-control hardware programs. Decision makers will thus be provided with a man-computer interactive capability to help them examine and evaluate alternative courses of action.

The Naval Warfare Research Center (NWRC) of SRI International (formerly Stanford Research Institute) has been a continuing participant in the program under Contract N00014-75-C-0742. NWRC has evaluated and produced a specific task force decision-aiding procedure called the Strike Outcome Calculator (SOC). SOC, which is an automated decision aid for use

in estimating outcomes associated with naval air strikes, is described here in detail to allow its employment by decision makers. For further background on SOC, refer to the following two documents produced by NWRC: "Augmentation of the Naval Task Force Decision-Aiding System: The Outcome Calculator" (April 1977) and "Evolution and Preliminary Tests of the Strike Outcome Calculator" (March 1978).

B. SOC Concept

SOC is a decision aid that enables a user quickly and easily to estimate battle outcomes associated with alternative courses of action (COA). SOC consists not only of a computational algorithm, but of an interactive medium to facilitate both the description of alternative COAs and the display of the associated battle outcomes. A decision aid like SOC may prove useful in decisions concerned with long-range planning, contingency planning, and short-range tactical execution. SOC is tailor-made for use as a component of a naval task force decision-aiding system. In addition to this role, SOC is, in its own right, a decision aid and can be used independently of any structured decision problem solving system.

The SOC concept is illustrated in Figure 1. As the figure indicates, the major parameters that enter into the choice among alternative COAs are timing, use of assets, force position, threat action, and weather. Combinations of parameters can produce several alternatives, which are sometimes represented as branches of a "decision tree." SOC evaluates outcomes associated with such branches. A user generally describes a possible COA by prescribing values of these parameters in aggregate, often qualitative, terms. SOC then allows the user to easily describe alternative COAs over a wide range of values of associated parameters, and yet maintain a level of detail consistent with his needs.

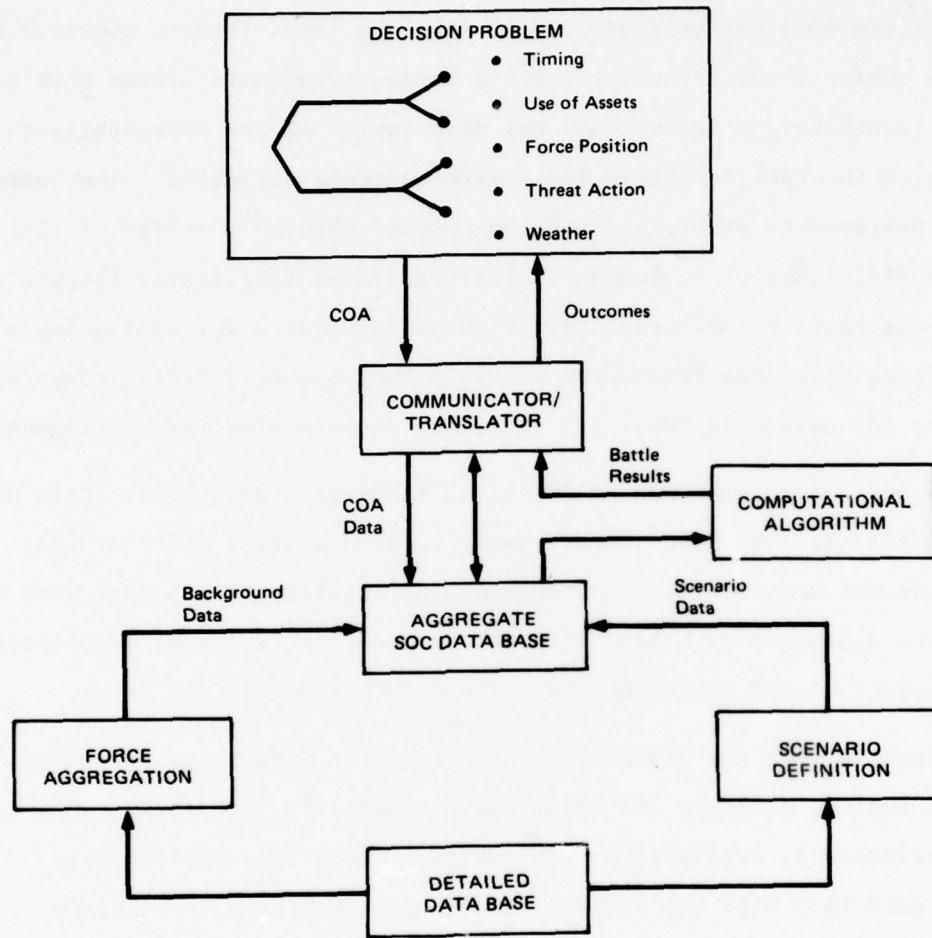


FIGURE 1. SOC CONCEPT

As shown in Figure 1, the SOC system contains three major components: the communicator/translator, the aggregate SOC data base, and the computational algorithm. The communicator/translator is the man-machine interface between the decision-maker and the computational algorithm and data base. The translator is sufficiently familiar with the decision problem to translate aggregate qualitative descriptions into the input formats required by SOC. The communicator is an interactive hardware/software system that permits the translator to communicate the description of the COAs easily to the computer in order to obtain the desired outcome estimates. The communicator has been designed to enable the user to proceed through a series of alternative COAs efficiently. The communicator/translator facilitates the use of SOC as a component of an overall decision-aiding system for evaluating a set of alternative COAs identified within a decision structuring component as well as for answering "what if" questions and planning for contingencies.

A second major component of SOC is the aggregate data base. This data base contains data of three types: background, scenario, and COA. All data in the SOC are at a level of aggregation consistent with that used by a TFC. The aggregate data base facilitates the translation of descriptions of the situations and alternative decisions into quantitative terms.

Background data are generally constant over a wide range of scenarios. Such data include friendly and enemy force capability data (e.g., weapon system performance, availability, and effectiveness characteristics). Scenario data vary with the scenario under consideration, and include friendly and enemy resources, force concentration, and environment. COA data include the timing, use of assets, force position, and threat action data that describe a particular COA.

Background data are generated by manual aggregation of the detailed weapon system performance and operations data to the level required by SOC. Similarly, developing scenario data involves manual procedures in

which more detailed operational and intelligence data are used to obtain the aggregate data for SOC.

In general, background and scenario data will have been loaded previously into the aggregate SOC data base. For various decision problems, the user provides COA data by means of the communicator/translator and receives outcome estimates. The battle results are generated by the computational algorithm from the current background, scenario, and COA data in the data base. A series of alternatives are investigated by modifying the COA data to move from one alternative to another. The communicator provides the user with direct access to all the data in the aggregate data base and allows him to display and modify all data.

The computational algorithm uses the data in the aggregate data base for straightforward computation of battle results. The relative simplicity of the algorithm, which enables interested users to understand the number-generating routines, does not imply a lack of credibility. The algorithm is consistent with the level of detail of the data, and therefore the integrity of the overall system is maintained. In fact, the absence of internal assumptions, which are common in more detailed algorithms, enhances the credibility of the SOC algorithm. In essence, the SOC computational algorithm provides a consistent, understandable, automated means of transforming a COA, as described by the user in terms of the data, into the implications of the action in terms of attrition and battle accomplishment results.

The data are structured to permit model credibility without overburdening detail. In most battle simulations, as the inputs become more numerous or detailed (with a resulting increase in model sophistication), the user cannot rapidly and easily see the underlying numerical routines; this reduces his ability to relate the causal effects of the input to the output. The aggregated data of SOC allow the user to avoid lack of

confidence associated with more detailed data inputs that are characterized by hidden and inflexible assumptions, doctrines, and numerical techniques within the model. Additionally, with SOC users can employ their judgement and experience as necessary or desired.

C. Document Purpose and Organization

This document provides the information required to exercise SOC methodology properly. Because the audience for this volume is comprised of naval decision makers, users are assumed to understand relevant Navy systems, procedures, and terminology. Use of this document does not require a technical data processing background. However, a general knowledge of the basic principles of data processing will aid the user.

The document sets forth computer hardware and software requirements (Section I D) necessary to exercise the SOC and also states the input data requirements (Section II B). These requirements appear early in the document to help explain the computational algorithm used within SOC (Section III). The inputs are stated in conjunction with sample data generated from the problem scenario in Appendix A. Section II on SOC data closes with an interpretation of the generated output.

Later sections detail the major subroutines used within the SOC and describe the systems man-machine capabilities. Appendix B shows the SOC program listing.

D. Computer Hardware and Software Requirements

SOC is currently configured for a PDP-10 computer and uses the time-sharing, interactive capability of that system. It requires 35,000 words of core storage and is written primarily in the FORTRAN and MACRO (assembly) languages. Each MACRO routine has a single purpose and is called up via FORTRAN to perform a single function, primarily in the area of character manipulation and terminal (CRT) input/output (I/O).

SOC requires a CRT nonstorage screen with the following functions:

- Character-for-character transmission
- Erase screen via line
- Cursor movement via line
- Blink on/off via line (optional).

During the display and editing performed by SOC, the terminal needs to achieve an uninterrupted nonechoed character I/O. In some systems this is called BINARY or TRANSPARENT MODE. SOC is now programmed to handle two terminals, the DATAMEDIA 1520 and 2500, but can be expanded to accommodate more terminals with minor modifications. These modifications involve changes in the INIT, NWSCRN, PTBLC, and PTUBLIC subroutines. A line speed of at least 2400 baud is recommended to enhance SOC interaction.

To use SOC, the program will have to be compiled and readied for execution by the time-share user. A support file known as FORMS is required and must be installed on a direct-access device during program execution. The program reads this file (composed of SOC computer table formats) throughout the program execution.

II SOC DATA

A. Introduction

The SOC data base is organized into 19 internally stored computer tables. The first 14 are used by the decision maker to input parameters into the SOC computational algorithm and the last 5 are used to receive the associated results. The computer table titles are listed in Table 1. (To avoid confusion with other tables in this report, the computer tables discussed herein are portrayed as "exhibits.")

Table 1
SOC COMPUTER TABLES

<u>Computer Table</u>	<u>Title</u>	<u>Type</u>
1	Blue and Red Force Elements	
2	Blue Force Units	
3	Red Force Units	
4	Engagement Statistics, Blue Attacking Red	
5	Engagement Statistics, Red Attacking Blue	
6	Weapon Platform Availability	
7	Operations and Damage Repair Capabilities	
8	Blue Force Complexes	
9	Red Force Complexes	
10	Miscellaneous Inputs	
11	Blue Operations Plans	
12	Red Operations Plans	
13	Relative Force Positions	
14	Weather Days	
15	Blue Mission Accomplishment Results	
16	Red Mission Accomplishment Results	
17	Blue Battle Attrition Results	
18	Red Battle Attrition Results	
19	Aircraft Expenditure Summary	

This document examines the contents of each computer table, by looking first at the input computer tables and their associated three types of data: background, scenario, and COA. Their explanation is geared to an example from the ONRODA problem scenario in Appendix A. Examination of that scenario will aid the reader as he studies the input data examples that are shown. When applicable, suggestions are given to the user about how he might generate particular data elements. The physical entry of the data elements is presented in Section IV, SOC Man-Machine Interactions. Shaded areas of the exhibits indicate areas in the computer tables the user is not expected to, and, indeed, cannot change.

B. Input

1. Background Data

Blue and Red Force Elements--In the SOC concept, the friendly (Blue) and enemy (Red) forces are assumed to be composed of generic force elements. Exhibit 1 shows the setup for the Blue force elements and the Red force elements. Blue forces may be composed of any subset of the following six generic force elements: attack aircraft (ATTACK), all-weather attack aircraft (AW-ATTACK), low performance fighter (VF-LO), high performance fighter (VF-HI), Carrier (CV), and support ship. Red forces are composed of eight generic elements: low-performance attack aircraft (BOMBER-LO), high-performance attack aircraft (BOMBER-HI), fighter-bomber (VBF), interceptor fighter (VFI), surface-to-surface missile (SSM-SHIP), airbase, surface-to-missile site (SAM-SITE), and supply line.

The generic elements that are to make up the forces appear in the first column and cannot be changed. The second column contains user designators as desired. These designators indicate the implicit implications of the following input computer tables. Performance and operational data, input subsequently for each element of those tables, will be representative

EXHIBIT 1

BLUE FORCE ELEMENTS			RED FORCE ELEMENTS		
ELEMENT	EXAMPLE	CLASS	ELEMENT	EXAMPLE	CLASS
ATTACK	A-7E	OA	BOMBER-LO	PADGER A	OA
AW-ATTACK	A-6E	OA	BOMBER-HI	BADGER C	OA
VF-LO	F-4J	DA	VBF	SU-7B	DA
VF-HI	F-14A	DA	VFI	MIG-21	DA
CV	KITTY HAWK	OS	SSM-SHIP	SS-N-3	OS
SUPPORT SHIP	DLG LEAHY	DS	AIRBASE	ONRODA	OS
			SAM SITE	SAM-3	DS
			SUPPLY LINE	AAA	LS

OA-OFFENSIVE AIR
 DA-DEFENSIVE AIR
 OS-OFFENSIVE SURFACE
 DS-DEFENSIVE SURFACE
 LS-LOGISTICS SUPPORT

of the weapon system specified. Each generic element is also given a class designation: offensive air (OA), defensive air (DA), offensive surface (OS), defensive surface (DS), and logistics support (LS).

Blue and Red Force Units--In performing offensive or defensive missions, certain combinations of the force elements are used in SOC. The user can define a variety of force units by specifying that units be composed of combinations of the elements specified in computer table 1, which is shown in Exhibit 1.

Exhibit 2 shows a sample definition of Blue force units. VFCAP, SUCAP, and DLI are SOC-defined reserved words. Any unit to be used as a fighter combat air patrol must be labeled as type VFCAP; any unit to serve as a surface combat air patrol must be labeled as type SUCAP; and any unit used as a deck launch interceptor must be labeled as type DLI. All other units may be designated by any type name the user desires. Several different force units within a given type may be defined by designating unit subtypes. Every unit must be defined by a type designation. However, it is not necessary for each unit type to be further delineated by subtype. Two different units may have the same elements in their makeup if the user intends to use the units differently during the battle (i.e., the same aircraft flying different tactics in various weather conditions with varying weapon loads can have different associated engagement statistics).

For each Blue unit defined, the numbers of attack, all-weather attack, low and high performance fighters, and suppression aircraft^{*} must be specified. All-weather attack aircraft are used for suppression. In addition, several characteristics must be assigned to each unit, including: the maximum range of the unit (LONG or SHORT); the long-range default force unit (i.e., the force unit subtype to replace it if its maximum range

* Suppression aircraft are assumed to be specially configured, all-weather attack aircraft.

EXHIBIT 2

BLUE FORCE UNITS												
B FRC UNIT		ELEMENTS PER UNIT					UNIT CHARACTERISTICS					
TYPE	SUB	VAI	AWI	VF	VFI	SUP	MAX	RANGE	WORST	WX	MAX	SPEED
				LO	HI	A/C	RANGE	DEF	WX	DEF	DET	
ALPHA-1A	-	3	-	-	1	1	LONG	-	GOOD	C	100	.9
ALPHA-1P	-	3	-	-	1	1	LONG	-	GOOD	C	100	.9
ALPHA-1C	-	-	2	-	1	1	LONG	-	BAD	-	100	.9
ESCOR-1A	-	-	-	-	1	1	LONG	-	GOOD	C	100	.9
ESCOR-1C	-	-	-	-	1	1	LONG	-	BAD	-	100	.9
SUCAP-1	-	1	1	-	-	-	LONG	-	BAD	-	100	.9
VFCAP-1	-	-	-	-	-	1	SHORT	-	BAD	-	100	2.2
DLI	-	-	-	-	-	1	SHORT	-	BAD	-	100	2.2
-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-

EXHIBIT 3

RED FORCE UNITS												
R FRC UNIT		ELEMENTS PER UNIT					UNIT CHARACTERISTICS					
TYPE	SUB	BRI	BRI	VBF	VFI	SSM	MAX	RANGE	WORST	WX	MAX	SPEED
		LO	HI			SHP	RANGE	DEF	WX	DEF	DET	
FREE-1A	-	4	-	4	-	-	LONG	-	GOOD	B	300	.9
FREE-1B	-	4	-	-	-	-	LONG	-	BAD	-	300	.9
VBF-1A	-	-	-	8	-	-	LONG	-	GOOD	-	300	.9
SSM-1	-	-	-	-	-	1	SHORT	-	GOOD	-	200	.9
SLI-1	-	-	-	-	-	1	SHORT	-	BAD	-	-	1.5
-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-

is "SHORT" and it is assigned a long-range mission); the worst weather conditions (GOOD or BAD) in which it can perform; the bad weather default unit (i.e., the force unit subtype to replace it if the worst weather condition in which it can perform is GOOD and it is assigned a mission in BAD weather), maximum range (miles) at which it can be detected by the enemy; and the speed (mach) at which the unit operates. The default units differ only in subtype from the original unit. No default unit need be specified. If no range is specified, LONG is assumed. If no worst weather is specified, BAD is assumed.

Exhibit 3 shows a sample definition of Red force units. SLI and SSM are SOC defined words reserved for Red forces. Thus, any Red unit that is to operate as a strip launched interceptor must be specified as type SLI, and any unit used to simulate SSM attack must be specified as type SSM. The rest of computer table 3 is filled in using the same rules described in relation to computer table 2.

To arrive at the unit definitions, listing the capability of the elements is helpful, as is analyzing their employment (sortie rates). The degree of effort required to arrive at the unit definition and the composition of the unit must be a judgment of the TFC and his staff. These judgments for the most part are influenced by the defensive strength of the threat (e.g., SAMs and interceptors) and the planned duration, in days, of the friendly strikes.

With regard to Blue, if the target is heavily defended with enemy SAM batteries and interceptor aircraft, then the Blue support elements (VF and AW) could be strengthened at the expense of increased sortie rates or weakening the task force defense. Alpha A in Exhibit 2 is an example of this stronger unit:

3 attack 1 AW attack 1 VF

A weather contingency is also provided by establishing the Alpha C unit of 2 AW attack aircraft and 1 VF.

As many as 15 units can be described for each of the Blue and Red forces.

Engagement Statistics--Having defined the force units, the battle engagement statistics that provide the basis for determining Blue and Red losses in combat must be specified. The engagement statistics for Blue attacking Red are shown in Exhibit 4, and for Red attacking Blue in Exhibit 5. As these exhibits indicate, engagement statistics are fairly aggregate measures of the battle effectiveness of the force units previously defined. These measures include the number of attacking force units that can be killed by air-to-air and surface-to-air means per defensive element of the force under attack. A second set of statistics gives the number of the defending fighters that can be killed air-to-air per unit of attacking force. The third set of statistics gives the number of the defending force elements that can be killed surface-to-air per unit of attacking force that penetrates the defenses.

In specifying the effectiveness of an attacking force unit, the synergistic effects of the various assets in the unit (e.g., attack, fighter, and suppression aircraft) should be considered. The importance of such synergistic effects as well as the application of user judgment were primary factors in the decision to treat force assets in terms of units.

It is unnecessary for units appearing in computer tables 2 or 3 to also appear in computer tables 4 or 5; however, any unit that is specified for use in its respective operations table (computer tables 11 and 12) must appear in both the unit definition computer table and engagement statistics computer table.

EXHIBIT 4

EXHIBIT 5

Data that can be utilized to support the input elements of computer tables 4 and 5 include air-to-air exchange ratios, SAM effectiveness data, and air-to-surface effectiveness data. Examples of this type of data are illustrated in Table 2.

The data in Table 2 can be used with some simple computations to fill in parts of Exhibit 4. Air-to-air data are used in the VBF/VFI columns, SAM data in the SAM and SUPPLY LINE columns, and air-to-surface data in the surface effects columns. The following numerical examples use the data presented in Table 2 to manually yield the first row of entries in Exhibit 4.

AIR TO AIR

It will take 14 Red VBF to kill 1 Alpha A unit as follows:

10 VBF to 1 VF-HI

4 VBF to 4 VA

14 VBF to 1 unit

Therefore, the Blue loss to 1 VBF is said to be $1/14 = 0.07$ unit.

It will take 8 Red VFI to kill 1 Alpha A unit as follows:

4 VFI to 1 VF-HI

4 VFI to 4 VA

8 VFI to 1 unit

Therefore, the Blue loss to 1 VFI is said to be $1/8 = 0.125$ unit.

Maximum Red air losses per Blue unit:

1 VF-HI kills 4 VBF

4 VA kill 4 VBF

1 unit kills 8 VBF

Table 2
SAMPLE ENGAGEMENT DATA BASE

1. Air-to-Air Exchange Ratio

	MIG-21	MIG-19/SU-7	BADGER	SSM
F-14	1/4	1/10	0/4	0/3
A-7 or A-6	1/.67	1/1	0/1	NA

These kills must be limited by the armaments that are usually carried. Maximum kills per aircraft are:

F-14--4 MIG-21--1
A7/A6--1 SU-7--1/2

2. SAM Effectiveness

These data can be taken from historical attrition figures.

	Losses per 1000 Sorties		
	AIRFIELD ATTACK	WAR AT SEA	CAS/ARREC
A-7	15	3	5
A-6	7	3	3

3. Air-to-Surface

These data can be taken from China Lake studies or Joint Munitions Effectiveness Manual (JMEM) and will depend on weapons carried, altitude released, number of passes, mil accuracy of aircraft, weapon P_k , and type of target.

An example would be an A-7 loaded with laser-guided bombs. This scores a $0.9 P_k$ against an enemy aircraft on the ground. Therefore 3 A-7s in a unit may be said to "kill" 2.7 enemy aircraft on the ground.

1 VF-HI kills 4 VFI
4 VA kill 2.7 VFI
1 unit kills 6.7 VFI*

SAM EFFECTIVENESS

Blue unit Alpha A

$$3 \text{ VA sorties lose } \frac{3 \times 15}{1000} = 0.045$$

$$1 \text{ AW sortie lose } \frac{1 \times 7}{1000} = 0.007$$

$$\text{Blue units lost to SAM} = \underline{\underline{0.052}}$$

(Blue VF-HI are not engaged)

AIR-TO-SURFACE

Blue Unit Alpha (VF-HI do not attack ground targets.)

3 VA attack Red aircraft on ground
(with 2 LGB ($P_k = 0.9$))

$$\text{Red kills} = 3 \times 0.9 = 2.7$$

1 A6 attack SAM batteries (site radar)
(with Standard ARM) $\bar{P}_k = 0.9$

$$\text{Red SAM site kill} = 1 \times 0.9 = 0.9$$

The data for computer table 5 are manually computed in a similar manner. Badger and VBF bomb loads and CEPs must be estimated to give probable damage to the Blue CV and support ships. Probable damage must also be estimated for the SSM attack. The Blue VFCAP, DLI, and Blue SAM ships will operate on the defensive to kill those attackers.

Weapon Platform Availability--The next type of background data required is weapon platform availability statistics. The form for these

* During an engagement, weighted averages of the two maximums are computed, based on the defensive force composition.

data is shown in Exhibit 6. For each of the force elements, normal and surge sortie rates and a refuel-rearm time must be specified. The normal sortie rate for a weapon platform is the average number of sorties that can be mounted per day on a sustained basis (usually 30 days). The surge sortie rate is the maximum number of sorties that can be mounted in a 24-hour period. The refuel-rearm time is the minimum time before a weapon platform returning from a mission can be ready to depart on another mission. In the computational algorithm, the battle is simulated in 3-hour increments. Thus, the refuel-rearm time is specified in terms of the number of 3-hour time periods. If a refuel-rearm time of zero is specified for an element, that element can return from one mission at the beginning of a time period and leave on another mission in the same time period.

Capabilities of Aircraft-Related Elements--The last of the background data are the operations and damage repair capabilities of the Blue carrier and Red airbase elements. The form of these data is shown in Exhibit 7. For various damage levels, the maximum number of operations per 3-hour period and the maximum damage repair capability per period must be specified for each element of Blue CV and Red airbase.

In determining the maximum number of operations for an undamaged Blue carrier, the sortie rate times the total number of aircraft aboard is often used as the index of carrier operational capability. The question of whether to use a surge rate or a sustained rate should be considered; it may depend on the estimate of the length of the operation (or war). A sustained rate could be used for campaigns of one or more months, but a surge rate is particularly appropriate for use in analyses involving time-sensitive targets such as the ONRODA scenario. Damage parameters can be developed with the kind of data found in the David Taylor Model Basin Technical Note SML-740-37, "The Operational Capabilities of a Forrestal-Class Aircraft Carrier After Successful Missile Hits," F. Weinberger

EXHIBIT 6

WEAPON PLATFORM AVAILABILITY				
FORCE ELEMENT	NORMAL SR	SURGE SR	R/R TIME	
VA	1	2	1	
AW	1	2	1	
VF LO	1	2	1	
VF HI	1	2	1	
BR LO	1	1.5	1	
BR HI	1	2	1	
VBF	1	2	1	
VFI	1	2	1	
SSMSHIP	1	2	4	

EXHIBIT 7

CAPABILITIES OF A/C RELATED ELEMENTS					
DAMAGE LEVEL	MAX OPERATIONS PER PERIOD		MAX DMGE REPAIR PER PERIOD		
	--BLUE--	--RED--	--BLUE--	--RED--	
	CV	AIRBASE	CV	AIRBASE	
0.0	250	300	0	0	
0.1	225	260	.07	.07	
0.2	200	230	.12	.12	
0.3	125	200	.15	.15	
0.4	100	175	.16	.16	
0.5	75	150	.15	.15	
0.6	24	24	.12	.12	
0.7	24	24	.07	.07	
0.8	0	0	0	0	
0.9	0	0	0	0	
1.0	0	0	0	0	

and R. Santa Maria (March 1965). However, the numbers used for computer table 7 shown in Exhibit 7 for "MAX DAMAGE REPAIR PER PERIOD" could be the TFC's estimates of the capability of his damage control units aboard a specific carrier. That is, for 20% damage to a carrier the repair crews might repair 12/20 of the damage in a 3 hour period.

This concludes the background data inputs. These data are generally derived by aggregating more detailed data found in Navy data bases and publications. As indicated in the description above, these data are fairly static, depending on the weapon systems and operating doctrine involved.

2. Scenario Data

Force Complexes--A concept employed in defining scenario data is that of force complexes. Force complexes are groupings of Blue or Red force elements. Each force can have as many as 8 complexes. The forms for the data used to define complexes are shown in Exhibits 8 and 9. As seen from these exhibits, a complex is defined by specifying the numbers of the various force elements associated with it, and the resources for the defense of a complex are assumed to be drawn from these elements. Similarly, resources for offensive actions are assumed to be drawn from the elements of a particular complex, with another complex being the target of the action. Complexes are defined as collections of elements, which are defined in computer table 1. Units are composed of collections of elements taken from the complexes they operate from. For all Blue complexes with aviation assets, including land bases, at least one unit of CV must be specified to guarantee operational capabilities as noted in computer table 7. Conversely, all Red complexes having aviation assets, including ships, must have at least one unit of air base.

EXHIBIT 8

BLUE FORCE COMPLEXES												
BLUE COMPLEX	ELEMENTS						CHARACTERISTICS					
	VA	AW	VF	VF	CV	SUP	DLI	SURV	FRACT	CYCL	REPL	REPL
	LO	HI	LO	HI	SHIPS	LNCH	RANGE	A/C	TIME	TIME	OPS	
						TIME	FACT	SHEL				
						(MIN)				DAYS	DAYS	
CTFRC	-	48	24	0	48	21	121	.51	11	.3	41	1150
	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-

EXHIBIT 9

RED FORCE COMPLEXES													
RED COMPLEX	ELEMENTS						CHARACTERISTICS						
	BR	BR	VBF	VFI	SSM	AIR	SAM	SLY	DLI	SURV	FRACT	CYCL	REPL
	LO	HI			SHP	BSE	STE	LNE	LNCH	RANGE	A/C	TIME	OPS
									TIME	FACT	SHEL		
									(MIN)			DAYS	DAYS
ONRAF	-	24	0	72	72	-	11	19	0	.51	11	0	61
ORGAF	-	36	0	72	72	-	11	27	0	.51	11	.3	61
REDCA	-	-	-	-	-	1	-	-	-	-	11	0	41
	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-

In addition to physical location, one may define more than one Blue or Red complex for a number of reasons. For example, Red and Blue forces may be composed of several allies who act independently, or several groups of force elements may initiate or be the targets of attacks independently of each other. For illustration, assume that Red and Orange are allies, as is the case of the problem scenario, with Orange having airbases on the island (ONRODA) and on the mainland. These two airbases may be called ONRAF and ORGAF. Suppose further that Red has a surface missile ship in the area. This ship can be designated as a third Red complex, REDCA. If Blue is operating a carrier task force against Orange, it might be designated as a Blue complex called CTFRC.

In addition to numbers of elements at a complex, several characteristics must be specified. For Blue complexes the time (minutes) between successive DLI launches must be given. For Red complexes the time (minutes) between successive SLI launches must be given. The surveillance range factor, which multiplies the maximum detection range of incoming enemy force units against the complex, must be given. This factor is meant to reflect individual complex surveillance capabilities. For example, in Exhibit 2 the maximum detection range for ALPHA A unit is given as 100 nmi, while in Exhibit 9 the surveillance range factor for ONRAF is given as 1. This means that all ALPHA A units attacking ONRAF will be detected at 100 nmi. The number of sheltered aircraft assigned to each complex must be specified. Sheltered aircraft are not subject to air-to-surface attrition by enemy raids. If a complex, such as a carrier, operates in a replenishment cycle, the length of the cycle is specified, in days as well as the number of days of the cycle during which replenishment is conducted. The maximum percentage of normal operations that can be conducted during replenishment is also specified. Operations during replenishment can be restricted to defensive operations only, by including a "D" after the percentage of normal operations.

EXHIBIT 10

MISCELLANEOUS INPUTS	
MAXIMUM NUMBER OF DAYS THE BATTLE IS TO RUN(LESS THAN 11)	10
MISSION TIME FOR LONG RANGE MISSIONS(NO. OF 3HR TIME STEPS)	0
MISSION TIME FOR SHORT RANGE MISSIONS(NO. OF 3HR TIME STEPS)	0
NO.OF OPERATIONS PER AIRCRAFT LANDING	2
NO.OF OPERATIONS PER AIRCRAFT LAUNCH	1
FRACTION OF MAX UNIT DETECTION RANGE REALIZED IN BAD WEATHER	.5
DELAY IN MINUTES BEFORE FIRST SLI/DLI CAN BE LAUNCHED	3
DO YOU WANT LONG RANGE FORCE UNIT DEFAULT IN EFFECT(1=YES,0=NO)	1
DO YOU WANT BAD WEATHER FORCE UNIT DEFAULT IN EFFECT(1=YES,0=NO)	1
STOP BTLE WHEN ALL MISSIONS COMPLETE(ELSE OFFENSE ONLY)(1=Y,0=N)	0

Miscellaneous Input--In addition to the complex definitions, scenario data include several miscellaneous inputs. These inputs, which are shown in Exhibit 10 are fairly self-explanatory. A mission time of "0" indicates that aircraft will be recovered in the same 3-hour period in which they are launched.

The bottom line refers to a stop criterion for the battle. If a "1" is entered, all missions including defensive missions must go to their completion by either side to stop the battle. If a "0" is entered, only offensive missions, as specified in the respective operations plan computer tables must be completed. In either case a maximum of 10 days of action is generated. Defensive missions are those that employ VFCAP, DLI, or SUCAP units for Blue and SLI units for Red.

3. COA Data

COA data are expected to be of major concern to the user in solving decision problems. These data provide the flexibility for describing various COAs such as timing, use of assets, force position, weather, and threat action.

Operations Plans--The timing, use of assets, and threat action are specified in the Blue and Red operations plan computer tables, shown in Exhibits 11 and 12. In the operations plan table, each offensive and defensive mission is defined by specifying a mission name, priority, origin complex, target complex, start and stop mission times, type of unit, desired number of units, minimum number of units, and number of ready units. Each force can have as many as 17 missions.

The mission name is arbitrarily assigned to identify each mission and may be the same as a unit. The priority of the mission indicates the preferred order for assigning elements to perform missions, with priority 1 missions allocated resources first. Missions may be assigned priorities

EXHIBIT 11

BLUE OPERATIONS PLANS

MISSION	P	ORIGIN COMPLX	TARGET COMPLX	START	STOP	MISSION TIMES	UNIT TYPE	DES SUB	MIN UNT	IREV UNT	UNIT UNT
-ESCR1	-12	CTFRC	ONRAF	DAY1	DAY1	1234	ESCORIA	A	4	2	
-STRK1	-12	CTFRC	ONRAF	DAY1	DAY1	1234	ALPHA	A	6	4	
-ESCR3	-12	CTFRC	ONRAF	DAY2	DDA75	13	ESCORIA	A	4	2	
-STRK3	-12	CTFRC	ONRAF	DAY2	DDA75	13	ALPHA	A	6	4	
-ESCR4	-12	CTFRC	ONRAF	DDA75	DDA90	13	ESCORIA	A	2	2	
-CLEAN	-12	CTFRC	ONRAF	DDA75	DDA90	13	ALPHA	B	6	4	
-STRK2	-12	CTFRC	ONRAF	DAY1	DDA75	6	ALPHA	C	2	1	
-	-	-	-	-	-	-	-	-	-	-	
-CAP1	-11	CTFRC		DAY1	ENDSTRK3	12345678	VFCAP		3	1	
-DLI1	-11	CTFRC		DAY1	ENDSTRK3	23	DLI		1	1	
-CAP2	-11	CTFRC		ENDSTRK3	ENDCLEAN	12345678	VFCAP		1	1	
-DLT2	-11	CTFRC		ENDSTRK3	ENDCLEAN	23	DLI		1	1	
-SUCAP	-11	CTFRC	REDCA	DAY1	DOS100	1234	SUCAP		1	1	
-	-	-	-	-	-	-	-	-	-	-	
-	-	-	-	-	-	-	-	-	-	-	
-	-	-	-	-	-	-	-	-	-	-	

EXHIBIT 12

RED OPERATIONS PLANS

MISSION	P	ORIGIN COMPLX	TARGET COMPLX	START	STOP	MISSION TIMES	UNIT TYPE	DES SUB	MIN UNT	IREV UNT	UNIT UNT
-VBFHT	-11	ONRAF	CTFRC	DAY2	DOS50	23	VBF	A	2	1	
-FREE	-12	URGAF	CTFRC	DAY1	DOS50	23	FREE	A	3	2	
-SLI1	-11	ONRAF		DAY1		1234	SLI		16	1	
-SSMHT	-11	REDCA	CTFRC	DAY2	DOS50	23	SSM		2	1	
-	-	-	-	-	-	-	-	-	-	-	
-	-	-	-	-	-	-	-	-	-	-	
-	-	-	-	-	-	-	-	-	-	-	
-	-	-	-	-	-	-	-	-	-	-	
-	-	-	-	-	-	-	-	-	-	-	
-	-	-	-	-	-	-	-	-	-	-	
-	-	-	-	-	-	-	-	-	-	-	
-	-	-	-	-	-	-	-	-	-	-	
-	-	-	-	-	-	-	-	-	-	-	

0 through 9. Priority 1 and 2 missions that cannot be filled in a particular period are rescheduled for a later period the same day. No missions are rescheduled for following days, however. No mission having a priority of 3 or greater will be rescheduled if unfilled. If a mission has a priority 0, it is not scheduled. Thus, assigning a mission a priority of 0 is a convenient way to eliminate a mission for a particular run.

The start and stop times for a mission can be specified in several ways. One way is to enter DAY, followed by the number of the day in the battle when the subject mission is to be initiated or terminated. A second way is to enter END, followed by the name of a previous mission. In this case, for start times, the mission will begin the day after the previous mission terminates. When END is used as a stop time, the mission will be terminated the day the previous mission is terminated. A third way of designating start and stop times is to enter DOA, DDA, DOS, DDS, or DLS (meaning destroy offensive air, destroy defensive air, destroy offensive surface, destroy defensive surface, or destroy logistics support, respectively), and then to enter the level of destruction of the associated elements at the target complex desired before the mission is to be initiated or terminated. For example, the DDA75 stop time for mission STRK3 in Exhibit 11 means that mission STRK3 is to be terminated when 75% of the defensive air elements at the target complex, ONRAF, have been destroyed. A blank entry for a stop time indicates that the mission is to be continued indefinitely.

Each mission must have an origin complex specified from which all units that comprise the mission must be formed. All offensive missions must be assigned a target complex. A target complex needs be assigned only to those defensive missions with a start or stop time given as one of the following contingencies: DOA, DDA, DOS, DDS, or DLS. The target complex specified for a defensive mission refers to the complex to which the start or stop contingency applies. The complexes named as the origin

and target of a mission must be defined in the scenario data (computer tables 8 and 9).

For computing battle results, the day is divided into eight, 3-hour periods. For mission times, the user specifies the period in which each mission is to be performed. The type and subtype of unit that is to perform each mission must also be specified by name. Force unit names must have been defined in the background data (in both force unit definition and engagement statistics computer tables). The desired units indicate the number of units the user desires to have assigned at each repetition of the mission. The minimum number of units designates the smallest number of force units to be assigned at each repetition of the mission. If resources are insufficient for assigning the minimum number of units to a mission, the mission is not flown during that time step. Ready units, which are used for defensive missions, are those units reserved to replace resources that become exhausted during a period such as in DLI or VFCAP.

Force Position and Weather--COA data that indicate force position and weather parameters are provided in computer tables in the form shown in Exhibits 13 and 14, respectively. The relative position (LONG or SHORT) between each pair of Red and Blue complexes must be specified. If no relative position is specified, LONG is assumed. The use of (LONG or SHORT) relative positions is an option to allow the attacking TFC to employ his forces to fullest advantage. If the TFC (user) is attacking an enemy air-base, he may launch his attack from a LONG position in order to use his aircraft range advantage and decrease the threat against him by any lesser range enemy aircraft.

The days of battle when weather (GOOD or BAD) changes occur are specified in computer table 14. The weather remains the same between days of change and is assumed to be GOOD if no weather is otherwise specified. If one or more days of battle are changed to BAD by the TFC, alternate

EXHIBIT 13

BLUE		RELATIVE COMPLEX POSITIONS							
RED	CIPRC								
ONRAF	LONG								
ORGAF	LONG								
REDCA	SHORT								

EXHIBIT 14

INITIATING WX DAYS									
DAY	1								
WX	GOOD								

force units are employed, thereby allowing the TFC to examine bad weather contingencies.

C. Output

Three types of battle outcomes are generated by SOC: mission accomplishment results, battle attrition results, and an aircraft expenditure summary. Blue and Red mission results are given in Exhibits 15 and 16, and basically show the operations plans specified by the user. The start and stop times, which became activated because of the termination of missions or destruction of targets, are displayed by day of occurrence. Otherwise, the data element does not change. The total number of units requested for each mission during the battle is shown, together with units actually engaged. If a default unit is activated, the number of units requested and engaged contributes to those statistics. As with all the output computer tables, a user-supplied run identification (i.e., ONRODA) is printed out along with the length of the battle (i.e., 2 days).

Blue and Red battle attrition results are shown in Exhibits 17 and 18. For each complex, the number of elements of each type attrited, together with the total elements assigned, is shown. The number of airbase and CV elements at a complex shown as lost does not include those damaged and subsequently repaired.

An aircraft expenditure summary is also generated as shown in Exhibit 19. The daily cumulative expenditure of Blue aviation sorties and aircraft attrition due to air-to-air, ground-to-air (SAM), and air-to-ground (enemy attack) engagements is shown. Red is summarized by offensive and defensive aircraft categories as categorized by computer table 1. The final cumulative attrition results should correlate with computer tables 17 and 18. In practice, however, they may deviate slightly because of round off error in computer table 19. Thus, for example we see a total of 5 Blue AW losses in Exhibit 19 and 6 Blue AW losses in Exhibit 17.

EXHIBIT 15

ONRODA						BLUE MISSION ACCOMPLISHMENT RESULTS				DAYS= 2			
MISSION	P	COMPLX	COMPLX	START	STOP	MISSION	UNIT	DES	UNT	UNT	REQ	ENG	
ESCR1	12	CTFRC	ONRAF	DAY	1 DAY	11	1234 ESCORIA	4	16	16			
STRK1	12	CTFRC	ONRAF	DAY	1 DAY	11	1234 ALPHAIA	6	24	24			
ESCR3	12	CTFRC	ONRAF	DAY	2 DAY	11	13 ESCORIA	4	0	0			
STRK3	12	CTFRC	ONRAF	DAY	2 DAY	11	13 ALPHAIA	6	0	0			
ESCR4	12	CTFRC	ONRAF	DAY	2 DAY	21	13 ESCORIA	2	4	4			
CLEAN	12	CTFRC	ONRAF	DAY	2 DAY	21	13 ALPHAIB	6	12	12			
STRK2	12	CTFRC	ONRAF	DAY	1 DAY	21	6 ALPHAIC	2	4	4			
CAP1	11	CTFRC		DAY	1 DAY	1 12345678 VFCAP	3	24	24				
DLI1	11	CTFRC		DAY	1 DAY	11	23 DLI	1	2	2			
CAP2	11	CTFRC		DAY	2 DAY	2 12345678 VFCAP	1	8	8				
DLI2	11	CTFRC		DAY	2 DAY	21	23 DLI	1	2	2			
SUCAP	11	CTFRC	REDCA	DAY	1 DOS	100	1234 SUCAP	1	8	8			

EXHIBIT 16

ONRODA						RED MISSION ACCOMPLISHMENT RESULTS				DAYS= 2			
MISSION	P	COMPLX	COMPLX	START	STOP	MISSION	UNIT	DES	UNT	UNT	REQ	ENG	
VBFHT	11	ONRAF	CTFRC	DAY	2 DOS	50	23 VBF	1	4	2			
FREE	12	ORGAF	CTFRC	DAY	1 DOS	50	23 FREE	1	3	12	9		
SLI1	11	CNRAF		DAY	1		1234 SLI	16	128	62			
SSMHT	11	REDCA	CTFRC	DAY	2 DOS	50	23 SSM	2	4	1			

EXHIBIT 17

ONRODA		BLUE CMPLX BATTLE ATTRITION RES. DAYS= 2					
BLUE COMPLEX		ELEMENTS ATTRITED / TOTAL ELEMENTS AT COMPLEX					
		VA	AW	VF LO	VF HI	CV	SUPSHIP
CTFRC		19.	6.	0.	22.	0.0	0.0
		48.	24.	0.	48.	2.0	12.0

EXHIBIT 18

ONRODA		RED COMPLEX BATTLE ATTRITION RESULTS						DAYS= 2
RED COMPLEX		ELEMENTS ATTRITED / TOTAL ELEMENTS AT COMPLEX						SUPLINE
		BR LO	BR HI	VRF	VFI	ISSMSHIP	AIRBASE	
ONRAF		24.	0.	72.	72.	0.0	0.0	18.0
		24.	0.	72.	72.	0.0	1.0	18.0
ORGAF		36.	0.	36.	0.	0.0	0.0	0.0
		36.	0.	72.	72.	0.0	1.0	27.0
REDCA		0.	0.	0.	0.	0.6	0.0	0.0
		0.	0.	0.	0.	1.0	0.0	0.0

EXHIBIT 19

ONRODA		AIRCRAFT EXPENDITURE SUMMARY										DAYS= 2		
		BLUE VA	BLUE AW	BLUE VF-LO	BLUE VF-HI	CUM RED								
DISCUM	CUM LOSS TO	CUM	CUM LOSS TO	CUM	CUM LOSS TO	CUM	CUM LOSS TO	CUM	CUM LOSS TO	CUM	LOSSES			
AISOR	-----	ISOR	-----	SOR	-----	SOR	-----	SOR	-----	SOR	-----			
YITIES	A-A G-A A-G TIES	A-A G-A A-G TIES	A-A G-A A-G TIES	A-A G-A A-G TIES	A-A G-A A-G TIES	A-A G-A A-G TIES	A-A G-A A-G TIES	A-A G-A A-G TIES	A-A G-A A-G TIES	A-A G-A A-G TIES	DAT			
1	76 11 7 0 32 3 2 0 0 0 0 0 68 17 2 0 87 71													
2	116 11 8 0 52 3 2 0 0 0 0 0 96 19 2 0 168 72													

III SOC BATTLE LOGIC

A. General Description

The logic in SOC's computational algorithm is straightforward. The algorithm basically carries out the operations plans specified in the COA data, using the scenario and background data for information on force composition, effectiveness, and availability. Numerical computations are at the level of aggregation of the data in the aggregate data base. Thus, credible estimates of battle outcomes, at the level of concern of the user, are generated.

A macroflow chart of the SOC algorithm is shown in Figure 2. Each day of the battle is simulated successively. For each day, the day's missions are scheduled in advance. The activities for each time step of the day are then performed. In each time step, first the schedule is modified in regard to past action (i.e., aircraft loss). Next, aircraft returning from missions begun in previous time steps are recovered. After this recovery has taken place, the aircraft to be used in missions begun during the current time step are launched. Next, forces engage followed by the recovery of surviving aircraft that performed missions during the current time step. After the day's activities the algorithm either ends or the next day's activities are initiated.

B. NEWRUN

At the beginning of each run an initialization process takes place. After this initialization has taken place, each day of the battle is simulated.

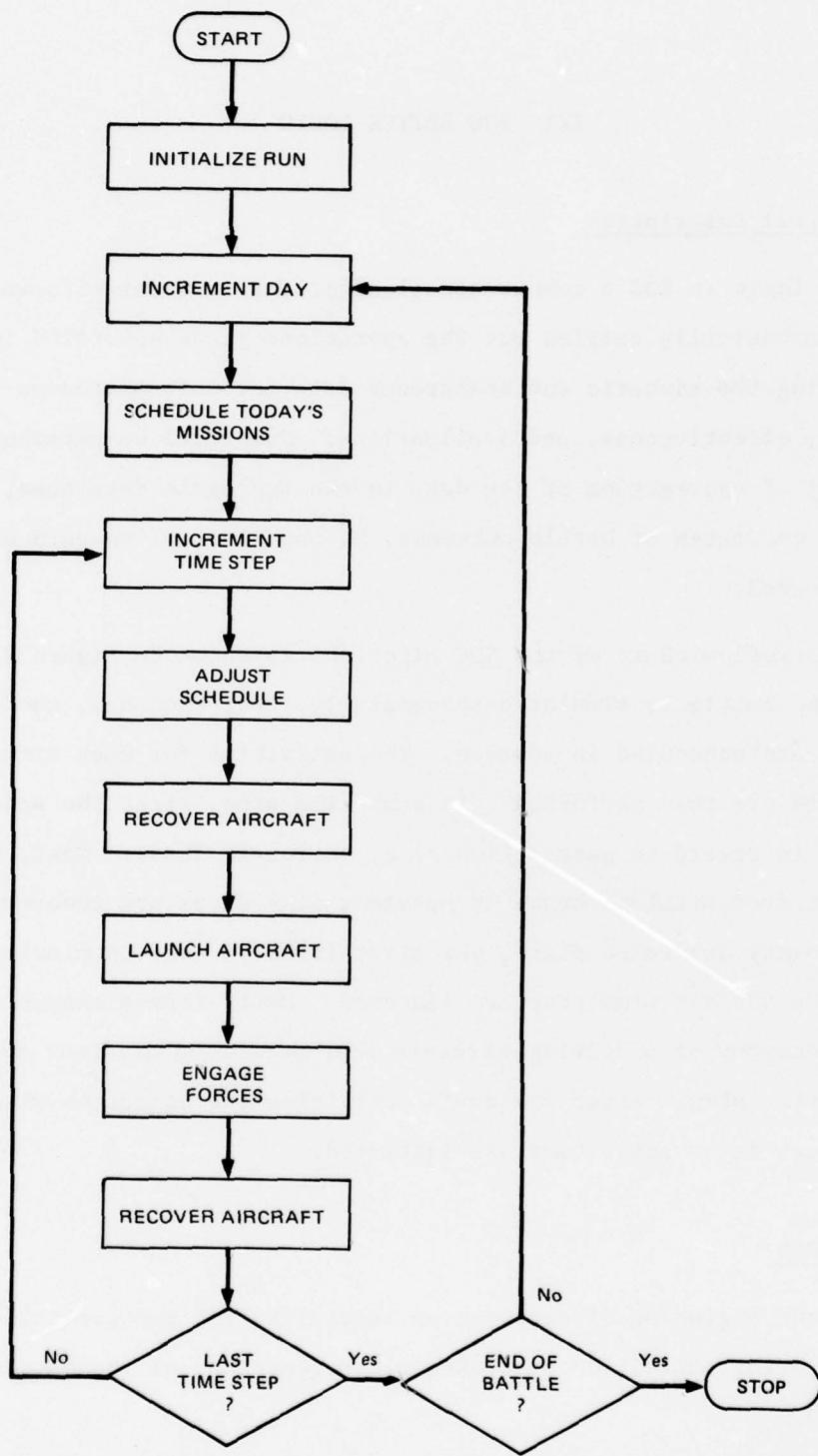


FIGURE 2. MACROFLOW CHART OF SOC COMPUTATIONAL ALGORITHM

C. NEWDAY

At the beginning of each new day, initialization for the day takes place and the live and killed elements at each complex are accounted for. In addition, mission start and stop times are updated. Depending on how the start times are specified in the input tables, mission start times remain the same, are set to the day following the completion of a previous mission, or are assigned to the day after a specified destruction level has been reached at a given complex. Likewise, depending on information in the input tables, stop times remain the same, are set to the day a previous mission ends, or are assigned to the day when a specified destruction level is reached at a given complex.

D. SCHEDULE

Each day is broken into eight 3-hour time periods that are sequentially simulated. At the beginning of each new day, the day's missions for both Blue and Red forces are scheduled. A mission is not scheduled if the current day precedes the start date of a mission or occurs after the stop date of the mission. If a mission is assigned a priority of zero or there is less than 1 unit requested by a mission, the mission is not scheduled. If a complex is designated as able to perform only defensive missions during replenishment and is undergoing replenishment during the current days, no offensive missions originating from that complex are scheduled. Note that missions using type SUCAP, VFCAP, or DLI units for Blue or SLI units for Red are defensive missions. All other type missions are offensive missions.

Under poor weather conditions, any mission requesting good weather units, for which no bad weather default units are specified, is not scheduled. Similarly, any long-range mission that requests short-range units that have no long-range default units specified, is not scheduled. If a long-range

mission is requested in bad weather, a long-range poor-weather unit must be specified in the input table as able to perform the mission; otherwise, the mission is cancelled.

At the beginning of each time step the ability of each complex to perform operations is increased according to the repairs that can be made in one time period. The level of operation of each complex can never be brought above the maximum capacity specified in the input tables. Additionally, the number of elements available at each complex is increased by making elements available, that have been rearmed and refueled.

With each time step, the schedule of missions is modified. Priority 1 and 2 missions unfilled in the previous time step of the current day are rescheduled for the current time step, provided an identical mission is not already scheduled. No rescheduling of missions from previous days occurs.

Once all missions for the remainder of the day have been updated, a check is made to determine if sortie rates will be exceeded. The missions are checked in order of priority, with priority 1 missions checked first. Missions of equal priority are checked starting with the current time step and proceeding sequentially throughout the day. Missions of the same priority, occurring during the same time step, are checked according to the order they occur in the operation plan tables. Priority 1 missions will be cancelled only if the 30-day maximum sortie rate is predicted to be exceeded. Priority 2 missions will be cancelled for the same reason and also if the surge sortie rate is predicted to be exceeded. Priority 3 through 9 missions will be cancelled for the same reasons and if the normal sortie rate is estimated to be exceeded. Note that any mission cancelled at this point is not rescheduled and that each mission checked adds to the estimated number of sorties for the aircraft involved. Thus, if priority 1 and 2 missions exceed the normal sortie rate, but not the surge sortie rate of an aircraft, those missions will remain scheduled. However, all missions of priorities 3 through 9 using that type of aircraft that day will be cancelled.

E. RECOVER

Once desired missions for both Blue and Red have been scheduled for the current time period, aircraft at all complexes are recovered one at a time from previous missions without preference for aircraft type or mission. When the number of operations available at a complex is reached, recovery ceases, and unrecovered aircraft wait until the next step to be recovered.

F. LAUNCH

After the recovery of aircraft has taken place, an attempt to fill missions and launch aircraft is made. Missions are filled in order of priority. Missions of the same priority are filled in the order they appear in the operation plans table. Each mission is assigned as many requested units as possible, within the maximum requested, subject to unit availability, the availability of launch operations, and sortie restrictions. Priority 1 and 2 missions can be flown, provided the surge sortie rate is not exceeded. Priority 3 through 9 missions can be flown only if the normal sortie rate is not exceeded. Under no circumstances can the 30-day sortie rate be exceeded.

A mission is flown only if the minimum number of specified units can be assigned to it. A mission that requests ready units (i.e., units reserved for launch as needed for replacements to a mission such as DLI) is assigned as many ready units as requested or possible if the mission has enough aircraft assigned to it to be flown. A mission need not have any ready units assigned to it to be flown. Aircraft assigned a ready status are unavailable for other missions. It is assumed that all aircraft assigned to missions in a given period are launched at the same time.

G. ENGAGE

The battle losses incurred when one side strikes the other are computed by using the engagement statistics provided in the SOC aggregate data base. The aircraft or surface-to-surface missiles of an incoming strike are subject of attrition by air and surface defenses before they can inflict air-to-surface damage. The defending units are those associated with the target complex of the strike.

Offensive strikes are considered one at a time. First, Blue strikes are considered according to the order they appear in the operation plans table. Next, each Red strike is considered according to the order in Red's operation plans table.

An incoming strike is first engaged by any airborne VFCAP at the target complex. Then, any DLI or SLI may engage the strike. Ready units are used to refill the VFCAP and DLI or SLI positions vacated by the engaging defenders. The refilled VFCAP are then used to attack the incoming strike with subsequent defense by the refilled DLI or SLI. This refilling and attacking sequence continues. There are, however, constraints on the use of the air defenders assigned to VFCAP, DLI, and SLI missions to attack incoming strikes. First, no more air defenders are engaged than necessary to annihilate an incoming raid. Second--based on the range at which the incoming strike force is detected, the speed of the incoming strike, the delay until the first defender (on the deck or ground) can be launched, and the time between launches--, the number of defenders that can be engaged with the strike before its attack on the target complex is limited.

The losses incurred by the incoming strike and the air defenders during the air-to-air combat are based on the engagement statistics supplied in computer tables 4 and 5. The losses are calculated by postulating a simultaneous exchange between the forces. These losses are

assumed to be a linear function of the number of force units in the strike and the number of defenders engaging the strike.

After the air-to-air battle, any surface-to-air defensive elements are assumed to have a shot at the remaining attackers. Surface-to-air capabilities may be specified by the user for Blue carriers and support ships and for Red SAM-sites and supply lines.

The portion of strike force that penetrates the defenses inflicts air-to-surface losses on the elements at the target complex. Only the parked, unsheltered aircraft (as specified by the user in the scenario data) are subject to destruction. The number of parked aircraft of each type that is destroyed is proportional to the ratio of parked aircraft of each type to the total parked aircraft.

Some logic is peculiar to the Red SSM attack. If an SSM attack is imminent, the program assumes that airborne SUCAP (associated with the Blue complex to be attacked) attack the SSM-ships from which the SSM attack is to be mounted. Only the portion of the ships that survive this attack is assumed to launch their missiles (at a rate of 8 missiles per undamaged ship). Ready units associated with a SUCAP mission are launched and attack the remaining SSM-ships after the ship's missiles have been launched. All attacking SUCAP are subject to attrition by any air-to-air and surface-to-air defenses associated with the force complex to which the SSM-ships belong. The missiles launched are subject to air-to-air and surface-to-air attrition by the defenses of the Blue complex under attack. The sequencing of the engagements of the SUCAP and SSM missiles with air and ground defenses and of the air-to-surface attacks is treated as previously described for offensive strikes.

The aircraft, both defensive and offensive, that survive are scheduled to land either in the current time step or some future time step. The

scheduled landing at the home complex is based on the user-specified (computer table 10) mission times for LONG and SHORT missions.* VFCAP, DLI, and SLI are scheduled to land at the end of the time step in which they were launched. Aircraft ready to land are recovered at the end of each time step, subject to the operations available at carriers and airbases.

All attrition computations are assumed to be linear functions of the numbers of units of force elements engaging. The losses at each of the force elements at each of the force complexes are computed and used to maintain an updated inventory of Blue and Red assets. These results are needed to determine the capability to perform future missions; they are also presented to the user in the battle results computer tables.

*If aircraft are scheduled to land at the end of the time step in which they are launched, the mission time is specified as zero time steps.

IV SOC MAN-MACHINE INTERACTION

A. Interactive Commands

Given a computer terminal with a CRT and a printer, the communicator portion of SOC allows the user to display and modify all the data in the data base, exercise the computational algorithm, and obtain printed copy of any of the data and results. The communicator commands follow:

1. LOAD INPUT

This command is used to retrieve a data base that has previously been created and saved.

2. SAVE INPUT

This command saves the current data for future use.

3. DISPLAY/INPUT TABLE

This command displays a particular table with its associated data, if any. Once a table is displayed, it may be edited. During editing, either current data may be changed, or a new data base can be created if the current data base is empty.

4. PRINT TABLE

This command is used to print tables on a line printer.

5. RUN

This command causes the execution of the battle logic using the currently loaded data base.

6. STOP

This command terminates SOC and returns control to the operating system.

7. LIST TABLES

This command lists the titles of all the tables on the CRT screen.

8. EXPLAIN ABOVE FUNCTIONS

This command provides the user with explanations of each of the other commands.

B. Command Use

A typical session for using SOC will begin by turning on the console and following a standard procedure for getting on-line to the computer. When SOC is loaded and entered, queries and instructions that require user interaction will appear on the screen. The questions and instructions posed to the user are self-explanatory. They begin with a choice of a CRT that SOC runs with. The user corresponds with a number as indicated to show the terminal type he is using. Only specified types will work with SOC. Quickly thereafter the user is led to the function menu that presents the 8 commands above for him to execute.

If the user wishes to refresh his memory, he can issue the EXPLAIN ABOVE FUNCTIONS command and receive a brief description of SOC communicator functions on the console. The user then depresses the space bar to return to the function menu. The user can then ask for a list of the titles of the aggregate data base computer tables by issuing the LIST TABLES command. If background, scenario, or COA data that were used and saved in a previous session, are of interest, they can be retrieved and placed in the current aggregate data base by issuing the LOAD INPUT command.

If a new data base is desired, no old data base will be loaded but the DISPLAY/INPUT TABLE command should be repeated until all 14 input tables are established. This must be done sequentially from table 1 to table 14 so that error routines can be established to check for logical consistency. When the user executes this command, he is queried for the table of interest. When a table number is specified, the table is displayed on the screen. If there is a current data base, the information associated with the specified table is displayed in addition to the table form. The user can now enter new table data or modify the existing table.

Data are changed or entered by moving the cursor to the appropriate field, using tab, carriage return, and cursor control keys, and then entering the new data. After a table has been filled with new data, the user hits the transmit key (or combination of the CONTROL and Q keys if no such key exists) to initiate error checking. If no errors occur, the table is entered into the current data base and the user is returned to the function menu. If errors occur, indications are given and the table is redisplayed and the editing process repeated. If during the editing process the user wishes to revert to the original table rather than continue editing or he merely wished to display the table, he hits the transmit key immediately after the table is displayed and he is then returned to the function menu. Once a data base has been established as above, or loaded using the LOAD INPUT command, it can be edited in random order.

When the current data base properly describes a COA that the user wants to evaluate, the RUN command is issued. On executing this command, the algorithm computes the battle results using the data in the current first 14 computer tables of the aggregate data base and places the results in the computer tables 15 through 19. Using the DISPLAY/INPUT TABLE command, the user can now have the results displayed on the console. Printouts (hard copy) of any of the data or results tables can be obtained using the PRINT TABLE command.

The user may evaluate a series of alternative COAs by sequentially modifying data tables, running the algorithm, and then displaying and/or printing the results. After all COAs have been evaluated, the user may issue the SAVE INPUT command. This will transfer the current contents of the aggregate data base to computer storage for future use. The STOP command terminates the session.

Appendix A

ONRODA PROBLEM SCENARIO

Appendix A

ONRODA PROBLEM SCENARIO

The problem scenario and resulting concept of operations described in this appendix have been extracted from the following document: "ONRODA Warfare Scenario," Research Memorandum NWRC-RM-83, Stanford Research Institute (June 1975). This document can be used to augment the information presented in this appendix if it is desired; however, it is broader in its scope and a few of its details are different. Every effort has been made in this appendix to present the major elements of the problem as simply as possible. A synopsis of the scenario follows.

Synopsis

Grey and Orange (see Figure A-1) have been ideologically opposed and hostile toward each other for a long time. Orange has supported rebel activities in Grey. ONRODA Island has been politically aligned with Grey but has a significant segment of Orange sympathizers. Grey's military capability has diminished and Orange responds by more active support of the rebels in Grey and by capturing ONRODA Island. Blue has previously indicated that this was an unacceptable action, supported Grey's appeal to the UN, and asked for congressional approval for unilateral support of Grey if favorable UN reaction was not immediate. At the same time, Blue orders the fleet to prepare to stabilize the military situation in the area and prevent Orange from using ONRODA Island as a base for future military action against Grey. Red, who has supplied Orange with most of her combat systems, also has a naval force in the area. A Blue carrier task force is formed and given the mission: "When directed, begin operations to neutralize Orange forces and facilities on ONRODA Island in order

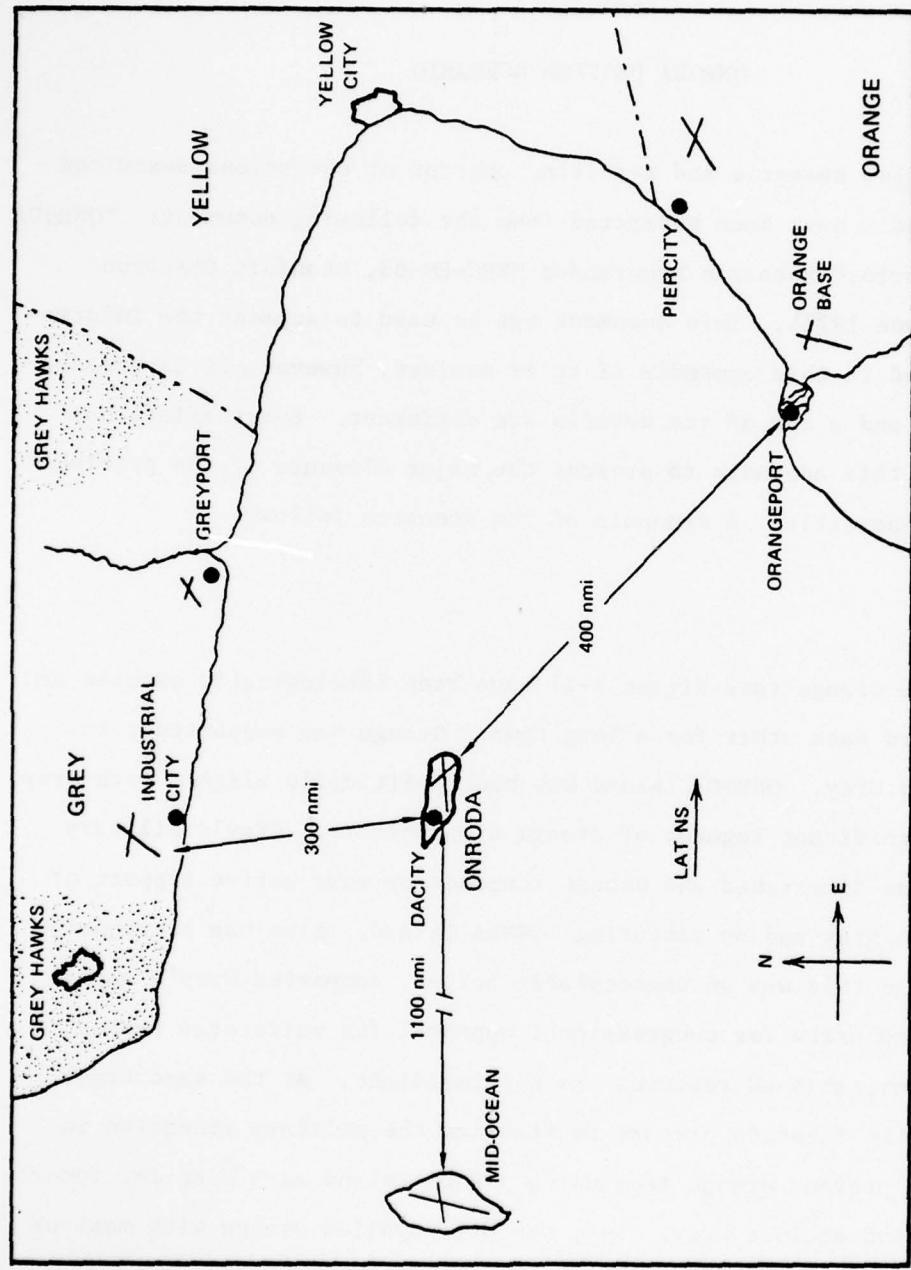


FIGURE A-1 AREA MAP

to defend Grey. Do not attack targets on Orange mainland or in Orange ports. Take defensive measures to protect your force from Orange or Red retaliations."

The Blue task force is divided into two carrier groups and consists of the elements listed in Table A-1. The enemy forces in the area are listed in Table A-2.

Concept of Operations

The tentative choices for Blue COAs are:

- Neutralizing enemy forces by preemptive or reactive air strike*
- Neutralizing enemy forces by air blockade
- Employing both reactive air strike and air blockade to neutralize enemy forces.

After much consideration, the TFC selects the third alternative and orders that a concept of operations be developed for the reactive air strike and subsequent air blockade. The concept of operations follows.

To defend Grey, the Blue task force will conduct air strikes, when directed, against Orange forces on ONRODA Island. This operation will defend Grey from Orange air attacks, in particular from ONRODA Island. The physical targets are the Orange combat aircraft and support facilities on ONRODA Island. Orange forces on the Orange mainland or in Orange ports cannot be attacked. It is expected that no nuclear weapons will be used.

Red units may be sighted in the area. There is a low probability that Red will initiate hostile action toward Blue, but Red is expected

* Note: The different elements in a preemptive air strike, as compared to a reactive air strike, might be considered to be more surprise to the enemy, stronger offensive forces, higher mission accomplishment in less time, and fewer defensive forces needed.

Table A-1

BLUE TASK FORCE MAJOR COMBATANTS

<u>CARRIER GROUP ONE</u>	
Type	Class
CV	KITTY HAWK
CG	ALBANY
DDG	CHARLES F. ADAMS
DD	SPRUANCE
DD	GEARING (FRAM I)
DE	KNOX (with BPDMS and LAMPS)
DE	KNOX (with BPDMS and LAMPS)

<u>CARRIER GROUP TWO</u>	
Type	Class
CV	FORRESTAL
CLG	converted CLEVELAND
DLG	LEAHY
DD	SPRUANCE
DD	GEARING (FRAM I)
DD	GEARING (FRAM I)
DE	KNOX (with BPDMS and LAMPS)

<u>AIR WING COMPOSITION (each CV)</u>	
24	F-14
24	A-7E
12	A-6E
8	S-3A
5	E-2C
4	EA-6
4	KA-6
4	SH-3
1	C-1

Table A-2
ENEMY MAJOR COMBATANTS

<u>ORANGE NAVAL FORCES</u>		
<u>Number</u>	<u>Type</u>	<u>Class</u>
4	Destroyer	SKORY
6	Missile Boat	OSA-1
6	Missile Boat	KOMAR
12	Torpedo Boat	P6
2	Minesweeper	T-43
5	Amphibious Craft	VYDRA
2	Submarine	W

<u>RED NAVAL FORCES</u>		
<u>Number</u>	<u>Type</u>	<u>Class</u>
1	Cruiser	KRESTA I
1	Destroyer	KASHIN
2	Submarine	ECHO II

<u>ORANGE AIR FORCES</u>	
<u>at ONRODA</u>	<u>at ORANGE</u>
24 MIG-19	48 MIG-19
72 MIG-21	72 MIG-21
48 SU-7	24 SU-7
24 IL-28	12 IL-28
	24 BADGER A

to harass and surveil individual Blue units. Orange units may also surveil the Blue forces before air strikes begin. Blue forces are ordered not to take preemptive hostile action against such activities but to defend themselves by return fire if fired upon.

The two carrier groups making up the Blue task force will operate independently, approximately 50 to 100 nmi apart. The strike launch point will be 400 nmi west of ONRODA Island. On the first strike day (S-day) four Alpha strikes will be launched against ONRODA--two strikes from each carrier. Each Alpha strike will be composed of 18 A-7s, 6 A-6s, 10 F-14 escorts, 1 E-2 strike control, 1 EA-6 for EW, and 3 KA-6 refuelers. On each succeeding day (S+1, S+2, etc.) strike carrier duty will be alternated between carriers, with each carrier launching two strikes a day. Additionally, an Intruder strike will be flown by four A-6s plus 2 F-14s at night. The Intruder strike composition will be substituted for the Alpha strikes if the weather turns bad. The off-duty strike carrier will fly the task force defensive sorties. "Clean-up" strikes are desired when the enemy defensive air assets have been reduced 75%.

The defensive carrier will provide fighter aircraft to support three CAP stations during its 24-hour defensive duty. Each carrier group will provide its own ASW daily aircraft requirements. If enemy surface units threaten the task force, a SUCAP (surface cap) of A-7s and A-6s will be called on. It is highly probable that Badger-A bombers escorted by MIG-21s with external tanks may attack the task force from Orange mainland sometime after Blue strikes ONRODA.

After Orange aircraft are neutralized on ONRODA Island, the task force plans to change station to a point midway between ONRODA and Grey from which it will set up an air blockade to protect both Grey from attack and ONRODA from reinforcements.

Appendix B
SOC PROGRAM LISTINGS

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
00057      NIT=14
00058      MAXFLD=16
00059      MAXROW=17
00060      DO 30 I=1,2
00061      DO 30 J=1,MAXCMP
00062      20 ICMPLX(I,J)=NULL
00063      DO 50 I=1,NIT
00064      DO 50 J=1,MAXFLD
00065      DO 50 K=1,MAXROW
00066      50 ITABLE(I,J,K)=NULL
00067      C
00068      C-----PRINT PROGRAM TITLE
00069      WRITE(5,1000)
00070      C
00071      C-----INITIALIZE TERMINAL DATA
00072      CALL INIT
00073      C
00074      C-----OUTPUT GENERAL PREFACE
00075      CALL NWSCPN(0)
00076      WRITE(5,1001)
00077      WRITE(5,1002)
00078      C
00079      C-----MONITOR AND COMPLY WITH TERMINAL DIALOGUE
00080      CALL MONTR(ITABLE)
00081      STOP
00082      C
00083      C-----FORMATS
00084      1000  FORMAT('OSOC PROGRAM VFR(4.0), SRI INTERNATIONAL'//)
00085      1001  FORMAT(69H THE OUTCOME CALCULATOR PRODUCED BY NWRC/SRI IS NOW AT
00086      1OUR DISPOSAL./1H ,
00087      264H THE PURPOSE OF THE OUTCOME CALCULATOR IS TO PROVIDE QUANTITATIV
00088      3E/1H ,
00089      464HESTIMATES OF OUTCOMES FOR NAVAL AIR STRIKE ALTERNATIVE COURSE O
00090      5F/1H ,
00091      617HACTION DECISIONS./1H ,/1H ,/1H )
00092      1002  FORMAT( ' ANSWERS TO QUESTIONS ASKED OF YOU BY THE OUTCOME ',
00093      1 'CALCULATOR',//, ' MUST BE FOLLOWED BY A CARRIAGE RETURN.'//)
00094      END
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
00001 C
00002 C      THIS PROGRAM CONSTITUTES THE INTERACTIVE VERSION
00003 C      OF THE STRIKE OUTCOME CALCULATOR, BUILT FOR THE OMR ODA PROGRAM.
00004 C
00005 C      NRRC/SRI  1976,1977
00006 C
00007 C      DISK20=FILE TO READ PREVIOUSLY SAVED DATA
00008 C      DISK21=FILE TO WRITE DATA SAVED FROM THIS RUN
00009 C      DISK23=PFERMFILE CONTAINING FORMS OUTLINES
00010 C
00011 C      DOUBLE PRECISION IUTYPE,IETYPE,IUSUP,IESUB,ICMPLX,MISS
00012 C      DOUBLE PRECISION ITABLE,ITEMP,ISTORE,NULL
00013 C      COMMON /IFACE/ IUTYPE(2,15),IETYPE(2,15),IUSUB(2,15,15),
00014 C                  +           IESUB(2,15,15),ICMPLX(2,8),MISS(2,17)
00015 C      COMMON /SPECs/ ITABER, ITABLR, ITABNR, ITABNE, IFlds(14,15),
00016 C      1           IBLNK(15,17), IFORMT(2,19)
00017 C      COMMON /MAX/ MAXCMP,MAXUNT,MAXMIS,NIT,MAXFLD,MAXPOW,NULL
00018 C      DIMENSION ITABLE(14,16,17),ITEMP(19,17)
00019 C
00020 C-----THIS IS AN ERROR CHECKING ACTION TABLE. THE FOLLOWING
00021 C      IS A DESCRIPTION OF THE CURRENT ENTRIES.
00022 C
00023 C      VALUE    ERROR ACTION
00024 C      0        CHECK FOR ALPHA
00025 C      1        CHECK FOR NUMERIC
00026 C      2        CHECK FORCE UNIT DEFINITION
00027 C      3        CHECK FOR COMPLEX ON LIST
00028 C      4        CHECK FOR MISSION PRESENTS ON APPROPRIATE LIST(NOT IMPL)
00029 C      5        CHECK FOR "LONG" OR "SHORT"
00030 C      6        CHECK FOR "COOP" OR "PAT"
00031 C      7        CHECK REPL OPS
00032 C      8        CHECK START/STOP FIELDS
00033 C
00034 C      DATA(IFlds(01,K),K=1,15) /0,0,C,0,0, 0,0,0,0,0, 0,0,0,0,C/
00035 C      DATA(IFlds(02,K),K=1,15) /0,0,1,1,1, 1,1,5,2,6, 2,1,1,0,0/
00036 C      DATA(IFlds(03,K),K=1,15) /0,0,1,1,1, 1,1,5,2,6, 2,1,1,0,0/
00037 C      DATA(IFlds(04,K),K=1,15) /0,0,1,1,1, 1,1,1,1,1, 1,1,1,0,0/
00038 C      DATA(IFlds(05,K),K=1,15) /0,0,1,1,1, 1,1,1,1,1, 1,0,0,0,0/
00039 C      DATA(IFlds(06,K),K=1,15) /1,1,1,0,0, 0,0,0,0,0, 0,0,0,0,0/
00040 C      DATA(IFlds(07,K),K=1,15) /1,1,1,1,0, 0,0,0,0,0, 0,0,0,0,0/
00041 C      DATA(IFlds(08,K),K=1,15) /0,1,1,1,1, 1,1,1,1,1, 1,1,7,0,0/
00042 C      DATA(IFlds(09,K),K=1,15) /0,1,1,1,1, 1,1,1,1,1, 1,1,1,1,7/
00043 C      DATA(IFlds(10,K),K=1,15) /1,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0/
00044 C      DATA(IFlds(11,K),K=1,15) /0,1,3,3,8, 8,1,0,2,1, 1,1,0,0,0/
00045 C      DATA(IFlds(12,K),K=1,15) /0,1,3,3,9, 8,1,0,2,1, 1,1,0,0,0/
00046 C      DATA(IFlds(13,K),K=1,15) /5,5,5,5,5, 5,5,5,0,0, 0,0,0,0,0/
00047 C      DATA(IFlds(14,K),K=1,15) /6,6,6,6,6, 6,6,6,6,6, 0,0,0,0,0/
00048 C
00049 C
00050 C      DATA NULL/6H      /
00051 C
00052 C-----INITIALIZE TABLE DATA
00053 C      CALL ERSET(0)
00054 C      MAXCMP=8
00055 C      MAXUNT=15
00056 C      MAXMIS=17
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
00001      SUBROUTINE MNTR(ITABLE)
00002      C
00003      C THIS ROUTINE MONITORS INTERACTIVE VERSION OF THE OUTCOME CALCULATOR
00004      C
00005      DOUBLE PRECISION NULL,ITABLE,ITFMP,CDATE
00006      DOUBLE PRECISON IUTYPE,IETYPE,TUSUP,IESUP,ICMPLX,MISS
00007      DIMENSION ITABLE(14,16,17),ITEMP(19,17)
00008      COMMON/MODS/IDRUN,IDAY
00009      DOUBLE PRECISION IDRUN
00010      COMMON /IFACE/ IUTYPE(2,15),IETYPE(2,15),IUSUR(2,15,15),
00011      +                  IESUB(2,15,15),ICMPLX(?,8),MISS(?,17)
00012      COMMON /SPECS/ ITABFR, ITABLR, ITABNR, ITABNF, IFLDS(14,15),
00013      1                  TBLNK(15,17), IFORMT(2,19)
00014      COMMON /MAX/ MAXCMP,MAXUNT,MAXMIS,NIT,MAXFLD,MAXROW,NULL
00015      INTEGER LINES(15)
00016      C
00017      C ATTACH PERM FILE WITH FORMS LOGICAL UNIT 23
00018      C
00019      CALL DEFINE FILE(23,80,TIN,"SRIFORMS.DAT",0,0)
00020      C
00021      C
00022      IDRUN=NULL
00023      WRITE(5,1002)
00024      WRITE(5,1001)
00025      CALL DUMRD
00026      C
00027      C FUNCTION MENU CONTROL POINT
00028      C
00029      100 CONTINUE
00030      CALL NWSCP(0)
00031      WRITE(5,1003)
00032      WRITE(5,1057)
00033      CALL GETINT(IF)
00034      IF(IF .EQ.-3.OR.(IF .GE.1.AND.IF .LE.8)) GO TO 105
00035      C
00036      C ABORT FUNCTION
00037      C
00038      102 CONTINUE
00039      CALL NWSCFN(0)
00040      WRITE(5,1006)
00041      CALL DLAY(2)
00042      GO TO 100
00043      C
00044      C PROCESS LEGAL FUNCTION
00045      C
00046      105 CONTINUE
00047      IF(IF .EQ.-3) IF =9
00048      CALL NWSCRN(0)
00049      GO TO (150,200,300,400,500,600,700,800,300),IF
00050      C
00051      C LOAD INPUT TABLES
00052      C
00053      150 CONTINUE
00054      CALL SETUP(ITABLE)
00055      GO TO 100
00056      C
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
00057 C      SAVE INPUT TABLES
00058 C
00059    200 CONTINUE
00060      CALL SAVEIT(ITABLE)
00061      GO TO 100
00062 C
00063 C      DISPLAY/INPUT TABLE
00064 C
00065    300 CONTINUE
00066      IF(IF .EQ. 9) GO TO 301
00067      WRITE(5,1008)
00068      WRITE(5,2008)
00069    301 CONTINUE
00070      WRITE(5,1009)
00071      WRITE(5,1061)
00072      CALL GETINT(IT)
00073      IF(IT.GE.1 .AND. IT.LE.NIT) GO TO 304
00074          IF(IT.LT.NIT .OR. IT.GT.19) GO TO 102
00075          CALL RDYOUT(IT,ITEMP)
00076          GO TO 306
00077    304 CONTINUE
00078      DO 305 I=1,MAXFLD
00079      DO 305 J=1,MAXROW
00080    305 ITEMPI(I,J)=ITABLE(IT,I,J)
00081 C
00082 C      PRINT SCREEN
00083 C
00084    306 CALL SCRNO(IT,ITEMP)
00085      IF(IT .LE. NIT) GO TO 308
00086      CALL DUMPR
00087      GO TO 100
00088 C
00089 C      INPUT DATA
00090 C
00091    308 CONTINUE
00092      DO 309 I=1,15
00093      DO 309 J=1,MAXROW
00094    309 IBLNK(I,J)=0
00095      IERR=0
00096      CALL SCRNI(IT,ITEMP)
00097 C
00098 C      CHECK FOR NO CHANGE IN TABLE
00099 C
00100      IF(IT .EQ.0) GO TO 100
00101 C
00102 C      SET ERROR FLAG IF FIELDS WERE IN ERROR
00103 C
00104      IF(IT.GT.0) GO TO 310
00105      IERR=1
00106      IT=-IT
00107    310 CONTINUE
00108 C
00109 C      DO SUBSEQUENT PROCESSING
00110 C
00111      CALL SUBPRO(IT,ITEMP,ITABLE)
00112      CALL ERROR(IT,ITEMP,IERR,1)
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
00113      IF(IERR.EQ.1) GO TO 306
00114          CALL DCODE(IT,ITEMP)
00115      C
00116      C     SAVE ERROR FREE TABLE
00117      C
00118          WRITE(5,1010)
00119          CALL TRSFN(IT,ITABLE,ITEMP)
00120          CALL DLAY(2)
00121          GO TO 100
00122      C
00123      C     PRINT TABLE
00124      C
00125      400 CONTINUE
00126          WRITE(5,1023)
00127          WRITE(5,1063)
00128          CALL GFTINT(IT)
00129      C
00130      C-----CHECK FOR VALID ENTRIES IN RESPONSE TO QUESTION FOR TAB NO.
00131          IF(IT.GE.1 .AND. IT.LE.19) GO TO 410
00132          IF(IT.EQ. 98) GO TO 420
00133          IF(IT.EQ. 99) GO TO 425
00134          IF(IT.EQ. 100) GO TO 430
00135          GO TO 102
00136      C
00137      C-----PRINT OUT INDIVIDUAL TABLE
00138      410 CONTINUE
00139          IE = IT
00140          IE = IT
00141          GO TO 440
00142      C
00143      C-----PRINT OUT INPUT TABLES
00144      420 CONTINUE
00145          IB = 1
00146          IE = NIT
00147          GO TO 440
00148      C
00149      C-----PRINT OUT OUTPUT TABLES
00150      425 CONTINUE
00151          IB = NIT+1
00152          IE = 19
00153          GO TO 440
00154      C
00155      C-----PRINT OUT ALL TABLES
00156      430 CONTINUE
00157          IB = 1
00158          IF = 19
00159          GO TO 440
00160      C
00161      C-----GENERAL LOOP FOR TABLE PRINTING
00162      440 CONTINUE
00163          CALL DATE(CDATE)
00164          CALL TIME(CTIME)
00165          NPP=99
00166      C
00167      C
00168          DO 490 ITT=IB,IE
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
00169      NPP=NPP+1
00170      IF(NPP.LE.2) GO TO 450
00171      NPP=1
00172      WRITE(3,1011) CDATE,CTIME
00173  450  CONTINUE
00174      WRITE(3,1012) ITT
00175      IF(ITT .GT. NIT) GO TO 460
00176      C
00177      C-----PROCESSING AN INPUT TYPE TABLE
00178      DO 455 I = 1,MAXFLD
00179      DO 455 J = 1,MAXROW
00180      ITEMP(I,J) = ITABLE(ITT,I,J)
00181  455  CONTINUE
00182      GO TO 470
00183      C
00184      C-----PROCESSING AN OUTPUT TYPE TABLE
00185  460  CONTINUE
00186      CALL RDYOUT(ITT,ITEMP)
$FTNDIM LINE:00186 POSSIBLE DO INDEX MODIFICATION INSIDE LOOP
00187      C
00188      C-----OUTPUT TABLE
00189  470  CONTINUE
00190      CALL LISTO(ITT,ITEMP)
$FTNDIM LINE:00190 POSSIBLE DO INDEX MODIFICATION INSIDE LOOP
00191      C
00192      C
00193  490  CONTINUE
00194      C
00195      C
00196      WRITE(5,1024)
00197      CALL DLAY(2)
00198      GO TO 100
00199      C
00200      C     RUN BATTLE
00201      C
00202  500  CONTINUE
00203      WRITE(5,1125)
00204      READ(5,1126) IDRUN
00205      WRITE(5,1025)
00206      CALL BATTLE
00207      WRITE(5,1026)
00208      CALL DLAY(2)
00209      GO TO 100
00210      C
00211      C     STOP
00212      C
00213  600  CONTINUE
00214      WRITE(5,1027)
00215      CALL RELEASE(23)
00216      RETURN
00217      C
00218      C     LIST TABLES
00219      C
00220  700  CONTINUE
00221      WRITE(5,1028)
00222      WRITE(5,2028)
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
00223      WRITE(5,3028)
00224      WRITE(5,1001)
00225      CALL DUMPD
00226      GO TO 100
00227      C
00228      C      FUNCTION EXPLANATION
00229      C
00230      800 CONTINUE
00231      WRITE(5,1029)
00232      WRITE(5,2029)
00233      WRITE(5,3029)
00234      WRITE(5,1001)
00235      CALL DUMRD
00236      GO TO 100
00237      C
00238      C-----FORMATS
00239      1001 FORMAT($,1H0,'TO CONTINUE STRIKE SPACE BAR. ')
00240      1002 FORMAT(72F0TO OPERATE THE OUTCOME CALCULATOR, THE USER REPEATEDLY
00241      1 SELCTS FROM THE /1H ,
00242      228HFOLLOWING LIST OF FUNCTIONS.)
00243      1003 FORMAT (16H THE FOLLOWING FUNCTIONS ARE AVAILABLE TO YOU //1H ,
00244      113H1. LOAD INPUT//1H ,
00245      213H2. SAVE INPUT//1H ,
00246      322H3. DISPLAY/INPUT TABLE//1H ,
00247      414H4. PRINT TABLE//1H ,
00248      56H5. RUN//1H ,
00249      67H6. STOP//1H ,
00250      714H7. LIST TABLES//1H ,
00251      826H8. EXPLAIN ABOVE FUNCTIONS//)
00252      1006 FORMAT (18H FUNCTION ABORTED.)
00253      1008 FORMAT(71H TO SUPPRESS THE FOLLOWING TEXT, CALL FOR THIS FUNCTION W
00254      1ITH A "-3" NEXT/1H ,
00255      25HTIME./1H ,
00256      359H ONLY TABLES NUMBERED 1-14 CAN BE INITIALIZED OR CHANGED./1H ,
00257      4'1H IF A TABLE IS INITIALIZED OR CHANGED, IT IS CHECKED FOR ENTRY
00258      5 ERRORS./1H ,
00259      625H IF NO ERRORS ARE FOUND /1H ,
00260      765H 1. THE CURRENT DATA TABLE IS REVISED TO REFLECT THE CHANGE
00261      8S.)
00262      1009 FORMAT("OTO REINITIALIZE A TABLE, LEAVE EDITING MODE "
00263      1 " IMMEDIATELY AFTER TABLE IS DISPLAYED."/
00264      2 " THE USER IS THEN RETURNED TO THE FUNCTION MENU."//
00265      3" TABLE CUES: 1-ELMTS 2-BFU 3-RFU 4-BENG 5-RENG 6-WPAV 7-ACCAP"/
00266      4" 8-BCPLX 9-RCPLX 10-MISC 11-BOPS 12-ROPS 13-POSIT 14-WX"/
00267      5" 15-BMAC 16-RMAC 17-BATT 18-RATT 19-ACEXP") )
00268      1010 FORMAT(73H CURRENT TABLE OF INTEREST WAS FOUND TO BE ERROR FREE. I
00269      1T NOW CONSTITUTES/1H ,
00270      229HTHE CURRENT INPUT DATA TABLE./ )
00271      1011 FORMAT("1 LISTING OF SOC PROGRAM TABLES.",5X,A10,5X,A5)
00272      1012 FORMAT(///,1X,"TABLE NUMBER ",I2,/ )
00273      1023 FORMAT (51H YOU MAY NOW SELECT ANY TABLE FOR PRINTING OFFLINE./
00274      +1H ,
00275      145HADDITIONALLY, WHEN ASKED FOR THE TABLE NUMBER /1H ,
00276      262H      A RESPONSE OF 98 WILL GIVE ALL OF THE INPUT TABLES(1-14)./
00277      +1H ,
00278      364H      A RESPONSE OF 99 WILL GIVE ALL OF THE OUTPUT TABLES(15-19)
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

00279	4./*	A RESPONSE OF 100 WILL GIVE ALL OF THE TABLES(1-19)./*)	
00280	1024 FORMAT (27H0OFFLINE PRINTING EXECUTED.)		
00281	1025 FORMAT (27H0THE OUTCOME CALCULATOR IS EXECUTING.)		
00282	1026 FORMAT(61H0OUTCOME CALCULATIONS FOR CURRENT INPUT DATA TABLES COMP		
00283	1LETE.)		
00284	1027 FORMAT (31H OUTCOME CALCULATOR TERMINATED.)		
00285	1028 FORMAT (42H	TABLE DIRECTORY/	
00286	156H NO.	TITLE	TYPE/
00287	256H 1	FORCE ELEMENTS	INPUT/
00288	356H 2	BLUE FORCE UNITS	INPUT/
00289	456H 3	RFD FORCE UNITS	INPUT/
00290	556H 4	ENGAGEMENT STATISTICS FOR BLUE ATTACKING RED	INPUT/
00291	656H 5	ENGAGEMENT STATISTICS FOR RFD ATTACKING BLUE	INPUT/
00292	756H 6	WEAPON PLATFORM AVAILABILITY	INPUT/
00293	1029 FORMAT (42H	FUNCTION EXPLANATIONS/1H ,/1P ,	
00294	168H1. LOAD INPUT - INPUT DATA TABLES WILL BE ENTERED FROM A PREVIOUSLY		
00295	2USL 1H ,		
00296	337H CONSTRUCTED DATA SET./1H ,/1H ,		
00297	471H2. SAVE INPUT - CURRENT INPUT DATA TABLES WILL BE SAVED FOR FUTURE USE./1H ,/1H ,		
00298	672H3. DISPLAY/INPUT - SUBSEQUENTLY SELECTED TABLE WILL BE DISPLAYED		
00299	7D. IF THE/1H ,		
00300	871H TABLE	TABLE IS IN THE INPUT DATA SET IT CAN BE INITIALIZED)	
00301			
00302	1057 FORMAT(\$,46HWHAT FUNCTION NUMBER DO YOU WISH TO EXECUTE?)		
00303	1061 FORMAT(\$,42H WHAT TABLE DO YOU WISH TO DISPLAY/INPUT?)		
00304	1063 FORMAT(\$,41H WHAT TABLE NUMBER DO YOU WISH TO PRINT?)		
00305	1125 FORMAT(' WHAT IS THE RUN ID (10 CHAR MAX)? ')		
00306	1126 FORMAT(%10)		
00307	2008 FORMAT (51H	2. THE USER IS RETURNED TO THE FUNCTION MENU./	
00308	11H ,23H IF AN ERROR IS FOUND /1H ,		
00309	269H 1. THE TABLE IS DISPLAYED AGAIN FOR CORRECTION(BAD FIELDS		
00310	3 NOTED)/1H ,		
00311	449H 2. NO ACTION IS TAKEN ON CURRENT DATA TABLE.)		
00312	2023 FORMAT(56H 7 CAPABILITIES OF A/C RELATED ELEMENTS		INPUT
00313	1/		
00314	255H 8	BLUE FORCE COMPLEXES	INPUT/
00315	356H 9	RFD FORCE COMPLEXES	INPUT/
00316	456H 10	MISCELLANEOUS INPUT	INPUT/
00317	556H 11	BLUE OPERATIONS PLANS	INPUT/
00318	656H 12	RFD OPERATIONS PLANS	INPUT/
00319	756H 13	RELATIVE COMPLEX POSITIONS	INPUT/
00320	856H 14	INITIATING WX DAYS	INPUT/
00321	2029 FORMAT (30H	OR EDITED./1H ,/1P ,	
00322	169H4. PRINT TABLE - SUBSEQUENTLY SELECTED TABLE WILL BE PRINTED OF		
00323	2FLINE./1H ,		
00324	369H5. RUN - CURRENT INPUT DATA TABLES WILL BE USED TO GENERATE OUTCOMES./1H ,		
00325	548H6. STOP - OUTCOME CALCULATOR WILL BE TERMINATED./)		
00326	3023 FORMAT(57H 15 BLUE MISSION ACCOMPLISHMENT RESULTS		OUTPUT
00327	1T/		
00328	257H 16	RED MISSION ACCOMPLISHMENT RESULTS	OUTPUT/
00329	357H 17	BLUE COMPLEX BATTLE ATTRITION RESULTS	OUTPUT/
00330	457H 18	RFD COMPLEX BATTLE ATTRITION RESULTS	OUTPUT/
00331	557H 19	AIRCRAFT EXPENDITURE SUMMARY	OUTPUT/
00332	3029 FORMAT(68H 7. LIST TABLES - REFERENCE NUMBERS AND TITLES OF TABLES		

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

00335 1 USED BY THE/1H ,
00336 254H OUTCOME CALCULATOR WILL BE DISPLAYED.//)
00337 END

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
00001
00002      SUBROUTINE BATL
00003      C
00004      C      THIS ROUTINE CONTROLS THE OUTCOME COMPUTATION ALGORITHM
00005      C
00006      CALL NEWRUN(IEND, IDAY)
00007      200 CALL NEWDAY(IEND, IDAY)
00008      IF(IEND.EQ.1) RETURN
00009      DO 400 ITS=1,8
00010      CALL SCHED(IDAY,ITS)
%FTNDIM LINE:00010 POSSIBLE DO INDEX MODIFICATION INSIDE LOOP
00011      CALL RECOVR
00012      CALL LAUNCH(ITS, IDAY)
%FTNDIM LINE:00012 POSSIBLE DO INDEX MODIFICATION INSIDE LOOP
00013      CALL ENGAGE(IDAY)
00014      CALL RECOVR
00015      400 CONTINUE
00016      GO TO 200
00017      END
```

```

00001
00002
00003 C SUBROUTINE NEWRUN(IEND, IDAY)
00004 C THIS ROUTINE INITIALIZES AN OUTCOME COMPUTATION
00005 C
00006 C DOUBLE PRECISION DCOMPS,DPLFX,ORIG,TARG,TTS
00007 C REAL LRDEF
00008 C INTEGER ORIGIN,TARGET,START,STOP,UMAX,UMIN,UREADY
00009 C INTEGER UASK,UCOT
00010 C REAL LIVPLX
00011 C COMMON/PLOCKB/TSUNIT(2,20,2),SCOMPS(2,20,5),COMPS(2,20,5),
00012 C +STRAT(2,20,4),ADDEFAT(2,20,2),AGKILL(2,20,10),SPRATE(2,2,5),
00013 C +DCOMPS(2,10),MAXR(2,20),LRDEF(2,20),WSTWX(2,20),BWDEF(2,20),
00014 C +DRMAX(2,20),SPEED(2,20),TRRTIM(2,5),SR30MX(2,5),IMAXR(2,20),
00015 C +ILRDEF(2,20),IWSTWX(2,20),IEWDEF(2,20),NCOMPS(2),NSUNIT(2)
00016 C COMMON/BLOCKC/COMPLX(2,10,10),
00017 C +DPLFX(2,10),PTSLL(2,10),SURFAC(2,10),DEFAC(2,10),TCYCLE(2,10),
00018 C +IREPL(2,10),RFPOPS(2,10,2),OPS(2,11),REPAIR(2,11),
00019 C +NPFLX(2),MTIME(2),NDAYS,JWKDEF,JLRDEF,WXSURV,DELAY,
00020 C +OPSLCH,OPSLND
00021 C COMMON/BLOCKD/IONOFF(2,20,2),JONOFF(2,20,2),S1(2,20,2),S2(2,20,2),
00022 C +MTS(2,20,8),UNIT(2,20,2),SA(2,20,2),SP(2,20,2),TTS(2,20),
00023 C +RANGE(10,10),IRANGE(10,10),PMISS(2,20),IPRI(2,20),ORIG(2,20),
00024 C +TARG(2,20),UMAX(2,20),UMIN(2,20),ORIGIN(2,20),TARGET(2,20),
00025 C +IUNIT(2,20),START(2,20),STOP(2,20),UREADY(2,20),
00026 C +DAY(20),WX(30),IWX(30),NMIS(2)
00027 C COMMON/PLOCKE/LIVPLX(2,10,10),UASK(2,20,3),UGUT(2,20,3),
00028 C +SR30(2,10,5),AIRUP(20,5,8),AIRDN(20,5,8),ADBASE(2,10)
00029 C COMMON /AIREXP/ EXP(19,10)
00030 C IDAY=0
00031 C IEND=0
00032 C
00033 C SET UP FLAGS TO INDICATE START AND STOP CRITERIA FOR MISSIONS
00034 C N=1 FOR START
00035 C N=2 FOR STOP
00036 C
00037 DO 490 L=1,2
00038 NM=NHTSS(L)
00039 DO 485 M=1,NM
00040 IONOFF(L,M,1)=0
00041 IONOFF(L,M,2)=0
00042 DO 480 N=1,2
00043 SA(L,M,N)=S1(L,M,N)
00044 SB(L,M,N)=S2(L,M,N)
00045 IF(S1(L,M,N).NE. 2HEND) GO TO 350
00046 DECODE(10,1000,S2(L,M,N)) A
00047 IS=A+.5
00048 IF(N.EQ.1) START(L,N)=IS
00049 IF(N.EQ.2) STOP(L,N)=IS
00050 GO TO 490
00051 350 IF(S1(L,M,N).NE. 2HEND) GO TO 450
00052 DO 440 J=1,NM
00053 IF(S2(L,M,N).NE.DHTSS(L,J)) GO TO 440
00054 IONOFF(L,M,N)=1
00055 JUNOFF(L,M,N)=J
00056 IF(N.EQ.1) START(L,N)=100

```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
00057      IF(N.EQ.2) STOP(L,M)=100
00058      GO TO 480
00059      440 CONTINUE
00060      450 IF(S1(L,M,N).EQ.3H ) GO TO 475
00061      DECODE(10,1000,S2(L,M,N)) A
00062      IPERCT=A+.5
00063      1000 FORMAT(F5.0,5X)
00064      IF(S1(L,M,N).EQ. 3HDOA) IA=2
00065      IF(S1(L,M,N).EQ. 3HDDA) IA=3
00066      IF(S1(L,M,N).EQ. 3HDOS) IA=4
00067      IF(S1(L,M,N).EQ. 3HDS) IA=5
00068      IF(S1(L,M,N).EQ. 3HOLS) IA=6
00069      IONOFF(L,M,N)=IA
00070      JONOFF(L,M,N)=IPERCT
00071      475 IF(N.EQ.1) START(L,M)=100
00072      IF(N.EQ.2) STOP(L,M)=100
00073      480 CONTINUE
00074      485 CONTINUE
00075      490 CONTINUE
00076
C      VARIABLE EXPLANATIONS
00077      C      NMISS=NO. OF MISSIONS L=1 FOR PLUE L=2 FOR RED
00078      C      NPLEX=NO. OF COMPLEXES
00079      C      NCOMPS=NO. OF FORCE COMPONENTS
00080      C      UASK=CUM NO. OF FORCE UNITS REQUESTED FOR M-TH MISSION
00081      C      UGOT=CUM NO. OF RECEIVED UNITS
00082      C      LIVPLX=NO. OF LIVE TYPE T UNITS AT K-TH COMPLEX
00083      C      ADBASE=CUM NO. OF OPERATIONS REPAIRED AT K-TH COMPLEX
00084      C      AIRON=AVAIL. OF FORCE UNITS ON GROUND
00085      C      J INDICATES TIME STEP(1=CURRENT)
00086      C      AIRUP=STATUS OF UNITS OUT ON MISSIONS
00087      C
00088      C
00089      DO 650 L=1,2
00090      NM=NMISS(L)
00091      NP=NPLEX(L)
00092      NC=NCOMPS(L)
00093      DO 500 M=1,NM
00094      DO 500 N=1,3
00095      UASK(L,M,N)=0
00096      UGOT(L,M,N)=0
00097      500 CONTINUE
00098      DO 600 K=1,NP
00099      ADBASE(L,K)=0.0
00100      DO 600 I=1,NC
00101      LIVPLX(L,K,I)=COMPLX(L,K,I)
00102      IF(I.LE.5) SR30(L,K,I)=0.0
00103      600 CONTINUE
00104      650 CONTINUE
00105      N=0
00106      DO 700 L=1,2
00107      NP=NPLEX(L)
00108      DO 700 M=1,NP
00109      N=N+1
00110      DO 700 I=1,3
00111      DO 700 J=1,8
00112      AIPDN(M,I,J)=0.0
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
00113      IF(J.EQ.1) AIRDN(N,I,J)=LTVPLX(L,K,I)
00114      IF(L.EQ.1.AND.J.EQ.1) AIRDN(N,5,J)=0.0
00115      AIRUP(N,I,J)=0.0
00116      700 CONTINUE
00117          DD 900 I=1,10
00118          DD 800 J=1,19
00119          EXPS(J,I)=0.0
00120      800      CONTINUE
00121      RETURN
00122      END
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
00001
00002      SUBROUTINE NEWDAY(IDAY, IDAY)
00003      C
00004      C      THIS ROUTINE INITIALIZES A NEW DAY IN THE ENGAGEMENT
00005      C
00006      DOUBLE PRECISION DCOMP, DPLEX, UASK, TARG, TTS
00007      REAL LRDF
00008      INTEGER ORIGIN, TARGET, START, STOP, UMAX, UMIN, UREADY
00009      INTEGER UASK, UGOT
00010      REAL LIVPLX
00011      DIMENSION D(10,6), XNTCER(31)
00012      COMMON/MODS/IDRUN, JDAY
00013      DOUBLE PRECISION IDRUN
00014      COMMON/ENDC/JENDC
00015      COMMON/RLOCKB/PSUNIT(2,20,2), SCOMPS(2,20,5), COMPS(2,20,5),
00016      +STRAT(2,20,4), ADFFAT(2,20,2), AGKILL(2,20,10), SRATE(2,2,5),
00017      +DCOMPS(2,10), MAXR(2,20), LRDEF(2,20), WSTWX(2,20), BWDEF(2,20),
00018      +DRMAX(2,20), SPEED(2,20), IRRTIM(2,5), SR30MX(2,5), IMAXR(2,20),
00019      +ILRDEF(2,20), IWSTWX(2,20), IBWDEF(2,20), NCOMPS(2), NSUNIT(2)
00020      COMMON/BLOCKC/COMPLX(2,10,10),
00021      +DPLEX(2,10), DTSLI(2,10), SURFAC(2,10), DEFAC(2,10), ICYCLE(2,10),
00022      +IREPL(2,10), REPOPS(2,10,2), OPS(2,11), REPAIK(2,11),
00023      +NPLEX(2), NTIME(2), NDAYS, JWXDEF, JLRDFF, WXSURV, DELAY,
00024      +OPSLCH, OPSLND
00025      COMMON/RLOCKD/JONOFF(2,20,2), JONOFF(2,20,2), S1(2,20,2), S2(2,20,2),
00026      +MTS(2,20,8), UNIT(2,20,2), SA(2,20,2), SR(2,20,2), TTS(2,20),
00027      +RANGE(10,10), IPANGE(10,10), DMISS(2,20), IPRI(2,20), ORIG(2,20),
00028      +TARG(2,20), UMAX(2,20), UMIN(2,20), ORIGIN(2,20), TARGET(2,20),
00029      +IUNIT(2,20), START(2,20), STOP(2,20), UREADY(2,20),
00030      +DAY(30), WX(30), IWX(30), NMIS(2)
00031      COMMON/BLOCKE/LIVPLX(2,10,10), UASK(2,20,3), UGOT(2,20,3),
00032      +SR30(2,10,5), AIRUP(20,5,8), AIRDN(20,5,8), ADRASE(2,10)
00033      COMMON/RLOCKF/SR(2,10,5), IPTY(2,9)
00034      COMMON /AIREXP/ EXP(19,10)
00035      DATA XNTCER/SH 0,
00036      +5H 1 ,5H 2 ,5H 3 ,5H 4 ,5H 5 ,
00037      +      SH 6 ,5H 7 ,5H 8 ,5H 9 ,5H 10 ,
00038      +      SH 11 ,5H 12 ,5H 13 ,5H 14 ,5H 15 ,
00039      +      SH 16 ,5H 17 ,5H 18 ,5H 19 ,5H 20 ,
00040      +      SH 21 ,5H 22 ,5H 23 ,5H 24 ,5H 25 ,
00041      +      SH 26 ,5H 27 ,5H 28 ,5H 29 ,5H 30 /
00042      C
00043      C      INITIALIZE VARIABLES AT THE BEGINNING OF A NEW DAY
00044      C
00045      JDAY=IDAY
00046      IDAY=IDAY+1
00047      DO 500 L=1,2
00048      NP=NPLEX(L)
00049      DO 150 IK=1,9
00050      IPTY(L,IK)=0
00051      150 CONTINUE
00052      DO 330 K=1,NP
00053      DO 200 I=1,6
00054      IF(I.LE.5) SR(L,K,I)=0.0
00055      D(K,I)=0.0
00056      200 CONTINUE
```

```

00057 C
00059 C COMPUTE DESTRUCTION LEVEL AT EACH COMPLEX FOR EACH ELEMENT CL
00059 C D=FRACTION OF UNITS I AT K DESTROYED
00060 C I=(? FOR DA,3 DA,4 DS,5 DS,6 LS)
00061 C
00062 C IF(L.EQ.2) GO TO 300
00063 XN=LIVPLX(L,K,1)+LIVPLX(L,K,2)
00064 XD=COMPLX(L,K,1)+COMPLX(L,K,2)
00065 IF(YF.GT..001)D(K,2)=1.-XN/XD
00066 XN=LIVPLX(L,K,3)+LIVPLX(L,K,4)
00067 XD=COMPLX(L,K,3)+COMPLX(L,K,4)
00068 IF(YD.GT..001)D(K,3)=1.-XN/XD
00069 IF(COMPLX(L,K,5).GT..001)D(K,4)=1.-LIVPLX(L,K,5)/
00070 1 COMPLX(L,K,5)
00071 IF(COMPLX(L,K,6).GT..001)D(K,5)=1.-LIVPLX(L,K,6)/
00072 1 COMPLX(L,K,6)
00073 GO TO 330
00074 300 CONTINUE
00075 XN=LIVPLX(L,K,1)+LIVPLX(L,K,2)+LIVPLX(L,K,3)
00076 XD=COMPLX(L,K,1)+COMPLX(L,K,2)+COMPLX(L,K,3)
00077 IF(XF.GT..001)D(K,2)=1.-XN/XD
00078 IF(COMPLX(L,K,4).GT..001)D(K,3)=1.-LIVPLX(L,K,4)/
00079 1 COMPLX(L,K,4)
00080 XN=LIVPLX(L,K,5)+LIVPLX(L,K,6)
00081 XD=COMPLX(L,K,5)+COMPLX(L,K,6)
00082 IF(XN.GT..001)D(K,4)=1.-XN/XD
00083 IF(COMPLX(L,K,7).GT..001)D(K,5)=1.-LIVPLX(L,K,7)/
00084 1 COMPLX(L,K,7)
00085 IF(COMPLX(L,K,8).GT..001)D(K,6)=1.-LIVPLX(L,K,8)/
00086 1 COMPLX(L,K,8)
00087 330 CONTINUE
00088 C
00089 C UPDATE START AND STOP IN OPERATIONS PLAN
00090 C
00091 J=2
00092 IF(L.EQ.2) J=1
00093 NM=NMISS(J)
00094 DO 450 ITF=1,2
00095 DJ 400 M=1,NM
00096 DO 350 N=1,2
00097 IST=0
00098 IF(N.EQ.1) S=START(J,M)
00099 IF(N.EQ.2) S=STOP(J,M)
00100 IO= IONOFF(J,M,N)
00101 JO= JONOFF(J,M,N)
00102 C
00103 C CHECK FOR DAY SPECIFICATION
00104 C
00105 IF(IO .EQ.0) GO TO 350
00106 IF(IO.NE.1) GO TO 340
00107 IF(STOP(J,JO).GT.99) GO TO 350
00108 C
00109 C START/STOP CRITERIA IN TERMS OF END MISSION
00110 C
00111 IONOFF(J,M,N)=0
00112 SA(J,M,N)=3HDAY

```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
00113      IS=STOP(J,JO)+2-N
00114      IF(N.EQ.1) START(J,M)=IS
00115      IF(N.EQ.2) STOP(J,V)=IS
00116      SB(J,M,N)=XNTGER(IS+1)
00117      GO TO 350
00118  C
00119  C      START/STOP DEPENDS ON DESTRUCTION LEVEL
00120  C
00121      340 JT=TARGET(J,M)
00122      AD=JONOFF(J,M,N)/100.
00123      IF( S.GT.99.AND. C(JT,IO).GE.AD) IST=1
00124      IF(IST.EQ.0) GO TO 350
00125      JONOFF(J,F,N)=0
00126      IS=IDAY+1-N
00127      IF(N.EQ.1) START(J,V)=IS
00128      IF(N.EQ.2) STOP(J,V)=IS
00129      SA(J,M,N)=3HDAY
00130      SB(J,M,N)=XNTGER(IS+1)
00131      350 CONTINUE
00132      400 CONTINUE
00133      450 CONTINUE
00134      500 CONTINUE
00135  C
00136  C      CHECK TO SEE IF EITHER SIDE HAS COMPLETED ALL OFFENSIVE MISSIONS
00137  C
00138  C      VARIABLES ARE
00139  C          IDAY=CURRENT DAY
00140  C          NDAYS=MAX DAYS
00141  C          IDEF=FLAG FOR DEFENSE MISSION
00142  C          JENDC=FLAG TO CONTINUE EVEN IF ONLY DEFENSE LEFT
00143      IEND=1
00144      IF(IDAY.GT.NDAYS) GO TO 800
00145      DO 700 L=1,2
00146      NM=NMISS(L)
00147      DO 650 M=1,NM
00148      IDEF=0
00149      IF(UNIT(L,M,1).EQ.5HSUCAF.OR.UNIT(L,M,1).EQ.5HVFCAP.OR.UNIT(L,M,1)
00150      +.EQ.5HDLI .OR.UNIT(L,M,1).EQ.5HSLI ) IDEF=1
00151      IF(IDAY.GT.STOP(L,M).OR.IPRI(L,M).EQ.0) GO TO 650
00152      IF(IDEF.EQ.1.AND.JENDC.EQ.0) GO TO 650
00153      GO TO 700
00154      650 CONTINUE
00155      GO TO 800
00156      700 CONTINUE
00157      IEND=0
00158      RETURN
00159  C
00160  C      TERMINATE BATTLE
00161  C
00162      800 DO 900 L=1,2
00163      NP=ENDLEY(L)
00164      NC=VCOMPS(L)
00165      DO 900 K=1,NC
00166      DO 950 I=1,NC
00167      LIVPLY(L,K,I)=COMPLY(L,K,I)-LIVPLY(L,K,I)
00168      950 CONTINUE
00169      900 CONTINUE
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
00169 900 CONTINUE
00170 IF(JDAY.EQ.1) GO TO 983
00171 DO 982 I=2,JDAY
00172 DO 982 J=1,19
00173 EXPS(J,I)=EXPS(J,I)+EXPS(J,I-1)
00174 982 CONTINUE
00175 983 CONTINUE
00176 RETURN
00177 END
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
00001
00002      SUBROUTINE SCHED(IDAY,ITS)
00003      C
00004      C      THIS ROUTINE SETS UP THE MISSION SCHEDULE FOR THE DAY
00005      C
00006      REAL LRDEF
00007      DOUBLE PRECISION DCOMPS,DPLEX,ORIG,TARG,TTS
00008      INTEGER ORIGIN,TARGET,START,STOP,UMAX,UMIN,UREADY
00009      INTEGER UASK,UGOT
00010      REAL LIVPLX,NOPS
00011      DIMENSION SRT(2,10,5),SR30T(2,10,5),A(5),JRPLEN(2,10)
00012      COMMON/BLOCKB/DSUNIT(2,20,2),SCOMPS(2,20,5),COMPS(2,20,5),
00013      +STRAT(2,20,4),ADEFAT(2,20,2),AGKILL(2,20,10),SRATE(2,2,5),
00014      +DCOMPS(2,10),MAXR(2,20),LRDEF(2,20),WSTWX(2,20),BWDEF(2,20),
00015      +DRMAX(2,20),SPEED(2,20),IRRTIM(2,5),SR30MX(2,5),IMAXR(2,20),
00016      +ILRDEF(2,20),IWSTWX(2,20),IBWDEF(2,20),NCOMPS(2),NSUNIT(2)
00017      COMMON/BLOCKC/COMPLX(2,10,10),
00018      +DPLEX(2,10),DTSLI(2,10),SURFAC(2,10),DEFAC(2,10),ICYCLE(2,10),
00019      +IREPL(2,10),REPOPS(2,10,2),OPS(2,11),REPAIR(2,11),
00020      +NPLEX(2),MTIME(2),          NDAYS,JWXDEF,JLRDEF,WXSURV,DELAY,
00021      +OPSLCH,OPSLND
00022      COMMON/BLOCKE/IONOFF(2,20,2),JONOFF(2,20,2),S1(2,20,2),S2(2,20,2),
00023      +MTS(2,20,8),UNIT(2,20,2),SA(2,20,2),SB(2,20,2),TTS(2,20),
00024      +RANGE(10,10),IPANGE(10,10),DMISS(2,20),IPRI(2,20),ORIG(2,20),
00025      +TARG(2,20),UMAX(2,20),UMIN(2,20),ORIGIN(2,20),TARGET(2,20),
00026      +IUNIT(2,20),START(2,20),STOP(2,20),UREADY(2,20),
00027      +DAY(30),WX(30),IWX(30),NMISS(2)
00028      COMMON/BLOCKF/LIVPLX(2,10,10),UASK(2,20,3),UGOT(2,20,3),
00029      +SR30(2,10,5),AIRUP(20,5,8),ATRDN(20,5,8),ADBASE(2,10)
00030      COMMON/BLOCKF/SR(2,10,5),IPTY(2,9)
00031      COMMON/BLOCKH/ISCHED(2,20,8),IU(2,20),NOPS(2,10)
00032      C
00033      IF(ITS.NE.1) GO TO 350
00034      C
00035      C      DETERMINE WHICH COMPLEXES ARE IN REPLENISHMENT
00036      C
00037      DO 150 L=1,2
00038      NP=NPLEX(L)
00039      DO 100 K=1,NP
00040      JRPLEN(L,K)=0
00041      IF(ICYCLE(L,K).LE.0) GO TO 100
00042      J=(IDAY-1)/ICYCLE(L,K)
00043      JJ=IDAY-J*ICYCLE(L,K)
00044      JL=ICYCLE(L,K)-JJ
00045      IF(JL.GE.IREPL(L,K)) GO TO 100
00046      JRPLEN(L,K)=1
00047      100 CONTINUE
00048      150 CONTINUE
00049      C
00050      C      CONSTRUCT TODAYS MISSION SCHEDULE
00051      C
00052      DO 300 L=1,2
00053      NM=NMISS(L)
00054      DO 250 M=1,NM
00055      DO 200 KK=1,3
00056      C
```

```
00057 C      ISCHED=0 IF NO MISSION ELSE PRIORITY OF MISSION
00058 C          M=MISSION NO.
00059 C          KK=TIME STEP
00060 C
00061 C      ISCHED(L,M,KK)=0
00062 C 200 CONTINUE
00063 C
00064 C      MISSION SCHEDULER FILTER
00065 C
00066 C      IF(IDAY.LT.START(L,M).OR.IDAY.GT.STOP(L,M).OR.UMAX(L,M).LT.1.OR.
00067 C          +IPRI(L,M).LT.1) GO TO 250
00068 C          I1=ORIGIN(L,M)
00069 C          I2=TARGET(L,M)
00070 C          IUN=IUNIT(L,M)
00071 C          IUN1=IUN
00072 C          IDEF=0
00073 C          IF(UNIT(L,M,1).EQ.5HSUCAP.OR.UNIT(L,M,1).EQ.5HVFCAP.OR.UNIT(L,M,1)
00074 C          +.EQ.5HDLI .OR.UNIT(L,M,1).EQ.5HSLI ) IDEF=1
00075 C
00076 C      IF ORIGIN IN REPLENISHMENT AND "D" no offense scheduled
00077 C
00078 C      IF(JRPLEN(L,I1).EQ.1.AND.RFPOPS(L,I1,2).EQ.1HD.AND.
00079 C          +IDEF.EQ.0) GO TO 250
00080 C
00081 C      SCHEDULFR WILL GET WX OR RANGE DEFAULTS AS NEEDED AND AVAIL
00082 C          IW=1 FOR BAD ELSE 2 FOR GOOD
00083 C          IR=1 FOR LONG ELSE 2 FOR SHORT
00084 C          DEFENSE MISSIONS DO NOT LOOK AT RANGE
00085 C          N=1 FOR NORMAL UNIT
00086 C          N=2 FOR BAD WX DEFAULT
00087 C          N=3 FOR LONG RANGE DEFAULT
00088 C
00089 C          N=1
00090 C          IW=IWX(IDAY)
00091 C          IR=2
00092 C          IF(L.EQ.1.AND.I2.GT.0) IR=IRANGE(I1,I2)
00093 C          IF(L.EQ.2.AND.I2.GT.0) IR=IRANGE(I2,I1)
00094 C          IF(IW.GE.IWSTWX(L,IUN)) GO TO 210
00095 C          IF(IBWDEF(L,IUN).EQ.0.OR.JWXDEF.EQ.0) GO TO 250
00096 C          IUN1=IBWDEF(L,IUN)
00097 C          N=2
00098 C          210 IF(IR.GE.1MAXR(L,IUN1).OR.IDEF.EQ.1) GO TO 230
00099 C          IF(ILRDFF(L,IUN) .EQ.0.OR.JLRDEF.EQ.0) GO TO 250
00100 C          IUN1=ILRDFF(L,IUN)
00101 C          N=3
00102 C          IF(IW.LT.1WSTWX(L,IUN1)) GO TO 250
00103 C          230 IU(L,M)=IUN1
00104 C          JU(L,M)=N
00105 C          IP=IPRI(L,M)
00106 C          IPTY(L,IP)=1
00107 C          DO 240 K=1,R
00108 C          IT=MTS(L,M,K)
00109 C          IF(IT.EQ.0) GO TO 240
00110 C
00111 C      SCHEDULE MISSION      MASK=DESIRED PLUS READY
00112 C
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
00113      ISCHED(L,M,IT)=TPRI(L,M)
00114      UASK(L,M,N)=UASK(L,M,N)+UMAX(L,M)+UREADY(L,M)
00115      240 CONTINUE
00116      250 CONTINUE
00117      300 CONTINUE
00118      C
00119      C      COMPUTE THE NUMBER OF POSSIBLE OPERATIONS AT CV AND AIRBASE ELEMENTS
00120      C
00121      350 DO 500 L=1,2
00122      NP=NPLEX(L)
00123      JB=5
00124      IF(L.EQ.2) JB=6
00125      DO 400 K=1,NP
00126      NOPSL(K)=0.0
00127      IJ=1
00128      IF(COMPLX(L,K,JB).GT.0.001)
00129      +IJ=(COMPLX(L,K,JB)-LIVPLX(L,K,JB))/COMPLX(L,K,JB)*10.0+1.99
00130      C
00131      C      COMPUTE AVAIL OPERATIONS AND ADD IN REPAIRS
00132      C
00133      F=1.0
00134      IF(JRPLFN(L,K).EQ.1) F=REPOPS(L,K,1)/100.
00135      ADD=A MIN1(COMPLX(L,K,JB)-LIVPLX(L,K,JB),REPAIR(L,IJ)*COMPLX(L,K,JB)
00136      +))
00137      ADBASE(L,K)=ADBAS(E(L,K)+ADD
00138      LIVPLX(L,K,JB)=LIVPLX(L,K,JB)+ADD
00139      NOPSL(K)=OPS(L,IJ)*COMPLX(L,K,JB)*F
00140      400 CONTINUE
00141      500 CONTINUE
00142      C
00143      C      RESCHEDULE UNFILLED PRIORITY 1 AND 2 MISSIONS FROM LAST TIME STEP
00144      C      IF SAME MISSION NOT SCHEDULED
00145      C
00146      IF(ITS.EQ.1) GO TO 535
00147      DO 530 L=1,2
00148      NM=NMISS(L)
00149      DO 520 M=1,NM
00150      IT1=ITS-1
00151      IF(ISCHED(L,M,IT1).EQ.0.OR.ISCHED(L,M,IT1).GT.2) GO TO 520
00152      IF(ISCHED(L,M,ITS).GT.0) GO TO 520
00153      ISCHED(L,M,ITS)=ISCHED(L,M,IT1)
00154      520 CONTINUE
00155      530 CONTINUE
00156      C
00157      C      UPDATE AIRCRAFT STATUS
00158      C
00159      535 IC=0
00160      DO 555 L=1,2
00161      NP=NPLEX(L)
00162      DO 550 K=1,NP
00163      IC=IC+1
00164      DO 545 I=1,5
00165      IF(L.EQ.1.AND.I.EQ.5) GO TO 545
00166      AIRDN(IC,I,1)=AIRDN(IC,I,1)+AIRDN(IC,I,2)
00167      AIRUP(IC,I,1)=AIRUP(IC,I,1)+AIRUP(IC,I,2)
00168      DO 540 J=2,7
```

```
00169      AIRDN(IC,I,J)=AIRDN(IC,I,J+1)
00170      AIRUP(IC,I,J)=AIRUP(IC,I,J+1)
00171      540 CONTINUF
00172      AIRDN(IC,I,8)=0.0
00173      AIRUP(IC,I,8)=0.0
00174      545 CONTINUE
00175      550 CONTINUE
00176      555 CONTINUE
00177      C
00178      C      PROJECT SORTIE REQUIREMENTS FORDING FOR PRIORITY 1 AND 2
00179      C      SURGE FOR PRIORITY 1 NEVER EXCEEDING 30 DAY UTILIZATION
00180      C      CANCEL MISSIONS FOR WHICH THERE ARE INSUFFICIENT RESOURCES
00181      C
00182      DO 800 L=1,2
00183      II=4
00184      IF(L.EQ.2) II = 5
00185      NM=NMISS(L)
00186      NP=NPLEX(L)
00187      DO 570 K=1,NP
00188      DO 570 I=1,II
00189      SR30T(L,K,I)=SR30(L,K,I)
00190      SRT(L,K,I)=SR(L,K,I)
00191      570 CONTINUE
00192      DO 700 IP=1,9
00193      IF(IPTY(L,IP).EQ.0) GO TO 700
00194      DO 680 IT=ITS,8
00195      DO 670 M=1,NM
00196      IF(ISCHED(L,M,IT).NE.IP) GO TO 670
00197      IO=ORIGIN(L,M)
00198      IUN=IU(L,M)
00199      DO 660 I=1,II
00200      A(I)=SRT(L,IO,I)
00201      ANEED=(UMAX(L,M)+UREADY(L,M))*SCOMPS(L,IUN,I)
00202      IF(ANEED.LT.0.001) GO TO 660
00203      IF(LIVPLX(L,IO,I).LT.0.001) GO TO 665
00204      A(I)=(ANFED/LIVPLX(L,IO,I))+SRT(L,IO,I)
00205      IF(A(I).GT.SRATE(L,1,I).AND.IP.GT.2) GO TO 665
00206      IF(A(I).GT.SRATE(L,2,I).AND.IP.GT.1) GO TO 665
00207      IF(A(I)+SR30T(L,IO,I).GT.SR30MX(L,I)) GO TO 665
00208      660 CONTINUE
00209      GO TO 666
00210      665 ISCHED(L,M,IT)=0
00211      GO TO 670
00212      666 DO 667 I=1,II
00213      SRT(L,IO,I)=A(I)
00214      667 CONTINUE
00215      670 CONTINUF
00216      680 CONTINUE
00217      700 CONTINUF
00218      800 CONTINUF
00219      RETURN
00220      END
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
00001
00002      SUBROUTINE RECOVR
00003      C
00004      C      THIS ROUTINE RECOVERS AIRCRAFT WAITING TO LAND
00005      C
00006      REAL LRDEF
00007      DOUBLE PRECISION DCOMPS,DPLEX,ORIG,TARG,TTS
00008      INTEGER UASK,UGOT
00009      REAL LIVPLX,NOPS
00010      COMMON/BLOCKB/DSUNIT(2,20,2),SCOMPS(2,20,5),COMPS(2,20,5),
00011      +STRAT(2,20,4),ADEFAT(2,20,2),ACKILL(2,20,10),SRATE(2,2,5),
00012      +DCOMPS(2,10),MAXR(2,20),LRDEF(2,20),WSTWX(2,20),BWDEF(2,20),
00013      +DRMAX(2,20),SPFED(2,20),TRRTIM(2,5),SR30MX(2,5),IMAXR(2,20),
00014      +ILRDEF(2,20),IWSTWX(2,20),IRWDEF(2,20),NCOMPS(2),NSUNIT(2)
00015      COMMON/BLOCKC/COMPLX(2,10,10),
00016      +DPLEX(2,10),DTSLI(2,10),SURFAC(2,10),DEFAC(2,10),ICYCLE(2,10),
00017      +IREPL(2,10),REPOPS(2,10,2),OPS(2,11),REPAIR(2,11),
00018      +NPLEX(2),MTIME(2),NDAYS,JWXDEF,JLRDEF,WXSURV,DELAY,
00019      +OPSLCH,OPSLND
00020      COMMON/PLOCKE/LIVPLX(2,10,10),UASK(2,20,3),UGOT(2,20,3),
00021      +SR30(2,10,5),AIRUP(20,5,8),AIRDN(20,5,8),ADBASE(2,10)
00022      COMMON/BLOCKH/ISCHED(2,20,8),TU(2,20),JU(2,20),NOPS(2,10)
00023      C
00024      C      RECOVER AIRCRAFT WAITING TO LAND
00025      C
00026      C      RECOVERY          RR   TIME          POSSIBLE TAKE OFF
00027      C      BEGINNING        0           CURRENT
00028      C      BEGINNING        1           NEXT
00029      C      END            0 OR 1        NEXT
00030      C
00031      C
00032      DO 500 L=1,2
00033      NP=NPLEX(L)
00034      II=4
00035      DO 400 K=1,NP
00036      ICX=K
00037      IF(L.EQ.2) ICX=NPLEX(1)+K
00038      DO 300 J=1,1000
00039      NJ=0
00040      DO 200 I=1,II
00041      IF(ATRUP(ICX,I,1).LT.0.0001.OR.NOPS(L,K).LT..001) GO TO 200
00042      A=AMIN1(1.0,AIRUP(ICX,I,1),NOPS(L,K)/OPSLND)
00043      NJ=1
00044      NOPS(L,K)=AMAX1(NOPS(L,K)-A*OPSLND,0.)
00045      AIRUP(ICX,I,1) = AIRUP(ICX,I,1)-A
00046      IRR=IRRTIM(L,I)+1
00047      IRR = VTNJ(IRR,8)
00048      AIRDN(ICX,I,IRR)=AIRDN(ICX,I,IRR)+A
00049      200 CONTINUE
00050      IF(NJ.EQ.0) GO TO 400
00051      300 CONTINUE
00052      400 CONTINUE
00053      500 CONTINUE
00054      RETURN
00055      END
```

```
00001
00002      SUBROUTINE LAUNCH(IT, IDAY)
00003
00004      C THIS ROUTINE LAUNCHES AIRCRAFT TO FILL SCHEDULED MISSIONS
00005      C
00006      REAL LRFFF
00007      DOUBLE PRECISION DCOMPS,DPLEX,ORIG,TARG,TTS
00008      INTEGER DFIGIN,TARGET,START,STOP,UMAX,UMIN,UREADY
00009      INTEGER MASK,UGOT
00010      REAL LIVPLX,NOPS
00011      DIMENSION A(5),ANeed(5)
00012      COMMON/RLOCKB/DSUNIT(2,20,2),SCOMPS(2,20,5),COMPS(2,20,5),
00013      +STRAT(2,20,4),ADESAT(2,20,2),AGKILL(2,20,10),SPRATE(2,2,5),
00014      +DCOMPS(2,10),MAXR(2,20),LRDEF(2,20),WSTWX(2,20),BWDEF(2,20),
00015      +DRMAX(2,20),SPEED(2,20),IRRTIME(2,5),SR30MX(2,5),IMAXR(2,20),
00016      +ILRDEF(2,20),IWSTWX(2,20),IBWDEF(2,20),NCOMPS(2),NSUNIT(2)
00017      COMMON/RLOCKC/COMPLX(2,10,10),
00018      +DPLEX(2,10),DTSLI(2,10),SURFAC(2,10),DEFAC(2,10),ICYCLE(2,10),
00019      +IREPL(2,10),REPOPS(2,10,2),OPS(2,11),REPAIR(2,11),
00020      +NPLEX(2),FTIME(2),NDAYS,JWXDEF,JLRDEF,WXSURV,DELAY,
00021      +OPSLCH,OPSLND
00022      COMMON/RLOCKD/IONOFF(2,20,2),JONOFF(2,20,2),S1(2,20,2),S2(2,20,2),
00023      +MTS(2,20,2),UNIT(2,20,2),SA(2,20,2),SB(2,20,2),TTS(2,20),
00024      +RANGE(10,10),IRANGE(10,10),BMISS(2,20),IPRI(2,20),ORIG(2,20),
00025      +TARG(2,20),UMAX(2,20),UMIN(2,20),ORIGIN(2,20),TARGET(2,20),
00026      +IUNIT(2,20),START(2,20),STOP(2,20),UREADY(2,20),
00027      +DAY(30),WX(30),IWX(30),NMIS(2)
00028      COMMON/RLOCKE/LIVPLX(2,10,10),MASK(2,20,3),UGOT(2,20,3),
00029      +SR30(2,10,5),AIRUP(20,5,8),AIREN(20,5,8),ADBASE(2,10)
00030      COMMON/RLOCKF/SR(2,10,5),TPTY(2,9)
00031      COMMON/RLOCKH/ISCHED(2,20,8),IU(2,20),JU(2,20),NOPS(2,10)
00032      COMMON/RLOCKI/NMISUP(2,4),MISUP(2,20,4),NMUNUP(2,20,4),IREADY(2,20)
00033      COMMON /AIRFYP/ EXP(19,10)
00034
00035      C NMISUP=NO. OF MISSIONS ENGAGED IN CURRENT PERIOD
00036      C     IN=(1 SUCAP, 2 VFCAP, 3 SLI/DLI, 4 OFFENSIVE)
00037      C MISUP =MISSION NO. FOR EACH ENGAGED MISSION
00038      C NMUNUP=FORCE UNIT NO. FOR EACH ENGAGED MISSION
00039      C SR=DAILY SORTIE RATE
00040      C SR30=30 DAY UTILIZATION FOUND
00041      C ANFED=NO. OF FORCE UNITS NEEDED TO FILL MISSION
00042
00043      C                         MISSION PRIORITY
00044      C SURTIE RATE          1 OR 2           >2
00045      C LF NORMAL            FILL           FILL
00046      C NORMAL LF SURCE      FILL           NO
00047      C CT SURGE              NO            NO
00048
00049      C 30 DAY UTILIZATION NEVER EXCEEDED
00050
00051      DO 500 L=1,?
00052      IT=4
00053      1F(L.EQ.2) 11=F
00054      NM=NMISS(L)
00055      DO 100 IN=1,4
00056      NMISUP(L,IN)=0
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
00057      100 CONTINUE
00058          DO 400 I=1,9
00059          IF(IPTY(L,IP).NE.1) GO TO 400
00060          DO 300 M=1,NM
00061          IF(ISCHED(L,M,IT).NE.IP) GO TO 300
00062          IREADY(L,M)=0
00063          ITHRU=0
00064          IO=ORIGIN(L,M)
00065          ICX=IO
00066          IF(L.EQ.2) ICX=IO+NPLEX(1)
00067          IUN=IU(L,M)
00068          NUN= UMAX(L,M)
00069          180 IF(ITHRU.FQ.1) NUN=UREADY(L,M)
00070          IF(NUN.LE.0) GO TO 300
00071          190 OPNEED=0.0
00072          DO 200 I=1,II
00073          A(I)=SR(L,IO,I)
00074          ANEED(I)=NUN*SCOMPS(L,IUN,I)
00075          IF(ANEED(I).LT.0.001) GO TO 200
00076          OPNEED=OPNEED+ANEED(I)*OPSLCH
00077          IF(UNIT(L,M,1).EQ.5HSSM ) OPNEED=0.0
00078          IF(ANEED(I).GT.AIRDN(ICX,I,1)) GO TO 250
00079          IF(NOPS(L,IO).LT.OPNEED) GO TO 250
00080          IF(LIVPLX(L,IO,I).LT.0.001) GO TO 250
00081          A(I)= ANEED(I)/LIVPLX(L,IO,I)+SR(L,IO,I)
00082          IF(A(I).GT.SRATE(L,1,I).AND.IP.GT.2) GO TO 250
00083          IF(A(I).GT.SRATE(L,2,I)) GO TO 250
00084          IF(A(I)+SR30(L,IO,I).GT.SR30MX(L,I)) GO TO 250
00085          200 CONTINUE
00086          IF(L.EQ.2) GO TO 208
00087          IEX=-2
00088          DO 204 I=1,4
00089          IEX=IEX+4
00090          EXPS(IEX,TDAY)=FXPS(IEX,TDAY)+NUN*SCOMPS(1,IUN,I)
00091          204 CONTINUE
00092          208 CONTINUE
00093          IF(ITHRU.FQ.0) GO TO 210
00094          IREADY(L,M)=NUN
00095          GO TO 230
00096          210 IN=4
00097          IF( UNIT(L,M,1).EQ.5HSUCAP ) IN=1
00098          IF( UNIT(L,M,1).EQ.5HVFCAP ) IN=2
00099          IF( UNIT(L,M,1).EQ.5HSLI .OP. UNIT(L,M,1).EQ.5HDLI ) IN=3
00100          NMISUP(L,IN)=NMISUP(L,IN)+1
00101          N=N+NMISUP(L,IN)
00102          NMISUP(L,N,IN)=N
00103          NUNUP(L,M,IN)=NUN
00104          ISCHFD(L,M,IT)=0
00105          230 DO 240 I=1,II
00106          SR(L,IO,I)=A(I)
00107          IF(ANEED(I).LT.0.001) GO TO 240
00108          SR30(L,IO,I)=SR30(L,IO,I)+ANEED(I)/LIVPLX(L,IO,I)
00109          AIRDN(ICX,I,1)=AIRDN(ICX,I,1)-ANEED(I)
00110          240 CONTINUE
00111          K=JU(L,M)
00112          UGOT(L,M,K)=UGOT(L,M,K)+NUN
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
00113      NOPS(L,IO)=NOPSL,IO)-OPNEED
00114      ITHRU=ITHRU+1
00115      IF(ITHRU.GT.1) GO TO 300
00116      GO TO 180
00117      C
00118      C      TRY TO FILL WITH LESS UNITS DOWN TO MIN
00119      C
00120      250 NUN=NUN-1
00121      IF(NUN.LE.0) GO TO 300
00122      IF(NUN.LT. UMIN(L,M).AND.ITHRU.EQ.0) GO TO 300
00123      GO TO 190
00124      300 CONTINUE
00125      400 CONTINUE
00126      500 CONTINUE
00127      RETURN
00128      END
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
00001
00002      SUBROUTINE ENGAGE(IDAY)
00003  C
00004  C      THIS ROUTINE CONDUCTS THE ENGAGEMENT OF FORCES IN THE SCHEDULED MI
00005  C
00006      REAL LRDEF
00007      DOUBLE PRECISION PCOMPS,DPLEX,ORIG,TARG,TTS
00008      INTEGER ORIGIN,TARGET,START,STOP,UMAX,UMIN,UREADY
00009      INTEGER UASK,UGOT
00010      REAL LIVPLX,NOPS
00011      DIMENSION VF(2),VFKill(2),AIRGD(5),AIRLOS(5)
00012      COMMON/BLOCKB/DSUNIT(2,20,2),SCOMPS(2,20,5),COMPS(2,20,5),
00013      +STRAT(2,20,4),ADEFAT(2,20,2),ACKILL(2,20,10),SPATE(2,2,5),
00014      +DCOMPS(2,10),MAXP(2,20),LRDEF(2,20),WSTWX(2,20),BWDEF(2,20),
00015      +DRMAX(2,20),SPEED(2,20),IRRTIM(2,5),SR30MX(2,5),IMAXR(2,20),
00016      +ILRDEF(2,20),IWSTWX(2,20),IBWDEF(2,20),NCOMPS(2),NSUNIT(2)
00017      COMMON/BLOCKC/COMPLX(2,10,10),
00018      +DPLEX(2,10),DTSLI(2,10),SURFAC(2,10),DEFAC(2,10),ICYCLE(2,10),
00019      +IREPL(2,10),REPOPS(2,10,2),OPS(2,11),REPAIR(2,11),
00020      +NPLEX(2),NTIME(2),
00021          NDAYS,JWDEF,JLRDEF,WKSURV,DELAY,
00022      +OPSLCH,OPSLND
00023      COMMON/BLOCKD/JONOFF(2,20,2),JONOFF(2,20,2),S1(2,20,2),S2(2,20,2),
00024      +MTS(2,20,8),UNIT(2,20,2),SA(2,20,2),SR(2,20,2),TTS(2,20),
00025      +RANGE(10,10),IRANGE(10,10),DMISS(2,20),IPRI(2,20),ORIG(2,20),
00026      +TARG(2,20),UMAX(2,20),UMIN(2,20),ORIGIN(2,20),TARGET(2,20),
00027      +IUNIT(2,20),START(2,20),STOP(2,20),UREADY(2,20),
00028      +DAY(30),WK(30),NMIS(2)
00029      COMMON/BLOCKE/LIVPLX(2,10,10),UASK(2,20,3),UGOT(2,20,3),
00030      +SR30(2,10,5),AIRUP(20,5,8),AIRDN(20,5,8),ADBASE(2,10)
00031      COMMON/BLOCKH/ISCHED(2,20,8),IU(2,20),JU(2,20),NOPS(2,10)
00032      COMMON/BLOCKI/NMISUP(2,4),MISUP(2,20,4),NUNUP(2,20,4),IREADY(2,20)
00033      COMMON /AIREXP/ EXP(19,10)
00034  C
00035  C      SEND IN AN ATTACK
00036      DO 800 L=1,2
00037      LL=?
00038      IF(L.EQ.2) LL=1
00039      NM=NMISUP(L,4)
00040      IF(NM.EQ.0) GO TO 800
00041  C
00042  C      SEQUENTIALLY ENGAGE OFFENSIVE MISSIONS    BLUE FIRST
00043  C
00044      DO 700 J=1,NM
00045      M=MISUP(L,J,4)
00046      NUN=NUNUP(L,J,4)
00047      JIN=NIN
00048      UBACK=NUN
00049      IO=ORIGIN(L,M)
00050      IT=TARGET(L,M)
00051      ICO=IO
00052      ICT=NPLEX(1)+IT
00053      IF(L.EQ.2) ICO=NPLEX(1)+IO
00054      IF(L.EQ.2) ICT=IT
00055      IUN=IU(L,M)
00056      IC=0
```

```

00057      IF( L.EQ.1.OR.SCOMPS(L,TUN,5).EQ.0) GO TO 600
00058      C
00059      C      ENGAGE ATTACKING SSM SHIPS WITH SUCAP
00060      C
00061          ITM = IRANGE(IT,IO)
00062          ITM = MTIME(ITM) + 1
00063          ITM = MINO(ITM,8)
00064          IRR = MINO(9,IRRTIM(2,5)+1)
00065          N=1
00066          NMDS=MISUP(LL,N)
00067          IF(NMD.FQ.0) GO TO 600
00068          200 ITHRU=0
00069      C
00070      C      ENGAGE SSM ATTACK WITH EACH SUCAP AT TARGET
00071          NO CONSTRAINT ON GETTING OFF DECK LIKE DLI/SLI
00072      C      AS IN ALL CASES DO NOT USE MORE THAN IS REQUIRED
00073      C
00074          DO 500 K=1,NMD
00075          MD=MISUP(LL,K,N)
00076          IUD=IU(LL,MD)
00077          NUND=NUNUP(LL,K,N)
00078          JJUD=JU(LL,MD)
00079          IF(ORIGIN(LL,MD).NE.IT.OR.NUND.EQ.0) GO TO 500
00080          IF(ACKILL(LL,IUD,5).LT.0.00001) GO TO 500
00081          INEED=UBACK/AGKILL(LL,IUD,5)+0.999
00082          IF(INEED.LE.0) GO TO 500
00083          IUSED=MINO(INFED,NUND)
00084          ITHRU=1
00085          NUNUP(LL,K,N)=NUNUP(LL,K,N)-IUSED
00086          IR=MINO(IUSED,IREADY(LL,MD))
00087          NUNUP(LL,K,N)=NUNUP(LL,K,N)+IR
00088          IREADY(LL,MD)=IREADY(LL,MD)-IR
00089          UIIND=IUSFP
00090          UPMAX=9999.
00091          F=1.0
00092          IF(IWX(IDAY).EQ.1) F=WXSURV
00093          IF(SPEED(1,IUD).GT.0.0.AND.DTSLI(2,IO).GT.0.0)
00094          +UPMAX=((PRMAX(1,IUD)*SURFAC(2,IO)*F-25.)/SPEED(1,IUD)/12.0-DELAY)
00095          +/DTSLI(2,IO)+1.0
00096          UPMAX=AMAY1(0,C,UPMAX)
00097      C
00098      C      ENGAGE SUCAP WITH AIR AND SURFACE DEFENDERS
00099      C
00100          CALL DEFEND(LL,L,IO,IUD,IUSED,UPMAX,UIIND,VF,VFKILL,IDAD)
%FTNDIM LINE:00100 POSSIBLE TO INDEX MODIFICATION INSIDE LOOP
00101      C
00102      C          LL=DEFENDER  L=ATTACKER   IO=ORIGIN
00103      C
00104      C      ACCOUNT FOR SUCAP LOSSES AND SCHEDULE LANDING OF SURVIVING SUCAP
00105      C
00106          DO 450 II=1,4
00107      C
00108      C          OFFENSE LOSSES ACCOUNTED FOR IN DEFEND
00109      C
00110          ALOST=(IUSED-UIIND)*SCOMPS(1,IUD,II)
00111          ALAND=UIIND*SCOMPS(1,IUD,II)

```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
00112      LIVPLX(1,IT,II)=LIVPLX(1,IT,II)-AMIN1(ALOST,LIVPLX(1,IT,II))
00113      AIRUP(1CT,II,ITM)=AIRUP(1CT,II,ITM)+ALAND
00114      450 CONTINUE
00115      C
00116      C      COMPUTE AIR-TO-GROUND KILLS BY SUCAP
00117      C
00118      IF(IC.EQ.0) UIN=UIN-AMIN1(UIN,UIND*AGKILL(1,IUD,5))
00119      UBACK=UBACK-AMIN1(UBACK,UIND*AGKILL(1,IUD,5))
00120      C
00121      C      ACCOUNT FOR DEFENDERS KILLED BY SUCAP
00122      C
00123      LIVPLX(2,IO,3) = LIVPLX(2,IO,3) - VFKILL(1)
00124      LIVPLX(2,IO,4) = LIVPLX(2,IO,4) - VFKILL(2)
00125      EXPFS(18, IDAY)=EXPFS(18, IDAY)+VFKILL(1)
00126      EXPFS(19, IDAY)=EXPFS(19, IDAY)+VFKILL(2)
00127      AIRUP(ICO,3,1) = AIRUP(ICO,3,1) + VF(1) - VFKILL(1)
00128      AIRUP(ICO,4,1) = AIRUP(ICO,4,1) + VF(2) - VFKILL(2)
00129      IF(UBACK.LT.0.001) GO TO 550
00130      500 CONTINUE
00131      IC=1
00132      IF(ITHRU.EQ.1) GO TO 200
00133      C
00134      C      ACCOUNT FOR SSM SHIP LOSSES AND SCHEDULE RETURN OF SURVIVING SSM S
00135      C
00136      550  AIRDN(ICO,5,IRR)=AIRDN(ICO,5,IRR)+UBACK*SOMPS(2,IUN,5)
00137      LIVPLX(2,IO,5)=LIVPLX(2,IO,5)-(NUN-UBACK)*SOMPS(2,IUN,5)
00138      C
00139      C      ENGAGE ATTACKING AIRCRAFT AND SSM WITH AIR AND SURFACE DEFENSES
00140      C
00141      C      VFKILL=FIGHTERS LOST DEFENDING
00142      C      TAIRCD=TOTAL A/C ON GROUND
00143      C      TAIRLS=TOTAL A/C LOST (DISTRIBUTE WITH AIRLOS)
00144      C      A/C LOSS RULES
00145      C          1. IN RR AND LEAST READY
00146      C          2. READY FOR SLI CAP OR SUCAP
00147      C          3. SLT/DLI NOT USED IN DEFENSE
00148      C          THESE RULES APPLY TO EACH TYPE A/C WHICH ARE LOST
00149      C          BASED ON RELATIVE NUMBER PRESENT FOR LOSS
00150      C
00151      600 UPMAX=9999.
00152      F=1.0
00153      IF(IWX(IDAY).EQ.1) F=WXSURV
00154      IF(SPEED(L,IUN).GT.0.0.AND.DTSLT(LL,IT).GT.0.0)
00155      +UPMAX=((DFMAX(L,IUN)*SURFAC(LL,IT)*F-25.)/SPEFD(L,IUN)/12.0-DELAY)
00156      +/DTSLI(LL,IT)+1.0
00157      UPMAX=AMAX1(0.0,UPMAX)
00158      CALL DEFEND(L,LL,IT,IUN,NUN,UPMAX,UIN,VF,VFKILL,IDAY)
00159      LINE:00159 POSSIBLE DO INDEX MODIFICATION INSIDE LOOP
00159      C
00160      C      ACCOUNT FOR AIRCRAFT KILLED IN DEFENSE AND ATTACKERS KILLED
00161      C
00162      C      SCHEDULE LANDING OF SURVIVORS
00163      C
00164      DO 620 I=1,4
00165      C
00166      C      OFFENSIVE LOSSES ACCOUNTED FOR IN DEFEND
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
00167 C
00168 C      LIVPLX(L,IO,I)=LIVPLX(L,IO,I)-(NUN-UIN)*SCOMPS(L,IUN,I)
00169 C      IC1=IO
00170 C      IC2=IT
00171 C      IF(L.EQ.2) IC1=IT
00172 C      IF(L.EQ.2) IC2=IO
00173 C      ITM=IRANGE(IC1,IC2)
00174 C      IL=MIN0(MTIME(ITM),7)
00175 C      AIRUP(ICO,I,IL+1)=AIRUP(ICO,I,IL+1)+UIN*SCOMPS(L,IUN,I)
00176 C      IF(I.NE.3.AND.I.NE.4) GO TO 620
00177 C
00178 C      ACCOUNT FOR DFFNSE FIGHTER LOSS
00179 C
00180 C      IF(LL.EQ.2) GO TO 610
00181 C      IEX=11
00182 C      IF(I.EQ.4)IFX=15
00183 C      EXP(S(IEX,1DAY)=EXP(S(IEX,1DAY)+VFKILL(I-2)
00184 C      GO TO 612
00185 610    CONTINUE
00186 C      IEX=13
00187 C      IF(I.EQ.4) IEX=19
00188 C      EXP(S(IEX,1DAY)=EXP(S(IEX,1DAY)+VFKILL(I-2)
00189 612    CONTINUE
00190 C      LIVPLX(LL,IT,I)=LIVPLX(LL,IT,I)-VFKILL(I-2)
00191 C      AIRUP(ICK,I,1)=AIRUP(ICK,I,1)+VF(I-2)-VFKILL(I-2)
00192 620    CONTINUE
00193 C
00194 C      COMPUTE NUMBER OF AIRCRAFT ON THE GROUND
00195 C
00196 C      TAIRD=0.0
00197 C      DO 650 I=1,4
00198 C      AIRGD(I)=0.0
00199 C      DO 630 IJ=1,8
00200 C      AIRGD(I)=AIRGD(I)+AIREM(ICK,I,IJ)
00201 630    CONTINUE
00202 C      DO 640 N=1,3
00203 C      NMD=NMSUP(LL,N)
00204 C      IF(NMD.EQ.0) GO TO 640
00205 C      DO 635 K=1,NMD
00206 C      MD=MISUP(LL,K,N)
00207 C      IUD=IU(LL,MD)
00208 C      NUND=NUNUF(LL,K,N)
00209 C      NNN=0
00210 C      IF(N.EQ.3) NNN=NUND
00211 C      IF(ORIGIN(LL,MF).NE.IT.OP.NUND.EQ.0) GO TO 635
00212 C      AIRGD(I)=AIRGD(I)+SCOMPS(LL,IUD,I)*(NNN+IREADY(LL,MD))
00213 635    CONTINUE
00214 640    CONTINUE
00215 C      TAIRD=TAIRD+AIRGD(I)
00216 650    CONTINUE
00217 C
00218 C      COMPUTE AIR TO GROUND ATTRITION
00219 C
00220 C      TAIRLS= AVIN1(AGKILL(L,IUN,4)*UIN,TAIRD*(1.0-DEFAC(LL,T)))
00221 C      NC=NCOMPS(LL)
00222 C      IEX=1
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
00223      DO 680 I=1,NC
00224      IF(I.LE.4) GO TO 670
00225      LIVPLX(LL,IT,I)=LIVPLX(LL,IT,I)-AMIN1(LIVPLX(LL,IT,I),
00226      +UIN*ASKILL( L,IUN,I))
00227      GO TO 680
00228      670 AIRLOS(I)=0.0
00229      IF(TAIRGD.GT.0.001)
00230      +AIRLOS(I)=AIRGD(I)/TAIRGD *TAIPLS
00231      IF(LL.EQ.2) GO TO 675
00232      IEX=IEX+4
00233      GO TO 678
00234      675 CONTINUE
00235      IEX=18
00236      IF(I.EQ.4)IEX=19
00237      679 CONTINUE
00238      EXPSC(IEX, IDAY)=EXPSC(IEX, IDAY)+*
00239      1          AMIN1(LIVPLX(LL,IT,I),AIRLOS(I))
00240      LIVPLX(LL,IT,I)=LTVPPLX(LL,IT,I)-AMIN1(LIVPLX(LL,IT,I),ATRLOS(I))
00241      680 CONTINUE
00242      C
00243      C ACCOUNT FOR PARKED AIRCRAFT ATTRITED
00244      C
00245      DO 695 I=1,4
00246      SUB=AIRLOS(I)
00247      IF(SUB.LT.0.0001) GO TO 695
00248      DO 685 IJ=1,3
00249      JI=9-IJ
00250      A=AMIN1(SUB,AIRDN(ICK,I,JI))
00251      AIRDN(ICK,I,JI)=AIRDN(ICK,I,JI)-A
00252      SUB=SUB-A
00253      IF(SUB.LT.0.0001) GO TO 695
00254      685 CONTINUE
00255      DO 690 NN=1,3
00256      N=4-NN
00257      NM0=NMSUP(LL,N)
00258      IF(NM0.EQ.0) GO TO 690
00259      DO 687 K=1,NM0
00260      MD=MISUP(LL,K,N)
00261      IUD=IU(LL,MD)
00262      IF(ORIGIN(LL,MD).NE.IT) GO TO 687
00263      IF(SCOMPS(LL,IUD,I).LT.0.001) GO TO 687
00264      ISUR=SUB/SCOMPS(LL,IUD,I)+.999
00265      IB=MINO(IREADY(LL,MD),ISUB)
00266      A=AMIN1(SUB,IB*SCOMPS(LL,IUD,I))
00267      SUB=SUB-A
00268      IREADY(LL,MD)=IREADY(LL,MD)-IB
00269      AIRDN(ICK,I,1)=AIRDN(ICK,I,1)+IB*SCOMPS(LL,IUD,I)-A
00270      DO 696 II=1,4
00271      IF(II.EQ.1) GO TO 686
00272      AIRDN(ICK,II,1)=AIRDN(ICK,II,1)+IB*SCOMPS(LL,IUD,II)
00273      685 CONTINUE
00274      IF(SUB.LT.0.0001) GO TO 695
00275      687 CONTINUE
00276      690 CONTINUE
00277      N=3
00278      NM0=NMSUP(LL,N)
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
00279      IF(NMD.EQ.0) GO TO 695
00280      DO 694 K=1,NMD
00281      MD=MISUP(LL,K,N)
00282      IUD=IU(LL,MD)
00283      NUND=NUNUP(LL,K,N)
00284      IF(ORIGIN(LL,MD).NE.IT) GO TO 694
00285      IF(SCOMPS(LL,IUD,I).LT.0.001) GO TO 694
00286      ISUB=SUP/SCOMPS(LL,IUD,I)+.999
00287      IB=MIN0(NUND,ISUB)
00288      A=AMIN1(SUB,IB*SCOMPS(LL,IUD,I))
00289      NUNUP(LL,K,N)=NUNUP(LL,K,N)-IB
00290      AIRDN(ICK,I,1)=AIRDN(ICK,I,1)+IB*SCOMPS(LL,IUD,I)-A
00291      SUB=SUB-A
00292      DO 692 II=1,4
00293      IF(II.EQ.I) GO TO 692
00294      AIRDN(ICK,II,1)=AIRDN(ICK,II,1)+IB*SCOMPS(LL,IUD,II)
00295      692 CONTINUE
00296      IF(SUB.LT.0.0001) GO TO 695
00297      694 CONTINUE
00298      695 CONTINUE
00299      700 CONTINUE
00300      800 CONTINUE
00301      C
00302      C      SCHEDULE LANDING OF UNUSED SUCAP AND CAP AND RELEASE SLI AND READY
00303      C
00304      DO 980 L=1,2
00305      DO 950 N=1,2
00306      NM=NMISUP(L,N)
00307      IF(NM.EQ.0) GO TO 950
00308      DO 940 K=1,NM
00309      M=MISUP(L,K,N)
00310      IO=ORIGIN(L,M)
00311      IP=IO
00312      IF(L.EQ.2) IP=NPLEX(1)+IO
00313      IUN=IU(L,M)
00314      NUN=NUNUP(L,K,N)
00315      NNU=0
00316      NND=0
00317      IF(N.EQ.3) NND=NUN
00318      IF(N.NE.3) NNU=NUN
00319      DO 930 I=1,4
00320      AIRDN(IP,I,1)=AIRDN(IP,I,1)+(NND+IREADY(L,M))*SCOMPS(L,IUN,I)
00321      AIRUP(IP,I,1)=AIRUP(IP,I,1)+NNU*SCOMPS(L,IUN,I)
00322      930 CONTINUE
00323      940 CONTINUE
00324      950 CONTINUE
00325      980 CONTINUE
00326      RETURN
00327      END
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
00001
00002      SUBROUTINE DEFEND(L,LL,IT,IUN,NUN,UPMAX,UIN,VF,VFKILL,IDAY)
00003      C
00004      C   THIS ROUTINE CONDUCTS THE AIR DEFENSE ENGAGEMENT OF INCOMING ATTACK
00005      C
00006      DOUBLE PRECISION DCOMPSS,DPLEX,ORTG,TARG,TTS
00007      REAL LRDEF
00008      INTEGER ORIGIN,TARGET,START,STOP,UMAX,UMIN,UREADY
00009      INTEGER UASK,UCOT
00010      REAL LIVPLX,NOPS
00011      DIMENSION VF(2),VFKILL(2)
00012      COMMON/FLOCKB/DSUNIT(2,20,2),SCOMPSS(2,20,5),COMPSS(2,20,5),
00013      +STRAT(2,20,4),ADEFAT(2,20,2),AGKILL(2,20,10),SRATE(2,2,5),
00014      +DCOMPSS(2,10),MAXR(2,20),LRDFF(2,20),WSTWX(2,20),BWDEF(2,20),
00015      +DRMAX(2,20),SPEED(2,20),IRRTIM(2,5),SR30MX(2,5),IMAXR(2,20),
00016      +ILRDEF(2,20),IWSTWX(2,20),IBWDFF(2,20),NCOMPSS(2),NSUNIT(2)
00017      COMMON/BLOCKC/COMPLX(2,10,10),
00018      +DPLEX(2,10),TSLI(2,10),SURFAC(2,10),DEFAC(2,10),ICYCLE(2,10),
00019      +IREPL(2,10),RFPOPS(2,10,2),OPS(2,11),REPAIR(2,11),
00020      +NPLEX(2),NTIME(2),NDAYS,JWXDEF,JLRDEF,WXSURV,DELAY,
00021      +OPSLCH,OPSLND
00022      COMMON/BLOCKD/JONOFF(2,20,2),JONOFF(2,20,2),S1(2,20,2),S2(2,20,2),
00023      +MTS(2,20,8),UNIT(2,20,2),SA(2,20,2),SB(2,20,2),TTS(2,20),
00024      +RANGE(10,10),IPANGE(10,10),DMISS(2,20),IPRI(2,20),ORIG(2,20),
00025      +TARG(2,20),UMAX(2,20),UMIN(2,20),ORIGIN(2,20),TARGET(2,20),
00026      +IUNIT(2,20),START(2,20),STOP(2,20),UREADY(2,20),
00027      +DAY(30),WX(30),IWX(30),NMIS(2)
00028      COMMON/BLOCKE/LIVPLX(2,10,10),UASK(2,20,3),UGOT(2,20,3),
00029      +SR30(2,10,5),AIRUP(20,5,8),AIRDN(20,5,3),ADBSE(2,10)
00030      COMMON/BLOCKH/ISCHED(2,20,8),IU(2,20),JU(2,20),NOPS(2,10)
00031      COMMON/BLOCKI/NMISUP(2,4),MISUP(2,20,4),NUNUP(2,20,4),IREARY(2,20)
00032      COMMON /AIREXP/ EXPSS(19,10)
00033      C
00034      C   UIN=UPDATED COUNT OF SURVIVING ATTACKING UNITS
00035      C   AA=NO. OF ATTACKING UNITS THAT CAN BE KILLED PER DEFENSE UNIT(AIR
00036      C   IUSFD=NO. OF DEFENSIVE UNITS ENGAGED
00037      C
00038      VF(1)=0.0
00039      VF(2)=0.0
00040      VFKILL(1)=0.0
00041      VFKILL(2)=0.0
00042      IC=0
00043      100 ITTHRU=0
00044      DO 600 N=2,3
00045      NMD=NUNUP(LL,N)
00046      IF(NMD.EQ.0) GO TO 550
00047      DO 500 K=1,NMD
00048      IF(UIN.LT.0.0001) GO TO 700
00049      MD=MISUP(LL,K,N)
00050      IUD=IU(LL,MD)
00051      NUND=NUNUP(LL,K,N)
00052      JJUD=JU(LL,MD)
00053      IUMAX=NUND
00054      IF(IC.EQ.0) GO TO 450
00055      AIK=0.0
00056      DO 400 IT=1,4
```

```

00057      AIR=AIR+SCOMPS(LL,IUD,1)
00058      400 CONTINUE
00059      IUMAX=UPMAX/ATR
00060      450 IF(ORIGIN(LL,MD).NE.IT.OR.NUND.FQ.0.OR.IUMAX.LF.0) GO TO 500
00061      AA=SCOMPS(LL,IUD,3)*STRAT(L,IUN,1)+SCOMPS(LL,IUD,4)*STRAT(L,IUN,2)
00062      IF(AA.LT.0.000001) GO TO 500
00063      INEED=UIN/AA+0.99
00064      IF(INEED.LE.0) GO TO 500
00065      IUSED=MIN0(INEED,NUND,IUMAX)
00066      IF(IC.EQ.1) UPMAX=UPMAX-IUSED*ATR
00067      ITHRU=1
00068      NUNUP(LL,K,N)=NUNUP(LL,K,N)-IUSED
00069      IR=MIN0(IUSED,IREADY(LL,MD))
00070      NUNUP(LL,K,N)=NUNUP(LL,K,N)+IR
00071      IREADY(LL,MD)=IREADY(LL,MD)-IR
00072          AALOSS=UIN
00073      UIN=UIN-AMIN1(AA*IUSED,UIN)
00074          AALOSS=AALOSS-UIN
00075          IEX=-1
00076          DO 455 III=1,4
00077          IEX=IEX+4
00078          IF(L.EQ.2) IEX=18
00079          IF(L.EQ.2 .AND. III.EQ.4)IEX=19
00080          EXP(S(IEX, IDAY)=EXP(S(IEX, IDAY)+AALOSS*SCOMPS(L,IUN,III)
00081      455 CONTINUE
00082          VF(1)=VF(1)+SCOMPS(LL,IUD,3)*IUSED
00083          VF(2)=VF(2)+SCOMPS(LL,IUD,4)*IUSED
00084          IF(UIN.LT.0.0001) GO TO 700
00085      500 CONTINUE
00086      550 IC=1
00087      600 CONTINUE
00088          IF(ITHRU.FQ.1) GO TO 100
00089      C
00090      C      COMPUTE SURFACE-TO-AIR LOSSES
00091      C
00092      700 I1=5
00093          I2=6
00094          IF(LL.EQ.2) I1=7
00095          IF(LL.EQ.2) I2=8
00096          GALOSS=UIN
00097          UIN=UIN-AMIN1(UIN,STRAT(L,IUN,3)*LIVPLX(LL,IT,I1))
00098          UIN=UIN-AMIN1(UIN,STRAT(L,IUN,4)*LIVPLX(LL,IT,I2))
00099          CALOSS=GALOSS-UIN
00100          IEX=0
00101          DO 705 III=1,4
00102          IEX=IEX+4
00103          IF(L.EQ.2) IEX=18
00104          IF(L.EQ.2 .AND. III.EQ.4)IEX=19
00105          EXP(S(IEX, IDAY)=EXP(S(IEX, IDAY)+
00106          1          GALOSS*SCOMPS(L,IUN,III)
00107      705 CONTINUE
00108          IF((VF(1)+VF(2)).LT.0.001) RETURN
00109      C
00110      C      COMPUTE AIR TO AIR LOSS OF DEFENDERS
00111      C
00112      VFKTLL(1)=AMIN1(VF(1),VF(1)/(VF(1)+VF(2))*ADEFAT(L,IUN,1)*NUN)

```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

00113 VFKILL(2)=AMIN1(VF(2),VF(2)/(VF(1)+VF(2))*ADEFAT(L,IUN,2)*NUN)
00114 RETURN
00115 END

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
00001      SUBROUTINE SAVED(ITABLE)
00002      C
00003      C   THIS ROUTINE SAVES INPUT TABLES ON PERM FILE LOGICAL UNIT 21
00004      C
00005      DOUBLE PRECISON INBUF,NAME
00006      DOUBLE PRECISION ITABLE
00007      DOUBLE PRECISION IUTYPE,IETYPE,IUSUB,IESUB,ICMPLX,MISS
00008      COMMON /IFACE/ IUTYPE(2,15),IETYPE(2,15),IUSUB(2,15,15),
00009      +                  IESUB(2,15,15),ICMPLX(2,8),MISS(2,17)
00010      DIMENSION ITABLE(14,16,17)
00011      C
00012      C-----OPEN OUTPUT DESTINATION FILE.
00013          WRITE(5,1000)
00014          READ(5,1001) NAME
00015          OPEN(UNIT=21,FILE=NAME,ACCESS='SEQOUT',MODE='BINARY')
00016          100 WRITE(21) IUTYPE,IETYPE,IUSUB,IESUB,ICMPLX,MISS,ITABLE
00017          CALL RELEAS(21)
00018          WRITE(5,1054)
00019          1054 FORMAT(1HO,'FILE HAS BEEN SAVED.')
00020          CALL DLAY(2)
00021          RETURN
00022      C
00023      C-----FORMATS
00024          1000  FORMAT($,' ENTER OUTPUT FILE NAME.(6 CHAR MAX) ')
00025          1001  FORMAT(F10)
00026          END
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
00001      SUBROUTINE SETUP(ITABLE)
00002      C
00003      C      THIS ROUTINE LOADS INPUT TABLES FROM PERM FILE LOGICAL UNIT 20
00004      C
00005      DOUBLE PRECISION INBUF,NAME
00006      DOUBLE PRECISION ITABLE,ITEMP,NULL
00007      DOUBLE PRECISION IUTYPE,IETYPE,IUSUB,IESUB,ICMPLX,MISS
00008      COMMON /IFACE/ IUTYPE(2,15),IETYPE(2,15),IUSUB(2,15,15),
00009      +                   IESUB(2,15,15),ICMPLX(2,8),MISS(2,17)
00010      COMMON /MAX/ MAXCMP,MAXUNT,MAXNIS,NIT,MAXFLE,MAXROW,NULL
00011      COMMON /CURTAB/ FORIND,FORM(24,80)
00012      INTEGER FORMNO,FORM,UFLD
00013      DIMENSION ITABLE(14,16,17),ITEMP(19,17)
00014      DATA UFLD/"725004020100/
00015      C
00016      C-----OPEN INPUT SOURCE FILE.
00017      WRITE(5,1000)
00018      READ(5,1001) NAME
00019      OPEN(UNIT=20,FILE=NAME,ACCESS="SEQIN",MODE="BINARY")
00020      C
00021      C
00022      100 READ (20) IUTYPE,IETYPE,IUSUB,IESUB,ICMPLX,MISS,ITABLE
00023      CALL PELEAS(20)
00024      C
00025      C      DECODE VARIABLES FOR RATTLE VARIABLES
00026      C
00027      DO 200 I=1,NIT
00028      DO 198 J=1,MAXFLE
00029      DO 193 K=1, MAXROW
00030      198 ITEMP(J,K)= ITABLE(I,J,K)
00031      FORMNO = I
00032      DO 199 IROW=1,24
00033      INX = IROW+24*(I-1)
00034      READ(23#INX,1002)(FORM(IROW,ICOL),ICOL=1,80)
00035      199 CONTINUE
00036      CALL FORMS(UFLD)
00037      CALL DCODE(I,ITEMP)
3FTNDIM LINE:00037 POSSIBLE DO INDEX MODIFICATION INSIDE LOOP
00038      200 CONTINUE
00039      WRITE(5,1004)
00040      1004 FORMAT(1H0,"FILE HAS BEEN LOADED.")
00041      CALL QLBY(2)
00042      RETURN
00043      C
00044      C-----FORMATS
00045      1000  FORMAT($," ENTER INPUT FILE NAME.(6 CHAR MAX)   ")
00046      1001  FORMAT(A10)
00047      1002  FORMAT(P0A1)
00048      END
```

AD-A061 363

SRI INTERNATIONAL MENLO PARK CA NAVAL WARFARE RESEAR--ETC F/G 15/7
STRIKE OUTCOME CALCULATOR (SOC)--DESCRIPTION, AND OPERATING INS--ETC(U)
MAR 78 R S GARNERO, J C BOBICK, D AYERS N00014-75-C-0742

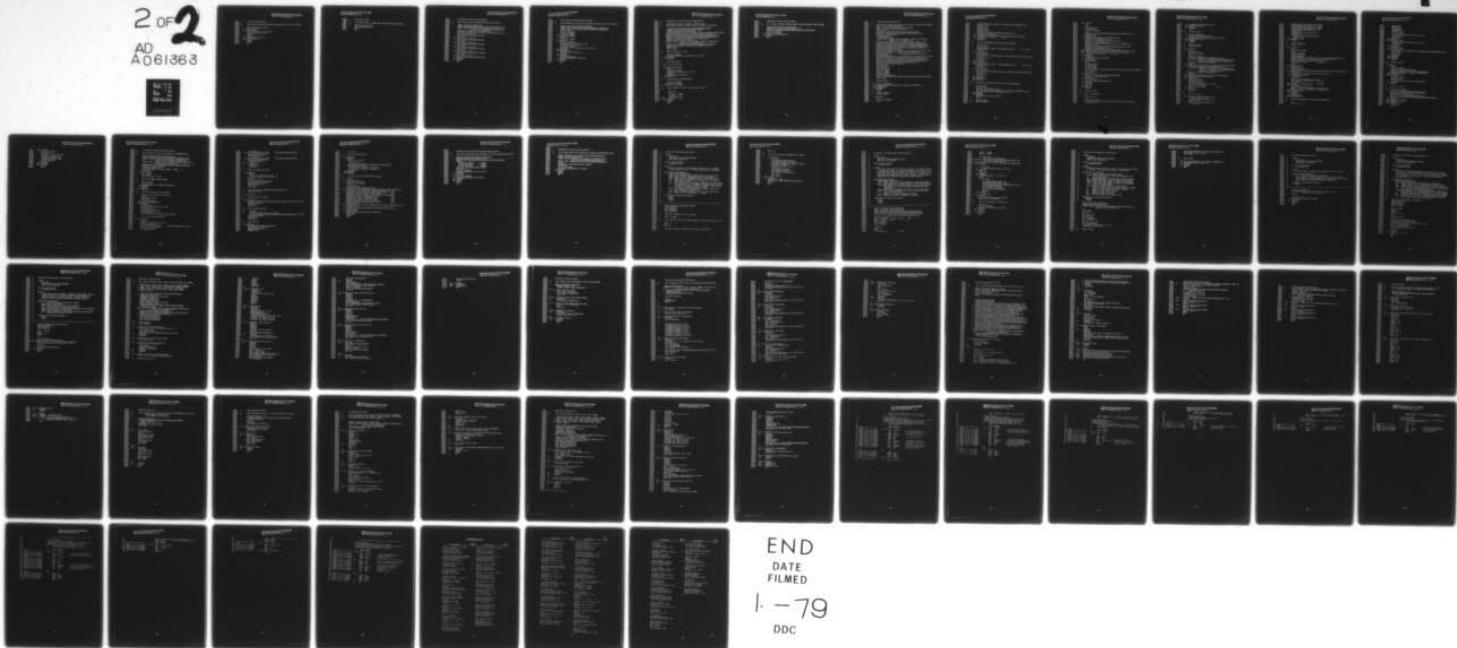
UNCLASSIFIED

NWRC-TR-15

NL

2 OF 2
AD
A061363

EEF
EAT



END
DATE
FILED

1 - 79
DDC

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
00001      SUBROUTINE GETINT(IVAL)
00002      C
00003      C   THIS ROUTINE CHECKS AND CONVERTS RESPONSE TO EIGHT ADJ. INTEGER
00004      C
00005      DOUBLE PRECISION IV,V,IVT
00006      READ(5,4000)IV
00007      4000 FORMAT(A10)
00008      CALL NFIELD(IV,1,10,IVT,IERR)
00009      IF(IERR.EQ.0) GO TO 100
00010      IVAL=-1
00011      RETURN
00012      100 DECODE(10,4001,IVT) V
00013      4001 FORMAT(F10.0)
00014      IVAL=V
00015      RETURN
00016      END
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
00001      SUBROUTINE DUMRD
00002  C
00003  C      THIS ROUTINE DOES A DUMMY READ TO REGAIN PROGRAM CONTROL
00004  C
00005      DOUBLE PRECISION ITEMP
00006      CALL SCRNI(0,ITEMP)
00007      RETURN
00008      END
```

```
00001      SUBROUTINE SUBPRO(IT,ITEMP,ITABLE)
00002      C
00003      C      THIS ROUTINE DOES SUBSEQUENT PROCESSING FOR TABLES
00004      C
00005      DOUBLE PRECISION ITABLE,ITEMP
00006      DOUBLE PRECISION IUTYPE,IETYPE,IUSUB,IESUB,ICMPLX,MISS
00007      COMMON /IFACE/ IUTYPE(2,15),IETYPE(2,15),IUSUB(2,15,15),
00008      +                  IESUB(2,15,15),ICMPLX(2,8),MISS(2,17)
00009      DIMENSION ITABLE(14,16,17),ITEMP(19,17)
00010      IF(IT.EQ.1.OR.IT.EQ.6.OR.IT.EQ.7.OR.IT.GE.10)GO TO 390
00011      IF(IT.GT.2) GO TO 310
00012      CALL ULIST(0,IUTYPE,IUSUB,ITEMP)
00013      GO TO 390
00014      310 IF(IT.GT.3) GO TO 320
00015      CALL ULIST(1,IUTYPE,IUSUB,ITEMP)
00016      GO TO 390
00017      320 IF(IT.GT.4) GO TO 330
00018      CALL ULIST(0,IETYPE,IESUB,ITEMP)
00019      GO TO 390
00020      330 IF(IT.GT.5) GO TO 340
00021      CALL ULIST(1,IETYPE,IESUB,ITEMP)
00022      GO TO 390
00023      340 IF(IT.GT.8) GO TO 350
00024      CALL CLIST(0,ICMPLX,ITEMP,ITABLE)
00025      GO TO 390
00026      350 IF(IT.GT.9) GO TO 390
00027      CALL CLIST(1,ICMPLX,ITEMP,ITABLE)
00028      390 CONTINUE
00029      RETURN
00030      END
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
00001      SUBROUTINE ULIST(IRED,NYPE,NUB,ITEMP)
00002      C
00003      C      THIS ROUTINE BUILDS LISTS OF RED/BLUE UNITS FOR UNIT AND FNG TABLES
00004      C
00005      DOUBLE PRECISION NULL,NYPE,NUB,ITEMP
00006      COMMON /MAX/ MAXCNP,MAXUNT,MAXMIS,NIT,MAXFLD,MAXROW,NULL
00007      DIMENSION NYPE(2,15),NUB(2,15,15),ITEMP(19,17),NNUB(15)
00008      IND= 1 +IPED
00009      DO 100 J=1,MAXUNT
00010      NYPE(IND,J)=NULL
00011      DO 100 K=1,MAXUNT
00012      100 NUB(IND,J,K)=NULL
00013      NNYPE=0
00014      DO 102 I=1,MAXUNT
00015      102 NNUB(I)=0
00016      DO 400 J=1,MAXUNT
00017      IF(ITEMP(1,J).EQ.NULL) GO TO 400
00018      IF(NNYPE.EQ.0) GO TO 200
00019      DO 150 JJ=1,NNYPE
00020      IF(ITEMP(1,J).EQ.NYPE(INT,JJ)) GO TO 250
00021      150 CONTINUE
00022      200 NNYPE=NNYPE+1
00023      NYPE(IND,NNYPE)=ITEMP(1,J)
00024      JJ=NNYPE
00025      250 CONTINUF
00026      NNUB(JJ)=NNUB(JJ)+1
00027      NUB(IND,JJ,NNUB(JJ))=ITEMP(2,J)
00028      400 CONTINUE
00029      RETURN
00030      END
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
00001      SUBROUTINE CLIST(IRED,ICMPLX,IITEMP,ITABLE)
00002      C
00003      C      THIS ROUTINE BUILDS RED/BLUE COMPLEX LISTS
00004      C      AND RESETS RELATIVE POSITION TABLE IF COMPLEX LIST CHANGES
00005      C
00006      INTEGER ORIGIN,TARGET,START,STOP,UMAX,UMIN,UREADY
00007      DOUBLE PRECISION NULL,ICMPLX,IITEMP,ITABLE
00008      DOUBLE PRECISION TTS,ORIG,TARG
00009      COMMON/PLOCKD/IONOFF(2,20,2),JONOFF(2,20,2),S1(2,20,2),S2(2,20,2),
00010      +MTS(2,20,8),UNIT(2,20,2),SA(2,20,2),SB(2,20,2),TTS(2,20),
00011      +RANGE(10,10),IRANGE(10,10),DMISS(2,20),IPRI(2,20),ORIG(2,20),
00012      +TARG(2,20),UMAX(2,20),UMIN(2,20),ORIGIN(2,20),TARGET(2,20),
00013      +IUNIT(2,20),START(2,20),STOP(2,20),UREADY(2,20),
00014      +DAY(30),WX(30),IWX(30),NMISS(2)
00015      COMMON /SPECs/ ITABFR, ITABLR, ITABNR, ITABNF, IFLDS(14,15),
00016      1          IBLNK(15,17), IFORMT(2,19)
00017      COMMON /MAX/ MAXCMP,MAXUNT,MAXMIS,NIT,MAXFLD,MAXROW,NULL
00018      DIMENSION ICMPLX(2,8),ITFMP(19,17),ITABLE(14,16,17)
00019      INDT=8+IRED
00020      IND= 1 +IRED
00021
00022      C      CHECK FOR CHANGE TO LIST
00023      C
00024      DO 100 J=1,MAXCMP
00025      IF(ITABLE(INDT,1,J).NE.IITEMP(1,J)) GO TO 200
00026      100 CONTINUE
00027      GO TO 300
00028
00029      C      LIST HAS CHANGED
00030      C
00031      200 DO 250 J=1,MAXCMP
00032
00033      250 ICMPLX(IND,J)=NULL
00034      NC=0
00035      DO 260 J=1,MAXCMP
00036      IF(IITEMP(1,J).EQ.NULL) GO TO 260
00037      NC=NC+1
00038      ICMPLX(IND,NC)=IITEMP(1,J)
00039      260 CONTINUE
00040
00041      C      BLANK RELATIVE POSITION TABLE
00042      C
00043      DO 270 J=1,MAXCNP
00044      DO 270 K=1,MAXCNP
00045      270 ITABLE(13,J,K)=NULL
00046
00047      C      RESET INDEXING IN TABLE 13(RFL POS NOW LONG)
00048      C
00049      NPS = 8
00050      NPR = 8
00051      DO 275 J = 1,NPR
00052      DO 275 K = 1,NPR
00053      IRANGE(K,J) = 1
00054      275   RANGE(K,J) = NULL
00055      300 CONTINUE
00056      RETURN
00057      END
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
00001      SUBROUTINE TRSFN(IT,ITABLE,ITEMP)
00002      C
00003      C      THIS ROUTINE REPLACES CURRENT INPUT TABLE WITH ERROR FREE NEW TABL
00004      C
00005      DOUBLE PRECISION NULL,ITABLE,ITEMP
00006      DIMENSION ITABLE(14,16,17),ITEMP(19,17)
00007      COMMON /MAX/ MAXCNP,MAXUNT,MAXMIS,NIT,MAXFLD,MAXROW,NULL
00008      DO 100 J=1,MAXFLD
00009      DO 100 K=1,MAXROW
00010      100 ITABLE(IT,J,K)=ITEMP(J,K)
00011      RETURN
00012      END
```

```

00001      SUBROUTINE DCODE(ITABL,JEMP)
00002      C
00003      C THIS ROUTINE DECODES AND PROCESSES A TABLE OF DATA INTO VARIABLES
00004      C NEEDED FOR COMPUTING OUTCOMES
00005      DOUBLE PRECISION DCOMPS,DPLEX,ORIG,TARG,TTS
00006      DOUBLE PRECISION JEMP(19,17),NULL10
00007      REAL LRDEF
00008      INTEGER ORIGIN,TARGET,START,STOP,UMAX,UMIN,UREADY
00009      DIMENSION ESUNIT(2,20,2),ETRAT(2,20,4),EDEFAT(2,20,2),
00010      +EGKILL(2,20,10),NESUNT(2)
00011      DIMENSION IN(14)
00012      COMMON//ENDC/JENDC
00013      COMMON //SPECs// ITABFR, ITABLR, ITABNR, ITABNF, IFLDS(14,15),
00014      1          IBLNK(15,17), TFORMT(2,19)
00015      COMMON//BLOCKB/DSUNIT(2,20,2),SCOMPS(2,20,5),COMPS(2,20,5),
00016      +STRAT(2,20,4),ADEFAT(2,20,2),AGKILL(2,20,10),SRATE(2,2,5),
00017      +DCOMPS(2,10),MAXR(2,20),LRDEF(2,20),WSTWX(2,20),BWDEF(2,20),
00018      +DRMAX(2,20),SPEED(2,20),IRRTIM(2,5),SR30MX(2,5),IMAXR(2,20),
00019      +ILRDEF(2,20),IWSTWX(2,20),IBWDEF(2,20),NCOMPS(2),NSUNIT(2)
00020      COMMON//BLOCKC/COMPLX(2,10,10),
00021      +DPLEX(2,10),DTSLI(2,10),SURFAC(2,10),DEFAC(2,10),ICYCLE(2,10),
00022      +IREPL(2,10),REPOPS(2,10,2),OPS(2,11),REPAIR(2,11),
00023      +NPLEX(2),NTIME(2),NDAYS,JWXDEF,JLRDEF,WXSURV,DELAY,
00024      +OPSLCH,OPSLND
00025      COMMON//BLOCKD/IONOFF(2,20,2),JONOFF(2,20,2),S1(2,20,2),S2(2,20,2),
00026      +MTS(2,20,8),UNIT(2,20,2),SA(2,20,2),SR(2,20,2),TTS(2,20),
00027      +RANGE(10,10),TRANGE(10,10),DMISS(2,20),IPRI(2,20),ORIG(2,20),
00028      +TARG(2,20),UMAX(2,20),UMIN(2,20),ORIGIN(2,20),TARGET(2,20),
00029      +IUNIT(2,20),START(2,20),STOP(2,20),UREADY(2,20),
00030      +DAY(30),WX(30),IWX(30),NMISS(2)
00031      DATA IN/14*0/
00032      DATA NCOMPS/6,8/
00033      DATA NULL/1H /
00034      DATA NULL10/10H /
00035      NLINES = ITABNR
00036      NFLDS = ITABNF
00037      NCHARS=NFLDS*10
00038      IN(ITABL)=IN(ITABL)+1
00039      GO TO (100,200,200,400,400,600,700,800,800,1000,1100,1100,1300,
00040      +1400) ITABL
00041      C
00042      C      PROCESS TABLE 1
00043      C
00044      100 DO 150 J=1,NLINES
00045      DECODE(NCHARS,9010,JEMP(1,J)) DCOMPS(1,J),DCOMPS(2,J)
00046      9010 FORMAT(2A10)
00047      150 CONTINUE
00048      RETURN
00049      C
00050      C      PROCESS TABLE 2
00051      C      PROCESS TABLE 3
00052      C
00053      200 L=ITABL-1
00054      I=0
00055      DO 205 J=1,NLINES
00056      I=I+1

```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
00057      DECODE(NCHARS,9020,JEMP(1,J)) DSUNIT(L,I,1),DSUNIT(L,I,2),  
00058      +COMPS(L,I,1),  
00059      +COMPS(L,I,2),  
00060      +COMPS(L,I,3),  
00061      +COMPS(L,I,4),  
00062      +COMPS(L,I,5),  
00063      + MAXR(L,I),LRDEF(L,I),WSTWX(L,I),BWDEF(L,I),  
00064      +DRMAX(L,I), SPEED(L,I)  
00065  9020 FORMAT(A5,5X,A3,7X,5(F3.0,7X),4(A4,6X),F5.0,5X,F6.0,4X)  
00066      IF(DSUNIT(L,I,1).EQ.5H      ) I=I-1  
00067      205 CONTINUE  
00068      NSUNIT(L)=I  
00069      C  
00070      C      REPLACE ALPHANUMERIC WITH NUMERIC DESCRIPTORS  
00071      C  
00072      NS=NSUNIT(L)  
00073      DO 265 J=1,NS  
00074      IF(MAXR(L,J).NE.5H)LONG .AND.MAXR(L,J).NE.5H      ) GO TO 210  
00075      IMAXR(L,J)=1  
00076      GO TO 230  
00077      210 IMAXR(L,J)=2  
00078      DO 220 K=1,NS  
00079      IF(DSUNIT(L,J,1).NE.DSUNIT(L,K,1).OR.DSUNIT(L,K,2).NE.LRDEF(L,J))  
00080      +GO TO 220  
00081      ILRDEF(L,J)=K  
00082      GO TO 240  
00083      220 CONTINUE  
00084      230 ILRDEF(L,J)=0  
00085      240 IF(WSTWX(L,J).NE.5H)RAD .AND.WSTWX(L,J).NE.5H      ) GO TO 250  
00086      IWSTWX(L,J)=1  
00087      GO TO 260  
00088      250 IWSTWX(L,J)=2  
00089      DO 255 K=1,NS  
00090      IF(DSUNIT(L,J,1).NE.DSUNIT(L,K,1).OR.DSUNIT(L,K,2).NE.BWDEF(L,J))  
00091      +GO TO 255  
00092      IBWDEF(L,J)=K  
00093      GO TO 265  
00094      255 CONTINUE  
00095      260 IBWDEF(L,J)=0  
00095      265 CONTINUE  
00097      C  
00098      C      ALL-WEATHER ATTACK AIRCRAFT ARE USED FOR SUPPRESSION  
00099      C  
00100      DO 280 J=1,NS  
00101      DO 270 I=1,5  
00102      SCOMPS(L,J,I)=COMPS(L,J,I)  
00103      IF(I.EQ.2.AND.L.EQ.1) SCOMPS(L,J,2)=COMPS(L,J,2)+COMPS(L,J,5)  
00104      IF(I.EQ.5.AND.L.EQ.1) SCOMPS(L,J,5)=0.0  
00105      270 CONTINUE  
00106      280 CONTINUE  
00107      IF(IN(ITABL+2).GT.0) GO TO 415  
00108      RETURN  
00109      C  
00110      C      PROCESS TABLE 4  
00111      C      PROCESS TABLE 5  
00112      C
```

```
00113    400 L=ITABL-3
00114    I=0
00115    LL=2
00116    IF(L.EQ.2) LL=1
00117    NT=NCOMP3(LL)
00118    DO 410 J=1,NLINES
00119    I=I+1
00120    IF(L.EQ.1) DECODE(NCHARS,9040,JFMP(1,J)) ESUNIT(L,I,1),
00121    +ESUNIT(L,I,2),
00122    +ETRAT(L,I,1),
00123    +ETRAT(L,I,2),ETRAT(L,I,3),ETRAT(L,I,4),
00124    +EDEFAT(L,I,1),EDEFAT(L,I,2),
00125    +EGKILL(L,I,4),EGKILL(L,I,5),EGKILL(L,I,6)
00126    +,EGKILL(L,I,7),EGKILL(L,I,8)
00127    IF(L.EQ.2) DECODE(NCHARS,9050,JEMP(1,J)) ESUNIT(L,I,1),
00128    +ESUNIT(L,I,2),
00129    +ETRAT(L,I,1),ETRAT(L,I,2),ETRAT(L,I,3),ETRAT(L,I,4),
00130    +EDEFAT(L,I,1),EDEFAT(L,I,2),
00131    +EGKILL(L,I,4),EGKILL(L,I,5),EGKILL(L,I,6)
00132    9040 FORMAT(A5,5X,A3,7X,4(F4.0,6X),F7.0,3X,F8.0,2X,5(F4.0,6X))
00133    9050 FORMAT(A5,5X,A3,7X,4(F4.0,6X),2(F8.0,2X),3(F7.0,3Y))
00134    IF(ESUNIT(L,I,1).EQ.5H      ) I = I-1
00135    410 CONTINUE
00136    NES=NT(L)=I
00137    IF(IN(ITABL-2).EQ.0) RETURN
00138    415 NS=NSUNIT(L)
00139    NES=NESUNT(L)
00140    LL=2
00141    IF(L.EQ.2) LL=1
00142    NT=NCOMP3(LL)
00143    DO 450 J=1,NS
00144    DO 445 K=1,NES
00145    IF(DSUNIT(L,J,1).EQ.ESUNIT(L,K,1).AND.DSUNIT(L,J,2).EQ.ESUNIT(L,K,
00146    +2)) GO TO 430
00147    GO TO 445
00148    430 DO 435 KK=1,4
00149    IF(KK.LT.2) ADEFAT(L,J,KK)=EDEFAT(L,K,KK)
00150    STRAT(L,J,KK)=ETRAT(L,K,KK)
00151    435 CONTINUE
00152    DO 440 KK=4,NT
00153    AGKILL(L,J,KK)=EGKILL(L,K,KK)
00154    440 CONTINUE
00155    GO TO 450
00156    445 CONTINUE
00157    450 CONTINUE
00158    RETURN
00159    C
00160    C      PROCESS TABLE 6
00161    C
00162    600 DO 650 J=1,NLINES
00163    L=1
00164    IF(J.GT.4) L=2
00165    K=J
00166    IF(L.EQ.2) K=J-4
00167    DECODE(NCHARS,9060,JFMP(1,J)) SRATE(L,1,K),SRATE(L,2,K),A
00168    1ERTIV(L,K)=A+.5
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
00169      SR30MX(L,K)=SRATE(L,1,K)*30.0
00170      9060 FORMAT(3(F6.0,4X))
00171      650 CONTINUE
00172      RETURN
00173      C
00174      C      PROCESS TABLE 7
00175      C
00176      700 DO 750 J=1,NLINES
00177      DECODE(NCHARS,9070,JEMP(1,J))
00178      +OPS(1,J),OPS(2,J),REPAIR(1,J),REPAIR(2,J)
00179      9070 FORMAT(4(F7.0,3X))
00180      750 CONTINUE
00181      RETURN
00182      C
00183      C      PROCESS TABLE 8
00184      C      PROCESS TABLE 9
00185      C
00186      800 L=ITABL-7
00187      I=0
00188      NC=NCOMPSS(L)
00189      DO 850 J=1,NLINES
00190      I=I+1
00191      DPLEX(L,I)=NULL10
00192      IF(L.EQ.1) DECODE(NCHARS,9080,JEMP(1,J))
00193      +DPLEX(L,I),COMPLX(L,I,1),COMPLX(L,I,2),COMPLX(L,I,3),
00194      +      COMPLX(L,I,4),COMPLX(L,I,5),COMPLX(L,I,6),
00195      +DTSLI(L,I),SURFAC(L,I),DEFAC(L,I
00196      +),A,B,C
00197      9080 FORMAT(A6,4X,5(F4.0,6X),F6.0,4X,3(F5.0,5X),3(F4.0,6X))
00198      IF(L.EQ.2) DECODE(NCHARS,9090,JEMP(1,J))
00199      + DPLEX(L,I),COMPLX(L,I,1),COMPLX(L,I,2),COMPLX(L,I,3)
00200      +      ,COMPLX(L,I,4),COMPLX(L,I,5),COMPLX(L,I,6),
00201      +      COMPLX(L,I,7),COMPLX(L,I,8),
00202      +DTSLI(L,I),SURFAC(L,I),DEFAC(L,I
00203      +),A,B,C
00204      9090 FORMAT(A6,4X,8(F3.0,7X),3(F5.0,5X),3(F4.0,6X))
00205      DECODE(10,9091,JEMP(16,J)) C
00206      9091 FORMAT(F10.0)
00207      ICYCLE(L,I)=A+.50
00208      IREPL(L,I)=B+.50
00209      REPOPS(L,I,2)=NULL
00210      IF(C.LT.0.0000) REPOPS(L,I,2)=1HF
00211      REPOPS(L,I,1)=ABS(C)
00212      IF( DPLFX(L,I).EQ.10H ) I=I-1
00213      850 CONTINUE
00214      NPLEX(L)=I
00215      RETURN
00216      C
00217      C      PROCESS TABLE 10
00218      C
00219      1000 DECODE(NCHARS,9100,JEMP(1,1)) A
00220      NDAYS= MIN1(A+.5,10.)
00221      DECODE(NCHARS,9100,JEMP(1,2)) A
00222      MTIME(1)=A+.5
00223      DECODE(NCHARS,9100,JEMP(1,3)) A
00224      MTIME(2)=A+.5
```

```
00225      DECODE(NCHARS,9100,JEMP(1,4)) OPSLND
00226      DECODE(NCHARS,9100,JEMP(1,5)) OPSLCH
00227      DECODE(NCHARS,9100,JEMP(1,6)) WXSURV
00228      DECODE(NCPARS,9100,JEMP(1,7)) DFLAY
00229      DECODE(NCHARS,9100,JEMP(1,8)) A
00230      JLRDEF=A+.1
00231      DECODE(NCPARS,9100,JEMP(1,9)) A
00232      JWXDEF=A+.1
00233          DECODE(NCHARS,9100,JEMP(1,10)) A
00234          JENDC = A+.1
00235 9100 FORMAT(F4.0,6X)
00236      RETURN
00237      C
00238      C      PROCESS TABLE 11
00239      C      PROCESS TABLE 12
00240      C
00241 1100 L=ITABL-10
00242      I=0
00243      DO 1150 J=1,NLINES
00244      I=I+1
00245      ORIG(L,I)=NULL10
00246      TARG(L,I)=NULL10
00247      TTS(L,I)=NULL10
00248      DECODE(NCHARS,9110,JEMP(1,J)) DMISS(L,I),IPRI(L,I),ORIG(L,I),
00249      +TARG(L,I),S1(L,I,1),S2(L,I,1),S1(L,I,2),S2(L,I,2),
00250      +TTS(L,I),UNIT(L,I,1),UNIT(L,I,2),A,B,C
00251      IF(DMISS(L,I).NE.5H) GO TO 1120
00252      I=I-1
00253      GO TO 1150
00254 1120 UMAX(L,I)=A+.5
00255      UMIN(L,I)=B+.5
00256      UREADY(L,I)=C+.5
00257 9110 FORMAT(A5,5X,I1,9X,2(A6,4X),A3,A5,2X,A3,A5,2X,A8, 2X,A5,5X,A3,7X,
00258      +3(F3.0,7X))
00259      DECODE(30,9111,JEMP(13,J))S1(L,I,1),S1(L,I,2)
00260 9111 FORMAT(A3,17X,A3,7X)
00261      DO 1145 N=1,2
00262      JJ=N*2+12
00263      IF(S1(L,I,N).EQ.3HEND) GO TO 1140
00264      DECODE(10,9112,JEMP(JJ,J)) S2(L,I,N)
00265      GO TO 1145
00266 1140 DECODE(10,9113,JEMP(JJ,J)) S2(L,I,N)
00267 1145 CONTINUE
00268      DECODE(10,9114,TTS(L,I))
00269      +MTS(L,I,1),MTS(L,I,2),MTS(L,I,3),MTS(L,I,4),
00270      +MTS(L,I,5),MTS(L,I,6),MTS(L,I,7),MTS(L,I,8)
00271 9114 FORMAT(8I1,2X)
00272 1150 CONTINUE
00273      NMIS(L)=I
00274 9112 FORMAT(5X,A5)
00275 9113 FORMAT(A5,5X)
00276      C
00277      C      REPLACE ALPHANUMERIC WITH NUMERIC DESCRIPTORS
00278      C
00279      LL=?
00280      IF(L.EQ.2) LL=1
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
00281      NM=NMISS(L)
00282      NO=NPLEX(L)
00283      NT=NPLEX(LL)
00284      NU=NSUNIT(L)
00285      DO 1235 M=1,NM
00286      DO 1205 K=1,NO
00287      IF(ORIG(L,M).NE.DPLEX(L,K)) GO TO 1205
00288      ORIGIN(L,M)=K
00289      GO TO 1210
00290      1205 CONTINUE
00291      1210 DO 1215 K=1,NT
00292      IF(TARG(L,M).NE.DPLEX(LL,K)) GO TO 1215
00293      TARGET(L,M)=K
00294      GO TO 1220
00295      1215 CONTINUE
00296      TARGET(L,M)=0
00297      1220 DO 1230 J=1,NU
00298      IF(UNIT(L,M,1).NE.DSUNIT(L,J,1).OR.UNIT(L,M,2).NE.DSUNIT(L,J,2))
00299      +GO TO 1230
00300      IUNIT(L,M)=J
00301      GO TO 1235
00302      1230 CONTINUE
00303      1235 CONTINUE
00304      RETURN
00305      C
00306      C      PROCESS TABLE 13
00307      C
00308      1300 NPR=8
00309      NPR=8
00310      DO 1350 J=1,NPR
00311      DECODE(NCHARS,9130,JEMP(1,J))
00312      +RANGE(1,J),RANGE(2,J),RANGE(3,J),RANGE(4,J),
00313      +RANGE(5,J),RANGE(6,J),RANGE(7,J),RANGE(8,J)
00314      9130 FORMAT(8(A5,5X))
00315      1350 CONTINUE
00316      DO 1360 J=1,NPP
00317      DD 1360 K=1,NPR
00318      IRANGE(J,K)=1
00319      IF( RANGE(J,K).EQ.5HSHORT) IRANGE(J,K)=2
00320      1360 CONTINUE
00321      RETURN
00322      C
00323      C      PROCESS TABLE 14
00324      C
00325      1400 DO 1450 J=1,NLINES
00326      IF(J.EQ.1) DECODE(NCHARS,9140,JEMP(1,J))
00327      +DAY(1),DAY(2),DAY(3),DAY(4),DAY(5),DAY(6),
00328      +DAY(7),DAY(8),DAY(9),DAY(10)
00329      IF(J.EQ.2) DECODE(NCHARS,9141,JEMP(1,J))
00330      +WX(1),WX(2),WX(3),WX(4),WX(5),WX(6),WX(7),WX(8),WX(9),
00331      +WX(10)
00332      9140 FORMAT(10(F5.0,5X))
00333      9141 FORMAT(10(A5,5X))
00334      1450 CONTINUE
00335      DO 1460 K=1,30
00336      IWX(K)=2
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
00337 1460 CONTINUE
00338 DO 1480 K=1,NFLDS
00339 ID=DAY(K)+.5
00340 IF(ID.GT. 30) GO TO 1480
00341 IF(ID.EQ.0) GO TO 1480
00342 IF(WX(K).EQ.4HBAD) IW=1
00343 IF(WX(K).EQ.4HGOOD) IW=2
00344 DO 1470 KK=ID,30
00345 IWX(KK)=IW
00346 1470 CONTINUE
00347 1480 CONTINUE
00348 RETURN
00349 END
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
00001      SUBROUTINE ERROR(IT,ITFMP,IERR,INTER)
00002      C
00003      C   THIS ROUTINE SCREENS DATA ENTRIES FOR INCONSISTANCIES
00004      C
00005      DOUBLE PRECISION NULL,ITEMP,IVAL,IWDS
00006      DOUBLE PRECISION IUTYPE,IETYPE,IUSUB,IESUB,ICMPLX,MISS
00007      COMMON /IFACE/ IUTYPF(2,15),IETYPE(2,15),IUSUB(2,15,15),
00008      +                   IESUB(2,15,15),ICMPLX(2,8),MISS(2,17)
00009      COMMON /SPECs/ ITABFR, ITABLR, ITABNR, ITABMF, IFLDS(14,15),
00010      1                   IBLNK(15,17), IFORMT(2,19)
00011      COMMON /MAX/ MAXCMP,MAXUNT,MAXMIS,NIT,MAXFLD,MAXROW,NULL
00012      DIMENSION IBADS(9),IWDS(4)
00013      DIMENSION ITEMP(19,17)
00014      DATA IWDS/6HLONG ,6HSHORT ,6HGOOD ,6HBAD /
00015      DO 100 I=1,9
00016      100 IBADS(I)=0
00017      NR = ITAPNR
00018      NF = ITABMF
00019      IF(IERR.EQ.0) GO TO 200
00020      C
00021      C   COUNT NON NUMERIC ENTRY ERRORS
00022      C
00023      DO 110 J=1,NF
00024      DO 110 J=1,NR
00025      IF(ILNFK(I,J).EQ.1) IBADS(9)=IBADS(9)+1
00026      110 CONTINUE
00027      IERR=0
00028      200 CONTINUE
00029      C
00030      C   CONSTRUCT MISSION LISTS IF NEEDED
00031      C
00032      IF(IT.LE.10.OR.IT.GE.13) GO TO 400
00033      IRED=0
00034      IF(IT.EQ.12) IRED=1
00035      IND=IRED+1
00036      DO 350 I=1,MAXMIS
00037      350 MISS(IND,I)=ITEMP(1,I)
00038      400 CONTINUE
00039      DO 990 J=1,NR
00040      IF(ITEMP(1,J).EQ.NULL) GO TO 990
00041      DO 900 I=1,NF
00042      ISEP=IFLPS(IT,I)-1
00043      IF(ISEP.LE.0) GO TO 900
00044      IVAL=ITFMP(I,J)
00045      GO TO (810,820,830,840,850,860,870),ISEP
00046      C
00047      C   CHECK FORCE UNIT DEFINITION
00048      C
00049      *10 CONTINUE
00050      IRED=0
00051      IF(IT.EC.3.OR.IT.FC.12) IRED=1
00052      IF(IT.GT.3) G3 TO *15
00053      CALL CKTS(ITEMP(1,J),IVAL      ,IRED,IUTYPE,IUSUB,IERR,IT)
00054      IF(IEPP.FC.0) GO TO 900
00055      IBADS(1)=IBADS(1)+1
00056      IFLNK(I,J) = 1
```

```
00057      GO TO 900
00058      815 CALL CKTS(ITEMP(8,J),IVAL      ,IRED,IUTYPE,IUSUB,IERR,IT)
00059      IF(IERR.EQ.0) GO TO 816
00060      IBADS(1)=IBADS(1)+1
00061      GO TO 818
00062      816 CALL CKTS(ITEMP(8,J),IVAL      ,IRED,IETYPE,IESUB,IERR,IT)
00063      IF(IERR.EQ.0) GO TO 900
00064      IBADS(2)=IBADS(2)+1
00065      818   CONTINUE
00066      IBLNK(I,J)=1
00067      IBLNK(I-1,J) = 1
00068      GO TO 900
00069      C
00070      C      CHECK FOR COMPLEX ON LIST
00071      C
00072      820 CONTINUE
00073      IRED=0
00074      IF(IT.EQ.11.AND.I.EQ.4) IRED = 1
00075      IF(IT.EQ.12.AND.I.EQ.3) IRED = 1
00075      IBOK=0
00077      IF(I.EQ.4) IBOK=1
00078      CALL CKC(IVAL,IRED,IERR,IBOK)
00079      IF(IERR.EQ.0) GO TO 900
00080      IBADS(3)=IBADS(3)+1
00081      IBLNK(I,J)=1
00082      GO TO 900
00083      C
00084      C      CHECK FOR MISSION PRESENCE ON APPROPRIATE LIST
00085      C      DONE WHEN ISEP=7
00086      C
00087      830 GO TO 900
00088      C
00089      C      CHECK FOR LONG OR SHORT
00090      C
00091      840 CONTINUE
00092      IF(IVAL.EQ.IWDS(1).OR.IVAL.EQ.IWDS(2).OR.IVAL.EQ.NULL) GO TO 900
00093      IBADS(5)=IBADS(5)+1
00094      IBLNK(I,J)=1
00095      GO TO 900
00096      C
00097      C      CHECK FOR GOOD OR BAD
00098      C
00099      850   CONTINUE
00100      IF(IT.EQ.14.AND.J.EQ.1) GO TO 990
00101      IF(IVAL.EQ.IWDS(3).OR.IVAL.EQ.IWDS(4).OR.IVAL.EQ.NULL) GO TO 900
00102      IBADS(6)=IBADS(6)+1
00103      IBLNK(I,J)=1
00104      GO TO 900
00105      C
00106      C      CODE REPL OPS
00107      C
00108      860 CONTINUE
00109      CALL ERR1(IVAL,ITEMP(16,J),IERR)
00110      IF(IERR.EQ.0) GO TO 900
00111      IBADS(7)=IBADS(7)+1
00112      IBLNK(I,J)=1
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDG

```
00113      GO TO 900
00114      C
00115      C      CHECK START/STOP FIELD
00116      C
00117      870 CONTINUE
00118      IRED=0
00119      IF(IT.EQ.12) IRED=1
00120      KK = 13
00121      IF(I.EQ.6) KK = 15
00122      CALL ERP?(IVAL,ITEMP(KK,J),ITEMP(KK+1,J),IERR,IRED)
00123      IF(IERR.EQ.0) GO TO 900
00124      IF(IERR.EQ.1) IBADS(8) = IBADS(8)+1
00125      IF(IERR.EQ.2) IBADS(4) = IBADS(4) + 1
00126      IBLNK(I,J)=1
00127      900 CONTINUE
00128      990 CONTINUE
00129      C
00130      C      IF ERRORS HAVE BEEN FOUND WRITE MESSAGES
00131      C
00132      IBADT=0
00133      DO 995 I=1,9
00134      995 IBADT=IBADT+IBADS(I)
00135      IF(IBADT.EQ.0) RETURN
00136      IERR=1
00137      CALL NWSCFN(0)
00138      WRITE(5,4000)IT,(IBADS(I),I=1,9)
00139      4000 FORMAT(33H INPUT ERRORS WERE FOUND IN TABLE,I3,14X, 6HNUMBER/
00140      148H0FORCE UNIT NOT DEFINED IN UNIT DEFINITION TABLE,I7/
00141      248H0FORCE UNIT NOT DEFINED IN ENGAGEMENT TABLE ,I7/
00142      348H0COMPLEX NOT DEFINED IN COMPLEX TABLE ,I7/
00143      448HOMISSION NAME IN START OR STOP FIELD NOT DEFINED,I7/
00144      548H0ILLEGAL ENTRY IN LONG/SHORT FIELD ,I7/
00145      648H0ILLEGAL ENTRY IN GOOD/BAD FIELD ,I7/
00146      748H0ILLEGAL ENTRY IN REPL OPS FIELD ,I7/
00147      848H0ILLEGAL ENTRY IN START OR STOP FIELD ,I7/
00148      948H0NON NUMERIC ENTRY IN NUMERIC FIELD ,I7/)
00149      WRITE(5,1022)
00150      1022 FORMAT(64H WHEN THE TABLE REAPPEARS THE FIELDS IN ERROR WILL BE IN
00151      1DICATED./1H ,
00152      253HYOU MAY CORRECT THE ERRORS OR RESTORE ORIGINAL TABLE.)
00153      WRITE(5,2022)
00154      2022 FORMAT($,1H0,
00155      136HTO RETURN TO TABLE STRIKE SPACE BAR.)
00156      CALL DUMRD
00157      RETURN
00158      END
```

```
00001      SUBROUTINE CKTS(NYPE,NUP,IRED,INYPE,INUB,IERR,IT)
00002
00003      C
00004      C   THIS ROUTINE VALIDATES THE PRESENCE OF A UNIT ON APPROPRIATE LIST
00005      C
00006      DOUBLE PRECISION NYPE,NUP,INYPE,INUB,NULL
00007      DIMENSION INYPE(2,15),INUB(2,15,15)
00008      COMMON /MAX/ MAXCNP,MAXUNT,MAXNIS,NIT,MAXFLD,MAXROW,NULL
00009      IERR=0
00010      IF(NYPE.EQ.10HVFCAP      ) RETURN
00011      IF(NYPE.EQ.10HDLI       ) RRETURN
00012      IF(NYPE.EQ.10RSLI       ) RETURN
00013      IF(IT.LE.3.AND.NUB.EQ.NULL) RETURN
00014      IND=IRED+1
00015      DO 100 J=1,MAXUNT
00016      IF(NYPE.EQ.INYPE(IND,J)) GO TO 200
00017 100  CONTINUE
00018      GO TO 900
00019      200 DO 300 K=1,MAXUNT
00020      IF(NUB.EQ.INUB(IND,J,K)) RETURN
00021      300 CONTINUE
00022      900 IERR=1
00023      RETURN
00024      END
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
00001
00002      SUBROUTINE CKC(ICOMP,IRED,IEPR,IBOK)
00003      C
00004      C      THIS ROUTINE CHECKS FOR PRESENCE OF COMPLEX ON APPROPRIATE LIST
00005      C
00006      DOUBLE PRECISION IUTYPE,IETYPE,IUSUB,IESUB,ICMPLX,MISS
00007      DOUBLE PRECISION NULL,ICOMP
00008      COMMON /IFACE/ IUTYPE(2,15),IETYPE(2,15),IUSUB(2,15,15),
00009      +                  IESUB(2,15,15),ICMPLX(2,8),MISS(2,17)
00010      COMMON /MAX/ MAXCMP,MAXUNT,MAXMIS,MIT,MAXFLD,MAXROW,NULL
00011      IERR=0
00012      IF(IBOK.EQ.1.AND.ICOMP.EQ.NULL) RETURN
00013      IND=IRED+1
00014      DO 100 J=1,MAXCMP
00015      IF(ICOMP.EQ.ICMPLX(IND,J)) RETURN
00016      100 CONTINUE
00017      IERR=1
00018      RETURN
00019      END
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

00001
00002 SUBROUTINE ERP1(IWORD,JWORD,IERP)
00003 C
00004 C AUTHOR
00005 C BOB RINGO
00006 C STANFORD RESEARCH INSTITUTE/NWRC
00007 C MENLO PARK CALIFORNIA
00008 C
00009 C DATE OF LAST REVISION
00010 C 1 DECEMBER 1976
00011 C
00012 C PURPOSE
00013 C TO RIGHT JUSTIFY A LEFT JUSTIFIED NUMERIC FIELD. IF INPUT
00014 C FIELD CONTAINS A *D*, A (-) SIGN IS PREFIXED TO OUTPUTWORD.
00015 C
00016 C DESCRIPTION OF VARIABLES
00017 C IBLANK-BLANK FILLED WORD
00018 C IBLNK -RIGHT ADJUSTED BLANK CHARACTER IN ZERO FILLED WORD
00019 C ICHAR -RIGHT ADJUSTED CHARACTER IN ZERO FILLED WORD
00020 C IDEE -RIGHT ADJUSTED CHARACTER *D* IN A ZERO FILLED WORD
00021 C IERR -ERROR TAG--(0) SUBROUTINE PROCESSED OK,(1) ERROR IN
00022 C INPUT WORD *IWORD*
00023 C INUM -TAG INDICATING (0) NO NUMERIC CHARACTER HAS BEEN FOUND
00024 C IN *IWORD* OR (1) FIRST NUMERIC CHARACTER HAS BEEN FOUND
00025 C IS -CHARACTER POSITION IN *IWORD* INDICATING CHARACTER TO
00026 C BE EXAMINED AND TRANSFERRED TO *JWORD*
00027 C IWORD -LEFT ADJUSTED NUMERIC CHARACTER FIELD(4 BYTES) THAT
00028 C MAY CONTAIN THE CHARACTER *D*
00029 C MINUS -RIGHT ADJUSTED CHARACTER (-) IN A ZERO FILLED WORD
00030 C NDS -NUMBER OF *D*'S ENCOUNTERED IN *IWORD*
00031 C
00032 C FUNCTIONS OR SUBROUTINES REQUIRED
00033 C GETCHA
00034 C PUTCHA
00035 C NUMCHK
00036 C
00037 C-----
00038 C
00039 C DOUBLE PRECISION IWORD,JWORD,IBLANK
00040 C DATA IBLNK/1H /
00041 C DATA IBLANK/6H /
00042 C DATA MINUS/1H-/
00043 C DATA IDEE/1HD/
00044 C
00045 C INITIALIZE OUTPUT WORD TO ALL BLANKS
00046 C
00047 C JWORD = IBLANK
00048 C
00049 C SET ERROR TAG, NO. OF *D*'S ENCOUNTERED, AND FIRST DIGIT TAG TO ZERO
00050 C
00051 C IERR = 0
00052 C NDS = 0
00053 C INUM = 0
00054 C
00055 C INITIALIZE STARTING CHARACTERS IN *IWORD* AND *JWORD*
00056 C

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
00057      IS = 5
00058      JS = 10
00059      C
00060      C      LOOP OVER FIRST FOUR CHARTRS OF *IWORD*
00061      C
00062      DO 100 I=1,4
00063          IS=IS-1
00064          CALL GETCHA(ICHAR,IWORD,IS)
00065          IF(ICHAR.EQ.IBLNK) GO TO 100
00066          IF(ICHAR.EQ.IDEE.AND.INUM.EQ.1) IERR = 1
00067          IF(IERR.EQ.1) RETURN
00068          IF(ICHAR.EQ.IDEE) GO TO 50
00069          INUM = 1
00070          CALL NUMCHK(ICHAR,IERR)
00071          IF(IERR.EQ.1) RETURN
00072          CALL PUTCHA(ICHAR,JWORD,JS)
00073          JS = JS - 1
00074          GO TO 100
00075      50      NDS = NDS + 1
00076      100 CONTINUE
00077          IF(NDS.GT.1) IERR = 1
00078          IF(NDS.EQ.1) CALL PUTCHA(MINTS,JWORD,JS)
00079          RETURN
00080          END
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

00001
00002 SUBROUTINE ERR2(IWORD,JWORD,KWORD,IERR,IRED)
00003 C
00004 C AUTHOR
00005 C BOB RINGO
00006 C STANFORD RESEARCH INSTITUTE/NWRC
00007 C MENLO PARK, CALIFORNIA
00008 C
00009 C DATE OF LAST REVISION
00010 C 1 DECEMBER 1976
00011 C
00012 C PURPOSE
00013 C TO DIVIDE INPUT WORD *IWORD* INTO 2 WORDS. THE FIRST 3 CHARAC
00014 C OF *IWORD* ARE PLACED IN *JWORD* BLANK FILLED. IF THE FIRST 3
00015 C CHARACTERS OF *IWORD* ARE *END*, THE NEXT 5 CHARACTERS OF *IWORD*
00016 C ARE LEFT JUSTIFIED BLANK FILLED INTO *KWORD*. IF THE FIRST 3
00017 C CHARACTERS OF *IWORD* ARE NOT *END*, THE NEXT 5 CHARACTERS OF
00018 C *IWORD* ARE RIGHT JUSTIFIED BLANK FILLED INTO *KWORD*.
00019 C
00020 C DESCRIPTION OF VARIABLES
00021 C IBLANK-BLANK FILLED WORD
00022 C IBLNK -BLANK CHARACTER, RIGHT JUSTIFIED IN ZERO FILLED WORD
00023 C IERR -ERROR FLAG--(0) ROUTINE PROCESSED OK,(1) PROCESSING ERR
00024 C IRBLNK-TAG INDICATING--(0) RIGHT BLANKS IN CHARACTER STRING TO
00025 C TRANSFERRED, (1) FIRST NON-BLANK CHARACTER HAS BEEN
00026 C ENCOUNTERED
00027 C IWORD -INPUT WORD CONTAINING 2 FIELDS OF 3 AND 5 CHARACTERS
00028 C JWORD -OUTPUT WORD CONTAINING 1ST 3 CHARS OF *TWORD*, LEFT
00029 C JUSTIFIED BLANK FILLED
00030 C KODES -VECTOR OF LEGITIMATE VALUES OF *JWORD*
00031 C NPROD -NUMBER OF PERIODS ENCOUNTERED IN *TWORD*
00032 C
00033 C SUBROUTINES OR FUNCTIONS REQUIRED
00034 C AFIELD
00035 C GETCHA
00036 C PUTCHA
00037 C
00038 C-----
00039 C
00040 DOUBLE PRECISION KODES,IRBLANK,NULL
00041 DOUBLE PRECISION IWORD,JWORD,KWORD
00042 DOUBLE PRECISION IUTYPE,IETYPE,IUSUB,IESUB,ICMPLX,MISS
00043 COMMON /MAX/ MAXCMP,MAXUNT,MAXMIS,NIT,MAXFLD,MAXROW,NULL
00044 COMMON /IFACE/ IUTYPE(2,15),IETYPE(2,15),IUSUB(2,15,15),
00045 + IFSUB(2,15,15),ICMPLX(2,8),MISS(2,17)
00046 DIMENSION KODES(7)
00047 DATA KODES/6H0A ,6H0A ,6H0A ,6H0A ,6H0A ,6H0A ,6H0A /
00048 16HDOS ,6H0DS ,6HEND /
00049 DATA IBLNK/1H /
00050 DATA IPRD/14./
00051 DATA IBLANK/6H /
00052
00053 IRBLNK = 0
00054 NPROD = 0
00055 IERR = 0
00056 IF(IWORD.EQ.10#) RETURN

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
00057      JWORD = IBLANK
00059      KWORD = IBLANK
00059      C
00060      DO 100 I=1,3
00061      CALL GETCHA(ICHAR,IWORD,I)
8FTNDIM LINL:00061 POSSIBLE TO INDEX MODIFICATION INSIDE LOOP
00062      CALL PUTCHA(ICHAR,JWORD,I)
8FTNDIM LINE:00062 POSSIBLE TO INDEX MODIFICATION INSIDE LOOP
00063      100 CONTINUE
00064      C
00065      DO 200 I=1,7
00066      IF(JWORD.EQ.KODES(I)) GO TO 300
00067      200 CONTINUE
00068      IERR = 1
00069      GO TO 600
00070      300 CONTINUE
00071      IF(JWORD.EQ.6HFND) GO TO 500
00072      KS=10
00073      DO 400 I=4,8
00074      IS = 12 - I
00075      CALL GETCHA(ICHAR,IWORD,IS)
00075      IF(ICHAR.EQ.IPRD) Nprd = Nprd + 1
00077      IF(ICHAR.EQ.IBLNK.AND.IRBLNK.EQ.0) GO TO 400
00078      CALL NUMCHK(ICHAR,IERR)
00079      IF(IERR.EQ.1) GO TO 600
00080      CALL PUTCHA(ICHAR,KWORD,KS)
00081      KS=KS-1
00082      IRBLNK = 1
00083      400 CONTINUE
00084      GO TO 600
00085      500 CONTINUE
00086      CALL AFIELD(IWORD,4,8,KWORD,TERP)
00087      IF(TERP.EC.1) GO TO 600
00088      C
00089      C      CHECK FOR PRESENCE ON VISION LIST
00090      C
00091      IND=IPED+1
00092      DO 550 J=1,MAXVIS
00093      IF(KWORD.EQ.MTSS(IND,J)) RETURN
00094      550 CONTINUE
00095      TERP=2
00096      600 CONTINUE
00097      RETURN
00098      END
00099
```

```
00001
00002      SUBROUTINE GETLR(INBUF,ITN,ITEMP,IERR)
00003  C
00004  C      AUTHOR
00005  C          BOB RINGO
00006  C          STANFORD RESEARCH INSTITUTE/NWRC
00007  C          MENLO PARK, CALIFORNIA
00008  C
00009  C      DATE OF LAST REVISION
00010  C          24 NOVEMBER 1976
00011  C
00012  C      PURPOSE
00013  C          TO DECODE VECTOR *INBUF* BY LINE BY FIELD AND PLACE RESULTS
00014  C          IN *ITEMP*. LIMITED ERROR CHECKING IS PERFORMED.
00015  C
00016  C      DESCRIPTION OF VARIABLES
00017  C          I -INDEX TO LINE WITHIN VECTOR *INBUF*
00018  C          IBLNK -ARRAY INDICATING IF A FIELD HAS AN ERROR(1) OR
00019  C          WAS DECODED OK (0)
00020  C          IE -ENDING CHARACTER TO BE DECODED IN VECTOR *INBUF*
00021  C          IFLDS -ARRAY INDICATING LENGTH OF FIELD TO BE DECODED
00022  C          INBUF -CHARACTER STRING VECTOR OF TABLE TO BE DECODED
00023  C          ITABS -ARRAY INDICATING NUMBER OF LINES AND FIELDS PER
00024  C          LINE FOR A GIVEN TABLE
00025  C          ITEMp -ARRAY OF FIELDS DECODED BY THIS ROUTINE
00026  C          IS -STARTING CHARACTER TO BE DECODED IN VECTOR *INBUF*
00027  C          J -INDEX TO FIELD WITHIN LINE TO BE DECODED
00028  C          LNF -LENGTH OF FIELD TO BE DECODED
00029  C          NF -NUMBER OF FIELDS PER LINE
00030  C          NL -NUMBER OF LINES PER TABLE
00031  C          NORA -(0) ALPHA FIELD, (1) NUMERIC FIELD
00032  C
00033  C      SUBROUTINES OR FUNCTIONS REQUIRED
00034  C          AFIELD
00035  C          NFIELD
00036  C
00037
00038      DOUBLE PRECISION INBUF,ITEMP
00039      DIMENSION INBUF(1),ITEMP(19,17)
00040      COMMON /SPECs/ ITABFR, ITABLR, ITABNR, ITABNF, IFLDS(14,15),
00041      1           IBLNK(15,17), IFORMT(2,19)
00042  C
00043      IERR = 0
00044      NL = ITABNR
00045      IE = 0
00046      DO 1000 T=1,NL
00047      NF = ITABNF
00048      DO 1000 J=1,NF
00049      LNF = IFORMT(2,J)
00050      IS = IE + 1
00051      IE = IS + LNF - 1
00052      NORA = IFLDS(ITN,J)
00053      IF(ITN.EQ.14.AND.I.EQ.1) NORA = 1
00054  C
00055  C          NUMERIC FIELD
00056  C
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
00057      CALL NFIELD(INBUF,IS,IE,ITEMP(J,I),IBLNK(J,I))
00058      IF(IBLNK(J,I).EQ.1) IERR = 1
00059      GO TO 1000
00060
00061      C
00062      C      ALPHA FIELD
00063      C
00064      500 CALL AFIELD(INBUF,IS,IE,ITEMP(J,I),IBLNK(J,I))
00065      IF(IBLNK(J,I).EQ.1) IERR = 1
00066      1000 CONTINUE
00067      RETURN
00068      END
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
00001      SUBROUTINE NUMCHK(ICHAR,IERR)
00002      C
00003      C      AUTHOR
00004      C          ROB RINGO
00005      C          STANFORD RESEARCH INSTITUTE/NWRC
00006      C          MENLO PARK, CALIFORNIA
00007      C
00008      C      DATE OF LAST REVISION
00009      C          24 NOVEMBER 1976
00010      C
00011      C      PURPOSE
00012      C          TO CHECK NUMERIC CHARACTERS TO ENSURE THEY ARE 0-9 OR (.)
00013      C
00014      C      DESCRIPTION OF VARIABLES
00015      C          I - INDEX OF CHARACTER IN *LENUM*
00016      C          ICHAR - CHAR TO BE CHECKED, RIGHT JUSTIFIED, ZERO FILLED
00017      C          IERR - ERROR TAG--(0) *ICHAR* OK, (1) ILLEGAL CHAR IN *ICHAR*
00018      C          LEGNUM-VECTOR OF LEGITIMATE NUMERIC CHARACTERS
00019      C
00020      C      FUNCTIONS OR SUBROUTINES REQUIRED
00021      C          NONE
00022      C
00023      C-----C
00024      C
00025      C      DIMENSION LEGNUM(13)
00026      C      DATA LEGNUM /1H-, 1H0, 1H1, 1H2, 1H3, 1H4, 1H5, 1H6,
00027      C          +           1H7, 1H8, 1H9, 1H., 1H /
00028      C          IERR=0
00029      C
00030      C          DO 100 I=1,13
00031      C          IF(ICHAR.EQ.LENUM(I)) RETURN
00032      C 100 CONTINUE
00033      C          IERR = 1
00034      C          RETURN
00035      C          END
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
00001      SUBROUTINE NFIELD(INBUF,IS,IE,ITEMP,IERR)
00002      C
00003      C      AUTHOR
00004      C          BOB RINGO
00005      C          STANFORD RESEARCH INSTITUTE/NWRC
00006      C          MENLO PARK, CALIFORNIA
00007      C
00008      C      DATE OF LAST REVISION
00009      C          24 NOVEMBER 1976
00010      C
00011      C      PURPOSE
00012      C          TO TRANSFER A NUMERIC CHARACTER STRING FROM VECTOR *INBUF*
00013      C          STARTING AT *IS* AND ENDING AT *IE* TO *ITEMP*. RIGHT BLANKS
00014      C          ARE STRIPPED OFF. ITEMP IS LEFT FILLED WITH BLANKS, AND EACH
00015      C          CHAR IS CHECKED FOR 0-9,(.),(. ). CHECK IS ALSO MADE FOR
00016      C          MORE THAN ONE (.).
00017      C
00018      C      DESCRIPTION OF VARIABLES
00019      C          IBLANK-BLANK FILLED WORD
00020      C          ICHAR -CHAR OF *INBUF* TO BE EXAMINED AND TRANSFERRED
00021      C          IE   -ENDING CHARACTER TO BE TRANSFERRED IN VECTOR *INBUF*
00022      C          IERR -ERROR TAG--(0) FIELD IS OK, (1) BAD DATA IN NUMERIC FILE
00023      C          INBUF -INPUT VECTOR CONTAINING CHARACTER STRING TO BE TRANSFERRED
00024      C          AND EXAMINED FOR LEGITIMATE NUMERALS
00025      C          IRBLNK-TAG INDICATING--(0) RIGHT BLANKS IN CHAR STRING TO BE
00026      C          TRANSFERRED, (1) FIRST NON-BLANK CHAR HAS BEEN ENCOUNTERED
00027      C          IS   -STARTING CHARACTER TO BE TRANSFERRED IN VECTOR *INBUF*
00028      C          ISC  -STARTING CHAR OF *ITEMP* TO RECEIVE CHARS FROM *INBUF*
00029      C          ITEMP -TARGET WORD FOR NUMERIC CHARACTER STRING TRANSFERRED
00030      C          FROM *INBUF*, LEFT FILLED WITH BLANKS
00031      C
00032      C      DOUBLE PRECISION INBUF,IPLANK,ITEMP
00033      C      DIMENSION INBUF(1)
00034      C      DATA IPRD /1H./
00035      C      DATA IRBLNK/1H /
00036      C      DATA IPLANK/64      /
00037      C
00038      C          ITEMP = IBLANK
00039      C          IERR=0
00040      C          NPRD = 0
00041      C          IRBLNK = 0
00042      C          ISC = IE - IS + 1
00043      C          J=IE+1
00044      C          DO 100 I=IS,IE
00045      C          J=J-1
00046      C          CALL GETCHA(ICHAR,INBUF(1),J)
00047      C          IF(ICHAR.EQ.IPRD) NPRD = NPRD + 1
00048      C          IF(ICHAR.EQ.IBLNK.AND.IRBLNK.EQ.0) GO TO 100
00049      C          CALL NUMCHK(ICHAR,JERR)
00050      C          IF(JERR.EQ.1) IERR=1
00051      C          CALL PUTCHA(ICHAR,ITEMP,ISC)
00052      C          IRBLNK = 1
00053      C          ISC = ISC -1
00054      C          100 CONTINUE
00055      C          IF(NPRD.GT.1) IERR = 1
00056      C          RETURN
00057      C      END
```

```
00001      SUBROUTINE AFIELD(INBUF,IS,IE,ITEMP,IERR)
00002      C
00003      C      AUTHOR
00004      C          BOB RINGO
00005      C          STANFORD RESEARCH INSTITUTE/NWRC
00006      C          MENLO PARK, CALIFORNIA
00007      C
00008      C      DATE OF LAST REVISION
00009      C          24 NOVEMBER 1976
00010      C
00011      C      PURPOSE
00012      C          TO TRANSFER AND LEFT JUSTIFY A CHARACTER STRING FROM A VECTOR
00013      C          *INBUF* STARTING AT CHARACTER POSITION *IS* AND ENDING AT
00014      C          CHARACTER POSITION *IE*. LEADING BLANKS ARE
00015      C          STRIPPED OFF AND TRAILING CHARACTERS ARE BLANK FILLED.
00016      C
00017      C      DESCRIPTION OF VARIABLES
00018      C          I -CURRENT CHARACTER POSITION OF *INBUF*
00019      C          IBLANK-BLANK FILLED WORD
00020      C          ICHAR -RIGHT JUSTIFIED, ZERO FILLED CHARACTER
00021      C          IERR -ERROR TAG--(0) NO ERROR, (1) ERROR
00022      C          IFIRST-TAG INDICATING FIRST NON-BLANK CHARACTER HAS BEEN ENCOU
00023      C          INBUF -INPUT CHARACTER STRING VECTOR
00024      C          ITEMP -OUTPUT CHARACTER STRING--LEFT JUSTIFIED, BLANK FILLED
00025      C          K -CHARACTER POSITION OF *ITEMP*
00026      C
00027      C      FUNCTIONS OR SUBROUTINES REQUIRED
00028      C          GETCHA
00029      C          PUTCHA
00030      C
00031      C-----+
00032      C
00033      DOUBLE PRECISION INBUF,ITEMP,IBLANK
00034      DIMENSION INBUF(1)
00035      DATA IBLANK/6H      /
00035      DATA IBLNF/1H      /
00037      C
00038      C      ITEMP = IBLANK
00039      C      IERR = 0
00040      C      IFIRST = 0
00041      C      K = 0
00042      C
00043      DO 100 I=IS,IE
00044      CALL GETCHA(ICHAR,INBUF(1),I)
00044      LINE:00044 POSSIBLE DO INDEX MODIFICATION INSIDE LOOP
00045      IF(ICHAR.EQ.IBLNK.AND.IFIRST.EQ.0) GO TO 100
00046      IFIRST = 1
00047      K = K + 1
00048      CALL PUTCHA(ICHAR,ITEMP,K)
00049      100 CONTINUE
00050      RETURN
00051      END
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
00001      SUBROUTINE SCRNO(IT,ITFMP)
00002      C
00003      C      THIS ROUTINE OUTPUTS TO THE SCREEN, FORMAT AND DATA FOR A TABLE
00004      C
00005      INTEGER FORM, UFLD, FORMNO, OTSTR, FCHAR, GETCHR, OUTPTR,
00006      1      COLN, USRPTR, TRMTYP, UPC, DOWNC, LEFTC, RIGHTC, TRANSC,
00007      2      HOMEC, TABC, CRC, ERASEC, PROC, UNPROC, BLC, UNRLC,
00008      3      CPXPTP, CFLD, UROW, XCHAR, RFLD, DFLD, DDAY, BLANK,
00009      4      RUNPTR, DAYPTR
00010
00011      DOUBLE PRECISION ITEMP, NULL, IUTYPE, IETYPE, IUSUB,
00012      1      ICMLPX, MISS, IESUB, KTEMP
00013      DIMENSION ITEMP(19,17), KTEMP(20)
00014      DIMENSION OTSTR(500)
00015      COMMON/MODS/IDRUN, IDAY
00016      DOUBLE PRECISION IDRUN
00017      COMMON /INBUFF/ INRPTR, INBUF(200)
00018      DOUBLE PRECISION INBUF
00019      COMMON/TIO/TRMTYP, UPC, DOWNC, LEFTC, RIGHTC, UDX, TRANSC,
00020      1      HOMEC, TABC, ADSBP
00021      COMMON /SPECS/ ITABFR, ITABLR, ITABNR, ITABNF, IFlds(14,15),
00022      1      IPLNK(15,17), IFORMT(2,19)
00023      COMMON /CURTAB/ FORMNO, FORM(24,80)
00024      COMMON /IFACE/ IUTYPE(2,15), IETYPE(2,15), IUSUB(2,15,15),
00025      1      IESUB(2,15,15), ICMLPX(2,8), MISS(2,17)
00026      DATA UFLD/"725004020100/
00027      DATA CFLD/"615004020100/
00028      DATA PFLD/"711004020100/
00029      DATA DFLD/"621004020100/
00030      C
00031      C-----CLEAR SCREEN
00032      CALL NWSCRN(0)
00033      C
00034      C-----READ-IN TABLE FROM FORMS FILE
00035      IF(IT.LT.15 .AND. IT.EQ.FORMNO) GO TO 20
00036      DO 10 IPOW = 1,24
00037      INX = IPOW+24*(IT-1)
00038      READ(23#INX,1000) (FORM(IROW,ICOL),ICOL=1,80)
00039      10  CONTINUE
00040      FORMNO=IT
00041      C
00042      C-----REPLACE "U" WITH DATA CHAR COUNT
00043      CALL FORMS(UFLD)
00044      C
00045      C-----GET COMPLEX HEADERS FOR TABLE 13
00046      IF(IT .NE. 13) GO TO 17
00047      DO 16 I=1,8
00048      KTEMP(I)=ICMLPX(1,I)
00049      KTEMP(I+8)=ICMLPX(2,I)
00050      15  CONTINUE
00051      17  CONTINUE
00052      C
00053      C-----ENCODE PATTLE DAYS FOR OUTPUT TABLES
00054      IF(IT .GT. 14) ENCODE(2,1001,DDAY) IDAY
00055      C
00056      C-----INITIALIZE PRINTERS
```

```
00057    20      CONTINUE
00058          IROW=0
00059          NFLD=0
00060          NCFLD=0
00061          OUTPTR=0
00062          INBPTR=0
00063          C
00064          C-----BEGIN NEW ROW
00065          30      CONTINUE
00066          IROW=IROW+1
00067          IF(IROW .GT. 24) GO TO 200
00068          ICOL=0
00069          USRPTR=0
00070          CPXPTR=0
00071          RUNPTR=0
00072          DAYPTR=0
00073          JBLNK=0
00074          NFLD=0
00075          C
00076          C-----NEXT COLUMN
00077          40      CONTINUE
00078          ICOL=ICOL+1
00079          IF(ICOL .GT. 80) GO TO 30
00080          OUTPTR=OUTPTR+1
00081          FCHAR=FORM(IROW,ICOL)
00082          IF(GETCHR(FCHAR,1) .EQ. 0) GO TO 50
00083          IF(FCHAR .EQ. CFLD) GO TO 70
00084          IF(FCHAR .EQ. RFLD) GO TO 80
00085          IF(FCHAR .EQ. DFLD) GO TO 90
00086          C
00087          C-----OUTPUT A FORMS CHARACTER
00088          CPXPTR=0
00089          USRPTR=0
00090          RUNPTR=0
00091          DAYPTR=0
00092          IF(JBLNK .EQ. 0) GO TO 45
00093          JBLNK=0
00094          CALL PTUBL(C(OTSTR,OUTPTR)
00095          45      CONTINUE
00096          IF(FCHAR .EQ. -1) GO TO 100
00097          CALL PUTCHA(FCHAR, OTSTR, OUTPTR)
00098          GO TO 40
00099          C
00100          C-----BEGIN A USER DATA FIELD "u"
00101          50      CONTINUE
00102          RUNPTR=C
00103          DAYPTR=C
00104          CPXPTR=0
00105          IF(USRPTR .NE. 0) GO TO 60
00106          NFLD = NFLD+1
00107          UROW = IROW-ITABFR
00108          IF(YPLNK(NFLD,UROW) .EQ. 0) GO TO 60
00109          CALL PUTCHR(LEFTC,OTSTR,OUTPTR)
00110          OUTPTR=OUTPTR+1
00111          CALL PUTCHR(62,OTSTR,OUTPTR)
00112          OUTPTR=OUTPTR+1
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
00113      CALL PTPLC(OTSTR,OUTPTR)
00114      JBLNK=1
00115      60      CONTINUE
00116      USRPTR=USRPTR+1
00117      CALL GETCHA(UCHAR, ITEMP(NFLD,UROW), USRPTR)
00118      CALL PUTCHA(UCHAR, OTSTR, OUTPTR)
00119      INBPTR = INBPTR+1
00120      CALL PUTCHA(UCHAR,INBUF,INBPTR)
00121      GO TO 40
00122      C
00123      C-----PROCESS A COMPLEX HEADER FIELD "c"
00124      70      CONTINUE
00125      RUNPTR=0
00126      DAYPTR=0
00127      USRPTR=0
00128      IF(CPXPTR .EQ. 0) NCFLD=NCFLD+1
00129      CPXPTR=CPXPTR+1
00130      CALL GETCHA(XCHAR, KTEMP(NCFLD), CPXPTR)
00131      CALL PUTCHA(XCHAR, OTSTR, OUTPTR)
00132      GO TO 40
00133      C
00134      C-----PROCESS A RUNITD FIELD "r"
00135      80      CONTINUE
00136      CPXPTR=0
00137      USRPTR=0
00138      DAYPTR=0
00139      RUNPTR=RUNPTR+1
00140      XCHAR=BLANK
00141      IF(RUNPTR .LE. 10) CALL GETCHA(XCHAR, IDRUN,RUNPTR)
00142      CALL PUTCHA(XCHAR,OTSTR,OUTPTR)
00143      GO TO 40
00144      C
00145      C-----PROCESS A BATTLE DAYS FIELD "d"
00146      90      CONTINUE
00147      CPXPTR=0
00148      USRPTR=0
00149      RUNPTR=0
00150      DAYPTR=DAYPTR+1
00151      XCHAR=BLANK
00152      IF(DAYPTR .LE. 2) CALL GETCHA(XCHAR,DDAY,DAYPTR)
00153      CALL PUTCHA(XCHAR,OTSTR,OUTPTR)
00154      GO TO 40
00155      C
00156      C-----END OF LINE ENCOUNTERED
00157      100     CONTINUE
00158      CALL PUTCHR(13,OTSTR,OUTPTR)
00159      ICOL = 80
00160      IF(IROW .EQ. 24) GO TO 40
00161      OUTPTR = OUTPTR+1
00162      CALL PUTCHR(10,OTSTR,OUTPTR)
00163      GO TO 40
00164      C
00165      C-----
00166      200     CONTINUE
00167      CALL PTUBLIC(OTSTR,OUTPTR)
00168      CALL PUTCHR(0, OTSTR, OUTPTR)
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
00169      CALL OUTSTR(DUM,OTSTR)
00170      RETURN
00171      C
00172      C-----FORMATS
00173      1000   FORMAT(80A1)
00174      1001   FORMAT(I2)
00175      END
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
00001      SUBROUTINE SCRNI(IT,ITEMP)
00002      C
00003      C      THIS ROUTINE READS UNPROTECTED DATA FROM THE SCREEN
00004      C
00005      DOUBLE PRECISION ITEMP,INBUF
00006      DIMENSION ITEMP(19,17)
00007      COMMON /INBUFF/ INBPTR,INBUF(200)
00008      C
00009      C-----CHECK FOR DUMMY READ
00010      IF(IT .NE. 0) GO TO 20
00011      CALL BUFIN(IT,INBUF,MODS)
00012      GO TO 200
00013      C
00014      C-----PROCESS REGULAR TABLE INPUT REQUEST
00015      20      CONTINUE
00016      CALL BUFIN(IT,INBUF,MODS)
00017      C
00018      C-----CHECK FOR ANY CHANGES MADE TO TABLE
00019      IF(MODS .GT. 0) GO TO 100
00020      IT=0
00021      GO TO 200
00022      C
00023      C-----CHANGE HAS BEEN MADE
00024      100     CONTINUE
00025      CALL GETLR(INBUF,IT,ITFMP,IERR)
00026      IF(IERR .EQ. 1) IT=-IT
00027      C
00028      C-----RETURN TO CALLER
00029      200     CONTINUE
00030      RETURN
00031      END
```

```
00001      SUBROUTINE RUFIN(TABLE,OUTARA,MODS)
00002      C
00003      C      THIS SUBROUTINE INPUTS CHARS TO PERFORM TABLE MODIFICATION
00004      C
00005      IMPLICIT INTEGER(A-Z)
00006      COMMON /SPECs/ ITABFR, ITABLR, ITABNR, ITABNF, IFLDS(14,15),
00007      1          IPLNK(15,17), IFORMT(2,19)
00008      1          COMMON/TIO/TRMTYP,UPC,DOWNC,LEFTC,RIGHTC,UDX,TRANSC,
00009      1          HOMEc,TABC,ADSBP
00010      COMMON/CURTAB/FOPMNO,FORM(24,80)
00011      C
00012      DIMENSION OUTARA(1)
00013
00014      CRC=13
00015      ROWS=ITABLR+1
00016      MODS=0
00017      C
00018      C
00019      C
00020      C-----SET TERMINAL IN BINARY MODE FOR CHAR MAPING
00021      CALL PTERM
00022      C
00023      C-----CHECK FOR FOR DUMMY READ REQUEST
00024      IF(TABLE.NE.0) GO TO 100
00025      C
00026      C-----GETTING CHARACTERS FROM TERMINAL
00027      20      CONTINUE
00028      CALL GETT(INCHAR)
00029      C
00030      C
00031      IF(TABLE .EQ. 0) GO TO 150
00032      IF(INCHAR.EQ.CRC) GO TO 40
00033      IF(INCHAR.EQ.LEFTC) GO TO 50
00034      IF(INCHAR.EQ.RIGHTC) GO TO 60
00035      IF(INCHAR.EQ.UPC) GO TO 70
00036      IF(INCHAR.EQ.DOWNC) GO TO 80
00037      IF(INCHAR.EQ.TABC) GO TO 90
00038      IF(INCHAR.EQ.HOMEc) GO TO 100
00039      IF(INCHAR.EQ.TRANSC) GO TO 110
00040      C
00041      C-----NORMAL CHARACTER PROCESSING
00042      30      CONTINUE
00043      IF(INCHAR.LT.32 .OR. INCHAR.GT.126) GO TO 20
00044      MODS=MODS+1
00045      CALL SFND(INCHAR)
00046      X = FORM(ROW,POS)
00047      CALL PUTCHR(INCHAR,OUTARA,X)
00048      POS = POS+1
00049      IF(POS.LT.80 .AND. GETCHR(FORM(ROW,POS),1).EQ.0) GO TO 20
00050      CALL SEND(LEFTC)
00051      POS = POS-1
00052      GO TO 20
00053      C
00054      C-----CARRAGE RETURN PROCESSING
00055      40      CONTINUE
00056      CALL SEND(CRC)
```

~~SELECTED~~
THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
00057      IF(TRMTYP.EQ.2) CALL SEND(DOWNC)
00058      POS = 1
00059      ROW=ROW+1
00060      IF(ROW .GT. ROWS) GO TO 100
00061      IF(GETCHR(FORM(ROW,POS),1) .EQ. 0) GO TO 20
00062      GO TO 60
00063      C
00064      C-----LEFT CURSOR PROCESSING
00065      50      CONTINUE
00066          IF = POS-1
00067          IF(IP.LT.1) GO TO 60
00068          CALL SEND(LEFTC)
00069          POS = POS-1
00070          IF(GETCHR(FORM(ROW,POS),1) .EQ. 0) GO TO 20
00071          GO TO 50
00072      C
00073      C-----RIGHT CURSOR PROCESSING
00074      60      CONTINUE
00075          IP = POS+1
00076          IF(IP.GT.80) GO TO 40
00077          CALL SEND(RIGHTC)
00078          POS = POS+1
00079          IF(GETCHR(FORM(ROW,POS),1) .EQ. 0) GO TO 20
00080          GO TO 60
00081      C
00082      C-----UP CURSOR PROCESSING
00083      70      CONTINUE
00084          IR = ROW-ITABFR-1
00085          IF(IR.LT.1) GO TO 20
00086          CALL SFND(UPC)
00087          ROW = ROW-1
00088          IF(GETCHR(FORM(ROW,POS),1) .EQ. 0) GO TO 20
00089          GO TO 70
00090      C
00091      C-----DOWN CURSOR PROCESSING
00092      80      CONTINUE
00093          IR = ROW+1
00094          IF(IR.GT.ROWS) GO TO 20
00095          CALL SEND(DOWNC)
00096          ROW = ROW+1
00097          IF(GETCHR(FORM(ROW,POS),1) .EQ. 0) GO TO 20
00098          GO TO 80
00099      C
00100      C-----TAB CURSOR PROCESSING
00101          C SCAN TO END OF CURRENT FIELD
00102          90      CONTINUE
00103          IF(POS.GE.80) GO TO 95
00104          CALL SEND(RIGHTC)
00105          POS = POS+1
00106          IF(GETCHR(FORM(ROW,POS),1) .EQ. 0) GO TO 90
00107      C-----SCAN TO PEG OF NEXT FIELD
00108          GO TO 60
00109      C-----END OF CURRENT LINE HIT WHILE TAPING
00110          95      CONTINUE
00111          IF(ROW.GE.ROWS) GO TO 100
00112          GO TO 40
```

```
00113 C
00114 C-----HOME CURSOR PROCESSING
00115 100 CONTINUE
00116 CALL SEND(HOME)
00117 ROW = 1
00118 POS = 1
00119 IR = ITABFR
00120 IF(TR.EQ.0) GO TO 105
00121 DO 104 I=1,IR
00122 CALL SEND(DOWNC)
00123 ROW = ROW+1
00124 104 CONTINUE
00125 105 CONTINUE
00126 IF(GETCFR(FORM(ROW,POS),1) .EQ. 0) GO TO 20
00127 GO TO 60
00128 C
00129 C-----XMIT DATA PROCESSNG
00130 110 CONTINUE
00131 GO TO 150
00132 C
00133 C-----EXIT PROCESSING
00134 150 CONTINUE
00135 CALL STERM
00136 CALL MNSCRN(0)
00137 RETURN
00138 END
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
00001      SUBROUTINE RDYOUT(IT, ITEMP)
00002      C
00003      C      THIS SUBROUTINE PREPARES OUTPUT TABLE DATA FOR DISPLAY
00004      C
00005      DOUBLE PRECISION ITEMP(19,17),NULL
00006      DOUBLE PRECISION IUTYPE,IETYPE,IUSUB,IESUP,ICMPLX,MISS,
00007      1 DCOMPS,DPLEX,ORIG,TARG,TTS
00008      REAL LRDEF,LIVPLX
00009      INTEGER ORIGIN,TARGET,START,STOP,UMAX,UMIN,UREADY,UASK,
00010      1 UGOT
00011      C
00012      C
00013      COMMON/MODS/IDPRUN,IDAY
00014      DOUBLE PRECISION IDRUN
00015      COMMON/BLOCKB/DSUNIT(2,20,2),SCOMPS(2,20,5),COMPS(2,20,5),
00016      1 STRAT(2,20,4),ADEFAT(2,20,2),AGKILL(2,20,10),SRATE(2,2,5),
00017      2 DCOMPS(2,10),MAXP(2,20),LRDFF(2,20),WSTWX(2,20),BWDEF(2,20),
00018      3 DRMAX(2,20),SPEFD(2,20),IRRTIM(2,5),SR30MX(2,5),IMAXR(2,20),
00019      4 ILRDEF(2,20),IWSTWX(2,20),IBWDEF(2,20),NCOMPS(2),NSUNIT(2)
00020      COMMON/PLOCKC/COMPLY(2,10,10),DPLEX(2,10),DTSLI(2,10),
00021      1 SURFAC(2,10),DEFAC(2,10),ICYCLE(2,10),IREPL(2,10),
00022      2 RFPUPS(2,10,2),OPS(2,11),REPAIR(2,11),NPFLX(2),MTIME(2),
00023      3 NDAYS,JWXDEF,JLRDEF,WXSUM,VDELAY,OPSLCH,OPSLND
00024      COMMON/BLOCKP/IONOFF(2,20,2),JONOFF(2,20,2),S1(2,20,2),
00025      1 S2(2,20,2),MTS(2,20,8),UNIT(2,20,2),SA(2,20,2),SB(2,20,2),
00026      2 TTS(2,20),RANGE(10,10),IRANGE(10,10),DMISS(2,20),IPRI(2,20),
00027      3 ORIG(2,20),TARG(2,20),UMAX(2,20),UMIN(2,20),ORIGIN(2,20),
00028      4 TARGET(2,20),IUNIT(2,20),START(2,20),STOP(2,20),UREADY(2,20),
00029      5 DAY(30),WX(30),IWK(30),NMIS(2)
00030      COMMON/PLOCKF/LIVPLX(2,10,10),UASK(2,20,3),UGOT(2,20,3),
00031      1 SR30(2,10,5),AIRUP(20,5,8),AIRDN(20,5,8),ADPASE(2,10)
00032      COMMON/IFACE/IUTYPF(2,15),IETYPE(2,15),IUSUB(2,15,15),
00033      1 IESUP(2,15,15),ICMPLX(2,9),WTSS(2,17)
00034      COMMON /SFCSS/ ITARFR, ITAPLR, ITAENR, ITAENF, IFLDS(14,15),
00035      1 IBLNK(15,17), IFORMT(2,19)
00036      COMMON/MAX/MAXCMP,MAXUNT,MAXMIS,NIT,MAXFLD,MAXROW,NULL
00037      COMMON/AIREXP/ EXP(19,10)
00038      *INVERSION IEXPS(19,10)
00039      C
00040      C-----INITIALIZE ALL FIELDS TO BLANKS
00041      DO 20 I=1,19
00042      DO 20 J=1,MAXROW
00043      ITEMP(I,J) = NULL
00044      20 CONTINUE
00045      C
00046      C-----
00047      IF(IT .EQ. 19) GO TO 80
00048      IRED = 0
00049      IF(IT.EQ.16 .OR. IT.EQ.18) IREF=1
00050      L = IRED+1
00051      IF(IT .GT. 16) GO TO 70
00052      NM = NMIS(L)
00053      DO 55 M=1,NM
00054      IT1 = UASK(L,M,1)+UASK(L,M,2)+UASK(L,M,3)
00055      IT2 = UGOT(L,M,1)+UGOT(L,M,2)+UGOT(L,M,3)
00056      E"CDPF(120,1000,ITEMP(1,M)) DMSS(L,M),IPRI(L,"),
```

```
00057      1 ORIG(L,M),TARG(L,M),SA(L,M,1),SB(L,M,1),SA(L,M,2),  
00058      2 SB(L,M,2),TTS(L,M),UNIT(L,M,1),UNIT(L,M,2),UMAX(L,M),  
00059      3 IT1,IT2  
00060      55 CONTINUE  
00061      GO TO 100  
00062      C  
00063      C-----  
00064      70 CONTINUE  
00065      NC = NCOMPS(L)  
00066      NP = NPLEX(L)  
00067      NCHAR = 70  
00068      IF(IT .EQ. 18) NCHAR=90  
00069      NM = 0  
00070      DO 75 K=1,NP  
00071      NM = NM+1  
00072      ENCODE(NCHAR,1001,ITEMP(1,NM)) DPLEX(L,K),  
00073      (LIVPLX(L,K,I),I=1,NC)  
00074      1 NM = NM+1  
00075      ENCODE(NCHAR,1002,ITEMP(1,NM)) (COMPLX(L,K,I),I=1,NC)  
00076      75 CONTINUE  
00077      GO TO 100  
00078      C  
00079      C-----  
00080      90 CONTINUE  
00081      DO 84 I=1,1DAY  
00082      IEXPS(1,I)=I  
00083      DO 84 J=2,19  
00084      IEXPS(J,I)=EXPS(J,I)  
00085      84 CONTINUE  
00086      IF(IDAY.LE.9) IEXPS(1,1DAY+1)=0  
00087      C  
00088      C-----FORMING TABLE 19 INTO ITEMP  
00089      MDY=0  
00090      IROW=0  
00091      90 CONTINUE  
00092      MDY=MDY+1  
00093      IF(IEXPS(1,MDY).EQ.0 .OR. MDY.GT.10) GO TO 95  
00094      IROW=IROW+1  
00095      ENCODE(100,1003,ITEMP(1,IROW))(IEXPS(I,MDY),I=1,10)  
00096      ENCODE(90,1004,ITEMP(11,IROW))(IEXPS(I,MDY),I=11,19)  
00097      GO TO 90  
00098      95 CONTINUE  
00099      C  
00100      C-----RETURN TO CALLER  
00101      100 CONTINUE  
00102      RETURN  
00103      C  
00104      C-----FORMATS  
00105      1000 FORMAT(A5,5X,I1,9X,A6,4X,A6,4X,A3,A5,2X,A3,A5,2X,A8,2X,  
00106      1 A5,5X,A3,7X,I3,7X,I3,7X,I3,7X)  
00107      1001 FORMAT(A6,4X,4(F6.0,4X),4(F6.1,4X))  
00108      1002 FORMAT(10X,4(F6.0,4X),4(F6.1,4X))  
00109      1003 FORMAT(I2,8X,I4,6X,3(I3,7X),I4,6X,3(I3,7X),I4,6X)  
00110      1004 FORMAT(3(I3,7X),I4,6X,5(I3,7X))  
00111      END
```

130

```
00001      SUBROUTINE PTUBLIC(OTSTR,OUTPTR)
00002 C      THIS SUBROUTINE PUTS THE APPROPRIATE CONTROL CHARACTERS INTO THE
00003 C      OUTPUT STRING 'OTSTR' TO TURN ON BLINKING.
00004      IMPLICIT INTEGER (A-Z)
00005      INTEGER OTSTR(1)
00006      COMMON/TIO/TRMTYP,UPC,DOWNC,LEFTC,RIGHTC,UDX,TRANSC,
00007      1      HOMEC,TARC,ADSPP
00008      IF(TRMTYP.EQ.1) GO TO 100
00009      IF(TRMTYP.EQ.3) GO TO 300
00010      IF(TRMTYP.EQ.2) GO TO 200
00011      TYPE 9999,TRMTYP
00012      9999  FORMAT(* PTUBLIC DOES NOT SUPPORT TERMINAL TYPE ',16)
00013      RETURN
00014      100   CALL PUTCHR(24,OTSTR,OUTPTR)
00015      OUTPTR = OUTPTR + 1
00016      RETURN
00017      300   CALL PUTCHR(15,OTSTR,OUTPTR)
00018      OUTPTR = OUTPTR + 1
00019      RETURN
00020      200   CALL PUTCHR(127,OTSTR,OUTPTR)
00021      OUTPTR = OUTPTR + 1
00022      RETURN
00023      END
```

00001 SUBROUTINE PTBLC(OTSTR,OUTPTR)
00002 C THIS SUBROUTINE PUTS THE APPROPRIATE CONTROL CHARACTERS INTO THE
00003 C OUTPUT STRING 'OTSTR' TO TURN OFF BLINKING.
00004 IMPLICIT INTEGER (A-Z)
00005 INTEGER OTSTR(1)
00006 COMMON/TIO/TRMTYP,UPC,DOWNC,LEFTC,RIGHTC,UDY,TRANS,
00007 POMEC,TABC,ADSBP
00008 IF(TRMTYP.EQ.1) GO TO 100
00009 IF(TRMTYP.EQ.3) GO TO 300
00010 IF(TRMTYP.EQ.2) GO TO 200
00011 TYPE 9999,TRMTYP
00012 9999 FORMAT(' PTBLC DOES NOT SUPPORT TERMINAL TYPE ',I6)
00013 RETURN
00014 100 CALL PUTCHR(14,OTSTR,OUTPTR)
00015 OUTPTR = OUTPTR + 1
00016 RETURN
00017 300 CALL PUTCHR(31,OTSTR,OUTPTR)
00018 OUTPTR = OUTPTR + 1
00019 CALL PUTCHR(14,OTSTR,OUTPTR)
00020 OUTPTR = OUTPTR + 1
00021 RETURN
00022 200 CALL PUTCHR(127,OTSTR,OUTPTR)
00023 OUTPTR = OUTPTR + 1
00024 RETURN
00025 END

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
00001      SUBROUTINE INIT
00002      C
00003      C      THIS SUBROUTINE OBTAINS THE TERMINAL TYPE NUMBER FROM THE
00004      C      USER AND SETS THE VALUES OF APPROPRIATE CHARACTERS.
00005      C
00006      IMPLICIT INTEGER (A-Z)
00007      COMMON/TIO/TRMTYP,UPC,DOWNC,LEFTC,RIGHTC,UDX,TRANS,
00008      1      HOMEC,TABC,ADS8P
00009      C
00010      C-----ASK USER FOR TERMINAL TYPE
00011      20     CONTINUE
00012      TYPE 1001
00013      TYPE 1000
00014      CALL GETINT(TRMTYP)
00015      C
00016      C-----CHECK FOR VALID TYPES
00017      IF(TRMTYP .EQ. 1) GO TO 50
00018      IF(TRMTYP .EQ. 2) GO TO 60
00019      GO TO 20
00020      C
00021      C-----SETUP CHAR DEFINATION FOR TABLE TOP DATAMEDIA 2500
00022      50     CONTINUE
00023      CALL STERM
00024      UPC = 26
00025      DOWNC = 10
00026      LEFTC = 3
00027      RIGHTC = 28
00028      TRANS = 17
00029      HOMEC = 2
00030      TABC = 9
00031      CRC = 13
00032      ERASEC = 30
00033      PROC = 127
00034      UNPROC = 127
00035      BLC = 14
00036      UNBLC = 24
00037      GO TO 200
00038      C
00039      C-----SETUP CHAR DEFINATIONS FOR PORTABLE DATAMEDIA 1520
00040      60     CONTINUE
00041      CALL STERM
00042      UPC = 31
00043      DOWNC = 10
00044      LEFTC = 8
00045      RIGHTC = 28
00046      TRANS = 17
00047      HOMEC = 25
00048      TABC = 9
00049      CRC = 13
00050      ERASEC = 12
00051      PROC = 127
00052      UNPROC = 127
00053      BLC = 127
00054      UNBLC = 127
00055      GO TO 200
00056      C
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
00057 C-----RETURN TO CALLER
00058 200      CONTINUE
00059          RETURN
00060          C
00061 C-----FORMATS
00062 1000    FORMAT(" TERMINAL TYPE?")
00063 1001    FORMAT(" ACCEPTABLE TYPES ARE",/,,
00064     1   " 1 = TABLE-TOP DATAMEDIA MODEL 2500.",/,,
00065     2   " 2 = PORTABLE DATAMEDIA MODEL 1520.",")
00066          END
```

~~THIS PAGE IS BEST QUALITY PRACTICABLE~~
~~FROM COPY FURNISHED TO DDC~~

```
00001      SUBROUTINE MOVC(Y,X)
00002      C
00003      C      THIS SUBROUTINE POSITIONS THE CURSOR ON THE TERMINAL VIA X,Y ADD
00004      C          Y=ROW NUMBER, FIRST ROW IS 1
00005      C          X=CUL NUMBER, FIRST COL IS 1
00006      C
00007      IMPLICIT INTEGER (A-Z)
00008      COMMON/TIO/TRMTYP,UPC,DOWNC,LEFTC,RIGHTC,UDX,TRANS,
00009      1          HOMEC,TABC,ADSBP
00010      IF(TRMTYP .EQ. 1) GO TO 100
00011      IF(TRMTYP .EQ. 2) GO TO 110
00012      GO TO 500
00013      C
00014      C
00015      100     CALL PTERM
00016          CALL SFNO(12,UDX)
00017          J=X+96
00018          IF(X.GT.31) J=J-64
00019          IF(X.GT.63) J=J-64
00020          CALL SEND(J,UDX)
00021          J=Y+96
00022          CALL SEND(J,UDX)
00023          CALL STERM
00024          GO TO 500
00025      C
00026      C-----
00027      110     CONTINUE
00028          CALL BTERM
00029          CALL SEND(30,UDX)
00030          J=X+31
00031          CALL SEND(J,UDX)
00032          J=Y-1
00033          CALL SEND(J,UDX)
00034          CALL STERM
00035          GO TO 500
00036      C
00037      C
00038      500     CONTINUE
00039          RETURN
00040          END
```

```
00001      SUBROUTINE NWSCRN(ACLR)
00002      C
00003      C      THIS SUBROUTINE CLEARS THE SCREEEN AND HOMES THE CURSOR
00004      C
00005      IMPLICIT INTEGER (A-Z)
00006      COMMON/TIO/TRMTYP,UPC,DOWNC,LEFTC,RIGHTC,UDX,TRANS,
00007      1      HOMEC,TABC,ADSRR
00008      CRC=13
00009      ERASEC=30
00010      IF(TRMTYP.EQ.2) ERASEC=12
00011      C
00012      C-----CHECK FOR ERASABLE TERM TYPE
00013      IF(TRMTYP.EQ.1) GO TO 10
00014      IF(TRMTYP.EQ.2) GO TO 10
00015      GO TO 200
00016      C
00017      C-----ERASE SCREEN
00018      10     CONTINUE
00019      CALL SEND(CRC,UDX)
00020      CALL BTFRM
00021      CALL SEND(ERASEC,UTX)
00022      CALL SEND(29,UDX)
00023      CALL STFRM
00024      GO TO 200
00025      C
00026      C-----RETURN TO CALLER
00027      200     CONTINUE
00028      RETURN
00029      END
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
00001      SUBROUTINE FORMS(UFLD)
00002      C
00003      C      THIS SUBROUTINE SCANS A FORM REPLACING USER FIELD CHARACTERS
00004      C      WITH THE CHARACTER COUNT INDEX. IT ALSO RECORDS DESCRIPTIVE
00005      C      DATA ABOUT THE FORM FOR OTHER ROUTINES.
00006      C
00007      C
00008      COMMON /CURTAB/ FORMNO, FORM(24,80)
00009      COMMON /SPECS/ ITABFR, ITABLR, ITABNR, ITABNF, IFLDS(14,15),
00010      1          IBLNK(15,17), IFORMT(2,19)
00011      INTEGER FORM, UFLD, BLANK, LNEND, FORMNO
00012      DATA BLANK,LNEND/1H , -1/
00013      C
00014      C-----INITIALIZE VARIABLES
00015      ITABFR = -1
00016      ITABLR = -1
00017      ITABNR = 0
00018      ITABNF = 0
00019      COLN = 0
00020      DO 10 I=1,15
00021      IFORMT(1,I) = 0
00022      IFORMT(2,I) = 0
00023      10      CONTINUE
00024      JFLD = 1
00025      IROW = 0
00026      ICOL = 0
00027      C
00028      C-----BEGIN SCAN OF NEXT ROW/LINE
00029      20      CONTINUE
00030      IROW = IROW+1
00031      IF(IROW .GT. 24) GO TO 100
00032      LCOLN = COLN
00033      ICOL = 0
00034      LEN = 0
00035      NCHAR = 0
00036      C
00037      C-----SCAN NEXT CHARACTER
00038      30      CONTINUE
00039      ICOL = ICOL+1
00040      IF(ICOL .GT. 90) GO TO 70
00041      C
00042      C-----PROCESS A USER FIELD CHARACTER
00043      IF(FORM(IROW,ICOL) .NE. UFLD) GO TO 50
00044      LFN = ICOL
00045      COLN = COLN+1
00046      FORM(IPDN,ICOL) = COLN
00047      IF(ITABFR .NE. -1) GO TO 30
00048      NCHAR = NCHAR+1
00049      IF(NCHAR .EQ. 1) IFORMT(1,JFLD) = ICOL-1
00050      GO TO 30
00051      C
00052      C-----PROCESS A NON-USER FIELD CHARACTER
00053      50      CONTINUE
00054      IF(FORM(IROW,ICOL) .NE. BLANK) LEN=ICOL
00055      IF(NCHAR .EQ. 0) GO TO 30
00056      IFORMT(2,JFLD) = NCHAR
```

```
00057      NCHAR = 0
00058      JFLD = JFLD+1
00059      GO TO 30
00060      C
00061      C-----END OF CHARACTER SCAN FOR THIS LINE
00062      70      CONTINUE
00063      IF(NCHAR .EQ. 0) GO TO 80
00064      IFORMT(2,JFLD) = NCHAR
00065      NCHAR = 0
00066      JFLD = JFLD+1
00067      30      CONTINUE
00068      C
00069      C-----MARK ACTUAL END OF LINE IF LESS THAN 80 CHARACTERS
00070      IF(LEN .LT. 80) FORM(IROW,LEN+1)=LNEND
00071      C
00072      C-----NOTE IF ANY USER FIELD CHARACCTERS WERE IN THIS LINE
00073      IF(COLN .EQ. LCOLN) GO TO 90
00074      IF(LCOLN .EQ. 0) ITABFR = IROW-1
00075      ITABNR = ITABNR+1
00076      ITABLR = IROW-1
00077      90      CONTINUE
00078      C
00079      C-----GO FOR NEXT LINE TO SCAN
00080      GO TO 20
00081      C
00082      C-----ALL LINES OF THIS FORM SCANNED, NOTE NUMB. OF USER FLDS.
00083      100      CONTINUE
00084      ITABNF = JFLD-1
00085      RETURN
00086      END
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
00001      SUBROUTINE LIST0(IT,ITFMP)
00002      C
00003      C      THIS ROUTINE OUTPUTS A TABLE TO THE LINE PRINTER
00004      C
00005      C      INTEGER FORM, UFLD, FORMNO, OTSTR, FCHAR, GETCHR, OUTPTP,
00006      1      COLN, USR PTR, TRMTYP, UPC, DOWNC, LEFTC, RIGHTC, TRANSC,
00007      2      HOMEc, TABC, CRC, ERASEC, PROC, UNPROC, BLC, UNBLC,
00008      3      CPXPTR, CFLD, UROW, XCHAR, RFLD, DFLD, DDAY, BLANK,
00009      4      RUNPTR, DAYPTR
00010      C
00011      C      DOUBLE PRECISION ITEMP, NULL, IUTYPE, IETYPE, IUSUB,
00012      1      ICMLPX, MISS, IESUB, KTEMP
00013      DIMENSION ITFMP(19,17), KTEMP(20)
00014      DIMENSION OTSTR(16)
00015      COMMON/MODS/IDRUN, IDAY
00016      DOUBLE PRECISION IDRUN
00017      COMMON /SPECs/ ITABFR, ITABLR, ITABNF, IFLDS(14,15),
00018      1      IBLNK(15,17), IFORMT(2,19)
00019      COMMON /CURTAB/ FORMNO, FORM(24,80)
00020      COMMON /IFACE/ IUTYPE(2,15), IETYPE(2,15), IUSUB(2,15,15),
00021      1      IESUB(2,15,15), ICMLPX(2,8), MISS(2,17)
00022      DATA UFLD/"725004020100/
00023      DATA CFLD/"615004020100/
00024      DATA RFLD/"711004020100/
00025      DATA DFLD/"621004020100/
00026      DATA BLANK/1H /
00027      C
00028      C-----READ-IN TABLE FROM FORMS FILE
00029      IF(IT.LT.15 .AND. IT.EQ.FORMNO) GO TO 20
00030      DO 10 IROW = 1,24
00031      INX = IROW+24*(IT-1)
00032      READ(23$INX,1000) (FORM(IROW,ICOL),ICOL=1,80)
00033      10    CONTINUE
00034      FORMNO=IT
00035      C
00036      C-----REPLACE "u" WITH DATA CHAR COUNT
00037      CALL FORMS(UFLD)
00038      C
00039      C-----GET COMPLEX HEADERS FOR TABLE 13
00040      IF(IT .NE. 13) GO TO 17
00041      DO 16 I=1,8
00042      KTEMP(I)=ICMLPX(1,I)
00043      KTEMP(I+8)=ICMLPX(2,I)
00044      16    CONTINUE
00045      17    CONTINUE
00046      C
00047      C-----ENCODE RATTLE DAYS FOR OUTPUT TABLES
00048      IF(IT .GT. 14) ENCODE(2,1001,DDAY) TODAY
00049      C
00050      C-----INITIALIZE POINTERS
00051      20    CONTINUE
00052      IROW=0
00053      NFLD=0
00054      NCFLD=0
00055      C
00056      C-----BEGIN NEW ROW
```

```
00057    30    CONTINUE
00058    IROW=IROW+1
00059    IF(IROW .GT. 23) GO TO 200
00060    ICOL=0
00061    USRPTR=0
00062    CPXPTR=0
00063    RUNPTR=0
00064    DAYPTR=0
00065    DO 35 I=1,16
00066    OTSTR(I) = BLANK
00067    35    CONTINUE
00068    NFLD=0
00069    C
00070    C-----NEXT COLUMN
00071    40    CONTINUE
00072    ICOL=ICOL+1
00073    IF(ICOL .GT. 80) GO TO 100
00074    FCHAR=FORM(IROW,ICOL)
00075    IF(GETCHR(FCHAR,1) .EQ. 0) GO TO 50
00076    IF(FCHAR .EQ. CFLD) GO TO 70
00077    IF(FCHAR .EQ. RFLD) GO TO 80
00078    IF(FCHAR .EQ. DFLD) GO TO 90
00079    IF(FCHAR .EQ. -1) GO TO 100
00080    C
00081    C-----OUTPUT A FORMS CHARACTER
00082    CPXPTR=0
00083    USRPTR=0
00084    RUNPTR=0
00085    DAYPTR=0
00086    CALL PUTCHA(FCHAR, OTSTR, ICOL)
00087    GO TO 40
00088    C
00089    C-----BEGIN A USER DATA FIELD "u"
00090    50    CONTINUE
00091    RUNPTR=0
00092    DAYPTR=0
00093    CPXPTR=0
00094    IF(USRPTR .NE. 0) GO TO 60
00095    NFLD = NFLD+1
00096    UROW = IROW-ITABFR
00097    IF(IPLNK(NFLD,UROW) .EQ. 0) GO TO 60
00098    CALL PUTCHR(62,OTSTR,ICOL-1)
00099    60    CONTINUE
00100    USRPTR=USRPTR+1
00101    CALL GETCHA(UCHAR, ITEMP(NFLD,UROW), USRPTR)
00102    CALL PUTCHA(UCHAR, OTSTR, ICOL)
00103    GO TO 40
00104    C
00105    C-----PROCESS A COMPLEX HEADER FIELD "c"
00106    70    CONTINUE
00107    RUNPTR=0
00108    DAYPTR=0
00109    USRPTR=0
00110    IF(CPXPTR .EQ. 0) NCFLD=NCFLD+1
00111    CPXPTR=CPXPTR+1
00112    CALL GETCHA(XCHAR, KTEMP(NCFLD), CPXPTR)
```

THIS PAGE IS BEST QUALITY PRACTICABLY
FROM COPY FURNISHED TO DDC

```
00113      CALL PUTCHA(XCHAR, OTSTR, ICOL)
00114      GO TO 40
00115      C
00116      C-----PROCESS A RUNID FIELD "r"
00117      80      CONTINUE
00118      CPXPTR=0
00119      USRPTR=0
00120      DAYPTR=0
00121      RUNPTR=RUNPTR+1
00122      XCHAR=BLANK
00123      IF(RUNPTR .LE. 10) CALL GETCHA(XCHAR, IDRUN, RUNPTR)
00124      CALL PUTCHA(XCHAR, OTSTR, ICOL)
00125      GO TO 40
00126      C
00127      C-----PROCESS A BATTLE DAYS FIELD "d"
00128      90      CONTINUE
00129      CPXPTR=0
00130      USRPTR=0
00131      RUNPTR=0
00132      DAYPTR=DAYPTR+1
00133      XCHAR=BALNK
00134      IF(DAYPTR .LE. 2) CALL GETCHA(XCHAR, ODAY, DAYPTR)
00135      CALL PUTCHA(XCHAR, OTSTR, ICOL)
00136      GO TO 40
00137      C
00138      C-----END OF LINE ENCOUNTERED
00139      100     CONTINUE
00140      WRITE(3,1002) (OTSTR(I), I=1,16)
00141      GO TO 30
00142      C
00143      C-----ENTIRE TABLE LISTED, RETURN TO CALLER
00144      200     CONTINUE
00145      RETURN
00146      C
00147      C-----FORMATS
00148      1000    FORMAT(P0A1)
00149      1001    FORMAT(J2)
00150      1002    FORMAT(1X,16A5)
00151      END
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
1          TITLE    SLIB3 Fortran Library of usefull routines
2          SUBTTL   GETCHA Getting a char from a char-string
3
4          ; Calling Sequence:
5          ; CALL GETCHA (LCHAR, NWORD, NCHAR)
6          ; LCHAR=variable to receive the character (leftmost) that is gotten
7          ; NWORD=variable or array to bias NCHAR's from for source
8          ; NCHAR=integer, character position number relative to the
9          ;       leftmost character of NWORD. The first character is 1,
10         ;       the second character is 2, etc.
11
12
13 000000* 47 45 64 43 50 41
14 000001*
15 000001* 550 00 1 16 000002
16 000002* 360 00 0 00 000000
17 000003* 231 00 0 00 000005
18 000004* 201 02 0 16 000001
19 000005* 200 02 0 02 000000
20 000006* 603 02 0 00 000020
21 000007* 324 00 0 00 000005*
22 000010* 553 00 0 00 000002
23 000011* 270 00 0 02 000002
24 000012* 500 00 0 01 000020*
25 000013* 134 01 0 00 000000
26 000014* 242 01 0 00 000025
27 000015* 434 01 0 00 000025*
28 000016* 202 01 1 16 000000
29 000017* 261 17 0 00 000000
30
31
32 000020*
33 000020* 44 07 0 00 000000
34 000021* 35 07 0 00 000000
35 000022* 26 07 0 00 000000
36 000023* 17 07 0 00 000000
37 000024* 10 07 0 00 000000
38 000025* 001004 020100
39
40          ENTRY    GETCHA
        S16BIT  "GETCHA"
        GETCHA:
        HRRZ    0,0? (16)      ; get char position value
        SDJ    0,                   ; correct position to zero base
        IDIVI   0,5                ; calc words & chars
        MOVEI   2,1(16)           ; get 2nd item in arg list
        END:    MOVEF   2,(7)          ; get 2nd item in arg list
        TNE    2,20                ; test for inj. bit on
        JUMPA   IND
        HRRZS   2
        ADD    0,2                ; add word's to wd addr
        HLL    0,PTAB(1)           ; get ptr bits from table
        ILOR   1,0                ; get char
        LSH    1,35                ; shift char to left of word
        IOR    1,BMASK             ; blank out remaining chars
        MOVEV   1,20(16)           ; put char in return word
        POPJ    17,                 ; return to caller
        ;
        ;
        PTAB:
        POINT   7,0
        POINT   7,0,6
        POINT   7,0,13
        POINT   7,0,20
        POINT   7,0,27
        BMASK:
        OCT    1004020100
        PPGEND
```

**THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC**

```
41          SUBTTL PUTCHA Putting a char into a char-string
42
43          ; Calling Sequence:
44          ; CALL PUTCHA (LCHAR, NWORD, NCHAR)
45          ; LCHAR=variable containing character(leftmost) to be put
46          ; NWORD=variable or array to bias NCHAR's from for dest.
47          ; NCHAR=Integer, character position number relative to
48          ;   the leftmost character of NWORD. The first
49          ;   character is 1, the second character is 2, etc.
50
51          ENTRY    PUTCHA
52          STABIT  "PUTCHA"
53
54          PUTCHA:
55          HRRZ    0,@2(16)      ; get char position value
56          SOJ     0,
57          IDIVI  0,5           ; correct position to zero base
58          MOVEI   2,1(16)
59          TND:    MOVE   2,(2)
60          TLNE   2,20
61          JUMPA  1"0
62          HRRZS  2
63          ADD    0,2
64          HLL    0,PTAB(1)      ; add wds to word address
65          MOVE   1,@(16)        ; get char ptr bits from table
66          ROT    1,7           ; get char to be loaded (leftmost)
67          IDPR   1,0           ; shift char to rightmost
68          PDRJ   1"              ; deposit char into string
69
70          PTAB:
71          POINT  7,0
72          POINT  7,0,6
73          POINT  7,0,13
74          POINT  7,0,20
75          POINT  7,0,27
76          PGEND
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
77          SUBTTL OUTSTR Output a string to the controlling terminal
78          SEARCH  MNSYM      ; search system monitor library
79
80
81
82
83
84
85
86
87
88
89 000000* 57 65 44 63 64 62
90 000001*
91 000001* 201 02 0 16 000001
92 000002*
93 000002* 200 02 0 02 000000
94 000003* 603 02 0 00 000020
95 000004* 324 00 0 00 000002*
96 000005* 550 01 0 00 000002
97 000006* 500 01 0 00 000011*
98 000007* 104 00 0 00 000076
99 000010* 263 17 0 00 000000
100          PRGFEND

          ; Calling Sequence:
          ; CALL OUTSTR (DUMMY, ISTRNG)
          ; DUMMY=unused argument
          ; ISTRNG=variable or array containing the string to be output.
          ; The string is expected to begin with the leftmost character
          ; of ISTRNG and proceed to a zero byte. The zero byte will
          ; not be output.

          ; ENTRY    OUTSTR
          ; SIXPIT  "OUTSTR"
          OUTSTR:   MOVEI   2,L(16)
          IND:      MOVEF   2,(2)
                     TLNF    2,20
                     JUMPA   IND
                     HRFZ   1,2
                     HLL    1,POINT 7,01 ; string addr to pntr AC
                     PSOUT  17,           ; pntr hdr to pntr AC
                     POPJ   17,           ; output string to zero byte
                     ; return to caller
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
101          SUBTL  SEND Output a byte to the controlling terminal
102          SEARCH  MONSYM      ; search system monitor library
103
104          ; Calling Sequence:
105          ; CALL SEND( CHAR, DUMMY)
106          ; CHAR=character to output, right-justified
107          ; DUMMY=unused argument
108
109          ENTRY   SEND
110 000000* 03 4E #t 44 00 00    SEND:
111 000001*           SIXPIT  "SEND"
112 000002* 20C 01 1 16 000000    MOVE    1,60(16)      ; get word containing char right-just.
113 000002* 104 00 0 00 000074    PPUT    17,             ; output one byte
114 000003* 263 17 0 00 000000    POPJ    17,             ; return to caller
115
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
116          SUBTTL SLP Pauses process for specified time
117          SEARCH MONSVM      ; search system monitor library
118
119          ; Calling Sequence:
120          ; CALL SLP( MILSEC )
121          ; MILSEC=Integer, number of milliseconds to dismiss process
122
123          *          ENTFY   SLP
124 000000* 63 54 60 00 00 00      STX9IT "SLP"
125 000001*           SLP:
126 000001* 200 01 1 16 000000      MOVE    1,30(16)      ; get dismissal time period
127 000002* 104 00 0 00 000167      DISW3      ; dismiss process
128 000003* 263 17 0 00 000000      POPJ    17,          ; return to caller
129
130          PRGEND
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
130          SUBTTL GET Get next byte from control terminal
131          SFARSH MONSYM      ; search system monitor library
132          ;
133          ; Calling Sequence:
134          ; CALL GETT (INCHAR)
135          ; INCHAR=contains the next byte from the controlling terminal
136          ;           right justified zero filled
137          ;
138          ENTRY  GETT
139          000000* 47 45 64 64 00 00
140          000001*          SIXBIT "GETT"
141          000001* 104 00 C 00 000073
142          000002* 405 01 0 00 000177
143          000003* 202 C1 1 16 000000
144          000004* 263 17 0 00 000000
145          CETT:
          PRIN          ; get nxt byte from term
          ANDI 1,177      ; mask off parity and garbage
          MOVFM 1,00(16)  ; store in argument
          POPJ 17,         ; return to caller
          P2GFND
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```

146          SUBTIL PUTCHR Putting rightmost data into byte stream
147
148
149
150
151
152
153
154
155
156
157 000000* 60 55 64 43 50 62
158 000001*
159 000001* 550 00 1 16 000002
160 000002* 360 00 0 00 000000
161 000003* 231 00 0 00 000005
162 000004* 201 02 0 16 000001
163 000005*
164 000005* 200 02 0 02 000000
165 000006* 603 02 0 00 000020
166 000007* 324 00 0 00 000005*
167 000010* 533 00 0 00 000002
168 000011* 270 00 0 00 000002
169 000012* 500 00 0 01 000016*
170 000013* 200 01 1 15 000000
171 000014* 136 01 0 00 000009
172 000015* 262 17 0 00 000000
173
174
175 000016*
176 000016* 44 07 0 00 000000
177 000017* 35 07 0 00 000000
178 000020* 26 07 0 00 000000
179 000021* 17 07 0 00 000000
180 000022* 10 07 0 00 000000
131
;
```

; Calling Sequence:
; CALL PUTCHR (IRCHAR, NWORD, NCHAR)
; IRCHAR=variable containg the data to be put rightmost 7 bits
; NWORD=variable or array to bias NCHAR's from for dest.
; NCHAR=Integer, character position number relative to
; the leftmost character of NWORD. The first
; character is 1, the second character is 2, etc.

;

ENTRY PUTCHR
SIXBIT *PUTCHR*

PUTCHR:

HRKZ 0,02(16) ; get char position value
SQJ 0, ; correct position to zero base
IDIVI 0,5 ; calc position in wds & chars
MOVEI 2,1(16)

IND:

MOVE 2,(0)
TLNE 2,20
JUMPA IND
HRRZS 2
ADD 0,2 ; add wds to word address
HLL 0,PTAB(1) ; get char ptr bits form table
MOVE 1,0(16) ; get data to be loaded (rightmost)
IDPB 1,0 ; deposit char into string
PDEJ 17, ; return to caller

;

;

PTAB:

POINT 7,0
POINT 7,0,6
POINT 7,0,13
POINT 7,0,20
POINT 7,0,27
PRGFND

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

182	SUBTTL	STERM Initializes term characteristics
183	SEARCH	MCSYMS ; search system monitor library
184	ENTRY	STERM
185 000000* 63 64 45 62 55 00	SIXBIT	"STERM"
186 000001*	STERM:	
187 000001* 200 01 0 00 000006*	MOVE	1,EWWD 0,.PRIOUJ
188 000002* 104 00 0 00 000107	RFMOD	
189 000003* 435 02 0 00 174300	ORI	2,174300
190 000004* 104 00 0 00 000110	SFMOD	
191 000005* 263 17 0 00 000000	POPJ	17,
192	PRCEND	

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

193		SHTTL	TERM Puts terminal in binary mode
194		SFACCH	MNSYM
195		ENTRY	BTERM
196	000000* 42 64 45 62 55 00	SIXPIT	"BTERM"
197	000001*	BTERM:	
198	000001* 200 01 0 00 000007*	MOVE	1,EYWD 0,.PRINW
199	000002* 104 00 0 00 000107	REMOD	
200	000003* 404 02 0 00 000005*	AND	2,EMASK
201	000004* 104 00 0 00 000110	SEMOD	
202	000005* 262 17 0 00 000000	POPJ	17,
203	000006* 777777 773477	BMASK:	XWD 777777,773477
204		PRGEND	

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
205          SUBTTL GETCHR Getting a char(r-format) from a string
206
207          ; Calling Sequence:
208          ; INTEGER FUNCTION ICHAR = GETCHR(LSTRING, NCHAR)
209          ; ICHAR=will contain the extracted char right-justified zero filled
210          ; LSTRING=variable or array containing the string
211          ; NCHAR=Integer, relative position of char in string
212
213          ; ENTRY      GETCHR
214          000000* 47 45 54 43 50 62
215          000001*      S1KEIT  "GETCHR"
216          000001* 550 00 1 16 000001
217          000002* 360 00 0 00 000000
218          000003* 231 00 0 00 000005
219          000004* 202 07 0 00 000024*
220          000005* 201 02 0 16 000000
221          000006*      MOVEI   2,(16)      ; get addr of fst item in arg list
222          000006* 200 02 0 02 000000
223          000007* 603 02 0 00 000020
224          000010* 324 00 0 00 000006*
225          000011* 553 00 0 00 000002
226          000012* 270 00 0 00 000002
227          000013* 500 00 0 01 000017*
228          000014* 134 00 0 00 000000
229          000015* 200 07 0 00 000024*
230          000016* 262 17 0 00 000000
231          ;           IND:
232          ;           MOVE  2,(?)      ; replace addr with contents of addr
233          000017*      PTAB:    TLNF  2,20      ; test for ind. bit on, skip if off
234          000017* 44 07 0 00 000000
235          000020* 35 07 0 00 000000
236          000021* 26 07 0 00 000000
237          000022* 17 07 0 00 000000
238          000023* 10 07 0 00 000000
239          000024* 000000 000000
240          SV7:      POINT 7,0
                  POINT 7,0,6
                  POINT 7,0,13
                  POINT 7,0,20
                  POINT 7,0,27
                  OCT   0
                  END
```

DISTRIBUTION LIST

<u>ORGANIZATION</u>	<u>NO. OF COPIES</u>	<u>ORGANIZATION</u>	<u>NO. OF COPIES</u>
Director, Engineering Psychology Programs (Code 455) Office of Naval Research 800 North Quincy Street Arlington, Virginia 22217	5	Fleet Analysis and Support Division, Code 230 Office of Naval Research 800 North Quincy Street Arlington, Virginia 22217	1
Organizational Effectiveness Programs Office of Naval Research (Code 452) 800 North Quincy Street Arlington, Virginia 22217	1	Naval Analysis Programs, Code 431 Office of Naval Research 800 North Quincy Street Arlington, Virginia 22217	1
Defense Documentation Center Cameron Station Alexandria, Virginia	12	Operations Research Program, Code 434 Office of Naval Research 800 North Quincy Street Arlington, Virginia 22217	1
Lt. Col. Henry L. Taylor, USAF OOD (E&LS) ODDR&E Pentagon, Room 3D129 Washington, D.C. 20301	1	Statistics and Probability Program, Code 436 Office of Naval Research 800 North Quincy Street Arlington, Virginia 22217	1
Capt. Roger Gramum Office of Assistant Secretary of Defense (Intelligence), Pentagon Washington, D.C. 20301	1	Information Systems Program, Code 437 Office of Naval Research 800 North Quincy Street Arlington, Virginia 22217	1
Dr. Robert Young Director Cybernetics Technology Office Defense Advanced Research Projects Agency 1400 Wilson Boulevard Arlington, Virginia 22209	1	Director, ONR Branch Office 495 Summer Street Boston, Massachusetts 02210 Attn: Dr. J. Lester	1
Human Factors Plans, OP987P7 Office of the Chief of Naval Operations Department of the Navy Washington, D.C. 20350	1	Director, ONR Branch Office 536 South Clark Street Chicago, Illinois 60605 Attn: Psychologist	1
Personnel Logistics Plans, OP987P10 Office of the Chief of Naval Operations Department of the Navy Washington, D.C. 20350	1	Director, ONR Branch Office 1030 East Green Street Pasadena, California 91106 Attn: Dr. E. Gloye	1
Dr. A. L. Slafkosky Scientific Advisor Commandant of the Marine Corps Code RD-1 Washington, D.C. 20380	1	Director, ONR Branch Office 1030 East Green Street Pasadena, California 91106 Attn: Mr. R. Lawson	1
Assistant Chief for Technology, Code 200 Office of Naval Research 800 North Quincy Street Arlington, Virginia 22217	1	Office of Naval Research (Code 102IP) Department of the Navy 800 North Quincy Street Arlington, Virginia 22217	6

ORGANIZATION	NO. OF COPIES	ORGANIZATION	NO. OF COPIES
Dr. Fred Muckler Manned Systems Design, Code 311 Navy Personnel Research and Development Center San Diego, California 92152	1	Dr. Joseph Zeidner Director, Organization and Systems Research Laboratory U.S. Army Research Institute 1300 Wilson Boulevard Arlington, Virginia 22209	1
Dr. Charles Gettys, Code 305 Navy Personnel Research and Development Center San Diego, California 92152	1	Dr. Donald A. Topmiller Chief, Systems Effectiveness Branch Human Engineering Division Wright Patterson AFB, Ohio 45433	1
Navy Personnel Research and Development Center Management Support Department, Code 210 San Diego, California 92152	1	Dr. H. W. Sinaiko Smithsonian Institution 801 N. Pitt Street Alexandria, Virginia 22314	1
Naval Electronics System Command Human Factors Engineering Branch Code 4701 Washington, D.C. 20360	6	Dr. Gary Lucas System Planning Corporation 1500 Wilson Boulevard Arlington, Virginia 22209	1
Director, Naval Research Laboratory Technical Information Division Code 2627 Washington, D.C. 20375	6	Dr. C. Peterson Decisions and Designs, Inc. 8400 Westpart Drive, Suite 600 McLean, Virginia 22101	1
Mr. Arnold Rubinstein Naval Material Command, NAVMAT 0344 Department of the Navy Washington, D.C. 20360	1	Navy C3 Architecture Division OP-943 Office of the Chief of Naval Operations 3801 Nebraska Ave., N.W. Washington, D.C. 20390 Attn: LCDR D. A. Spaugy	1
Mr. Richard Coburn Head, Human Factors Division Naval Electronics Laboratory Center San Diego, California 92152	1	Mr. L. A. Aarons R&D Plans Division Office of the Chief of Naval Operations OP-987C Washington, D.C. 20350	1
Dr. Jesse Orlansky Institute for Defense Analyses 400 Army-Navy Drive Arlington, Virginia 22202	1	Commander, Naval Electronics Systems Command Command and Control Division, Code 530 Washington, D.C. 20360	1
Human Factors Department, Code N215 Naval Training Equipment Center Orlando, Florida 32813	1	Commander, Naval Electronics Systems Command C3 Project Office PME 108-1 Washington, D.C. 20360 Attn: LCDR E. Neely	1
Dr. Alfred F. Smode Training Analysis and Evaluation Group Naval Training Equipment Center Code N-00T Orlando, Florida 32813	1	LCDR Charles Theisen Naval Air Development Center, Code 4024 Warminster, Pennsylvania 18974	1
Dr. Gary Pock Operations Research Department Naval Postgraduate School Monterey, California 93940	1	Dr. S. D. Epstein Analytics 2500 Maryland Road Willow Grove, Pennsylvania 19090	1

<u>ORGANIZATION</u>	<u>NO. OF COPIES</u>	<u>ORGANIZATION</u>	<u>NO. OF COPIES</u>
Mr. Harold Crane CTEC, Inc. 7777 Leesburg Pike Falls Church, Virginia 22043	1	Lt. Col. David Dianich HQs Tactical Air Command Langley AFB, Virginia 22065	1
Dr. Robert Andrews Organization and Systems Research Laboratory U.S. Army Research Lab 1300 Wilson Boulevard Arlington, Virginia 22209	1	Mr. Victor Monteleon Naval Electronics Laboratory Center Code 230 San Diego, California 92152	1
Mr. George Pugh General Research Corporation 7655 Old Springhouse Road McLean, Virginia 22101	1	CDR Richard Schlaff NIPSSA Hoffman Bldg. No. 1 2461 Eisenhower Avenue Alexandria, Virginia 22331	1
Mr. J. W. Stump Grumman Aerospace Corporation Bethpage, New York 11714	1	Commander, Naval Electronics Systems Command ELEX-03 Washington, D.C. 20360	1
Mr. Gary W. Irving Integrated Sciences Corporation 1532 Third Street Santa Monica, California 90401	1	Dr. Chantee Lewis Management Department Naval War College Newport, Rhode Island 02840	1
Dr. Amos Freedy Perceptronics, Inc. 6271 Variel Avenue Woodland Hills, California 91364	1	Dr. John Shore Code 5403 Communications Sciences Division Naval Research Laboratory Washington, D.C. 20375	1
Dr. Miley Merkhofer Stanford Research Institute Decision Analysis Group Menlo Park, California 94025	1	Dr. Allen C. Miller Applied Decision Analysis 3000 Sand Hill Road Menlo Park, California 94025	1
Mr. Victor Rowney Stanford Research Institute Naval Warfare Research Center Menlo Park, California 94025	11		
Dr. H. L. Morgan University of Pennsylvania Wharton School Philadelphia, Pennsylvania 19174	1		
Lt. Brenneman COMOPTEVFOR Norfolk, Virginia 23511	1		
M. L. Metersky NAVAIRDEVcen, Code 5424 Warminster, Pennsylvania 18974	1		
Ralph M. Tucker Martin Marietta Aerospace Mail Stop 8105 Denver Division P.O. Box 179 Denver, Colorado 80201	1		