

AD-A090 682

RAND CORP SANTA MONICA CA

TSARINA: USER'S GUIDE TO A COMPUTER MODEL FOR DAMAGE ASSESSMENT--ETC(U)

JUL 80 D EMERSON

F49620-77-C-0023

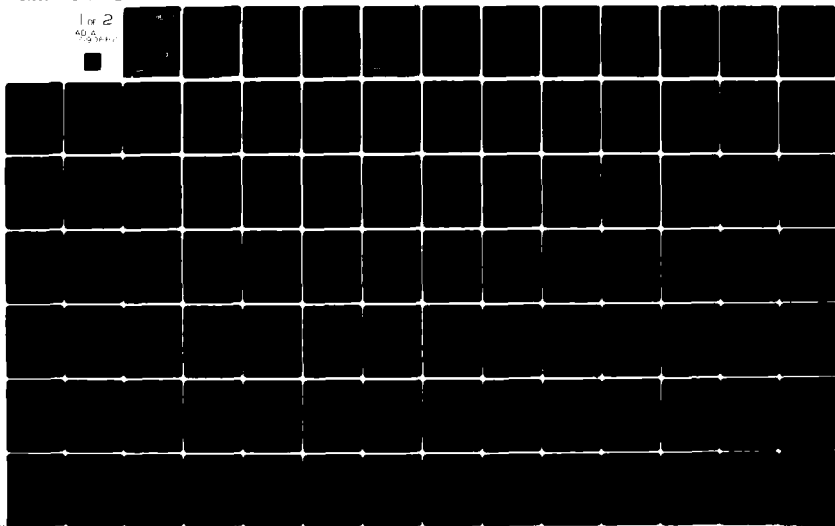
RAND/N-1460-AF

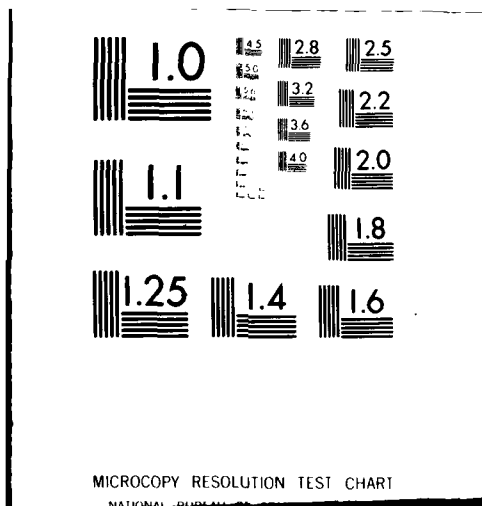
NL

UNCLASSIFIED

1 of 2

AD-A090 682





52

A RAND NOTE

AD A090682

Prepared For

LEVEL II

2

TSARINA--USER'S GUIDE TO A COMPUTER MODEL FOR
DAMAGE ASSESSMENT OF COMPLEX AIRBASE TARGETS

Donald Emerson

July 1980

N-1460-AF

The United States Air Force

DTIC
ELECTE
OCT 21 1980
S D
E

FILE COPY

RAND
SANTA MONICA, CA 90406

80 10 17 071

The research reported here was sponsored by the Directorate of Operational Requirements, Deputy Chief of Staff/Research, Development, and Acquisition, Hq USAF, under Contract F49620-77-C-0028. The United States Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright notation hereon.

The Rand Publications Series: The Report is the principal publication documenting and transmitting Rand's major research findings and final research results. The Rand Note reports other outputs of sponsored research for general distribution. Publications of The Rand Corporation do not necessarily reflect the opinions or policies of the sponsors of Rand research.

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

DD FORM 1473 EDITION OF 1 NOV 68 IS OBSOLETE

~~SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)~~

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

RPN 3509

N-1460-AF TSARINA: User's Guide to a
Computer Model for Damage Assessment of
Complex Airbase Targets.# D. Emerson.
July 1980.

Description of the TSARINA computer program, developed to examine conventional air attacks against complex targets and to assess losses and damage to categories of resources and to buildings and other facilities. TSARINA permits damage assessments of attacks on an airbase complex composed of up to 500 individual targets (buildings, taxiways, etc.), and 1000 packets of resources. Targets may be grouped into 20 vulnerability categories, and different types of personnel, equipment, munitions, spare parts, and other support resources can be distinguished. TSARINA determines the actual impact points by Monte Carlo procedures and the losses and damage are assessed using "cookie-cutter" weapon-effects approximations. TSARINA may be employed separately as a general-purpose model or used in conjunction with the TSAR (Theatre Simulation of Airbase Resources) computer model to assess the impact of airbase damage on sortie generation capabilities and to evaluate proposals for improving those capabilities at an airbase or set of airbases. Detailed user instructions and a listing of the program are included. 123 pp. (Author)

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

A RAND NOTE

TSARINA--USER'S GUIDE TO A COMPUTER MODEL FOR DAMAGE ASSESSMENT OF COMPLEX AIRBASE TARGETS

Donald Emerson

July 1980

N-1460-AF

Prepared For

The United States Air Force

Accession For	
NTIS GRA&I	☒
DDC TAB	☐
Unannounced	☐
Justification	
By	
Distribution/	
Availability Codes	
Dist.	Avail and/or special
A	



PREFACE

This Note describes TSARINA, a special modification of Rand's Airbase Damage Assessment (AIDA) computer model that has been developed for examining conventional air attacks against complex targets and for assessing losses and damage to various categories of resources as well as to various buildings and other facilities. TSARINA may be employed as a general purpose damage assessment model, or as a special purpose model for use with the TSAR (Theater Simulation of Airbase Resources) computer model. This Note includes detailed user instructions as well as a listing of the program.

TSARINA was developed by Rand for use with the TSAR model for studying means of sustaining and improving wartime sortie generation capabilities, despite unexpected demands and sudden unpredictable resource shortages imposed by air attacks. TSARINA is basically a Monte Carlo computer model that can generate sample patterns of airbase damage for incorporation into the TSAR simulation, or statistical summaries of the damage and resource losses for multiple samples of the specified attack.

TSARINA has a variety of possible applications. It can be used separately to assess the casualties and losses that would be sustained from air attacks on airbases (or other complex targets), and to assess the impact of various dispersal and/or hardening proposals on the expected losses. It can also be used in conjunction with the TSAR simulation model to assess the impact of airbase damage on sortie generation capabilities, and to evaluate proposals for improving those capabilities at an airbase or a set of airbases.

TSARINA has been used in a study for the Air Staff of the personnel casualties and War Reserve Materiel (WRM) losses that might be sustained in high-level conventional conflict in Europe or Korea, and it is currently being applied in conjunction with TSAR to examine alternative proposals for improving the Air Force's wartime sortie generation capabilities, under Project AIR FORCE. The model, which has been discussed with many groups within the Air Force, has been transferred to the Office of the Assistant Chief of Staff, Studies & Analyses. This Note is being published to provide documentation for the model and to introduce it to a wider audience. The computer program is available from The Rand Corporation.

This work was conducted under the Project AIR FORCE research project entitled "Strategies To Improve Sortie Production in a Dynamic Wartime Environment."

SUMMARY

This Note describes a new version of the AIDA airbase damage assessment computer program that has been developed to assess losses to various on-base resources, as well as damage to runways, taxiways, buildings, and other facilities. The model may be used either as a general-purpose, complex-target damage assessment model, or as a special-purpose model in support of the TSAR simulation program. When used with TSAR, multiple trials of a multi-base airbase-attack campaign can be assessed with TSARINA, and, in a continuous computer operation, the impact of those attacks on sortie generation can be derived using the TSAR simulation model.

TSARINA, as presently configured, permits damage assessments of attacks on an airbase (or other) complex that is composed of up to 500 individual targets (buildings, taxiways, etc.), and 1000 packets of resources. The targets may be grouped into 20 different vulnerability categories, and many different types of personnel, equipment, munitions, spare parts, TRAP (tanks, racks, adaptors, and pylons), building materials, and POL (petroleum, oils, and lubricants) can be distinguished. The attacks may involve as many as 50 weapon-delivery passes and 10 types of weapons. Both point-impact weapons (such as general-purpose bombs and precision-guided munitions) and area weapons (such as cluster bomb units (CBUs)) can be accommodated.

TSARINA determines the actual impact points (pattern centroids for CBUs) by Monte Carlo procedures--i.e., by random selections from the appropriate error distributions. Weapons that impact within a specified distance of each target type are classed as hits, and estimates of the damage to the structures and to the various classes of support resources are assessed using "cookie-cutter" weapon-effects approximations. In addition to the weapon-effects procedures used with AIDA, this model also permits use of a novel two-level cookie-cutter representation for assessing damage to the various classes of resources.

For each trial computation of an attack, the program determines the fraction of each target covered by the circular damage coverage patterns, and the results include estimates of the overall damage to each target and to all resource classes that are colocated with that target. In addition, the output includes an estimate of the total damage sustained by each type of resource at its various storage locations. The attack may be repeated automatically for several trials to provide statistics on the average damage levels to each of the targets and to each type of resource.

A maximum of five targets may be designated as runways or taxiways suitable for aircraft operations, and the model will examine these to see if an area of a user-specified size is available for aircraft operations; if not, the minimum number of craters that would need to be repaired to obtain an area of that size is determined.

The TSARINA program is written in FORTRAN IV, and should be readily adaptable to other computer systems, as was the widely used AIDA model. This Note provides a full discussion of the use of TSARINA as a general-purpose damage assessment model, and outlines in detail the special requirements for its use in conjunction with the TSAR simulation program. Most features of the model are illustrated with a sample problem. Appendixes include a description of TSARINA input requirements, definitions of all variables and arrays found in TSARINA common statements, and a listing of the complete TSARINA source code.

GLOSSARY

Resource class	All airbase support resources are grouped into seven classes: personnel, equipment and AGE, aircraft spare parts, munitions, TRAP, building materials, and POL.
Resource type	Different types of resources may be distinguished within each resource class; e.g., different categories of aircraft maintenance specialists. Resource, when used alone, implies a resource type.
Resource packet	A user-specified percentage of a given resource is referred to as a resource packet; resources are located within the target complex as packets.
Target	A target is represented by a rectangle that is located in an X-Y coordinate system; individual buildings, runways, taxiways, parking areas, etc., can be designated as targets.
Target complex	A target complex, such as an airfield or an industrial area, is a collection of rectangular targets.
Target type	A target type is specified for each target; all targets of the same type have the same vulnerability, and all resource types of the same resource class located at the same type of target have the same vulnerability.
GP	General-purpose (bomb).
PGM	Precision-guided munition.
POL	Petroleum, oils, and lubricants.
TRAP	Tanks, racks, adaptors, and pylons.
WRM	War reserve materiel.
AGE	Aerospace ground equipment.

ACKNOWLEDGMENTS

The author would like particularly to acknowledge Felix Kozaczka and Louis Wegner of the Rand staff for a variety of suggestions that should make the TSARINA program, and this documentation, more useful and understandable.

CONTENTS

PREFACE.....	iii
SUMMARY.....	v
GLOSSARY.....	vii
ACKNOWLEDGMENTS.....	ix
Section	
I. INTRODUCTION.....	1
II. TARGET DATA.....	3
III. DAMAGE ASSESSMENTS FOR TSAR.....	5
TSAR Relationships Between Facilities and Resources.....	6
Identification of Airbase Facilities.....	8
Procedures to Ensure Resource Category Compatability....	12
IV. WEAPON EFFECTIVENESS DATA.....	15
Point-Impact Weapons.....	15
CBU Munitions.....	19
V. SAMPLE PROBLEM.....	20
Input.....	20
Output.....	27
Appendix	
A. DETAILED DESCRIPTION OF TSARINA INPUT.....	37
B. DEFINITIONS OF VARIABLES AND ARRAYS USED IN TSARINA "COMMON" STATEMENTS.....	59
C. TSARINA SOURCE CODE AND COMMENTS.....	70
D. PROGRAMMERS' NOTES.....	122

I. INTRODUCTION

TSARINA (TSAR Inputs using AIDA) is a modified version of the AIDA (Airbase Damage Assessment) computer model*; it was developed to generate airbase damage estimates for a campaign of air attacks, and to organize those results for direct entry into the TSAR (Theater Simulation of Airbase Resources) sortie generation model,** which can assess the impact of the destructive effects of attacks. Several key changes have been made to the AIDA model so that the on-base location of resources (e.g., personnel, munitions, aircraft spare parts, etc.) can be readily associated with various targets (structures/facilities), and so that different MAEs (mean areas of effectiveness) and/or Pks (kill probabilities) can be defined for the different resources. These changes also permit a novel two-level "cookie-cutter"*** representation of the effectiveness of weapons against the various classes of resources. In addition, the various effectiveness values may be different for direct hits and for near misses. With these added input data, TSARINA generates estimates of the losses among the various on-base resources, in addition to the estimates of hits and facility damage that are generated by the original AIDA model.

TSARINA may be used either as a special-purpose model in support of the TSAR simulation, or as a general-purpose damage assessment model. When used with TSAR, multiple trials of a multi-base airbase-attack campaign can be evaluated with TSARINA, and, in a continuous computer operation, the impact of

*R-1872-PR, AIDA: An Airbase Damage Assessment Model, D. E. Emerson, The Rand Corporation, September 1976.

**R-2584-AF, An Introduction to the TSAR Simulation Program: Model Features and Logic, D. E. Emerson, The Rand Corporation, June 1980.

***See p. 15.

those attacks on sortie generation can then be derived using the TSAR simulation model. When TSARINA is used for damage estimates only, the various protocols required for use with TSAR (Section III) may be ignored.

Since AIDA, and this new version of AIDA, employ identical target and attack representations, the reader is referred to the AIDA report for a discussion of these concepts. The emphasis in this note will be on the special features associated with TSARINA. With both AIDA and TSARINA the user is able to specify the size, location, and nature of several hundred rectangular targets and the characteristics of up to 50 weapon-delivery passes.* Targets can be categorized into 20 vulnerability classes, and up to 10 types of weapons may be employed in any given attack; point-impact and CBU munitions may be used in the same attack. Both TSARINA and AIDA are basically Monte Carlo models; however, an expected-value mode is available as an option for evaluating damage to aircraft shelters.

*The maximum numbers of targets and weapon-delivery passes are readily changed.

II. TARGET DATA

In AIDA and TSARINA, the facilities on an airbase are represented as a target complex consisting of a number of rectangular-shaped targets (e.g., runways, parking ramps, buildings); the size, location,* and type of each target are specified. With TSARINA the user may also specify the resources that are associated with each target. The resources to be identified may be grouped into seven categories; personnel, equipment, aircraft spare parts, munitions, TRAP, building materials, and POL. And within each class, different subclasses may be distinguished by type; for example, the personnel class may distinguish pilots, crew chiefs, radar repair specialists, and weapons loaders. Hereafter, the term resource will refer to a particular resource class and type.

The user may specify the percentages of the different resources that are located at each target. Thus, personnel with different specialties may be located at different facilities; AGE and other equipment can be located in various buildings or parked in designated areas; and different kinds of munitions, TRAP, etc., can similarly be located in various proportions in various on-base locations.

The losses estimated for each resource depend upon the attack weapon type, weapon impact location, resource class, and target type, location, and orientation.** In each case, it is assumed that the resource is distributed uniformly

*AIDA's restrictions on target location have been relaxed for TSARINA. It is no longer necessary that all targets be located in the first quadrant of the X-Y coordinate system, and the targets may cover an area as large as 32000 x 32000 dimension units. When the target location entries are not entirely within the first quadrant, TSARINA automatically translates the coordinate system to place all targets (and attacks) in the first quadrant for processing. Furthermore, an auxiliary program is available that will convert dimensional information structured for the Eglin AFB MASSIVE computer program into the format required for TSARINA.

**All subcategories of a given resource class, located within targets of the same type, are assumed to have the same vulnerability.

within the target area, and that the aggregate losses for that resource are the sum of the losses estimated to be sustained at each of the target locations. Additional weapon effectiveness data must be supplied to complement these extended target descriptors and the user is given considerable flexibility as to how these data are expressed, as discussed in Section IV.

Any number of resource packets may be associated with each target, except that there may be no more than 1000* packets in total. The designations used to specify the different classes of resources are defined in the subsequent discussion of supplementary TGT cards. The "integer" designators that are to be assigned to each type of resource are selected by the user; the only constraint is that the integers chosen are not greater than the size of the corresponding storage array.

TSARINA's treatment of runways is identical with that described for AIDA. Runways must always be identified as type #1 targets; up to five may be entered. When the minimum clear length and width needed for flight operations are identified, and the "minimum repair requirements" option is requested (on the CONT card), all runway targets are searched to find whether or not an uncratered area of the required size exists, and if not, what the smallest number of crater repairs would be to attain that amount of clear space.

*This restraint is easily changed by redefining MXITEM (line 85, page 72) and redimensioning the STOCKS array (line 34, page 71 et al.)

III. DAMAGE ASSESSMENTS FOR TSAR

When TSARINA is to be used to generate damage assessments for the TSAR simulation, it is necessary to make the specifications for the targets and the resources consistent with the conventions used in TSAR. If TSARINA is not to be used with TSAR, but as a general-purpose damage assessment model, the reader may skip to Section IV.

The TSAR computer model is a large, complex, task-oriented event-simulation model that has been developed to interrelate the number of effective sorties that can be generated in wartime at a set of airbases to the resources that are available. It has been designed to provide a means of assessing the potential contributions of various ideas for improving and sustaining sortie capabilities, despite unexpected wartime demands and sudden unpredictable resource shortages imposed by airbase attacks. When used with TSAR, TSARINA generates sample patterns of airbase damage and loss that are incorporated into the simulated TSAR scenario. Special requirements must be satisfied when using TSARINA in order for TSARINA outputs to interface satisfactorily with TSAR.

These additional requirements are a necessary reflection of the considerable complexity and flexibility of the TSAR simulation, and arise as a result of

- o The mechanism employed in TSAR to associate resources with facilities
- o The conventions used in TSAR to identify facilities
- o The procedures provided to permit disparate resource categorizations in the two models.

The following subsections will expand on each of these items.

TSAR RELATIONSHIPS BETWEEN FACILITIES AND RESOURCES

The TSARINA results include estimates of (1) the percentage loss for each resource class at each target and (2) the overall percentage loss for each type of resource at all locations. When used in TSAR, one or the other of these percentages are applied to the quantities of each resource at risk at the time of the attack. For most resources, losses are estimated using the second of these two type of estimates. Since the derivation of this loss percentage in TSARINA is based on a nominal user-specified on-base resource disposition, and is not affected by the particular consumption experience in the TSAR simulation, the use of this overall estimate effectively assumes that the on-base disposition of a given subcategory of resource, expressed in percent, is not dependent upon the absolute level of such resources. The first type of loss estimate is used only for on-duty maintenance personnel, aircraft maintenance equipments, and reparable spare parts; these resources are accounted for individually within TSAR, so that their assigned location at the time of the attack is known.

The TSAR simulation is able to associate these resources with particular facilities because it assigns personnel and equipment to each individual task and those tasks (are normally assumed to) occur in a designated facility. The maintenance personnel engaged in on-equipment maintenance are assumed to be in close proximity to the aircraft to which they are assigned. On-duty maintenance personnel who are not actively engaged in on-equipment maintenance are assumed to be awaiting assignment at their respective work centers, or, in the case of the flightline personnel in a COMO (Combat Oriented Maintenance Organization), in a particular flightline facility. Specialists involved in parts repair jobs are assumed to be in the facility designated for the repair of the particular type of part that they are repairing, as are all reparable spare parts.

Losses among resources engaged in on-equipment maintenance always are conditioned by a separate assessment of the likelihood that the aircraft is exposed to damage (this relationship will be explained shortly). For the others, the nominal TSAR-TSARINA logic presumes that unassigned on-duty personnel and equipment, and personnel and equipment engaged in off-equipment parts repair, are in their designated facilities, and that the loss percentages to these resources are related to the damage to those facilities. In this way, appropriate loss percentages are assigned these mobile resources at the instant of the attack. Damage estimates for facilities and their associated resources are only passed to TSAR for designated targets; this is done by entering the facility number (as defined in the TSAR data structure) on the appropriate TSARINA TGT cards. If the user wishes the target-dependent loss percentages to be transferred to TSAR for some targets, but not for others, the facility numbers for the latter targets should not be entered.

Also, if the user wants the target-dependent loss percentage to be used for some of the resources that are nominally associated with a particular facility, but not for others of those resources, the target-dependent value will be overridden in TSAR for whichever resources have a specific damage estimate transmitted to TSAR. Thus, if on-duty radar technicians have been located in a variety of facilities for the TSARINA airbase attack simulation, the TSARINA estimate of percent damage for those technicians will normally be applied to all such personnel not engaged in on-equipment aircraft maintenance within TSAR, even though a different personnel loss percentage is associated with their normal work facility, and it is passed to TSAR. To prevent the target-dependent estimate from being overridden, the user must either not include the specific locations for these technicians, or specifically instruct TSARINA not to transfer the estimate by use of the EQUI card format, as discussed later in this section.

*When the damage to some, but not all, resources is specified, the dimensions in TSAR's subroutine BOMB currently permit at most 50 damage specifications for each resource class.

Careful attention to which targets are identified with a TSAR facility number, and which resources are located specifically in TSARINA, provides the user substantial flexibility for controlling the damage information transferred from TSARINA to TSAR and thereby representing his knowledge of expected on-base conditions in the simulation data base.

IDENTIFICATION OF AIRBASE FACILITIES

Some airbase facilities are handled in an aggregate manner in TSAR--aircraft shelters, runways, taxiways, and aircraft parking ramps--and for these facilities the user simply specifies the target-type number (on the DATA card, see p. 41) that is used in TSARINA to designate these kinds of facilities. Other facilities--those that relate to specific maintenance functions--are identified within TSAR with specific facility numbers, and these facility numbers must be identified in TSARINA if damage to these facilities is to be communicated to TSAR from TSARINA. The following paragraphs discuss the particular data requirements for each of the several target classes.

Runways

The maximum number of facilities that are distinguished in TSAR is designated by the variable NOFAC. By TSAR convention, the runway is identified by that maximum number, i.e., NOFAC.* For TSARINA the value of NOFAC is specified in the source code of the MAIN subroutine, along with the dimensions of other key TSAR arrays. The user should consult the comments in the MAIN routine for a description of the particular TSARINA configuration he has available; if the configuration is at variance with his TSAR configuration, changes to the TSARINA configuration are easily accomplished.

*Since TSAR is only concerned with the repairs that are to be accomplished to provide the minimum area for flight operations, without regard to which runway is to be repaired, a single datum is required.

Aircraft Parking Ramps

For aircraft parking ramps, TSARINA generates an estimate for the percentage of exposed aircraft that are damaged by computing the expected percentage of aircraft parts that would be destroyed, if they were dispersed at random in these areas, using the rules given in Section IV for specifying aircraft parts vulnerability. In using this estimate, TSAR first subtracts the number of aircraft that could be sheltered from the number of aircraft on the ground, and assumes that the remainder are unsheltered. The survival of the unsheltered aircraft is then checked by comparing a random number for each aircraft with the exposed aircraft damage percentage estimated in TSARINA.

Taxiways

For taxiways TSARINA simply counts the number of point-impact weapons that hit on, or within the weapon radius of, these targets. The TSARINA output identifies these hits with the TSAR target number for taxiways. Hits on the taxiways are communicated to TSAR as damage to facility number NOFAC-1.

Maintenance Shops

In TSAR each on-equipment task and parts repair job is associated with a particular shop or facility. If the shop is damaged, the parts repair jobs and certain designated on-equipment tasks can not be accomplished. Facilities #1 through #24 are reserved for the locations of the various specific maintenance functions (i.e., work-centers or shops) that are designated by the same number. When the facility number for one of these maintenance functions is entered on the TGT card in TSARINA, damage to that facility is communicated to TSAR.

Other facility numbers are reserved for other functions: #27 relates to aircraft reconfiguration, #28 to munitions loading, #29 to fuel servicing, and #30 to munitions assembly; facilities #31, #32, and #33 are used to define the assembly points for the flightline specialists associated with squadrons

one, two, and three, respectively, in a AFR 66-5, or COMO, maintenance organization. Damage to each of these facilities will also be transmitted to TSAR when their facility number is identified on the appropriate TGT card.

When any of these facilities sustain one or more "hits," TSARINA generates an estimate of the percentage of the facility that sustains damage, as well as estimates of the percentage losses sustained by the personnel, AGE, and parts present in the facility at the time of the attack.

Aircraft and Aircraft Shelters

TSAR assesses aircraft damage and loss by drawing a random number for each aircraft on-base at the moment of attack, and by comparing that number with an estimate of the fraction of the aircraft that are damaged and/or killed. In TSARINA a damage fraction is generated both for sheltered and unsheltered aircraft, and TSAR applies the latter fraction to that number of on-base aircraft that exceed the then-current capacity of the shelters, as noted earlier.

Damage to shelters and sheltered aircraft are handled uniquely in TSARINA, in that the damage estimate may be generated either with the Monte Carlo mode (as for other targets) or with the expected-value mode. The Monte Carlo mode is required if the user wishes to consider specific resources that might be stored in the shelters; if he does not have that requirement, he may reduce TSARINA computer processing by use of the expected-value mode.*

With TSARINA the user designates the target type that he has assigned to shelters on the DATA card (see p. 41), and enters the location and size data for each shelter either with the TGT or TGT2 type cards, depending upon whether the Monte Carlo or expected-value mode is to be used for assessing damage.

*Since TSAR compares a random number with the damage estimate for each aircraft, losses will vary from trial to trial even though the expected value mode is used. When the Monte Carlo mode is used, the damage estimate used for these comparisons will also vary between trials.

Weapon effectiveness entries are also handled differently for these targets. In all, 14 different weapon effectiveness data may be entered; 13 for each type of weapon, and one that is presumed to apply for all weapons. To enter these data in TSARINA, 14 cards are entered for each point-impact type weapon, rather than the one or two cards that are used in AIDA. The entry on the first card is interpreted as the effective miss distance against buttoned-up shelters; a hit is recorded whenever a weapon strikes within this distance of a shelter. Data entered on the second through ninth cards control the damage estimates for personnel, AGE, spare parts, munitions, TRAP, and building materials that might be stored in the shelters when a weapon strikes within the first effective miss distance. The several resource damage criteria that are available are explained in Appendix A. When the expected value mode is used, only the third through fifth of these cards is used, and the entries are interpreted as the percent losses sustained by personnel, AGE, and spare parts in a shelter that has closed doors when a hit has been recorded.

Four additional data are entered on the tenth, twelfth, thirteenth and fourteenth EMD cards; the first defines the effective miss distance against aircraft in a shelter with an open door, the next two define the probability of damage to an aircraft when the shelter door is closed, and when it is open, and the last defines the probability of kill of the shelter itself, when a weapon strikes within the first effective miss distance. The factor that controls the fraction of the damaged aircraft that are not reparable is entered on the DATA card, as explained on page 41. This factor is presumed to apply for whatever mix of weapons has been used in the attack.

The various assessments of aircraft and shelter damage, and of the losses among resources associated with the aircraft at the moment of the attack, are communicated to TSAR as class 8 and class 10 in a particular facility; the data structure for these transmissions can be inferred from the code in subroutine DAMAGE.

PROCEDURES TO ENSURE RESOURCE CATEGORY COMPATIBILITY

A requirement for the proper functioning of TSAR and TSARINA is that the level of detail at which resources are specified must be the same for both models. Although it might be adequate for some damage assessments for resource types to be lumped into broad categories, such aggregate results would not be useful for the simulation if these resources were treated in more specific terms in TSAR. Thus, one cannot specify "air-to-air missile losses" in the TSARINA output if TSAR requires that losses of AIM-7s and AIM-9s be distinguished.

The required consistency can be achieved in three different ways. First, TSARINA inputs may specify the location data separately for each type of, say, munition that is to be distinguished in TSAR. Or, if the user cannot distinguish the locations for some types of munitions, he may locate them as a group, and then specify that the same damage should be reported to TSAR for each type of munition in the group; the EQUIvalence card-format is used for this purpose. Lastly, if the individual locations for many, or all, of the various types of any class of resources cannot be distinguished, the user may locate these resources and report their common damage to TSAR by the simple expedient of not specifying a resource number. To clarify these options, consider the following table:

<u>Type-Number Specified in TSARINA Location Data</u>	<u>Types Specified in TSAR</u>	<u>User Action Required</u>
1	1	EQUI data card
	2	
3	3	None
4	4	EQUI data card
	5	
	6	
	7	
0	8	None
	9	
	10	
	11	
	12	

Assuming that the resource class is TRAP, for example, this is how one would handle the situation in which the TSAR data base distinguished 12 different types of TRAP, but location data could only be specified for four different categories, and only one corresponded to the TSAR data base, e.g., TRAP #3. Location data for TRAP #1 and TRAP #2, for TRAP #4, #5, and #6, and for TRAP #7 through #12 cannot be distinguished, presumably, for this example. The only special action the user needs to take is to define the actual types of TRAP, implied by the TSARINA type numbers 1 and 4, using the EQUI data cards. It should be noted that the numbers need not be in any special sequence and that type #0 always implies "all types not otherwise specified."

When dealing with personnel and munitions, two other special features come into play. The first of these is concerned with the distinction between on-duty and off-duty personnel, which are treated differently within TSAR. To indicate that a particular category of personnel is off-duty, the number 1000 is added to the nominal personnel type when their location data are entered.* When this is done, casualties among on- and off-duty personnel are estimated separately in TSARINA and reported separately to TSAR without any additional user input. Specification of personnel type #1000 implies all off-duty personnel types that are not otherwise specified.

The other special number convention permits the user to distinguish the locations and damage to assembled and unassembled munitions; unassembled munitions are identified simply by adding the number 100 to the nominal munition type designation. Thus, if munition #5 were assembled AIM-7s, munition #105 would refer to unassembled AIM-7s. These special identity numbers for personnel

*This feature, as currently implemented, behaves as described when TSAR is dimensioned for 1 to 100 different personnel types. Defining that dimension as NOPEOP, aircrews are identified for TSAR with the number NOPEOP +1; thus, if NOPEOP = 100, aircrews are personnel type #101.

and munitions may either be assigned to the resources when their location data are specified, or subsequently, using the EQUIvalence card type.

IV. WEAPON EFFECTIVENESS DATA

For each weapon type the user has several options for specifying its effectiveness against the various types of targets (facilities) and against the various resources that may be present at those targets. These options are expressed by up to 17 data for each target type and for each type of weapons; these 17 data are entered using the EMD card (see page 50) and up to 16 supplementary cards for each type of weapon.

As described more fully in the AIDA report, the basic mathematical representation that is used for the effectiveness of a point-impact weapon is what has been called a cookie-cutter--that is, a uniform probability of kill over a circle of specified radii. Integration of actual weapon effects kill probability contours over the many rectangular targets was rejected in order to limit computer processing. Although the cookie-cutter approach to representing weapon effects has been retained in TSARINA, an attempt has been made to provide the user somewhat greater flexibility for these approximate weapon effects representations. As developed below, there are eight different possibilities provided for estimating losses for each of the different classes of resources, including what might be defined as a two-level cookie-cutter; i.e., an inner circle with a specified Pk, and an outer circle defined such that the average Pk is just one-fourth that for the inner circle.

POINT IMPACT WEAPONS

For Each Target Type Except Aircraft Shelters

The first and tenth data for each target type are the mean radius of effectiveness (R1) against the structures of that target type; these data are entered on the first and tenth of the EMD cards. The first entry applies to near misses and the tenth to

direct hits; if a radius is not entered for direct hits, the first data applies in both cases. The TSARINA results include the fraction of each target that is covered by circles of these radii, one circle for each weapon delivered; the covered area is interpreted as the percent of target damage. Entries on the second and eleventh EMD cards are also effectiveness radii (R2); the interpretation of these radii is dependent upon the assessment criteria in use. The entries on the third through eighth, and the twelfth through seventeenth, EMD cards control damage assessments of the various resources in all facilities. These assessments, for the various resource classes, depend upon damage factors that are entered in the following locations:

<u>EMD Card</u>		
<u>Near Miss</u>	<u>Direct Hit</u>	<u>Resource Class</u>
Third	12th	Personnel
Fourth	13th	AGE and equipment
Fifth	14th	Spare parts
Sixth	15th	Munitions and POL
Seventh	16th	TRAP
Eighth	17th	Building materials

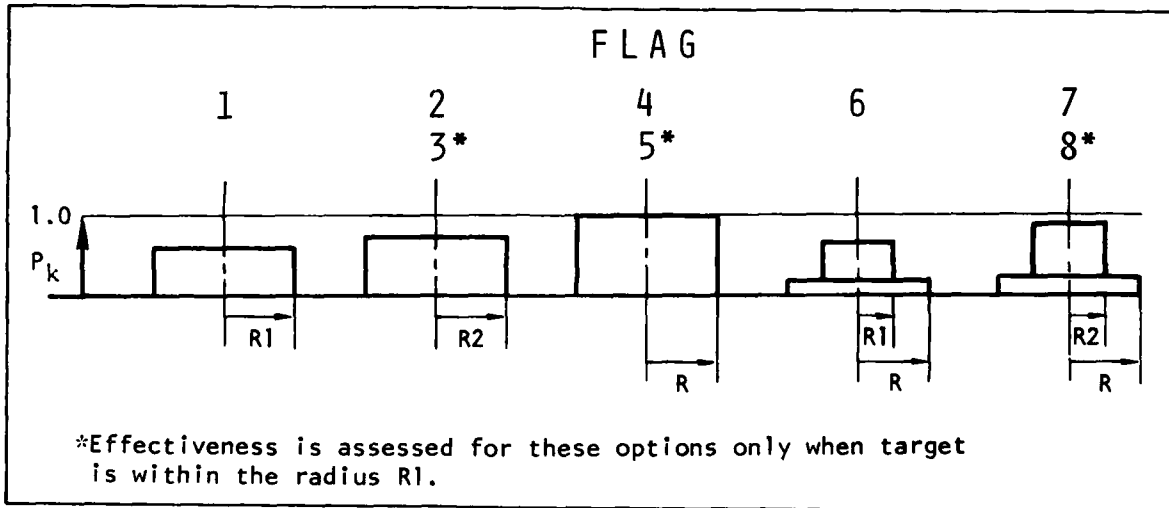
As noted above, the "near miss" values apply when data are not entered for direct hits. If any data are entered for a particular target type on the eleventh through seventeenth cards, only data from those cards will be used in assessing direct hits; i.e., blank entries are interpreted as zero.

The values entered with these 12 resource damage cards are interpreted either (1) as effective kill radii against these resources, (2) as probabilities of kill of such resources within the radii R1 or R2, or (3) as a radius and a kill probability that jointly define a "two-level" cookie-cutter, depending upon the value of the FLAG input described below.

Since munitions and POL would not be expected to be present together, little flexibility is lost with the dual definition used with the sixth (and fifteenth) card; it is important, of course, that the user be clear as to the distinction.

The interpretations of the weapon effectiveness parameters on these twelve cards are specified by the user on the ninth EMD card--i.e., by the value of the variable called "FLAG" in the program.

The user has eight options for handling the vulnerability of each of the six classes of resources, and FLAG communicates these six choices with a 6-digit code. These eight options are illustrated below:



The eight weapon effectiveness parameter options that are identified by the 6-digit code are defined as:

- 0 Ignore this class of resource
- 1 Value represents probability of kill of these resources within a circle of radius R1
- 2 Value represents probability of kill of these resources within a circle of radius R2
- 3 As in 2, given that radius R1 intersects the target perimeter
- 4 Value is the radius of kill of these resources
- 5 Value is the radius of kill of these resources, given that radius R1 intersects the target perimeter

The last three values are a combination of a radius and a probability of kill.

- 6 Value is (1) the radius (times 1000) of an area within which the Pk is one-fourth that value of Pk within R1, plus (2) the value of Pk within R1 (thus 60080, for example, specifies that Pk is 0.80 within R1, and 0.20 in the annular area between R1 and 60 feet).
- 7 Value is the radius (times 1000) of an area within which the Pk is one-fourth that value of Pk within R2, plus the value of Pk within R2
- 8 As in 7, given that radius R1 intersects the target perimeter

Thus FLAG = 321475 would imply that the six classes of resources are to be assessed by options 3, 2, 1, 4, 7, and 5, respectively, for the particular weapon type and target type for which it is listed.

Aircraft Shelters

No distinction is made between a direct hit and a near miss for aircraft shelters; the entry on the first EMD card is interpreted as the effective miss distance against aircraft in shelters with closed doors, and the entry on the tenth card is interpreted as the effective miss distance against aircraft in shelters with open doors. When shelters are handled with the Monte Carlo mode the entries on the second through ninth cards are used to assess losses to resources in a closed shelter in a manner consistent with other target types, as just explained. When the user does not identify specific resources within the aircraft shelters, and they are treated with the expected-value mode, the entries on the third through fifth cards are interpreted as the loss probabilities for personnel, AGE, and spare parts when a weapon strikes within the radius specified on the first card. In either case, the probabilities that aircraft are damaged in closed and open shelters, and the probability that the shelter itself is lost, are given by the entries on the twelfth through fourteenth cards, respectively.

CBU MUNITIONS

The entries on the first EMD card for CBU munitions specify the reliability of the dispenser and the length and width of a rectangular pattern of bomblets; the weapon-type number and the number of cards to be entered for these weapons are also specified. The user may enter up to seven supplementary cards with each EMD card for CBU munitions. The first supplementary card is blank, and the entries on the second through seventh supplementary cards are to be interpreted as the percentage of the six resource classes that would be expected to be lost if the bomblet pattern covered their location. Intermediate cards must not be omitted; e.g., if only equipment and TRAP losses were of interest, two blank supplementary cards would still have to precede the third card and two more precede the sixth card.

V. SAMPLE PROBLEM

The layout of the test base is shown in Figure 1. This base consists of a 200 ft x 7000 ft runway, a parallel taxiway, two connecting taxiways, four shops, three aircraft parking ramps, six aircraft shelters, two vehicle parking areas, and four barracks. To examine the effectiveness of a bombing attack against this target complex with TSARINA, one first must describe the target elements and the attack in a common coordinate system. Each target is defined by its westernmost corner, its size, and its orientation;* the attack heading and the desired mean point of impact fix the attacks. For this illustration four aircraft are assumed to attack each of two points on the runway in an effort to "cut" it; in addition three aircraft attack each of the two sets of three shelters, and three aircraft attack the complex of four shops. Each attacker drops either 18 or 24 bombs with a range error probable (REP) and deflected error probable (DEP) of 300 feet and 150 feet, respectively. The intervalometers are set for stick lengths between 1500 and 2500 feet.

Several different types of resources are present in these facilities and are at risk to the attack. Varying quantities of personnel and AGE are in the four shops. In addition, several types of AGE are in the vehicle parking areas. Many off-duty personnel are in the barracks. Several of the aircraft shelters have TRAP and munitions stored internally.

INPUT

Figures 2 and 3 reproduce the card images needed to describe this sample problem and to control the assessment. For clarity, the control (CONT) card, the TSAR data (DATA) card, the target (TGT) cards, and the attack (ATT) cards are listed in order; the

*An auxiliary program is available for converting dimension data prepared for the MASSIVE program (developed at Eglin AFB) into the format required for TSARINA.

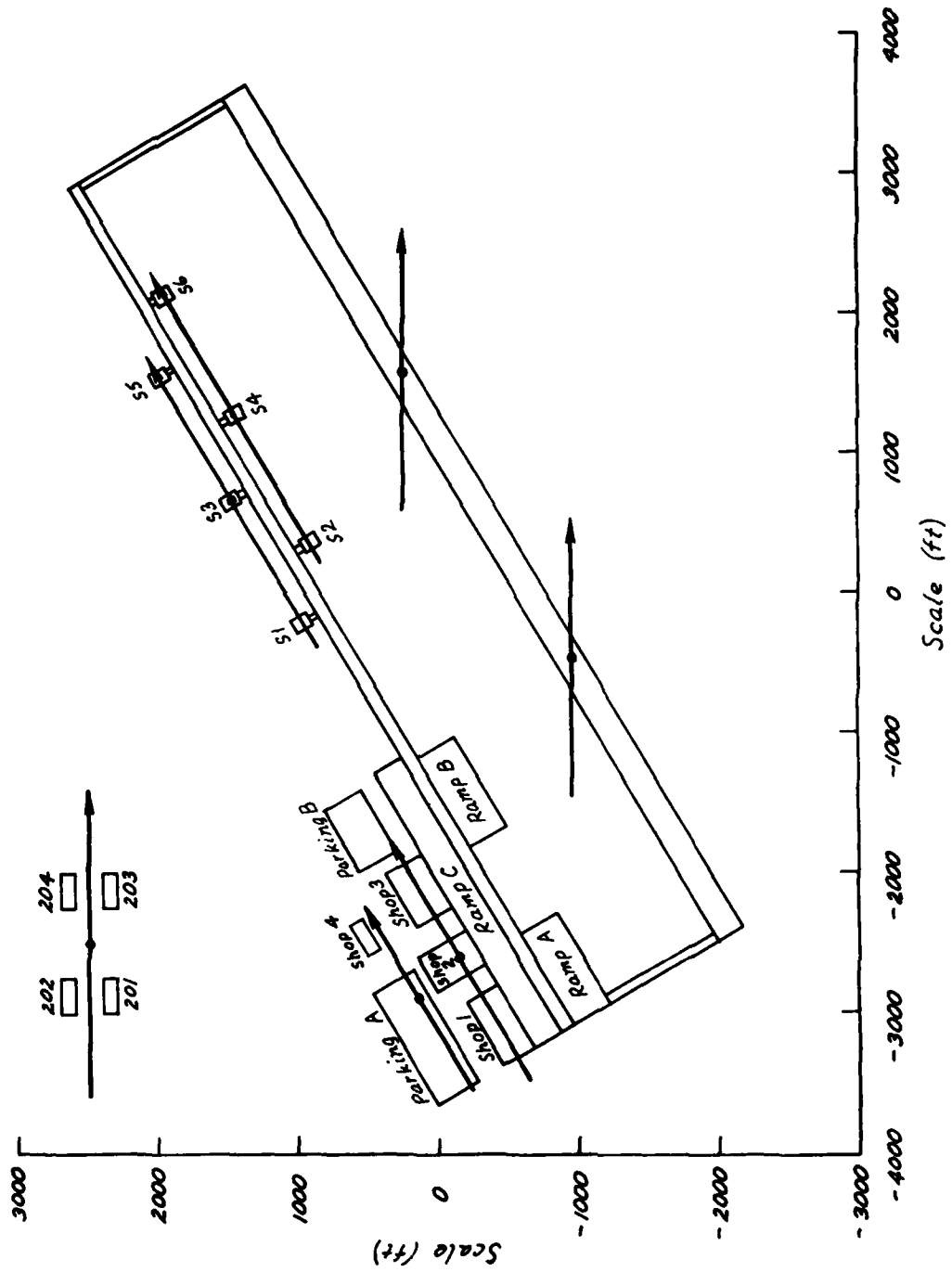


Fig. 1 — Test base

Model Operations are Specified with the CONTROL card

CONT	1	0	8	10	1	1	3	0	4250	65	1	0	10	250	TESTBASE
------	---	---	---	----	---	---	---	---	------	----	---	---	----	-----	----------

The DATA card controls the interactions with TSAR.

DATA	0	0	5	1	5	45	2	3	4	50
------	---	---	---	---	---	----	---	---	---	----

The target data for these sample calculations are entered next.

TGT	-2500	-2000	7000	200	60		1								RUNWAY MAINTENANCE
TGT	-3150	-880	7000	80	60		1								
TGT	-3380	-450	500	275	60		5			1				2	SHOP #1
	C 1	1	20 C 1	4	33 C	2		2		10					
	C 1	2	18 C 1	3	35										
TGT	-2860	0	275	450	60		6			2				1	SHOP #2
	C 1	2	30 C 2	3	25 C	2		4		50 C	1		1	35	
TGT	-2390	150	425	300	60		5			3				1	SHOP #3
	C 1	1	18 C 1	2	22 C	1		3		40					
TGT	-2600	500	250	100	60		6			4			1	7	SHOP #4
TGT	-3100	-960	750	300	60		4								RAMP A
TGT	-1810	-240	750	300	60		4								RAMP B
TGT	-3240	-700	2250	200	60		4								RAMP C
TGT	-3650	0	900	300	60		7							1	PKG AREA
	C 2	1	80 C 2	2	30 C	2		3		25 C	2		4	10	
TGT	-2000	550	500	275	60		7							1	PKG AREA
	C 2	1	20 C 2	2	50 C	2		3		50 C	2		4	40	
TGT	-2950	-1240	50	850	60		3								X TAXY 1
TGT	2940	2500	50	850	60		3								X TAXY 2
TGT	-200	930	50	50	60		3								STUB 1
TGT	310	1010	50	50	60		3								STUB 2
TGT	670	1430	50	50	60		3								STUB 3
TGT	1200	1530	50	50	60		3								STUB 4
TGT	1550	1960	50	50	60		3								STUB 5
TGT	2070	2050	50	50	60		3								STUB 6

The aircraft shelter data are entered with TGT type cards, rather than TGT2 type cards, since munitions and TRAP are stored in these facilities in this example.

[illegible]

Various off-duty personnel are in the barracks.

TGT	-3000	2300	100	250	0	8	1	BARRACKS #201
C	1	1000	20	1000	44			
TGT	-3000	2600	100	250	0	8	1	BARRACKS #202
C	1	1000	20	1000	22			
TGT	-2250	2300	250	250	0	8	1	BARRACKS #203
C	1	1000	20	1000	34			
TGT	-2250	2600	100	250	0	8	1	BARRACKS #204
C	1	1000	20					

1 2 3 4 5 6 7 8

Fig. 2 — TSARINA input-control data and target information

The attack data, equivalence data and the weapon effectiveness data are entered last.

ATT	3	90	-450	-950	300	150	75	55	24	2000	1	95
ATT	3	90	1600	250	300	150	75	55	24	2000	1	95
ATT	2	60	-2600	-150	300	150	75	55	24	2000	1	95
ATT	1	60	-2900	150	300	150	75	55	24	1500	1	95
ATT	4	60	650	1550	300	150	75	55	24	2500	1	95
ATT	4	60	1310	1400	300	150	75	55	24	2500	1	95
ATT	2	90	-2500	2500	300	150	75	55	24	2200	1	95

The equivalence data are used to assure compatibility of resource types at the ISAR-TSARINA interface.

EQUI	5001	1	-1	5001	4	4	5	5004	1	1	101
EQUI	4	104	5005	2	2	1	7				
EQUI	5001	1003	1003	1004	1005	5001	2	101	0		

Fourteen cards are used to input the effectiveness data; direct hits and near hits are assessed with the same effectiveness factors, except for the eighth target type.

END 14	1	95	20	2	20	20	80	100	20	120
				150		80	110	140	40	35
				75			85	75	75	80
				55			28	15	15	70
				35	100	80	35	24	26	60
				40			30	17	30	
				45			44	29	44	
							11	0	11	
			3333		7	111545123120222545777				
			100							150
										43
				55					100	80
				69					90	80
				12					80	80

END

1	2	3	4	5	6	7	8
12345678901234567890123456789012345678901234567890123456789012345678901234567890							

Fig. 3 — TSARINA input - attacks and weapons data

ordering of cards for a given case is generally immaterial except that (1) the CONTROL and DATA cards should be entered first, (2) all supplementary cards (used with the TGT, ATT2, and EMD cards) must be listed immediately following the card that they supplement, and (3) the EQUI cards should be entered as a group. The END card concludes the input stream and specifies the end of the assessment; alternatively, a REDO card can be entered to end the input for one assessment and call for a new case in which the attack is changed, or both the target complex and the attack are changed. In this sample we have not used either the TGT2 or ATT2 type cards; the TGT2 card types could have been used for the aircraft shelters if resources had not been stored in these facilities.

A careful review of these entries along with a reading of Appendix A should lead to a full and rapid understanding of the various input requirements. The CONTROL card data specify, for example, that ten trials are to be computed, and the runway availability is to be based on a 65 feet x 4250 feet minimum operating surface. Required repairs are to be assessed (the 1 in col. 45), but plots of the impact points are not to be generated (the 0 in col. 48). The DATA card specifies that TSAR is to simulate this attack at 5:45 am on the first day at base #1. Aircraft shelters, taxiways, and aircraft parking ramps are to be designated, respectively, as target types #2, #3, and #4.

As will be noted, the shop number has been entered in columns 53-54 of the TGT cards for the four shops, and the number of cards that are to be used to specify the resources that are associated with each target is specified in columns 68-70, except for shop #4. Since 100 percent of the on-duty type #7 personnel are in shop #4, columns 55-66 on the TGT card can be used to locate this resource. Shop #1 contains 20 percent of the type #1 personnel, 33 percent of the type #4 personnel, and 10 percent of the type #2 AGE. The larger vehicle parking area contains 80 percent of the type #1 AGE, and 30, 25, and 10 percent of AGE types #2, #3, and #4, respectively. The target data for the aircraft shelters and barracks are listed next; 0.7

percent of the type #1 munitions and 2.8 percent of the type #2 TRAP are in aircraft shelter #1; other shelters contain other quantities of munitions and TRAP. Eighty percent of the off-duty personnel are distributed in the four barracks, except for off-duty type #3 personnel who are located in the first three barracks.

The ATT cards in Figure 3 specify the attack headings, intended aim points, aiming errors, ballistic dispersion, number of weapons, bomb stick length, weapon type, and probability of arrival of the attacks. The EQUIvalence cards enter instructions for structuring the results to be transmitted to TSAR. The first entry specifies that type #1 personnel casualties are not to be reported to TSAR. The second entry prescribes that the percentage casualties to TSARINA type #4 personnel are to be imposed on types #4 and #5 personnel in TSAR. The next entry specifies that the percentage losses sustained by type #1 munitions are to be imposed on both assembled and unassembled #1 and #4 type munitions in TSAR; note that the "equivalence" list starts on the first EQUI card and is completed on the second. The last entry on the second card specifies that the casualty percentage sustained in TSARINA by the #2 type TRAP is to be imposed on three TSAR types of TRAP: #1, #2, and #7. The third EQUI card specifies that the casualties sustained by the off-duty type #3 personnel should be applied to the #3, #4, and #5 off-duty personnel types in TSAR. (Casualties among other off-duty types will be controlled by the losses to the type #1000 personnel who were located in the barracks.) The last entry on the third EQUI card specifies that the percentage loss sustained by personnel type #2 should be imposed on the aircrews (i.e., #101) at risk in TSAR.

The various weapon effectiveness data are entered with the EMD card and its supplementary cards. The first entry specifies that 14 cards are to be used to specify the weapon effects for this weapon. The second and third entries denote that the weapon is type #1 and that its reliability is 95 percent. The eight entries that follow are the radii of weapon effects against the eight target types found at the test base. The first of these specifies an effective disrupted radius of 20 feet on runways (target type #1).

The same value is indicated for the other pavement targets: e.g., the taxiways, aircraft parking ramps, and vehicle parking areas (target types #3, #4, and #7). Larger radii are specified for structural effects against the buildings (target types #5, #6, and #8).

The entries on the first supplementary EMD card specify the secondary weapon effects radii. The role and interpretation of these radii, and of the factors on the following six cards, are determined by the entries on the eighth supplementary card, as explained at length in Appendix A. For example, the values listed for the sixth type of target specify that personnel and munitions (or POL) that are within a radius of 100 feet, sustain 75 and 17 percent losses, respectively; i.e., their losses are governed by the first criterion. AGE and TRAP (the third and sixth supplementary cards) are governed by the second criterion, and 15 and 29 percent of these resources within 140 feet of a hit are lost. All spare parts within 140 feet of the burst sustain 24 percent losses, if the burst was within 100 feet of the target boundary (i.e., the third criterion). No assessment is provided for building materials in the sixth type of target.

The last three entries for the second target type--the aircraft shelters--are the three special factors that control aircraft and shelter damage estimates. The first two numbers are the probabilities that aircraft that are in the shelters at the time of the attack will be killed (1) if the shelter is buttoned up and a weapon strikes within the two foot radius entered on the first card, and (2) if the shelter door is open and a weapon strikes within the 100 foot radius noted on the tenth card. The last of these three entries specifies that there is a 12 percent chance that the shelter itself is destroyed, given a hit within two feet (R1) of the shelter walls.

The two-level cookie-cutter damage function is used for target type #8; when weapons fall outside the target, 80 percent of the personnel, equipment, and parts within the target are lost within 35 feet of an impact and 20 percent (one-fourth of 80) of these resources are lost within 80, 70, and 60 feet from an impact, respectively.

If the weapon hits the target, 80 percent of the resources are lost within 43 feet of an impact, and 20 percent are lost within 100, 90, and 80 feet, respectively.

OUTPUT

The initial TSARINA output provides a record of the input data. The first of these data, shown in Figure 4, provides a record of all resources, including the number and name of the target at which the resources are stored. These are followed by the formatted title block shown in Figure 5 that indicates the values for several of the key control parameters. The scheduled time of the attack, and an indication of any coordinate translation that was required, are indicated next. Full particulars on the targets, attacks, and weapon effects data conclude the input data as shown in Figures 5 and 6. To distinguish aircraft shelters from other numbered facilities, a "1000" is added to the number for each shelter to avoid ambiguity.

TSARINA output for each trial is illustrated in Figures 7 and 8, using the results for the eighth trial. As will be noted, both the runway and the main taxiway were hit; of the 26 bombs that affected the runway, six did not impact the runway itself, but hit close enough for the runway to be within the bomb's 20 foot radius of effectiveness. Two of the six aircraft shelters each received one hit; in one case the impact was outside the target, but within the 2-foot radius of effectiveness. The fractional losses of any resources that were stored in each of these facilities are also noted.

The only hits that affect the taxiways, or the stub taxiways in front of the aircraft shelters, were on those stubs; stub #1 sustained a direct hit, and stub #5 sustained a near miss.

Three of the shops sustained several hits; the expected losses to the various classes of resources can be noted. For shop #2, for example, even though 5 out of the 9 hits were external to the facility, damage would be expected to nearly 70 percent of the structure, and 58 percent of the personnel in that facility are estimated to be lost.

RESOURCE STORAGE DATA										
TARGET NUMBER	NAME									
3	SHOP #1		-3380	-450						
		C 1	1 200	C 1	4 330	C 2	2 100			
4	SHOP #2	C 1	2 180	C 1	3 350					
			-2860	0						
5	SHOP #3	C 1	2 300	C 2	3 250	C 2	4 500	C 1	1 350	
			-2390	150						
6	SHOP #4	C 1	1 180	C 1	2 220	C 1	3 400			
			-2600	500						
10	PKG AREA	C 1	7 1000							
			-3650	0						
11	PKG AREA	C 2	1 800	C 2	2 300	C 2	3 250	C 2	4 100	
			-2000	550						
20	SHEL #1	C 2	1 200	C 2	2 500	C 2	3 500	C 2	4 400	
			-330	1070						
21	SHEL #2	C 4	1 7	C 5	2 24					
			310	920						
23	SHEL #4	C 4	3 31	C 5	4 43					
			1200	1435						
24	SHEL #5	C 4	3 31	C 5	4 43					
			1400	2040						
25	SHEL #5	C 4	1 7	C 5	2 24	C 4	2 4	C 5	3 10	
			2090	1950						
26	BARRACKS #201	C 4	2 24	C 5	3 12					
			-3000	2300						
27	BARRACKS #202	C 1	1000 200	C 1	1000 440					
			-3000	2400						
28	BARRACKS #203	C 1	1000 200	C 1	1000 120					
			-2250	2300						
29	BARRACKS #204	C 1	1000 200	C 1	1000 340					
			-2250	2600						

STORAGE OF THE RESOURCE LOCATIONS REQUIRED 55 LOCATIONS IN THE STOCKS ARRAY,
AND THE EQUIVALENCE DATA USED 21 ELEMENTS OF THE EQUIV ARRAY.

Fig. 4 — Listing of resource location data

```

*****
***** ISARINA *** DAMAGE ASSESSMENTS FOR ISAR *****
*****
      BASED ON THE AIDA MODEL OF AIRBASE DAMAGE ASSESSMENT
      DEVELOPED BY D. E. EMERSON AT THE RAND CORPORATION
*****
NO OF TRIALS TO MERIT 3  DAMAGE 3  MODE C  NCL 4250(250)  NCM 65(10)  MIN REPAIR 1  PLOT HITS 3  TEST 3
*****

      CASE # 1
      IN THE ISAR SIMULATION, THIS ATTACK AND DAMAGE OCCUR AT BASE # 1 ON DAY 1 AT 5:45

      ***** BASE COMPLEX NAME - TESTBASE *****

      ALL TARGET LOCATION DIMENSIONS WERE INCREASED BY 4000 IN THE X-DIMENSION AND 2000 IN THE Y-DIMENSION

      NUMBER      X-DIM      Y-DIM      TARGET DATA      ANGLE      TGT TYPE      STORE      BLDG NO
      1          1500          0          7500      230      60      1      0      0      0      0      0      0      0      0      0      0      0      0      0
      2          850          1120      7500      80      60      1      0      0      0      0      0      0      0      0      0      0      0      0      0
      20          3670          3370          75      125      60      2      0      0      0      0      0      0      0      0      0      0      0      0      0
      21          4310          2920          75      125      60      2      0      0      0      0      0      0      0      0      0      0      0      0      0
      22          4530          3540          75      125      60      2      0      0      0      0      0      0      0      0      0      0      0      0      0
      23          5200          3435          75      125      60      2      0      0      0      0      0      0      0      0      0      0      0      0      0
      24          5400          4330          75      125      60      2      0      0      0      0      0      0      0      0      0      0      0      0      0
      25          6090          3950          75      125      60      2      0      0      0      0      0      0      0      0      0      0      0      0      0
      12          1050          760          50      450      60      3      0      0      0      0      0      0      0      0      0      0      0      0
      13          840          4530          50      450      60      3      0      0      0      0      0      0      0      0      0      0      0      0
      14          380          2910          50      450      60      3      0      0      0      0      0      0      0      0      0      0      0      0
      15          4310          3710          50      50      60      3      0      0      0      0      0      0      0      0      0      0      0      0
      16          4670          3430          50      50      60      3      0      0      0      0      0      0      0      0      0      0      0      0
      17          5200          3530          50      50      60      3      0      0      0      0      0      0      0      0      0      0      0      0
      18          5550          3660          50      50      60      3      0      0      0      0      0      0      0      0      0      0      0      0
      19          6070          4050          50      50      60      3      0      0      0      0      0      0      0      0      0      0      0      0
      7           920          1240          750      300      60      4      0      0      0      0      0      0      0      0      0      0      0      0
      8          4190          1760          750      300      60      4      0      0      0      0      0      0      0      0      0      0      0      0
      9          760          1300          2250      200      60      4      0      0      0      0      0      0      0      0      0      0      0      0
      3           620          1550          500      275      60      5      0      0      0      0      0      0      0      0      0      0      0      0
      5          1610          2150          425      300      60      5      0      0      0      0      0      0      0      0      0      0      0      0
      4          1140          2600          275      450      60      6      0      0      0      0      0      0      0      0      0      0      0      0
      6          1400          2500          250      100      60      6      0      0      0      0      0      0      0      0      0      0      0      0
      10          350          2300          900      300      60      7      0      0      0      0      0      0      0      0      0      0      0      0
      11          200          2550          500      275      60      7      0      0      0      0      0      0      0      0      0      0      0      0
      26          1000          4300          100      250      0      8      0      0      0      0      0      0      0      0      0      0      0      0
      27          1000          4600          100      250      0      8      0      0      0      0      0      0      0      0      0      0      0      0
      28          1750          4300          250      250      0      8      0      0      0      0      0      0      0      0      0      0      0      0
      29          1750          4600          100      250      0      8      0      0      0      0      0      0      0      0      0      0      0      0

```

Fig. 5 — Listing of control and target data

NUMBER	HOG	ATTACK DATA					R-DISP	D-DISP	NO WPNS	LENGTH	WPN TYPE	APRVAL
		X-DHPI	Y-DHPI	REP	DEP							
1	90	3550	1350	300	150	75	55	24	2000	1	95	
2	90	3550	1350	300	150	75	55	24	2000	1	95	
3	90	3550	1350	300	150	75	55	24	2000	1	95	
4	90	5600	2250	300	150	75	55	24	2000	1	95	
5	90	5600	2250	300	150	75	55	24	2000	1	95	
6	90	5600	2250	300	150	75	55	24	2000	1	95	
7	60	1400	1850	300	150	75	55	18	2000	1	95	
8	60	1400	1850	300	150	75	55	18	2000	1	95	
9	60	1100	2150	300	150	75	55	18	1500	1	95	
10	60	4650	3550	300	150	75	55	18	2500	1	95	
11	60	4650	3550	300	150	75	55	18	2500	1	95	
12	60	4650	3550	300	150	75	55	18	2500	1	95	
13	60	4650	3550	300	150	75	55	18	2500	1	95	
14	60	5310	3400	300	150	75	55	18	2500	1	95	
15	60	5310	3400	300	150	75	55	18	2500	1	95	
16	60	5310	3400	300	150	75	55	18	2500	1	95	
17	60	5310	3400	300	150	75	55	18	2500	1	95	
18	90	1500	4500	300	150	75	55	18	2200	1	95	
19	90	1500	4500	300	150	75	55	18	2200	1	95	

MISS DISTANCE AND KILL PROBABILITY DATA

WPN TYPE	WPN REL	TARGET TYPES									
		1	2	3	4	5	6	7	8	9	10
1	0.950	20	2	20	20	80	100	20	120	0	0
		0	150	0	40	110	140	40	45	0	0
		0	75	0	0	45	75	75	80000	0	0
		0	55	0	0	28	15	15	70000	0	0
		0	35	0	100000	35	24	26	60000	0	0
		0	40	0	0	30	17	30	0	0	0
		0	45	0	0	44	24	44	0	0	0
		0	0	0	0	11	0	11	0	0	0
		0	33330	0	7000	111545	123120	222545	777300	0	0
		-1	100	-1	-1	-1	-1	-1	150	-1	-1
		0	0	0	0	0	0	0	40	0	0
		0	55	0	0	0	0	0	100000	0	0
		0	64	0	0	0	0	0	80000	0	0
		0	12	0	0	0	0	0	40000	0	0

Fig. 6 — Listing of attack and weapons data

***** TRIAL # 8 *****													
TARGET HIT SUMMARY				TRIAL				B					
TGT NO	NUMBER HITS	CPH COVERAGE	TOT OUT	COVERAGE	R1	P2	PEOPLE	AGE	PARTS	AMMO	TRAP	MATERL	RLOG NO
1	26	4	1.3	** TARGET TYPE #	1	**	3.3	3.3	3.3	0.3	0.3	0.3	0
2	2	1	0.3	** TARGET TYPE #	2	**	0.0	0.3	3.3	0.3	0.3	0.3	0
20	1	3	3.3	** TARGET TYPE #	1	**	1.003	3.753	0.350	0.400	0.450	0.0	1001
21	3	3	3.3	** TARGET TYPE #	2	**	1.003	0.3	0.3	0.3	0.3	0.3	1002
22	3	3	3.3	** TARGET TYPE #	3	**	0.956	3.3	0.3	0.3	0.3	0.3	1003
23	1	1	3.0	** TARGET TYPE #	1	**	1.003	3.753	0.350	0.400	0.450	0.0	1004
24	3	3	3.3	** TARGET TYPE #	2	**	0.553	3.3	0.3	0.3	0.3	0.3	1005
25	3	3	3.3	** TARGET TYPE #	3	**	0.818	0.3	3.3	0.3	0.3	0.3	1006
14	1	3	3.3	** TARGET TYPE #	1	**	3.443	3.3	3.3	0.3	0.3	0.3	0
18	1	1	3.3	** TARGET TYPE #	2	**	3.389	3.3	3.3	0.3	0.3	0.3	0
9	5	1	3.3	** TARGET TYPE #	1	**	3.322	3.3	3.356	0.3	0.3	0.3	0
3	3	3	3.3	** TARGET TYPE #	2	**	3.523	3.461	3.179	3.399	3.197	0.014	1
5	3	3	3.3	** TARGET TYPE #	3	**	0.336	3.3	3.3	0.3	0.3	0.3	3
4	9	5	3.3	** TARGET TYPE #	1	**	3.696	3.583	3.266	3.394	3.457	0.3	2
6	4	3	3.3	** TARGET TYPE #	2	**	3.096	3.784	3.334	3.241	3.567	0.3	4
13	4	1	3.3	** TARGET TYPE #	3	**	3.127	3.397	3.320	3.335	3.153	0.010	0
26	5	4	3.3	** TARGET TYPE #	1	**	3.835	3.143	3.171	3.156	3.3	0.3	0
27	3	3	3.3	** TARGET TYPE #	2	**	3.918	3.246	3.226	3.204	3.3	0.3	0
28	3	3	3.3	** TARGET TYPE #	3	**	3.482	3.381	3.073	3.066	3.3	0.3	0
29	4	4	3.3	** TARGET TYPE #	1	**	3.838	3.143	3.115	3.391	3.3	0.3	0
				** TARGET TYPE #	2	**							
				** TARGET TYPE #	3	**							
				** TARGET TYPE #	4	**							
				** TARGET TYPE #	5	**							
				** TARGET TYPE #	6	**							
				** TARGET TYPE #	7	**							
				** TARGET TYPE #	8	**							
				** TARGET TYPE #	9	**							
				** TARGET TYPE #	10	**							
				** TARGET TYPE #	11	**							
				** TARGET TYPE #	12	**							
				** TARGET TYPE #	13	**							
				** TARGET TYPE #	14	**							
				** TARGET TYPE #	15	**							
				** TARGET TYPE #	16	**							
				** TARGET TYPE #	17	**							
				** TARGET TYPE #	18	**							
				** TARGET TYPE #	19	**							
				** TARGET TYPE #	20	**							
				** TARGET TYPE #	21	**							
				** TARGET TYPE #	22	**							
				** TARGET TYPE #	23	**							
				** TARGET TYPE #	24	**							
				** TARGET TYPE #	25	**							
				** TARGET TYPE #	26	**							
				** TARGET TYPE #	27	**							
				** TARGET TYPE #	28	**							
				** TARGET TYPE #	29	**							
				** TARGET TYPE #	30	**							
				** TARGET TYPE #	31	**							
				** TARGET TYPE #	32	**							
				** TARGET TYPE #	33	**							
				** TARGET TYPE #	34	**							
				** TARGET TYPE #	35	**							
				** TARGET TYPE #	36	**							
				** TARGET TYPE #	37	**							
				** TARGET TYPE #	38	**							
				** TARGET TYPE #	39	**							
				** TARGET TYPE #	40	**							
				** TARGET TYPE #	41	**							
				** TARGET TYPE #	42	**							
				** TARGET TYPE #	43	**							
				** TARGET TYPE #	44	**							
				** TARGET TYPE #	45	**							
				** TARGET TYPE #	46	**							
				** TARGET TYPE #	47	**							
				** TARGET TYPE #	48	**							
				** TARGET TYPE #	49	**							
				** TARGET TYPE #	50	**							
				** TARGET TYPE #	51	**							
				** TARGET TYPE #	52	**							
				** TARGET TYPE #	53	**							
				** TARGET TYPE #	54	**							
				** TARGET TYPE #	55	**							
				** TARGET TYPE #	56	**							
				** TARGET TYPE #	57	**							
				** TARGET TYPE #	58	**							
				** TARGET TYPE #	59	**							
				** TARGET TYPE #	60	**							
				** TARGET TYPE #	61	**							
				** TARGET TYPE #	62	**							
				** TARGET TYPE #	63	**							
				** TARGET TYPE #	64	**							
				** TARGET TYPE #	65	**							
				** TARGET TYPE #	66	**							
				** TARGET TYPE #	67	**							
				** TARGET TYPE #	68	**							
				** TARGET TYPE #	69	**							
				** TARGET TYPE #	70	**							
				** TARGET TYPE #	71	**							
				** TARGET TYPE #	72	**							
				** TARGET TYPE #	73	**							
				** TARGET TYPE #	74	**							
				** TARGET TYPE #	75	**							
				** TARGET TYPE #	76	**							
				** TARGET TYPE #	77	**							
				** TARGET TYPE #	78	**							
				** TARGET TYPE #	79	**							
				** TARGET TYPE #	80	**							
				** TARGET TYPE #	81	**							
				** TARGET TYPE #	82	**							
				** TARGET TYPE #	83	**							
				** TARGET TYPE #	84	**							
				** TARGET TYPE #	85	**							
				** TARGET TYPE #	86	**							
				** TARGET TYPE #	87	**							
				** TARGET TYPE #	88	**							
				** TARGET TYPE #	89	**							
				** TARGET TYPE #	90	**							
				** TARGET TYPE #	91	**							
				** TARGET TYPE #	92	**							
				** TARGET TYPE #	93	**							
				** TARGET TYPE #	94	**							
				** TARGET TYPE #	95	**							
				** TARGET TYPE #	96	**							
				** TARGET TYPE #	97	**							
				** TARGET TYPE #	98	**							
				** TARGET TYPE #	99	**							
				** TARGET TYPE #	100	**							

Fig. 7 — TSARINA trial results

Fig. 8 — Trial results of fractional losses

The fractional losses for all of the resource types present on the base are summarized in Figure 8; these results cumulate the fractional losses suffered by each type of resource at each of its locations. These same results, modified as required by the EQUI cards, and summary data describing the damage sustained by the runway, taxiways, aircraft shelters, and other facilities, are shown at the bottom of Figure 8, formatted for transfer to TSAR. To interpret these card images, the reader should consult the instructions for preparation of Input Card Type #40 in the TSAR User's Manual.

The statistical results for the ten trials are presented in Figures 9 and 10. The first of these figures provides an indication of the fraction of trials in which at least one hit was sustained, as well as the expected number, and standard deviation, of hits. Comparable results are provided for CBU weapons, when they have been used. The results labelled "Bomb Coverage" are the expected fractions of the facility floor-space that are affected by the two coverage radii, R1 and R2. The average losses that are sustained at each target by the six classes of resources are listed on the right side of Figure 9. A summary of the runway closures and the required runway repairs is noted at the bottom of this figure.

Figure 10 presents the average losses sustained by each type of resource for the ten trials, along with the standard deviation of those losses. When TSARINA is used as a general-purpose damage assessment model, these statistical results are the primary output; they are not transferred to TSAR, since TSAR only uses the trial-by-trial results illustrated in Figure 9.

TARGET DAMAGE STATISTICS FOR 1) TOTALS												
TARGET NO	PERCENT ATTACKS HIT	AVERAGE HITS PER ATTACK	STD DEV OF HITS	AVG CPU COVERAGE	STD DEV CPU COVERAGE	HMMR COVERAGE	P2	PEOPLE	AGE	KILL PRIORITIES	AMMO	TREAD MATERIAL
TARGET TYPE # 1												
1	100.0	20.80	6.68	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	50.0	21.60	12.55	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
TARGET TYPE # 2												
20	50.0	3.50	0.53	0.0	0.0	0.083	0.375	0.275	0.175	0.200	0.225	0.0
21	60.0	3.40	0.52	0.0	0.0	0.890	0.300	0.220	0.140	0.160	0.180	0.0
22	30.0	3.30	0.48	0.0	0.0	0.986	0.225	0.165	0.105	0.120	0.135	0.0
23	60.0	3.40	0.52	0.0	0.0	0.930	0.300	0.220	0.160	0.160	0.180	0.0
24	20.0	3.20	0.42	0.0	0.0	0.952	0.150	0.110	0.070	0.080	0.090	0.0
25	60.0	0.60	0.52	0.0	0.0	0.901	0.300	0.220	0.140	0.160	0.180	0.0
		46.67	3.37									
TARGET TYPE # 3												
13	20.0	3.20	0.42	0.0	0.0	0.705	0.0	0.0	0.0	0.0	0.0	0.0
14	20.0	0.40	0.97	0.0	0.0	0.980	0.0	0.0	0.0	0.0	0.0	0.0
15	10.0	0.10	0.32	0.0	0.0	0.931	0.0	0.0	0.0	0.0	0.0	0.0
16	30.0	0.30	0.68	0.0	0.0	0.998	0.0	0.0	0.0	0.0	0.0	0.0
17	20.0	0.20	0.42	0.0	0.0	0.916	0.0	0.0	0.0	0.0	0.0	0.0
18	10.0	0.10	0.32	0.0	0.0	0.936	0.0	0.0	0.0	0.0	0.0	0.0
19	60.0	0.50	0.71	0.0	0.0	0.972	0.0	0.0	0.0	0.0	0.0	0.0
TARGET TYPE # 4												
8	0.0	0.0	0.0	0.0	0.0	0.005	0.0	0.0	0.0	0.0	0.0	0.0
9	70.0	0.10	0.75	0.0	0.0	0.916	0.0	0.0	0.0	0.0	0.0	0.0
TARGET TYPE # 5												
3	100.0	7.10	2.18	0.0	0.0	0.422	0.377	0.160	0.188	0.085	0.168	0.012
5	50.0	7.20	5.47	0.0	0.0	0.636	0.576	0.360	0.186	0.208	0.166	0.012
TARGET TYPE # 6												
4	100.0	4.70	2.56	0.0	0.0	0.437	0.345	0.515	0.265	0.181	0.432	0.0
6	0.0	1.50	1.66	0.0	0.0	0.331	0.467	0.266	0.178	0.074	0.223	0.0
TARGET TYPE # 7												
10	70.0	4.70	4.27	0.0	0.0	0.390	0.376	0.058	0.012	0.044	0.090	0.006
11	70.0	1.80	1.69	0.0	0.0	0.315	0.355	0.382	0.316	0.032	0.066	0.005
TARGET TYPE # 1												
26	90.0	3.40	2.30	0.0	0.0	0.881	0.126	0.171	0.158	0.166	0.0	0.0
27	100.0	3.20	1.38	0.0	0.0	0.998	0.046	0.145	0.118	0.0	0.0	0.0
28	40.0	4.50	2.80	0.0	0.0	0.852	0.308	0.194	0.118	0.104	0.0	0.0
29	100.0	3.70	1.70	0.0	0.0	0.931	0.166	0.216	0.198	0.172	0.0	0.0
TARGET TYPE # 2												
30	100.0	4.70	4.27	0.0	0.0	0.390	0.376	0.058	0.012	0.044	0.090	0.006
31	70.0	1.80	1.69	0.0	0.0	0.315	0.355	0.382	0.316	0.032	0.066	0.005
TARGET TYPE # 3												
32	90.0	3.40	2.30	0.0	0.0	0.881	0.126	0.171	0.158	0.166	0.0	0.0
33	100.0	3.20	1.38	0.0	0.0	0.998	0.046	0.145	0.118	0.0	0.0	0.0
34	40.0	4.50	2.80	0.0	0.0	0.852	0.308	0.194	0.118	0.104	0.0	0.0
35	100.0	3.70	1.70	0.0	0.0	0.931	0.166	0.216	0.198	0.172	0.0	0.0
TARGET TYPE # 4												
36	100.0	4.70	4.27	0.0	0.0	0.390	0.376	0.058	0.012	0.044	0.090	0.006
37	70.0	1.80	1.69	0.0	0.0	0.315	0.355	0.382	0.316	0.032	0.066	0.005
TARGET TYPE # 5												
38	90.0	3.40	2.30	0.0	0.0	0.881	0.126	0.171	0.158	0.166	0.0	0.0
39	100.0	3.20	1.38	0.0	0.0	0.998	0.046	0.145	0.118	0.0	0.0	0.0
40	40.0	4.50	2.80	0.0	0.0	0.852	0.308	0.194	0.118	0.104	0.0	0.0
41	100.0	3.70	1.70	0.0	0.0	0.931	0.166	0.216	0.198	0.172	0.0	0.0
TARGET TYPE # 6												
42	100.0	4.70	4.27	0.0	0.0	0.390	0.376	0.058	0.012	0.044	0.090	0.006
43	70.0	1.80	1.69	0.0	0.0	0.315	0.355	0.382	0.316	0.032	0.066	0.005
TARGET TYPE # 7												
44	90.0	3.40	2.30	0.0	0.0	0.881	0.126	0.171	0.158	0.166	0.0	0.0
45	100.0	3.20	1.38	0.0	0.0	0.998	0.046	0.145	0.118	0.0	0.0	0.0
46	40.0	4.50	2.80	0.0	0.0	0.852	0.308	0.194	0.118	0.104	0.0	0.0
47	100.0	3.70	1.70	0.0	0.0	0.931	0.166	0.216	0.198	0.172	0.0	0.0
TARGET TYPE # 8												
48	100.0	4.70	4.27	0.0	0.0	0.390	0.376	0.058	0.012	0.044	0.090	0.006
49	70.0	1.80	1.69	0.0	0.0	0.315	0.355	0.382	0.316	0.032	0.066	0.005
TARGET TYPE # 9												
50	90.0	3.40	2.30	0.0	0.0	0.881	0.126	0.171	0.158	0.166	0.0	0.0
51	100.0	3.20	1.38	0.0	0.0	0.998	0.046	0.145	0.118	0.0	0.0	0.0
52	40.0	4.50	2.80	0.0	0.0	0.852	0.308	0.194	0.118	0.104	0.0	0.0
53	100.0	3.70	1.70	0.0	0.0	0.931	0.166	0.216	0.198	0.172	0.0	0.0
TARGET TYPE # 10												
54	100.0	4.70	4.27	0.0	0.0	0.390	0.376	0.058	0.012	0.044	0.090	0.006
55	70.0	1.80	1.69	0.0	0.0	0.315	0.355	0.382	0.316	0.032	0.066	0.005
TARGET TYPE # 11												
56	90.0	3.40	2.30	0.0	0.0	0.881	0.126	0.171	0.158	0.166	0.0	0.0
57	100.0	3.20	1.38	0.0	0.0	0.998	0.046	0.145	0.118	0.0	0.0	0.0
58	40.0	4.50	2.80	0.0	0.0	0.852	0.308	0.194	0.118	0.104	0.0	0.0
59	100.0	3.70	1.70	0.0	0.0	0.931	0.166	0.216	0.198	0.172	0.0	0.0
TARGET TYPE # 12												
60	100.0	4.70	4.27	0.0	0.0	0.390	0.376	0.058	0.012	0.044	0.090	0.006
61	70.0	1.80	1.69	0.0	0.0	0.315	0.355	0.382	0.316	0.032	0.066	0.005
TARGET TYPE # 13												
62	90.0	3.40	2.30	0.0	0.0	0.881	0.126	0.171	0.158	0.166	0.0	0.0
63	100.0	3.20	1.38	0.0	0.0	0.998	0.046	0.145	0.118	0.0	0.0	0.0
64	40.0	4.50	2.80	0.0	0.0	0.852	0.308	0.194	0.118	0.104	0.0	0.0
65	100.0	3.70	1.70	0.0	0.0	0.931	0.166	0.216	0.198	0.172	0.0	0.0
TARGET TYPE # 14												
66	100.0	4.70	4.27	0.0	0.0	0.390	0.376	0.058	0.012	0.044	0.090	0.006
67	70.0	1.80	1.69	0.0	0.0	0.315	0.355	0.382	0.316	0.032	0.066	0.005
TARGET TYPE # 15												
68	90.0	3.40	2.30	0.0	0.0	0.881	0.126	0.171	0.158	0.166	0.0	0.0
69	100.0	3.20	1.38	0.0	0.0	0.998	0.046	0.145	0.118	0.0	0.0	0.0
70	40.0	4.50	2.80	0.0	0.0	0.852	0.308	0.194	0.118	0.104	0.0	0.0
71	100.0	3.70	1.70	0.0	0.0	0.931	0.166	0.216	0.198	0.172	0.0	0.0
TARGET TYPE # 16												
72	100.0	4.70	4.27	0.0	0.0	0.390	0.376	0.058	0.012	0.044	0.090	0.006
73	70.0	1.80	1.69	0.0	0.0	0.315	0.355	0.382	0.316	0.032	0.066	0.005
TARGET TYPE # 17												
74	90.0	3.40	2.30	0.0	0.0	0.881	0.126	0.171	0.158	0.166	0.0	0.0
75	100.0	3.20	1.38	0.0	0.0	0.998	0.046	0.145	0.118	0.0	0.0	0.0
76	40.0	4.50	2.80	0.0	0.0	0.852	0.308	0.194	0.118	0.104	0.0	0.0
77	100.0	3.70	1.70	0.0	0.0	0.931	0.166	0.216	0.198	0.172	0.0	0.0
TARGET TYPE # 18												
78	100.0	4.70	4.27	0.0	0.0	0.390	0.376	0.058	0.012	0.044	0.090	0.006
79	70.0	1.80	1.69	0.0	0.0	0.315	0.355	0.382	0.316	0.032	0.066	0.005
TARGET TYPE # 19												
80	90.0	3.40	2.30	0.0	0.0	0.881	0.126	0.171	0.158	0.166	0.0	0.0
81	100.0	3.20	1.38	0.0	0.0	0.998	0.046	0.145	0.118	0.0	0.0	0.0
82	40.0	4.50	2.80	0.0	0.0	0.852	0.308	0.194	0.118	0.104	0.0	0.0
83	100.0	3.70	1.70	0.0	0.0	0.931	0.166	0.216	0.198	0.172	0.0	0.0
TARGET TYPE # 20												
84	100.0	4.70	4.27	0.0	0.0	0.390	0.376	0.058	0.012	0.044	0.090	0.006
85	70.0	1.80	1.69	0.0	0.0	0.315	0.355	0.382	0.316	0.032	0.066	0.005
TARGET TYPE # 21												
86	90.0	3.40	2.30	0.0	0.0	0.881	0.126	0.171	0.158	0.166	0.0	0.0
87	100.0	3.20	1.38	0.0	0.0	0.998	0.046	0.145	0.118	0.0	0.0	0.0
88	40.0	4.50	2.80	0.0	0.0	0.852	0.308	0.194	0.118	0.104	0.0	0.0
89	100.0	3.70	1.70	0.0	0.0	0.931	0.166	0.216	0.198	0.172	0.0	0.0
TARGET TYPE # 22												
90	100.0	4.70	4.27	0.0	0.0	0.390	0.376	0.058	0.012	0.044	0.090	0.006
91	70.0	1.80	1.69	0.0	0.0	0.315	0.355	0.382	0.316	0.032	0.066	0.005
TARGET TYPE # 23												
92	90.0	3.40	2.30	0.0	0.0	0.881	0.126	0.171	0.158	0.166	0.0	0.0
93	100.0	3.20	1.38	0.0	0.0	0.998	0.046	0.145	0.118	0.0	0.0	0.0
94	40.0	4.50	2.80	0.0	0.0	0.852	0.308	0.194	0.118	0.104	0.0	0.0
95	100.0	3.70	1.70	0.0	0.0	0.931	0.166	0.216	0.198	0.172	0.0	0.0
TARGET TYPE # 24												
96	100.0	4.70	4.27	0.0	0.0	0.390	0.376	0.058	0.012	0.044	0.090	0.006
97	70.0	1.80	1.69	0.0	0.0	0.315	0.355	0.382	0.316	0.032	0.066	0.005
TARGET TYPE # 25												
98	90.0	3.40	2.30	0.0	0.0	0.881	0.126	0.171	0.158	0.166	0.0	0.0
99	100.0	3.20	1.38	0.0	0.0	0						

ARXIV:1508.03462v1 [math.CO] 15 Aug 2015

Fig. 9 — Statistical results of target damage

RESOURCE CLASS	TYPE	AVERAGE LOSSES PERCENT	STD. DEV LOSSES
PEOPLE	1	32.830	14.394
	2	30.890	14.016
	3	27.940	16.968
	4	12.390	7.466
	7	26.330	31.526
	1000	13.260	6.380
	1000	15.230	11.599
AGE	1	1.080	0.932
	2	2.240	1.154
	3	7.030	2.788
	4	13.130	5.206
AMMO	1	0.150	0.220
	2	0.500	0.548
	3	0.960	0.824
TRAP	2	0.850	0.883
	3	0.300	0.306
	4	1.520	1.261

Fig. 10 — Statistical results of
resource losses

Appendix A
DETAILED DESCRIPTION OF TSARINA INPUT

The basic input cards employed with TSARINA are:

CONT	control card
DATA	TSAR data card
TGT	target card; one per target
TGT2	aircraft shelter card; one per shelter when the expected-value mode is to be used
ATT	attack card; one per weapon delivery pass (or group of identical passes)
ATT2	alternate attack card
EMD	effective miss distance card; one for each weapon type
REDO	controls sequential cases
END	terminates overall computation

The ATT2 card is actually two cards in sequence and the EMD card may have up to sixteen* supplementary cards. Each TGT card is followed by as many cards as are required to specify the resources located at that target. A detailed description of the entries for each type of card is presented on the pages that follow.

The general arrangement of data on all basic card types is similar; the card name is placed (left-adjusted) in the first four columns and the data are entered in the eleven 6-column fields between columns 7 and 72. All data are read with an I6 format, i.e., they are integers, except that, as will be noted

*Up to 33 supplementary cards, if there are more than ten target types.

from the descriptions defining data entry, two data are entered in certain fields of the CONT and DATA cards and on the supplementary target cards. Columns 5 and 6 are also used on several cards, as will be described. Furthermore, the name of the target complex being studied and a name for each target may be included in columns 73 through 80 of the CONT and TGT* cards, respectively; any alphanumeric names are acceptable.

All linear dimensions should be in consistent units** (e.g., feet) and the target orientation and the attack heading entries should be in degrees.

*When electronic card images are used, columns 73 through 88 may be used for alphanumeric target names by entering a "1" in column 18 of the CONT card.

**If ATT2 cards are to be used, all linear dimensions must be in feet.

CONT

The CONT card controls the mode of operation, the choice of random number generator, the number of trials (attack replications), and printout options; it also specifies the minimum clear length (MCL) and minimum clear width (MCW) for runway attack effectiveness calculations, and controls runway repair assessment. This card should be the first card to be entered.

ColumnsData Entry

1-4	CONT
6	If unity, program computes resource damage levels appropriate for entry into TSAR.
8-9	When 0, the seed for the random number generator is the same for all runs. If greater than 0, the seed is changed from run to run; if equal to -1, the random number generator is locked out.
10-12	Number of target types to be entered.
13-15	Desired number of replications. Default is 1.
16-18	If 1, descriptive data on the CONT and TGT cards may extend to column 88, rather than be constrained to an 80-column format.
19-21	Controls printout options as follows: If entry is: 5 Prints multiple trial statistics plus a condensed listing of hits by trial 4 Prints multiple trial statistics plus a condensed listing of runway status by trial 3 Prints multiple trial statistics only 2 All above plus runway results for each trial 1 All above plus hit summary for each trial -1 All above plus all hits and target corners -2 All above plus all impact points
23-24	Controls printout options for resource damage: 1 Damage fraction formatted only for user 2 Damage fraction formatted only for TSAR 3 Both formats

- 29-30 Normally set to zero; when initialized greater than zero, intermediate computational information is output for program test purposes. If set to greater than 7, the random number generator is locked out. See the program source listing in Appendix C for the effect of other values.
- 31-36 Minimum clear length (MCL) for aircraft operations. (Used to test if the runways are open.)
- 37-42 Minimum clear width (MCW) for aircraft operations. (Used to test if the runways are open.)
- 45 When the entry is 1, runway results will include the minimum number of craters to be repaired for the runway to meet the MCL and MCW criteria.
- 48 When the entry is 1, a plot of all impact points will be included for all of the runways. If the printout option entry is defined as 1 or less than 31; when the entry is 2, impact points are provided for each runway & other than 31 or less.
- 49-54 The distance across the runway that the "minimum runway rectangle" is to be shifted in checking for an adequate section; the default value is 1.
- 55-60 The distance along the runway that the minimum runway rectangle is to be shifted in checking for an adequate section; the default value is 250.
- 73-80 A name can be entered here for the entire target complex and it will appear in the heading of the output listing.

DATA

The DATA card controls the form of the output to TSAR, defines the time and location of the attack, and provides TSARINA the necessary resource identity data for communicating results to TSAR. This card is not required if the results are not to be used in TSAR.

<u>Columns</u>	<u>Data Entry</u>
1- 4	DATA
5- 6	If unity, statistical results are punched on cards for subsequent processing and reorganization using auxiliary programs.
12	Enter the number of trials for which damage data and resource loss data are to be stored for TSAR.
18	Enter the number of trials for which card copies of the damage data and resource loss data are to be punched.
24	Number of airbase under attack (as interpreted within TSAR).
29-30	Day of attack
35-36	Hour of attack
41-42	Minute of attack
47-48	Target type number assigned to aircraft shelters
53-54	Target type number assigned to aircraft taxiways
59-60	Target type number assigned to aircraft parking ramps
61-66	Percentage of the aircraft that are damaged by air attack that cannot be repaired.

TGT

Each TGT card designates the location, size, and orientation of a rectangular target.

<u>Columns</u>	<u>Data Entry</u>
1-3	TGT
7-12	The X-coordinate of the westernmost corner of the target. If the westernmost corner of any target does not fall in the first quadrant of the X-Y coordinate system, TSARINA automatically translates the origin (a multiple of 1000 feet) so that all targets are in the first quadrant. If after translation, targets do not fall within the allowed 32000 x 32000 area, they are "reset" to the edge of that area and the user is notified.
13-18	The Y-coordinate of the westernmost corner of the target. If a target boundary runs exactly north-south, the X and Y coordinates of the southwestern corner should be specified.
19-24	Target dimension along the boundary running north-east (or north) from the reference corner specified in the two previous fields.
25-30	Target dimension along the boundary running southeast (or east) from the reference corner.
31-36	Heading in degrees of the northeast (or north) heading boundary of the target (along the dimension specified in columns 19 to 24).
41-42	Target type. Targets may be grouped into up to 10 (or 20) different categories with like vulnerabilities. This entry is used in conjunction with the effective miss distance on the EMD card. Target type #1 is restricted to runways and taxiways that may be used for flight operations; there can be at most 5 targets of this type. The user may specify other target types as aircraft parking ramps, taxiways, and aircraft shelters; if used with TSAR, the target type number selected to designate each of these target sets must be entered on the DATA card. For all other targets, structures with materiel of like vulnerability can be assigned a common target type number; if additional stratification in results is desired, targets of like vulnerability may be grouped under two or more target types.

- 48 If greater than zero, all hit locations will be saved and printed when the entry in column 24 of the CONT card is zero or less.
- 49-54 Facility number as understood in TSAR, if the target is a maintenance shop or a flight-line personnel assembly area (should not be used if the results are not to be used in TSAR).
- 55-60 Class of resource stored in the facility, if storage is restricted to 100 percent of one resource class, or to 100 percent of one type of one resource class.
- 61-66 Type of resource, if only one type is stored in the facility (all types are inferred if blank or zero).
- 67-70 Number of subsequent cards used to describe the types and quantities of resources stored in this facility (use only when columns 55-66 are blank).
- 73-80 Target descriptions; columns 73-88 may be used if input is not restricted to 80 columns.

SUPPLEMENTARY TARGET CARDS

Each TGT card may be followed by as many supplementary cards as are necessary to define the resources that are located in that particular target. Each of these cards is read with a 5X, 5(3X,I2, I5, F5.0) format that provides for five descriptions of resource class, resource type, and percent at the target. Entry of the letter "C" (for class), preceding the resource class (in columns 7, 22, 37, 52, and 67), has been found helpful in reviewing the large data sets required to represent a complex airbase (see Figure 2).

<u>Columns</u>	<u>Data Entry</u>
9-10	Number identifying the resource class:
24-25	1 Personnel 5 TRAP
39-40	2 AGE and equipment 6 Building materials
54-55	3 Aircraft parts 7 POL
69-70	4 Munitions
11-15	Number identifying which type of the specified
26-30	resource class is located here.* If there is no
41-45	entry for "type", all types of the specified class
56-60	(that have not otherwise been specified) are
71-75	assumed to be present.
16-20	Percentage of the base stocks of the specified
31-35	type and class of resource that are located in
46-50	this target. Whole numbers are interpreted
61-65	as percentages; a decimal entry is required to
76-80	specify tenths of a percent. The output listing
	of resource storage data is in tenths of percent;
	e.g., 273 implies 27.3 percent.

*The number "1000" added to a personnel designation specifies off-duty personnel, and "100" added to a munitions designator specifies weapons that are not assembled.

TGT2

These cards are used to input the location of aircraft shelters when resources are not to be located in the shelters and the damage to these targets is to be estimated with the expected-value mode. The identification "TGT2" designates selection of this option. All aircraft shelters must be handled in a consistent fashion; i.e. their characteristics must either all be entered on TGT type cards, or all entered on TGT2 type cards. The inputs for the TGT2 cards are identical with the TGT cards for columns 7-48; entries in columns 49-80 are ignored.

ATT

The ATT card specifies the parameters of each weapon-delivery pass. Inputs required are the attack heading (measured from north in the coordinate system used to specify the targets), the desired mean point of impact (DMPI) for a single weapon or for the middle of a stick of weapons, the aiming error expressed as REP (range error probable) and DEP (deflection error probable), the ballistic error of the individual weapons, the number of weapons to be delivered in the pass, the stick length, the weapon type (related to the effective miss distance on the EMD card), and the probability of arrival at the target.

<u>Columns</u>	<u>Data Entry</u>
1-3	ATT
5-6	Total number of passes with identical characteristics; default = 1.
10-12	Attack heading; degrees from north.
13-18	The X-coordinate of the DMPI of a single weapon or the middle of a stick of weapons.
19-24	The Y-coordinate of the DMPI as above.
25-30	The REP
31-36	The DEP
37-42	Ballistic dispersion in range of individual weapons (R-DISP).
43-48	Ballistic dispersion in deflection of individual weapons (D-DISP). Default value is R-DISP.
49-54	The number of weapons in the stick.
55-60	The length of the stick (the distance between the first and last weapon of the stick in the absence of dispersion).
61-66	The weapon type (provides reference to the appropriate effectiveness data). An entry is required (an integer from 1 to 10); otherwise hits will not be recorded.
67-72	Probability of arrival at target; default = 100.

ATT2

The ATT2 card should be used in place of the ATT card when the user wishes assistance with trajectory calculations. With this card the user expresses the attack in terms of speed, altitude, dive angle, intervalometer settings, etc., and a special subroutine converts these inputs to those demanded on the ATT card. The conversion procedure is the JMEM/AS Open-End Method Zero as outlined in the Users' Manual for JMEM/AS Open-End Methods, Wang Labs., Inc., Tewksbury, Mass., August 1974.

Both ATT and ATT2 type cards may be used in the same run; the order of entry is of no importance. When ATT2 cards are used, the input data will be reproduced as submitted, as well as being tabulated in the normal manner, after conversion.

Data input with the ATT2 procedure require two cards. The first card is labeled ATT2 in the first 4 columns and has input similar to that on an ATT card (all fields are read with an I6 format); a second unlabeled card is mandatory following each ATT2 card. The format for both cards follows. When these cards are used, all linear dimensions in the input data must be in feet.

<u>Columns</u>	<u>Data Entry</u>
1-4	ATT2
5-6	Total number of passes with identical characteristics; default = 1.
10-12	Attack heading in degrees from north.
13-18	The X-coordinate of the desired mean point of impact (DMPI) of a single weapon or the middle of a stick of weapons.
19-24	The Y-coordinate of the DMPI as above.
25-30	The CEP in the normal plane in mils, or, if DEP is specified, a constant which, when divided by the sine of the impact angle, gives the REP, in mils.
31-36	The DEP in mils (if omitted, CEP controls).

37-42	Ballistic dispersion in mils.
49-54	The number of weapons in the stick.
61-66	The weapon type.
67-72	Probability of arrival at target; default = 100.

The data format for the second card of each ATT2 pair is as noted below (this card is used with a 6X, 916 format). Typical ballistic data required for this card are noted in Table A-1.

<u>Columns</u>	<u>Data Entry</u>
7-12	Aircraft velocity (kn).
13-18	Release altitude of last bomb (ft).
19-24	Dive angle at release (deg).
25-30	Terminal velocity of a low-drag weapon, or the first leg of a high-drag bomb (ft/sec) (See Table A-1) (VT1 in JMEM).*
31-36	Terminal velocity of a cluster bomblet or a high-drag bomb (ft/sec) (See Table A-1) (VT2 in JMEM).
37-42	Probable error in estimating and correcting for wind effects (ft/sec).
43-48	Cluster opening time or fin opening time for a high-drag bomb (msec) (TD in JMEM).
49-54	Intervalometer setting (msec).
55-60	Dispensor intervalometer setting (msec) (0 for clusters).

*Illustrative values are noted on Table A-1.

Table A-1
TYPICAL BALLISTIC PARAMETERS

Weapon	VT1 (fps)	VT2 (fps)	T or H
Mk-81 Mod 1	1850	0	0
Mk-81 SE	1100	208	300 msec
Mk-82 Mod 1	1900	0	0
Mk-82 SE	1200	240	350 msec
Mk-83	2250	0	0
Mk-84	2850	0	0
M-117 Unretarded	1950	0	0
M-117 Retarded	900	168	300 msec
M-118	2450	0	0
AN-M64A1	1600	0	0
AN-M65A1	2000	0	0
Mk-36 DST	1200	0	350 msec
CBU-38	450	0	0
CBU-52B/B	1000	230	Variable altitude (ft)
CBU-58/B	950	215	Variable altitude (ft)

SOURCE: Users' Manual for JMEM/AS Open-End Methods,
Wang Labs, Inc., Tewksbury, Mass., August 1974.

EMD

The EMD and supplementary cards provide information regarding weapon effectiveness against the several types of targets and the several classes of resources. The formats of the entries differ for point-impact weapons and for CBU munitions, and they differ between aircraft shelters and all other target types. Normally, each type of weapon will be represented by up to 17 cards (or 34 cards, if 11-20 target types are specified on the CONTROL card), although just the first card, the EMD card, could suffice for certain limited assessments.

For point-impact weapons (GP bombs or PGMs) the entries on the EMD card are:

<u>Columns</u>	<u>Data Entry</u>
1-3	EMD
5-6	Enter the maximum number of cards that are associated with each target type for this type of weapon.
8-9	Weapon type number
10-12	Weapon reliability (percentage)
13-18	R1 Radius of effectiveness versus target type #1
19-24	R1 Radius of effectiveness versus target type #2
o	
o	
o	
67-72	R1 Radius of effectiveness versus target type #10

The supplementary data for the several target types are located in the corresponding fields on the cards that immediately follow the first card. Definitions of the data to be entered on the EMD card and on the supplementary cards are noted below; somewhat different definitions apply for the target type that the user has designated as aircraft shelters, as will be outlined shortly.

<u>Card</u> <u>No.</u>	<u>All Target Types</u> <u>(except shelters)</u>
1 10	R1 Effective radius against target type
2 11	R2 Secondary effects radius
3 12	Personnel loss criteria
4 13	AGE loss criteria
5 14	Spare parts loss criteria
6 15	Munitions (and POL) loss criteria
7 16	TRAP loss criteria
8 17	Building materials loss criteria
9	Flag - Controls loss criteria interpretation

The data on cards 1 to 8 apply in the case of a near miss and those on cards 10 to 17 apply for a direct hit. If no data are entered for a direct hit, the near miss inputs are used; if any data are entered for a direct hit, only the values on the tenth through seventeenth cards are used. Null entries are interpreted as zero.

If 11 to 20 target types are treated, a second set of (up to 17) cards should be placed immediately after the first set described above; these cards are each read with a 12X, 1016 format.

The appropriate loss criterion for assessing resource damage is controlled by the user and is communicated with the value of the control variable Flag, which is entered on the ninth card. The value of Flag for each target type is a 6-digit code UVWXYZ, where U, V, W, X, Y, and Z express the user's choices regarding the treatment of the six resource classes. Their values are defined from 0 to 8 as:

- 0 Ignore this class of resource
- 1 Value represents probability of kill of these resources within a circle of radius R1
- 2 Value represents probability of kill of these resources within a circle of radius R2
- 3 As in 2, given that the R1 radius intersects the target perimeter.
- 4 Value is the radius of kill of these resources
- 5 Value is the radius of kill of these resources, given that the R1 radius intersects the target perimeter

Values 6, 7, and 8 are a combination of a radius and a probability of kill.

- 6 Value is the radius (times 1000) of an area within which the Pk is one-fourth that value of Pk within R1, plus the value of Pk within R1
- 7 Value is the radius (times 1000) of an area within which the Pk is one-fourth that value of Pk within R2, plus the value of Pk within R2
- 8 As in 7, given that the radius R1 intersects the target perimeter.

Thus, Flag = 321475 implies that personnel, AGE, parts, munitions (or POL), TRAP, and building materials are to be assessed by options 3, 2, 1, 4, 7, and 5, respectively, for the particular weapon type and target type for which it is listed.

The use of the EMD cards is somewhat different for aircraft shelters. When the Monte Carlo mode is used to assess damage to these targets, the second through ninth EMD cards are interpreted the same as for any other target type. When the expected value mode is used, only the third, fourth, and fifth of these cards are used, and the entries are interpreted as the damage probability to personnel, equipment, and spare parts when a weapon falls within a radius R1 from the shelter. The other entries, noted below, are interpreted identically in either mode.

<u>Card No.</u>	<u>Entry</u>
1	R1 Effective radius against aircraft in shelters with closed doors.
10	R3 Effective radius against aircraft in shelters with open doors.
12	Probability of aircraft damage in a shelter with a closed door, when a weapon falls within a radius of R1 from the shelter.
13	Probability of aircraft damage in a shelter with an open door, when a weapon falls within a radius of R3 from the shelter.
14	Probability the aircraft shelter is killed when a weapon strikes within a radius of R1 from the shelter.

If the weapons are CBU type munitions, the following entries are used with the EMD and the supplementary cards:

<u>Columns</u>	<u>Data Entry</u>
1-3	EMD
5-6	Enter the total number of cards (maximum = 8) that are associated with the first 10 target types for this type of weapon.
8-9	Weapon type number
10-12	Weapon reliability (percent)
13-18	Enter CBU pattern length as a negative entry
19-24	Enter CBU pattern width as a positive entry

The supplementary cards permit the user to specify the expected percentage loss of various classes of resources when they are within the CBU bomblet pattern:

Card	
<u>No.</u>	<u>Entry</u>
13-18	Expected percentage loss at #1 type targets
19-24	Expected percentage loss at #2 type targets
o	
o	
o	
67-72	Expected percentage loss at #10 type targets

The first supplementary card is blank; the others are organized similarly to those for point-impact weapons. That is, the third through eighth cards apply to personnel, AGE and equipment, spare parts, munitions (or POL), TRAP, and building materials, respectively.

EQUI

If several TSAR resources are to be assigned the same TSARINA damage percentage, each of their numbers should immediately follow the number for the equivalent TSARINA resource. If necessary, the designator list can be continued on a subsequent card image when the eleven data fields are full; the list is terminated either by a null entry or by another resource class designator. However, the equivalence card cannot be used to equate a TSARINA resource designation to a type #0 resource to signify "all types not otherwise specified" for TSAR; to take advantage of that option, the TSARINA resource designation should itself be #0 (see p. 13). If the first entry following the TSARINA designator is -1, the TSARINA damage estimate for that resource is not reported to TSAR.

EQUI	5003	5	1	3	5	7	5003	6	2	4	6
EQUI	10	12	0								

In this example, the first, third, fifth, and seventh types of aircraft spare parts, as defined for TSAR, are to be assigned the damage level assessed for the fifth type of spare part in TSARINA; similarly, the second, fourth, sixth, tenth, and twelfth TSAR spare parts are assigned the damage level assessed for the sixth type of spare part in TSARINA.

EQUI 5001 7 -1

In this case, estimates of casualties suffered by type #7 personnel are not to be reported to TSAR.

A final complication is introduced for differentiating between on-duty and off-duty personnel, and between assembled and unassembled munitions. TSAR personnel designators with values less than 1000 refer to on-duty personnel; off-duty personnel are specified by adding 1000 to their normal designator. Note that types 0 and 1000 refer, respectively, to all on-duty and off-duty personnel not otherwise specified. Unassembled munitions are designated by adding 100 to the nominal munition designation.

REDO

The REDO card is used to terminate the input for one case and initiate a new case with some or all of the previous inputs, as described earlier.

<u>Columns</u>	<u>Data Entry</u>
1-4	REDO
7-12	All targets will be retained unless the entry is unity (1); in that case a new set of targets and a new set of attacks are required.
13-18	The number of prior attacks to be retained when the targets are not changed. Each attack is numbered in the order in which it is entered; the attacks retained are selected from the top of that ordered list. All will be retained if there is no entry. Use a negative entry (-1) if none are to be retained.
19-24	An entry of unity (1) suppresses the input listings for targets and/or for attacks and weapons if no changes have been made in these data sets from the prior case.

58

END

An END card must be included at the end of all data entry cards.

Columns
1-3

Data Entry
END

Appendix B

DEFINITIONS OF VARIABLES AND ARRAYS USED IN TSARINA "COMMON" STATEMENTSKey Variables

ACLOSS	The percentage of damaged aircraft that are not reparable.
ALLMC	Switch; internally set to unity when aircraft shelters are handled with the Monte Carlo mode.
BASE	The number of the airbase, in TSAR, at which the attack occurs.
CHANGE	Switch; set to unity between cases when the target data are to be changed.
DAY	The day, during the TSAR simulation, on which the air attack is presumed to occur.
HOURL	The hour, during the TSAR simulation, during which the air attack is presumed to occur.
INL	Distance along the runway the "minimum runway rectangle" is shifted.
INTSAR	Switch; set to unity when results are to be generated for TSAR.
INW	Lateral distance the minimum runway rectangle is shifted in checking for an adequate section.
ISAVE	Switch; set to unity if resource damage results are to be generated for the auxiliary FORMATER program.
ITRIAL	Number of the current trial.
KCBU	Switch; set to unity if any weapons are CBUs.
KPTI	Switch; set to unity if any weapons are the point-impact type.
KTEST	Index controlling variety of debugging printout options.
LAST	Switch; set to unity for last case.
LIST	Switch; when set to unity, target and/or attack input lists are suppressed when unchanged.

MCL	Minimum adequate length for required runway.
MCR	Switch; set to unity when runway availability is to be checked.
MCW	Minimum adequate width for required runway.
MINUTE	The minute, during the TSAR simulation, at which the air attack is presumed to occur.
MODE	Index controlling mode of operation.
MTT	Largest target-type number in the target array.
MXITEM	Maximum number of entries in the STOCKS array.
NA	Total number of weapon-delivery passes.
NAM	Maximum permissible number of weapon-delivery passes.
ND	Number of types of weapons in overall attack.
NHITD	Switch; set to unity when the expected-value mode is specified.
NJMEM	Number of weapon-delivery passes that require trajectory calculations.
NOAGE	Maximum number of entries in the AGE array; one greater than the size of the AGESTK array in TSAR.
NOEQUI	Maximum number of entries in the EQUIV array.
NOFAC	Maximum number of entries in the FACLT array in TSAR.
NOMATL	Maximum number of entries in the MATERL array; one greater than the corresponding TSAR array.
NOMUN	Maximum number of entries in the AMMO array; 101 units greater than the MUNSTK array in TSAR.
NOPART	Maximum number of entries in the PARTS array; one greater than the corresponding TSAR array.
NOPEO	Dimension of PEOPLE array in TSARINA; equals $(2*NOPEOP + 2)$.
NOPEOP	Maximum number of entries in the PEOPLE array in TSAR.
NOPOL	Maximum number of entries in the POL array.

NOTRAP	Maximum number of entries in the TRAP array; one greater than the corresponding TSAR array.
NPLOT	Switch; set to 1 or 2 if runway impact plots are desired.
NPRINT	Index controlling results output.
NREDO	Switch; set to unity if an additional case is specified.
NREP	Switch; set to unity when repair requirements are to be assessed.
NSAVE	The number of weapon-delivery passes saved from one case to be used in the next case.
	Number of targets to be retained for a subsequent case.
	Number of weapon-delivery passes to be retained for a subsequent case.
NSM	Total number of aircraft shelters.
NST	Maximum number of targets for which hits can be stored.
NSTAT	Cumulative number of trials in which the minimum runway was available.
NT	Total number of targets entered using the TGT cards.
NT2	Total number of targets entered using the TGT2 cards.
NTM	Maximum permissible number of targets.
NTRIAL	Total number of trials specified.
PDAM	Switch; position controls output formats for trial-to-trial damage summaries.
PUNCH	When greater than zero, output for TSAR is card-punched.
RAMPS	Target-type chosen to designate aircraft parking aprons and ramps.
REPAIR	The minimum number of crater repairs required to clear the minimum area for flight operations.
SHELT	Target-type chosen to designate aircraft shelters.
TSAR	When greater than zero, output for TSAR is stored on disk.
TXWYS	Target-type chosen to designate taxiways.

KEY ARRAYS

All arrays listed in labeled Common in TSARINA are defined below. The first seven arrays store data pertaining to AGE and equipment, munitions, building materials, aircraft spare parts, personnel, POL, and TRAP, respectively. The definitions shown below the array names are the same for all of these arrays.

AGE(I,J)

AMMO(I,J)

MATERL(I,J)

PARTS(I,J)

PEOPLE(I,J)

POL(I,J)

TRAP(I,J)

I = Resource subcategory

J = 1 Pointer to the location in the STOCKS array, where the first quantity of this resource is stored.

2 Cumulative losses at all targets where this resource is stored.

3 Square of the cumulative losses.

4 Pointer to the location in the EQUIV array, where the first equivalent TSAR resource category designations are stored.

ATT(I,J) Storage array for weapon-delivery data.

I Weapon-delivery pass number; numbered internally in order of entry.

J = 1 Heading (deg).

2 X-coordinate of desired mean point of impact.

3 Y-coordinate of DMPI.

4 Range error probable of DMPI.

5 Deflection error probable of DMPI.

6	Dispersion in range (ground plane).
7	Number of weapons released in pass.
8	Length of stick (in ground plane).
9	Weapon type.
10	Dispersion in deflection.
11	Probability attacker arrives at target.
CBUHT(J,K)	Impact coordinates of the centroid of the Jth CBU pattern.
K = 1	X-coordinate.
2	Y-coordinate.
COV(L)	Fraction of target L covered by one or more CBU patterns.
COV2(I,J)	
J = 1	Expected number of weapons that impact within R1 feet of aircraft shelter "I".
2	Expected number of weapons that impact within R2 feet of aircraft shelter "I".
EMD(I,J,K)	Weapon effectiveness data.
I	Weapon type.
J	Target type.
K = 1	Effective miss distance R1 for a near miss.
2	Effective damage radius R2 for a near miss.
3	Personnel damage factor for a near miss.
4	Equipment damage factor for a near miss.
5	Aircraft spare parts damage factor for a near miss.
6	Munitions damage factor for a near miss.
7	TRAP damage factor for a near miss.
8	Building material damage factor for a near miss.

9 Coded Flag defining the criteria for assessing resource damage.

10 Effective miss distance R1 for a direct hit.

11 Effective damage radius R2 for a direct hit.

12 Personnel damage factor for a direct hit.

13 Equipment damage factor for a direct hit.

14 Aircraft spare parts damage factor for a direct hit.

15 Munitions damage factor for a direct hit.

16 TRAP damage factor for a direct hit.

17 Building material damage factor for a direct hit.

EQUIV(NOEQUI) Used to store the resource designators to be used for reporting damage to TSAR.

FACLT(I) Used to store the TSAR facility number for those structures whose damage is to be reported to TSAR.

HIT(I,J,K) Storage array for hit locations on specified targets.

I Ith of those targets for which hit data are to be stored.

J = 1 X-coordinate.

2 Y-coordinate.

3 Weapon type.

K Number of hits on the Ith target.

HITR(I,J,K) Storage array for hit locations on type #1 targets (i.e., runways and taxiways).

I,J,K See HIT(I,J,K).

IR(N) Switch; set to unity if the Nth weapon-delivery attacker fails to reach target.

IZ(I) Designates the zone for each target (see subroutine TGTZON).

IZONE(K,J)	Denotes which of the ordered targets fall in the Kth target zone.
J = 1	Lowest numbered target in the Kth zone.
2	Highest numbered target in the Kth zone.
MHIT(K)	Target number of the Kth target for which hit location data are to be stored.
MSTAT(J)	Storage array for accumulating trial results of runway availability tests.
J = 1	Minimum number of repairs required to open a minimum runway.
2	Square of J = 1, above.
3-8	Not used.
MTYPE(1)	Index that specifies whether or not supplementary data are to follow the EMD card for weapon type 1.
NAME(I,4)	Stores either a two-word or four-word alphanumeric name for each target.
I	Target number
NCBU(L)	Number of CBU weapon patterns that cover all or part of target L.
NHIT(L)	Number of hits on target L; by both point-impact and CBU weapons.
NRW(I)	Target number of the Ith runway entered.
OHIT(I)	Counts near misses for each target.
P(L,K)	Damage estimates for targets handled with the Monte Carlo mode.
K = 1	Expected fraction of target L that is within the radius R1 of point-impact weapons.
2	Expected fraction of target L that is within the radius R2 of point-impact weapons.
3	Fraction of personnel casualties expected at target L.

4	Fraction of equipment losses expected at target L.
5	Fraction of spare parts losses expected at target L.
6	Fraction of munitions (or POL) losses expected at target L.
7	Fraction of TRAP losses expected at target L.
8	Fraction of building material losses expected at target L.
P2(L,K)	Damage estimates for aircraft shelters.
K = 1	Expected fraction of target L within radius R1 of the weapon impacts.
2	Expected fraction of target L within radius R2 of the weapon impacts.
3	Probability that a sheltered aircraft is damaged when the shelter door is closed.
4	Probability that the shelter is killed.
5	Probability that a sheltered aircraft is damaged when the shelter door is open.
6	Fraction of personnel casualties expected at target L.
7	Fraction of equipment losses expected at target L.
8	Fraction of spare parts losses expected at target L.
SHEL(N)	The TSARINA-generated target number for the Nth shelter.
STAT(L,J)	Storage array for accumulating trial results.
L	Target number.
J = 1	Number of hits by point-impact weapons.
2	Square of J = 1, above.
3	Trials with at least one hit.
4	Fractional coverage by CBU weapons.
5	Square of J = 4, above.

6	Fractional target coverage within radius R1 of point-impact weapons.
7	Fractional target coverage within radius R2 of point-impact weapons.
8	Fractional personnel casualties.
9	Fractional equipment losses.
10	Fractional spare parts losses.
11	Fractional munitions losses.
12	Fractional TRAP losses.
13	Fractional building material losses.
STAT2(I,J)	Storage array for accumulating trial results for targets of a given type.
I	Target type.
J = 1	Fraction of the targets of type I that received at least one hit.
2	Square of J = 1, above.
STOCKS(I,J)	Resource storage location information.
J = 1	Target number at which resource is located.
2	The percent of the resource stored in this location (in tenths of percent).
3	Pointer to next target with the same type of resource.
TGT(L,J)	Storage array for target data.
L	Target number; numbered internally in order of entry.
J = 1	X-coordinate of westernmost corner (#1).
2	Y-coordinate of corner #1.
3	X-coordinate of corner #2.
4	Y-coordinate of corner #2.
5	X-coordinate of corner #3.

6	Y-coordinate of corner #3.
7	X-coordinate of corner #4.
8	Y-coordinate of corner #4.
9	Heading of northeast target leg.
10	Target type.
11	Switch; hits stored when reset to unity.
12	Dimension of northeast target leg.
13	Dimension of southeast target leg.
14	Facility number of target.
TGT2(L,J)	Storage array for aircraft shelter data.
L	Target number; numbered internally in order of entry.
J = 1	X-coordinate of westernmost corner (#1).
2	Y-coordinate of corner #1.
3	X-coordinate of corner #2.
4	Y-coordinate of corner #2.
5	X-coordinate of corner #3.
6	Y-coordinate of corner #3.
7	X-coordinate of corner #4.
8	Y-coordinate of corner #4.
9	Heading of northeast target leg.
10	Target type.
11	Switch; hits stored when reset to unity.
12	Dimension of northeast target leg.
13	Dimension of southeast target leg.
14	Facility number of target.

TO(I,J)	Target order array in which targets are ordered according to increasing values of the sum of the coordinates of the western corner.
I	Ith target in the ordered array.
J = 1	Value of (X+Y) for the 1th ordered target.
2	Number of the target as initially entered.
WPNREL(1)	Reliability of weapon type 1.

Appendix C

TSARINA SOURCE CODE AND COMMENTS

```

1.  C  MAIN - TSARINA - AIRBASE DAMAGE ASSESSMENT INPUTS FOR TSAR.
2.      IMPLICIT INTEGER *2 (A-Z)
3.      INTEGER *4 MSTAT, NBASE1, NBASE2
4.      REAL *4 STAT
5.  C  THE FOLLOWING JCL INSERTS TSARINA'S COMMON "BASIC"
6.  //  DD DSN=*,STEP1.COMMON,DISP=SHR
7.  //  DD *,DCB=BLKSIZE=800
8.  C
9.  C  ***** "BASIC" COMMON IS STORED IN TSARIN.COMMON *****
10. C  FILE AS "SAVE TSARIN.COMMON REP CARD (10)"
11. C
12. C  INTEGER*4 NAME, EMD
13. C  REAL *4 P, P2, COV, COV2, WPNREL
14. C
15. C  COMMON /BASIC/ NAME(500,4),P(500,8),P2(100,8),COV(500),
16. C  X EMD(10,20,17), MTYPE(10), COV2(100,2), WPNREL(10),
17. C  X IZ(500), NT, TGT(500,14), NT2, TGT2(100,13), SHEL(100),
18. C  X NA, ATT(50,11), ND, NTM, NSM, NAM, MTT, NST, ALLMC,
19. C  X MODE, KTEST, NPRINT, PDAM, NREDO, MCR, MCL, MCW, NRW(5),
20. C  X NAMES, LAST, BASE, DAY, HOUR, MINUTE, ACLOSS,
21. C  X INTSAR,REPAIR,FACLT(500),NOFAC,SPEC,SHELT,TXWYS,RAMPS,
22. C  X TO(500,2), IZCNE(100,2), NHIT(500), OHIT(500), MHIT(20),
23. C  X HIT(20,3,25), HITR(5,3,250), NCBU(500)
24. C
25. C  *****
26. C  THE LINEAR ARRAY 'ZO', AND THE 2 ZERGING LOOPS, BELOW, MUST
27. C  BE REDIMENSIONED IF ANY PART OF THE COMMON 'STORES' IS CHANGED.
28. C  **** NOTE THAT ALL ENTRIES ARE *4 EXCEPT 'STOCKS'
29. C  *****
30. C  INTEGER *4 MXITEM,NJPED,NOAGE,NOPART,NOMUN,NOTRAP,NOMATL,NOPOL
31. C  INTEGER *4 PEOPLE,AGE,PARTS,AMMO,TRAP,MATERL,PCL,NOEQUI,NOPECP
32. C  COMMON /STORES/ MXITEM,NOPEC,NOAGE,NOPART,NOMUN,NOTRAP,NOMATL,
33. C  X NOPOL,NOECUI,NOPECP,PEOPLE(20,4),AGE(101,4),PARTS(401,4),
34. C  X AMMO(126,4),TRAP(26,4),MATERL(26,4),PCL(10,4),STOCKS(1000,3),
35. C  X EQUIV(400)
36. C  *****
37. C  COMMON /STATS / STAT(500,17),MSTAT(8),NTRIAL,ITRIAL,NSTAT
38. C  COMMON /CONTRL/ NREP,NPLCT,INW,INL,CHANGE,NSAVE,LIST,NJMEM
39. C  COMMON /OUTPUT/ TSAR, PUNCH, NBASE1, NBASE2
40. C
41. C  THE CBU CODE IS LIMITED TO 200 CLUSTER BOMBLET CONTAINER IMPACTS
42. C
43. C  COMMON /CBUHIT/ CBUHT(200,2), IR(50), KCBU, KPTI
44. C  *****
45. C  DIMENSION ZO(7402)
46. C  EQUIVALENCE (ZO(1), MXITEM)
47. C  *****
48. C  NTM = 500
49. C  ***** NTM IS THE MAXIMUM NUMBER OF TARGETS *****
50. C  ARRAYS: TGT, TO, NHIT, OHIT, P, COV, NAME, NCBU, STAT,
51. C  FACLT, IZ
52. C  NT2M = 100
53. C  ***** NT2M IS THE MAXIMUM NUMBER OF SHELTERS, WHEN THEY ARE AGGREGATED
54. C  ARRAYS: TGT2, P2, COV2, SHEL (SEE ALSO EHIT1 IN EXPHIT)
55. C  NAM = 50
56. C  ***** NAM IS THE MAXIMUM NUMBER OF ATTACKS
57. C  ARRAYS: ATT, IR
58. C  NST = 20
59. C  ***** NST IS THE NUMBER OF TARGETS FOR WHICH HITS CAN BE STORED.
60. C  ARRAYS: HIT, MHIT
61. C  ***** TO CHANGE ANY OF THE PRECEDING DIMENSIONS, MAKE THE APPROPRIATE
62. C  ***** CHANGES IN THE ARRAYS AND THEN CHANGE THE LIMITING VALUE.
63. C  *****
64. C
65. C  THE OUTPUTS FOR TSAR MAY BE PRINTED, PUNCHED ON CARDS AND FILED
66. C  DIRECTLY ON DISK FOR PRODUCTION RUNS. THE 2ND AND 3RD OPTIONS
67. C  ARE CONTROLLED BY THE VARIABLES "PUNCH" AND "TSAR", RESPECTIVELY,
68. C  AND ARE EXERCISED WHEN THOSE VARIABLES ARE INITIALIZED TO UNITY.
69. C
70. C  *****
71. C
72. C  DO 1 I = 1, 10556
73. C  1 ZO(I) = 0
74. C

```

```

75. C THE NUMBER OF FACILITIES AND THE DIMENSIONS OF THE ARRAYS
76. C IN THE "STORES" COMMON MUST BE ENTERED HERE. THE DIMENSIONS
77. C SHOULD BE ONE LARGER THAN THE ARRAYS USED IN THE COMPANION
78. C VERSION OF THE TSAR SIMULATION MODEL, EXCEPT FOR MUNITIONS
79. C AND PERSONNEL. FOR MUNITIONS THE DIMENSION SHOULD BE LARGER
80. C BY +101 AND FOR PERSONNEL IT SHOULD BE 2*NOPEOP + 2, WHERE
81. C NOPEOP IS THE NUMBER OF PERSONNEL TYPES AS USED IN TSAR.
82. C
83. C      NJFAC = 60
84. C
85. C      MXITEM = 1000
86. C      NOPEO = 202
87. C      NJAGE = 101
88. C      NOPART = 401
89. C      NUMUN = 126
90. C      NUTRAP = 26
91. C      NUMATL = 26
92. C      NOPOP = 10
93. C      NDEQUI = 400
94. C      NOPEOP = (NOPEO - 2)/2
95. C
96. C      NT = 0
97. C      NT2 = 0
98. C      NA = 0
99. C      NE = 0
100. C      MER = 0
101. C      CASE = 0
102. C      MWPNI = 0
103. C      NTRIAL = 1
104. C      NTENT = 0
105. C      NPREDI = 0
106. C      CHANGE = 0
107. C      NEWTGT = 1
108. C      DO 2 I = 1,10
109. C      MTYPE(I) = 0
110. C      WPNR(I) = 1.0
111. C      DO 7 J = 1,20
112. C      DO 2 K = 1,27
113. C      2 5MUN(I,K) = 0
114. C      DO 3 I = 1, NIM
115. C      3 1FACTY(I) = 0
116. C      4 1TRIAL = 0
117. C      FLAG = 0
118. C
119. C      *REPLACE RESOURCE LOCATION IN CONTINERS AS REQUIRED
120. C
121. C      IF (CHANGE .NE. 1) GO TO 1000
122. C      NT = 0
123. C      NT2 = 0
124. C      CHANGE = 0
125. C      FLAG = 1
126. C      NEWTGT = 1
127. C      DO 1010 I = 21, 10556
128. C      1010 20(1) = 0
129. C      1020 CONTINUE
130. C
131. C      ***** BEGIN NEW CASE *****
132. C
133. C      CASE = CASE + 1
134. C      DO 5 I = 1, 5
135. C      5 1NRW(I) = 0
136. C
137. C      CALL INPUT(CASE,MWPNI,NEWTGT,NTSAR,NPUNCH)
138. C
139. C      IF (NTRIAL .LT. 2) GO TO 25
140. C      IF ((CASE .EQ. 1).OR.(FLAG .EQ. 1)) GO TO 14
141. C      DO 12 J = 2,3
142. C      DO 6 I = 1,NOPEO
143. C      6 1PEOPLE(I,J) = 0
144. C      DO 7 I = 1,NOAGE
145. C      7 1AGE(I,J) = 0
146. C      DO 8 I = 1,NOPART
147. C      8 1PARTS(I,J) = 0
148. C      DO 9 I = 1,NUMUN
149. C      9 1AMMO(I,J) = 0
150. C      DO 10 I = 1,NUTRAP

```

```

151.      10  TRAP(I,J) = 0
152.      DO 11 I = 1,NOMATL
153.      11  MATERL(I,J) = 0
154.      DO 12 I = 1,NOPOL
155.      12  POL(I,J) = 0
156.      14  CONTINUE
157.      NSTAT = 0
158.      DO 15 I = 1,NT
159.      DO 15 N = 1,13
160.      15  STAT(I,N) = 0.0
161.      25  DO 30 I = 1, NST
162.      30  MHIT(I) = 0
163.      40  CONTINUE
164.      DO 45 N = 1,8
165.      45  MSTAT(N) = 0
166.      CALL TGTDIM
167.      DO 50 I = 1,NT2M
168.      DO 50 J = 1,8
169.      50  P2(I,J) = 0.0
170.      IF (NT2.EQ. 0) GO TO 60
171.      DO 55 I = 1,NT2
172.      DO 55 N = 1,2
173.      55  COV2(I,N) = 0.0
174.      C
175.      CALL EXPFIT
176.      C
177.      IF (NT.EQ. 0) GO TO 200
178.      60  IF (KPTI.EQ. 0) GO TO 100
179.      C
180.      CALL TGTORD
181.      C
182.      CALL TGTZON
183.      C
184.      100 CONTINUE
186.      C ***** BEGIN NEW TRIAL *****
186.      ITRIAL = ITRIAL + 1
187.      IF (ITRIAL.GT. NTSAR) TSAR = 0
188.      IF (ITRIAL.GT. NPUNCH) PUNCH = 0
189.      DO 105 I1 = 1, NST
190.      DO 105 I2 = 1,3
191.      DO 105 I3 = 1, 25
192.      105  HIT(I1,I2,I3) = 0
193.      DO 110 I = 1,NT
194.      COV(I) = 0.0
195.      DO 108 N = 1,8
196.      108  P(I,N) = 0.0
197.      NCBU(I) = 0
198.      OHIT(I) = 0
199.      110  NHIT(I)=0
200.      IF (NSM.EQ. 0) GO TO 114
201.      DO 112 I = 1, NT2M
202.      DO 112 N = 1, 8
203.      112  P2(I,N) = 0.0
204.      114  CONTINUE
205.      DO 115 I1 = 1, 5
206.      DO 115 I2 = 1, 3
207.      DO 115 I3 = 1, 250
208.      115  HITR(I1,I2,I3) = 0
209.      CALL BOMB
210.      IF (KCBU.EQ. 1) CALL CBU
211.      REPAIR = 0
212.      IF (MCR.NE. 0) CALL CHECKR
213.      ITRIAL.LT. 2) GO TO 170
214.      DO 140 I = 1, NT
215.      AID = NHIT(I) - NCBU(I)
216.      STAT(I,1) = STAT(I,1) + AID
217.      STAT(I,2) = STAT(I,2) + AID*AID
218.      IF ( AID.GT. 0.0) STAT(I,3) = STAT(I,3) + 1.
219.      STAT(I,4) = STAT(I,4) + COV(I)
220.      STAT(I,5) = STAT(I,5) + COV(I)*COV(I)
221.      DO 140 J = 1,8
222.      STAT(I,J+5) = STAT(I,J+5) + P(I,J)
223.      140  CONTINUE
224.      170  CONTINUE
225.      IF (NPRINT.GT. 1) GO TO 180
226.      CALL PRINT

```

```

227.      180  CONTINUE
228.          IF (NPRINT .NE. 5)  GO TO 190
229.          DO 185  L = 1,NT
230.      185  WRITE(6,1001)  ITRIAL, L, NHIT(L)
231.      190  CONTINUE
232.          IF (INTSAR .EQ. 1)  CALL DAMAGE
233.          IF (ITRIAL .LT. NTPIAL)  GO TO 100
234.          IF (NTRIAL .GT. 1)  CALL STATIS
235.      200  IF ((INT .EQ. 0).AND.(INTSAR .EQ. 1))  CALL DAMAGE
236.          IF (NREDO .EQ. 1)  GO TO 4
237.          STOP
238.      1001  FORMAT(' ', 'TRIAL', I4, '  TGT', I4, '  HITS', I4)
239.          END

```

```

241.      SUBROUTINE INPUT(CASE,MWPN,NEWGT,NTSAP,NPUNCH)
242.      IMPLICIT INTEGER *2 (A-Z)
243.      INTEGER *4 DATA,LABEL,AN,WORDS,NBASE1,NBASE2,AID
244.      INTEGER *4 DIM,DIM2,MSTAT
245.      INTEGER *4 MXITEM,NOPEO,NOAGE,NOPART,NOMUN,NOTRAP,NOMATL,NOPOL
246.      INTEGER *4 PEOPLE,AGE,PARTS,AMMO,TRAP,MATERL,POL,NOPQUI,NOPPOP
247.      REAL *4 STAT
248.      C THE FOLLOWING JCL INSERTS TSAPINA'S COMMON "BASIC"
249.      // DD DSN=*.STEP1.COMMON,DISP=SHR
250.      // DD *,DCB=BLKSIZE=800
251.      C
252.      COMMON /STATS / STAT(500,17),MSTAT(8),NTRIAL,ITRIAL,NSTAT
253.      COMMON /CONTRL/ NREP,NELOT,INW,INL,CHANGE,NSAVE,LIST,NJMEM
254.      COMMON /CBUHIT/ CBUHT(200,2), IR(50), KCBU, KPTI
255.      COMMON /OUTPUT/ TSAR, PUNCH, NBASE1, NBASE2
256.      C
257.      C THE DATA STORAGE ARRAYS (AND DIMENSIONS) FOR THE RESOURCE
258.      C STORAGE LOCATION DATA ARE FILED IN LABELLED COMMON STORES.
259.      C
260.      C SEE THE MAIN ROUTINE FOR DIMENSIONING INSTRUCTIONS
261.      C
262.      COMMON /STORES/ MXITEM,NOPEO,NOAGE,NOPART,NOMUN,NOTRAP,NOMATI,
263.      X NOPOL,NOEQUI,NOPOP,PEOPLE(200,4),AGE(100,4),PARTS(400,4),
264.      X AMMO(126,4),TRAP(26,4),MATERL(26,4),POL(10,4),STOCKS(1000,3),
265.      X EQUIV(400)
266.      C
267.      COMMON / LISTER / ISAVE
268.      C
269.      DIMENSION LABEL(10), DIM(500,2), DIM2(100,2), DATA(11), WORDS(4),
270.      X CARDS(10)
271.      DATA LABEL /'TGT ','TGT2','ATT ','ATT2','END ','EQUI','CONT',
272.      X 'DATA','REDO','END ' /
273.      C
274.      C
275.      NS = 0
276.      XMIN = 0
277.      YMIN = 0
278.      NJMEM = 0
279.      LAST = 0
280.      LIST1 = 0
281.      LIST2 = 0
282.      ITEM = 0
283.      FLAG = 0
284.      IF (CASE .EQ. 1) NAMES = 2
285.      NRMAX = 0
286.      NAO = 1
287.      IF (NREDO .EQ. 0) GO TO 2
288.      C SEE NOTE AT LABEL '48'
289.      IF (NEWGT .EQ. 0) LIST1 = LIST
290.      IF (NSAVE .GT. 0) NA = NSAVE
291.      IF (NSAVE .LT. 0) NA = 0
292.      NAO = NA + 1
293.      IF (NSAVE .EQ. 0) LIST2 = LIST
294.      NSAVE = 0
295.      2 CONTINUE
296.      NREDO = 0
297.      6 READ (5,101) AN, NTYPE, (DATA(I), I=1,11), (WORDS(I), I=1, NAMES)
298.      IF (KTEST .GT. 2) WRITE (6,1101) AN, NTYPE, (DATA(I), I=1,11),
299.      X (WORDS(I), I=1, NAMES)
300.      C NO ENTRY IS REQUIRED IN COLUMNS 5 AND 6. IF AN INTEGER IS FOUND
301.      C ON AN ATTACK CARD, THE ATTACK WILL BE REPEATED SO THAT THERE WILL
302.      C BE THAT TOTAL NUMBER OF ATTACKS WITH THE STATED CHARACTERISTICS.
303.      C (ONE ATTACK IS ASSUMED IF THERE IS NO ENTRY.)
304.      C THE ENTRY IN COL 6 ON THE END CARDS AND COLUMNS 67-70 OF
305.      C THE TGT CARDS CONTROL THE NUMBER OF CARDS EXPECTED.
306.      C
307.      IF (AN .EQ. LABEL(1)) GO TO 10
308.      IF (AN .EQ. LABEL(2)) GO TO 15
309.      IF (AN .EQ. LABEL(3)) GO TO 20
310.      IF (AN .EQ. LABEL(4)) GO TO 26
311.      IF (AN .EQ. LABEL(5)) GO TO 30
312.      IF (AN .EQ. LABEL(6)) GO TO 36
313.      IF (AN .EQ. LABEL(7)) GO TO 40
314.      IF (AN .EQ. LABEL(8)) GO TO 45
315.      IF (AN .EQ. LABEL(9)) GO TO 48
316.      IF (AN .EQ. LABEL(10)) GO TO 50

```

```

317.      WRITE(6,1102) AN,TYPE,(DATA(I),I=1,11),(WORDS(I),I=1,NAMES)
318.      GO TO 6
319.      10  NT = NT + 1
320.      IF ((NT.EQ. 1).AND.(NPRINT.LE. 1))  WRITE(6,121)
321.      IF (NEWTGT.EQ. 0)  GO TO 170
322.      LIST1 = 0
323.      IF (NT.GT. NTM)  GO TO 120
324.      DO 11 I = 1,2
325.      11  DIM(NT,I) = DATA(I)
326.      IF (DATA(1).LT. XMIN)  XMIN = DATA(1)
327.      IF (DATA(2).LT. YMIN)  YMIN = DATA(2)
328.      DO 13 I = 3,4
329.      13  TGT(NT,I+9) = DATA(I)
330.      DO 131 I = 5,7
331.      131 TGT(NT,I + 4) = DATA(I)
332.      TGT(NT,14) = DATA(8)
333.      IF (DATA(6).NE. SHEL)  GO TO 133
334.      NS = NS + 1
335.      TGT(NT,14) = 1000 + NS
336.      SHEL(NS) = NT
337.      133 IF ((DATA(8).GT.C).AND.(DATA(3).LE.1000))  FACILY(DATA(8)) = NT
338.      DO 14 I = 1,NAMES
339.      14  NAME(NT,I) = WORDS(I)
340.      IF ((DATA(9)+DATA(11)).EQ. 0)  GO TO 142
341.      IF (NPRINT.LE. 1)  WRITE(6,122) NT, (NAME(NT,L),L=1,4),
342.      X      DATA(1), DATA(2)
343.      ITEM = ITEM + 1
344.      IF (ITEM.GT. MXITEM)  GO TO 160
345.      NCARDS = DATA(11)/100
346.      IF (NCARDS.EQ. 0)  GO TO 141
347.      C
348.      C  SUBROUTINE STOPE ORGANIZES RESOURCE DATA FOR
349.      C  FACILITIES WITH SEVERAL ITEMS.
350.      C
351.      C      CALL STORE( ITEM, NCARDS )
352.      C
353.      C      GO TO 142
354.      C
355.      C  WITH ONLY ONE CLASS AND ONE TYPE (OR ALL TYPES OF A CLASS),
356.      C  ENTRY FILE IN SUBROUTINE STORE IS USED.
357.      C
358.      141  STOCKS(ITEM,1) = NT
359.      STOCKS(ITEM,2) = 1000
360.      C
361.      C      CALL FILE(ITEM,DATA(9),DATA(10) )
362.      C
363.      142  CONTINUE
364.      GO TO 6
365.      15  NT2 = NT2 + 1
366.      IF (NEWTGT.EQ. 0)  GO TO 170
367.      IF (NT2.GT. 100)  GO TO 120
368.      DO 16 I = 1,2
369.      16  DIM2(NT2,I) = DATA(I)
370.      IF (DATA(1).LT. XMIN)  XMIN = DATA(1)
371.      IF (DATA(2).LT. YMIN)  YMIN = DATA(2)
372.      DO 17 I = 3,4
373.      17  TGT2(NT2,I+9) = DATA(I)
374.      DO 18 I = 5,7
375.      18  TGT2(NT2,I + 4) = DATA(I)
376.      GO TO 6
377.      20  NA = NA + 1
378.      LIST2 = 0
379.      IF ( NA.GT. NAM)  GO TO 130
380.      DO 22 I = 1,6
381.      22  ATT(NA,I) = DATA(I)
382.      ATT(NA,10) = DATA(7)
383.      IF (DATA(7).EQ. 0)  ATT(NA,10) = ATT(NA,6)
384.      IF (DATA(11).EQ. 0)  DATA(11) = 100
385.      ATT(NA,11) = DATA(11)
386.      DO 24 I = 7,9
387.      24  ATT(NA,I) = DATA(I+1)
388.      NTYPE = MAX0((NTYPE-1),0)
389.      IF (NTYPE.EQ. 0)  GO TO 6
390.      GO TO 20
391.      26  NA = NA + 1
392.      LIST2 = 0

```

```

393.      IF ( NA .GT. NAM)   GO TO 130
394.      IF (DATA(11) .EQ. 0) DATA(11) = 1.0
395.      CALL JHEMC (NJHEM,DATA,NA,KTEST)
396.      DO 27 I = 1,11
397. 27 ATT(NA,I) = DATA(I)
398. 28 NTYPE = MAXO((NTYPE-1),0)
399.      IF (NTYPE .EQ. 0) GO TO 6
400.      NA = NA + 1
401.      DO 29 I = 1,11
402. 29 ATT(NA,I) = ATT((NA-1),I)
403.      GO TO 28
404. 30 ND = ND + 1
405.      NR = NTYPE
406.      IF (NR .EQ. 0) NR = 1
407.      IF (NR .GT. NRMAX) NRMAX = NR
408.      CBU = 0
409.      IF (DATA(2) .LT. 0) CBU = 1
410.      M = DATA(1)/1000
411.      IF (M .GT. 10) GO TO 140
412.      IF (M .GT. MWPEN) MWPEN = M
413.      AID = DATA(1) - 1000*M
414.      IF (AID .EQ. 0) AID = 100
415.      WPNREL(M) = AID/100.
416.      MTYPE(M) = NR
417.      CARDS(M) = TT
418.      DO 32 N = 2,11
419. 32 EMD(M,N-1,1) = DATA(N)
420.      DO 34 NC = 2, NR
421. 34 READ(5,114) (EMD(M,N,NC), N = 1,10)
422.      IF (TT .EQ. 1) GO TO 6
423.      NF = CBU + 1
424.      DO 35 NC = NF, NR
425. 35 READ(5,114) (EMD(M,N,NC), N = 11,20)
426.      GO TO 6
427. C
428. C   STORE RESOURCE EQUIVALENCE DATA
429. C
430. 36 DO 39 I = 1,11
431.      IF ((DATA(I) .EQ. 0).AND.(FLAG .GT. 0)) GO TO 39
432.      IF (DATA(I) .GT. 5000) GO TO 38
433.      IF (TYPE .EQ. 0) GO TO 37
434. C
435. C   CALL SAVE(CLASS, TYPE, DATA(I))
436. C
437.      FLAG = FLAG + 1
438.      GO TO 39
439. 37 TYPE = DATA(I)
440.      GO TO 39
441. 38 CLASS = DATA(I) - 5000
442.      TYPE = 0
443.      FLAG = 0
444. 39 CONTINUE
445.      GO TO 6
446. C
447. C   TSAR INPUTS ARE PREPARED AND LISTED IF INTSAR IS SET TO UNITY.
448. C
449. 40 INTSAR = NTYPE
450.      MODE = DATA(1)/1000
451.      TTGT = DATA(1) - 1000*MODE
452.      IF (TTGT .LT. 0) TTGT = -TTGT
453.      IF (TTGT .LE. 10) TT = 1
454.      IF (TTGT .GT. 10) TT = 2
455.      AID = DATA(2)
456.      NTRIAL = AID/1000
457.      AID = AID - 1000*NTRIAL
458.      IF (NTRIAL .LT. 2) NTRIAL = 1
459.      IF (AID .GT. 0) NAMES = 4
460.      NPRINT = DATA(3)/1000
461.      PDAM = DATA(3) - 1000*NPRINT
462.      IF (PDAM .LT. 0) PDAM = -PDAM
463.      KTEST = DATA(4)
464.      CALL SAVER(C,KTEST,MAXLOC)
465.      MCR = 0
466.      IF (DATA(5) .GT. 0) MCR = 1
467.      MCL = DATA(5)
468.      MCW = DATA(6)

```

```

469.      NREP = DATA(7)/1000
470.      NPLOT= DATA(7) - 1000*NREP
471.      INW = DATA(8)
472.      INL = DATA(9)
473.      IF (INW .EQ. 0)   INW = 5
474.      IF (INL .EQ. 0)   INL = 250
475.      NBASE1 = WORDS(1)
476.      NBASE2 = WORDS(2)
477.      GO TO 6
478.  C
479.  C      THE DATA CARD ONLY NEED BE USED WHEN OUTPUT IS TO BE USED WITH TSAP
480.  C
481.  C      IF THERE IS AN ENTRY IN COLUMNS 5-6 FOR NTYPE (ISAVE),
482.  C      THE LOSSES FOR EACH SUBCATEGORY OF RESOURCES ARE
483.  C      PUNCHED IN CARD FORMAT SO AS TO BE READ DIRECTLY BY
484.  C      THE "FORMATER" AUXILIARY PROGRAM AND PROCESSED TO A
485.  C      FORM THAT IS USABLE DIRECTLY IN THE "PREPARE RESULTS"
486.  C      AUXILIARY PROGRAM.
487.  C
488.  C      ALL RELATED CARDS ARE IDENTIFIED
489.  C      BY A *** IN COLUMNS 73-75.
490.  C
491.  45  CONTINUE
492.      ISAVE = NTYPE
493.      TSAP = DATA(1)
494.      NTSAR = TSAR
495.      IF (NTSAR .GT. 0)   TSAR = 1
496.      PUNCH = DATA(2)
497.      NPUNCH = PUNCH
498.      IF (NPUNCH .GT. 0)   PUNCH = 1
499.      BASE = DATA(3)
500.      DAY = DATA(4)
501.      HOUR = DATA(5)
502.      MINUTE = DATA(6)
503.      SHELT = DATA(7)
504.      TKWYS = DATA(8)
505.      RAMPS = DATA(9)
506.  C
507.  C      ACLOSS IS THE EXPECTED LOSS-TO-DAMAGE PERCENTAGE FOR AIRCRAFT.
508.  C      (SEE SUBROUTINE DAMAGE FOR APPLICATION)
509.  C
510.      ACLOSS = DATA(10)
511.  C
512.      IF (ISAVE .EQ. 0)   GO TO 6
513.      N99 = 99
514.      WRITE(7,1111) N99,BASE,ISAVE
515.      FORMAT(13,2X,I2,5X,I2)
516.  C
517.      GO TO 6
518.  48  NREDO = 1
519.  C
520.  C      IF THE FIRST DATA ENTRY ON THE REDO CARD (COL 12) IS "1",
521.  C      ALL THE TARGETS ARE TO BE CHANGED FOR THE NEXT CASE.
522.  C
523.      CHANGE = DATA(1)
524.  C
525.  C      THE SECOND DATA ENTRY ON THE REDO CARD (COL 18) MAY BE USED
526.  C      TO SPECIFY THE NUMBER OF PRIOR ATTACKS TO BE INCLUDED IN
527.  C      THE NEW CALCULATION. THAT NUMBER WILL BE SELECTED IN
528.  C      RANK ORDER FROM THOSE INPUT PREVIOUSLY; IN NO CASE MAY THE
529.  C      NUMBER BE LARGER THAN THE NUMBER AVAILABLE. IF NO NUMBER
530.  C      IS ENTERED, ALL PRIOR ATTACKS WILL BE INCLUDED. IF A
531.  C      NEGATIVE NUMBER IS ENTERED (EG -1), NONE OF THE PRIOR
532.  C      ATTACKS WILL BE TREATED. NEW ATTACKS ARE REQUIRED
533.  C      IF NEW TARGETS ARE ENTERED.
534.  C
535.      NSAVE = DATA(2)
536.      IF (CHANGE .GT. 0)   NSAVE = -1
537.  C
538.  C      IF THE THIRD ENTRY (COL 24) IS SET TO UNITY, THE TARGET LIST
539.  C      AND/OR THE ATTACK/WEAPON LISTS WILL BE SUPPRESSED IF THE
540.  C      LATTER HAVE NOT BEEN CHANGED.
541.  C
542.      LIST = DATA(3)
543.  50  CONTINUE
544.  C

```

```

545. C PLACE A "-1" IN THE TENTH ROW OF EACH COLUMN FOR THOSE TARGET
546. C TYPES FOR WHICH THE EFFECTIVENESS WITH INTERNAL AND EXTERNAL HITS
547. C IS NOT TO BE DISTINGUISHED.
548. C
549. LMT = 10*TT
550. DO 54 M = 1,10
551. IF (MTYPE(M) .EQ. 0) GO TO 54
552. DO 54 N = 1,LMT
553. IF (M .EQ. SHELTT) GO TO 54
554. IF (NRMAX .LE. 9) GO TO 53
555. DO 51 I = 1,NRMAX
556. ROW = NRMAX - I + 1
557. IF (EMD(M,N,ROW) .NE. 0) GO TO 52
558. IF (ROW .EQ. 9) GO TO 53
559. 51 CONTINUE
560. 52 IF (ROW .EQ. 9) EMD(M,N,10) = -1
561. GO TO 54
562. 53 EMD(M,N,10) = -1
563. 54 CONTINUE
564. C
565. C NOTE WHEN SHELTERS ARE TO BE HANDLED WITH THE MONTE
566. C CARLO MODE RATHER THAN THE EXPECTED VALUE MODE.
567. C
568. ALLMC = 0
569. IF ((SHELTT .GT. 0) .AND. (NT2 .EQ. 0)) ALLMC = 1
570. NSM = NS
571. C
572. CALL SAVER(1,KTEST,MAXLOC)
573. C
574. WRITE(6,128) ITEM, MAXLOC
575. C
576. C PRINT OUTPUT HEADING BLOCK
577. C
578. IF (NREDO .EQ. 0) LAST = 1
579. IF (MODE .EQ. 0) CALL RSTART(7)
580. WRITE(6,111) NTRIAL, NPRINT, PDAM, MODE, MCL, INL, MCW, INW,
581. X NREP, NPLOT, KTEST, CASE, BASE, DAY, HOUR, MINUTE
582. IF ((NBASE1 .EQ. 0) .AND. (NBASE2 .EQ. 0)) GO TO 55
583. WRITE(6,100) NBASE1, NBASE2
584. 55 CONTINUE
585. IF (NPRINT .GT. 0) WRITE(6,129) NT, NA
586. IF (NPRINT .LE. 0) WRITE(6,102)
587. IF (NEWTGT .EQ. 0) GO TO 65
588. C
589. C TO FACILITATE THE PROCEDURE OUTLINED IN SUBROUTINE TGTZON,
590. C THE X-Y COORDINATE SYSTEM IS TRANSLATED SO THAT ALL TARGETS
591. C ARE IN THE FIRST QUADRANT, WHEN NECESSARY.
592. C
593. XM = (XMIN - 999)/1000
594. YM = (YMIN - 999)/1000
595. XM = - 1000*XM
596. YM = - 1000*YM
597. XMAX = 32000 - XM
598. YMAX = 32000 - YM
599. 56 IF ((XM+YM) .GT. 0) WRITE(6,123) XM, YM
600. MTT = 0
601. IF (NT .EQ. 0) GO TO 63
602. DO 59 I = 1,NT
603. IF (TGT(I,10) .GT. MTT) MTT = TGT(I,10)
604. 59 CONTINUE
605. DO 62 M = 1, MTT
606. FLAG = 0
607. DO 62 I = 1,NT
608. IF (TGT(I,10) .NE. M) GO TO 62
609. IF (FLAG .EQ. 1) GO TO 60
610. IF (NPRINT .LE. 0) WRITE(6,112) M
611. FLAG = 1
612. 60 IF ((DIM(I,1) .LE. XMAX) .AND. (DIM(I,2) .LE. YMAX)) GO TO 61
613. C
614. C IF A TARGET IS OUTSIDE THE ALLOWED 32000 X 32000 AREA, IT
615. C IS SHIFTED TO THE EDGE OF THAT AREA AND THE USER IS NOTIFIED.
616. C
617. IF (DIM(I,1) .GT. XMAX) DIM(I,1) = XMAX
618. IF (DIM(I,2) .GT. YMAX) DIM(I,2) = YMAX
619. WRITE(6,124) I
620. 61 TGT(I,1) = DIM(I,1) + XM
621. TGT(I,2) = DIM(I,2) + YM

```

```

622.      IF (NPRINT.LE.0)      WRITE (6,112) I, (IGT(I,J),J=1,2),
623.      X      (IGT(I,J),J=12,13), (TGT(I,J),J=9,11), TGT(I,14),
624.      X      (NAME(I,L),L=1,NAMES)
625.      62      CONTINUE
626.      IF (LIST1.EQ. 1)      GO TO 65
627.      63      IF (NT2.EQ. 0)      GO TO 65
628.      WRITE(6,115)
629.      DO 64 I = 1,NT2
630.      IGT2(I,1) = DIM2(I,1) + XM
631.      TGT2(I,2) = DIM2(I,2) + YM
632.      64      WRITE (6,113) I, (TGT2(I,J),J=1,2), (TGT2(I,J),J=12,13),
633.      X      (TGT2(I,J),J=9,11)
634.      C
635.      65      NEWTGT = 0
636.      IF (LIST2.EQ. 1)      GO TO 95
637.      WRITE (6,104)
638.      KPTI = 0
639.      DO 68 I = NA0, NA
640.      ATT(I,2) = ATT(I,2) + XM
641.      68      ATT(I,3) = ATT(I,3) + YM
642.      DO 70 I = 1, NA
643.      IF (EMD(ATT(I,9),1,1).GE. 0)      KPTI = 1
644.      70      WRITE (6,103) I, (ATT(I,J),J=1,6), ATT(I,10), (ATT(I,J),J=7,9)
645.      X      , ATT(I,11)
646.      IF ((ND.EQ. 0).OR.(NPRINT.GT. 0))      GO TO 95
647.      WRITE (6,105) (I, I=1,10)
648.      DO 90 I = 1, MWPN
649.      NR = NTYPE(I)
650.      IF (NR.EQ. 0)      GO TO 90
651.      WRITE(6,116) I, WPNSEL(I), (EMD(I,J,1),J=1,10)
652.      WRITE (6,107) ((EMD(I,J,K), J = 1,10), K=2,NR)
653.      90      CONTINUE
654.      IF (CARDS(I).EQ. 1)      GO TO 90
655.      CBU = 0
656.      IF (EMD(I,1,1).LT. 0)      CBU = 1
657.      NP = CBU + 1
658.      WRITE(6,127)
659.      WRITE(6,107) ((EMD(I,J,K), J=11,20),K=NP,NR)
660.      90      CONTINUE
661.      95      CONTINUE
662.      C
663.      C      TARGET TYPE #1 IS RESERVED FOR RUNWAYS AND TAXIWAYS (OR OTHER
664.      C      LARGE TARGETS IF MCR=0) AND HIT STORAGE IS PROVIDED FOR 250 HITS
665.      C      BUT FOR A MAXIMUM OF FIVE TARGETS OF TYPE #1.
666.      C
667.      PTX = 0
668.      DO 99 I = 1, NT
669.      IF (IGT(I,10).NE. 1.)      GO TO 99
670.      NIX = NTX + 1
671.      IF (NIX.GT. 5)      GO TO 150
672.      NRW(NTX) = I
673.      99      CONTINUE
674.      IF (KTEST.LT. 8)      RETURN
675.      WRITE(6,112) NAMES
676.      WRITE(6,118) ((STOCKS(I,J),I=1,30),J=1,3)
677.      WRITE(6,118) ((PEOPLE(I,J),I=1,30),J=1,4,3)
678.      WRITE(6,118) ((PEOPLE(I,J),I=102,131),J=1,4,3)
679.      WRITE(6,118) (AGE(I,1),I=1,30)
680.      WRITE(6,118) (PARTS(I,1),I=1,30)
681.      WRITE(6,118) (AMMO(I,1),I=1,26)
682.      WRITE(6,118) (TRAP(I,1),I=1,26)
683.      WRITE(6,118) (MATERL(I,1),I=1,26)
684.      WRITE(6,118) (POL(I,1),I=1,10)
685.      WRITE(6,118) (EQUIV(I),I=1,60)
686.      IF (KTEST.GT.15)      STOP
687.      RETURN
688.      C
689.      120      WRITE (6,108)
690.      STOP
691.      130      WRITE (6,109)
692.      STOP
693.      140      WRITE (6,110)
694.      STOP
695.      150      WRITE (6,116)
696.      STOP
697.      160      WRITE(6,117)
698.      STOP

```

```

699.      170      WRITE(6,119)
700.      STOP
701.  C
702.      100      FORMAT(' ',40X,'***** BASE COMPLEX NAME - ',2A4,' *****',//)
703.      101      FORMAT( A4,I2,11I6,4A4)
704.      102      FORMAT(' ', 20X,'TARGET DATA',//,'      NUMBER      X-DIM      Y-DI
705.      XM      NE LIMB      SE LIMB      ANGLE TGT TYPE      STORF      BLDG NO',
706.      X      // )
707.      103      FORMAT(' ',I4,2X,11I10 )
708.      104      FORMAT(' ',//, 20X,'ATTACK DATA',//,'      NUMBEP      HDG      X-DMPI
709.      X      Y-DMPI      REP      DEP      R-DISP      D-DISP      NO WPNS      LENG
710.      XTH WPN TYPE      ARRIVAL',//)
711.      105      FORMAT('1', 35X,'MISS DISTANCE AND KILL PROBABILITY DATA',//,
712.      X50X,'TARGET TYPES',//,20X,10(6X,I2,2X),//,' WPN TYPE      WPN PEL ',
713.      X//,//)
714.      106      FORMAT('0',I6,7X,F5.3,10I10)
715.      107      FORMAT(' ', 18X,10I10)
716.      108      FORMAT(' ',//,'***** TOO MANY TARGETS HAVE BEEN SPECIFIED *****')
717.      109      FORMAT(' ',//,'***** TOO MANY ATTACKS HAVE BEEN SPECIFIED *****')
718.      110      FORMAT(' ',//,'***** TOO MANY TYPES OF WEAPONS HAVE BEEN ',
719.      X      'SPECIFIED *****')
720.      111      FORMAT('1', 130(' '),//,
721.      X      40X,'*** TSARINA *** DAMAGE ASSESSMENTS FOR TSAP',//,
722.      X      37X,'BASED ON THE AIDA MODEL OF AIRBASE DAMAGE ASSESSMENT',//,
723.      X      39X,'DEVELOPED BY D. E. EMERSON AT THE RAND CORPORATION',//,
724.      X      130(' '),//,' NO OF TRIALS ', I3,' NPRINT ',I2,' DAMAGE ',I2,
725.      X      ' MODE ',I2,
726.      X      ' MCL ',I6,'(',I4,')      MCW ',I4,'(',I2,')      MIN REPAIR ',
727.      X      I1,' PLOT HITS ',I1,' TEST ',I2,/,130(' '),//,55X,'CASE ',
728.      X      I3,/,20X,'IN THE TSAR SIMULATION, THIS ATTACK AND DAMAGE ',
729.      X      'OCCUR AT BASE #',I2,' ON DAY',I3,' AT ',I2,':',I2,//)
730.      112      FORMAT(' ', I4,4X,8I10,2X,4A4)
731.      113      FORMAT(' ', I4,4X,7I10)
732.      114      FORMAT( 12X, 10I6 )
733.      115      FORMAT('0',20X,'***** SHELTERS ***** ',// )
734.      116      FORMAT(' ',//,'***** TOO MANY RUNWAYS/TAXIWAYS HAVE BEEN',
735.      X      ' SPECIFIED *****')
736.      117      FORMAT(' ',//,'***** THERE ARE TOO MANY FACILITY CONTENTS ',
737.      X      'DESCRIPTORS *****')
738.      118      FORMAT(' ',30I4,/,', ',30I4)
739.      119      FORMAT('0','TARGETS MAY NOT BE CHANGED FROM CASE TO CASE' )
740.      121      FORMAT('0',10X,'RESOURCE STORAGE DATA',//,' ', 'TARGET NUMBER',
741.      X      5X,'NAME' )
742.      122      FORMAT(' ', 6X, I4, 8X, 4A4, 2I7 )
743.      123      FORMAT(' ', 'ALL TARGET LOCATION DIMENSIONS WERE INCREASED BY',
744.      X      I6,' IN THE X-DIMENSION AND ',I6,' IN THE Y-DIMENSION',//)
745.      124      FORMAT('0',' ONE OR BOTH DIMENSIONS OF TARGET ',I4,' WERE ',
746.      X      'MODIFIED TO PLACE THE TARGET AT THE EDGE OF THE ALLOWED AREA')
747.      127      FORMAT('0' )
748.      128      FORMAT('0','STORAGE OF THE RESOURCE LOCATIONS REQUIRED ',I4,
749.      X      ' LOCATIONS IN THE STOCKS ARRAY',//,' ', 'AND THE EQUIVALENCE',
750.      X      ' DATA USED ',I4,' ELEMENTS OF THE EQUIV ARRAY',// )
751.      129      FORMAT('0','DATA WERE ENTERED FOR ',I4,' TARGETS AND ',I3,
752.      X      ' ATTACKS',//)
753.      132      FORMAT('0', 25X,'** TARGET TYPE # ',I3,' **',//)
754.      1101      FORMAT(' ',A4,I4,11I8,4A4)
755.      1102      FORMAT(' ', 'UNIDENTIFIED CARD IMAGE: ',A4,I2,11I6,4A4 )
756.      END

```

```

758.      SUBROUTINE TGTDIM
759.      IMPLICIT INTEGER *2 (A-Z)
760.      INTEGER *4 MSTAT
761.      REAL *4 THETA, S, C, SIN, COS, STAT
762.      C THE FOLLOWING JCL INSERTS TSARINA'S COMMON "BASIC"
763.      // DD DSN=*,STEP1.COMMON,DISP=SHR
764.      // DD *,DCB=BLKSIZE=800
765.      COMMON /STATS / STAT(500,17),MSTAT(8),NTRIAL,ITRIAL,NSTAT
766.      IF (NPRINT .LT. 0) WRITE (6,104)
767.      IF (NT .EQ. 0) GO TO 22
768.      DO 20 I = 1, NT
769.      C TGTDIM COMPUTES AND STORES THE LOCATION OF THE OTHER CORNERS.
770.      L1 = TGT(I,12)
771.      L2 = TGT(I,13)
772.      THETA = TGT(I, 9)/57.3
773.      S = SIN (THETA)
774.      C = COS (THETA)
775.      L1S = L1*S
776.      L1C = L1*C
777.      L2S = L2*S
778.      L2C = L2*C
779.      TGT(I,3) = TGT(I, 1) + L1S
780.      TGT(I,4) = TGT(I, 2) + L1C
781.      TGT(I,5) = TGT(I, 3) + L2C
782.      TGT(I,6) = TGT(I, 4) - L2S
783.      TGT(I,7) = TGT(I, 5) - L1S
784.      TGT(I,8) = TGT(I, 6) - L1C
785.      IF ((KTEST .GT. 2) .OR. (NPRINT .LT. 0))
786.      X WRITE (6,102) I, (TGT(I,K), K=1,8)
787.      20 CONTINUE
788.      22 IF (NT2 .EQ. 0) GO TO 28
789.      DO 26 I = 1, NT2
790.      C TGT2DIM COMPUTES AND STORES THE LOCATION OF THE OTHER CORNERS.
791.      L1 = TGT2(I,12)
792.      L2 = TGT2(I,13)
793.      THETA = TGT2(I, 9)/57.3
794.      S = SIN (THETA)
795.      C = COS (THETA)
796.      L1S = L1*S
797.      L1C = L1*C
798.      L2S = L2*S
799.      L2C = L2*C
800.      TGT2(I,3) = TGT2(I, 1) + L1S
801.      TGT2(I,4) = TGT2(I, 2) + L1C
802.      TGT2(I,5) = TGT2(I, 3) + L2C
803.      TGT2(I,6) = TGT2(I, 4) - L2S
804.      TGT2(I,7) = TGT2(I, 5) - L1S
805.      TGT2(I,8) = TGT2(I, 6) - L1C
806.      IF ((KTEST .GT. 2) .OR. (NPRINT .LT. 0))
807.      X WRITE (6,102) I, (TGT2(I,K), K=1,8)
808.      26 CONTINUE
809.      28 IF (NT .EQ. 0) GO TO 50
810.      NR = 0
811.      DO 30 I = 1, NT
812.      C FOR SPECIFIED TARGETS THE TARGET NUMBER IS STORED IN MHIT FOR
813.      C LATER REFERENCE.
814.      IF ((TGT(I,11) .LT. 1) .OR. (TGT(I,10) .EQ. 1)) GO TO 30
815.      NR = NR + 1
816.      IF (NR .GT. NST) GO TO 80
817.      MHIT(NR) = I
818.      IF ((KTEST .GT. 4) .AND. (ITRIAL .LT. 2))
819.      X WRITE (6,101) NR, MHIT(NR)
820.      30 CONTINUE
821.      IF (NR .EQ. NST) GO TO 50
822.      NR1 = NR+1
823.      DO 40 I = NR1, NST
824.      40 MHIT(I) = 0
825.      50 CONTINUE
826.      IF ((NPRINT .LT. 0) .OR. (KTEST .GT. 2)) WRITE(6,104)
827.      RETURN
828.      80 WRITE (6,103)
829.      STOP
830.      101 FORMAT(' ', 'MHIT(' , I2, ') = ', I2)
831.      102 FORMAT(' ', 'TARGET CORNER : TGT # ', I4, ' (4X, I6, 1X, I6)')
832.      103 FORMAT('0', ' COMPUTATION STOPPED: HIT DATA SPACE ',
833.      X 'REQUIRED FOR MORE THAN "NST" TARGETS')
834.      104 FORMAT('1')
835.      END

```

```

837.          SUBROUTINE TGTORD
838.          IMPLICIT INTEGER *2 (A-Z)
839.          C   THE FOLLOWING JCL INSERTS TSARINA'S COMMON "BASIC"
840.          //      DD      DSN=*.STEP1.COMMON,DISP=SHR
841.          //      DD      *,DCB=BLKSIZE=800
842.          C   THIS ROUTINE CREATES AN ARRAY IN WHICH THE TARGET NUMBERS ARE
843.          C   ORDERED ACCORDING TO INCREASING VALUES OF (X+Y) OF THE INDEX COPNPP
844.          C   OUTPUT VALUES : TO(I,1) IS THE (X+Y) AND TO(I,2) IS THE 'ORIGINAL'
845.          C   TARGET NUMBER; I.E. ITS POSITION IN THE INPUT LIST
846.          DO 10 I=1,NT
847.          C   INITIALIZES TO (TARGET ORDER) ARRAY
848.          IF (TGT(I,9) .GT. 45) GO TO 5
849.          TO(I,1)=TGT(I,1)+TGT(I,2)
850.          GO TO 8
851.          5   TO(I,1)=TGT(I,7)+TGT(I,8)
852.          8   TO(I,2)=I
853.          10  CONTINUE
854.          IF (NT .EQ. 1) GO TO 25
855.          DO 20 J=2,NT
856.          NTEST=0
857.          DO 15 K=2,NT
858.          C   REORGANIZES THE TO ARRAY INTO INCREASING VALUES OF THE INDEX COPNPP
859.          I=NT-K+2
860.          IF (TO(I,1) .GE. TO(I-1,1)) GO TO 15
861.          NTEST=1
862.          T=TO(I-1,1)
863.          TN=TO(I-1,2)
864.          TO(I-1,1)=TO(I,1)
865.          TO(I-1,2)=TO(I,2)
866.          TO(I,1)=T
867.          TO(I,2)=TN
868.          15  CONTINUE
869.          IF (NTEST .EQ. 0) GO TO 25
870.          20  CONTINUE
871.          25  CONTINUE
872.          IF (KTEST .LT. 3) GO TO 40
873.          DO 30 I = 1, NT
874.          NTO = TO(I,2)
875.          30  IF (KTEST .GT. 5) WRITE (6,101) I , NTO
876.          40  CONTINUE
877.          NT1 = NT + 1
878.          DO 50 I = NT1, NTM
879.          DO 50 J = 1,2
880.          TO(I,J) = 0.0
881.          RETURN
882.          101 FORMAT (' ', ' RANK ',I3,' TARGET # ', I3)
883.          END

```

```

885.          SUBROUTINE TGTZON
886.          IMPLICIT INTEGER *2 (A-Z)
887.          C   THE FOLLOWING JCL INSERTS TSARINA'S COMMON "BASIC"
888.          //      DD      DSN=*.STEP1.COMMON,DISP=SHR
889.          //      DD      *,DCB=BLKSIZE=800
890.          C
891.          C   TGTZON IDENTIFIES TARGET LOCATION IN TERMS OF ITS IZONE SO THAT THE
892.          C   SUBSEQUENT SEARCH PROCESS CAN BE REDUCED.  CONSIDER THE ENTIRE
893.          C   TARGET AREA MAPPED BY LINES OF CONSTANT (X+Y).  ALL TARGETS WITH '
894.          C   'INDEX CORNER' (X+Y) FALLING INTO THE K-TH 500 FOOT SEGMENT OF
895.          C   (X+Y) ARE IN THE K-TH ZONE.  THE ORDERED INDEX NUMBER FOR THE
896.          C   TARGET WITH THE LOWEST (X+Y) IN THE ZONE IS IZONE(K,1); THAT WITH
897.          C   THE HIGHEST, IS IZONE(K,2).  IF THERE ARE NO TARGETS IN A ZONE, THE
898.          C   IZONE VALUES ARE BOTH EQUAL TO THE INDEX NUMBER OF THE LAST TARGET
899.          C   AS IZONE(K-1,2)).
900.          C
901.          C   TO FACILITATE THIS PROCEDURE THE TARGET COORDINATE SYSTEM IS
902.          C   TRANSLATED, WHEN NECESSARY, SO THAT ALL TARGETS ARE IN THE
903.          C   FIRST QUADRANT.  IF ANY TARGET COORDINATE EXCEEDS 32000, THE
904.          C   TARGET IS PLACED AT THE EDGE OF THE ALLOWED 32000 X 32000 AREA,
905.          C   AND THE USER IS NOTIFIED.
906.          C
907.          DO 10 I=1,NT
908.          10  IZ(I) = TO(I,1)/500
909.          IZONE(1,1) = 0
910.          IZONE(1,2) = 0
911.          IF (IZ(1) .NE. 0) GO TO 14
912.          K = 1
913.          IZONE(1,1) = 1
914.          IZONE(1,2) = 1
915.          IF (NT .EQ. 1) GO TO 31
916.          DO 12 I = 2,NT
917.          IF (IZ(I) .GT. 0) GO TO 22
918.          12  IZONE(1,2) = I
919.          C   ALL HITS IN ZONE #1 HERE.
920.          GO TO 30
921.          14  DO 16 K = 2,100
922.          IF (IZ(1) .EQ. (K-1)) GO TO 18
923.          IZONE(K,1) = 0
924.          16  IZONE(K,2) = 0
925.          18  CONTINUE
926.          IZONE(K,1) = 1
927.          IZONE(K-1,2) = 1
928.          IZONE(K,2) = 1
929.          C   AT THIS POINT K IS ZONE OF FIRST HIT
930.          IF (NT .EQ. 1) GO TO 31
931.          DO 20 I = 2,NT
932.          IF (IZ(I) .GT. (K-1)) GO TO 22
933.          20  IZONE(K,2) = I
934.          C   ON TRANSFER TO '22' K IS FIRST OCCUPIED ZONE AND I IS FIRST HIT
935.          C   IN (K+1) ZONE.
936.          22  CONTINUE
937.          N = I
938.          DO 28 I = N,NT
939.          C   SKIP TO 26 IF HIT IN ZONE OF PRIOR HIT
940.          IF (IZ(I) .EQ. IZ(I-1)) GO TO 26
941.          24  K = K+1
942.          IZONE(K,1) = I-1
943.          IZONE(K,2) = I-1
944.          C   IF NO HITS IN ZONE INCREMENT ZONE
945.          IF (IZ(I) .GT. (K-1)) GO TO 24
946.          IZONE(K,1) = I
947.          IZONE(K,2) = I
948.          GO TO 28
949.          C   INCREMENT UPPER HIT IN ZONE
950.          26  IZONE(K,2) = I
951.          28  CONTINUE
952.          30  CONTINUE
953.          IF ((K+1) .GT. 100) GO TO 35
954.          C   FILL ALL EXCESS ZONES
955.          31  K1 = K + 1
956.          DO 32 L = K1,100
957.          IZONE(L,1) = NT + 1
958.          32  IZONE(L,2) = NT + 1
959.          36  CONTINUE
960.          IF (KTEST .LT. 3) GO TO 50
961.          WRITE (6,101)

```

```

962.      DO 40 K = 1, 100
963.      40 WRITE (6,102) K, IZONE(K,1), IZONE(K,2), TO(IZONE(K,1),2),
964.      X TO(IZONE(K,2),2)
965.      50 CONTINUE
966.      RETURN
967.      101 FORMAT ('1', 'TARGETS BY ZONE',/,
968.      X'ZONE LOWER UPPER (LOWER UPPER)')
969.      102 FORMAT (' ', I4,4X,I3,4X,I3,7X,I4,3X,I4)
970.      END

```

```

972.          SUBROUTINE BOMB
973.          IMPLICIT INTEGER *2 (A-Z)
974.          REAL*4  RAN, PHI, S, C, SIGR, SIGD, SIGX, SIGY, BDGZX, BDGZY, D, DX, DY,
975.          X       PRRF, DERR, X, Y, SIN, COS
976.          C      THE FOLLOWING JCL INSEPTS TSAPINA'S COMMON "BASIC"
977.          //      DD      DSN=*.STEP1.COMMON, DISP=SHR
978.          //      DD      *,DCB=BLKSIZE=800
979.          COMMON /CBUHT/ CBUHT(200,2), IR(50), KCBU, KFTI
980.          IF (KTEST.GT. 2)  WRITE (6,102)
981.          NCBUHT = 0
982.          KCBU   = 0
983.          DO 40 I=1,NA
984.          IR(I) = 0
985.          NW = ATT(1,9)
986.          EMDW = EMD(NW,1,1)
987.          IF ((MODE.LT. 0) .OR. (ATT(1,11) .EQ. 100))  GO TO 3
988.          C      IF (MODE.EQ. 0)  GO TO 1
989.          C      RN = 1 + 100*RANDT(1.)
990.          C      GO TO 2
991.          C 1  RN = 1 + 100*PAND(1.)
992.          RN = 1 + 100*PAN(1)
993.          2  IF (RN.LE. ATT(1,11))  GO TO 3
994.          IR(I) = 1
995.          GO TO 40
996.          3  CONTINUE
997.          IF (EMDW.LT. 0.0)  KCBU = 1
998.          DY=0
999.          DX=0
1000.          NW=ATT(1,7)
1001.          SIGR=ATT(1,1)/57.7
1002.          SIN=PHI
1003.          COS=PHI
1004.          EMD=1.48*ATT(1,4)
1005.          SIGD=1.48*ATT(1,5)
1006.          CALL GAUSS(SIGR, DERR, KTEST, MODE)
1007.          CALL GAUSS(SIGD, DERR, KTEST, MODE)
1008.          ASZX=ATT(1,2)*EMD*SIGR*DEFF*C
1009.          ASZY=ATT(1,3)*EMD*SIGD*DEFF*C
1010.          BDGZX=ASZX-S*ATT(1,8)/2.
1011.          BDGZY=ASZY-C*ATT(1,8)/2.
1012.          IF (NS.LT. 2)  GO TO 10
1013.          D=ATT(1,8)/(NS-1)
1014.          DX=S*D
1015.          DY=C*D
1016.          10 CONTINUE
1017.          SIGX = ATT(1,6)
1018.          SIGY = ATT(1,10)
1019.          DO 20 M=1,NS
1020.          IF (EMDW.LT. 0)  NCBUHT = NCBUHT + 1
1021.          IF (NCBUHT.GT. 200)  GO TO 60
1022.          IF (MODE.LT. 0)  GO TO 13
1023.          C      IF (MODE.EQ. 0)  GO TO 11
1024.          C      RN = PANDT(1.)
1025.          C      GO TO 12
1026.          C 11  RN = PAND(1.)
1027.          RN = PAN(1)
1028.          C 12  IF (RN.GT. WPNP*L(NW))  GO TO 17
1029.          IF (PN.GT. WPNREL(NW))  GO TO 17
1030.          13 CONTINUE
1031.          CALL GAUSS(SIGX, X, KTEST, MODE)
1032.          CALL GAUSS(SIGY, Y, KTEST, MODE)
1033.          BAGZX=BDGZX+X*S+Y*C
1034.          BAGZY=BDGZY+X*C-Y*S
1035.          IF ((KTEST.GT. 0) .OR. (NPRINI.LT. -1))
1036.          X      WRITE (6,101) I, M, BAGZX, BAGZY
1037.          IF (EMDW.GE. 0)  GO TO 16
1038.          CBUHT(NCBUHT,1) = BAGZX
1039.          CBUHT(NCBUHT,2) = BAGZY
1040.          GO TO 18
1041.          16  IF ((BAGZX+BAGZY) .LT. -500)  GO TO 18
1042.          INDEX = I
1043.          CALL TESTHT(INDEX, BAGZX, BAGZY, NW)
1044.          GO TO 18
1045.          17  IF (EMDW.LT. 0)  CBUHT(NCBUHT,1) = -10000
1046.          18  BDGZX=BDGZX+DX
1047.          BDGZY=BDGZY+DY

```

AD-A090 682

RAND CORP SANTA MONICA CA

F/6 1/5

TSARINA: USER'S GUIDE TO A COMPUTER MODEL FOR DAMAGE ASSESSMENT--ETC(U)

JUL 80 D EMERSON

F49620-77-C-0023

UNCLASSIFIED

RAND/N-1460-AF

NL

2 of 2

AD-A090 682

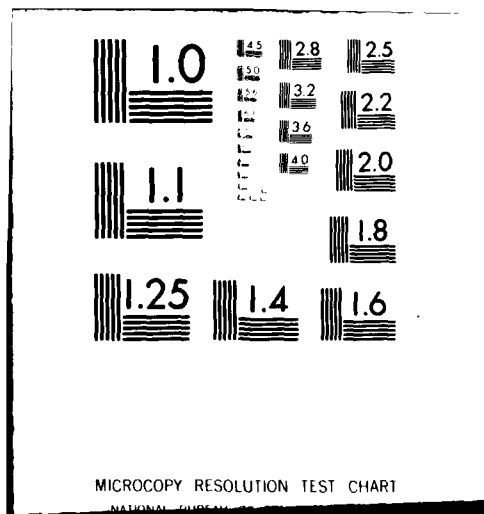


END

DATE

FILMED

DTIC



```

1048.      20  CONTINUE
1049.      40  CONTINUE
1050.      RETURN
1051.      60  WRITE(6,103)
1052.      STOP
1053.      101  FORMAT(' ', 'ATTACK #', I4, '   BOMB #', I4, '   X-DIM ', I8,
1054.      X      '   Y-DIM ', I8 )
1055.      102  FORMAT ('1', 30('*'), '   BOMB IMPACT AND HIT DATA ', 30('*') , //, //)
1056.      103  FORMAT('0', 'THE CBUHT ARRAY MUST BE ENLARGED TO ',
1057.      X      'ACCOMMODATE MORE CBU WEAPONS')
1058.      END

```

```

1060.      SUBROUTINE TESTHT(I,BX,BY,NW)
1061.      IMPLICIT INTEGER *2 (A-Z)
1062.      INTEGER *4 AID, HELP, HINO, MAXO, XY, K
1063.      REAL *4 DMAX, DR, PP
1064.      C THE FOLLOWING JCL INSERTS TSARINA'S COMMON "BASIC"
1065.      // DD DSN=*.STEP1.COMMON,DISP=SHR
1066.      // DD *.DCB=BLKSIZE=800
1067.      C
1068.      DIMENSION FLAG(6), DR(9), PP(6)
1069.      C
1070.      XY = BX + BY
1071.      NN=XY/500
1072.      K = MAXO(NN+1,1)
1073.      K = HINO(K,99)
1074.      LL = 0
1075.      IF (K.EQ. 1) GO TO 10
1076.      LL = IZONE (K-1, 1)
1077.      IF (K.EQ.2) GO TO 15
1078.      IF ((IZONE(K-1,1) .EQ. IZONE(K-2,2)) .AND.
1079.      X (IZONE(K-2,2) .NE. 1)) LL = IZONE(K,1)
1080.      GO TO 15
1081.      10 LL = IZONE (1, 1)
1082.      15 CONTINUE
1083.      LU = IZONE ((K+1), 2)
1084.      IF (KTEST .GT. 3) WRITE (6,102) I, LL,LU
1085.      DO 100 IL = LL, LU
1086.      C CONSIDER ALL TARGETS BETWEEN THE LIMITS OF LL AND LU
1087.      L = IO(IL,2)
1088.      IF (L .LE. 0) GO TO 100
1089.      IF ((TGT(L,12) + TGT(L,13)) .GT. 500) GO TO 100
1090.      NTGT = TGT(L,10)
1091.      HFLAG = 0
1092.      IF ((NTGT .NE. SHELTL).OP. (ALLMC .NE. 1)) GO TO 20
1093.      HFLAG = 1
1094.      DR(9) = EMD(NW,NTGT,10)
1095.      20 DR(1) = EMD(NW,NTGT,1)
1096.      DMAX = DR(1)
1097.      DR(2) = EMD(NW,NTGT,2)
1098.      IF (DMAX .LT. DR(2)) DMAX = DR(2)
1099.      HELP = EMD(NW,NTGT,9)
1100.      DO 25 FLG = 1,5
1101.      AID = 10**(6-FLG)
1102.      FLAG(FLG) = HELP/AID
1103.      25 HELP = HELP - AID*FLAG(FLG)
1104.      FLAG(6) = HELP
1105.      C
1106.      DO 30 FLG = 1,6
1107.      DR(FLG+2) = 0
1108.      IF (FLAG(FLG) .LT. 4) GO TO 30
1109.      IF (FLAG(FLG) .LT. 6) GO TO 28
1110.      AID = EMD(NW,NTGT,(FLG+2))
1111.      DR(FLG+2) = AID/1000
1112.      PP(FLG) = (AID - 1000*DR(FLG+2))/100.
1113.      IF (FLAG(FLG) .EQ. 8) GO TO 30
1114.      GO TO 29
1115.      28 DR(FLG+2) = EMD(NW,NTGT,(FLG+2))
1116.      IF (FLAG(FLG) .NE. 4) GO TO 30
1117.      29 IF (DR(FLG+2) .GT. DMAX) DMAX = DR(FLG+2)
1118.      30 CONTINUE
1119.      40 CONTINUE
1120.      D = DMAX
1121.      D = 1.414*D
1122.      IF ((TGT(L,1) - D) .GT. BX) GO TO 100
1123.      IF ((TGT(L,4) + D) .LT. BY) GO TO 100
1124.      IF ((TGT(L,5) + D) .LT. BX) GO TO 100
1125.      IF ((TGT(L,8) - D) .GT. BY) GO TO 100
1126.      IF (KTEST .GT. 4) WRITE (6,101) I,L,BX,BY,DMAX,HELP
1127.      CALL HITGT(I,L,BX,BY,NW,NTGT,DR,PP,FLAG,HFLAG)
1128.      100 CONTINUE
1129.      HFLAG = 0
1130.      DO 120 L = 1,NT
1131.      IF ((TGT(L,12) + TGT(L,13)) .LE. 500) GO TO 120
1132.      NTGT = TGT(L,10)
1133.      DR(1) = EMD(NW,NTGT,1)
1134.      DMAX = DR(1)
1135.      IF ((NTGT .EQ. 1).AND. (MCR .GT. 0)) GO TO 115

```

```

1136.      DR(2) = END(NW,NTGT,2)
1137.      IF (DHAX .LT. DR(2)) DHAX = DR(2)
1138.      HELP = END(NW,NTGT,9)
1139.      DO 105 FLG = 1,5
1140.      AID = 10**(6-FLG)
1141.      FLAG(FLG) = HELP/AID
1142. 105    HELP = HELP - AID*FLAG(FLG)
1143.      FLAG(6) = HELP
1144.
1145.  C      DO 110 FLG = 1,6
1146.      DR(FLG+2) = 0
1147.      IF (FLAG(FLG) .LT. 4) GO TO 119
1148.      IF (FLAG(FLG) .LT. 6) GO TO 106
1149.      AID = END(NW,NTGT,(FLG+2))
1150.      DR(FLG+2) = AID/100
1151.      PP(FLG) = (AID - 100C*DR(FLG+2))/100.
1152.      IF (FLAG(FLG) .EQ. 8) GO TO 110
1153.      GO TO 108
1154. 106    DR(FLG+2) = END(NW,NTGT,(FLG+2))
1155.      IF (FLAG(FLG) .NE. 4) GO TO 110
1156. 108    IF (DR(FLG+2) .GT. DHAX) DHAX = DR(FLG+2)
1157. 110    CONTINUE
1158. 115    IF (KTEST .GT. 4) WRITE (6,101) I,L,BX,BY,DHAX,HELP
1159.      CALL HITGT(I,L,BX,BY,NW,NTGT,DR,PP,FLAG,HFLAG)
1160. 120    CONTINUE
1161.      RETURN
1162. 101    FORMAT (' ',10I,'TEST HIT: ATTACK ',I3,' TGT ',I4,' X-DIM
1163.  X    ',I6,' Y-DIM ',I6,' P8.7, I8 )
1164. 102    FORMAT (' ',20X,'ATTACKER ',I3,' TARGET RANK LIMITS ',2I6,/)
1165.      END

```

```

1167.      SUBROUTINE HITGT(I,L,BX,BY,NW,NTGT,DP,PP,FLAG,MFLAG)
1168.      IMPLICIT INTEGER *2 (A-Z)
1169.      INTEGER *4 AID
1170.      REAL *4 DD,DR,S,C,CO,T,D,RHO,F,FX,FY,FAC,F1,F2,PP,
1171.      X TEST, OLD1,OLD2, YD,A,AL,DN,D2,R,RL,PS,TOT,Z, SIN, COS
1172.      C THE FOLLOWING JCL INSERTS TSARINA'S COMMON "BASIC"
1173.      // DD DSN=*.STEP1.COMMON,DISP=SHR
1174.      // DD *.DCB=BLKSIZE=800
1175.      DIMENSION DN(4), P(4), IS(4), D2(4), DR(9), FE(6), FLAG(6), FAC(9)
1176.      C
1177.      INSIDE = 0
1178.      HIT1 = 0
1179.      DO 2 NR = 1,9
1180.      2 PAC(NR) = 0.0
1181.      S = SIN (TGT (L,9)/57.3)
1182.      C = COS (TGT (L,9)/57.3)
1183.      CO = C
1184.      C
1185.      D = 0
1186.      T = 0
1187.      F1 = 0.0
1188.      F2 = 0.0
1189.      NSTART = 1
1190.      NEND = 9
1191.      C
1192.      4 CONTINUE
1193.      C
1194.      DO 100 NP = NSTART, NEND
1195.      IF (NP .GT. 1) GO TO 5
1196.      DD = 0
1197.      GO TO 30
1198.      5 NR = NP - 1
1199.      IF (NR .GT. 2) GO TO 10
1200.      IF ((HIT1.EQ.1).AND.(HCR.NE.0).AND.(NR.GT.1)) GO TO 105
1201.      IF ((NR.EQ.2).AND.(DR(NR).EQ.0.0)) GO TO 100
1202.      DD = DR(NR)
1203.      GO TO 20
1204.      10 IF (NR .LT. 9) GO TO 15
1205.      DD = DR(9)
1206.      GO TO 20
1207.      15 IF (FLAG(NR-2) .LT. 4) GO TO 85
1208.      DD = DR(NR)
1209.      IF ((FLAG(NR-2) .EQ. 5).AND.(HIT1 .EQ. 0)) GO TO 90
1210.      IF ((FLAG(NR-2) .EQ. 8).AND.(HIT1 .EQ. 0)) GO TO 90
1211.      20 IF (DD .EQ. 0.0) GO TO 30
1212.      T = (S + CO)*DD
1213.      D = (S - CO)*DD
1214.      30 X1 = TGT (L,1) - T
1215.      Y1 = TGT (L,2) + D
1216.      X2 = TGT (L,3) + D
1217.      Y2 = TGT (L,4) + T
1218.      X3 = TGT (L,5) + T
1219.      Y3 = TGT (L,6) - D
1220.      X4 = TGT (L,7) - D
1221.      Y4 = TGT (L,8) - T
1222.      IF ((BX .LT. X1) .OR. (BY .GT. X3)) GO TO 100
1223.      IF ((BY .LT. Y4) .OR. (BX .GT. Y2)) GO TO 100
1224.      IF (TGT (L, 9) .EQ. 0) GO TO 50
1225.      T = S/CO
1226.      C = 1./T
1227.      IF ((BX .LT. X2) .AND. (BY .GT. (Y1+C*(BX-X1)))) GO TO 100
1228.      IF ((BX .GT. X2) .AND. (BY .GT. (Y2-T*(BX-X2)))) GO TO 100
1229.      IF ((BX .GT. X4) .AND. (BY .LT. (Y4+C*(BX-X4)))) GO TO 100
1230.      IF ((BX .LT. X4) .AND. (BY .LT. (Y1-T*(BX-X1)))) GO TO 100
1231.      50 CONTINUE
1232.      IF (NP .NE. 1) GO TO 56
1233.      INSIDE = 1
1234.      IF ((END(NW,NTGT,10) .EQ. -1).OR.(NTGT .EQ. SHFLT)) GO TO 100
1235.      DR(1) = END(NW,NTGT,10)
1236.      DR(2) = END(NW,NTGT,11)
1237.      DRAX = DR(1)
1238.      IF (DRAX .LT. DR(2)) DRAX = DR(2)
1239.      DO 54 FLG = 1,6
1240.      DR(FLG+2) = 0
1241.      IF (FLAG(FLG) .LT. 4) GO TO 54
1242.      IF (FLAG(FLG) .LT. 6) GO TO 51
1243.      AID = END(NW,NTGT,(FLG+11))

```

```

1244.      DR(FLG+2) = AID/1000
1245.      PP(FLG) = (AID - 1000*DR(FLG+2))/100.
1246.      IF (FLAG(FLG) .EQ. 8) GO TO 54
1247.      GO TO 52
1248. 51 DR(FLG+2) = END(NW,NTGT,(FLG+11))
1249.      IF (FLAG(FLG) .NE. 4) GO TO 54
1250. 52 IF (DR(FLG+2) .GT. DHAX) DHAX = DR(FLG+2)
1251. 54 CONTINUE
1252.      GO TO 100
1253. 56 IF ((KTEST.GT.2) .OR. (NPRINT.LT.0)) WRITE(6,101) L,BX,BY,INSIDE,NR,DP
1254.      IF (NR .NE. 1) GO TO 60
1255.      HIT1 = 1
1256.      NHIT(L) = NHIT(L) + 1
1257.      IF (INSIDE .EQ. 0) OHIT(L) = OHIT(L) + 1
1258. 60 IF ((NTGT .EQ. 1) .AND. (RCR .NE. 0)) GO TO 110
1259. C
1260. C RESULTS INCLUDE AN ESTIMATE OF THAT FRACTION OF THE TARGET
1261. C AREA THAT IS COVERED BY CIRCLES OF RADIUS DD1 AND DD2.
1262. C
1263. C COMPUTE DISTANCES NORMAL TO THE FOUR SIDES OF THE TARGET.
1264. C
1265.      IF (TGT(L,9) .EQ. 0) GO TO 62
1266.      YD = Y1 + C*(BX-X1) - BY
1267.      DN(1) = S*YD
1268.      YD = Y2 - T*(BX-X2) - BY
1269.      DN(2) = CO*YD
1270.      YD = BY - Y4 - C*(BX-X4)
1271.      DN(3) = S*YD
1272.      YD = BY - Y1 + T*(BX - X1)
1273.      DN(4) = CO*YD
1274.      GO TO 64
1275. 62 CONTINUE
1276.      DN(1) = BX - X1
1277.      DN(2) = Y2 - BY
1278.      DN(3) = X3 - BX
1279.      DN(4) = BY - Y1
1280. 64 CONTINUE
1281.      TOT = 0.0
1282.      DO 66 N = 1,4
1283.      D2(N) = DD-DN(N)
1284.      IF (D2(N) .LE. 0.0) GO TO 66
1285.      TOT = TOT + D2(N)*D2(N)
1286. 66 CONTINUE
1287.      IF (KTEST.GT.5) WRITE(6,1102) TOT, (DN(N),N=1,4), (D2(N),N=1,4)
1288.      IF (TOT .LE. DD*DD) GO TO 68
1289.      IF (NR .NE. 1) GO TO 100
1290.      HIT1 = 0
1291.      NHIT(L) = NHIT(L) - 1
1292.      OHIT(L) = OHIT(L) - 1
1293.      GO TO 100
1294. C
1295. C SPECIAL TREATMENT IS REQUIRED WHEN SHELTERS
1296. C ARE HANDLED WITH THE MONTE CARLO MODE.
1297. C
1298. 68 IF (HFLAG .EQ. 0) GO TO 74
1299.      NS = TGT(L,14) - 1000
1300.      IF (NR .EQ. 1) GO TO 70
1301.      IF (NR .NE. 9) GO TO 74
1302.      GO TO 72
1303. 70 P2(NS,3) = 1. - (1. - P2(NS,3))*(100 - END(NW,NTGT,12))/100.
1304.      P2(NS,4) = 1. - (1. - P2(NS,4))*(100 - END(NW,NTGT,14))/100.
1305.      GO TO 74
1306. 72 P2(NS,5) = 1. - (1. - P2(NS,5))*(100 - END(NW,NTGT,13))/100.
1307.      IF (KTEST .GT. 5) WRITE(6,1101) NR, (P2(NS,K),K=3,5)
1308.      GO TO 102
1309. 74 CONTINUE
1310.      RL = DD
1311. 80 CONTINUE
1312.      AL = 3.1416*RL*RL
1313.      DO 84 N = 1,4
1314.      F(N) = 0.0
1315.      R = DD - DN(N)
1316.      IF ((R .GT. 0.0) .AND. (R .GE. RL)) F(N) = 1.0
1317.      IF (F(N) .EQ. 1.0) GO TO 83
1318.      IF ((R .LT. 0.0) .AND. (-R .GT. RL)) GO TO 83
1319.      IF (R .LT. 0.0) R = -R
1320.      IF (R .EQ. 0.0) GO TO 82

```

```

1321.      Z = R/RL
1322.      RHO = 2.*ATAN((1./(Z*Z)-1.))**.5)
1323.      A = RL*RL*(RHO - SIN(RHO))/2.
1324.      IF (DN(N) .GT. DD) A = AL - A
1325.      F(N) = 1. - A/AL
1326.      GO TO 83
1327.      82 F(N) = .5
1328.      83 IF (KTEST .GT. 5) WRITE(6,1101) N,R,Z,A,AL,F(N),DD
1329.      84 CONTINUE
1330.      FX = 1. - F(1) - F(3)
1331.      FY = 1. - F(2) - F(4)
1332.      IF (FX .LT. 0.0) FX = 0.0
1333.      IF (FY .LT. 0.0) FY = 0.0
1334.      IF (KTEST .GT. 4) WRITE(6,1002) FX,FY,R,FL,A,AL,(F(N),N=1,4)
1335.      C NOTE THAT THE USE OF FX AND FY PROVIDES ONLY AN APPROXIMATE RESULT
1336.      FAC(NR) = FX*FY*AL/(TGT(L,12)*TGT(L,13))
1337.      FAC(NR) = AMIN1(1.,FAC(NR))
1338.      IF (NR .LT. 3) GO TO 844
1339.      IF (FLAG(NR-2) .LT. 6) GO TO 844
1340.      IF (FLAG(NR-2) .EQ. 7) 841, 843, 842
1341.      841 FAC(NR) = (3.0*F1 + FAC(NR))*PP(NR-2)/4.0
1342.      GO TO 844
1343.      842 IF (HIT1 .EQ. 0) GO TO 844
1344.      843 FAC(NR) = (3.0*F2 + FAC(NR))*PP(NR-2)/4.0
1345.      844 CONTINUE
1346.      PS = 1. - FAC(NR)
1347.      P(L,NR) = 1. - (1. - P(L,NR))*PS
1348.      GO TO 90
1349.      85 XNR = NR
1350.      IF ((END(NW,NTGT,10) .NE. -1) .AND. (NTGT.NE.SHELT)) XNR=NR+9*INSTDE
1351.      IF (FLAG(NR-2) .EQ. 2) 86, 88, 87
1352.      86 IF (FLAG(NR-2) .EQ. 0) GO TO 90
1353.      P(L,NR) = 1. - (1. - P(L,NR))*(1. - (FAC(1)*END(NW,NTGT,XNR))/100.)
1354.      GO TO 90
1355.      87 IF (HIT1 .EQ. 0) GO TO 90
1356.      88 P(L,NR) = 1. - (1. - P(L,NR))*(1. - (FAC(2)*END(NW,NTGT,XNR))/100.)
1357.      90 CONTINUE
1358.      IF (NR .EQ. 2) F2 = FAC(2)
1359.      IF (NR .NE. 1) GO TO 100
1360.      F1 = FAC(1)
1361.      IF (TGT(L,11) .LT. 1.) GO TO 100
1362.      IF (NHIT(L) .GT. 25) GO TO 100
1363.      DO 95 J = 1, NST
1364.      IF (NHIT(J) .EQ. 0) GO TO 100
1365.      IF (NHIT(J) .NE. L) GO TO 95
1366.      HIT(J,1,NHIT(L))=BX
1367.      HIT(J,2,NHIT(L))=BY
1368.      HIT(J,3,NHIT(L))=NW
1369.      GO TO 100
1370.      95 CONTINUE
1371.      100 CONTINUE
1372.      IF ((INTGT .NE. SHELT) .OR. (NSTART .EQ. 10)) GO TO 102
1373.      NSTART = 10
1374.      NEND = 10
1375.      GO TO 4
1376.      102 CONTINUE
1377.      IF (KTEST .GT. 3) WRITE(6,1001) L,(FAC(K),K=1,8),(P(L,J),J=1,8)
1378.      105 CONTINUE
1379.      RETURN
1380.      110 CONTINUE
1381.      DO 120 J = 1,5
1382.      IF (NRW(J) .EQ. 0) GO TO 130
1383.      IF (NRW(J) .NE. L) GO TO 120
1384.      H1R(J,1,NHIT(L)) = BX
1385.      H1R(J,2,NHIT(L)) = BY
1386.      H1R(J,3,NHIT(L)) = NW
1387.      120 CONTINUE
1388.      130 CONTINUE
1389.      RETURN
1390.      101 FORMAT(' ',10X,'***** HITGT NHIT('',I3,''),N16,9P8.3)
1391.      1002 FORMAT(' ', ' FX ',P6.0, ' FY ', P6.0, 6X,4P8.0,4P8.4)
1392.      1001 FORMAT(' ', ' TGT ', I3, ' FAC ', 8P6.3, ' PK ',8P6.3)
1393.      1101 FORMAT(' ', I6,9P12.3)
1394.      1102 FORMAT(' ', P12.3, 8P8.3)
1395.      END

```

```

1397.      SUBROUTINE GAUSS (S,V,KTEST,MODE)
1398.      INTEGER *2 KTEST,MODE
1399.      IF ((KTEST .GT. 7) .OR. (MODE .LT. 0)) GO TO 50
1400.      A=0.0
1401.      C      IF (MODE .EQ. 0) GO TO 20
1402.      DO 10 I=1,12
1403.      Y = RAN(1)
1404.      C      Y = RANDT(1.)
1405.      10    A = A + Y
1406.      C      GO TO 40
1407.      C 20    DO 30 I=1,12
1408.      C      Y=RAND(1.)
1409.      C 30    A = A + Y
1410.      C 40    V = (A-6.0)*S
1411.      V = (A-6.0)*S
1412.      RETURN
1413.      50    CONTINUE
1414.      V = 0.0
1415.      RETURN
1416.      END

```

```

1418.      SUBROUTINE CHECKR
1419.      IMPLICIT INTEGER *2 (A-Z)
1420.      INTEGER *4 NSTAT, NHOLES
1421.      REAL *4 STAT
1422.      C      THE FOLLOWING JCL INSERTS TSARINA'S COMMON "BASIC"
1423.      //      DD DSN=*.STEP1.COMMON,DISP=SHR
1424.      //      DD *,DCB=BLKSIZE=800
1425.      COMMON /STATS / STAT(500,17),NSTAT(8),NTRIAL,ITRIAL,NSTAT
1426.      COMMON /CONTRL/ NREP,NPLOT,INW,INL,CHANGE,NSAVF,LIST,NJMEM
1427.      NC=0
1428.      NN=0
1429.      LHOLES = 10000
1430.      DO 40 NRW = 1, 5
1431.      C      CYCLE THRU AS MANY AS 8 RUNWAY/TAXIWAYS.
1432.      IRW = NRW(NRW)
1433.      C      EXIT IF NO TARGET NUMBER (IRW) FOUND.
1434.      IF (IRW .EQ. 0) GO TO 50
1435.      NN=NN+1
1436.      IF (NNIT(IRW) .EQ. 0) GO TO 40
1437.      IF (KTEST .GT. 4) WRITE (6,102) IRW
1438.      INDEX = NRW
1439.      CALL RUNWAY (INDEX, IRW, ICOND, NHOLES)
1440.      C      RUNWAY SUBROUTINE RETURNS ICOND = 0 IF RUNWAY HAS REQUIRED SPACE;
1441.      C      ICOND = 1 IF NOT.
1442.      C
1443.      IF (ICOND .EQ. 1) NC=NC+1
1444.      IF (NHOLES .LT. LHOLES) LHOLES = NHOLES
1445.      IF ((NPRINT .EQ. 4) .OR. (NPRINT .EQ. 6))
1446.      X      WRITE(6,103) ITRIAL, IRW, NNIT(IRW),NHOLES
1447.      40    CONTINUE
1448.      50    IF (NC .EQ. NN) GO TO 60
1449.      IF (NPRINT .LT. 3) WRITE (6,101)
1450.      NSTAT = NSTAT + 1
1451.      60    CONTINUE
1452.      IF ((NREP .EQ. 1) .AND. (NC .EQ. NN)) GO TO 70
1453.      RETURN
1454.      70    NSTAT(1) = NSTAT(1) + LHOLES
1455.      NSTAT(2) = NSTAT(2) + LHOLES*LHOLES
1456.      REPAIR = LHOLES
1457.      IF (LHOLES .EQ. 10000) REPAIR = 0
1458.      RETURN
1459.      101   FORMAT(' ', ' AT LEAST ONE RUNWAY IS AVAILABLE')
1460.      102   FORMAT(' ', ' CHECK TARGET #', I4)
1461.      103   FORMAT(' ', 'TRIAL', I4, ' TGT', I4, ' HITS', I4, ' REPAIRS', I2)
1462.      END

```

```

1464.      SUBROUTINE RUNWAY (MRW, IRW, ICOND, NHOLES)
1465.      IMPLICIT INTEGER *2 (A-Z)
1466.      INTEGER *4 NHOLES, IHOLE, MSTAT
1467.      REAL *4 STAT, TH, TH1, TH2, XX, YY, SIN, COS
1468.      C THE FOLLOWING JCL INSERTS TSARINA'S COMMON "BASIC"
1469.      // DD DEN=*.STEP1.COMMON,DISP=SHR
1470.      // DD *,DCB=BLKSIZE=800
1471.      COMMON /STATS / STAT(500,17),MSTAT(8),NTRIAL,ITRIAL,NSTAT
1472.      COMMON /HITS/ XN(250),YN(250),NZ(250)
1473.      COMMON /CONTRL/ NREP,NPLOT,INW,INL,CHANGE,NSAVE,LIST,NJMEM
1474.      DIMENSION NTEST(250), YH(250,2)
1475.      C
1476.      C CHECKS FOR THE EXISTENCE OF THE SPECIFIED RUNWAY MINIMUMS (MCL Y
1477.      C MCW) ON EACH RUNWAY AND DESIGNATED TAXIWAY (TYPE #1 TARGETS).
1478.      C STOPS SEARCHING A GIVEN RUNWAY WHENEVER REQUIREMENT IS SATISFIED.
1479.      C
1480.      TH=TGT(IRW,9)/57.3
1481.      C
1482.      C ESTABLISH ORIGIN (XO, YO) FOR A RECTANGULAR COORDINATE SYSTEM WITH
1483.      C THE X-AXIS ON THE MORE SOUTHERLY EDGE OF THE RUNWAY.
1484.      C
1485.      NHI = 0
1486.      NHOLES = 1000
1487.      DO 5 N = 1, 250
1488.      5 NZ(N) = 0
1489.      IF (TGT(IRW,12) .GT. TGT(IRW,13)) GO TO 10
1490.      NDIR=1
1491.      XO=TGT(IRW,1)
1492.      YO=TGT(IRW,2)
1493.      LTH = TGT(IRW,13)
1494.      WID = TGT(IRW,12)
1495.      GO TO 20
1496.      10 NDIR=2
1497.      XO=TGT(IRW,7)
1498.      YO=TGT(IRW,8)
1499.      LTH = TGT(IRW,12)
1500.      WID = TGT(IRW,13)
1501.      20 CONTINUE
1502.      IF (KTEST .GT. 4) WRITE (6,1004) IRW,XO,YO,LTH,WID,MCL,MCW
1503.      IF (MCW .GT. WID) GO TO 320
1504.      NHIT1 = NHIT(IRW)
1505.      DO 50 I = 1, NHIT1
1506.      IF (END(HITR(MRW,3,I),1,1) .LT. 0) GO TO 50
1507.      NTW = HITR(MRW,3,I)
1508.      GO TO 60
1509.      50 CONTINUE
1510.      60 CONTINUE
1511.      NON = 1
1512.      EMDW = EMD(NTW,1,1)
1513.      DO 70 I = 1, NHIT1
1514.      IF (HITR(MRW,3,I) .EQ. NTW) GO TO 70
1515.      IF (EMD(HITR(MRW,3,I),1,1) .LT. 0) GO TO 70
1516.      NON = 0
1517.      EMDW = 0.0
1518.      GO TO 80
1519.      70 CONTINUE
1520.      80 CONTINUE
1521.      DO 140 I = 1, NHIT1
1522.      C TRANSFORM HIT COORDINATES TO RUNWAY COORDINATES.
1523.      NHI = NHI + 1
1524.      XB = HITR(MRW,1,I)
1525.      YB = HITR(MRW,2,I)
1526.      IF (EMD(HITR(MRW,3,I),1,1) .LT. 0.) NZ(I) = 1
1527.      IF (TH .EQ. 0.0) GO TO 110
1528.      XX=XB-XO
1529.      YY=YB-YO
1530.      R = (XX*XX+YY*YY)**(0.5)
1531.      IF (KTEST .GT. 7) WRITE (6,1010) R,XX,YY
1532.      TH1 = ATAN(YY/XX)
1533.      IF (XX .LT. 0.0) TH1 = TH1 + 3.1416
1534.      TH2 = TH1 + TH
1535.      IF (NDIR .EQ. 2) TH2 = TH2 - 1.5706
1536.      XN(I)=R*COS(TH2)
1537.      YN(I)=R*SIN(TH2)
1538.      IF (KTEST .GT. 6) WRITE (6,1009) I,XN(I),YN(I)
1539.      GO TO 130
1540.      110 IF (NDIR .EQ. 2) GO TO 120

```

```

1541.      XM(I)=XB-XO
1542.      YN(I)=YB-YO
1543.      GO TO 130
1544. 120    XM(I)=YB-YO
1545.      YN(I)=XO-XB
1546. 130    IF (I.GT. 249) GO TO 150
1547. 140    CONTINUE
1548.      GO TO 160
1549. 150    WRITE (6,1001) IRW, ITRIAL
1550. 160    CONTINUE
1551.      NH = NHI
1552. C      IF NPLLOT.EQ. 2 RUNWAY IMPACTS ARE PLOTTED FOR ALL CONDITIONS.
1553. C      IF NPLLOT.EQ. 1 IMPACTS ONLY PLOTTED WHEN RUNWAY IS CLOSED.
1554.      IF ((NRW.EQ. 1).AND. (NPLLOT.EQ. 2)) WRITE(6,1012) ITRIAL
1555.      IF (NPLLOT.EQ. 2) CALL PLOTHT(NH,IRW,LTH,WID)
1556.      IF (KTEST.GT.6) WRITE (6,1006) NHIT(IRW),NH
1557.      DO 170 I=1,NH
1558.      YH(I,1) = YN(I)
1559. 170      YH(I,2)=I
1560.      IF (NH.EQ. 1) GO TO 190
1561.      DO 180 J=2,NH
1562.      DO 180 K=2,NH
1563. C      ORDER ALL HITS FROM LOWEST Y TO HIGHEST. YH(I,1) IS THE
1564. C      Y COORDINATE, YH(I,2) THE HIT NUMBER, OF THE I TH ORDERED HIT.
1565.      I=NH-K+2
1566.      IF (YH(I,1).GE. YH(I-1,1)) GO TO 180
1567.      T=YH(I-1,1)
1568.      TN=YH(I-1,2)
1569.      YH(I-1,1)=YH(I,1)
1570.      YH(I-1,2)=YH(I,2)
1571.      YH(I,1)=T
1572.      YH(I,2)=TN
1573. 180    CONTINUE
1574. 190    CONTINUE
1575.      XL = 0
1576.      XU = MCL
1577. 200      YL = 0
1578.      YU = MCW
1579.      NYL = 1
1580.      IF (NOM.EQ. 0) GO TO 210
1581.      YL = YL - EMDW
1582.      YU = YU + EMDW
1583. 210    CONTINUE
1584. 220    IHOLE = 0
1585.      DO 250 I = NYL, NH
1586.      IF (NZ(I).EQ. 1) GO TO 250
1587.      YT = YH(I,1)
1588.      IF (NOM.EQ. 1) GO TO 230
1589.      R = EMD(HITR(NRW,3,YH(I,2)),1,1)
1590.      YL = YL - R
1591.      YU = YU + R
1592. 230      IF (YT.LT. YL) GO TO 240
1593.      XT = XN(YH(I,2))
1594.      IF ((XT.LT. XL).OR. (XT.GT. XU)) GO TO 240
1595.      IF (YT.GT. YU) GO TO 260
1596.      IHOLE = IHOLE + 1
1597.      IF (KTEST.GT. 6) WRITE(6,1008) I, IHOLE, YL,YU
1598.      NTEST(IHOLE) = I
1599.      IF ((NREP.EQ. 0).OR. (IHOLE.GT. NHOLES)) GO TO 260
1600. 240      IF (NOM.EQ. 1) GO TO 250
1601.      IF (I.EQ. NH) GO TO 260
1602.      YL = YL + R
1603.      YU = YU - R
1604. 250    CONTINUE
1605. 260    CONTINUE
1606.      IF (IHOLE.EQ. 0) GO TO 300
1607.      IF (NOM.EQ. 1) NYL = NTEST(1)
1608.      NHOLES = NING(NHOLES, IHOLE)
1609.      IF (NOM.EQ. 0) GO TO 270
1610.      YL = YL + INW
1611.      YU = YU + INW
1612.      IF (YU.GT. (WID+EMDW)) GO TO 280
1613.      GO TO 220
1614. 270      YL = YL + R + INW
1615.      YU = YU - R + INW
1616.      IF (YU.GT. WID) GO TO 280
1617.      IF (KTEST.GT. 4) WRITE(6,1008) NHOLES

```

```

1618.      GO TO 220
1619. 280  CONTINUE
1620.      XL = XL + INL
1621.      XU = XU + INL
1622.      IF (XU .GT. LTH)  GO TO 290
1623.      GO TO 200
1624. 290  CONTINUE
1625.      IF (NPRINT .GE. 3)  GO TO 295
1626.      IF ((NRW .EQ. 1) .AND. (NPLOT .NE. 2))  WRITE (6,1012) ITRIAL
1627.      WRITE (6,1002) IRW
1628.      IF (NPEP .GT. 0)  WRITE (6,1011) NHOLES
1629.      IF (NPLOT .EQ. 1)  CALL PLOTH(NH,IRW,LTH,WID)
1630. 295  ICOND = 1
1631.      IF ((NPLOT .GT. 0) .AND. (NPRINT .LT. 3))  WRITE(6,1000)
1632.      RETURN
1633. 300  IF (NPRINT .GT. 2)  GO TO 310
1634.      IF (NRW .EQ. 1)  WRITE (6,1012) ITRIAL
1635.      WRITE (6,1003) IRW
1636. 310  ICOND = 0
1637.      IF ((NPLOT .EQ. 2) .AND. (NPRINT .LT. 3))  WRITE(6,1000)
1638.      NHOLES = 0
1639.      RETURN
1640. 320  WRITE(6,1005)  IRW
1641.      STOP
1642. 1000  FORMAT('1')
1643. 1001  FORMAT('0','ONLY FIRST 250 HITS TESTED FOR TARGET #',I4,
1644.  X    ' IN TRIAL #',I4,/,/)
1645. 1002  FORMAT('C',20X,'RUNWAY #',I3,' IS CLOSED',/,/)
1646. 1003  FORMAT('O',20X,'RUNWAY #',I3,' IS OPEN',/,/)
1647. 1004  FORMAT(' ', ' RUNWAY SPECS ',I4,4X,6I8)
1648. 1005  FORMAT('0','***** TARGET #',I3,' IS TOO NARROW FOR ',
1649.  X    ' FLIGHT OPERATIONS')
1650. 1006  FORMAT(' ', ' # HITS TO CHECK', 2I6)
1651. 1008  FORMAT(' ', ' TEST POINT #B ',3I10)
1652. 1009  FORMAT(' ', ' TEST POINT #E', I4,2I10)
1653. 1010  FORMAT(' ', ' R = ',F8.0,10X,2F10.0)
1654. 1011  FORMAT(' ', 17X,I4,' HOLES MUST BE REPAIRED TO MEET RUNWAY ',
1655.  X    ' MINIMUMS',/)
1656. 1012  FORMAT('1', ' ***** TRIAL #',I3,' *****',/)
1657.      END

```

```

1659.      SUBROUTINE PRINT
1660.          IMPLICIT INTEGER *2 (A-Z)
1661.          INTEGER *4 NSTAT
1662.          REAL *4 STAT, TOTAL
1663.      C   THE FOLLOWING JCL INSERTS SARINA'S COMMON "BASIC"
1664.      //   DD   DSN=*.STEP1.COMMON,DISP=SHR
1665.      //   DD   *,DCB=BLKSIZE=800
1666.          COMMON /STATS / STAT(500,17),NSTAT(8),NTRIAL,ITRIAL,NSTAT
1667.          IF (NTRIAL.EQ. 1) GO TO 1
1668.          IF ((NPRINT.LT.0).OR.((NPRINT.LT.3).AND.(MCR.EQ.2))) WRITE(6,107)
1669.          WRITE(6,108) ITRIAL
1670.          GO TO 2
1671.      1   CONTINUE
1672.          WRITE(6,106)
1673.      2   CONTINUE
1674.          WRITE(6,101)
1675.          DO 10 M=1,MTT
1676.              NN= 0
1677.              FLAG = 0
1678.              DO 10 I=1,NT
1679.                  IF (TGT(I,10) .NE. M) GO TO 10
1680.                  NN= NN+1
1681.                  NAID = NHIT(I) - NCBU(I)
1682.                  TOTAL = NAID + OHIT(I) + COV(I)
1683.                  DO 5 J = 1,8
1684.                      5 TOTAL = TOTAL + P(I,J)
1685.                  IF (TOTAL.EQ. 0.0) GO TO 10
1686.                  IF (FLAG.EQ. 1) GO TO 8
1687.                  FLAG = 1
1688.                  WRITE(6,109) M
1689.                  8   WRITE(6,102) I,NAID,OHIT(I),COV(I),(P(I,J),J=1,8),TGT(I,14),
1690.                  X   (NAME(I,J),J=1,NAMES)
1691.      10   CONTINUE
1692.          IF (NPRINT.GT. -1) GO TO 60
1693.          WRITE(6,103)
1694.          DO 20 M = 1, NST
1695.              IF (MHIT(M).EQ. 0) GO TO 30
1696.              NN=0
1697.              NL = NHIT(MHIT(M))
1698.              IF (NL.EQ. 0) GO TO 20
1699.              DO 15 N=1, NL
1700.                  NN=NN+1
1701.                  IF (NN.EQ. 26) GO TO 20
1702.                  IF (NN.EQ. 1) WRITE(6,104) MHIT(M)
1703.                  X=HIT(M,1,NN)
1704.                  Y=HIT(M,2,NN)
1705.                  WPN=HIT(M,3,NN)
1706.                  WRITE(6,105) X,Y,WPN
1707.      15   CONTINUE
1708.      20   CONTINUE
1709.      30   CONTINUE
1710.          DO 50 M = 1, 5
1711.              IF (NRW(M).EQ. 0) GO TO 60
1712.              NN = 0
1713.              NL = NHIT(NRW(M))
1714.              IF (NL.EQ. 0) GO TO 50
1715.              DO 40 N = 1, NL
1716.                  NN = NN + 1
1717.                  IF (NN.EQ. 251) GO TO 50
1718.                  IF (NN.EQ. 1) WRITE(6,104) NRW(M)
1719.                  X = HITR(M,1,NN)
1720.                  Y = HITR(M,2,NN)
1721.                  WPN = HITR(M,3,NN)
1722.                  WRITE(6,105) X, Y, WPN
1723.      40   CONTINUE
1724.      50   CONTINUE
1725.      60   CONTINUE
1726.          RETURN
1727.      101  FORMAT('0',' IGT   NUMBER ',4X,'CPU',16X,'COVERAGE',5X,'BLDG',/,
1728.      X   ' NO   HITS   COVERAGE',12X,'P1',6X,'R2   PEOPLE   AGE   ',
1729.      X   ' PARTS   AMMO   TRAP   NAIRL ',7X,'NO',/, ' ',7X,
1730.      X   'TOT OUT' )
1731.      102  FORMAT(' ',I3,3X,I3,2X,I3,3X,F6.2,10X,8(3X,F5.3),6X,I4,3X,4A4)
1732.      103  FORMAT('0',15X,'HIT LOCATION AND WPN TYPE FOR SELECTED TARGETS'
1733.      X,/,/)

```

```

1734.      104  FORMAT(' ', '      TARGET NUMBER', I4, '      X-DIM      Y-DIM      ',
1735.      X 'WPN TYPE', '/')
1736.      105  FORMAT(' ', '      21X, 2I9, I7)
1737.      106  FORMAT('1', 25X, 'TARGET HIT SUMMARY', '/', '/')
1738.      107  FORMAT('1')
1739.      108  FORMAT('0', 25X, 'TARGET HIT SUMMARY      TRIAL', I5)
1740.      109  FORMAT('0', 25X, '** TARGET TYPE # ', I3, ' **', '/')
1741.      END

```

```

1743. SUBROUTINE STATIS
1744. IMPLICIT INTEGER *2 (A-Z)
1745. INTEGER *4 MSTAT
1746. INTEGER *4 MXITEM, NOPEQ, NCAGE, NOPART, NOMUN, NCTRAP, NOMATL, NOPOL
1747. INTEGER *4 PEOPLE, AGE, PARTS, AMMO, TRAP, MATERL, POL, NOEQUI, NOPEOP
1748. REAL *4 PK, FHIT, AHITS, ACOV, AVGREP, CLSD, FCPEN, SCC, SDH,
1749. X STAT, TOTAL, TOTAL1, STA, SDREP, TRIAL, SUM1, SUM2
1750. C THE FOLLOWING JCL INSERTS TSARINA'S COMMON "BASIC"
1751. // DD USN=*.STEP1.COMMON.DISP=SHR
1752. // DD *.CCB=BLKSIZE=800
1753. COMMON /STATS / STAT(500,17), MSTAT(8), NTRIAL, ITRIAL, NSTAT
1754. COMMON /CONTRL/ NREP, NPLT, INW, INL, CHANGE, NSAVE, LIST, NJMEM
1755. COMMON /STORES/ MXITEM, NOPEQ, NOAGE, NOPART, NOMUN, NCTRAP, NOMATL,
1756. X NOPOL, NOEQUI, NOPEOP, PEOPLE(202,4), AGE(101,4), PARTS(401,4),
1757. X AMMO(126,4), TRAP(26,4), MATERL(26,4), POL(10,4), STOCKS(1000,3),
1758. X EQUIV(400)
1759. DIMENSION PK(8)
1760. WRITE (6,101) NTRIAL
1761. AVGREP = 0.0
1762. DO 12 M = 1, MTT
1763. NN = 0
1764. IF (M.NE. SHEL1) GO TO 2
1765. SUM1 = 0.0
1766. SUM2 = 0.0
1767. 2 CONTINUE
1768. TOTAL1 = 0.0
1769. DO 10 I = 1, NT
1770. IF (TGT(I,10).NE. M) GO TO 10
1771. NN = NN + 1
1772. IF (NN.EQ. 1) WRITE (6,102) M
1773. FHIT = (STAT(I,3)/NTRIAL)*100.
1774. AHITS = STAT(I,1)/NTRIAL
1775. TRIAL = NTRIAL
1776. SDH = (STAT(I,2) - TRIAL*AHITS*AHITS)/(TRIAL - 1.)
1777. SDH = SDH**(.5)
1778. ACOV = STAT(I,4)/TRIAL
1779. SOC = (STAT(I,5) - TRIAL*ACOV*ACOV)/(TRIAL - 1.)
1780. SOC = SOC**(.5)
1781. DO 5 L = 1,8
1782. 5 PK(L) = STAT(I,L+5)/TRIAL
1783. TOTAL = FHIT + AHITS + SDH + ACOV + SOC
1784. DO 6 L = 1,8
1785. 6 TOTAL = TOTAL + PK(L)
1786. IF (TOTAL.EQ. 0.0) GO TO 8
1787. C
1788. FLAG = 0
1789. IF ((TGT(I,14).GT.0).AND.(TGT(I,14).LT.1000)) FLAG = 1
1790. C
1791. IF (FLAG.EQ. 0) WRITE (6,103) I, FHIT, AHITS, SDH, ACOV, SOC,
1792. X (PK(L), L=1,8), (NAME(I,L), L=1, NAMES)
1793. C
1794. IF (FLAG.EQ. 1) WRITE (6,106) I, FHIT, AHITS, SDH, ACOV, SOC,
1795. X (PK(L), L=1,8), TGT(I,14), (NAME(I,L), L=1, NAMES)
1796. C
1797. TOTAL1 = TOTAL1 + PK(1)
1798. 8 IF (M.NE. SHEL1) GO TO 10
1799. SUM1 = SUM1 + FHIT
1800. SUM2 = SUM2 + AHITS
1801. 10 CONTINUE
1802. IF ((M.NE. SHEL1).OR.(NN.EQ. 0)) GO TO 11
1803. SUM1 = SUM1/NN
1804. SUM2 = SUM2/NN
1805. WRITE(6,104) SUM1, SUM2
1806. GO TO 12
1807. 11 IF (NN.EQ. 0) GO TO 12
1808. IF (M.LT. 5) GO TO 12
1809. TOTAL1 = TOTAL1/NN
1810. WRITE(6,109) TOTAL1
1811. 12 CONTINUE
1812. IF (MCR.EQ. 0) GO TO 50
1813. STA = NSTAT
1814. FOPEN = (STA/TRIAL)*100.
1815. NCLSD = NTRIAL - STA
1816. IF (NCLSD.EQ. 0) GO TO 40
1817. CLSD = NCLSD
1818. AVGREP = *STAT(1)/CLSD
1819. SDREP = 0.0
1820. IF (NCLSD.GT. 1)
1821. X SDREP = ((MSTAT(2)-CLSD*AVGREP*AVGREP)/(CLSD-1.))**(.5)

```

```

1822.      40  CONTINUE
1823.      WRITE (6,104) FOPEN
1824.      IF (NRFP.EQ. 1) WRITE(6,105) AVGREP, SOREP
1825.      50  CONTINUE
1826.      C
1827.      C  USE SUBROUTINE RESTAT TO SUMMARIZE THE RESOURCE LOSS STATISTICS
1828.      C
1829.      CALL RESTAT(PEOPLE(1,1),NOPEC, 1,NTRIAL)
1830.      CALL RESTAT( AGE(1,1),NOAGE, 2,NTRIAL)
1831.      CALL RESTAT( PARTS(1,1),NOPART,3,NTRIAL)
1832.      CALL RESTAT( AMMO(1,1),NOMUN, 4,NTRIAL)
1833.      CALL RESTAT( TRAP(1,1),NCTRAP,5,NTRIAL)
1834.      CALL RESTAT(MATERL(1,1),NOMATL,6,NTRIAL)
1835.      CALL RESTAT( POL(1,1),NOPCL, 7,NTRIAL)
1836.      C
1837.      RETURN
1838.      101  FORMAT ('1', 30X, 'TARGET DAMAGE STATISTICS FOR',14,' TRIALS',/,/,
1839.      X ' TARGET PERCENT AVERAGE HITS STD DEV AVG CBU STD '
1840.      X, 'DEV BCMB COVERAGE',10X, ' KILL PROBABILITIES',22X, 'BLDG ',/,
1841.      X ' NO ATTACKS HIT PER ATTACK OF HITS CCVERAGE COVE',
1842.      X 'RAGE R1 R2 PEOPLE AGE PARTS AMMO TRAP',
1843.      X ' MATERL BLDG NO/NAME ',/,/)
1844.      102  FORMAT (' ',/, 40X, 'TARGET TYPE # ',13,/)
1845.      103  FORMAT(' ',14,3X,F6.1,3X,F7.2,7X,F6.2,3X,F6.2,3X,F6.2,
1846.      X 8F7.3, 6X, 4A4)
1847.      104  FORMAT(' ',/, ' AT LEAST ONE MINIMUM RUNWAY SECTION WAS OPEN AFTE
1848.      XR', F6.1, ' PERCENT OF THE ATTACKS',/,/)
1849.      105  FORMAT(' ', 'WHEN ALL RUNWAYS WERE CLOSED, ',F6.2, '( ',F6.2,
1850.      X' ) HOLES REQUIRED REPAIR, ON THE AVERAGE, TO PROVIDE',
1851.      X ' A MINIMUM RUNWAY',/,/)
1852.      106  FORMAT(' ',14,3X,F6.1,3X,F7.2,7X,F6.2,3X,F6.2,3X,F6.2,
1853.      X 8F7.3, 2X,13,1X, 4A4)
1854.      108  FORMAT(' ', 8X, '-----',4X, '-----',/, ' ', 8X,F6.2,3X,F6.2 )
1855.      109  FORMAT(' ',55X, '-----',/, ' ',54X,F6.2 )
1856.      END

```

```

1858.      SUBROUTINE CBU
1859.      IMPLICIT INTEGER *2 (A-Z)
1860.      INTEGER *4 MSTAT
1861.      REAL *4 PHI, S, C, T, CT, A0, A1, A2, A3, A4, A5, XT, YT,
1862.      X      STAT, PK, PSP, PST, PSCOV, TOTC, SIN, COS
1863.      C      THE FOLLOWING JCL INSERTS TSARINA'S COMMON "BASIC"
1864.      //      DD      DSN=*.STEP1.COMMON,DISP=SHR
1865.      //      DD      *,DCB=BLKSIZE=800
1866.      COMMON /STATS / STAT(500,17),MSTAT(8),NTRIAL,ITRIAL,NSTAT
1867.      COMMON /CBUHT/ CBUHT(200,2), IR(50), KCBU, KPTI
1868.      DIMENSION NCOV(16,16),ICOV(16,16),PSCOV(16,16,6),PST(6),
1869.      X      X(4), Y(4), XX(4), YY(4), WD(10)
1870.      C      COMPUTE PATTERN DIAGONAL DIMENSION
1871.      DO 2      NWPN = 1, 10
1872.      WD(NWPN) = 0
1873.      LTH = -END(NWPN,1,1)
1874.      IF (LTH .LE. 0)      GO TO 2
1875.      WID = END(NWPN,2,1)
1876.      WD(NWPN) = ((LTH*LTH + WID*WID)**(.5))/2.
1877.      2      CONTINUE
1878.      C      AT THIS POINT THE PROGRAM NOW CHECKS, TARGET BY TARGET, FOR
1879.      C      WHATEVER CBU'S MAY HAVE COVERED ANY PART OF THE TARGET.
1880.      DO 200      L = 1, NT
1881.      C      *****      FOR EACH TARGET
1882.      INIT1 = 0
1883.      C      COMPUTE TARGET CENTER
1884.      TCX=(TGT(L,1)+TGT(L,5))*0.5
1885.      TCY=(TGT(L,2)+TGT(L,6))*0.5
1886.      C      TARGET DIAGONAL
1887.      XA=TGT(L,1)-TGT(L,5)
1888.      YA=TGT(L,2)-TGT(L,6)
1889.      TD = 0.5*((XA*XA + YA*YA)**(.5))
1890.      NCBUHT = 0
1891.      FLAG = 0
1892.      DO 40      I = 1, NA
1893.      C      *****      FOR EACH ATTACK
1894.      IF (IR(I) .EQ. 1)      GO TO 40
1895.      INIT2 = 0
1896.      NWPN = ATT(I,9)
1897.      LTH = -END(NWPN,1,1)
1898.      IF (LTH .LE. 0)      GO TO 40
1899.      WID = END(NWPN,2,1)
1900.      TOT = WD(NWPN) + TD
1901.      NS = ATT(I,7)
1902.      INIT3 = 0
1903.      DO 20      M = 1, NS
1904.      C      *****      FOR EACH WEAPON
1905.      NCBUHT = NCBUHT + 1
1906.      XB = CBUHT(NCBUHT, 1)
1907.      IF (XB .EQ. -10000)      GO TO 20
1908.      YB = CBUHT(NCBUHT, 2)
1909.      C      DISTANCE BETWEEN TARGET AND PATTERN CENTERS
1910.      D = ((XB-TCX)*(XB-TCX) + (YB-TCY)*(YB-TCY))**(.5)
1911.      C      TARGET CANNOT BE HIT IF D GREATER THAN TOT
1912.      IF (D .GT. TOT)      GO TO 20
1913.      FLAG = 1
1914.      IF (INIT3 .GT. 0)      GO TO 16
1915.      INIT3 = 1
1916.      PHI = ATT(I,1)/57.3
1917.      S = SIN(PHI)
1918.      C = COS(PHI)
1919.      SL = S*LTH
1920.      SW = S*WID
1921.      CL = C*LTH
1922.      CW = C*WID
1923.      DO 5      J = 1, 16
1924.      DO 5      K = 1, 16
1925.      5      NCOV(J,K) = 0
1926.      16      CONTINUE
1927.      XX(1) = XB - (SL+CW)/2.
1928.      XX(2) = XX(1) + SL
1929.      XX(3) = XX(2) + CW
1930.      XX(4) = XX(1) + CW
1931.      YY(1) = YB + (SW-CL)/2.
1932.      YY(2) = YY(1) + CL

```

```

1933.      YY(3) = YY(2) - SW
1934.      YY(4) = YY(1) - SW
1935.      IF (INIT2 .GT. 0) GO TO 18
1936.      INIT2 = 1
1937. C     FIND WESTERLY CORNER
1938.      ILX = 1
1939.      DO 15 NN = 1,4
1940.      15 IF (XX(NN) .LT. XX(ILX)) ILX = NN
1941.      IF (S .EQ. 1.0) ILX = 4
1942. C     RENUMBER CORNERS SO THAT CORNER #1 IS THE MOST WESTERN
1943.      IDIF = ILX - 1
1944. C     COMPUTE AND ADJUST TAN AND COTAN AS REQUIRED
1945.      IF ((S .EQ. 0.0) .OR. (C .EQ. 0.0)) GO TO 28
1946.      IF ((ILX .EQ. 2) .OR. (ILX .EQ. 4)) GO TO 26
1947.      T = S/C
1948.      GO TO 27
1949.      26 T = -C/S
1950.      27 CT = 1./T
1951.      28 CONTINUE
1952.      18 DO 22 NN = 1,4
1953.          NEW = NN - IDIF
1954.          IF (NEW .LT. 1) NEW = NEW + 4
1955.          X(NEW) = XX(NN)
1956.          Y(NEW) = YY(NN)
1957.      22 IF (KTEST .GT. 4) WRITE(6,1003) NEW, X(NEW), Y(NEW)
1958.      IF (INIT1 .GT. 0) GO TO 31
1959. C     CREATE A 16-POINT GRID ON TARGET - USE MORE POINTS FOR LARGE TGTS
1960.      INIT1 = 1
1961.      NX0 = 8
1962.      NY0 = 8
1963.      IF (TGT(L,12) .GT. 250.) NY0 = 16
1964.      IF (TGT(L,12) .GT. 1000.) NY0 = 32
1965.      IF (TGT(L,13) .GT. 250.) NX0 = 16
1966.      IF (TGT(L,13) .GT. 1000.) NX0 = 32
1967.      NXT = NX0/2
1968.      NYT = NY0/2
1969.      DO 29 J = 1, NXT
1970.      DO 29 K = 1, NYT
1971.      ICOV(J,K) = 0
1972.      DO 29 NP = 3,8
1973.      29 PSCOV(J,K,(NP-2)) = 1.0
1974.      NX1 = NX0 - 1
1975.      NY1 = NY0 - 1
1976.      X0 = NX0
1977.      Y0 = NY0
1978.      A0 = TGT(L,1)
1979.      A1 = (TGT(L,7) - TGT(L,1))/X0
1980.      A2 = (TGT(L,3) - TGT(L,1))/Y0
1981.      A3 = TGT(L,2)
1982.      A4 = (TGT(L,4) - TGT(L,2))/Y0
1983.      A5 = (TGT(L,8) - TGT(L,2))/X0
1984.      31 CONTINUE
1985. C     TEST TO SEE IF TARGET CORNERS COVERED BY PATTERN
1986.      NIN = 0
1987.      NGIN = 0
1988.      DO 10 NC = 1, 7, 2
1989.      XT = TGT(L,NC)
1990.      YT = TGT(L,NC+1)
1991.      IF (KTEST .GT. 4) WRITE(6,1001) XT, YT
1992.      IF ((XT .LE. X(1)) .OR. (XT .GE. X(3))) GO TO 10
1993.      IF ((YT .LE. Y(4)) .OR. (YT .GE. Y(2))) GO TO 10
1994.      IF ((S .EQ. 0.) .OR. (C .EQ. 0.)) GO TO 9
1995.      IF (YT .GE. (Y(1) + CT*(XT - X(1)))) GO TO 10
1996.      IF (YT .GE. (Y(2) - T*(XT - X(2)))) GO TO 10
1997.      IF (YT .LE. (Y(4) + CT*(XT - X(4)))) GO TO 10
1998.      IF (YT .LE. (Y(1) - T*(XT - X(1)))) GO TO 10
1999.      9 NIN = NIN + 1
2000.      10 CONTINUE
2001.      IF (KTEST .GT. 3) WRITE(6,1002) NIN
2002.      IF (NIN .LT. 4) GO TO 34
2003. C     IF ALL CORNERS COVERED BY PATTERN, TARGET FULLY COVERED
2004.      DO 32 J = 1, NXT
2005.      DO 32 K = 1, NYT
2006.      32 MCOV(J,K) = MCOV(J,K) + 1
2007.      GO TO 33

```

```

2008. C IF PARTIALLY COVERED, ESTIMATE FFACTION THAT IS COVERED
2009. 34 CONTINUE
2010. DO 30 NX = 1,NX1,2
2011. DO 30 NY = 1,NY1,2
2012. J = (NX+1)/2.
2013. K = (NY+1)/2.
2014. C GRID-POINT DIMENSIONS
2015. XT = A0 + NX*A1 + NY*A2
2016. YT = A3 + NY*A4 + NX*A5
2017. C CHECK IF WITHIN RECTANGLE ENCLOSING PATTERN THAT IS PARALLEL TO
2018. C AXES
2019. IF ((XT .LT. X(1)).OR.(XT .GT. X(3))) GO TO 30
2020. IF ((YT .LT. Y(4)).OR.(YT .GT. Y(2))) GO TO 30
2021. IF ((S .EQ. 0.) .OR. (C .EQ. 0.)) GO TO 35
2022. C CHECK IF POINT IS WITHIN ACTUAL CBU PATTERN
2023. IF (YT .GT. (Y(1)+CT*(XT-X(1)))) GO TO 30
2024. IF (YT .GT. (Y(2)-T*(XT-X(2)))) GO TO 30
2025. IF (YT .LT. (Y(4)+CT*(XT-X(4)))) GO TO 30
2026. IF (YT .LT. (Y(1)-T*(XT-X(1)))) GO TO 30
2027. 35 NGIN = NGIN + 1
2028. NCOV(J,K) = NCOV(J,K) + 1
2029. IF (KTEST .GT. 5) WRITE(6,1005) NX,NY,XT,YT,NGIN,NCOV(J,K)
2030. 30 CONTINUE
2031. 33 CONTINUE
2032. IF ((MIN + NGIN) .EQ. 0) GO TO 20
2033. C RECORD ANY COVERAGE AS A 'HIT'
2034. NHIT(L) = NHIT(L) + 1
2035. NCBU(L) = NCBU(L) + 1
2036. IF (KTEST .GT. 4) WRITE(6,1006) L, NHIT(L), NCBU(L)
2037. IF ((TGT(L,1) .LT. 1) .OR. (TGT(L,10) .EQ. 1.)) GO TO 130
2038. IF (NHIT(L) .GT. 25) GO TO 130
2039. DO 120 J = 1, NST
2040. IF (MHIT(J) .EQ. 0) GO TO 130
2041. IF (MHIT(J) .NE. L) GO TO 120
2042. HIT (J,1,NHIT(L)) = XB
2043. HIT (J,2,NHIT(L)) = YB
2044. HIT (J,3,NHIT(L)) = NWPW
2045. IF (NHIT(L) .EQ. 25) WRITE (6,1007) L, ITrial
2046. GO TO 130
2047. 120 CONTINUE
2048. 130 CONTINUE
2049. IF (TGT(L,10) .NE. 1) GO TO 150
2050. IF (NHIT(L) .GT. 250) GO TO 150
2051. DO 140 J = 1, 5
2052. IF (NRW(J) .EQ. 0) GO TO 150
2053. IF (NRW(J) .NE. L) GO TO 140
2054. HIR(J,1,NHIT(L)) = XB
2055. HIR(J,2,NHIT(L)) = YB
2056. HIR(J,3,NHIT(L)) = NWPW
2057. IF (NHIT(L) .EQ. 250) WRITE(6,1009) L, ITrial
2058. 140 CONTINUE
2059. 150 CONTINUE
2060. C ***** RECYCLE FOR MORE WEAPONS
2061. 20 CONTINUE
2062. IF (INIT2 .EQ. 0) GO TO 40
2063. DO 160 NP = 3, 8
2064. PK = EMD(NWPW,TGT(L,10),NP)/100.
2065. PSP = 1. - PK
2066. DO 160 J = 1, NXT
2067. DO 160 K = 1, NYT
2067.1 IF (NP .GT. 3) GO TO 160
2068. ICOV(J,K) = ICOV(J,K) + NCOV(J,K)
2069. 160 PSCOV(J,K,NP-2) = PSCOV(J,K,NP-2)*PSP**NCOV(J,K)
2070. C ***** RECYCLE FOR MORE ATTACKS
2071. 40 CONTINUE
2072. IF (FLAG .EQ. 0) GO TO 200
2073. TCOV = 0
2074. DO 165 NP = 3, 8
2075. 165 PST(NP-2) = 0.0
2076. DO 170 J = 1, NXT
2077. DO 170 K = 1, NYT
2078. IF (ICOV(J,K) .GT. 0) TCOV = TCOV + 1
2079. DO 170 NP = 3, 8
2080. 170 PST(NP-2) = PST(NP-2) + (1. - PSCOV(J,K,NP-2))
2081. TOTC = NXT*NYT
2082. COV(L) = TCOV/TOTC

```

```

2093.      DO 180 NP = 3, 8
2094.      180 P(L,NP) = 1 - (1.-P(L,NP))*(1.-PST(NP-2)/TOIC)
2095.      IF (KTEST.GT. 3) WRITE(6,1004) L,TCOV,TOTC, (PST(I),I=1,6),
2096.      X      COV(L),(P(L,I),I=1,8)
2097.      C ***** RECYCLE FOR MORE TARGETS
2098.      200 CONTINUE
2099.      RETURN
2100.      1001 FORMAT(' ', ' XT ',F8.0, ' YT ',F8.0)
2101.      1002 FORMAT(' ', ' NIN ',I4)
2102.      1003 FORMAT(' ', ' NEW ',I3,2I10)
2103.      1004 FORMAT(' ', ' TGT ',I4, ' COV ',I4, ' TOT ',F6.0, ' PST ',
2104.      X 6F6.3, ' COV ',F6.4, ' PK ',6F6.3 )
2105.      1005 FORMAT(' ', ' NX ',I3, ' NY ',I3,2F10.2,2I10)
2106.      1006 FORMAT(' ', ' TGT ',I4, ' WHIT ',I5, ' NCEU ',I5)
2107.      1007 FORMAT('0','***** ONLY 25 HITS WERE STORED FOR TARGET ',
2108.      X I3, ' DURING TRIAL ',I4, ' *****')
2109.      1009 FORMAT('0','***** ONLY 250 HITS WERE STORED FOR TARGET ',
2110.      X ' ',I3, ' DURING TRIAL ',I4, ' *****')
2111.      EN

```

```

2103.      SUBROUTINE PLOTHT(NH,WR,LTH,WID)
2104.      IMPLICIT INTEGER *2 (A-Z)
2105.      REAL *4 Y
2106.      COMMON /HITS/ XN(250),YN(250),NZ(250)
2107.      DIMENSION ICOL(139)
2108.      DATA IBK / 1H /,IX/ 1H* /,IY/ 1H+ /,IS/ 1H- /,IE/ 1H' /
2109.      THIS ROUTINE PLOTS THE IMPACT POINTS (BUT NOT CRATERPS) FOR
2110.      ALL HITS THAT HAVE BEEN STORED FOR A RUNWAY/TAXIWAY. IT
2111.      WILL PLOT ALL HITS THAT AFFECT RUNWAY OPERATION UP TO 50 'FEET'
2112.      OF EITHER SIDE OF (UP TO) A 300 'FOOT' RUNWAY. RUNWAY LENGTH
2113.      IS LIMITED TO 13000 'FEET'.
2114.      IWID = WID/10 + 5
2115.      LEN = LTH/100 + 1
2116.      IF (LEN .GT. 129) LEN = 129
2117.      LU = LEN/10
2118.      LI = 10*LU + 1
2119.      LU = LU + 1
2120.      DO 40 J = 1,40
2121.          I = 41-J
2122.          DO 10 N = 1,129
2123.      10 ICOL(N) = IBK
2124.          ICOL(1) = IE
2125.          ICOL(LEN) = IE
2126.          IF ((I .NE. 5) .AND. (I .NE. IWID)) GO TO 14
2127.          DO 11 NS = 1,LEN
2128.      11 ICOL(NS) = IS
2129.          DO 12 NS = 1,LI,10
2130.      12 ICOL(NS) = IF
2131.      14 CONTINUE
2132.          DO 20 N = 1,NH
2133.              NY = YN(N)/10 + 5
2134.              IF (NY .NE. I) GO TO 20
2135.              NX = XN(N)/100 + 1
2136.              IF ((NX .LT. 1) .OR. (NX .GT. 129)) GO TO 20
2137.              ICOL(NX) = IX
2138.              IF (NZ(N) .EQ. 1) ICOL(NX) = IY
2139.      20 CONTINUE
2140.              Y = I/5.
2141.              LY=Y
2142.              IF ((Y-LY) .NE. 0.2) GO TO 30
2143.                  LYY = 5*LY - 5
2144.                  WRITE (6,101) LYY, (ICOL(M),M=1,129)
2145.                  GO TO 40
2146.      30 WRITE (6,102) (ICOL(M),M=1,129)
2147.      40 CONTINUE
2148.              WRITE (6,103) (I, I=1,12), WR
2149.              RETURN
2150.      101 FORMAT (' ',I2,129A1)
2151.      102 FORMAT (' ',2X,129A1)
2152.      103 FORMAT (' ', ' 0', 12(8X,I2),/,
2153.      X 40X, ' TENS BY THOUSANDS OF LENGTH UNITS ',/,/,
2154.      X 40X, ' IMPACT POINTS ON RUNWAY NUMBER ',I2,/,
2155.      X 37X, ' (* = POINT IMPACT WPN + = CBU CENTERCID)')
2156.      END

```

```

2158.      SUBROUTINE JMEMO(NJMEM,D,NA,KTEST)
2159.      INTEGER *2 NJMEM, D, NA, KTEST
2160.      INTEGER *4 EE
2161.      DIMENSION E(9), EE(9), D(11)
2162. C   THIS SUBROUTINE PROVIDES THE USER 'METHOD 2' AS OUTLINED IN THE
2163. C   8-74 WANG 700 USERS'S MANUAL FOR JMEM OPEN-END METHODS.
2164. C   THIS TRAJECTORY PROGRAM PERMITS AIDA USERS TO PRESCRIBE THE
2165. C   ATTACK DATA AS IN JMEM. THE SUBROUTINE LOGIC IS TAKEN DIRECTLY FROM
2166. C   THE REFERENCED PUBLICATION AND USES NOTATION CLOSELY PARALLELING
2167. C   THE ORIGINAL. ONLY THE 'PATTERN RADIUS' COMPUTATION (USED WITH
2168. C   ROCKEYE) IS OMITTED.
2169.      NJMEM = NJMEM + 1
2170.      NCNT = 0
2171.      IF (NJMEM.EQ. 1) WRITE (6,101)
2172.      READ (5,102) (EE(I),I=1,9)
2173.      DO 16 I = 1,6
2174. 16    E(I) = EE(I)
2175.      DO 18 I = 7,9
2176. 18    E(I) = EE(I)/1000.
2177.      WRITE (6,103) NA, (D(I),I=1,6)
2178.      WRITE (6,104) (E(I),I=1,9)
2179.      DO 20 I = 7,9
2180. 20    D(I) = D(I+1)
2181.      V1 = 1.688*E(1)
2182.      V = V1
2183.      TH = E(3)/57.3
2184.      VX = V*COS(TH)
2185.      VY = -V*SIN(TH)
2186.      TR = (D(7) - 1.)*E(8) + (E(5)-1.)*E(9)
2187.      DX = TR*VX/2.
2188.      Y1 = E(2) - VY*TR
2189.      YI = (Y1 + E(2))/2.
2190.      Y = YI
2191.      TP = 0.0
2192.      VP = 0.0
2193.      THP=0.0
2194.      IF (E(7) .GT. 500.) GO TO 1
2195.      YT = 0.0
2196.      IF (E(7) .EQ. 0.0) GO TO 2
2197.      TPD=E(7)
2198.      GO TO 3
2199. 1    YT = E(7)
2200. 2    TPD= 99.
2201. 3    CONTINUE
2202.      DG = 32.17/(E(4)*E(4))
2203. 7    CONTINUE
2204.      IF (KTEST .GT. 5) WRITE (6,106) TP,Y,V
2205.      NCNT = NCNT + 1
2206.      IF (NCNT .GT. 2000) GO TO 30
2207.      NCNT2 = 0
2208.      C = V/220. - 3.
2209.      IF (C .GE. 3.) C = 3.
2210.      IF (C .LT. 1.) C = 1.
2211.      DD = C*DG*EXP(-Y/31000.)
2212.      DT= 10./(DG*V*V)
2213.      IF (DT .GT. 0.5) DT = 0.5
2214.      GO TO 13
2215. 5    CONTINUE
2216.      TP = TPD
2217.      VY = VYC
2218.      Y = Y0
2219.      IF (KTEST .GT. 5) WRITE (6,107) DT, VY
2220.      NCNT2 = NCNT2 + 1
2221.      IF (NCNT2 .GT. 100) GO TO 30
2222. 13   CONTINUE
2223.      TPO = TP
2224.      TF = TP + DT
2225.      IF (TF .LT. TPD) GO TO 4
2226.      TP = TPD
2227.      DT = TF - TPO
2228. 4    CONTINUE
2229.      VY0 = VY
2230.      VY = VY*(1.-V*DT*DD) - 32.17*DT
2231.      Y0 = Y
2232.      Y = Y + DT*(VY+VY0)/2.
2233.      IF ((Y - YT) .GE. -1.) GO TO 6

```

```

2234.      ZZ = VYG*V*DD + 32.17
2235.      Z = VY0*VY0 + 2.0*(Y0-Y1)*ZZ
2236.      DT = (VY0 + Z**(.5))/ZZ
2237.      GO TO 5
2238. 6 CONTINUE
2239.      VX = VX
2240.      VX = VX*(1.-V*DT*DD)
2241.      DX = DX + DT*(VX0+VX)/2.
2242.      V = (VX*VX + VY*VY)**(.5)
2243.      IF (TF .EQ. TPD) GO TO 12
2244.      IF (E(7) .GT. 500.) GO TO 11
2245.      GO TO 10
2246. 11 AID = ABS((Y-Z(7)))
2247.      IF (AID .LT. 1.) GO TO 12
2248.      IF (KTEST .GE. 5) WRITE (6,109) AID, TF
2249. 10 CONTINUE
2250.      IF (KTEST .GE. 5) WRITE(6,108) Y
2251.      IF (Y .GE. 1.) GO TO 7
2252.      GO TO 15
2253. 12 CONTINUE
2254.      Z = -VY/V
2255.      ZZ = (1.-Z*Z)**(.5)
2256.      THF=ATAN(Z/ZZ)
2257.      DG = 32.17/(E(5)*E(5))
2258.      TPD= 99.
2259.      YT = 0.0
2260.      IF (E(7) .GE. 500.) E(7) = 0.0
2261.      GO TO 10
2262. 15 Z = -VY/V
2263.      ZZ = (1.-Z*Z)**(.5)
2264.      AI = ATAN(Z/ZZ)
2265.      SR = (DX*DX + Y1*Y1)**(.5)
2266.      IF (D(5) .EQ. 0.0) GO TO 8
2267.      BP = D(4)/1000.
2268.      CP = D(5)/1000.
2269.      GO TO 9
2270. 8 BP = 0.000573*D(4)
2271.      CP = BP
2272. 9 CONTINUE
2273.      U = (E(6)*E(6)+TF*TF)
2274.      D(4) = (U + (SR*SR*BP/Y1)**2.)*(.5)
2275.      D(5) = (U+CP*CP*SR*SR)**(.5)
2276.      SRT = SR/1000.
2277.      D(8) = V1*TR*SIN(AI-TH)/SIN(AI)
2278.      DISP = D(6)
2279.      D(10) = DISP*SRT
2280.      D(6) = D(10)/Z
2281.      AID = 57.3*AI
2282.      THPD= 57.3*THF
2283.      WRITE(6,105) TF, SR, AID, THPD
2284.      RETURN
2285. 30 WRITE (6,110)
2286.      STOP
2287. 101 FORMAT('1',20X,'JHEH FORMAT ATTACK DATA',/,/,',',
2288.      X'ATT NO HDG X-MPI Y-MPI CEP(REP) (DEP) DIS
2289.      X',/,/,',',
2290.      X' SPEED ALTITUDE DIVE TERM1 TERM2 WIN
2291.      XD TD/HF T TD',/,/,')
2292. 102 FORMAT( 6X, 9I6 )
2293. 103 FORMAT ('0',16,7I10 )
2294. 104 FORMAT (' ',6X, 6F10.0, 3F10.3)
2295. 105 FORMAT(' ',30X,'TF ',F5.2,' SEC SR ',F7.0,' FEET ',
2296.      X 'IMPACT ANGLE ',F5.2,' DEG (FUZING ANGLE ',F5.2,' DEG)')
2297. 106 FORMAT(' ',',',',F6.3,' SEC ',F8.0,' FEET ',F8.1,' FT/SEC')
2298. 107 FORMAT(' ',10X,' DT ',F10.4,' VERT VEL',F10.3)
2299. 108 FORMAT(' ',10X,' ALT ',F10.2)
2300. 109 FORMAT(' ',10X,' ALT DIFF',F10.2,' TF ',F10.2)
2301. 110 FORMAT(' ',/,/,',','LOOPING IN JHEH. CHECK INPUTS AND/OR TEST',
2302.      X ' WITH KTEST = 6.')
2303.      END

```

```

2305.      SUBROUTINE EXPHIT
2306.      IMPLICIT INTEGER *2 (A-Z)
2307.      REAL *4 PI, PHI, REL, S, C, CT, TH, CHI, SC, CC, AID1, AID2, DD, DR,
2308.      X STG, CTG, XT, YT, XX, YY, U, R, DEN, DEN1, EHIT1, TCOV, BLDGHT, PS,
2309.      X SEXP, SIN, COS, SIGRS, SIGDS, SIGH, SIGD, C1, C2, C3, S1, S2,
2310.      X X, Y, P, EM, DLS, DIL1, DIL2, DIV, TSRS, TSDS, PSR
2311.      C THE FOLLOWING JCL INSERTS TSARINA'S COMMON "BASIC"
2312.      // DD DSN=*.STEP1.COMMON,DISP=SHR
2313.      // DD *.DCB=BLKSIZE=800
2314.      DIMENSION ID(8), EHIT1(100), PSR(3), TCOV(2), BLDGHT(2)
2315.      C THIS ROUTINE ESTIMATES THE EXPECTED NUMBERS OF HITS FOR ATTACKS
2316.      C WITH POINT-IMPACT WEAPONS ONLY.
2317.      NCYCLE = 0
2318.      IF (KT*ST.GT.4) WRITE(6,1101) NTIRIAL, NPRINT, MODE
2319.      CCUNT = 0
2320.      PI = 3.14159
2321.      C
2322.      C EXAMINE THE ATTACKS ONE AT A TIME
2323.      C
2324.      DO 200 I = 1, NA
2325.      MM = ATT(I,9)
2326.      IF (END(MM,1,1) .GE. 0) GO TO 5
2327.      COUNT = COUNT + 1
2328.      IF (COUNT .EQ. 1) WRITE(6,1004)
2329.      GO TO 200
2330.      5 REL = #PANEL(MM)
2331.      NV = NTYPE(MM)
2332.      IF (NV .GT. 3) NV = 3
2333.      NFLAG = 0
2334.      NS = ATT(I,7)
2335.      LS = ATT(I,8)
2336.      PHI = ATT(I,1)/57.3
2337.      S = SIN(PHI)
2338.      C = COS(PHI)
2339.      IF ((ATT(I,1) .EQ. 0) .OR. (ATT(I,1) .EQ. 180)) GO TO 10
2340.      CT = C/S
2341.      GO TO 20
2342.      10 NFLAG = 1
2343.      20 CONTINUE
2344.      NP = NS
2345.      DLS = 0
2346.      IF (NP .EQ. 1) GO TO 24
2347.      DLS = LS/(NP - 1)
2348.      24 CONTINUE
2349.      X = ATT(I,2) - S*LS/2.
2350.      Y = ATT(I,3) - C*LS/2.
2351.      SIGRS = 2.200*ATT(I,4)*ATT(I,4) + ATT(I,6)*ATT(I,6)
2352.      SIGDS = 2.200*ATT(I,5)*ATT(I,5) + ATT(I,10)*ATT(I,10)
2353.      TSRS = 2*SIGRS
2354.      TSDS = 2*SIGDS
2355.      SIGH = SIGRS**(.5)
2356.      SIGD = SIGDS**(.5)
2357.      P = 1. / (SIGH*SIGD*6.2832)
2358.      IF (KTEST .GT. 7) WRITE(6,1008) I, LS, NP, DLS, X, Y, SIGH,
2359.      X SIGD, P
2360.      NR = 1
2361.      ENDI = END(MM, SHLT, 1)
2362.      28 CONTINUE
2363.      C
2364.      C FOR EACH TARGET, COMPUTE THE EXPECTED NUMBER OF HITS FOR ATTACK "I"
2365.      C
2366.      DO 180 L = 1, NT2
2367.      DEN = 0.0
2368.      IF ((NR.EQ.2) .AND. (ENDI.EQ.0)) GO TO 180
2369.      TGT1 = TGT2(L,12)
2370.      TGT2 = TGT2(L,13)
2371.      IF (ENDI .NE. 0) GO TO 30
2372.      EN = 0
2373.      GO TO 32
2374.      C
2375.      C THE AVERAGE HIT DENSITY IS DETERMINED BY AVERAGING THE HIT DENSITY
2376.      C OVER SEVERAL POINTS OF A RECTANGLE WHOSE DIMENSIONS ARE CHOSEN TO
2377.      C MAKE ITS AREA EQUAL IN SIZE TO THE AREA WITHIN WHICH THE TARGET
2378.      C COULD BE HIT, TAKING INTO ACCOUNT THE ROUNDED CORNERS OF THAT AREA.
2379.      C
2380.      30 C1 = (TGT1 + TGT2)/2.0
2381.      C2 = 3.1415926/4.0

```

```

2382.      C3 = - C1*ENDI - C2*ENDI*ENDI
2383.      EM = (-C1 + (C1*C1 - 4*C3)**(0.5))/2.0
2384. 32      TL1 = TGT1 + 2*EM
2385.      TL2 = TGT2 + 2*EM
2386.      TH = TGT2(L,9)/57.3
2387.      STG = SIN(TH)
2388.      CTG = COS(TH)
2389.      U = 0.0
2390.      V = 0.0
2391.      IF (EM .EQ. 0.0) GO TO 35
2392.      U = (STG + CTG) * EM
2393.      V = (STG - CTG) * EM
2394. 35      CONTINUE
2395.      XD(1) = TGT2(L,1) - U
2396.      XD(2) = TGT2(L,2) + V
2397.      XD(3) = TGT2(L,3) + V
2398.      XD(4) = TGT2(L,4) + U
2399.      XD(5) = TGT2(L,5) + U
2400.      XD(6) = TGT2(L,6) - V
2401.      XD(7) = TGT2(L,7) - V
2402.      XD(8) = TGT2(L,8) - U
2403.      CHI = TH - PHI
2404.      SC = SIN(CHI)
2405.      CC = COS(CHI)
2406.      S1 = (SIGRS*CC*CC + SIGDS*SC*SC)**(.5)
2407.      S2 = (SIGRS*SC*SC + SIGDS*CC*CC)**(.5)
2408.      N1 = (4*TL1/S1) + 1
2409.      N2 = (4*TL2/S2) + 1
2410.      IF (KTEST .GT. 9) WRITE(6,1009) N1, N2, S1, S2
2411.      IF ((N1 + N2) .GT. 2) GO TO 100
2412.      DO 90 K = 1,7,2
2413.      XT = XD(K)
2414.      YT = XD(K+1)
2415.      XL = X - XT
2416.      IF (NFLAG .EQ. 1) GO TO 40
2417.      YY = YT - Y + CT*XL
2418.      D = -YY*S
2419.      R = YY*C - XX/S
2420.      GO TO 50
2421. 40      D = XX
2422.      R = (YT-Y)*C
2423. 50      AID1 = D*U/TSDS
2424.      IF (AID1 .GT. 12.) GO TO 90
2425.      DD = SEXP(-AID1)
2426.      NCYCLE = NCYCLE + 1
2427.      DO 80 M = 1,NP
2428.      AID2 = R*M/TSRS
2429.      IF (AID2 .GT. 12.) GO TO 70
2430.      DR = SEXP(-AID2)
2431.      DEN = DEN + DR*DD
2432.      NCYCLE = NCYCLE + 1
2433.      IF (KTEST .GT. 10) WRITE(6,1005) I,L,K,XT,YT,D,R,DD,DR,DEN
2434. 70      R = R - ELS
2435. 80      CONTINUE
2436. 90      CONTINUE
2437.      DEN = DEN/(4.*NP)
2438.      GO TO 160
2439. C      IF THE TANGENT DIMENSIONS ARE SMALL (I.E. LESS THAN ONE-QUARTER
2440. C      THE PROJECTION OF SIGMA PARALLEL TO THE TARGET EDGE) THE HIT
2441. C      DENSITY IS TAKEN AS THE AVERAGE OF THE VALUES AT THE FOUR CORNERS.
2442. C      IF IT IS LARGER, A GRID OF INTERNAL POINTS IS ESTABLISHED AND
2443. C      THE HIT DENSITY IS TAKEN AS THE AVERAGE OVER THE INTERNAL POINTS.
2444. 100     CONTINUE
2445.      DIL1 = TL1/(N1+1)
2446.      DIL2 = TL2/(N2+1)
2447.      NN1 = N1 + 2
2448.      NN2 = N2 + 2
2449.      DO 150 M = 1,NN1
2450.      DO 150 N = 1,NN2
2451.      XT = XD(1) + (M-1)*DIL1*STG + (N-1)*DIL2*CTG
2452.      YT = XD(2) + (M-1)*DIL1*CTG - (N-1)*DIL2*STG
2453.      XL = X - XT
2454.      IF (NFLAG .EQ. 1) GO TO 110
2455.      YY = YT - Y + CT*XL
2456.      D = -YY*S
2457.      R = YY*C - XX/S
2458.      DO 10 120

```

```

2459. 110 J = XX
2460.      A = (Y1-Y)*C
2461. 120 AID1 = J*D/TSDS
2462.      IF (AID1.GT. 12.) GO TO 150
2463.      DD = SEXP(-AID1)
2464.      DIV = J.0
2465.      IF ((N.EQ.1).OR.(M.EQ.NN1)) DIV = 2.0
2466.      IF ((N.EQ.1).OR.(N.EQ.NN2)) DIV = DIV + 2.0
2467.      IF (DIV.NE. J.0) DD = DD/DIV
2468.      NCYCLE = NCYCLE + 1
2469.      DO 140 K = 1,NP
2470.      AID2 = R*A/TSDS
2471.      IF (AID2.GT. 12.) GO TO 130
2472.      DR = SEXP(-AID2)
2473.      DEN = DEN + DR*DD
2474.      NCYCLE = NCYCLE + 1
2475.      IF (KTEST.GT. 12) WRITE(6,1006) I,L,XT,YT,D,A,DD,DR,DEN
2476.      R = R - DCS
2477. 140 CONTINUE
2478. 150 CONTINUE
2479.      DEN = DEN/((1 + (N1 + N2) + N1*N2)*NP)
2480. 160 CONTINUE
2481.      DEN1 = DEN
2482.      DEN = NS*F*DEL*AIT(I,11)*DEN1/100.
2483.      IF (KTEST.GT. 7) WRITE(6,1007) I,L,DEN1,DEN
2484.      EHIT1(L) = DEN*1L1*TL2
2485.      COV2(L,NR) = COV2(L,NR) + EHIT1(L)
2486. 170 CONTINUE
2487. C
2488. C REPEAT FOR EACH TARGET
2489. C
2490.      IF ((NR.EQ. 2).OR.(NV.LT. 2)) GO TO 195
2491.      PSR(1) = (100. - EMD(NW,SHELT, 12))/100.
2492.      PSR(2) = (100. - EMD(NW,SHELT, 14))/100.
2493.      DO 190 L = 1,NT2
2494.      DO 190 N = 2,NV
2495. 190 P2(L,N+1) = 1. - (1. - P2(L,N+1))*(PSR(N-1)**EHIT1(L))
2496. C
2497.      IF (KTEST.GT. 4) WRITE(6,1011) PSR,(EHIT1(K),K=1,6)
2498. C
2499.      PSR(1) = (100.-EMD(NW,SHELT,3))/100.
2500.      PSR(2) = (100.-EMD(NW,SHELT,4))/100.
2501.      PSR(3) = (100.-EMD(NW,SHELT,5))/100.
2502.      DO 192 L = 1, NT2
2503.      DO 192 N = 6, 8
2504. 192 P2(L,N) = 1. - (1. - P2(L,N))*(PSR(N-5)**EHIT1(L))
2505. C
2506.      IF (KTEST.GT. 4) WRITE(6,1011) PSR,(EHIT1(K),K=1,6)
2507. C
2508. 195 CONTINUE
2509.      NR = NR + 1
2510.      EMD1 = EMD(NW,SHELT,10)
2511.      IF (NR.EQ. 2) GO TO 28
2512.      DO 198 L = 1,NT2
2513.      PS = (100. - EMD(NW,SHELT, 13))/100.
2514. 198 P2(L,5) = 1. - (1. - P2(L,5))*(PS**EHIT1(L))
2515. 200 CONTINUE
2516. C
2517. C REPEAT FOR EACH ATTACK
2518. C
2519.      DO 205 L = 1, NT2
2520.      P2(L,1) = 1. - SEXP(-COV2(L,1))
2521. 205 P2(L,2) = 1. - SEXP(-COV2(L,2))
2522.      IF (NPRINT.GT. 2) RETURN
2523.      WRITE(6,1000) NCYCLE
2524.      WRITE(6,1002)
2525.      NN = 0
2526.      TCOV(1) = 0.0
2527.      TCOV(2) = 0.0
2528.      BLDGHT(1) = 0.0
2529.      BLDGHT(2) = 0.0
2530.      DO 210 L = 1,NT2
2531.      NN = NN + 1
2532.      DO 208 I = 1,2
2533.      TCOV(I) = TCOV(I) + COV2(L,I)
2534. 208 BLDGHT(I) = BLDGHT(I) + (1. - SEXP(-COV2(L,I)))

```

```

2535.      WRITE (6,1003) L,COV2(L,1), COV2(L,2), (P2(L,J),J=1,8)
2536. 210 CONTINUE
2537.      BLDGHT(1) = BLDGHT(1)/NN
2538.      BLDGHT(2) = BLDGHT(2)/NN
2539.      WRITE (6,1010) TCOV(1), BLDGHT(1), TCOV(2), BLDGHT(2)
2540.      RETURN
2541. 1000 FORMAT ('0', 25X, 'CYCLES', 17, '//', ' ', 30X, '** SHELTERS ** ')
2542. 1002 FORMAT ('0', 10X, 'TARGET', 13X, 'HITS', 38X, 'PROBABILITIES', //, 12X,
2543. X 'NO', 14X, 'EXPECTED', 14X, 'HIT(R1) HIT(R2) AC(R1) SHELTER ',
2544. X 'AC(R2) PEOPLE AGE PARTS' )
2545. 1003 FORMAT (' ', 10X, 14, 6X, F6.3, 9X, F6.3, 6X, 8F8.3 )
2546. 1004 FORMAT ('1', ' COMPUTATION IGNORED THE CBU WEAPONS IN THE',
2547. X ' EXPECTED VALUE CALCULATION' )
2548. 1005 FORMAT(' ', 'CD ATT ', I3, ' TGT ', I3, ' COR ', I2,
2549. X 4F6.0, 3E12.4 )
2550. 1006 FORMAT (' ', 'IP ATT ', I3, ' TGT ', I3, 4F6.0, 3E12.5)
2551. 1007 FORMAT(' ', ' ATT ', I3, ' TGT ', I3, ' AVG DEN ',
2552. X F8.5, ' NOR DEN ', F14.10, //)
2553. 1008 FORMAT (' ', 'ATT ', I3, 2I8, F8.2, 4F8.1, E12.5)
2554. 1009 FORMAT(' ', 'N1 ', I4, ' N2 ', I4, 6X, 2F10.1)
2555. 1010 FORMAT (' ', 20X, '-----', //, 19X, F8.3, '(', F5.3, ')',
2556. X F8.3, '(', F5.3, ')')
2557. 1011 FORMAT(' ', 20X, 'PSR', 10F8.4 )
2558. 1101 FORMAT(' ', ' ***** EXPHIT ', 3I8 )
2559.      END

```

```

2561.      FUNCTION SEXP(X)
2562.      IF (X .LT. -0.025) GO TO 10
2563.      SEXP = 1.+X
2564.      RETURN
2565. 10 SEXP = EXP(X)
2566.      RETURN
2567.      END

```

```

2569.      SUBROUTINE STORE( ITEM, NCAEDS)
2570.      IMPLICIT INTEGER *2 (A-Z)
2571.      REAL *4 A
2572.      C THE FOLLOWING JCL INSERTS TSARINA'S COMMON "BASIC"
2573.      // DD DSN=*.STEP1.COMMON,DISP=SHR
2574.      // DD *,DCP=BLKSIZE=800
2575.      INTEGER *4 CLASS, TYPE
2576.      INTEGER *4 MXITEM, NOPEO, NOAGE, NOPART, NOMUN, NOTRAP, NOMATL, NOPOL
2577.      INTEGER *4 PEOPLE, AGE, PARTS, AMMO, TRAP, MATERL, POL, NOEQUI, NOPEOP
2578.      COMMON /STORES/ MXITEM, NOPEO, NOAGE, NOPEO, NOMUN, NOTRAP, NOMATL,
2579.      X NOPOL, NOEQUI, NOPEOP, PEOPLE(202,4), AGE(101,4), PARTS(401,4),
2580.      X AMMO(126,4), TRAP(26,4), MATERL(26,4), POL(10,4), STOCKS(1000,3),
2581.      X EQUIV(400)
2582.      DIMENSION A(5), D(15)
2583.      FLAG = 0
2584.      CAPD = 1
2585.      10 READ(5,1001) D(1),D(2),A(1),D(4),D(5),A(2),D(7),D(8),A(3),
2586.      X D(10),D(11),A(4),D(13),D(14),A(5)
2587.      1001 FORMAT( 5X, 5(3X,I2,I5,P5,C) )
2588.      DO 15 N = 1,5
2589.      IP (D(3*N-2) .EQ. 0) GO TO 16
2590.      15 D(3*N) = 10.0*(A(N) + C.0001)
2591.      MAX = 15
2592.      GO TO 17
2593.      16 MAX = 3*(N-1)
2594.      17 IF (NPRINT.LE.2) WRITE(6,1002) (D(N),N=1,MAX)
2595.      1002 FORMAT(' ',25X,5(' C',I3,2I5,4X) )
2596.      N = 1
2597.      20 CLASS = D(N)
2598.      TYPE = D(N+1) + 1
2599.      STOCKS(ITEM,1) = NT
2600.      STOCKS(ITEM,2) = D(N+2)
2601.      GO TO 50
2602.      C *****
2603.      ENTRY FILE(ITEM,CLASS,TYPE)
2604.      C *****
2605.      FLAG = 1
2606.      N1000 = 1000
2607.      IF (NPRINT.LE.2) WRITE(6,1002) CLASS, TYPE, N1000
2608.      TYPE = TYPE + 1
2609.      50 GO TO (100,200,300,400,500,600,700), CLASS
2610.      C
2611.      100 IF (TYPE .LT. 1000) GO TO 110
2612.      TYPE = TYPE - 999 + NOPEOP
2613.      110 NF = PEOPLE(TYPE,1)
2614.      IF (NF .NE. 0) GO TO 800
2615.      PEOPLE(TYPE,1) = ITEM
2616.      GO TO 1000
2617.      200 NF = AGE(TYPE,1)
2618.      IF (NF .NE. 0) GO TO 800
2619.      AGE(TYPE,1) = ITEM
2620.      GO TO 1000
2621.      300 NF = PARTS(TYPE,1)
2622.      IF (NF .NE. 0) GO TO 800
2623.      PARTS(TYPE,1) = ITEM
2624.      GO TO 1000
2625.      400 NF = AMMO(TYPE,1)
2626.      IF (NF .NE. 0) GO TO 800
2627.      AMMO(TYPE,1) = ITEM
2628.      GO TO 1000
2629.      500 NF = TRAP(TYPE,1)
2630.      IF (NF .NE. 0) GO TO 800
2631.      TRAP(TYPE,1) = ITEM
2632.      GO TO 1000
2633.      600 NF = MATERL(TYPE,1)
2634.      IF (NF .NE. 0) GO TO 800
2635.      MATERL(TYPE,1) = ITEM
2636.      GO TO 1000
2637.      700 NF = POL(TYPE,1)
2638.      IF (NF .NE. 0) GO TO 800
2639.      POL(TYPE,1) = ITEM
2640.      GO TO 1000
2641.      C
2642.      800 NX = STOCKS(NF,3)
2643.      IF (NX .EQ. 0) GO TO 900
2644.      NF = NX
2645.      GO TO 800

```

```

2646.      900   STOCKS(NF,3) = ITEM
2647.      1000  STOCKS(ITEM,3) = 0
2648.      C
2649.      IF (KTEST.GT.4) WRITE(6,1101) ITEM,CLASS,TYPE, (STOCKS(ITEM,K),K=1,3)
2650.      1101   FORMAT(' ', '      STORE ',10I8 )
2651.      C
2652.      C
2653.      ITEM = ITEM + 1
2654.      IF (FLAG.EQ. 1)      RETURN
2655.      N = N + 3
2656.      IF ((N.NE. 16) .AND. (D(N) .GT. 0))   GO TO 20
2657.      IF (CARD.EQ. NCARDS)  RETURN
2658.      CARD = CARD + 1
2659.      GO TO 10
2660.      END

```

```

2662.      SUBROUTINE SAVEDIFLAG, KTEST, MAXLOC )
2663.      IMPLICIT INTEGER *2 (A-Z)
2664.      INTEGER *4 FLAG, KIND
2665.
2666.      C
2667.      INTEGER *4 MXITEM, NCPEO, NCAGE, NOPART, NCMUN, NCTRAP, NOMATL, NOPOL
2668.      INTEGER *4 PEOPLE, AGE, PARTS, AMMO, TRAP, MATERL, POL, NOEQUI, NOPEOP
2669.      COMMON /STORES/ MXITEM, NCPEO, NCAGE, NOPART, NCMUN, NCTRAP, NOMATL,
2670.      X NOPOL, NOEQUI, NOPEOP, PEOPLE(232,4), AGE(101,4), PARTS(401,4),
2671.      X AMMO(126,4), TRAP(26,4), MATERL(26,4), POL(10,4), STOCKS(1000,3),
2672.      X EQUIV(400)
2673.      COMMON / ZERO / EQUI(7)
2674.
2675.      C
2676.      IF (FLAG .EQ. 1) GO TO 10
2677.      LOC = 1
2678.      CLASS = 0
2679.      LAST = 0
2680.      DO 5 I = 1,7
2681.      5 EQUI(I) = 0
2682.      RETURN
2683.
2684.      C
2685.      10 EQUIV(LOC) = 0
2686.      MAXLOC = LOC
2687.      RETURN
2688.      C
2689.      *****
2690.      ENTRY SAVE(CLAS, TYP, KIND)
2691.      C
2692.      *****
2693.      TYPE = TYP + 1
2694.
2695.      C
2696.      IF (KTEST .GT. 5) WRITE(6,1101) LOC,CLAS,CLASS,TYPE,LAST,KIND
2697.      1101 FORMAT(' ', SAVER ', 818 ')
2698.
2699.      C
2700.      IF (CLAS .NE. CLASS) GO TO 50
2701.      IF (TYPE .EQ. LAST) GO TO 800
2702.      EQUIV(LOC) = 0
2703.      LOC = LOC + 1
2704.      LAST = TYPE
2705.      CLASS = CLAS
2706.
2707.      C
2708.      GO TO (100,200,300,400,500,600), CLASS
2709.
2710.      C
2711.      100 IF (TYPE .LT. 1000) GO TO 110
2712.      TYPE = TYPE - 999 + NOPEOP
2713.      PEOPLE(TYPE,4) = LOC
2714.      GO TO 800
2715.      200 AGE(TYPE,4) = LOC
2716.      GO TO 800
2717.      300 PARTS(TYPE,4) = LOC
2718.      GO TO 800
2719.      400 AMMO(TYPE,4) = LOC
2720.      GO TO 800
2721.      500 TRAP(TYPE,4) = LOC
2722.      GO TO 800
2723.      600 MATERL(TYPE,4) = LOC
2724.      GO TO 800
2725.
2726.      C
2727.      800 EQUIV(LOC) = KIND
2728.      IF (KIND .EQ. 0) EQUI(CLASS) = 1
2729.      LOC = LOC + 1
2730.      IF (LOC .LT. NOEQUI) RETURN
2731.      WRITE(6,1001) CLAS, TYP, KIND
2732.      1001 FORMAT(' ', 'THE EQUIVALENCE ARRAY IS TOO SMALL; OVERFLOWED ',
2733.      X 'AT ',318 )
2734.      STOP
2735.      END

```

```

2727.      SUBROUTINE DAMAGE
2728.      IMPLICIT INTEGER *2 (A-Z)
2729.      INTEGER *4 NBASE1, NBASE2, NSTAT, AREA, TOTAL, NABRA, DISK, MXMT
2730.      INTEGER *4 NITEM, NOPEO, NOAGE, NOPAST, NOMUN, NOTRAP, NOMATL, NOPOL
2731.      INTEGER *4 PEOPLE, AGE, PARTS, AMMO, TRAP, MATERL, POL, NOEQUI, NOPEOP
2732.      REAL *4 PK, STAT, TOTCO1, TOTCO2, TOTCO3, TOTCO4, TOTCO5, TOTCC6
2733.      C THE FOLLOWING JCL INSERTS TSARINA'S COMMON "BASIC"
2734.      // DO DSN=*.STEP1.COMMON,DISP=SHR
2735.      // DC *,DCB=BLKSIZE=800
2736.      COMMON /STORES/ NITEM, NOPEO, NOAGE, NCPART, NOMUN, NOTRAP, NOMATL,
2737.      X NOPOL, NOEQUI, NOPEOP, PEOPLE(202,4), AGE(101,4), PARTS(401,4),
2738.      X AMMO(126,4), TRAP(26,4), MATERL(26,4), POL(10,4), STOCKS(1000,3),
2739.      X EQUIV(400)
2740.      COMMON /STATS / STAT(500,17), NSTAT(8), NTRIAL, ITRIAL, NSTAT
2741.      COMMON /OUTPUT/ TSAR, PUNCH, NBASE1, NBASE2
2742.      DIMENSION DATA(15)
2743.      C
2744.      IF ((NPRINT .GE. 3) .AND. (ITRIAL .EQ. 1)) WRITE(6,2222)
2745.      2222 FORMAT('1')
2746.      C
2747.      C *****
2748.      C
2749.      C THE OUTPUTS FOR TSAR MAY BE PRINTED, PUNCHED ON CARDS AND FILED
2750.      C DIRECTLY ON DISK FOR PRODUCTION RUNS. THE 2ND AND 3RD OPTIONS
2751.      C ARE CONTROLLED BY THE VARIABLES PUNCH AND TSAR, RESPECTIVELY,
2752.      C AND ARE EXERCISED WHEN THE VARIABLES ARE INITIALIZED TO UNITY.
2753.      C
2754.      C *****
2755.      MXMT = NTH
2756.      DISK = ITRIAL + 20
2757.      CARDTY = 40
2758.      JBASE = BASE
2759.      DC 10 K = 1,15
2760.      10 DATA(K) = 0
2761.      DATA(1) = DAY
2762.      DATA(2) = HOUR
2763.      DATA(3) = MINUTE
2764.      C
2765.      IF (PDAN.GT.1) WRITE(6,7777) CARDTY,JBASE,(DATA(K),K=1,15)
2766.      IF (TSAR.EQ.1) WRITE(DISK) CARDTY,JBASE,(DATA(K),K=1,15)
2767.      IF (PUNCH.EQ.1) WRITE(7,8888) ITRIAL,JBASE,NBASE1,NBASE2,DAY,HOUR
2768.      IF (PUNCH.EQ.1) WRITE(7,8888) CARDTY,JBASE,(DATA(K),K=1,15)
2769.      C
2770.      C RECORD DAMAGE SUSTAINED BY VARIOUS RESOURCE STORAGE AREAS
2771.      C
2772.      CALL IREPT(MXMT, NITEM, NOEQUI, STOCKS(1,1), P(1,1), TGI(1,1),
2773.      X EQUIV(1), NOPEOP, PCAN, KTEST)
2774.      C
2775.      CALL RREPORT(1, PEOPLE(1,1), NOPEO, 3)
2776.      CALL RREPORT(2, AGE(1,1), NOAGE, 4)
2777.      CALL RREPORT(3, PARTS(1,1), NCPART, 5)
2778.      CALL RREPORT(4, AMMO(1,1), NOMUN, 6)
2779.      CALL RREPORT(5, TRAP(1,1), NCTRAP, 7)
2780.      CALL RREPORT(6, MATERL(1,1), NCMATL, 8)
2781.      CALL RREPORT(7, POL(1,1), 10, 6)
2782.      C
2783.      C PREPARE TSAR INPUT DATA FOR THE SHOPS
2784.      C
2785.      DO 20 K = 1,3
2786.      20 DATA(K) = 0
2787.      JBASE = 0
2788.      N = 3
2789.      DO 40 SHOP = 1,30
2790.      I = FACLTY(SHOP)
2791.      IF (I.EQ.0) GO TO 40
2792.      IF (P(I,1).EQ.0.0) GO TO 40
2793.      IF (N.EQ.15) N = 3
2794.      DATA(N+1) = 9
2795.      DATA(N+2) = SHOP
2796.      DATA(N+3) = 100*(P(I,1) + 0.005)
2797.      DO 30 K = 4,6
2798.      30 DATA(N+K) = 100*(P(I,(K-1)) + 0.005)
2799.      N = N + 6
2800.      C
2801.      IF ((N.EQ.15) .AND. (PDAN.GT.1)) WRITE(6,7777) CARDTY,JBASE,
2802.      X (DATA(K),K=1,15)
2803.      IF ((TSAR.EQ.1) .AND. (N.EQ.15)) WRITE(DISK) CARDTY,JBASE,
2804.      X (DATA(K),K=1,15)
2805.      IF ((PUNCH.EQ.1) .AND. (N.EQ.15)) WRITE(7,8888) CARDTY,JBASE,
2806.      X (DATA(K),K=1,15)
2807.      C
2808.      40 CONTINUE
2809.      IF (N=9) 70, 50, 70

```

```

2810.      DO 60 K = 10,15
2811.      CC DATA(K) = 0
2812.      C
2813.      IF (PGAM.EQ.1) WRITE(6,7777) CARETY,JBASE,(DATA(K),K=1,15)
2814.      IF (TBASE.EQ.1) WRITE(DISK) CARETY,JBASE,(DATA(K),K=1,15)
2815.      IF (PUNCH.EQ.1) WRITE(7,8888) CARETY,JBASE,(DATA(K),K=1,15)
2816.      C
2817.      70 CONTINUE
2818.      C
2819.      C BEGIN TO ESTIMATE THE FRACTION OF SHELTERED AND
2820.      C OF PARTIALLY EXPOSED AIRCRAFT THAT ARE LOST.
2821.      C
2822.      TOTC1 = 0
2823.      TOTC2 = 0
2824.      TOTC3 = 0
2825.      TOTC01 = 0.0
2826.      TOTC02 = 0.0
2827.      TOTC03 = 0.0
2828.      TOTC04 = 0.0
2829.      TOTC05 = 0.0
2830.      TOTC06 = 0.0
2831.      C
2832.      C CHECK WHETHER SHELTERS ARE HANDLED WITH TGT2 CARDS OR INDIVIDUALLY
2833.      C
2834.      IF (INT2.EQ.0).AND.(NSM.EQ.0) GO TO 215
2835.      MAXS = N12
2836.      IF (INT2.EQ.0) MAXS = NSM
2837.      C
2838.      DO 210 K = 1,MAXS
2839.      TOTC01 = TOTC01 + P2(K,3)
2840.      TOTC02 = TOTC02 + P2(K,4)
2841.      TOTC03 = TOTC03 + P2(K,5)
2842.      IF (INT2.EQ.0) GO TO 200
2843.      C
2844.      TOTC04 = TOTC04 + P2(K,6)
2845.      TOTC05 = TOTC05 + P2(K,7)
2846.      TOTC06 = TOTC06 + P2(K,8)
2847.      GO TO 210
2848.      C
2849.      200 TOTC04 = TOTC04 + P(SHEL(K),3)
2850.      TOTC05 = TOTC05 + P(SHEL(K),4)
2851.      TOTC06 = TOTC06 + P(SHEL(K),5)
2852.      210 CONTINUE
2853.      C
2854.      IF (KTEST.GT.3) WRITE(6,1102) TOTC01, TOTC02, TOTC03,
2855.      X TOTC04, TOTC05, TOTC06, ACLCSS
2856.      1102 FORMAT(' ', 'DAMAGE-A', 6F8.4, 18 )
2857.      C
2858.      TOTC1 = (100*(TOTC01 + 0.005))/MAXS
2859.      TOTC2 = (100*(TOTC02 + 0.005))/MAXS
2860.      TOTC3 = (100*(TOTC03 + 0.005))/MAXS
2861.      C
2862.      C ESTIMATE THE PERCENTAGE LOSS TO EXPOSED AIRCRAFT AS A FRACTION
2863.      C OF THE RAMP SPACE THAT IS COVERED BY GP OR CPU EFFECTS.
2864.      C
2865.      215 TOTAL = 0
2866.      KAREA = 0
2867.      UNSHEL = 0
2868.      TOTHT = 0
2869.      C
2870.      DO 230 I = 1,NT
2871.      IF (TGT(I,10).NE. RANPS) GO TO 220
2872.      AREA = TOT(I,12)*TGT(I,13)
2873.      PK = 1. - (1.-P(I,5))
2874.      TOTAL = TOTAL + AREA
2875.      KAREA = KAREA + PK*AREA
2876.      220 CONTINUE
2877.      IF (TOTAL.GT.0) UNSHEL = (100*KAREA)/TOTAL
2878.      IF (TGT(I,10).NE. TAXWS) GO TO 230
2879.      TOTHT = TOTHT + WHIT(I)
2880.      230 CONTINUE
2881.      C
2882.      C RECORD THE REPAIRS ON THE LEAST DAMAGED RUNWAY AND
2883.      C THE TOTAL NUMBER OF HITS ON THE TAXIWAYS.
2884.      C
2885.      IF ((REPAIR + TOTHT).EQ.0) GO TO 250
2886.      DATA(4) = 9
2887.      DATA(5) = NOPAC
2888.      DATA(6) = REPAIR
2889.      C
2890.      DO 240 K = 7, 9
2891.      240 DATA(K) = 0
2892.      C

```

```

2893.      DATA(10) = 9
2894.      DATA(11) = WOPAC - 1
2895.      DATA(12) = TOTHT
2896.      C
2897.      IF (PDAM.GT.1) WRITE(6,7777) CABDTY,JBASE,(DATA(K),K=1,12)
2898.      IF (TSAR.EQ.1) WRITE(DISK) CABDTY,JBASE,(DATA(K),K=1,12)
2899.      IF (PUNCH.EQ.1) WRITE(7,8888) CABDTY,JBASE,(DATA(K),K=1,12)
2900.      C
2901. 250  CONTINUE
2902.      C
2903.      C STORE THE SHELTER AND AIRCRAFT DAMAGE DATA USING
2904.      C THE SPECIAL CLASS #10 ENTRIES FOR TSAR.
2905.      C
2906.      IF (TOTC1+TOTC2+TOTC3+UNSHL) .EQ. C) GO TO 310
2907.      DATA(4) = 10
2908.      DATA(5) = 1
2909.      DATA(6) = TOTC2
2910.      DATA(7) = TOTC3
2911.      DATA(8) = UNSHL
2912.      DATA(9) = 0
2913.      C
2914.      DATA(10) = 10
2915.      DATA(11) = 2
2916.      DATA(12) = ACLOSS
2917.      C
2918.      IF (PDAM.GT.1) WRITE(6,7777) CABDTY,JBASE,(DATA(K),K=1,12)
2919.      IF (TSAR.EQ.1) WRITE(DISK) CABDTY,JBASE,(DATA(K),K=1,12)
2920.      IF (PUNCH.EQ.1) WRITE(7,8888) CABDTY,JBASE,(DATA(K),K=1,12)
2921.      C
2922.      C STORE THE EXPECTED DAMAGE TO AIRCRAFT
2923.      C
2924.      DATA(4) = 8
2925.      DATA(5) = 0
2926.      DATA(6) = TOTC1
2927.      C
2928.      C STORE THE RESOURCE LOSS RATES EXPECTED WHEN AIRCRAFT ARE DAMAGED
2929.      C
2930.      C ESTIMATE THE LIKELIHOOD THAT PEOPLE, AGE AND SPARE PARTS
2931.      C ARE LOST WHEN AN AIRCRAFT IS DAMAGED AS THE RATIO OF THE
2932.      C PROBABILITY THAT THE RESOURCE IS LOST, DIVIDED BY THE
2933.      C PROBABILITY THAT THE AIRCRAFT IS LOST.
2934.      C
2935.      C DATA ARE ENTERED IN THE ORDER: PEOPLE, AGE AND PARTS
2936.      C
2937.      PEODAM = 0
2938.      AGEDAM = 0
2939.      PRTDAM = 0
2940.      C
2941.      IF (TOTC01 .EQ. 0.0) GO TO 290
2942.      C
2943.      PEODAM = ((100.*TOTC04)/TOTC01 + 0.5)
2944.      AGEDAM = ((100.*TOTC05)/TOTC01 + 0.5)
2945.      PRTDAM = ((100.*TOTC06)/TOTC01 + 0.5)
2946.      C
2947.      GO TO 300
2948.      C
2949. 290  CONTINUE
2950.      C
2951.      C IF THE PA AGAINST AIRCRAFT IN SHELTERS IS ZERO, USE THE
2952.      C FOLLOWING RELATIONSHIPS TO PROVIDE A PROXY FOR THE LOSS
2953.      C RATES FOR PERSONNEL, AGE AND PARTS ASSOCIATED WITH
2954.      C AIRCRAFT THAT ARE IN THE OPEN.
2955.      C
2956.      PEODAM = (100*END(1,SHELT,3))/END(1,SHELT,13)
2957.      AGEDAM = (100*END(1,SHELT,4))/END(1,SHELT,13)
2958.      AGEPR = (100*END(1,SHELT,5))/END(1,SHELT,13)
2959.      C
2960. 300  CONTINUE
2961.      C
2962.      IF (PEODAM .GT. 100) PEODAM = 100
2963.      IF (AGEDAM .GT. 100) AGEDAM = 100
2964.      IF (PRTDAM .GT. 100) PRTDAM = 100
2965.      C
2966.      DATA(7) = PEODAM
2967.      DATA(8) = AGEDAM
2968.      DATA(9) = PRTDAM
2969.      C
2970.      IF (PDAM.GT.1) WRITE(6,7777) CABDTY,JBASE,(DATA(K),K=1,9)
2971.      IF (TSAR.EQ.1) WRITE(DISK) CABDTY,JBASE,(DATA(K),K=1,9)
2972.      IF (PUNCH.EQ.1) WRITE(7,8888) CABDTY,JBASE,(DATA(K),K=1,9)
2973.      C
2974. 310  CONTINUE
2975.      C

```

```

2976.      IF (LAST.EQ. 0) RETURN
2977.      CARDTY = 0
2978.      IF (PDAM.EI.1) WRITE(6,7777) CARDTY
2979.      IF (ISAR.EI.1) WRITE(DISK) CARDTY
2980.      IF (PUNCH.EI.1) WRITE(7,8888) CARDTY
2981.      RETURN
2982.  7777  FORMAT(' ',I2,I4, 15I7)
2983.  8888  FORMAT('***** TRIAL',I4,'      BASE ',I3,4X,'NAME ',2A4,4X,
2984.  X      '      DAY HB', 2I4 )
2985.  8888  FORMAT(I2,I3,15I5)
2986.      END

```

```

2988.      SUBROUTINE REPORT(CLASS, A, MA, NR)
2989.      IMPLICIT INTEGER *2 (A-Z)
2990.      INTEGER *4 MSTAT, CLASS, MXNT, MAX, NOEQUI, NOPEOP, A, MA, NR, DISK
2991.      INTEGER *4 NBASE1, NBASE2
2992.      REAL *4 STAT, P, FRAC, TOT, TOT1
2993.      COMMON /STATS/ STAT(500,17), MSTAT(8), NTRIAL, ITRIAL, NSTAT
2994.      COMMON /OUTPUT/ TSAR, PUNCH, NBASE1, NBASE2
2995.      COMMON /ZFRO/ EQUI(17)
2996.      DIMENSION STOCKS(MAX,3), A(MA,4), DATA(15), P(MXNT,8),
2997.      X          TGT(MXNT,14), EQUIV(NOEQUI)
2998.      C *****
2999.      C
3000.      C THE OUTPUTS FOR TSAR MAY BE PRINTED, PUNCHED ON CARDS AND FILED
3001.      C DIRECTLY ON DISK FOR PRODUCTION RUNS. THE 2ND AND 3RD OPTIONS
3002.      C ARE CONTROLLED BY THE VARIABLES PUNCH AND TSAR, RESPECTIVELY,
3003.      C AND ARE EXERCISED WHEN THE VARIABLES ARE INITIALIZED TO UNITY.
3004.      C
3005.      C THE PRINT OPTIONS ARE CONTROLLED BY THE CONTROL VARIABLE "PDAM"
3006.      C
3007.      C *****
3008.      DISK = ITRIAL + 20
3009.      SKIP = 0
3010.      IF ((CLASS.NE.1).AND.(A(1,1).EQ.0).AND.(EQUI(CLASS).EQ.0)) SKIP=1
3011.      CARDTY = 40
3012.      JBASE = 0
3013.      M = 3
3014.      DO 10 K = 1,15
3015.      10 DATA(K) = 0
3016.      20 IF (MA .EQ. 1) GO TO 160
3017.      DO 150 I = 2, MA
3018.      IF (CLASS.NE.1) GO TO 30
3019.      IF (I .EQ. ALL) GO TO 150
3020.      30 ITEM = A(I,1)
3021.      TOT = 0.0
3022.      IF (ITEM.NE.0) GO TO 40
3023.      IF (A(I,4).EQ.0) GO TO 150
3024.      GO TO 60
3025.      40 NT = STOCKS(ITEM,1)
3026.      FRAC = STOCKS(ITEM,2)/10.0
3027.      TOT = TOT + FRAC*P(NT,NR)
3028.      ITEM = STOCKS(ITEM,3)
3029.      IF (ITEM.GT.0) GO TO 40
3030.      IF (TOT.EQ.0.0) GO TO 50
3031.      A(I,2) = A(I,2) + (10*TCT)
3032.      A(I,3) = A(I,3) + (100*TCT*TOT)
3033.      50 TYPE = I - 1
3034.      TOT1 = TOT/100.
3035.      TTYPE = TYPE
3036.      IF ((CLASS.EQ.1).AND.(TYPE.GT.NOPEOP)) TTYPE = 999+TYPE-NCPEOP
3037.      IF((PDAM.EQ.1).OR.(PDAM.EQ.3)) WRITE(6,7766) CLASS, TTYPE, TOT1
3038.      C
3039.      IF(KTEST.GT.1) WRITE(6,1101) I,CLASS,A(I,4),TOT
3040.      1101 FORMAT(' ', ' REPORT', 3110, F10.4)
3041.      C
3042.      C A REPORT OF ZERO LOSSES MAY BE REQUIRED FOR TSAR
3043.      C
3044.      IF (A(I,4).EQ.0) GO TO 100
3045.      IF ((SKIP.EQ.1).AND.(TOT.EQ.0.0)) GO TO 150
3046.      C
3047.      C WHEN THE RESOURCE TYPE SPECIFIED IN THE TSARINA TARGET ARRAYS
3048.      C IS A COMPOSITE OF SEVERAL TYPES USED IN TSAR, REPORT THE DAMAGE
3049.      C TO TSAR FOR EACH TYPE.
3050.      C
3051.      60 LOC = A(I,4)
3052.      70 KIND = EQUIV(LOC)
3053.      IF (KIND.LT.0) GO TO 150
3054.      75 IF (KTEST.GT.5) WRITE(6,1101) I, CLASS, KIND, TOT
3055.      DATA(M+1) = CLASS
3056.      DATA(M+2) = KIND
3057.      DATA(M+3) = (TCT + 0.5)
3058.      M = M + 4
3059.      IF (M.NE.15) GO TO 90
3060.      IF (PDAM.GT.1) WRITE(6,7777) CARDTY, JBASE, (DATA(K), K=1,15)
3061.      IF (TSAR.EQ.1) WRITE(DISK) CARDTY, JBASE, (DATA(K), K=1,15)
3062.      IF (PUNCH.EQ.1) WRITE(7,8888) CARDTY, JBASE, (DATA(K), K=1,15)
3063.      DO 80 K = 1,15
3064.      80 DATA(K) = 0
3065.      M = 3
3066.      90 LOC = LOC + 1

```

```

3067.      KIND = EQUIV(LOC)
3068.      IF (KIND .EQ. 3) GO TO 150
3069.      GO TO 75
3070.
3071. C
3072. 100 CONTINUE
3073.      IF (CLASS .NE. 1) GO TO 105
3074.      IF (I .GT. ALL) GO TO 110
3075. 105 IF ((TOT .EQ. 0.) .AND. (A(1,1) .EQ. 0.)) GO TO 150
3076.      GO TO 115
3077. 110 IF ((TOT .EQ. 0.0) .AND. (A(ALL,1) .EQ. 0.0)) GO TO 150
3078. 115 CONTINUE
3079. C
3080. C ONLY ONE TYPE OF POL IS PRESENTLY RECOGNIZED IN TSAR: TYPE #1.
3081. C DAMAGE DATA ON OTHER POL TYPES IS NOT STORED FOR USE BY TSAR.
3082. C
3083.      IF ((CLASS .EQ. 7) .AND. (TYPE .NE. 1)) GO TO 120
3084.      DATA(M+1) = CLASS
3085.      DATA(M+2) = TYPE
3086.      IF ((CLASS .EQ. 1) .AND. (I .GT. MAXPEO)) DATA(M+2) = 1200 + I - ALL
3087.      DATA(M+3) = (TOT + 0.5)
3088.      M = M + 6
3089. 120 IF (M .NE. 15) GO TO 150
3090.      IF (PDAM .GT. 1) WRITE(6,7777) CARDTY,JBASE,(DATA(K),K=1,15)
3091.      IF (TSAR .EQ. 1) WRITE(DISK) CARDTY,JBASE,(DATA(K),K=1,15)
3092.      IF (PUNCH .EQ. 1) WRITE(7,8888) CARDTY,JBASE,(DATA(K),K=1,15)
3093. 130 DO 140 K = 1,15
3094. 140 DATA(K) = 0
3095.      M = 3
3096. 150 CONTINUE
3097.      IF (M .NE. 9) GO TO 160
3098.      IF (PDAM .GT. 1) WRITE(6,7777) CARDTY,JBASE,(DATA(K),K=1,9)
3099.      IF (TSAR .EQ. 1) WRITE(DISK) CARDTY,JBASE,(DATA(K),K=1,9)
3100.      IF (PUNCH .EQ. 1) WRITE(7,8888) CARDTY,JBASE,(DATA(K),K=1,9)
3101. 160 CONTINUE
3102. C
3103.      DO 170 K=1,15
3104. 170 DATA(K) = 0
3105. C
3106.      M = 3
3107.      NN = 1
3108.      ND = 1
3109.      IF (CLASS .NE. 1) GO TO 180
3110.      NN = ALL
3111.      ND = MAXPEO
3112. 180 CONTINUE
3113. C
3114.      DO 250 I = 1, NN, ND
3115. C
3116.      DATA(M+1) = CLASS
3117.      ITEM = A(I,1)
3118.      IF (ITEM .EQ. 0) GO TO 240
3119. 190 NT = STOCKS(ITEM,1)
3120.      FRAC = STOCKS(ITEM,2)/10.0
3121.      TOT = TOT + FRAC*P(NT,NR)
3122.      ITEM = STOCKS(ITEM,3)
3123.      IF (ITEM .GT. 0) GO TO 190
3124.      IF (TOT .EQ. 0.0) GO TO 210
3125.      A(NN,2) = A(NN,2) + (10*TOT)
3126.      A(NN,3) = A(NN,3) + (100*TOT*TOT)
3127. 210 TOT1 = TOT/100.
3128.      IF (I .EQ. 1) GO TO 220
3129.      DATA(M+2) = 1000
3130.      M2 = 1000
3131.      IF ((PDAM .EQ. 1) .OR. (PDAM .EQ. 3)) WRITE(6,7766) CLASS,M2,TOT1
3132.      GO TO 230
3133. 220 DATA(5) = 0
3134.      IF ((PDAM .EQ. 1) .OR. (PDAM .EQ. 3)) WRITE(6,7766) DATA(4),DATA(5),TOT1
3135. 230 DATA(M+3) = (TOT + 0.5)
3136. 240 IF (A(I,1) .NE. 0) M = M + 6
3137. 250 CONTINUE
3138.      IF (M .EQ. 3) RETURN
3139.      IF (PDAM .GT. 1) WRITE(6,7777) CARDTY,JBASE,(DATA(K),K=1,12)
3140.      IF (TSAR .EQ. 1) WRITE(DISK) CARDTY,JBASE,(DATA(K),K=1,12)
3141.      IF (PUNCH .EQ. 1) WRITE(7,8888) CARDTY,JBASE,(DATA(K),K=1,12)
3142.      RETURN
3143. C *****
3144.      ENTRY IRFPT(MAX,MXNT,NCEQUI,STOCKS,P,TGT,EQUIV,
3145. X NCPEOP,PDAM,KTEST)

```

```

3146. C *****
3147. C
3148.      MAXPEC = NCPEOP + 1
3149.      ALL     = NCPEOP + 2
3150. C
3151.      RETURN
3152. C
3153.      7766 FORMAT(' ','CLASS ',I2,' TYPE',I5,F7.3)
3154.      7777 FORMAT(' ',I2,I4,I5I7)
3155.      8888 FORMAT(I2,I3,I5I5)
3156.      END

```

```

3158.      SUBROUTINE RESTAT(A,MA,N,NTRIAL)
3159.      IMPLICIT INTEGER *2 (A-Z)
3160.      INTEGER *4 NAME, A, MA, N
3161.      INTEGER *4 TYPES(10)
3162.      REAL *4 TRIAL, AVG, SD, K
3163.      REAL *4 DATA(10)
3164.      C
3165.      COMMON / LISTER / ISAVE
3166.      C
3167.      DIMENSION A(MA,3), NAME(2,7)
3168.      C
3169.      DATA NAME /'PEC','PLF','A','GE','PA','RTS','AM','MO',
3170.      X          'TR','AP','MAT','ERL','P','CL' /
3171.      C
3172.      C
3173.      IF (N.EQ. 1) WRITE(6,1001)
3174.      C
3175.      TALLY = 0
3176.      C
3177.      TRIAL = NTRIAL
3178.      K = 10.*TRIAL
3179.      IF (N.EQ. 1) NCPEOP = (MA - 2)/2
3180.      C
3181.      ITEM = A(1,1)
3182.      IF (ITEM.EQ. 0) GO TO 20
3183.      AVG = A(1,2)/K
3184.      SD = (A(1,3)/100. - TRIAL*AVG*AVG)/(TRIAL - 1.)
3185.      IF (SD.LT. 0) GO TO 5
3186.      SD = SD**(0.5)
3187.      GO TO 10
3188.      5 SD = -1
3189.      10 WRITE(6,1002) NAME(1,N),NAME(2,N), AVG, SD
3190.      20 FLAG = 0
3191.      DO 40 I = 2,MA
3192.      ITEM = A(I,1)
3193.      IF (ITEM.EQ. 0) GO TO 40
3194.      TYPE = I - 1
3195.      IF ((N.EQ. 1).AND.(TYPE.GT. NCPEOP)) TYPE = TYPE - NCPEOP
3196.      AVG = A(I,2)/K
3197.      SD = (A(I,3)/100. - TRIAL*AVG*AVG)/(TRIAL - 1.)
3198.      C
3199.      TALLY = TALLY + 1
3200.      TYPES(TALLY) = TYPE
3201.      DATA(TALLY) = AVG
3202.      C
3203.      IF (SD.LT. 0) GO TO 25
3204.      SD = SD**(0.5)
3205.      GO TO 30
3206.      25 SD = -1
3207.      30 IF (FLAG.EQ. 1) GO TO 35
3208.      WRITE(6,1003) NAME(1,N),NAME(2,N),TYPE,AVG,SD
3209.      FLAG = 1
3210.      GO TO 40
3211.      35 WRITE(6,1004) TYPE,AVG,SD
3212.      40 CONTINUE
3213.      C
3214.      IF (ISAVE.EQ. 0) RETURN
3215.      IF (TALLY.GT.0) WRITE(7,1111) N, ((TYPES(I),DATA(I)),I=1,TALLY)
3216.      C
3217.      RETURN
3218.      1001 FORMAT('1','RESOURCE TYPE AVERAGE STD DEV',/,
3219.      X          ' ',' CLASS LOSSES LOSSES ',/,
3220.      X          ' ',' PERCENT ',/ )
3221.      1002 FORMAT('0','1X,2A3,' ALL ',2(4X,F6.3) )
3222.      1003 FORMAT('0','1X,2A3,10.2X, 2(4X,F6.3) )
3223.      1004 FORMAT(' ','10X,15.2X, 2(4X,F6.3) )
3224.      1111 FORMAT('13, 2X,10(12, F5.2) )
3225.      END

```

Appendix D

PROGRAMMERS' NOTES

The TSARINA program, as listed in Appendix C, requires about 250K bytes (or 64K words) of core memory. This requirement is the sum of about 115K bytes for program logic and 135K bytes for data storage. If TSARINA is to be run on a computer system that does not support half-word FORTRAN integers (e.g., INTEGER*2) the space required for data storage will be approximately doubled.

If these core requirements are excessive for some installations, they can be reduced in two different ways. First, the program is readily adaptable to a relatively efficient overlay structure that would reduce the space requirements for the program logic. The structure listed below should not affect operating speed seriously and would cut the core required for program logic from about 115K to about 25K bytes:

```
      INSERT MAIN,GAUSS
OVERLAY ZONE
      INSERT INPUT,STORE,SAVER,JMEMO
OVERLAY ZONE
      INSERT BOMB,TESTHT,HITTGT,TGTDIM,TGTORD,TGTZON
OVERLAY ZONE
      INSERT CBU,EXPHIT
OVERLAY ZONE
      INSERT CHECKR,RUNWAY,PLOHT
OVERLAY ZONE
      INSERT PRINT,STATIS,DAMAGE,REPORT,RESTAT
```

The space required for data storage may also be reduced dramatically, depending upon the nature of the problem of interest to the user. As can be seen from a careful review of the code and comments in the first half of the MAIN routine (pp. 71, 72), the version listed in Appendix C provides space for a relatively large and complex problem: 500 targets, 50 attacks, 1000 resource packets, and hundreds of different types of resources of various classes. For lesser problems, the dimensions of the appropriate storage arrays can be reduced before the program is compiled. With the few exceptions noted in the MAIN routine all of these changes are confined to COMMON statements.

The same procedure can also be used, of course, to increase the size of the storage arrays, if the user's problem exceeds the bounds of TSARINAs current dimensions. With the exception of the numbers of different types of the different classes of resources, there is no practical limit on the size of the problem that can be treated, other than those imposed by available core and the user's budget.