NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

EVALUATION OF MULTI-LEVEL SUPPORT STRUCTURE
REQUIREMENTS FOR NEW WEAPON SYSTEMS

by

Rainer Seth

September 1987

Thesis Advisor: F. Russell Richards

Approved for public release; distribution is unlimited

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b RESTRICTIVE MARKINGS
2a SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; Distribution is unlimited
2b DECLASSIFICATION/DOWNGRADING SCHEDULE	
4 PERFORMING ORGANIZATION REPORT NUMBER(S)	5 MONITORING ORGANIZATION REPORT NUMBER(S)
6a NAME OF PERFORMING ORGANIZATION Naval Postgraduate School; 6b OFFICE SYMBOL (If applicable) 55	7a NAME OF MONITORING ORGANIZATION Naval Postgraduate School
6c ADDRESS (City, State, and ZIP Code) Monterey, California 93943-5000	7b ADDRESS (City, State, and ZIP Code) Monterey, California 93943-5000
8a NAME OF FUNDING/SPONSORING ORGANIZATION; 8b OFFICE SYMBOL (If applicable)	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c ADDRESS (City, State, and ZIP Code)	10 SOURCE OF FUNDING NUMBERS: PROGRAM ELEMENT NO / PROJECT NO / TASK NO / WORK UNIT ACCESSION NO

11 TITLE (Include Security Classification)
Evaluation of Multi-Level Support Structure Requirements for New Weapon Systems

12 PERSONAL AUTHOR(S)
SETH, Rainer

13a TYPE OF REPORT	13b TIME COVERED	14 DATE OF REPORT (Year, Month Day)	15 PAGE COUNT
Master's Thesis	FROM ____ TO ____	1987 September	302

16 SUPPLEMENTARY NOTATION

17 COSATI CODES			18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Logistics, Multi-Level Support, Support Structures Requirements

19 ABSTRACT (Continue on reverse if necessary and identify by block number)

This thesis presents a computer program, CAESAR, for IBM XT/AT or compatible computers to compute the logistics demands of a new system in terms of manpower and supply.

CAESAR, (Computer Aided Evaluation of Supply and Support on All Repair Levels), uses inputs such as weapon system distribution, system characteristic data and major unit data to provide conclusions about the logistics behavior of failing weapon systems.

The modeling of system behavior with CAESAR has several advantages over the empitical approach. It can be utilized at any major unit level, and it can be used to simulate repair part consumption of weapon systems prior to their introduction into the Armed Forces.

Continued

20 DISTRIBUTION/AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	21 ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED
22a NAME OF RESPONSIBLE INDIVIDUAL F. Russell Richards	22b TELEPHONE (Include Area Code) 408-372-3439; 22c OFFICE SYMBOL 55Rh

DD FORM 1473, 84 MAR 83 APR edition may be used until exhausted. All other editions are obsolete

SECURITY CLASSIFICATION OF THIS PAGE
Unclassified

19. Abstract (continued)

CAESAR is not an optimizer. Instead, it is an evaluator to be used to assist decision making. It facilitates the examination of various logistics trade offs. Thus, it can be a valuable tool for managers responsible for introducting new equipment and for defining the accompanying support network.

Evaluation of Multi-Level Support Structure Requirements for New Weapon Systems

by

Rainer Seth
Captain, German Army
M.S. Mechanical Engineering
Fachhochschule des Heeres, Darmstadt, 1975

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
September 1987

ABSTRACT

This thesis presents a computer program, CAESAR, for IBM XT/AT or compatible computers to compute the logistics demands of a new system in terms of manpower and supply.

CAESAR, (Computer Aided Evaluation of Supply and support on All Repair levels), uses inputs such as weapon system distribution, system characteristic data and major unit data to provide conclusions about the logistics behavior of failing weapon systems.

The modelling of system behavior with CAESAR has several advantages over the empirical approach. It can be utilized at any major unit level, and it can be used to simulate repair part consumption of weapon systems prior to their introduction into the Armed Forces.

CAESAR is not an optimizer. Instead it is an evaluator to be used to assist decision making. It facilitates the examination of various logistics trade offs. Thus, it can be a valuable tool for managers responsible for introducing new equipment and for defining the accompanying support network.

TABLE OF CONTENTS

LIST OF FIGURES

I. THE PROBLEM

With the number of new systems that are introduced to replace outdated or worn-out equipment and with increasing monetary constraints, it is of paramount interest to know the logistics requirements of the new systems well in advance of actual deployment. Of specific interest are manpower requirements on the different maintenance levels and the stock requirements for spare parts at the different supply levels.

This study attempts to provide the decision maker with an array of options from which he can pick the one that looks favorable both as a reasonable compromise between combat readiness and budget constraints and as an acceptable choice for the units in the field.

Obviously, it would be beyond such a study to guarantee optimal solutions for all kinds of different situations. This thesis will not consider initial introduction of weapon systems or surge conditions brought about by increased usage of the equipment. Thus, the problem we consider involves the "sizing" of the logistics network for routine operations.

For this "steady state" case the output will suggest necessary, required, or otherwise pertinent data to plan the successful build-up of an efficient logistics network.

The output of CAESAR yields the following information:

(1) the number of hours spent on all the jobs,

(2) the manpower required to accomplish these jobs on the different levels,

(3) the numbers of repair parts for all the jobs,

(4) recommended supply stocks to cover the demand.

Further, the consumption behavior with suggested stock levels will be analyzed.

Failing systems are modelled as customers of a queuing network. This network extends through several maintenance levels. Sensitivity analysis is done in terms of manpower requirements. The sensitivity analysis considers, if possible, at least a 10 per cent bandwidth on both sides of an initial computed value. All possible combinations of strengths on all repair levels within these bands are computed. This gives the decision maker the opportunity to compare different scenarios and to consider different trade-offs.

Figure 1.1 demonstrates the simplified structure within the supporting organizations and the interactions among the user in the field and the supporting organizations. Assume the user, depicted by the tank, faces a failure of one of his weapon systems.

He will notify his organizational maintenance unit and after a delay, due to administrative and transportational times, the system will undergo repair by either changing the component that failed or by repairing this component. After another delay the system will be returned to the user and will be operational again. If the component was replaced, it will enter the next level repair cycle at the direct / general support unit, where the subcomponent will be replaced or repaired. The operational component is returned to the organizational maintenance unit. If further repair is necessary, the subcomponent is shipped to the depot / industry level where it is finally repaired. It will then be shipped back to the direct / general maintenance unit to wait for the next exchange cycle.

Usually, the components and subcomponents can be used again after repair. Only parts have to be provided through the supply chain. The model will not simulate total loss of components or subcomponents but it will provide a number of available components and subcomponents on the respective maintenance levels for direct exchange.

The structures of the underlying maintenance and supply organizations are described in more detail in Chapter C.2. and C.3.

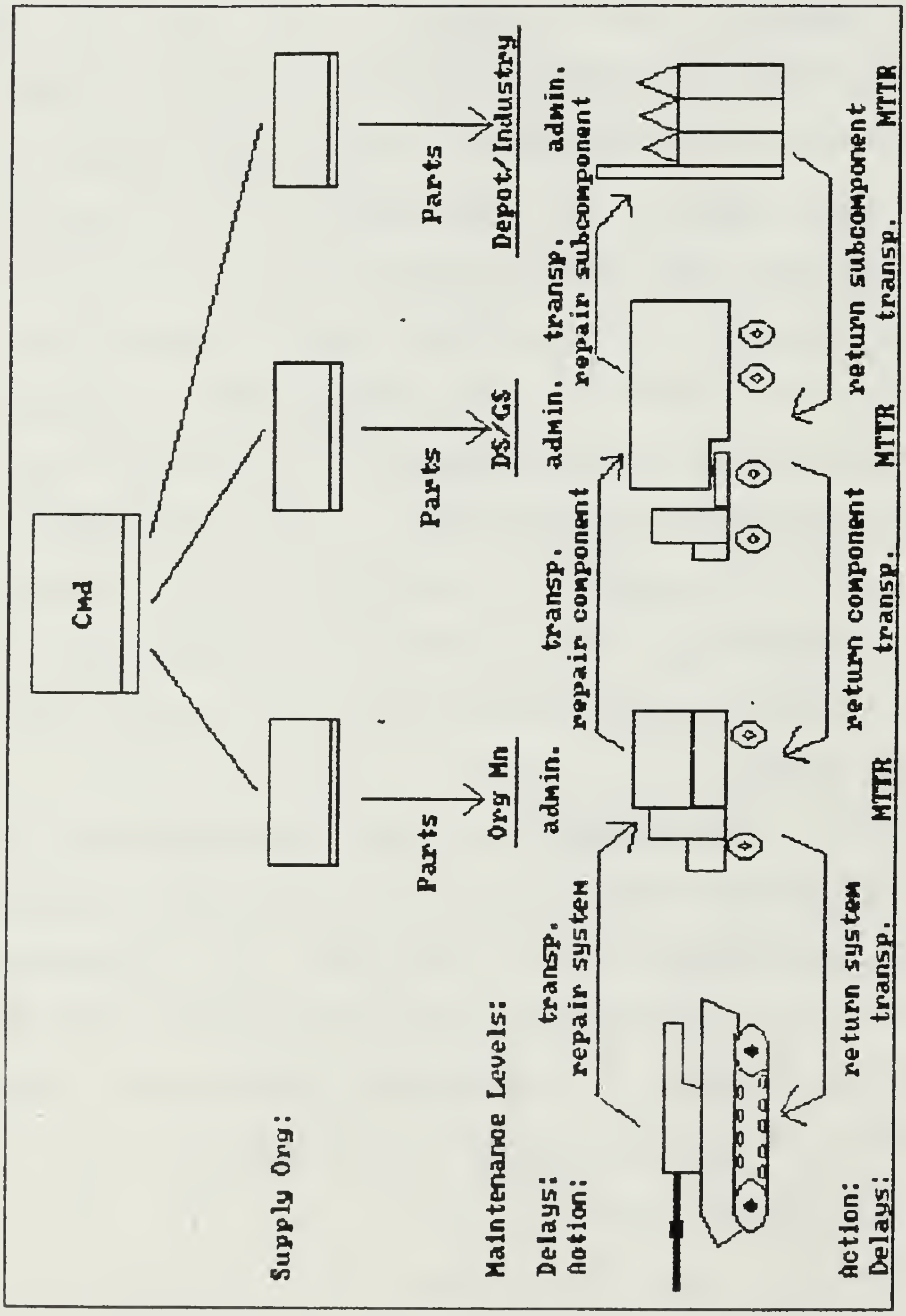


Figure 1.1 Support Structure

II. THE THEORY

A. DEFINITIONS

The following definitions are provided to make clear the terminology used in this thesis. Refer to Figure 2.1 for a graphical depiction of a weapon system.[Ref. 1]

System is the entity of all replaceable or repairable components, subcomponents and parts that make up the working piece of equipment.

Component is the first split-up level of replaceable or repairable items within a system.

Subcomponent is any item below the component level which is still repairable.

Parts are non-repairable items within a system, component or subcomponent.

Refer to Figure 2.3 for a graphical depiction of the underlying maintenance organization.

Organizational Maintenance (Org Mn) is the first repair level. It is usually performed at the battalion level.

Direct Support/General Support (DS/GS) is the next level above Org Mn. This is performed at the brigade, division or corps level for the respective supported units.

Depot Maintenance is the highest level of maintenance. It is usually performed by the military depot organization or a civilian contractor.

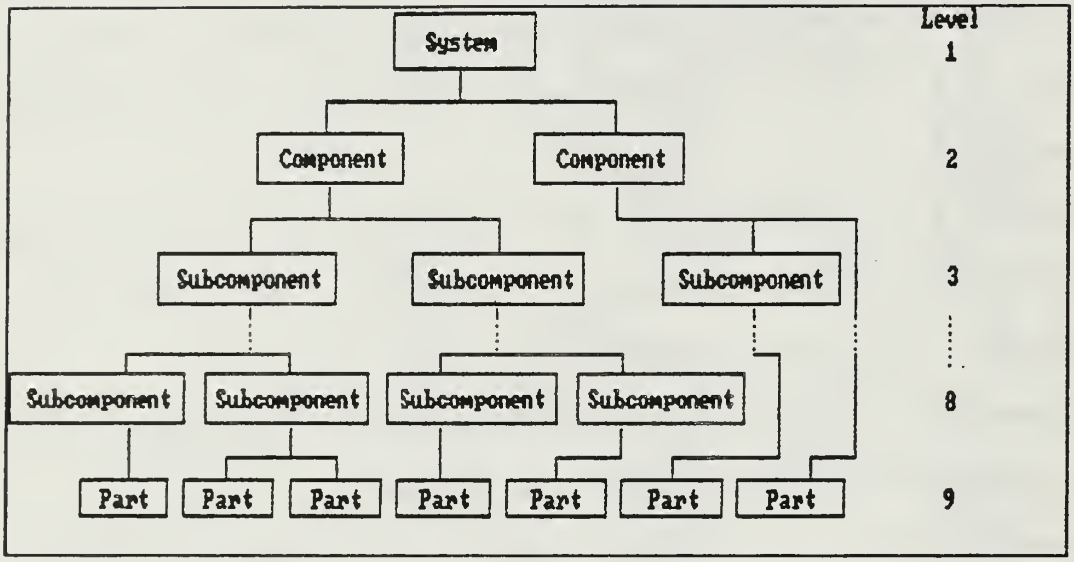


Figure 2.1 Tree - Structure of a Weapon System

Assignments of repair jobs to one of the repair levels is done on the basis of their need for expertise, special tools, amount of hours necessary for repair or severity of damage.

It may be possible that one level will be skipped or that work at a lower level is performed in connection with work on a higher level. This aspect will be disregarded in this study.

Refer to Figure 2.4 for a graphical depiction of the structure of the underlying supply organization.

Direct Supply Organization (DSO) is the supply unit which is responsible for the support of specific customers within the major unit.

Back-up Supply Organization (BSO) is a supply unit which can support another supply unit on its supply level.

Initial Demand Generator (IDG) is the customer of a DSO who demands service from the DSO.

Intermediate Supply Organization (ISO) is any supply unit in the supply channel that both accepts inputs of spares and passes on output of spares to other ISOs or IDGs.

End Supply Organization (ESO) is the facility where all orders that cannot immediately be filled or orders of restocks for ISO are centralized. It does not move physical items; it coordinates and routes the orders.

Mean Time Between Failures (MTBF) is the expected value of the interarrival time of failures for a working system. In this thesis equipment lifetimes are assumed to be exponentially distributed.

Mean time To Repair (MTTR) is the expected value of the time required to repair a failed equipment.

B. ASSUMPTIONS

1. Independence of Systems

Systems in any specific major unit are assumed to be independent of each other.

2. No Non-lethal Failures

Each item is assumed equally vital for the operation of the system. A failure in any part will lead to a failure for the complete system.

3. Unlimited Storage Capabilities

It is assumed that supply units can store any suggested number of spare parts. For the first simulation, unlimited supplies are assumed.

Later, limited numbers of spares are considered.

4. Hierarchical Logistical Structure

The underlying logistical system is assumed to be strictly hierarchical. There is no horizontal substitution for maintenance nor vertical substitution for supplies between different major units in the model.

C. THE LOGISTICS STRUCTURE

1. System Lay-out Dependencies

The physical configuration of any new system depends on many different parameters. Cornerstones for the design are the tradeoffs due to budget and the operability and system availability in the field.

These connections between the various aspects are displayed in Figure 2.2. If enough resources (budget) are available, then the quantity of components for a new system can be optimized, and where possible, the idea of modular system layout can be realized. The number of servers can be tailored to satisfy maximum demand. All these means lead to a reduction in system repairtime. This and the ability to subsidize higher quality material, which increases the time until a failure occurs, yields less stress upon the logistics organization. The resulting increase in system availability is clearly in favor of the user in the field. In reality, however, different users formulate priorities for acquisitions which demand resources. These demands have to be met as well as possible. Thus tradeoffs between optimality and feasibility have to be made.

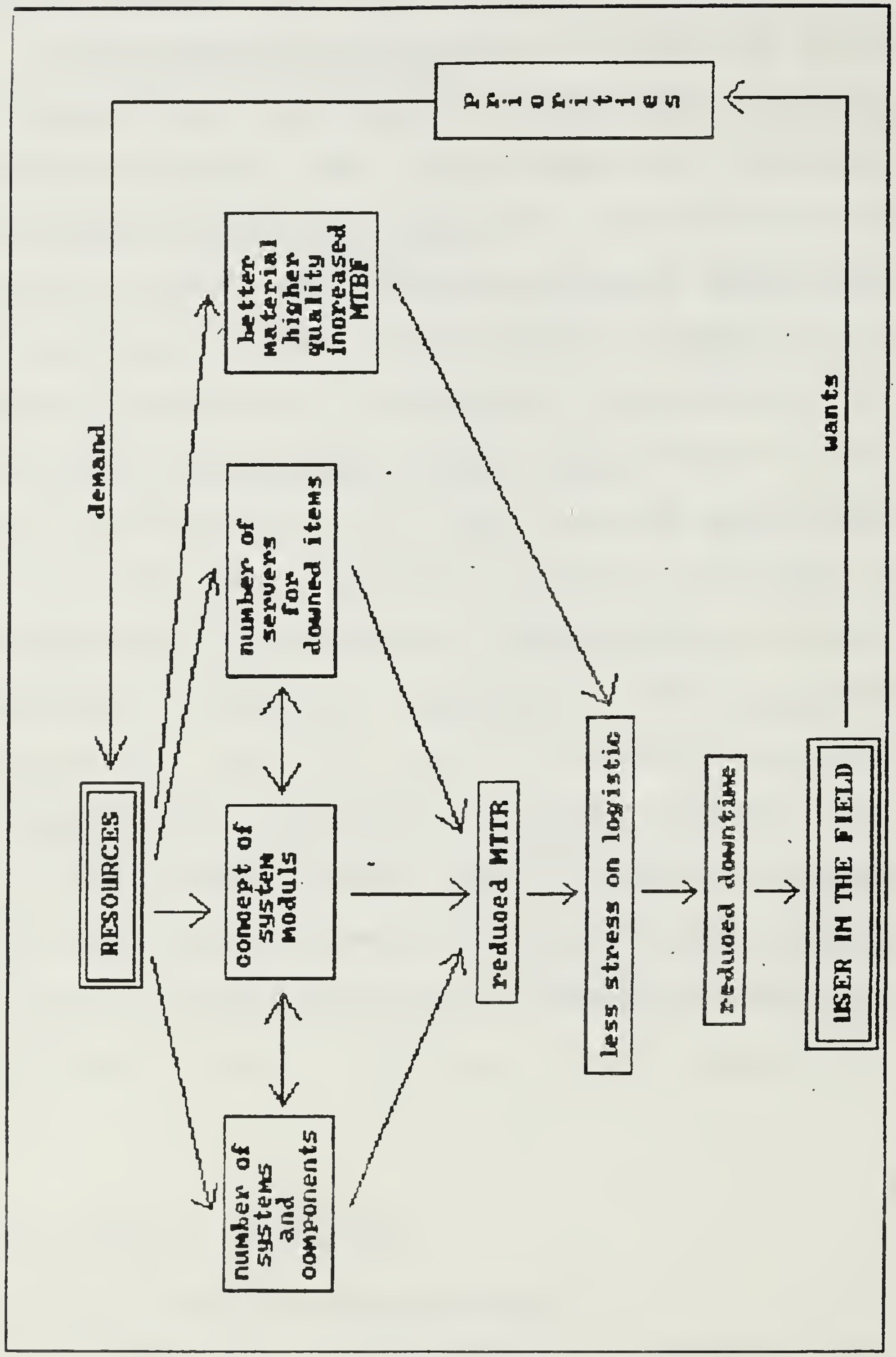


Figure 2.2 System Interdependencies

2. Underlying Maintenance Organization

As stated earlier, the maintenance structure is assumed strictly hierarchical. Each level serves specified jobs for specific customers. An exchange of capability cannot regularly be performed between different maintenance units. Figure 2.3 shows this assumed structure.

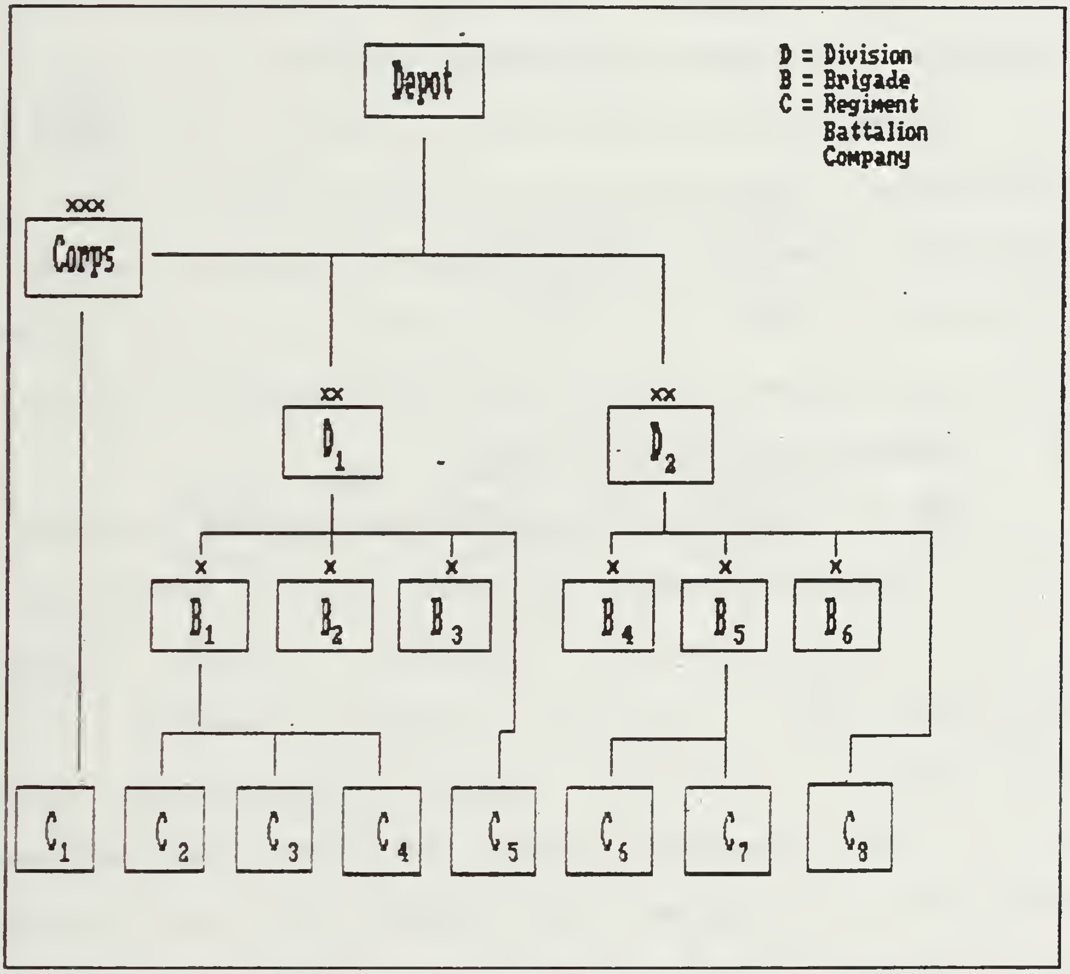


Figure 2.3 Schematic Maintenance Structure

In the example depicted in Figure 2.3 it is assumed that units (customers) C_1 to C_8 perform their own organizational maintenance. In case of a failure that requires direct support, the destination of the item depends on the type of customer and his supporting unit.

C_1 is supported directly (direct support, DS) by the corps unit. C_2,C_3,C_4 and C_6,C_7 are units which are supported by their respective brigades. C_5 and C_8 are supported by their divisional units.

In case of an overload of work at the DS level, the divisional units can back up the brigade units, and the corps can back up the divisions (general support, GS). However, this is only possible within certain limitations. The model does not allow general support.

3. Underlying Supply System

The structure assumed for the supply system is similar to the maintenance system. In this structure it will be distinguished between Direct Supply Organization (DSO), which is directly responsible for its customers and back-up Supply Organization (BSO), which is, within limits, capable of supplying customers of BSO units, if they are not immediately able to serve their customers. The initial demand generator (IDG) is the customer for the supply system, which is usually the maintenance unit.

Figure 2.4 shows a typical structure for a supply network.

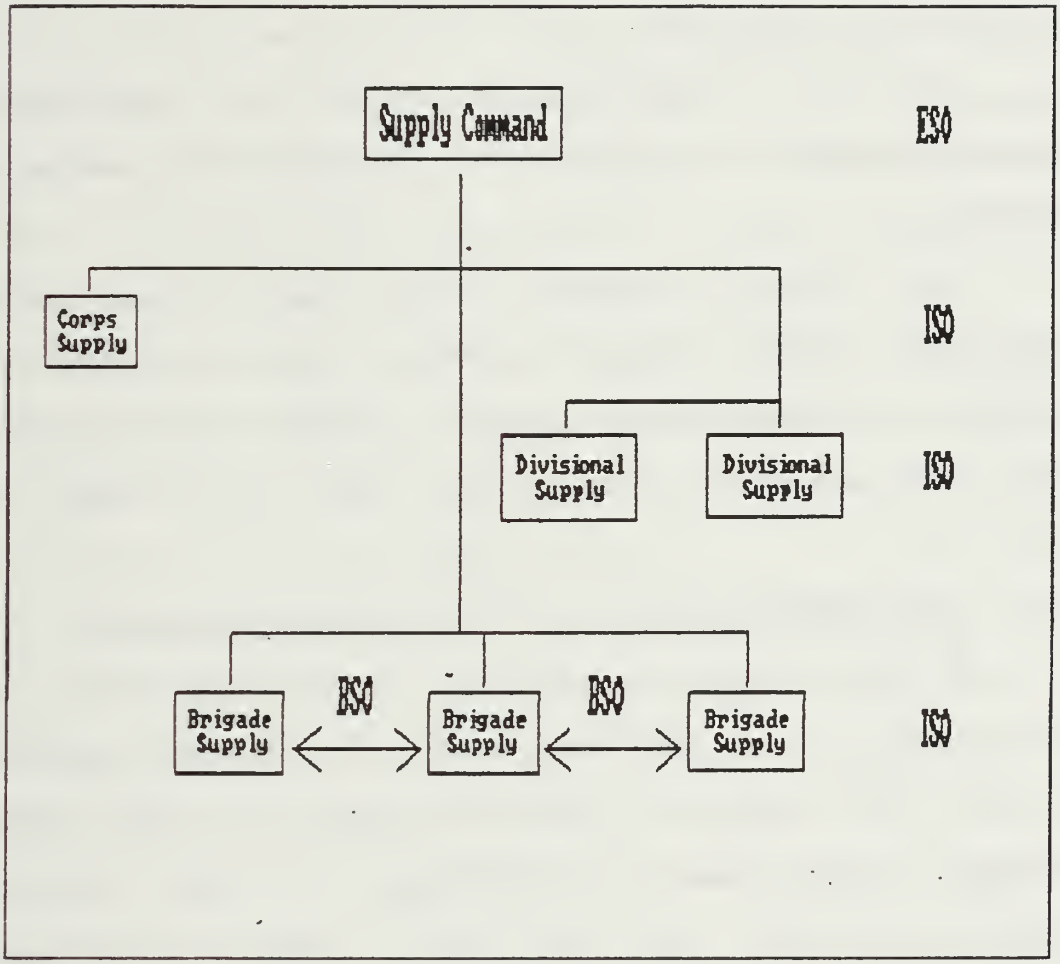


Figure 2.4 Schematic Structure of the Supply System

The Supply Command supports all supply units on corps, division and brigade levels in a parallel structure. Since the customers on these levels are all similarly structured, i.e. have the same equipment, type and density, a horizontal back-up (BSO) on the same level is possible.

Vertical substitution would be ineffective because customers at different levels have dissimilar equipment.

The Supply Command works as an end supply organization (ESO) since its job is it to provide the intermediate supply organizations (ISO) with the needed items. The customers, the IDGs, are not shown in Figure 2.4.

4. Cooperation within the Logistics Network

A user reports a failure of a system to his own Org Mn activity. At the same time, the user becomes an IDG for the supply network since a replacement component or part must be provided. If there is stock available at the Org Mn level, and if maintenance capability is also available, the customer is notified that the equipment will be worked on. If there are no spares on hand, the ISO will get spares for the IDG either from the ESO or the BSO.

After successful repair and delivery the system will be operational and functional and the first repair cycle (see Figure 1.1) is complete. The Org Mn unit becomes the customer for the DS unit with the broken component. This DS unit as the IDG will approach its supply unit to obtain the necessary items until the second repair cycle is completed. From here the same procedure continues at the depot level with its supply channel where the final broken item will be exchanged or repaired, if required. On either level the repair has been finished when the level 9 (non- repairable) item has been reached.

Should Org Mn not have enough manpower available then repairs might be passed on to the GS level. GS level is the division for brigade units or the corps for the divisional units.

Although the time for immediate repair and return of the system to its normal duty is of particular interest, it is further important to consider the whole chain of repairs. These repairs have an impact on the availability of spares under the assumption of limited supplies in the evaluation later in the program package.

Replacements have to be ordered for the parts, since these do not get back into circulation like components or subcomponents, which after repair regain their place on the shelves of the maintenance units. This inventory policy can be on a one-by-one basis, regularly according to fixed time intervals or upon reaching a minimum number of stocked items.

In this study, the restocking policy will be the (s,S) method. It will be explained in detail later. The number of restocked spare parts depends on the length of the chosen time interval and the demand within this interval.

D. THE COMPUTATIONAL FORMULAE

The simulation consists of two parts. First, in an idealized run, planning figures will be computed assuming unlimited resources of both manpower and spare parts. Thus waiting times in the system are considered to be zero. In the second run the planning figures of the idealized run are used and the actual queuing behavior with limited servers and spare parts is analyzed. In this run the waiting times are greater than zero. The major computational formulas used in the simulation are presented in the following paragraphs.

1. Random Number Generation

The basis for all the random numbers used in CAESAR are the uniform(0,1) random numbers [Ref. 2].

a. Uniform Random Number

To transform the uniform(0,1) random number to a uniform(LL,UL) number the following algorithm is used:

$U(LL,UL) = LL + (UL-LL) * U(0,1)$.

where
$LL = mean - SQRT[\ 3 * (std.dev.)^2]$
$UL = mean + SQRT[\ 3 * (std.dev.)^2]$

b. Exponential Random Number

To transform the uniform(0,1) random number to an exponential random number the following algorithm is used [Ref. 3] :

$$EXP = -\ (\ 1/lambda\ ) * \ln [\ U(0,1)\ ]$$

with
EXP ..... exp. distributed random variable
lambda .. parameter for exp. distr. variable
ln ...... natural lgarithm
U(0,1) .. uniform(0,1) random number.

where
Lambda = 1/mean

c. Gamma Random Number

A gamma (Erlang) random number with n degrees of freedom is generated from the following algorithm [Ref. 3] :

$$GAMMA = -\ (\ 1/lambda\ ) * \ln [\ \underset{n}{prod}[\ U(0,1)\ ]$$

with
GAMMA ... Gamma distributed random variable
lambda .. parameter for Gamma distr. variable
n .... degrees of freedom
ln ...... natural logarithm
U(0,1) .. uniform(0,1) random number
$\underset{n}{prod}[\ ]$ ... product from 1 to n of U(0,1).

where
$lambda = mean\ /\ (\ std.dev.)^2$
$n = (\ mean\ /\ std.dev.)^2$

d. Normal Random Number

To obtain a normally distributed random number a Box - Mueller algorithm is used [Ref. 3] :

$$N_1 = \{SQRT(-2 * \ln[U_1(0,1)])\} * \cos[2*pi*U_2(0,1)]$$

$$N_2 = \{SQRT(-2 * \ln[U_1(0,1)])\} * \sin[2*pi*U_2(0,1)].$$

with
- N_1,N_2 .. the normally distributed r.n.
- SQRT() . square root of the argument ()
- ln ..... natural logarithm
- pi 3.141592654
- cos() .. trigonometric cosine function
- sin() .. trigonometric sine function.

This algorithm produces two independent normally(0,1) distributed random numbers. To use one of them randomly, another Monte Carlo trial is performed. A U(0,1) random number is drawn, and depending on that outcome either N_1 or N_2 is chosen.

The norm(0,1) random numbers are converted to norm(mu,sigma) numbers as follows:

N(mu,sigma) = mu + N(0,1) * sigma

with
- mu .... desired mean of the norm. distr. r.n.
- sigma . standard deviation of the r.n.

e. Lognormal Random Number

For the lognormal distribution the following computation is made [Ref. 4] :

$$L(mu, sigma) = \exp[N(mu_n, sigma_n)]$$

with

- L[] .... desired lognormal random number
- N[] .... a normally distributed random number
- mu ..... desired mean of the lognormal r.n.
- mu_n .... mean of the normal random number
- sigma .. std.dev. of the lognormal r.n.
- $sigma_n$. the std.dev. of the normal r.n.

where

$$mu = \exp(mu_n + 1/2\ sigma_n^2)$$

$$sigma = SQRT[\exp(2\ mu + 2\ sigma_n^2) - \\ -\exp(2 * mu + sigma_n^2)]$$

$$mu_n = 2\log(mu) - 1/2\log(mu^2 + sigma^2)$$

$$sigma_n = SQRT[\log(sigma^2 + mu^2) - 2\log(mu)$$

2. <u>System Mean Time Between Failure ($MTBF_{System}$)</u>

Suppose a system is made up of n components. Let $Lambda_i$ be the failure rate for component i. Let T be the time to failure for the system. Let T_i be the time to failure of $component_i$.

Then, the probability of the lifetime of $component_i$ being greater than t is

$$P\{T_i > t\} = e^{-Lambda_i * t} \quad \text{for all i.}$$

For the system lifetime T the following holds:

$$P\{T > t\} = P\{\min(T_1, T_2, \ldots, T_k) > t\}$$

$$= P\{T_1 > t, T_2 > t, \ldots, T_k > t\}$$

$$= e^{-Lambda_1 * t}\ e^{-Lambda_2 * t} \ldots e^{-Lambda_k * t}$$

$$= e^{-(Lambda_1 + Lambda_2 + \ldots + Lambda_k) * t} .$$

Thus the system time to failure is exponential with parameter $Lambda_1 + Lambda_2 + \ldots + Lambda_n$. The mean time to failure for the system is

$$\frac{1}{Lambda_1 + \ldots + Lambda_n} = \frac{1}{\frac{1}{MTBF_1} + \ldots + \frac{1}{MTBF_n}}$$

Let us now determine the probability, that a system failure is due to component i. Let us consider the case in which the system has only two components. The probability, that the first system failure is due to component 1 is given by $P\{T_1 < T_2\}$.

Conditioning we get [Ref. 5] :

$$P\{T_1<T_2\} = \int_0^\infty P\{T1<T_2 \mid T_2=t\} * Lambda_2 * e^{-Lambda_2 * t}\, dt$$

$$= \int_0^\infty P\{T_1<t\} * Lambda_2 * e^{-Lambda_2 * t}\, dt$$

$$= \int_0^\infty (1-e^{-Lambda_1 * t}) * Lambda_2 * e^{-Lambda_2 * t}\, dt$$

$$= \int_0^\infty Lambda_2 * e^{-Lambda_2 * t}\, dt - $$

$$- Lambda_2 * \int_0^\infty e^{-(Lambda_1 + Lambda_2)t}\, dt$$

$$= 1 - \frac{Lambda_2}{Lambda_1 + Lambda_2}$$

$$= \frac{Lambda_1}{Lambda_1 + Lambda_2}$$

Because of the loss of memory property of the exponential the probability of any system failure being due to component 1 is the same,

$$= \frac{Lambda_1}{Lambda_1 + Lambda_2} .$$

This can be extended to the general case of n components in a straight forward manner. The probability that the failure is due to component i is :

$$P(X_i < X_k) = \frac{Lambda_i}{Sum(Lambda_k)}$$

$$= \frac{1/MTBF_i}{Sum(1/MTBF_k)}$$

for $i = 1,2,...,n$

$k = 1,2,...,i-1,i+1,...n.$

Example:

Suppose, a system consists of three parts X_1, X_2 and X_3.

The respective mean times to failure (MTTF) are:

$MTTF_1 = 1000$ hrs
$MTTF_2 = 500$ hrs
$MTTF_3 = 500$ hrs.

Then the expected MTTF for the system is

$$MTTF_{System} = \frac{1}{1/1000 + 1/500 + 1/500}$$

$$= \frac{1}{5/1000}$$

$$= 200 \text{ hrs.}$$

The probability the failure is caused by item 2 is

$$P\{X_2 < X_1, X_3\} = \frac{Lambda_2}{Lambda_1 + Lambda_2 + Lambda_3}$$

$$= \frac{1/500}{5/1000}$$

$$= 0.40$$

If a component j fails it is due to failure of one of its subcomponents. In the same way computed as above, the MTBF for the component depends on the MTBFs of its subcomponents.

$$MTBF_{comp.} = \frac{1}{Sum(1/MTBF_{subc.})}$$

and the conditional probability of a component failure being the result of subcomponent j is

$$P(comp._i \text{ fails} | \text{ subcomp.}_j \text{ fails}) = \frac{1/MTBF_j}{Sum(1/MTBF_i)}$$

This continues until the part level has been reached.

3. Total System Downtime

The time it takes to repair the system and get it back into service consists of several individual steps. First, the failure has to be reported to the next level maintenance facility. This facility will fill out a work order and inspect the system for damage. This time period will be called "administrative time delay" (TA).

Furthermore the system has to be shipped to the facility and back. The time for one-way transportation will be called "transportational time delay" (TT). Both delays are regarded as random variables. Then the system enters the repair queue and waits for an available server. This time will be called waiting time (WT). When a server is available the system enters repair. The repair time is RT.

The total system downtime will be:

$$DOWNTIME_{System} = TA_{Org\ Mn}+(2*TT_{Org\ Mn})+RT_{Comp.}+WT$$

Similarly, the downtime for a component will be:

$$DOWNTIME_{Comp.} = TA_{DS/GS}+(2*TT_{DS/GS})+RT_{Subcomp.}+WT$$

and the downtime for a subcomponent:

$$DOWNTIME_{Subc.} = TA+2*TT+RT+WT$$

(indices depend on the next level maintenance facility).

The following variables have to be known:

- TA
- TT
- RT

for each maintenance level.

For the idealized run, the WT's are assumed to be zero.

4. <u>The Queuing System</u>

In reality a system cannot expect to be served as soon as it fails and is inspected. A server at the maintenance facility has to be available and all needed spare parts must be available. Otherwise, the customer has to wait for the next available server or for the arrival of the necessary parts.

Service times are assumed to be independent of each other, and are distributed according to a user specified distribution. The service policy is FCFS[1]. The number of servers at a repair facility is k. The customer will always be served by the next available server.

If a spare part is not immediately available, the customer has to wait even to get into the server queue. These waiting times have to be accounted for. Therefore the total downtime of the system is:

$$DOWNTIME_{Sys}=WT_{parts}+WT_{labor}+TA_{OrgMn}+(2*TT_{OrgMn})+RT_{Sys}$$

with

WT_{parts} .... waiting time for spare parts

WT_{labor} .... waiting time for service.

5. Failure Generation

The system lifetime is computed from all the component lifetimes (MTBF). The lifetimes of the components are assumed to be exponentially distributed.

For purposes of saving memory and to be able to capture the feature that other components of a failed system do not fail while the system is shutdown, we generate system failures in accordance with the system

---

[1]FCFS = First Come First Served

failure distribution. Given the system failed, we then determine the component which failed by a uniform draw. In the same way, subcomponent failures and part failures are determined by the Monte Carlo method of drawing uniformly distributed random numbers and comparing these to a table of cumulative probabilities, which has previously been generated.

Example:

Let the system consist of three items with failure rates

$$\text{Lambda}_1 = 1/1000 \text{ hrs}$$

$$\text{Lambda}_2 = 1/500 \text{ hrs}$$

$$\text{Lambda}_3 = 1/500 \text{ hrs.}$$

Then the probability of a system failure due to item 1 is 0.2, due to item 2 is 0.4 and due to item 3 is 0.4. The cumulative table would look like this

0.00000 ... 0.20000	item 1
0.20001 ... 0.60000	item 2
0.60001 ... 1.00000	item 3.

Now a uniform(0,1) random number is drawn. Suppose this number was 0.6775. Thus the failing item would be item 3, because 0.6775 lies between 0.60001 and 1.0000, the range for item 3.

Using the same procedure would lead from the failing component to the failing subcomponent and finally the part, that caused the system to fail.

6. Restocking Policy

Since the program QBEHAVE analyses the behavior of the spare part stock levels,a restocking policy has to be defined.

We considered three common restocking policies:

- Restock on a one-for-one basis.

  Whenever an item is consumed, a substitute is immediately ordered and after a processing and shipping delay it fills the empty spot in the warehouse. Problems may be caused by shortcomings due to limited stocks and delays in the shipment.

- Restock after fixed time periods.

  After given time intervals a fixed number of items is shipped to replace the consumed items. Surpluses are possible with this method.

- Restock at the critical inventory level.

  This method is widely used. It combines the advantages of being able to satisfy most of the demand with the stock on hand with the smaller number of restocking transactions. Whenever the stock level reaches the refill level, the difference between this refill level and the stock maximum is ordered and shipped after delays. The problem is to keep the minimum level high enough to avoid stockout and low enough not to generate too large a holding cost.

In this simulation the third method, restocking at the critical lower levels, is used. This method is also known as the (s,S) - method, where s denotes the lower and S the upper limit.

S is taken to be the average demand during the four week planning period computed by program ESTIMATE. The value s is computed so as to satisfy a specified stockout risk.

The stockout risk factor r is the probability of running out of stock while the replacement is ordered and shipped to the consuming facility. This delay for generating the order and for receiving it is called leadtime.

With exponential times to failure the distribution of the number of failures during a leadtime L is POISSON with parameter (Lambda * L). We therefore compute the stock level s by finding the smallest s such that

$$P[X_i > s] <= r,$$

where X_i is POISSON($Lambda_i$*L).

To find s from this, compute

$$P[X_i <= s] >= 1 - r$$

For s=0: probability $p(0) = \exp(-Lambda_i * L)$

s=1: $p(1) = \frac{(Lambda_i * L)}{1} * p(0)$

s=2: $p(2) = \frac{(Lambda_i * L)}{2} * p(1)$ etc.

Compute p(k) recursively until the sum of the cumulative probabilities

$$p(0)+p(1)+p(2)+...+p(k) >= 1-r.$$

The value for s is then k.

E.g. $Lambda_i = 1/30$, $L = 5$, $r = 0.10$

k = 0: $p(0) = \exp(-Lambda_i * L) = \exp(-5/30) = 0.8465$

k = 1: $p(1) = \frac{(Lambda_i * L)}{1} * p(0)$

$= \frac{0.1667}{1} * 0.8465 = 0.1411$

$[p(0) + p(1) = 0.9875] > [1 - r = 1 - 0.1 = 0.9]$

That implies s = 1.

III. THE REALIZATION

A. DETERMINATION OF THE OUTPUT DATA

Before the simulation could be realized, the decision variables for the desired output data had to be determined. It seemed to be reasonable that the needed data would consist at least of information about:

- the number of maintenance personnel on each level of maintenance
- the number of spare parts to be provided for each supply level
- the use of the provided capabilities in terms of manpower and spare parts
- the total expected cost for the system.

These data should be based on a planning period. This period should not be too short, because that would unneccessarily extend the duration of program runs, and not too long, because the the resulting data would be unreal. A period of four weeks seemed to be adequate for planning purposes.

B. STANDARDIZATION OF THE INPUT DATA

Due to the large volume, the input data are stored in files. To improve access times, the data are written into random access files with linked lists used to navigate through the files.

This program is written for military use; knowledge about the military organization of major units is assumed. Appendix C displays all used tactical signs used in this thesis.

1. Major Unit Structural Data

To be able to extract the distribution of the system in question out of the desired major unit (brigade, division and/or corps), the structure of these major units has to be known and standardized. The program TACMIX gives the user the interface to enter the required data. Once the program is loaded, the user is prompted to chose among four options[2]:

```
Create New Organization          (1)
View Existing Organization       (2)
Modify Existing Organization     (3)
Terminate Program                (9)
```

If the user chooses option (1) he will see a screen with tactical signs of the following major unit types:

Corps
Armored Division
Mechanized Infantry Division
Infantry Division
Airborne Division
Air Assault Division
Mountaineer Division
Armored Brigade
Mechanized Infantry Brigade

---

[2]see Appendix A.1.

Infantry Brigade
Airborne Brigade
Mountaineer Brigade
Combat Aviation Brigade
Signal Brigade
Air Cavalry Combat Brigade.

After the decision for one major unit type, all unit types, regiments,battalions,companies, which may be possible under this major unit will be displayed. The user chooses the units, their number and creates the structure of the major unit. The newly created file will overwrite any existing file for this major unit.

Selection of option (2) will display the contents of the chosen unit type file in graphical form: the tactical signs for the units in the major unit and the number of each of these units.

Selection of option (3) will also display the contents of the major unit file. In addition, it allows the user to implement modifications into the file. The new file overwrites the previous one.

For each of the above major units there exists a file in the program package in accordance with reference 1.

2. <u>Distribution of Systems</u>

Next, the user calls the program INMILDAT for entry of the number of the systems into the military structure. The program displays all types of units from regiment to company and prompts for the number of systems in the respective unit. Another prompt will have the user check the correctness of the entry. After all units have been displayed the program will ask for reserve in either percentage or absolute numbers. Program ESTIMATE will use this number to find the total number of systems within a corps. ESTIMAT will give the amount of systems in the major units below the corps level and also the reserve if one exists.

At the end of the program INMILDAT it will prompt the user to enter a data file name for the data just created and write the data sequentially into this file. If no name is entered the default file name "DISTRIBN" will be used.

3. System Structural Data

Before the user can enter the system data, they have to be available in the format necessary for these inputs.

The following information will be entered[3]:

Item Type - to be able to answer this prompt, the system must be broken down such that the structure is similar to the one shown in Figure 1. A part, a non repairable item, will always require a 9, even if it is directly under the system level 1,a component will always be a 2, and the subcomponents have to be classified from 3 to 8 depending on their level within the system.

National Stock Number - this is the NATO standardized stock number in the format:

AAAA-BB-CCC-DDDD.

---

[3]see Appendix A.3.

All the groups of letters stand for numbers. In this program it will be assumed that the country code (BB) is the same throughout the system. If the country code is omitted, the stock number can be written as:

AAAADDDD.CCC

and can be used as a unique item file name.

Item Description - this is the name of the item

How many of these Items - this is not the overall total but rather the number within the next level item. If subcomponent A appears x times in the system but y times in component B, then enter y here.

Unit Price in US Dollars - self-explanatory

MTBF[4] (assumed) in Days - this information should be provided by the manufacturer of the item. If there are no data available, an estimate of this number is necessary. Fractions of hours can be entered.

Replacement Level - this is the maintenance level where the failed item is replaced.

Repair Level - this is the maintenance level where the replaced item will be repaired.

MTTR[5] in hours - enter here the repair time provided by the manufacturer. If no data are available, enter an estimate based on similar items.

After entering all data, a summary of these data will be displayed. The user has to acknowledge the correctness of the input. If they are correct the user will be asked if there are more items to be entered.

When the last item has been entered, they will be written onto the disk into a random access file with its transformed stock number name in the case of the

---

[4]MTBF = Mean Time Between Failures

[5]MTTR = Mean Time To Repair

system. In all other cases the name will be the transformed stock number of the next higher level item.

C. PERFORM IDEALIZED EVALUATION RUN

1. Macro - Flowchart Program ESTIMATE

The basic structure and sequence of computations of the program ESTIMATE starts with an initialization part including reading default data and string variables from the.memory and prompting the user for additional information necessary for the simulation (see Figure 2.5). It computes the system lifetime from the component data and requires the choice of major unit, for which the simulation will be run. Finally, a failure time is drawn for each of the systems in the major unit using the system lifetime. These failure times are sorted.

The sequence of steps depicted in Figure 2.6 is repeated for as many times as failures have to be generated. For the next failing system the failing component, subcomponent(s) and part are determined by the randomization procedure described in Chapter II. A repair time is drawn from the selected distribution and the total downtime for the system (ignoring any queuing waiting times) is computed. For this system the time to the next failure T is then determined and merged with the failure times of the other systems.

Once all the failures have been simulated and stored in the data files, the output is generated for each maintenance level. Some additional statistical data about the simulation are provided on the last output page (see Figure 2.7).

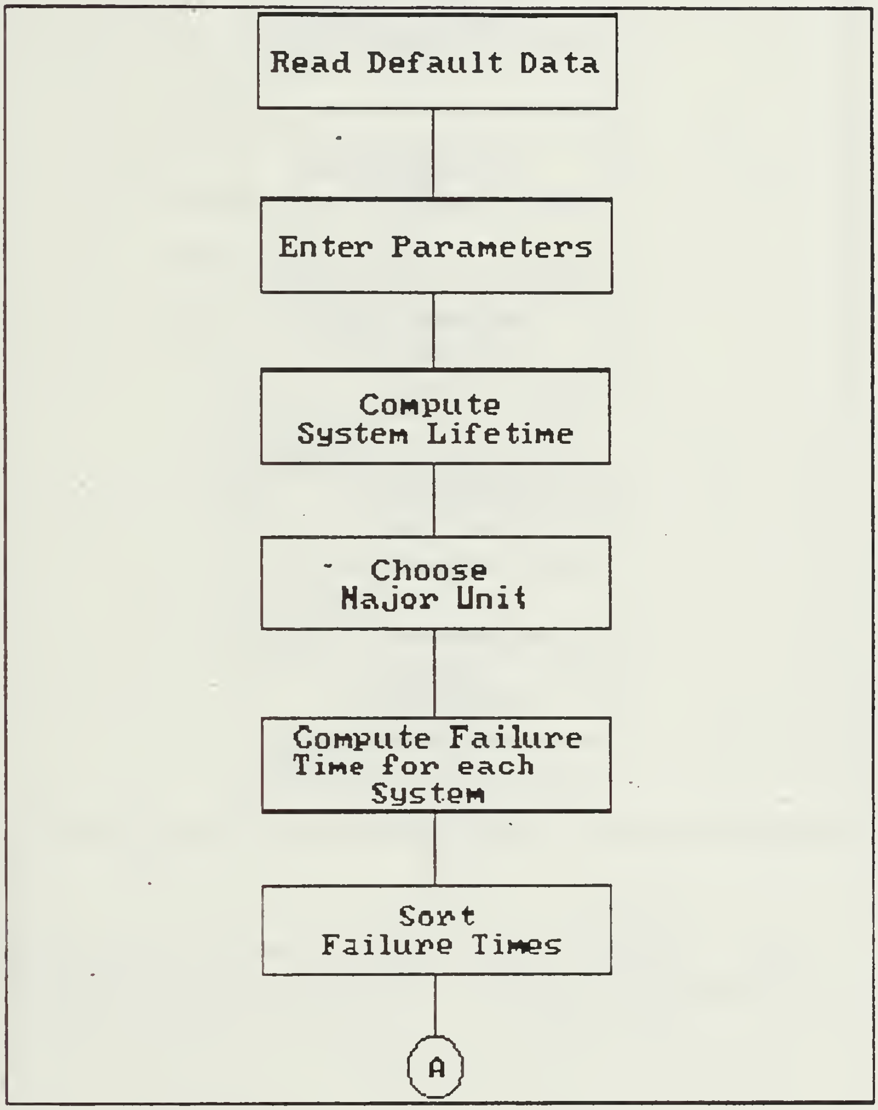


Figure 2.5 Initialization Phase of Program ESTIMATE

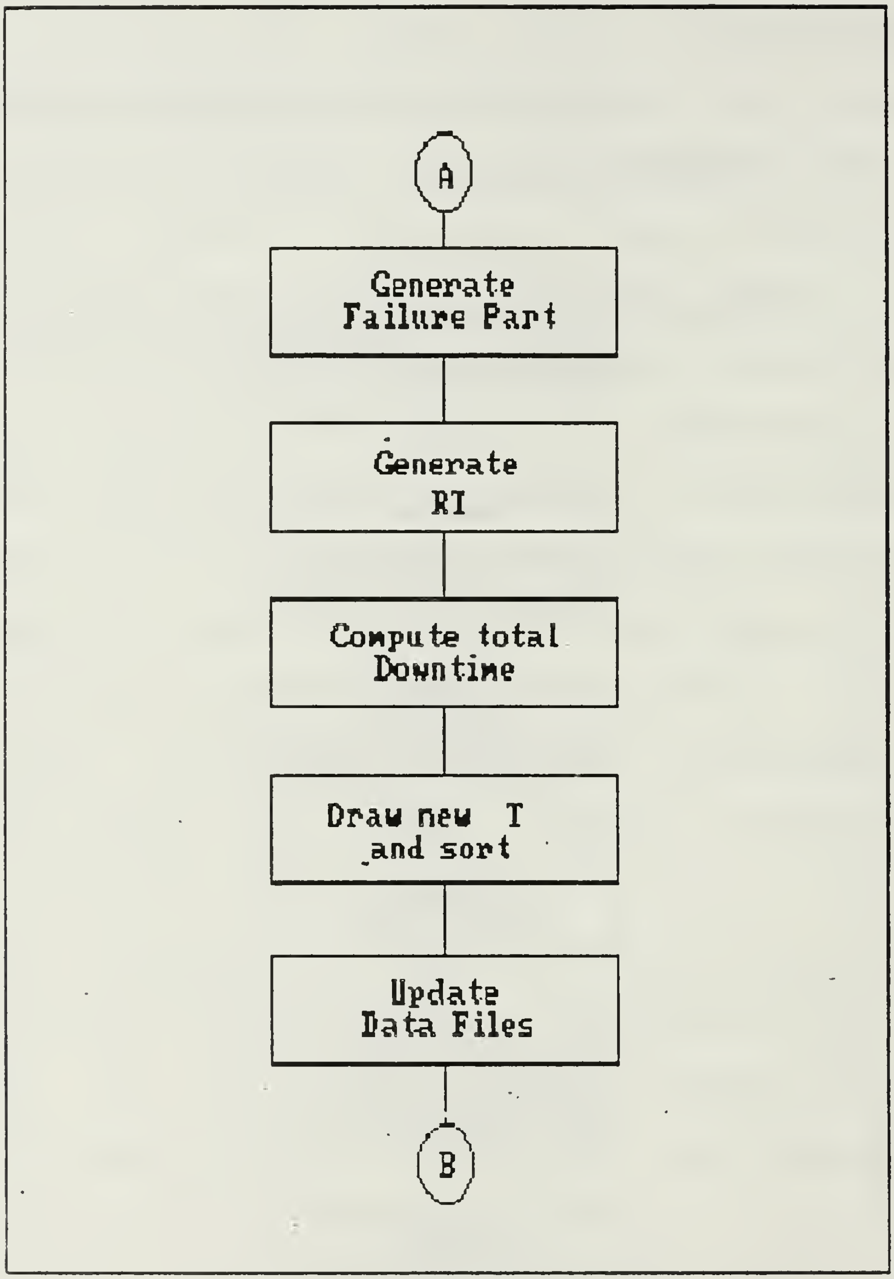


Figure 2.6 Simulation Core of Program ESTIMATE

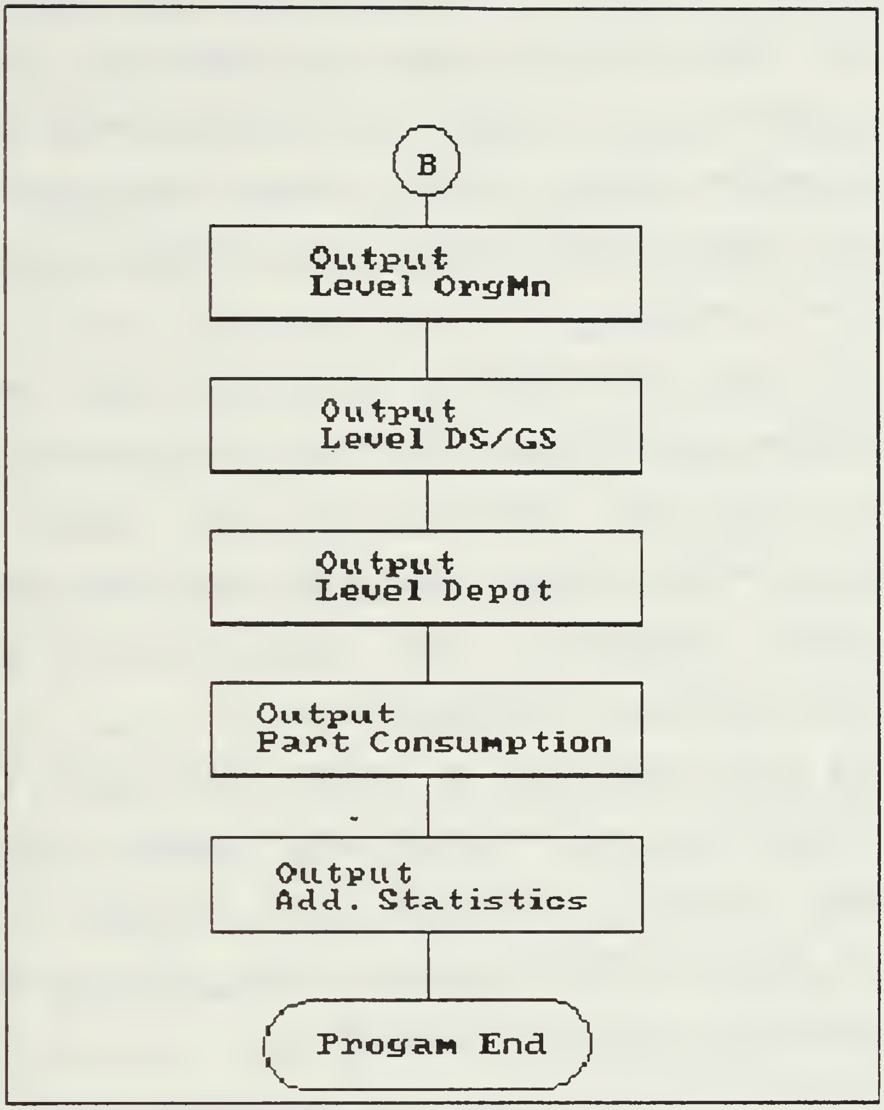


Figure 2.7 Output of Program ESTIMATE

2. Detailed Look at Program ESTIMATE

ESTIMATE takes the data in the item files and the military structure files and computes planning figures for the desired output information. Since the amount of data is so huge, the data are kept in auxiliary random access files during the program run. This allows rapid access to the data without cluttering the computer's memory.

In the introductory phase of the program, default values and strings are read into variables. Old auxiliary files are deleted and the random number generator is seeded with the DOS day/time group to achieve real randomness. The user then enters the number of failures to be generated.

The user also has to enter the distributions for administrative and transportation delay times. The first input screen[6] requires the entry of the maintenance level after which the type of distribution and its parameters must be entered.

Choices for the distributions and parameters are:

Uniform	Parameters: lower, upper limit
Normal	Parameters: mean, standard deviation
Exponential	Parameter : mean
Gamma	Parameters: mean, degrees of freedom
Lognormal	Parameters: mean, standard deviation
Fixed	Parameter : mean.

[6]see Appendix C.4.

If the user is uncertain about the parameter choice, default values are provided for each distribution. Both, the distributions and the parameters for each maintenance level are printed[7].

After input of the system stock number, the program computes the overall system failure rate $lambda_{System}$ from all the individual failure rates of the components, subcomponents and parts.

The next step is to find the distribution of the systems in the corps. The program takes the entries in the system distribution file created with the program INMILDAT and compares them with the military structure data created by the program TACMIX. The output will be a list of the major units in the corps with the respective number of systems[8]. This list will be displayed on the screen and printed together with the user's choice of major unit within the corps.

The program will compute one individual failure time for each of the systems in the major unit. The program will generate an exponentially distributed random number with parameter $lambda_{System}$ for each system and then order these failure times.

After having obtained all failure times the program will determine which component caused each

---

[7]see Appendix B.1.a.

[8]see Appendix B.1.b.

failure. It does this as described in chapter II. The same procedure is used to determine the failed subcomponent and part. As soon as the failed part is determined, all information (times, costs, etc.) about the chain of failed items within the system up to this part have been stored in auxiliary files.

Starting from the time of failure (FTIME), the program computes an idealized repair cycle for this system. It assumes that there are unlimited resources in spare parts and manpower available.

The time at which the system is operational again and back at its location, is computed as follows:

UPTIME = FTIME+2*TT+TA+MTTR

```
with FTIME ... time system fails
     TT ...... transportation delay time
     TA ...... administrative delay time
     MTTR .... mean time to repair.
```

The program will always serve the next failing system until the desired number of failures have occurred.

3. Output Program ESTIMATE

The output of the idealized run performed in the program ESTIMATE consists of information about the manpower requirements for each maintenance level and the spare part stocking levels[9].

The output consists of at least six pages. The first two pages provide data about manpower requirements and the consumed spare parts. The last page gives additional information about the simulation itself and the costs involved for the systems.

Page 1 of the output gives the number of failures, a listing of the distributions and parameters for the transportation and administrative delay times and the repair time distribution. These data were entered by the user in the initialization phase of ESTIMATE.

Page 2 of the output gives the distribution of the systems within the major units of the corps. It also provides information about the number of systems on reserve and the total number of systems in the corps.

Pages 3 to 5 give the summaries for manpower and spare part consumption for the three maintenance levels. The header shows the number of simulation replications. Next is a list of consumed items on each respective level. Provided are the national stock numbers, the item descriptions and the total number of units consumed.

---

[9] see Appendix A.4

The manpower data show the total number of manhours spent on the given number of simulations. This total time is converted to an average time within a four week planning period. Based on this number the average number of repairmen for this level is shown.

Page 6 gives additional information about the simulation. It lists the total time for the simulation in weeks and the number of four week periods.

It also provides system data. The average system downtime and the average system availability during the simulation are provided.

For the costs of the new systems the output shows the average maintenance cost per system (only parts are consumed), the initial acquisition costs for introducing the new system plus one four week spare part package into the given major unit and finally the cost for an additional average four week supply of spare parts.

D. RERUN THE SIMULATION WITH QUEUING

1. Macro - Flowchart Program QBEHAVE

The data processing in the program QBEHAVE starts with a reading of parameters generated in program ESTIMATE. Then, for each system, the first failure time is read. Taking transportation and administrative delays into account, the times of the actual arrival at the maintenance facility are computed and sorted. Then the program simulates the repair at the Org Mn level by checking the availability of servers.

To keep track of the times the servers are busy, the program uses two pointers for each server. One for the earliest time the server starts service (intime) and the second for the earliest time at which the server will be available (outtime). As soon as a customer (weapon system) enters the queuing system (entertime), the next available outtime is found. If the outtime is less or equal the entertime, the customer is immediately served. Otherwise the difference between the minimum of the outtimes and the entertime is the waiting time for this customer. The outtime pointer for the selected server is set to the value {intime + repair time}. Thus an entering customer always gets to see the first available server and the time at which service will be complete is determined when each customer arrives. Levels DS/GS and Depot are handled in the same way. The stock levels for each spare part at all levels are computed and printed.

This simulation is run with the numbers of servers equal to the numbers suggested by the program ESTIMATE or for each of these numbers within +/- 10 %

of that figure (unless not feasible, i.e. number <= 0 or percentage busy < 10 %).

Figure 2.8 shows the initialization and the first repair level (Org Mn) within the sensitivity analysis loop.

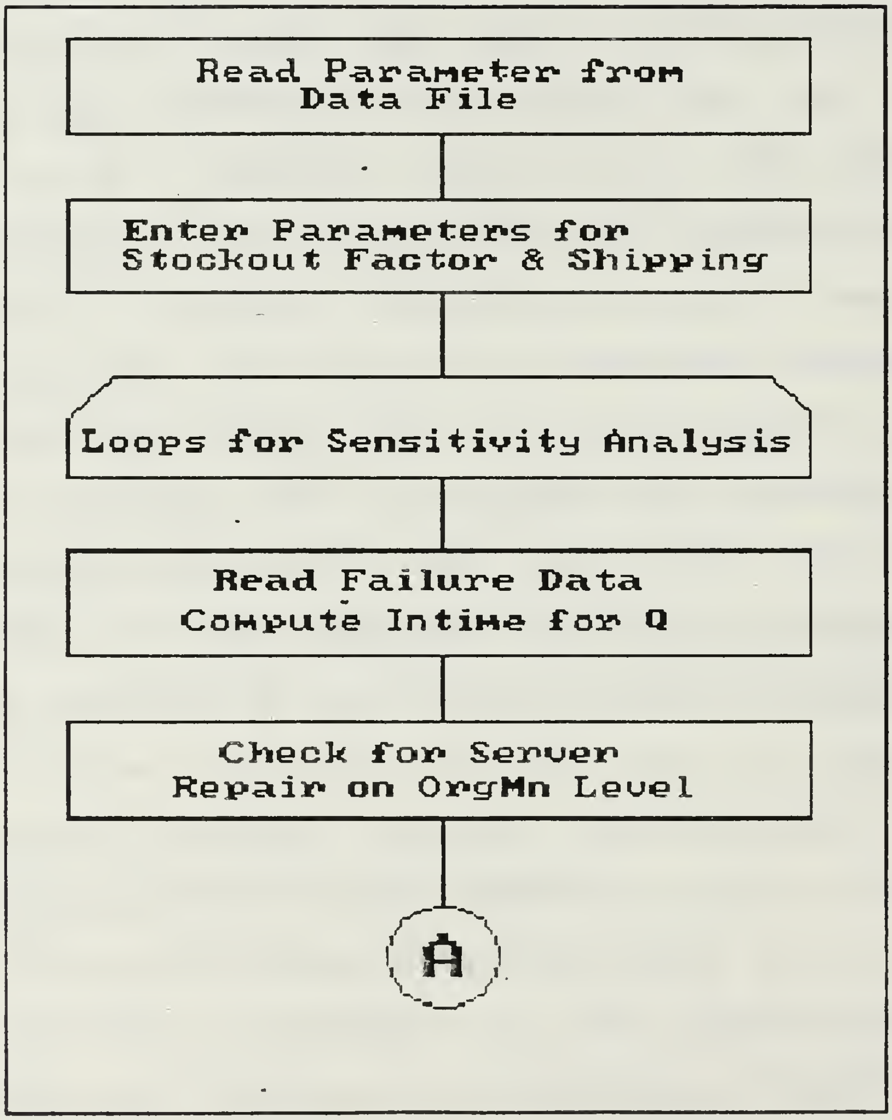


Figure 2.8 Repair Level OrgMn in Program QBEHAVE

Figure 2.9 shows the other two repair levels (DS/GS and Depot) within the sensitivity analysis loop.

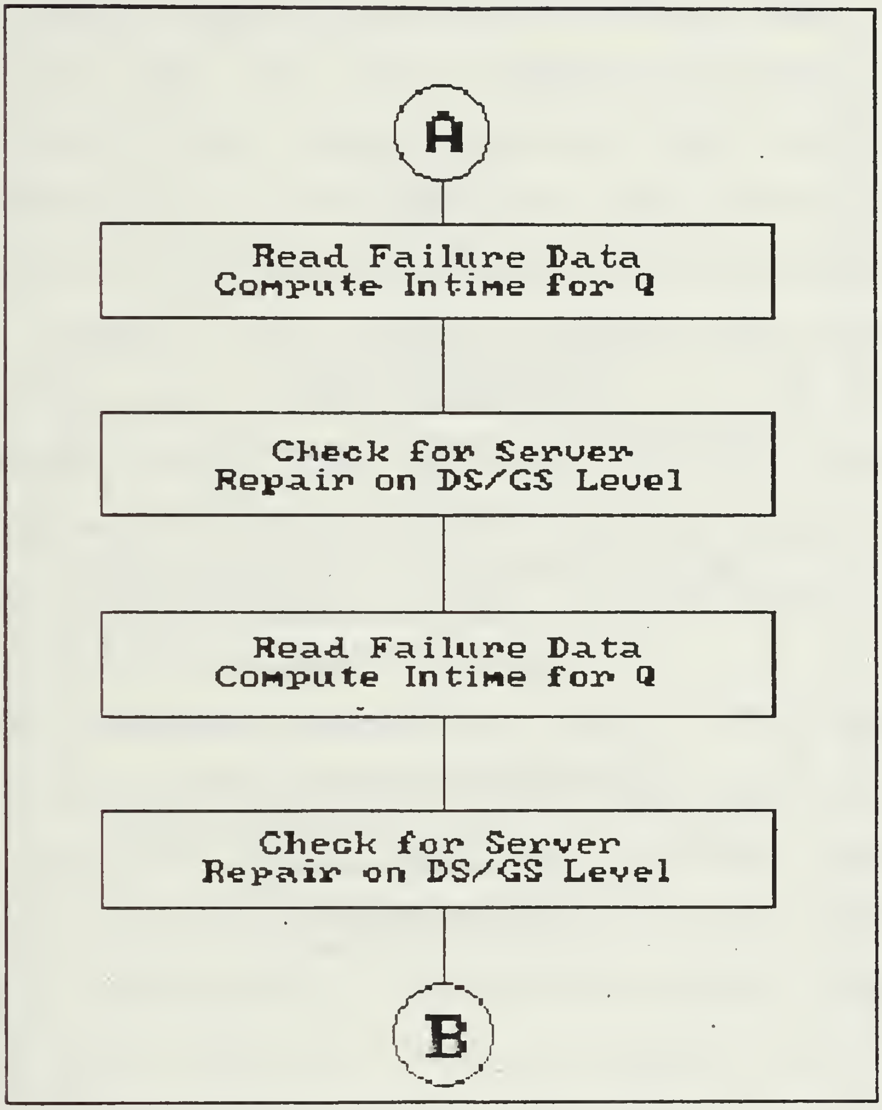


Figure 2.9 Repair Levels DS/GS and Depot in QBEHAVE

Figure 2.10 shows the computation of spare part consumption within the sensitivity analysis loop.

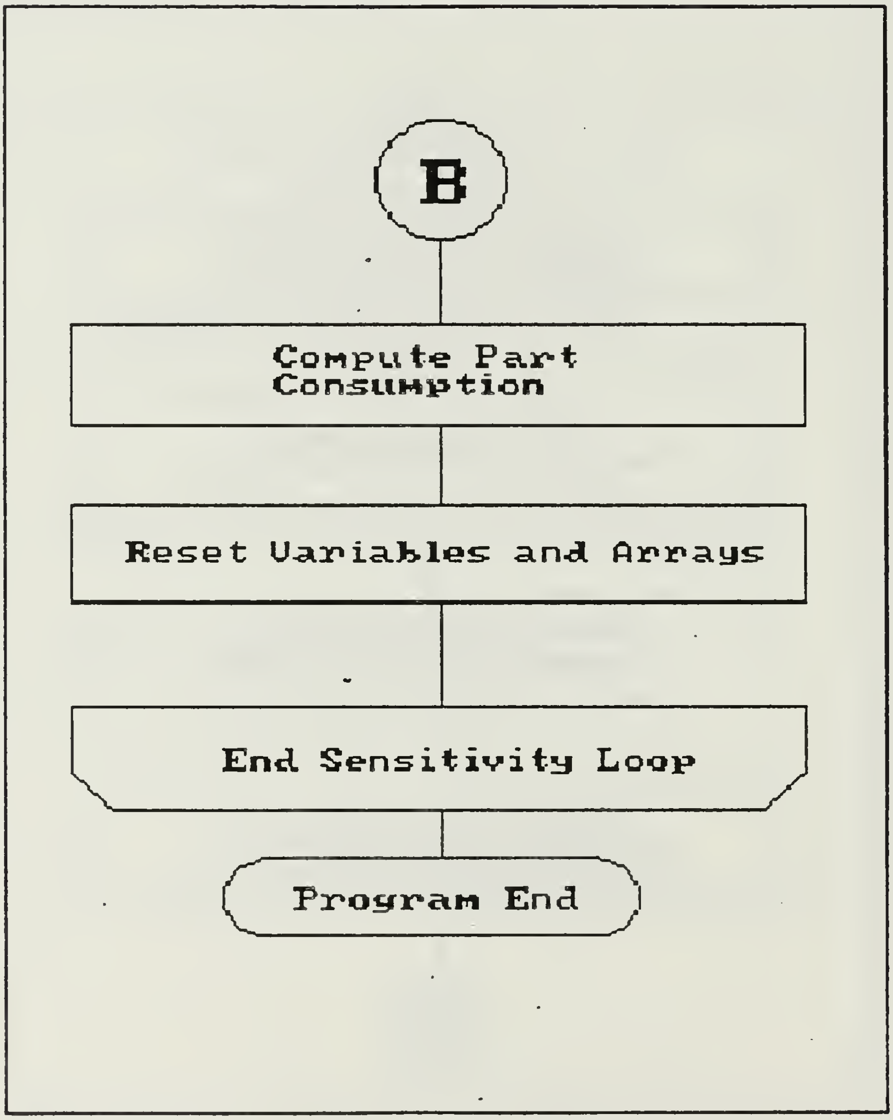


Figure 2.10 Spare Part Consumption in QBEHAVE

2. Detailed Look at Program QBEHAVE

QBEHAVE gives a more realistic view of the behavior of the systems in terms of downtimes than does the program ESTIMATE, which provides the data for QBEHAVE. QBEHAVE takes the ordered failure times of the systems and adds to these numbers one half of the administrative delay time, assuming that this delay occurs equally at the beginning and at the end of processing a workorder. One transportational delay time is also added. The ordered numbers are used to determine the entry time for the system into service. The customer with the earliest failure does not necessarily have to be the first one to be served. It might take him longer to get to the server or the administrative delay might be longer.

The service policy is FCFS. The algorithm checks for the next available server and if a server is free, the customer entry time plus the MTTR result in the customer departure time. To this time the remaining half of the administrative delay and another transportational delay time is added to obtain the time the customer is operational again. The time the customer leaves the server is also used as the failure time of the customer for the next level of maintenance, where the replaced item itself will now be repaired.

These computations continue until the part level is reached and no further repairs will be performed.

The calculations on the next level can only be performed after the previous level is completely computed, since the times the customer leaves the previous level have to be converted to arrival times at the next level. They then have to be sorted to obtain the proper sequence of service.

3. Output Program QBEHAVE

The output of QBEHAVE offers information about the manpower requirements for each of the maintenance levels and the spare part consumption[10]. Because this phase of the simulation is not an idealized run such as was the case with the program ESTIMATE, queuing waiting time data are also available.

The number of output pages is always a multiple of two. A set of output for each combination of servers consists of two consecutive pages - one for the manpower and the other for the spare part data. The total number of output pages depends on the program mode. If the sensitivity analysis is to be performed, then the number of pages depends on the number of different cases considered in the sensitivity analysis. For this mode the first set shows the data for the

---

[10]see Appendix A.5

combination of smallest server numbers on all repair levels. The next set usually shows the results for an increased number of servers on the Org Mn level unless the upper limit is reached or the percentage of time server 1 is busy is less than 10 per cent.

When all possible numbers of servers on the Org Mn have been tried, the number of servers on the DS/GS level is increased, the number of servers on the Org Mn level is reset to its lowest value and the computations start over again.

After all possible combinations of servers on Org Mn level and DS/GS level have been accounted for, then the number of servers on the Depot level is increased. The program terminates when all combinations of server numbers on all three levels have been tried. For each combination the spare part consumption is collected through all three levels and printed. The results are shown on the second page of the output set. Each combination of server numbers produces one set of output.

The first page of each output set shows the statistics for the maintenance queues on the various levels. On the Org Mn level the number of customers (systems in the major unit) and the number of simulations are provided. For the other levels these numbers do not change.

The output then shows the number of servers available for each maintenance level. The total number of delays, the total waiting time and the maximum number of customers waiting are provided. Also provided are the average waiting times per customer, (given the customer waited), and the expected average waiting for any customer entering the queuing system. Also shown is the percentage of time each server was busy. Finally, statistics about the simulation, like total elapsed time and the number of four week periods are given.

The second page offers a summary of stock behavior for all consumed items, repairable or not. Initial and final stock levels within the simulation period are shown. The last column shows the minimum stock number, where negative values represent stockouts.

IV. LIMITATIONS

To keep the size of the different programs and the amount of generated and accessed data manageable, a few limitations were built into the programs.

The number of iterations (trials) is restricted to 1000. Numbers close to 1000 cause excessive execution time for the program. The number of systems handled is resticted to 500. The total number of different components, subcomponents and parts is limited to 1000. The number of each individual part is not restricted. If a larger number of items is required, then the system has to be divided, and the simulation has to be run for for the different parts of the system. These partial results can then be added together to obtain the numbers for the system as a whole. The number of items under any other higher level item is limited to 99. The item price must not be greater than $ 99999.99 (of course, one can express costs in tens, hundreds and thousands of dollars to avoid this problem.

The maximum number of repairmen on each of the three maintenance levels is restricted to 50. It would be unreasonable to assign more people to one system.

The systems are strictly structured as serial systems.

V. SUMMARY AND CONCLUSIONS

The output of the program ESTIMATE yields the average numbers of repairmen per maintenance level for a four week period and the total number of spare parts for the whole simulation. Although these numbers are computed assuming no queuing they are useful for planning figures. The values are input to QBEHAVE which simulates more realistic behavior.

For a quick overview of the expected requirements and to obtain ballpark figures a run of ESTIMATE will likely yield reasonable numbers. For a more detailed view the user has to run QBEHAVE and select a suitable solution out of the array offered. When interpreting the output of QBEHAVE, the user should not only take the waiting times for the customers into consideration, but also observe the proportion of time the various servers are busy. Changes on a lower level directly influence the behavior on the higher levels. These must be taken into account when raising the number of repairmen.

CAESAR is capable of generating valuable predictions for both the maintenance and the supply organization by bringing a multitude of aspects into the consideration of the responsible manager.

CAESAR presents a broad spectrum of options for the deployment of manpower and supplies within the hierarchical logistics structure. CAESAR will not make the decisions for the manager. But CAESAR will make the decision process easier and the alternatives more transparent.

APPENDIX A

SAMPLE RUN

In all sample runs of CAESAR's programs the following conventions are used:

- a message from the computer will always be denoted in lower and upper case letters with an (M) at the beginning of the message,
- an input for the computer will always be denoted in upper case letters with an (I) at the beginning of the input,
- <enter> means to press the ENTER key after typing the input
- any comments will be denoted (C).

1. Program TACMIX

TACMIX is the graphic interface for CAESAR to provide the input of major unit structures. The program will present all units and major units with their respective tactical signs. If the number of units of any type is greater than 1, then an index outside the lower right corner of the symbol will indicate this number.

If you intend to create or modify a major unit, and you want to keep the old data, make a copy of the data file for this major unit. To do this you enter from DOS

C>COPY OLDNAME NEWNAME <enter>

using the old and the new name for the data file.

E.g. to copy BRIG16 into a back up file called BR16 you enter

C> COPY BRIG16 BR16 <enter>.

There will be a new file named BR16 containing the same data as BRIG16. Now you can modify BRIG16 in any way you like.

To retrieve the data after you are done, simply copy the contents of BR16 back into BRIG16, typing

C>COPY BR16 BRIG16 <enter>.

To start TACMIX, type at the DOS prompt:

C>TACMIX <enter>

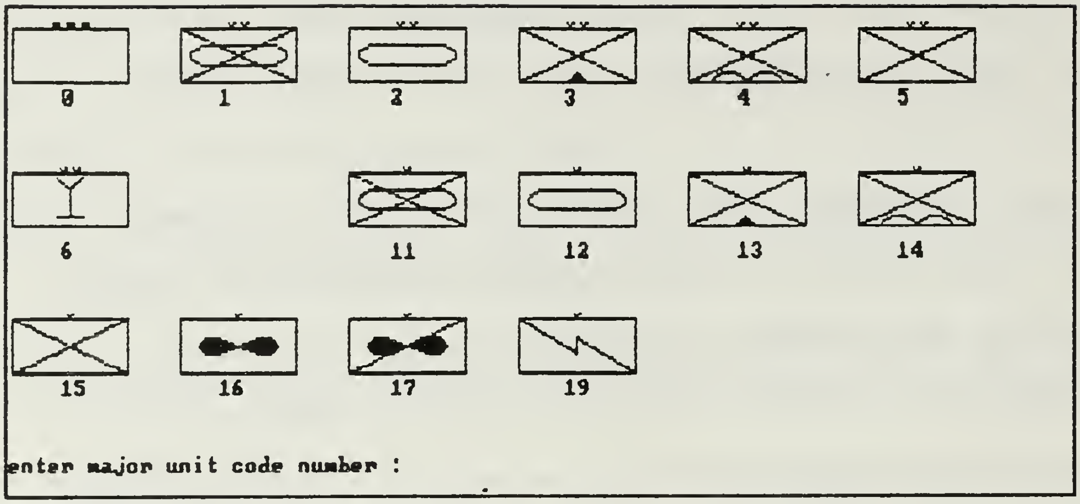
(M)

```
            ****************************
            ***     Program TACMIX   ***
            ****       Options:      ****
            ****************************

   - Create New Organization .......(1)

   - View Existing Organization ....(2)

   - Modify Existing Organization ..(3)


   - Terminate Program .............(9)


             Enter Number of Choice :
```

(C) Let us create a new organization for major unit 11.

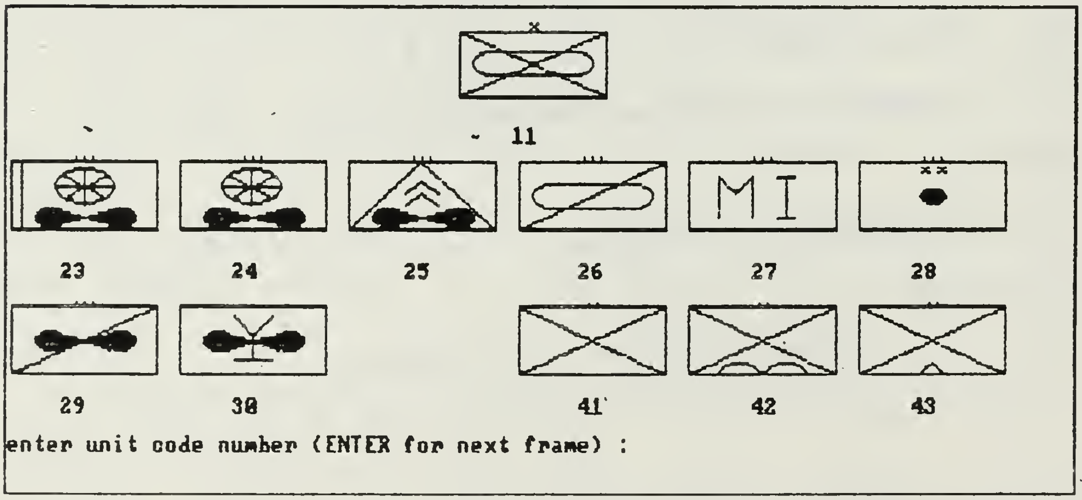
(I) 1 <enter>

(M)



(I) 11 <enter>

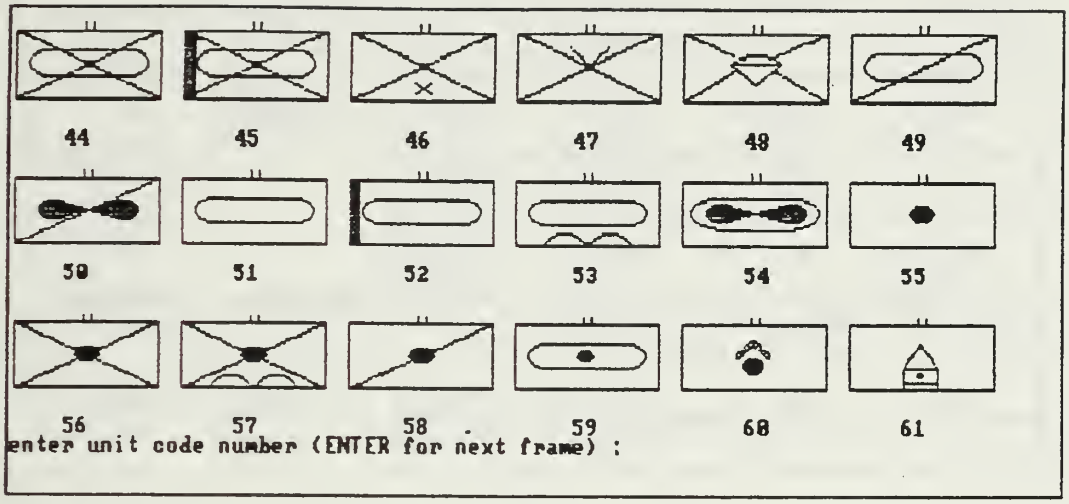
(M)



(C) We will only browse through the tactical signs.

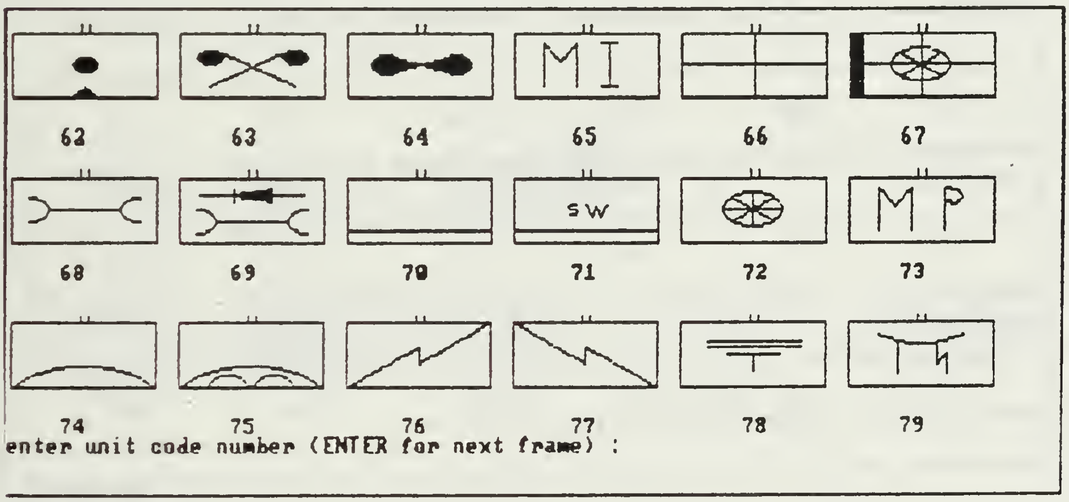
(I) <enter>

(M)



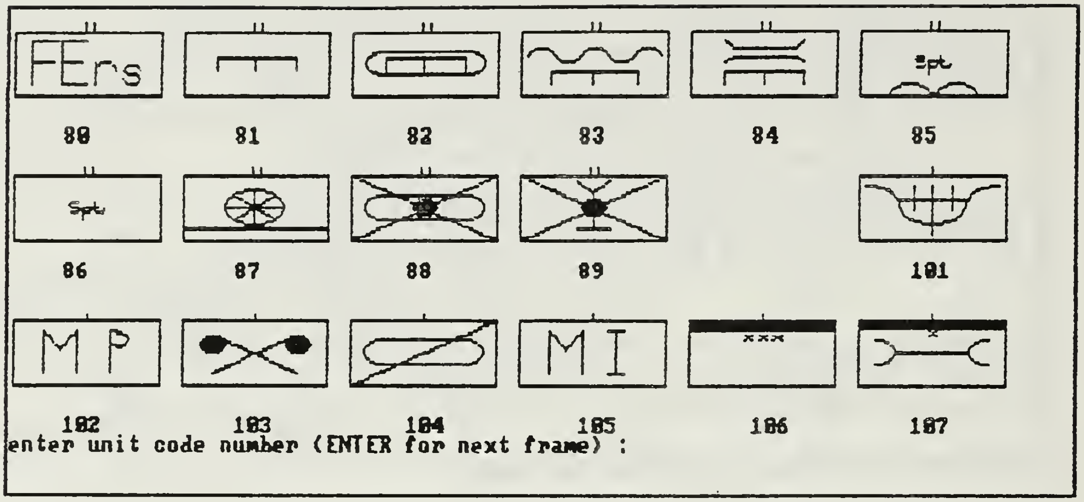
(I) <enter>

(M)



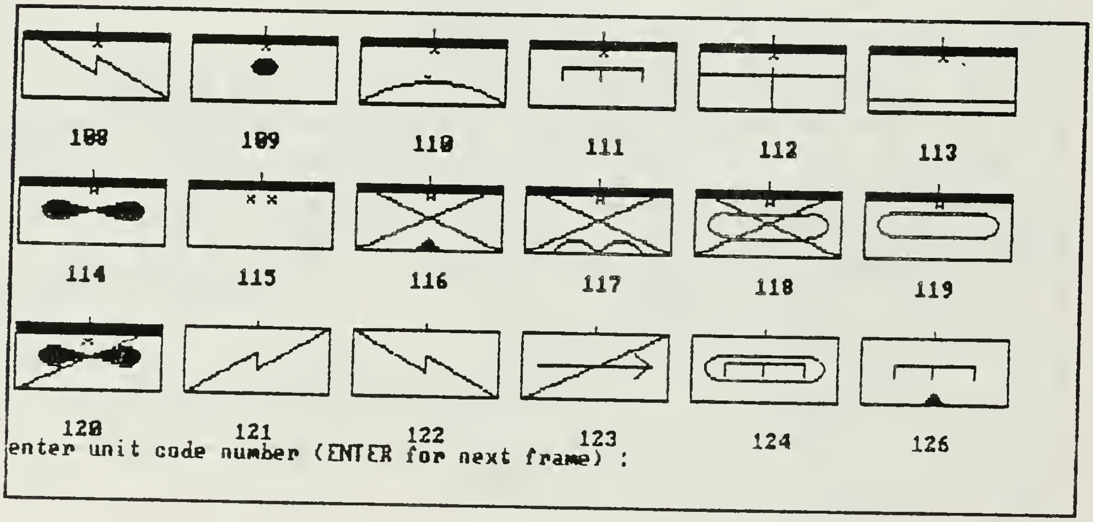
(I) <enter>

(M)



(I) <enter>

(M)

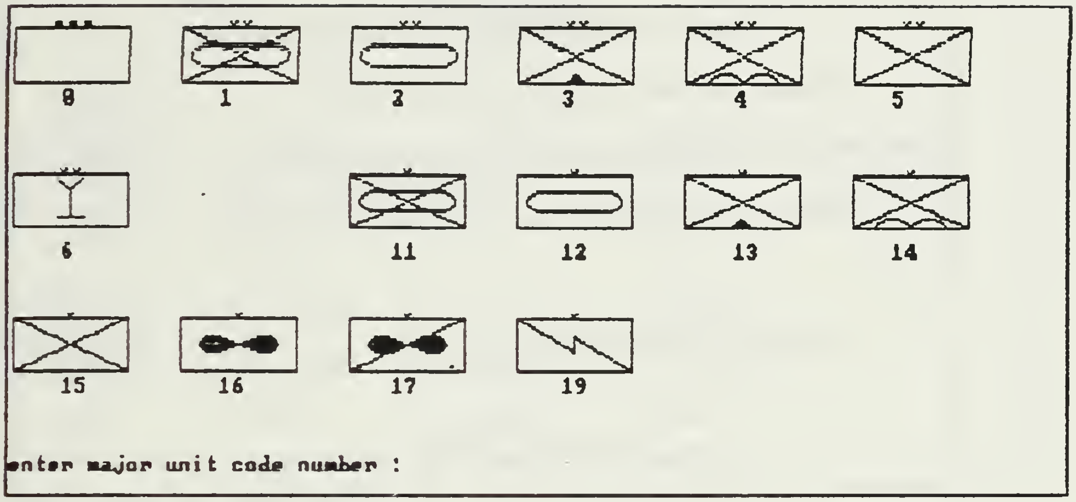


(I) <enter>

(C) Let us use option 2 (view existing organization) to take a close look at major unit 14.

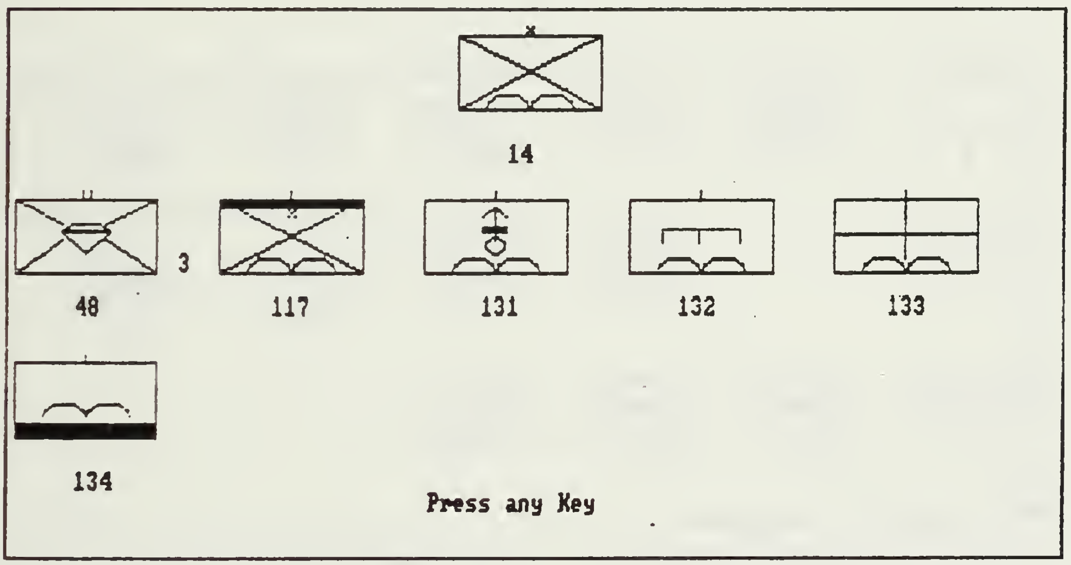
(I) 2 <enter>

(M)



(I) 14 <enter>

(M)



(I) <enter>

(C) This brings you back to the menu screen.

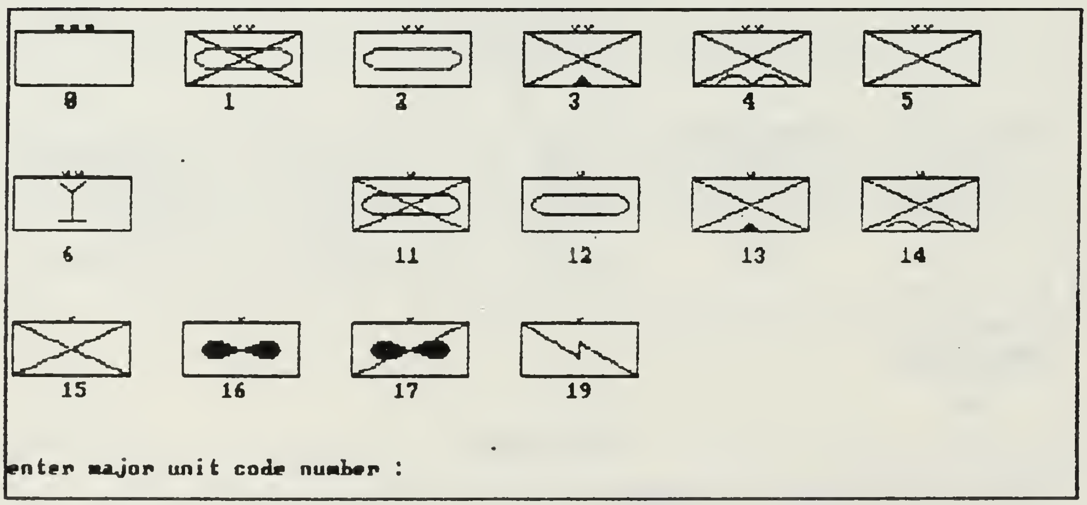
(M)

```
******************************
***     Program TACMIX     ***
***       Options:         ***
******************************

 - Create New Organization .......(1)

 - View Existing Organization ....(2)

 - Modify Existing Organization ..(3)


 - Terminate Program .............(9)


        Enter Number of Choice :
```

(C) Let us use option 3 (modify existing organization) and modify major unit 12.

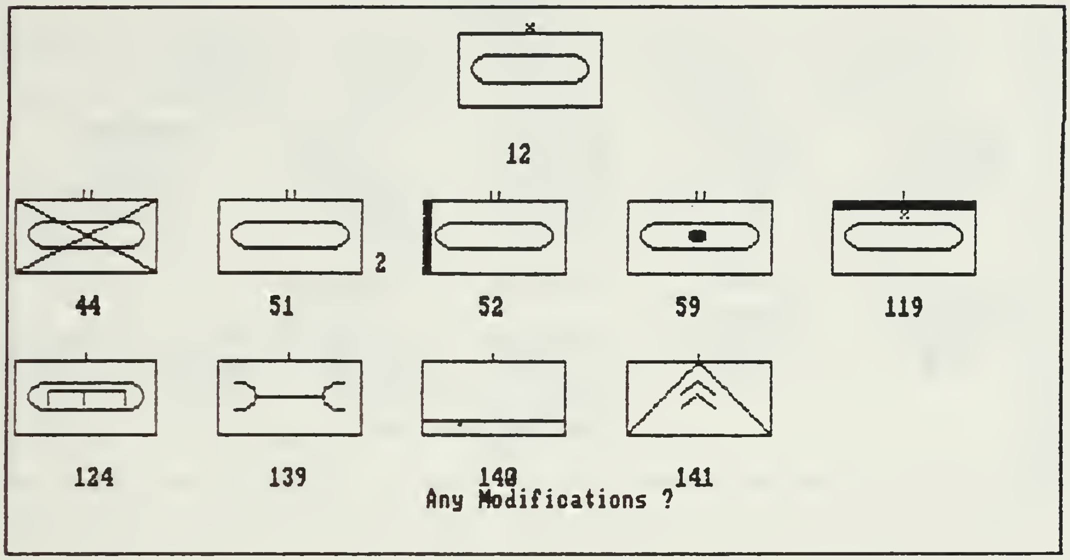
(I) 3 <enter>

(M)



(I) 12 <enter>

(M)



(I) y

(C) change the number of units type 51 from 2 to 3

(M) Which Modification ?(0=erase, 1=change number of units, 2=add a unit)

(I) 1

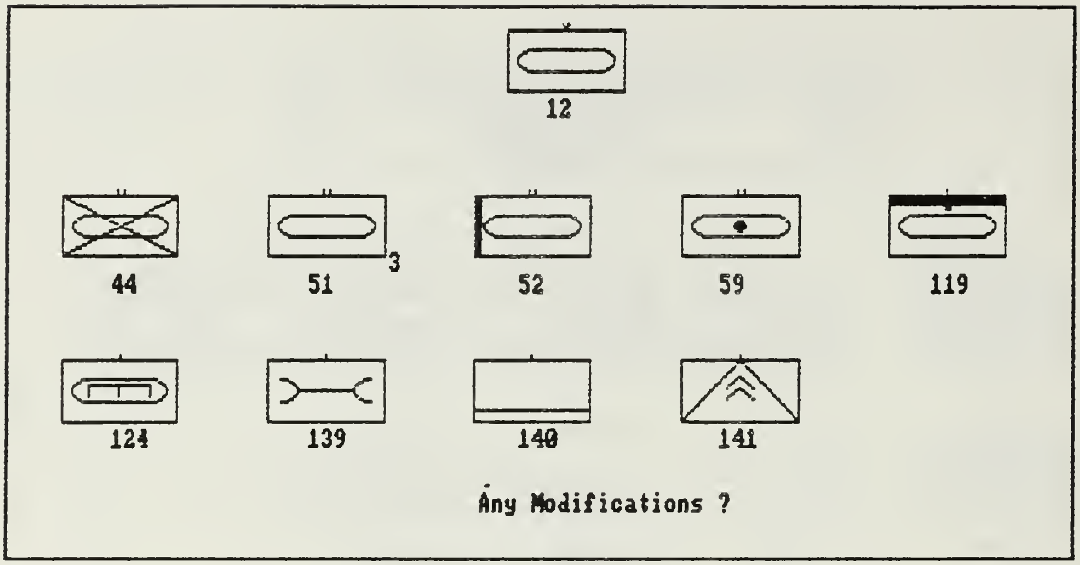
(M) Which Unit Number ?

(I) 51 <enter>

(M) Enter new number of units

(I) 3 <enter>

(M)



(I) n

(C) Now we are back in the main menu

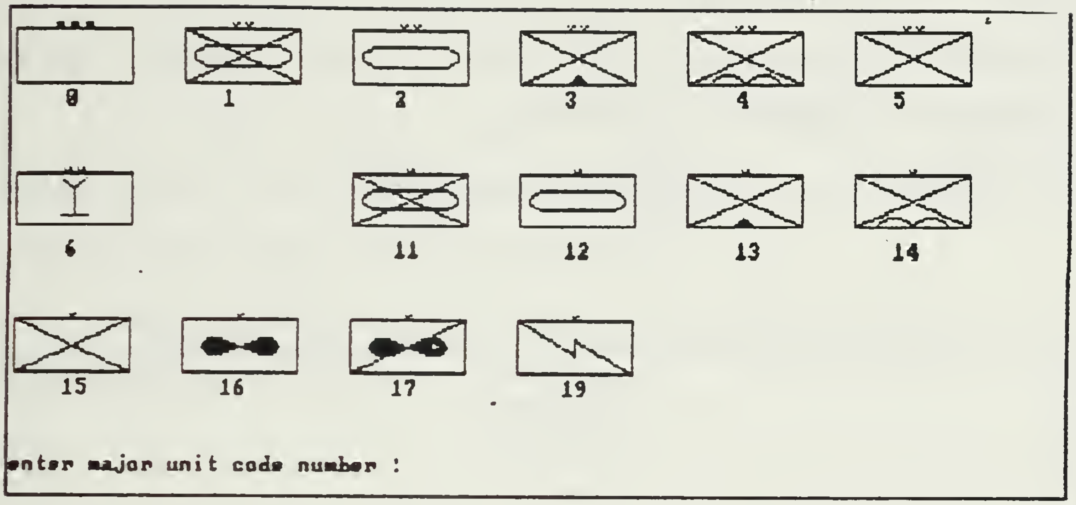
(C) Let us change the number back to 2

(M)

```
*******************************
***     Program TACMIX     ***
***        Options:        ***
*******************************

 - Create New Organization .......(1)

 - View Existing Organization ....(2)

 - Modify Existing Organization ..(3)


 - Terminate Program .............(9)


        Enter Number of Choice :
```

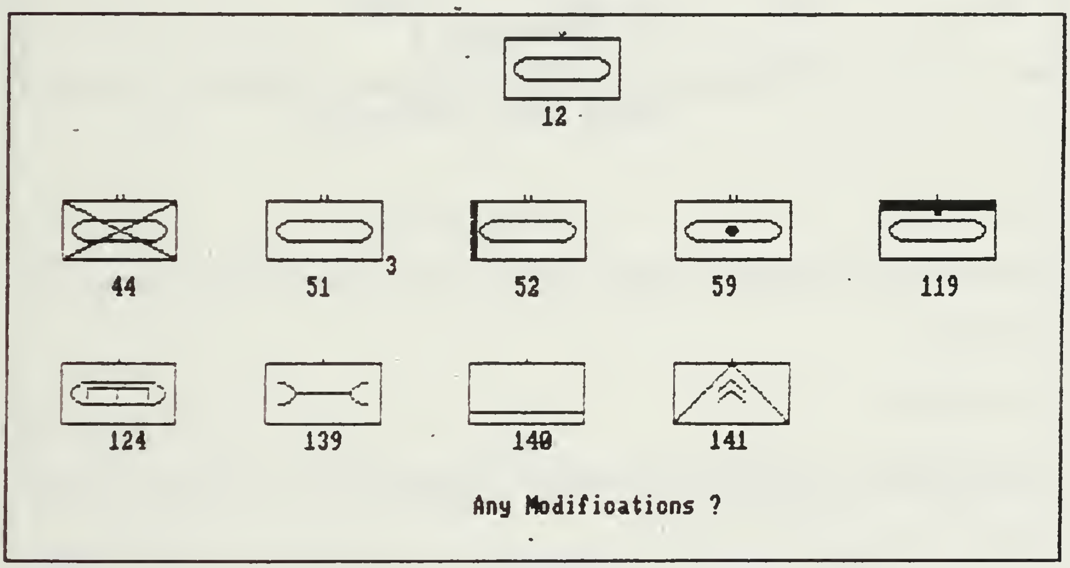
(I) 3 <enter>

(M)



(I) 12 <enter>

(M)



(I) y

(M) Which Modification .....

(I) 1 <enter>

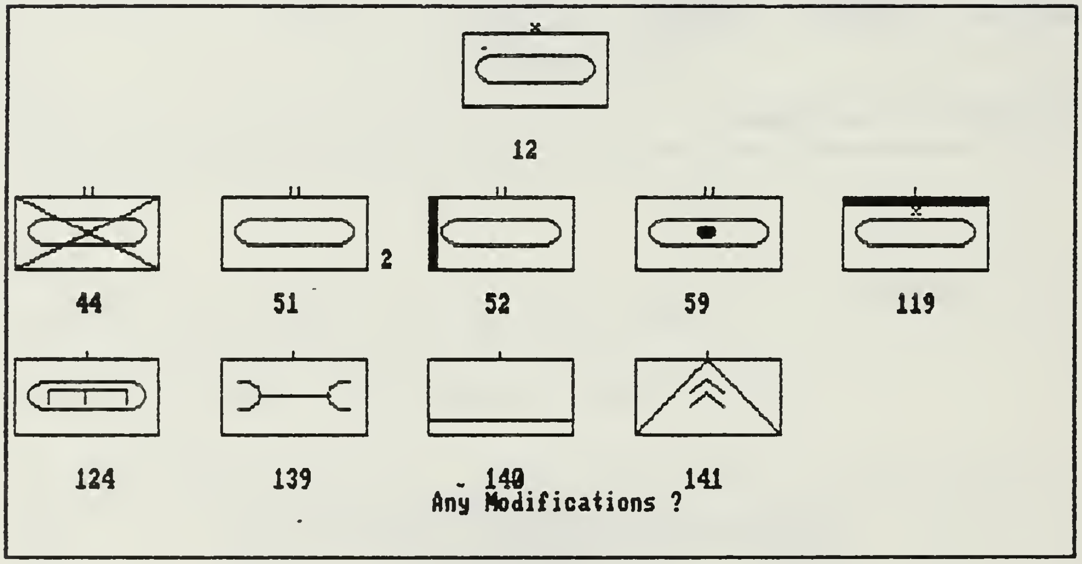
(M) Which Unit Number ?

(I) 51 <enter>

(M) Enter new number of units

(I) 2 <enter>

(M)



(I) n <enter>

(C) We are now in the main menu. Let us abort the program.

(I) 9 <enter>

This input brings us back to DOS.

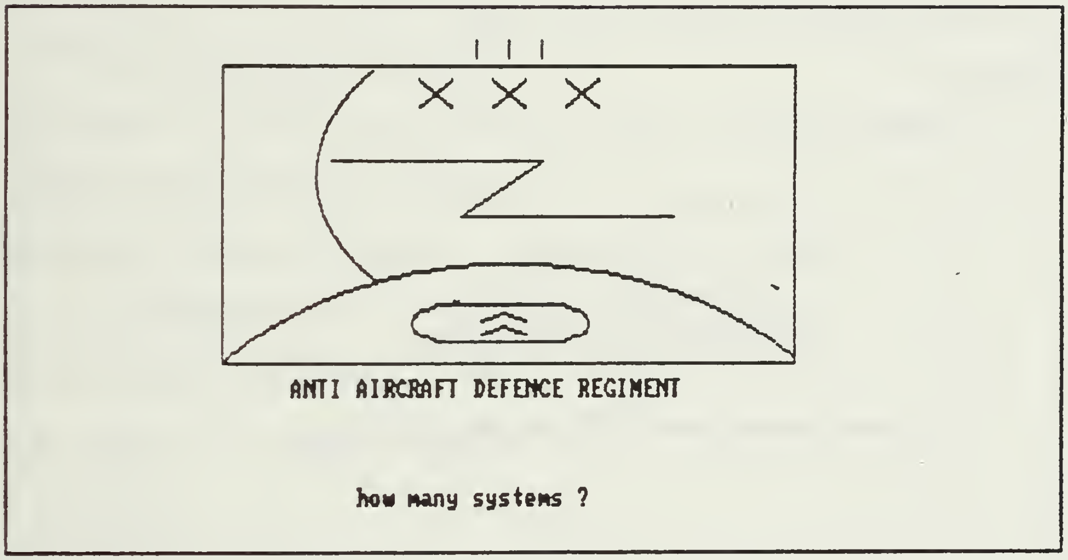
2. Program INMILDAT

INMILDAT is the graphical interface for CAESAR to provide the input of the distribution of the new systems within the corps. The program will represent all units and major units with their respective standardized tactical signs. Below each tactical sign is displayed the name of the unit.

To start INMILDAT type from the DOS prompt:

(I) C>INMILDAT <enter>

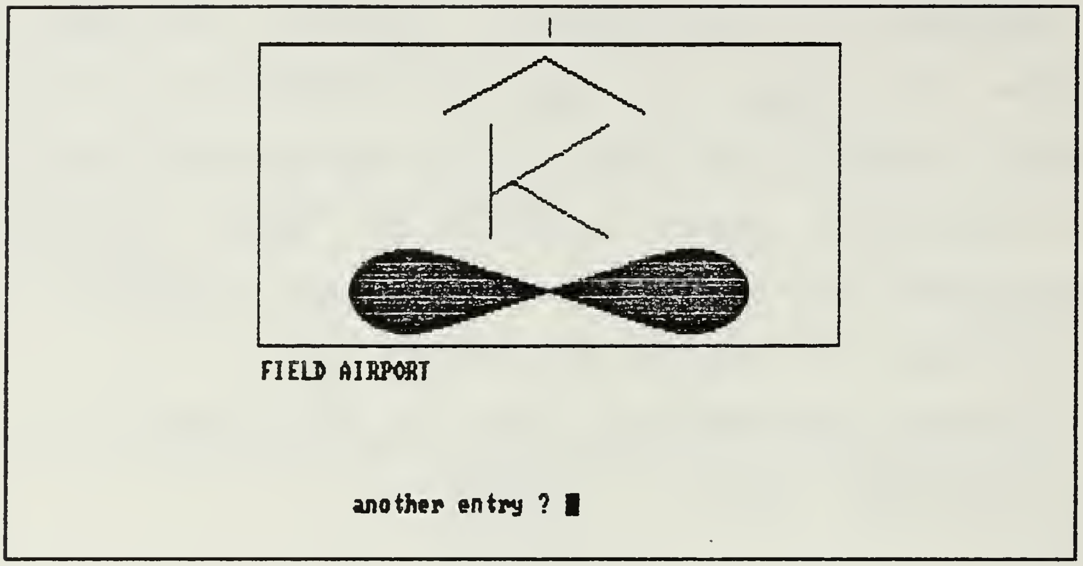
(M)



(I) 30 <enter>

(C) The program will successively display all unit types and ask for the number of systems in this unit until it reaches the last stored unit image:

(M)



(C) no systems in this unit type

(I) <enter>

(M)

To save Data, name File.
Default File Name is DISTRIBN

To accept Default press ENTER or enter File Name

(C) enter a name of your choice

(I) TESTFILE <enter>

(C) That is the end of INMILDAT

3. <u>Program INSYSDAT</u>

INSYSDAT is the interface for CAESAR to enter the relevant data about the new systems.

To start the program type from the DOS prompt:

(C) Let us enter a component called TEST ITEM for the system with the stock number 1234-12-999-0000.

(I) C>INSYSDAT <enter>

(M) Item Type(1=System, 2=Component, 3..8=Subcomponent, 9=Part)

(I) 2 <enter>

(M) National Stock Number:

(I) 1234-12-140-0000 <enter>

(M) Belongs to what item (next level stock number):

(I) 1234-12-999-0000 <enter>

(M) National Stock Number: 1234-12-140-0000

Item Description :

(I) Test Item <enter>

(M) How many of these Items:

(I) 5 <enter>

(M) Unit Price in US Dollars :$

(I) 253.98 <enter>

(M) MTBF (assumed) in Days:

(I) 120 <enter>

(M) Replacement Level (2=Org MN,3=DS/GS,4=Depot):

(C) assume replacement on ORG MN level

(I) 2<enter>

(M) Repair Level (0=no,2=Org MN,3=DS/GS,4=Depot):

(C) assume repair on DS/GS level

(I) 3 <enter>

(M) MTTR (Repair/Replace) in Hours :

(C) fractions are possible; (e.g. for 200 minutes enter 3.3)

(I) 3.3 <enter>

(M)

```
Type                          : Component
National Stock Number         : 1234-12-140-0000
Item Description              : Test Item
Unit Price in US Dollars      :$    253.98
Number of items               :  5
MTBF ( assumed )              :  120 days
Replacement Level             : Organ. Maintainance
Repair Level                  : Direct / General Support
MTTR (replace)                :    3.3 hours



  *******     Entries correct ? (y/n)     *******?
```

(I) y <enter>

(M) another entry ?

(C) For this sample run another entry is not necessary.

(I) n <enter>

This input brings us back to DOS.

4. Program ESTIMATE

ESTIMATE is the first of two simulation programs in CAESAR to compute the requirements of the system's logistics support organization. To start ESTIMATE type from the DOS prompt:

(I) C>ESTIMATE <enter>.

(M) Enter total number of failures to simulate (<=1000):

(C) let us chose 50 for this sample run

(I) 50 <enter>

(M) adjust paper in printer

(M) press a key to continue

(C) make sure you start with a new page on your printer

(C) and the printer is switched on and online.

(I) <enter>

(M)

> Enter desired value for the standard deviation of
> the delay distributions in per cent of the mean
>
> the default value is 10 per cent
> to accept press ENTER, else type percentage

(C) to use the default value of 10 % press ENTER key

(I) <enter>

(M)

```
Enter required information about the transportation times (one way):

    - on the user / org maint. level   ( 1 )
    - on the      DS / GS      level   ( 2 )
    - on the Depot / Industry  level   ( 3 )


    -        no more entries           ( 9 )
```

(C) enter for level Org Mn

(I) 1 <enter>

(M)

```
              transportation times (one way)

Enter desired distribution for the time delays:

uniform               ( 1 )
normal                ( 2 )
exponential           ( 3 )
Gamma                 ( 4 )
lognormal             ( 5 )
fixed                 ( 6 )


        default distribution is NORMAL
to accept default press ENTER else type number
```

(C) accept normal distribution as default

(I) <enter>

(M)

```
transportation times (one way)

On the user / org maint. level, enter for the normal distribution

the average (mean) value in hours
the default value is: 3


to accept default press ENTER else type value
```

(C) accept default

(I) <enter>

(C) back to the transportation delay level menu

(M)

```
Enter required information about the transportation times (one way):

    - on the user / org maint. level   ( 1 )
    - on the      DS / GS      level   ( 2 )
    - on the Depot / Industry  level   ( 3 )


    -        no more entries           ( 9 )
```

(C) for DS/GS level

(I) 2 <ENTER>

(C) now the same prompts as for the Org Mn level

(M) ... distribution ...

(I) <enter>

(M) ... default value ...

(I) <enter>

(C) back to the transportation delay menu

(C) for Depot level enter

(I) 3 <enter>

(C) now the same prompts as for the Org Mn level

(M) ... distribution ...

(I) <enter>

(M) ... default value ...

(I) <enter>

(C) back to the transportation delay menu

(C) no more entries here

(I) 9 <enter>

(M)

```
Enter required information about the admin time delays:

   - on the user / org maint. level     ( 1 )
   - on the      DS / GS      level     ( 2 )
   - on the Depot / Industry  level     ( 3 )


   -        no more entries             ( 9 )
```

(C) for the Org Mn level enter

(I) 1 <enter>

(M)

```
                admin time delays

     Enter desired distribution for the time delays:

     uniform                  ( 1 )
     normal                   ( 2 )
     exponential              ( 3 )
     Gamma                    ( 4 )
     lognormal                ( 5 )
     fixed                    ( 6 )


           default distribution is NORMAL
     to accept default press ENTER else type number
```

(C) let us change this distribution to exponential

(I) 3 <enter>

(M)

```
              admin time delays

  On the user / org maint. level, enter for the exponential distribution

              the average (mean) value in hours
              the default value is: 4


          to accept default press ENTER else type value
```

(C) accept default value and press

(I) <enter>

(C) now do the same for the other two levels

(M) Enter required ...

(I) 2 <enter>

(M) ... distribution ...

(I) 3 <enter>

(M) ... default value ...

(I) <enter>

(M) Enter required ...

(I) 3 <enter>

(M) ... distribution ...

(I) 3 <enter>

(M) ... default value ...

(I) <enter>

(M) Enter required ...

(C) you are finished with this part

(I) 9 <enter>

(C) you now choose the repair time distribution

(M)

```
Enter desired distribution for the repair time:

uniform             ( 1 )
normal              ( 2 )
exponential         ( 3 )
Gamma               ( 4 )
lognormal           ( 5 )
fixed               ( 6 )


the default distribution is LOGNORMAL

to accept press ENTER, else type number
```

(C) use default distribution lognormal

(I) <enter>

(M)

```
Enter desired value for the standard deviation

the default value is 10 per cent of the mean


to accept press ENTER, else type percentage
```

(C) accept default value

(I) <enter>

(M) Enter number of workhours per day

(C) for an average of 9 hours per day

(I) 9 <enter>

(M) Enter number of workdays per week

(C) for an average of 5 days per week

(I) 5 <enter>

(M) Enter system stock number (e.g. 1234-12-123-1234)

(C) use the CAESAR sample file

(I) 1234-12-100-0000 <enter>

(M) Compute Lambda for overall system

(M)

```
System Dislocation within Major Units

Input filename for system distribution file
Default filename is DISTRIBN

To accept default press ENTER or enter name
```

(C) use the default filename

(I) <enter>

(M)

```
==================================================
== Menu :  CHOICE OF MAJOR UNIT TYPE            ==
==================================================
==    Customer        # systems     selection   ==
==    -----------------------------------------  ==
==    MECH. BRIG      ( 98 )          :  1       ==
==    ARM'D BRIG      ( 98 )          :  2       ==
==    MOUNT.BRIG      ( 38 )          :  3       ==
==    MECH. DIV       ( 395 )         :  4       ==
==    DIV. UNITS      ( 125 )         :  5       ==
==    ARM'D. DIV      ( 395 )         :  6       ==
==    DIV. UNITS      ( 125 )         :  7       ==
==    CORPS           ( 1390 )        :  8       ==
==    CORPS UNITS     ( 285 )         :  9       ==
==    Reserve         ( 184 )         :  10      ==
==================================================

              Enter choice :
```

(C) the number of failures is 50. Therefore pick a major unit with less than 50 systems.

(I) 3 <enter>

(C) this was the last input for ESTIMATE.

(C) ESTIMATE generates various messages.

(C) Display when the MTBFs are drawn:

(M) Failure Time Generation

(C) The next message is updated for each simulation

(M) .. th Iteration for Simulation

(C) within each simulation this message is displayed:

(M) Subroutine Summation of consumed Items

(C) the last message will be:

(M) Elapsed Time for 50 Simulations : xxx Weeks

(M) PROGRAM END

The program has terminated correctly. We are back in the DOS environment.

The following six pages demonstrate the output of ESTIMATE with the data we just entered:

- parameters for delay distributions
- distribution of system within the corps
- summary of items and manhours on Org Mn level
- summary of items and manhours on DS/GS level
- summary of items and manhours on Depot level
- additional statistics

```
Number of Failures to be generated :   50


Distributions and Parameters for Transportational
=================================================
        and Administrative Delays
        =========================


transportation delays - level  user / org maint.
    distribution normal
Mean  =   3.000    Sigma =   0.300

transportation delays - level       DS / GS
    distribution normal
Mean  =   7.000    Sigma =   0.700

transportation delays - level  Depot / Industry
    distribution normal
Mean  =  16.000    Sigma =   1.600

administrative delays - level  user / org maint.
    distribution exponential
Lambda=   0.250

administrative delays - level       DS / GS
    distribution exponential
Lambda=   0.167

administrative delays - level  Depot / Industry
    distribution exponential
Lambda=   0.125
```

Dislocation of overload protector within Corps

Major Unit	# of systems
MECH. BRIG	90
ARM'D BRIG	90
MOUNT.BRIG	30
MECH. DIV	395
DIV. UNITS	125
ARM'D. DIV	395
DIV. UNITS	125
CORPS	1390
CORPS UNITS	205
Reserve	104
Total	1494

choice : 3 MOUNT.BRIG

Summary of Items and Manhours for 50 Simulations

For Level :Org. Maintenance

stock number	Item Description	# used
1234-12-130-0000	battery back-up	27
1234-12-131-0000	charging unit	27
1234-12-110-0000	case complete	13
1234-12-100-0001	wall mount	4
1234-12-120-0000	carrybag	6
1234-12-121-0000	bag	6

total consumed manhours on this level : 427.33 hrs

average manhrs within 4 weeks on this level : 140.22 hrs

average rounded number of mainten. personal : 1 repairman

For Level : DS/GS Maintenance

stock number	Item Description	# used
1234-12-131-0002	fuse 110V/2mA	10
1234-12-112-0000	monitor panel	8
1234-12-112-1000	function board	6
1234-12-131-0003	fuse 12V/.5A	12
1234-12-131-0001	capacitor 20V/.1mu	5
1234-12-121-0003	button	5
1234-12-112-0002	lamp mount	2
1234-12-113-0000	processing unit	5
1234-12-113-0003	output converter	2
1234-12-121-0002	cotton bag	1
1234-12-113-0002	data processor	3

total consumed manhours on this level : 196.76 hrs

average manhrs within 4 weeks on this level : 64.57 hrs

average rounded number of mainten. personal : 1 repairman

For Level :Depot Maintenance

stock number	Item Description	# used
1234-12-112-1003	relais	5
1234-12-112-1002	transformer	1

total consumed manhours on this level : 19.45 hrs

average manhrs within 4 weeks on this level : 6.38 hrs

average rounded number of mainten. personal : 1 repairman

Additional Statistics

total elapsed time for 50 simulations : 12.19 weeks

total number of 4 week periods for 50 simulations : 3.05

average system down time : 3.05 Days

average system availability : 95.00 per cent

average maintenance cost per system : $40.90

initial system cost for MOUNT.BRIG: $40766.00
(systems and spare parts for 4 weeks)

cost of additional 4 week supplies : $402.63

5. <u>Program QBEHAVE</u>

QBEHAVE is the second of the two simulation programs in CAESAR that compute the logistics requirements for new systems. QBEHAVE sends results to the printer. Thus it is necessary to switch the printer on and align the paper.

To start QBEHAVE, type from the DOS prompt:

(I) C>QBEHAVE <enter>

(M)

```
              *****************************
              **    Program Mode Menu    **
              *****************************


   ( 0 ) ... Check Queuing Behavior without
             Sensitivity Analysis (faster)

   ( 1 ) ... Check Queuing Behavior with
             Sensitivity Analysis

=============================================

                  Default Value is 0.
     To accept press ENTER, else enter value  :
```

(C) let us use the full computational power of CAESAR

(I) 1 <enter>

(M)

```
Enter sensitivity analysis bounds in per cent

The default value is 10.

To accept default press ENTER, else type value
```

(C) let us use the default value

(I) <enter>

(M)

```
Enter the stockout risk factor r
( 0.0 < r < 1.0 )

The default value is 0.1.

To accept default press ENTER, else type value
```

(C) let us use the default value

(I) <enter>

(M)

```
Enter ave. shipping time for ordered items in days

The default value is 5 days.

To accept default press ENTER, else type value
```

(C) let us use the default value

(I) <enter>

(C) QBEHAVE starts sorting data for level Org Mn

(M)

```
Level Org Mn

Sort Data Entries
```

(C) now assign servers on Org Mn level

(M)

```
number of servers for level Depot  :   1

number of servers for level DS/GS  :   1

number of servers for level Org Mn :   2

        Compute Queueing Behavior

              Level Org Mn

          Service Number    14
```

(C) assign servers on DS/GS and DEPOT level

(C) this screen looks like the one earlier, because the number of servers has not yet changed.

(M)

```
number of servers for level Depot  :   1

number of servers for level DS/GS  :   1

number of servers for level Org Mn :   2

        Compute Queueing Behavior

              Level DEPOT
```

(C) one combination of servers has been calculated. this continues until calculations with all combinations have been performed. Eventually, with two repairmen on each maintenance level, the screen looks like

(M)

```
number of servers for level Depot  :   2

number of servers for level DS/GS  :   2

number of servers for level Org Mn :   2


        Compute Queueing Behavior


         Service Number      28
```

(C) each combination of repairmen on the different levels is printed

(C) QBEHAVE starts analyzing the spare part consumption

(M)

Ordering of Part Consumption Times

(C) the final results of this process will be printed

(M)
```
*****************
** Program End **
*****************
```

(C) this concludes the program QBEHAVE

(C) the output contains the following information:

- number of customers
- number of failures
- number of servers for each level
- number of delays for each level
- max. queue length for each level
- total waiting time for each level
- percentages servers are busy for each level
- total elapsed time and number of 4 week periods

for the service organization and

- national stock number
- initial stock level
- final stock level
- minimum stock level

for the supply organization.

The following pages are the output of QBEHAVE's sample run with data created earlier by ESTIMATE.

Eight pages show different combinations of numbers of repairmen on the three maintenance levels. Compare the waiting times and the percentages of time the servers are busy!

The last of the output pages shows the behavior in terms of spare parts. Observe the bottleneck of insufficient stocks for item 13 in all of the combinations (the debits were only a maximum of one each). One might want to increase the stock level for this item by one.

Program Start Time is : 06-26-1987 11:11:03

Queuing Behavior for Varying Numbers of Servers PAGE 1

```
LEVEL ORG MN :

no of customers    =     30
no of failures     =     50
no of servers      =      1
no of delays       =     41
max queue length   =      4 customers
total waiting time =      413.94 hours

average  waiting time per  waiting customer =     10.10 hours
expected waiting time per entering customer =      8.28 hours

percentage server  1 is busy = 75.4 %

LEVEL DS/GS :

no of servers      =      1
no of delays       =     11
max queue length   =      4 customers
total waiting time =       59.95  hours

average  waiting time per  waiting customer =      5.45 hours
expected waiting time per entering customer =      1.20 hours

percentage server  1 is busy = 31.8 %

LEVEL DEPOT :

no of servers      =      1
no of delays       =      0
max queue length   =      0 customers
total waiting time =        0.00  hours

average  waiting time per waiting  customer =       0.00 hours
expected waiting time per entering customer =      0.00 hours

percentage server  1 is busy =  3.1 %

Total elapsed time for    50 failures =  12.59 weeks

Total number of 4 week periods for   50 failures =      3.1
```

Summary of Item Consumption and Stock Levels PAGE 2

	STOCK NO.	INITIAL STOCK	FINAL STOCK	MIN. STOCK
1	1234-12-130-0000	9	9	0
2	1234-12-131-0000	9	9	0
3	1234-12-131-0002	3	2	0
4	1234-12-110-0000	4	4	0
5	1234-12-112-0000	3	3	0
6	1234-12-112-1000	2	2	0
7	1234-12-112-1003	2	1	0
8	1234-12-100-0001	1	1	0
9	1234-12-131-0003	4	4	0
10	1234-12-131-0001	2	1	0
11	1234-12-120-0000	2	2	0
12	1234-12-121-0000	2	2	0
13	1234-12-121-0003	2	1	-1
14	1234-12-112-0002	1	1	0
15	1234-12-113-0000	2	2	0
16	1234-12-113-0003	1	1	0
17	1234-12-121-0002	1	1	0
18	1234-12-112-1002	1	1	0
19	1234-12-113-0002	1	1	0

LEVEL ORG MN :

```
no of customers    =    30
no of failures     =    50
no of servers      =     2
no of delays       =    12
max queue length   =     3 customers
total waiting time =      23.21 hours

average  waiting time per  waiting customer =      1.93 hours
expected waiting time per entering customer =      0.46 hours

percentage server  1 is busy = 45.9 %
percentage server  2 is busy = 31.1 %
```

LEVEL DS/GS :

```
no of servers      =     1
no of delays       =    11
max queue length   =     2 customers
total waiting time =      69.05  hours

average  waiting time per  waiting customer =      6.28 hours
expected waiting time per entering customer =      1.38 hours

percentage server  1 is busy = 32.5 %
```

LEVEL DEPOT :

```
no of servers      =     1
no of delays       =     0
max queue length   =     0 customers
total waiting time =       0.00  hours

average  waiting time per waiting  customer =       0.00 hours
expected waiting time per entering customer =      0.00 hours

percentage server  1 is busy =  3.2 %

Total elapsed time for    50 failures =  12.33 weeks

Total number of 4 week periods for   50 failures =      3.1
```

	STOCK NO.	INITIAL STOCK	FINAL STOCK	MIN. STOCK
1	1234-12-130-0000	9	9	0
2	1234-12-131-0000	9	9	0
3	1234-12-131-0002	3	2	0
4	1234-12-110-0000	4	4	0
5	1234-12-112-0000	3	3	0
6	1234-12-112-1000	2	2	0
7	1234-12-112-1003	2	2	0
8	1234-12-100-0001	1	1	0
9	1234-12-131-0003	4	3	0
10	1234-12-131-0001	2	1	0
11	1234-12-120-0000	2	2	0
12	1234-12-121-0000	2	2	0
13	1234-12-121-0003	2	1	-1
14	1234-12-112-0002	1	1	0
15	1234-12-113-0000	2	2	0
16	1234-12-113-0003	1	1	0
17	1234-12-121-0002	1	1	0
18	1234-12-112-1002	1	1	0
19	1234-12-113-0002	1	1	0

```
LEVEL ORG MN :

no of customers    =     30
no of failures     =     50
no of servers      =      1
no of delays       =     41
max queue length   =      4 customers
total waiting time =      413.94 hours

average  waiting time per  waiting customer =     10.10 hours
expected waiting time per entering customer =      8.28 hours

percentage server  1 is busy = 75.4 %

LEVEL DS/GS :

no of servers      =      2
no of delays       =      0
max queue length   =      0 customers
total waiting time =        0.00  hours

average  waiting time per  waiting  customer =      0.00 hours
expected waiting time per entering customer =      0.00 hours

percentage server  1 is busy = 31.2 %
percentage server  2 is busy =  0.6 %

LEVEL DEPOT :

no of servers      =      1
no of delays       =      0
max queue length   =      0 customers
total waiting time =        0.00  hours

average  waiting time per waiting  customer =      0.00 hours
expected waiting time per entering customer =      0.00 hours

percentage server  1 is busy =  3.1 %

Total elapsed time for    50 failures =  12.59 weeks

Total number of 4 week periods for   50 failures =      3.1
```

	STOCK NO.	INITIAL STOCK	FINAL STOCK	MIN. STOCK
1	1234-12-130-0000	9	9	0
2	1234-12-131-0000	9	9	0
3	1234-12-131-0002	3	2	0
4	1234-12-110-0000	4	4	0
5	1234-12-112-0000	3	3	0
6	1234-12-112-1000	2	2	0
7	1234-12-112-1003	2	1	0
8	1234-12-100-0001	1	1	0
9	1234-12-131-0003	4	4	0
10	1234-12-131-0001	2	1	0
11	1234-12-120-0000	2	2	0
12	1234-12-121-0000	2	2	0
13	1234-12-121-0003	2	1	-1
14	1234-12-112-0002	1	1	0
15	1234-12-113-0000	2	2	0
16	1234-12-113-0003	1	1	0
17	1234-12-121-0002	1	1	0
18	1234-12-112-1002	1	1	0
19	1234-12-113-0002	1	1	0

```
LEVEL ORG MN :

no of customers    =     30
no of failures     =     50
no of servers      =      2
no of delays       =     12
max queue length   =      3 customers
total waiting time =       23.21 hours

average  waiting time per  waiting customer =       1.93 hours
expected waiting time per entering customer =       0.46 hours

percentage server  1 is busy = 45.9 %
percentage server  2 is busy = 31.1 %

LEVEL DS/GS :

no of servers      =      2
no of delays       =      2
max queue length   =      1 customers
total waiting time =        1.74  hours

average  waiting time per  waiting customer =       0.87 hours
expected waiting time per entering customer =       0.03 hours

percentage server  1 is busy = 28.1 %
percentage server  2 is busy =  4.3 %

LEVEL DEPOT :

no of servers      =      1
no of delays       =      0
max queue length   =      0 customers
total waiting time =        0.00  hours

average  waiting time per waiting  customer =        0.00 hours
expected waiting time per entering customer =       0.00 hours

percentage server  1 is busy =  3.2 %

Total elapsed time for    50 failures =  12.33 weeks

Total number of 4 week periods for   50 failures =      3.1
```

	STOCK NO.	INITIAL STOCK	FINAL STOCK	MIN. STOCK
1	1234-12-130-0000	9	9	0
2	1234-12-131-0000	9	9	0
3	1234-12-131-0002	3	2	0
4	1234-12-110-0000	4	4	0
5	1234-12-112-0000	3	3	0
6	1234-12-112-1000	2	2	0
7	1234-12-112-1003	2	2	0
8	1234-12-100-0001	1	1	0
9	1234-12-131-0003	4	3	0
10	1234-12-131-0001	2	1	0
11	1234-12-120-0000	2	2	0
12	1234-12-121-0000	2	2	0
13	1234-12-121-0003	2	1	-1
14	1234-12-112-0002	1	1	0
15	1234-12-113-0000	2	2	0
16	1234-12-113-0003	1	1	0
17	1234-12-121-0002	1	1	0
18	1234-12-112-1002	1	1	0
19	1234-12-113-0002	1	1	0

APPENDIX B

USER'S GUIDE

The summary of CAESAR in this user's guide enables the user to use CAESAR without need of the theoretical overhead in the main part of the study.

Each program in CAESAR will be explained in detail. Inputs and outputs are clearly marked and the user can easily follow the program flow.

All programs of CAESAR were written and compiled with the program TURBO BASIC [Ref. 6].

1. Introduction

a. Purpose of CAESAR

The programs contained in CAESAR were written for the IBM XT/AT or compatible computers to estimate the logistics requirements for a new system in terms of manpower and supply stocks.

CAESAR, (Computer Aided Evaluation of Supply and support on All Repair levels), uses inputs like system distribution, system characteristic data and major unit data to create a scenario within any major unit that provides conclusions about the behavior of a given number of failing systems in this scenario before the system is introduced into the Armed Forces (or any other customer with similar structures).

CAESAR is strictly thought of as a decision making aid, it must not be confused with artificial intelligence. The output of CAESAR is able to explicitly suggest reasonable logistical structures and to show alternatives to the suggested solution. Thus it can be a valuable tool in the hands of a manager, responsible of introducing new equipment and the accompanying supporting network.

System requirements for CAESAR are at least 512 kByte of memory, one floppy disk drive and a harddisk. The use of extended or expanded memory and the installation of a RAM-Disk will speed up the programs significantly.

Operating system is DOS version 2.10 or higher. A printer has to be connected for the programs ESTIMATE and QBEHAVE, otherwise the programs will generate an error message and abort.

For the following instructions the use of a harddisk as drive C is assumed.

b. <u>How To Get Started</u>

To be able to use the program, boot-up your computer and after getting the DOS-prompt "C>" create a sub-directory on your harddisk with the following command:

```
C>MD\CAESAR.        (You do not have to type C> !)
```

Make this sub-directory your current directory:

```
C>CD\CAESAR.
```

Now insert the CAESAR I floppy disk into your A-drive. If you are not sure which drive is the A-drive consult your computer manual.

To copy the program files from the CAESAR I floppy disk to the harddisk enter the following command:

```
C>COPY A:*.* C:.
```

This command will copy all files from the floppy drive A to the sub-directory CAESAR on your harddisk C. Repeat the above procedure for the CAESAR II and III floppy disks.

Now you are ready to enter the structural data for the major units and the new system. These data will later be used in the programs that perform the simulations.

2. **Program TACMIX.EXE**

a. Start TACMIX

TACMIX is the graphic interface for CAESAR to provide the input of major unit structures. The program is written and compiled in TURBOBASIC. TACMIX is found on CAESAR disk I. Familiarity with the common tactical signs is assumed (see app. C). The program will present the tactical signs of both the US and West German Armies.

To start TACMIX simply type:

C>TACMIX.

The program will respond with the main menu as shown in figure B.1.

After your choice the display of all available major unit types will appear (except for option 9). Figure B.2 shows this screen. Now you have to chose the major unit you want to work with.

```
****************************
***     Program TACMIX    ***
***       Options:        ***
****************************

 - Create New Organization .......(1)

 - View Existing Organization ....(2)

 - Modify Existing Organization ..(3)


 - Terminate Program .............(9)


        Enter Number of Choice :
```

Figure B.1 Main Menu Program TACMIX

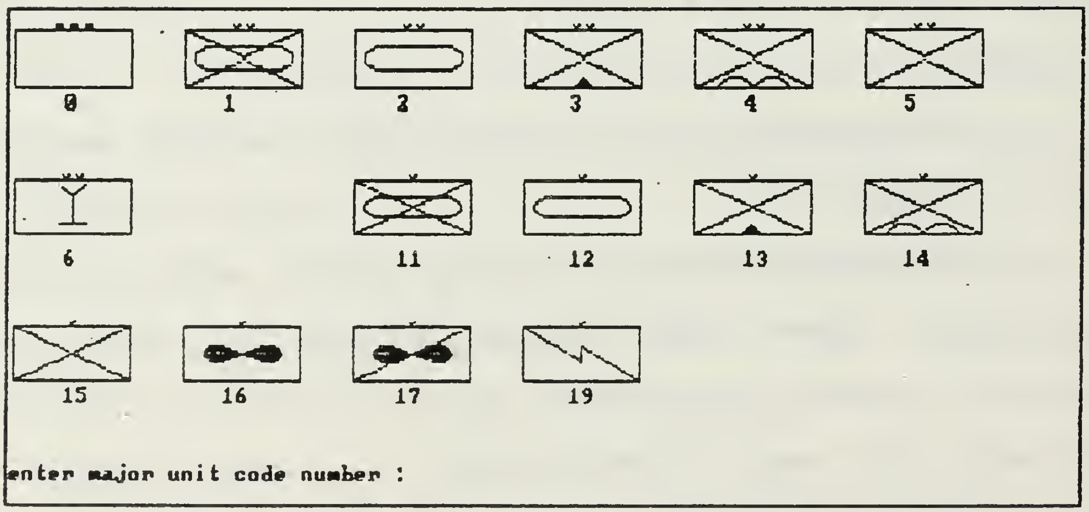


Figure B.2 Major Units Display Screen

b. Option 1: Create New Organization

In option 1, create a new organization, the program will present several screens like the one in Figure B.2 with all possible unit types for your selected major unit.

On the bottom of the screen you will read the following message:

"enter unit code number (ENTER for next frame) : ".

If you decided you need any of the depicted unit types for your major unit, simply enter the code number below the symbol. End your input with the ENTER key. The program wants to know how many of these unit types are in the major unit you chose and displays this message on the bottom of the screen:

"enter number of these units (default=1) :".

You can enter any number now. If you only have one of this unit type in your major unit, simply press the ENTER key and TACMIX will set the number to 1.

Repeat this for all unit types on this screen that you want to include in your major unit.

When you have entered all wanted units from the current graphic screen, press the ENTER key at the unit code number prompt to see the next available unit symbols. Take your choices as before. If you press the ENTER key at the unit code number prompt after the last screen is displayed, TACMIX will store your input in a sequential data file.

Depending on the type of major unit you chose, the file will be called "CORPS", "DIVx" or "BRIGxx", where the "x" or "xx" stand for the number of the major unit, you chose.

E.g. the contents of file "BRIG12" might look like depicted in Figure B.3.

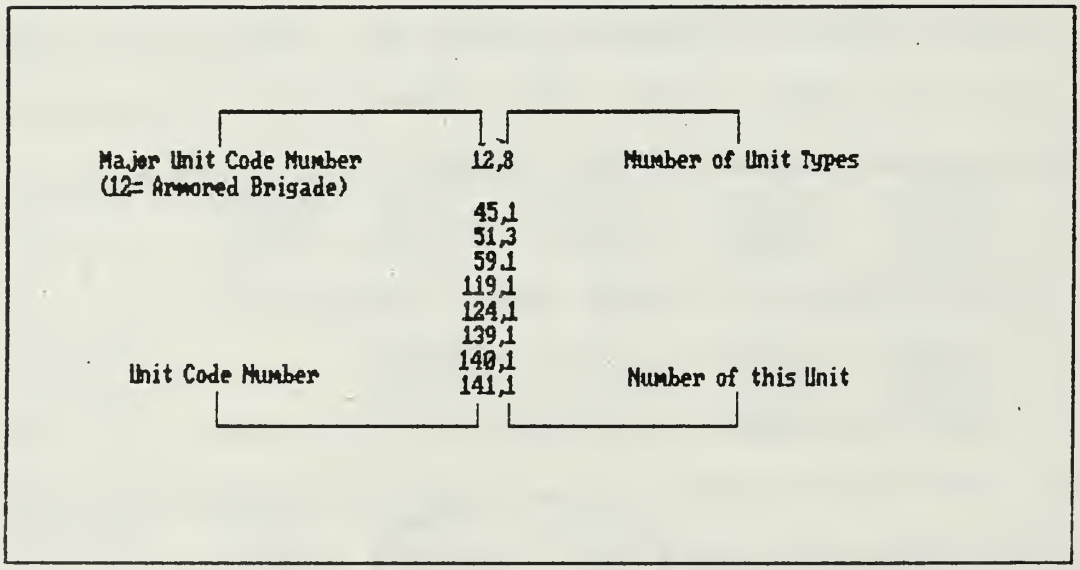


Figure B.3 Sample Contents of a Major Unit File

Working samples for all major unit types are included on the CAESAR data disk.

c. <u>Option 2: View Existing Organization</u>

In option 2, view existing organization, the program will display the screen shown in Figure B.2. After your choice of the major unit, the program will access the major unit data file and display all of the unit types within this major unit on consecutive screens. Figure B.4 gives an example for major unit 14 (airborne brigade).

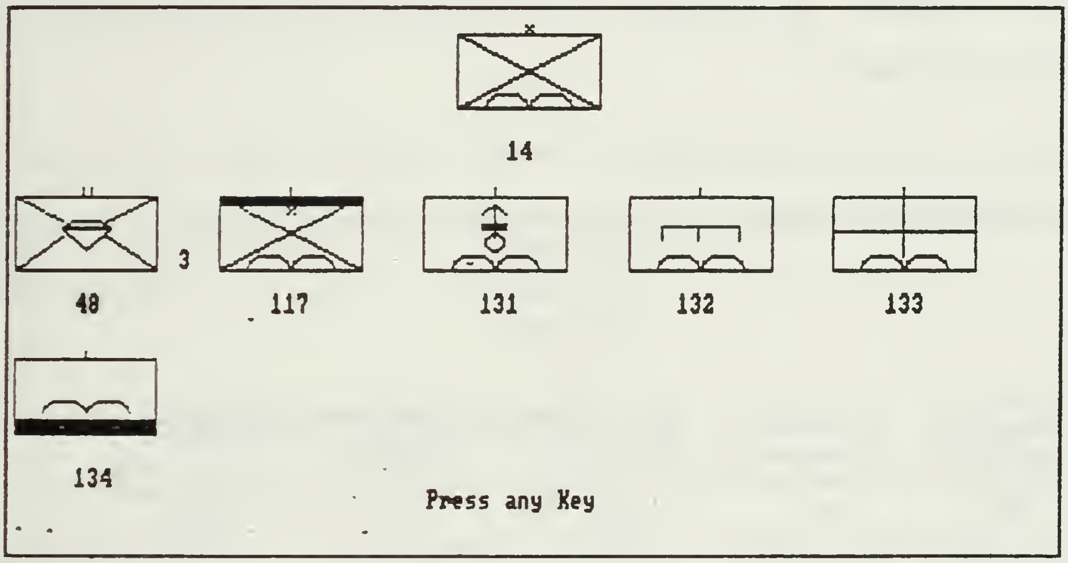


Figure B.4 View Airborne Brigade

If there are more than one units of a specific type, then there will be an index outside the lower right corner of the symbol, indicating the number of these units. In Figure B.4 there are three unit types 48 and one each of the others.

To get to the next screen, simply press any key. To get back to the main menu after the last screen for your chosen major you press any key.

d. <u>Option 3: Modify Existing Organization</u>

In option 3, modify existing organization, the program will display the screen shown in Figure B.2. After your choice of the major unit, the program will access the major unit data file and display all of the unit types within this major unit on consecutive screens. Figure B.5 gives an example for major unit 12 (armored brigade).

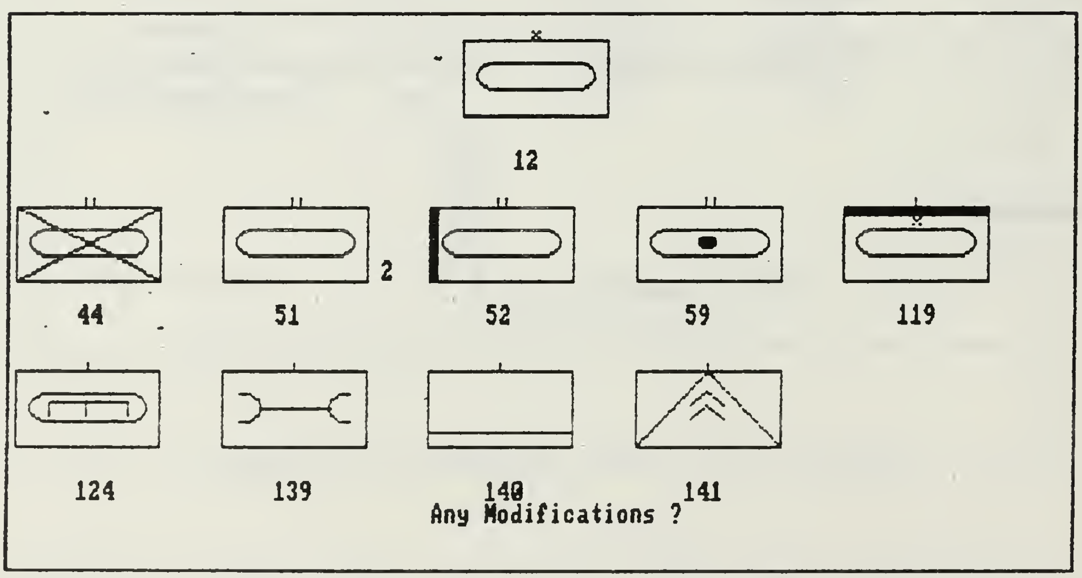


Figure B.5 Modify Major Unit 12

For each screen you will be asked:

"Any Modifications ?".

If you respond with "n" or "N" (do not type the quotes), then the next screen will be displayed. If there is only one screen or you have already reached the last one, "n" or "N" will get you back to the main menu. If you press any other key you will be asked:

"Which Modification ? (0=erase,1=change number of units,2=add a unit)".

You decide now, which kind of modification to perform and enter the appropriate number. The program will prompt you:

"Which Unit Number ?",

and you enter the code number for the unit to be changed.

If you wanted to <u>erase</u> a unit, the screen will now be redrawn without the erased unit - the file has been modified in the meantime.

For the <u>change of number</u> of units you see this prompt:

"Enter new number of units".

After you entered the required number the screen will be redrawn and the modification will have been made both on the screen and in the file.

If you want to add another unit type you enter the unit code number at the:

"Which Unit Number"

prompt. Now you will be asked to

"Enter number of new units".

When you enter this number the data file will be updated and the screen will be redrawn including the new unit type and the correct number of this type outside the lower right corner of the symbol for this unit type. Since the unit code numbers are in ascending order, the unit you have added might be on a later screen.

e. Option 9: Terminate Program

When you are finished with the program you chose option 9, termination of the program. Do not turn off your computer without running option 9. You might loose data if not all file are closed.

After you chose option 9, the program will get you back to the DOS prompt

"C>".

3. Program INMILDAT.EXE

a. Start INMILDAT

INMILDAT is the graphic interface for CAESAR to provide the input of the distribution of the new systems within the corps. INMILDAT can be found on the CAESAR I disk. Familiarity with the common tactical signs is assumed[1]. The program will present the most commonly used tactical signs of both the US and West German Armies.

INMILDAT will create a new data file with the unit code numbers of the selected units and the respective number of systems. If you want to keep the previous data file, make sure to select another name for the new one or rename the old one and type from the DOS prompt "C>":

"C>RENAME DISTRIBN DISTOLD" .

This DOS command will rename the data file "DISTRIBN" into "DISTOLD". You can use any other name up to 8 characters, as long as it is not the same as one already in use. Now, when you start INMILDAT, you can use the file name "DISTRIBN" without loosing any previous data.

---

[1] see Appendix C

To start INMILDAT simply type:

C>INMILDAT.

The program will immediately respond with the first unit symbol.

b. Operate INMILDAT

The first graphic screen will show the anti-aircraft defence regiment, code number 21. Figure B.6 shows this screen.

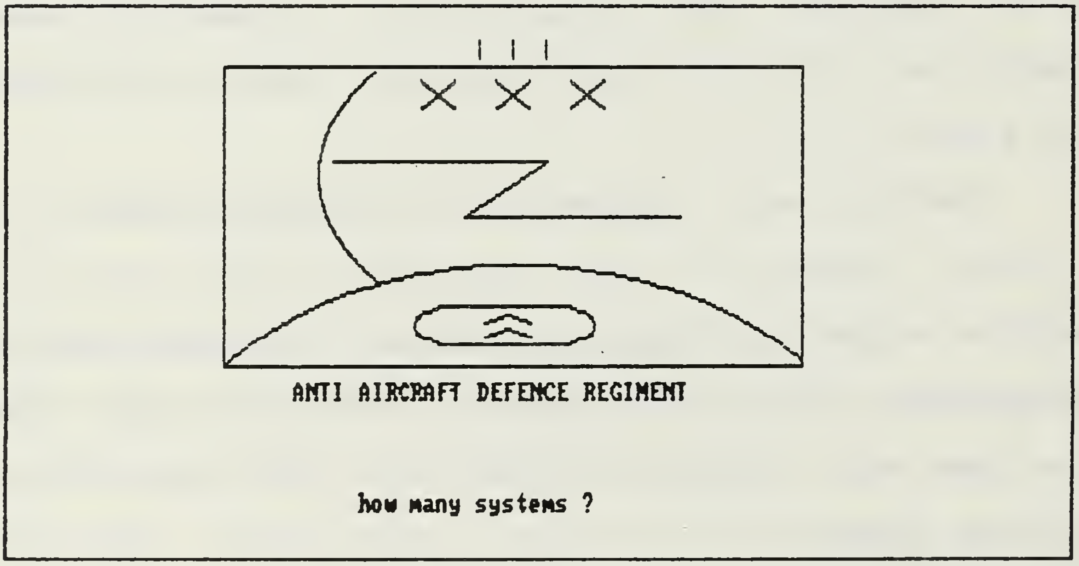


Figure B.6 Sample INMILDAT Screen

You will be asked:

"how many systems"

If there will be some of the new systems in this unit type, enter the number of the systems. If not, press ENTER to see the next unit symbol. Once you enter a number you will be asked to verify:

"entry ok".

Any keystroke except "n" or "N" will accept the input and the program will ask:

"another entry".

Any keystroke except "n" or "N" will lead to the next available symbol until <u>all</u> possible symbols have been presented in the described manner.

After the last symbol you will be asked:

"do you want a reserve".

If you decide to have one,answer with "y" or "Y". The next required information is:

"in per cent or absolute (p/a)".

Now enter the appropriate letter (P or A). Should you decide for per cent, the program will ask you to:

"enter percentage (e.g.:25)".

Please enter now the value of the percentage, not the fraction of 100. Given the example, for 25 % (.25) you would enter the number 25. Do not enter .25 unless you want .25 %.

In the case of a reserve in absolute numbers the program will ask you to:

"enter number of systems".

After this entry you will be asked to enter a name for the data file that will contain the system distribution data:

"To save data name file.
Default file name is DISTRIBN.
To accept default press ENTER or enter file name".

Now you can enter the file name you prefer or use the above default file name. If you decide to use the default, simply press the ENTER key and the data will be saved into "DISTRIBN". Otherwise they will be save into the file you named. You can enter any name up to 8 characters.

! Do not use a name that has already been used before ! The new file will overwrite the old one and all previous data will be lost !

4. Program INSYSDAT.EXE .

a. Input Format for INSYSDAT

INSYSDAT is the interface for CAESAR to enter the characteristic data about the new systems. To be able to enter the appropriate inputs you have to know the formats of these inputs. You will be asked for the following entries
(underlined entries will be explained in detail) :

<u>Item Type</u>
<u>National Stock Number</u>
Item Description
Number of Items
Price in Dollars
MTBF in days
<u>Replacement Level</u>
<u>Repair Level</u>
MTTR in hours.

<u>Item Type</u>: to find the required number you have to break up the system (see figure B.7).

The system (level 1) consists of several components (level 2). Each of these contains subcomponents (level 3) which themselves contain

subcomponents (level 4) etc. If an item cannot be split up further, then it is a part (level 9). Parts can be found on all levels except the system (level 1).
Figure B.7 shows the schematic structure of a system.

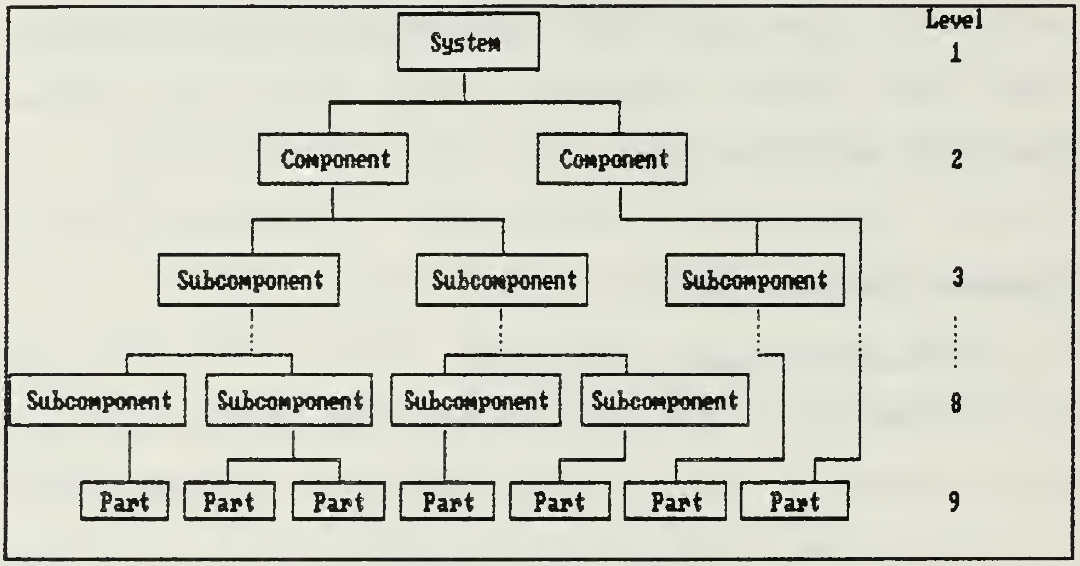


Figure B.7 System Structure

The above schematic requires the classification of all items, of which the system consists, in one of the levels 2 - 9. Use this classification as input for the "Item Number".

<u>EXAMPLE:</u>

Consider a car as the system. It will be coded "1". Now take the body. It is a component of the car and will be coded "2". The rear hood is a subcomponent "3" and the lock on the hood is a subcomponent "4". The opening mechanism of the lock is subcomponent "5" and consists of parts "9" like bolts, nuts, etc.

National Stock Number (NSN): this is a 13-digit number in the following format (e.g.):

1234-12-123-1234,

where the fifth and sixth digit (-12-) should be the same for all items in the system, because the program will not use these to discriminate between items. These two digits are the code for the country where the item is built. The name of the random access data file that is generated for each system, component and subcomponent is

AAAADDDD.CCC

derived from the National Stock Number

AAAA-BB-CCC-DDDD.

Replacement Level: this is the level where the failed item is replaced at a maintenance facility. For the system it is the level where the component is replaced, for the component, where the subcomponent is replaced etc. Parts can be replaced on any level.

These are the possible levels:

Organizational	Maintenance	(Org Mn)	level "2"
Direct/General	Support	(DS/GS)	level "3"
Depot/Industry	Maintenance	(Depot)	level "4".

Please enter the code number for the appropriate level.

Repair Level: this is the level where the replaced item is repaired at a maintenance facility. For the system it is the level where the component is repaired, for the component, where the subcomponent is repaired etc. Parts can be repaired on any level.

These are the possible levels:

No Repair			level "0"
Organizational Maintenance		(Org Mn)	level "2"
Direct/General Support		(DS/GS)	level "3"
Depot/Industry Maintenance		(Depot)	level "4".

Please enter the code number for the appropriate level.

b. Start INSYSDAT

To start INSYSDAT simply type at the DOS prompt:

C>INSYSDAT.

c. Operate INSYSDAT

For each entry you will find a sample input. To become familiar with the program, you are welcome to use these examples.

The first program prompt will ask you for:

"Item Type (1=System, 2=Component,
3...8=Subcomponent, 9=Part)".

You are required to enter the type classification code (1 to 9) as described in paragraph a.

E.g. for a component enter 2.

The next prompt will ask you for:

"National Stock Number :".

Please enter the stock number in the format described in the paragraph a (AAAA-BB-CCC-DDDD).

E.g. enter a stock number like 1234-12-140-0000.

If the item type code is 2 or greater, the program will now ask you:

"belongs to what item (next level stock number):"

You want to enter the national stock number of the next higher level item here, which in the case of a component would be the system, in the case of a level 3 subcomponent the component etc.

For the system in this example enter 1234-12-100-0000.

With this information the program is able to check if this item is already stored under the same next level item. If it is, you will see the following error message:

"input error:item already stored
press any key to continue"

and the program will start from the beginning. If the item is not yet stored the program will proceed and you will be asked to enter:

"Item Description :".

E.g. enter Test Item.

The next inputs only apply to item types 2 to 9:

"Unit Price in US Dollars : $".

Give the price for one item.

E.g. type 253.98 .

Now the program will ask you for:

"How many of these Items :"

Enter the actual number of these items within the next level item. Do not enter the overall total number of this item in the complete system.

E.g. enter 5 .

The next prompt will be:

"MTBF (assumed) in Days :".

Enter here either empirical data, an estimate or the proposed MTBF by the manufacturer.

E.g. type 120.

The next input will be:

"Replacement Level (2=Org Mn,3=DS/GS,4=Depot)".

Use the code number (2 to 4) for the replacement level as described in paragraph a.
E.g. enter 2 for organizational maintenance.
You will now be asked for:

"Repair Level (0=no,2=Org Mn,3=DS/GS,4=Depot)".

Again use the code number (0,2 to 4) for the repair level as described in paragraph a. For a part, repair level 0 is set by default and displayed on the screen.
E.g. enter 3 for direct/general support.
The next prompt will ask you for:

"MTTR (Repair/Replace) in Hours :"

Enter here the actually needed repair/replace times, an estimate or numbers given by the manufacturer in hours.
E.g. for 200 minutes you enter 3.3 .

You now have completed the input phase. To verify your inputs the screen in Figure B.8 with a summary of your inputs will be displayed.

Figure B.8 shows the data you just entered to verify the input.

```
Type                         : Component
National Stock Number        : 1234-12-140-0000
Item Description             : Test Item
Unit Price in US Dollars     :$    253.98
Number of items              :  5
MTBF ( assumed )             :  120 days
Replacement Level            : Organ. Maintainance
Repair Level                 : Direct / General Support
MTTR (replace)               :    3.3 hours



  *******    Entries correct ? (y/n)    *******?
```

Figure B.8 Summary of Inputs for INSYSDAT

If you want to accept the inputs, press "y" or "Y". If not press "n" or "N".

After accepting the data are written into random access files. This file is for item type code numbers 2 to 9 the next higher item stock number converted into a file name as described in paragraph b. For the system it is its own stock number converted into that format.

The last prompt in this sequence is:

"another entry ?".

"Y" or "y" get you to the program starting point and you can enter the next set of data. Any other key will end the program.

d. Error Messages

INSYSDAT handles three different types of input errors:

Invalid Repair Level

If the repair level for any item is less than the replacement level you will see the following error message:

"input error: repair level invalid
press any key to continue"

The program assumes an earlier input error and starts all over again.

Item Already Stored

If an item with the same stock number has already been stored under the given next level item stock number then this message will be dispayed:

"input error: item already stored
press any key to continue

The program starts from the beginning.

Number Exceeds Maximum

The number of data sets in the random access files is restricted to 95 for each file to guarantee reasonable run time of the program. If the entered data set exceeds this number you will see this message:

"input error: number of items exceeds maximum
press any key to continue"

To avoid this message recheck the structure of the system. It might be possible to define another item to decrease the number of items in this specific level.

Make sure that all data files for the system, the components and the subcomponents are in the same directory. Otherwise the program is not able to search for already stored items. Furthermore the use of the data files by the programs ESTIMATE and QBEHAVE requires that they will be in the same directory as these programs.

With the input of these data the process of generating data files is complete and the data can now be evaluated.

4. **Program ESTIMATE.EXE**

a. Start ESTIMATE

ESTIMATE is one of the two major programs in CAESAR to simulate and compute logistics requirements. The other one is QBEHAVE. ESTIMATE combines the data in the major unit structur files, created by TACMIX, the data in the system distribution file, created by INMILDAT and the system data, created by INSYSDAT. From these data it simulates the new systems in the chosen major unit for a given number of failures. After ESTIMATE finishes the SAMPSIZE trials for all levels of maintenance the number of manhours and comsumed parts as well as all related times will be known.

The generation of the various times is based on the assumption that all times have known distributions whose parameters are also known. If you are not sure about the distribution or the parameters for an input, take a guess for the distribution and then use the default values. That should give reasonable results.

For MTTR and MTBF the distributions are assumed to be exponential with the given values as means.

ESTIMATE requires a printer for the output, otherwise DOS will generate an error message and abort.

To start ESTIMATE simply type at the DOS prompt:

C>ESTIMATE.

b. Operate ESTIMATE

As soon as you start ESTIMATE, it will ask you to:

"Enter total number of failures to similate (<=1000):"

The number you enter here is the number of failures to generate. This is not the number of systems in a specific major unit. These numbers are independent of each other since the number of systems depend on the major unit type that is chosen later in the process. The number to be entered has to be at most 1000 and will be printed. The greater the number you pick, the better the prediction will be, but also the longer the runtime of the program.

After this the program will tell you to:

"adjust paper in printer
press a key to continue".

Make sure your printer is connected and online and that your paper is at the top of a new page. Press any key to proceed. The next required steps are to enter the distributions and parameters of the transportation and administrative delay times for each of the three maintenance levels.

Figure B.9 shows the screen for the selection of levels.

```
Enter required information about the transportation times (one way):

     - on the user / org maint. level    ( 1 )
     - on the       DS / GS      level    ( 2 )
     - on the Depot / Industry  level    ( 3 )

     -          no more entries           ( 9 )
```

Figure B.9 Menu for Administrative Delay Times

You have to enter data for all levels of maintenance. Do not skip any input since these data will be used later in the program. You may, however, use the default values for each distribution. So, enter a "1" for the Org Mn level and the screen will present the available distributions.

Figure B.10 shows your options for the distributions. Pick the one you want or guess at one of them. If you do not want to have a random time pick the fixed value. Your delay time for the selected level will in this case not be generated by a random number generator but will be treated as a constant.

These are your options for the distributions:

```
            transportation times (one way)
Enter desired distribution for the time delays:
uniform            ( 1 )
normal             ( 2 )
exponential        ( 3 )
Gamma              ( 4 )
fixed              ( 5 )
```

Figure B.10 Distributions of Delay Times

Figure B.11 shows a screen prompting for the lower limit of a uniform distribution as an example for a screen to enter parameters for the distributions.

As soon as a distribution with its respective parameters has been entered, the program will print these. When the input is complete for all maintenance levels this information will also be available in an auxiliary data file "006".

This is a sample screen to enter parameters:

```
            transportation times (one way)
On the user / org maint. level, enter for the uniform distribution


            the lower limit in hours
            the default value is : 1



            Want to accept default value - press ENTER key?
```

Figure B.11 Enter Parameters for Distribution

If you entered wrong data, simply pick the level again at the menu shown in Figure B.9 and enter the correct numbers. When you have entered all the transportation delay time distributions for the three levels correctly, chose option 9 to get to the screens for the administrative delay times. These screens look the same as the previous ones. Enter the data for all three levels and close the input with a "9" at the level menu (see Figure B.9). This will conclude this phase of the program.

Next enter a few additionally required data. To be able to compute the time frame ESTIMATE will ask you to:

"Enter number of workhours per day".

Give here the average number of both the system and the maintenance workhours on a workday. Then

"Enter number of workdays per week".

Again enter the average number of days here. Now the program needs to know for which system the simulation run has to be performed. Therefor:

"Enter the System Stock Number (e.g. 1234-12-123-1234)"

You enter the national stock number for the overall system in the format given in the parentheses.

For the last part of the interaction with the user, ESTIMATE needs to know where to find the information about the system distribution within the military structure.

Figure B.12 shows the screen that asks for the name of this file.

```
System Dislocation within Major Units

Input filename for system distribution file
Default filename is DISTRIBN

To accept default press ENTER or enter name
```

Figure B.12 Name of System Distribution File

After this entry the program takes one major unit after the other and finds in the appropriate data file all unit types within the major unit and checks them if they contain the system in question. Finally, after all units are checked, the total number for that major unit is determined. For division level there will be two numbers computed, one for the divisional units and the other one for the total number including the subordinate brigades. The same is valid for the corps, in which the number for the corps units and the total including the subordinate divisions will be computed. Finally the program will find the number for the reserve. Major units, that do not contain any of the systems, will not be taken into account. Figure B.13 shows the menu for the major units.

```
===============================================
== Menu :  CHOICE OF MAJOR UNIT TYPE         ==
===============================================
==    Customer       # systems    selection  ==
==   ---------------------------------------  ==
==    MECH. BRIG      ( 90 )        :  1     ==
==    ARM'D BRIG      ( 90 )        :  2     ==
==    MOUNT.BRIG      ( 30 )        :  3     ==
==    MECH. DIV       ( 395 )       :  4     ==
==    DIV. UNITS      ( 125 )       :  5     ==
==    ARM'D. DIV      ( 395 )       :  6     ==
==    DIV. UNITS      ( 125 )       :  7     ==
==    CORPS           ( 1390 )      :  8     ==
==    CORPS UNITS     ( 205 )       :  9     ==
==    Reserve         ( 104 )       :  10    ==
===============================================

              Enter choice :
```

Figure B.13 Menu for Major Units

The number in parentheses stands for the number of systems in that particular major unit. Enter the number (1 to 10) that represents your choice.

This was the last input for ESTIMATE. Now the computation starts. ESTIMATE will display what it is just doing with messages such as:

"Compute Lambda for overall system"

"Subroutine MTBF Generation"

"xx th Iteration for Simulation
Subroutine Failure Generation"

"xx th Iteration for Simulation
Subroutine Update Data Files"

"Subroutine Summation of consumed Items",

where the xx stand for the numbers inserted by the program at execution time. The final screen with the following message is shown in Figure B.14:

"elapsed time for xxxx simulations : xxx.xx weeks

PROGRAM END"

Before this message appears on the screen, the printer will have printed the consumed items and manhours for each level of maintenance and the planning figures for a four week period. Also statistical data about the elapsed time and the systems' status plus the total cost of parts will be provided.

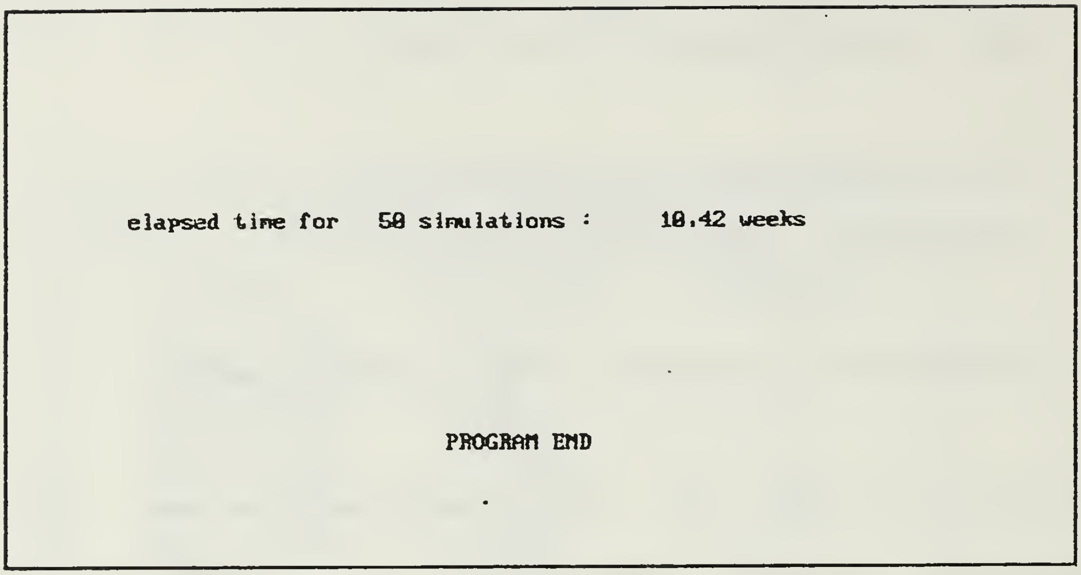


Figure B.14 End Screen Program ESTIMATE

5. Program QBEHAVE.EXE

a. Start QBEHAVE

QBEHAVE is the second of the two major programs in CAESAR to simulate and compute logistics requirements. The other one is ESTIMATE. QBEHAVE recomputes the data from ESTIMATE, the idealized run. But now it simulates the failing systems as customers in a real service environment. Customers have to form a queue and will be served as soon as a server is available.

The number of servers is determined by ESTIMATE. QBEHAVE takes this number and, if this option is chosen, computes for a given bandwidth, minimum one server, on either side the real behavior. It is assumed, that at least one server will be assigned to each level of maintenance.

A printer is required for QBEHAVE to avoid a DOS error message.

To start QBEHAVE type at the DOS prompt:

C>QBEHAVE.

b. Operate QBEHAVE

All required data, distributions and related information will be obtained from auxiliary data files, created by ESTIMATE and the data files containing major unit, system and organizational data. The first input for QBEHAVE is the choice of either the verification of the numbers obtained from ESTIMATE or a sensitivity analysis of the numbers (see Figure B.15). QBEHAVE computes its results for the same failures as ESTIMATE, to assure the comparability of the two sets of figures.

After this choice, the user has to determine the sensitivity bounds (see Figure B.16), the stockout risk factor (see Figure B.17) and the average shipping time for ordered items (see Figure B.18) to provide the basis for the restocking computations.

```
**************************
**   Program Mode Menu   **
**************************


( 0 ) ... Check Queuing Behavior without
          Sensitivity Analysis (faster)

( 1 ) ... Check Queuing Behavior with
          Sensitivity Analysis

============================================


          Default Value is 0.
  To accept press ENTER, else enter value  :
```

Figure B.15 Program Mode Menu

```
Enter sensitivity analysis bounds in per cent



          The default value is 10.



  To accept default press ENTER, else type value
```

Figure B.16 Choice of Sensitivity Bounds

Enter the stockout risk factor r

(0.0 < r < 1.0)

The default value is 0.1.

To accept default press ENTER, else type value

Figure B.17 Input of the Stockout Risk Factor

Enter ave. shipping time for ordered items in days

The default value is 5 days.

To accept default press ENTER, else type value

Figure B.18 Input of the Ave. Shipping Time

QBEHAVE will always display what it is just computing. That provides the user with the program status at each time.

When the program is started, the first thing to compute is the sequence of entry data for the Org Mn queue. This sequence determines all other times for the different maintenance levels.

Figure B.19 shows the message for this step.

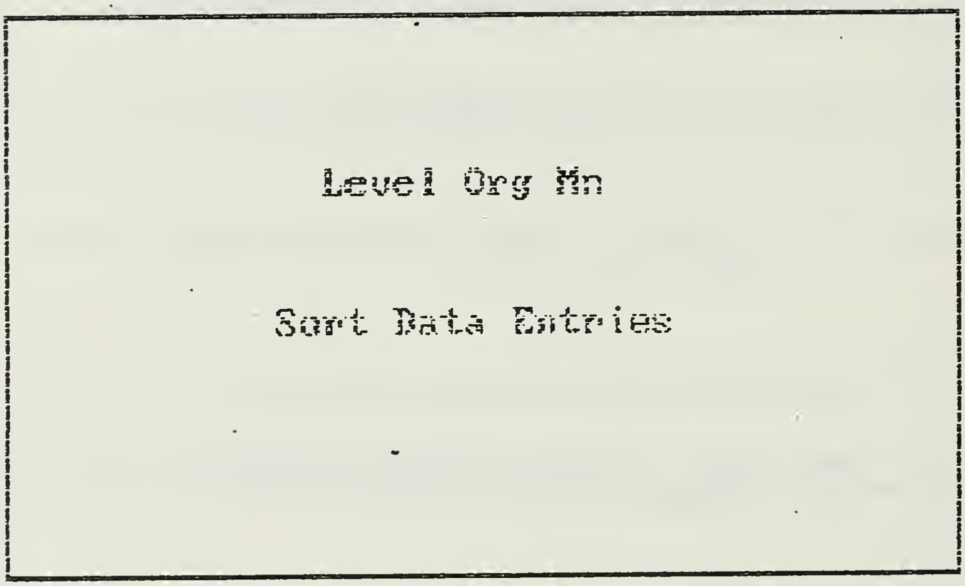


Figure B.19 QBEHAVE Sort Screen Org Mn

After these data have been sorted, the customers will be served in their sequence of entering level Org Mn. The time, the customers leave service will be computed and the time the systems are operational again. Furthermore, QBEHAVE keeps track of the part consumption and will, at the end of the program run, print the supply levels.

Figure B.20 displays the screen, that is shown during these computations.

```
number of servers for level Depot  :   1

number of servers for level DS/GS  :   1

number of servers for level Org Mn :   2

        Compute Queueing Behavior

              Level Org Mn

           Service Number   14
```

Figure B.20 Service on Org Mn Level

The times, the customers leave level Org Mn are the failure times for the next level, DS/GS. Transportational and administrative delay times will be added. The service enter times are computed and brought into sequence.

As before, the customers will be served and leave service. The spare parts consumption will be noted. The repaired items will go back into the supply chain as spares for the lower maintenance level.

The last step for the maintenance portion of the program are the computations for the Depot level. They are performed using the same algorithms as before. The times of failure for the customers at the Depot level are the times the customer leaves the lower level service. Transportational and administrative delay times are added.

After each sort routine, the program will display the service screen with the number of servers available on all levels and the customer being serviced. When all three levels are computed, the current combination of servers on each level is printed with:

- number of customers
- number of failures
- number of servers for each level
- number of delays for each level
- max. queue length for each level
- total waiting time for each level
- percentages servers are busy for each level
- total elapsed time and number of 4 week periods

Figure B.21 shows the screen for the sample run after all combinations of servers on all levels have been computed.

```
number of servers for level Depot  :   2

number of servers for level DS/GS  :   2

number of servers for level Org Mn :   2

        Compute Queueing Behavior

         Service Number      28
```

Figure B.21 Service on Depot Level

QBEHAVE will now analyze the spare part consumption. The times, when items were needed are known. ESTIMATE suggested the stock levels. QBEHAVE uses these suggestions. Refill of parts will be performed every four weeks with constant increments. All other repairable items will have the initial stock level. After repair they will be rerouted into the supply chain. There will be no attrition for these items.

Figure B.22 shows the screen, that is displayed during these computations.

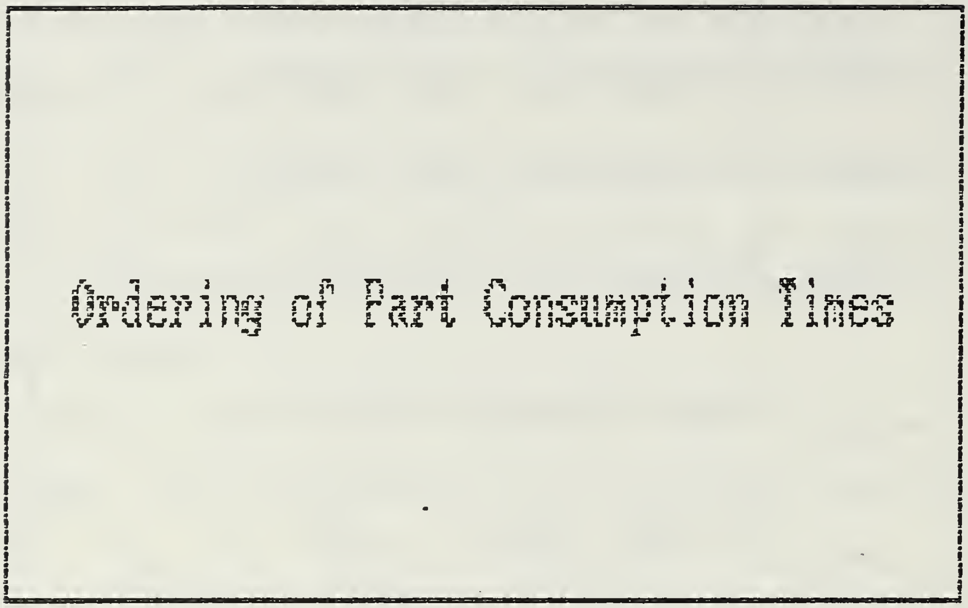


Figure B.22 Compute Spare Part Behavior

The spare part consumption will be protocolled on the printer including:

- national stock number
- initial stock level
- final stock level
- minimum stock level.

That concludes the program QBEHAVE. The output of QBEHAVE offers an array of alternatives for server numbers as well as the trade-offs between waiting times and the server numbers. From these the decision maker can chose his preferred option. The statistic of the part consumption can guide to decisions about the respective stock levels.

The total number of output pages can be computed as:

(total number of pages) =

= combinations of server numbers + 1

= (number of mutations level Org Mn) *

(number of mutations level DS/GS) *

(number of mutations level Depot) +

(one page for spare parts).

E.g. Org Mn level (5+/-1).servers: 4,5,6

DS/GS level (3+/-1) servers: 2,3,4

Depot level (1+/-1) servers: 1,2 (min 1!)

total number of pages :

3 * 3 * 2 + 1 = 19 pages

APPENDIX C

TACTICAL SIGNS

Appendix C summarizes the tactical signs, used in the programs of CAESAR. This is the standard notation for unit sizes:

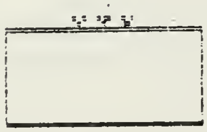
xxx	corps
xx	division
x	brigade
|||	regiment
||	battalion
|	company

The tactical signs are ordered with respect to unit size and the CAESAR code numbers. This order does not imply any other meaning.

The display format is as follows:

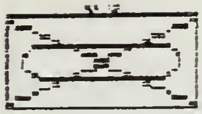
Symbol	CAESAR code number description

Here are the most commonly used tactical signs of both the US and GE Armies [Ref. 7 and 8] :

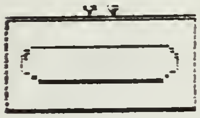


0
Corps

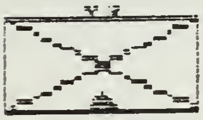
Figure C.1 Corps



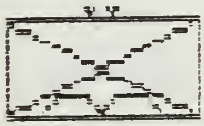
1
Mechanized Infantry Division



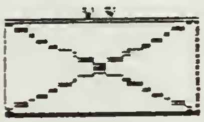
2
Armored Division



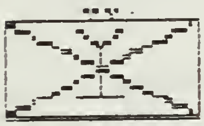
3
Mountaineer Division



4
Airborne Division



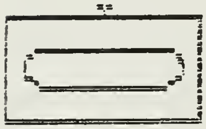
5
Infantry Division



6
Air Assault Division

Figure C.2 Division Types

11
Mechanized Infantry Brigade

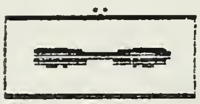


12
Armored Brigade

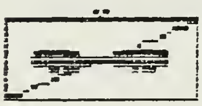
13
Mountaineer Brigade

14
Airborne Brigade

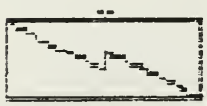
15
Infantry Brigade



16
Combat Aviation Brigade

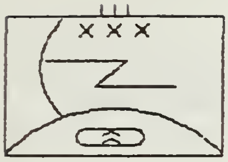


17
Air Cavalry Combat Brigade

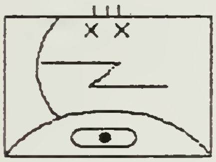


19
Signal Brigade

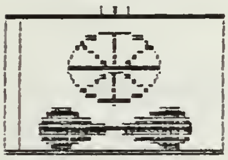
Figure C.3 Brigade Types



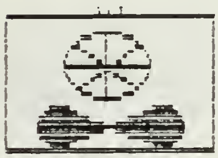
21
Anti Aircraft Defence Regiment
(Corps)



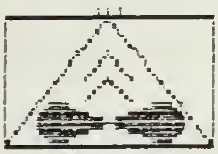
22
Anti Aircraft Defence Regiment
(Division)



23
Aviation Regiment

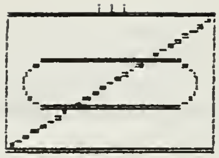


24
Aviation Regiment

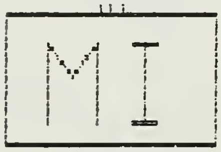


25
Combat Aviation Regiment

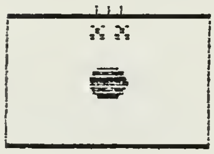
Figure C.4 Regiment Types I



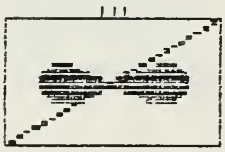
26
Armored Cavalry Regiment



27
Military Intelligence Regiment



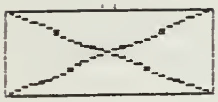
28
Field Artillery Regiment



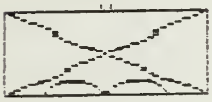
29
Cavalry Regiment Air Attack

30
Aviation Group

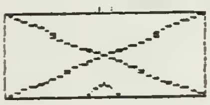
Figure C.5 Regiment Types II



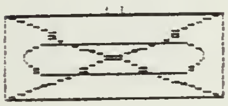
41
Motorized Infantry Battalion
(Division)



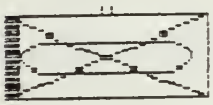
42
Airborne Battalion



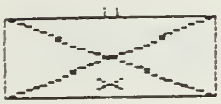
43
Mountaineer Battalion



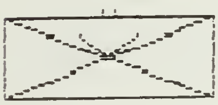
44
Mechanized Infantry Battalion
(Armored Brigade)



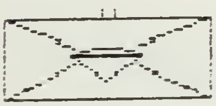
45
Mechanized Infantry Battalion
(Mech. Inf. Brigade)



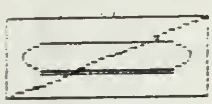
46
Motorized Infantry Battalion
(Division)



47
Air Assault Infantry Battalion

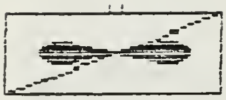


48
Airborne Battalion (GE)

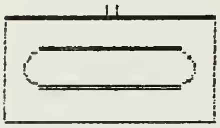


49
Armored Reconnaissance
Battalion

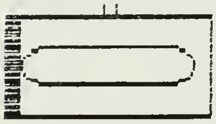
Figure C.6 Battalion Types I



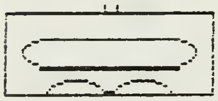
50
Air Cavalry Squadron



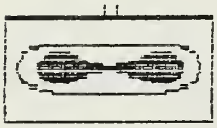
51
Armored Battalion
(Mech.Inf. Brig.)



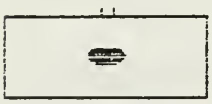
52
Armored Battalion
(Armored Brig.)



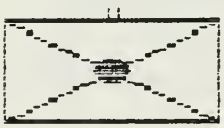
53
Armored Battalion
(Airborne Brig.)



54
Attack Helicopter Battalion

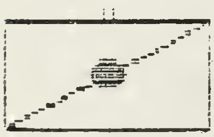


55
Field Artillery Battalion



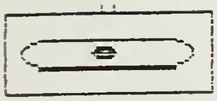
56
Field Artillery Battalion
(AASLT Division)

57
Field Artillery Battalion
(Airborne Division)



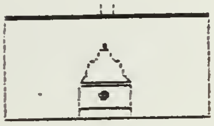
58
Reconnaissance Battalion

Figure C.7 Battalion Types II



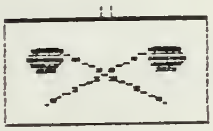
59
Field Artillery Battalion (GE)
(Brigade)

60
Missile Artillery Battalion



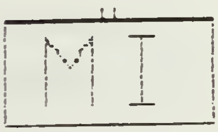
61
Missile Artillery Battalion
(GE) (Corps)

62
Field Artillery Battalion
(Mount. Brigade)



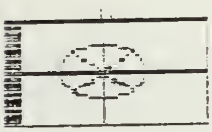
63
NBC Battalion

64
Combat Aviation Battalion



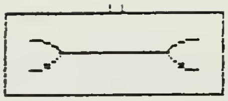
65
Military Intelligence Battalion

66
Medical Battalion

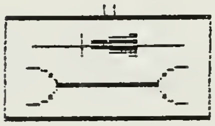


67
Medical Evacuation Battalion

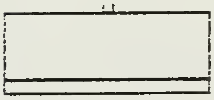
Figure C.8 Battalion Types III



68
Maintenance Battalion

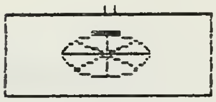


69
Maintenance Battalion (GE)
(Electronic Equipment)

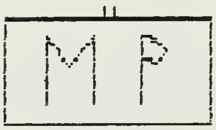


70
Supply Battalion

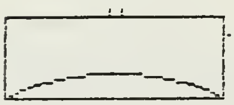
71
Supply Battalion
(Special Ordnance)



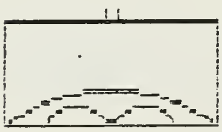
72
Transportation Battalion



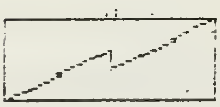
73
Military Police Battalion



74
Air Defence Artillery Battalion

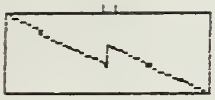


75
Air Defence Artillery Battalion
(Airborne Division)

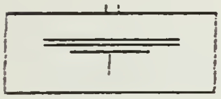


76
Electronic Warfare Battalion

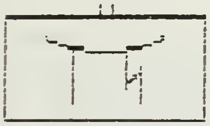
Figure C.9 Battalion Types IV



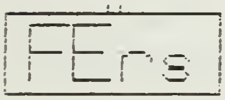
77
Signal Battalion



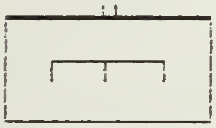
78
Signal Construction Battalion
(GE)



79
Signal Construction Battalion
(GE)

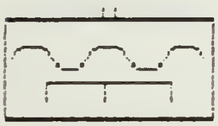


80
Feldersatz Bataillon (GE)

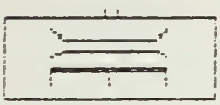


81
Engineer Battalion

82
Armored Engineer Battalion



83
Engineer Battalion

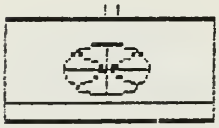


84
Engineer Battalion

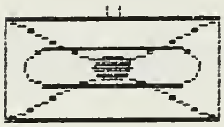
85
Support Battalion
(Airborne Div.)

Figure C.10 Battalion Types V

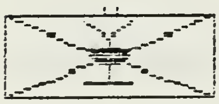
86
Support Battalion



87
Supply And Transportation Battalion



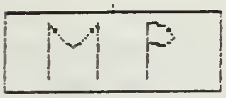
88
Field Artillery Battalion (Mech. Inf. Division)



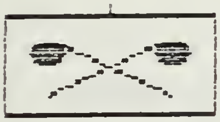
89
Field Artillery Battalion (AASLT Division)

Figure C.11 Battallion Type VI

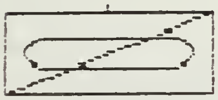
101
Military Band



102
Military Police Company



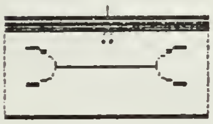
103
NBC Company



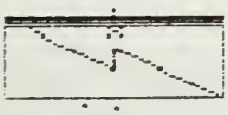
104
Cavalry Troop

105
Military Intelligence Company

106
HHQ Corps Command



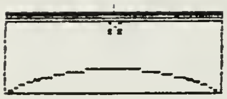
107
HHQ Corps Maintenance Command



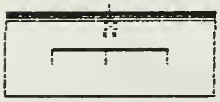
108
HHQ Corps Signal Command

109
HHQ Corps Artillery Command

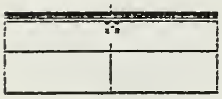
Figure C.12 Company Types I



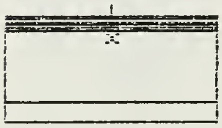
110
HHQ Corps ADA Command



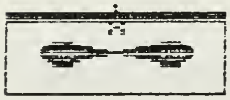
111
HHQ Corps Engineer Command



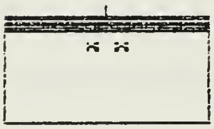
112
HHQ Corps Medical Command



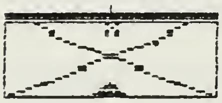
113
HHQ Corps Supply Command



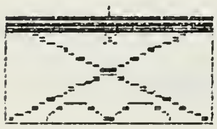
114
HHQ Aviation Brigade



115
HHQ Division Command



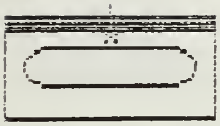
116
HHQ Mountaineer Brigade



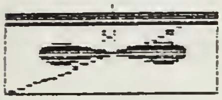
117
HHQ Airborne Brigade

118
HHQ Mechanized Infantry Brigade

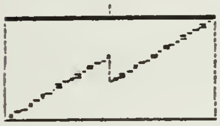
Figure C.13 Company Types II



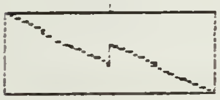
119
HHQ Armored Brigade



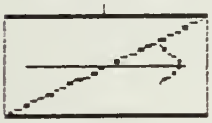
120
HHQ Air Cavalry Combat Brigade



121
Electronic Warfare Company



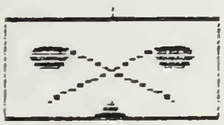
122
Signal Company



123
Fernspaeh-Kompanie (GE)

124
Armored Engineer Company

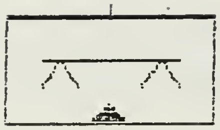
126
Engineer Company
(Mount. Brig.)



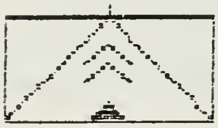
127
NBC Company
(Mount. Brig.)

128
Supply & Support Company
(Mount. Brig.)

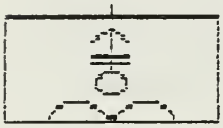
Figure C.14 Company Types III



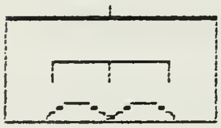
129
Tragtier Kompanie (GE)
(Mount. Brig.)



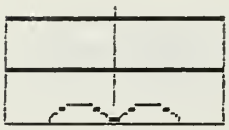
130
Antitank Company
(Mount. Brig.)



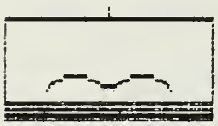
131
Mortar Company
(Airborne Brig.)



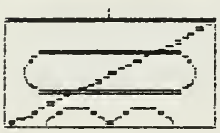
132
Engineer Company
(Airborne Brig.)



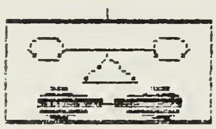
133
Medical Company
(Airborne Brig.)



134
Supply & Support Company
(Airborne Brigade)



135
Cavalry Troop
(Airborne Brig.)

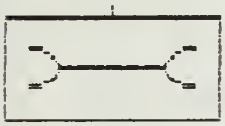


136
Aviation Liaison

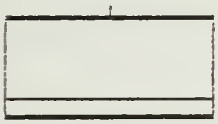
137
Aviation Liaison

Figure C.15 Company Types IV

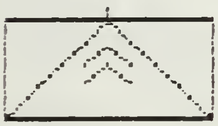
138
Field Airport



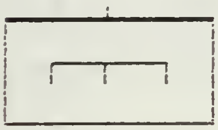
139
Maintenance Company



140
Supply Company



141
Antitank Company



142
Engineer Company

144
HHQ Infantry Brigade

Figure C.16 Company Types V

APPENDIX D

SOURCE CODE OF CAESAR

1. Source Code Program TACMIX

```
REM program TACMIX                          JUNE 1987
REM
REM This program provides the graphical interface for
REM the user to enter, view or modify the structure
REM of major units.
REM
REM ***********************************************
REM **               Variable Directory          **
REM ***********************************************
REM ** A        : auxiliary variable             **
REM ** B        : auxiliary variable             **
REM ** C        : counting variable              **
REM ** E        : auxiliary variable             **
REM ** E$       : number of units                **
REM ** F        : error flag                     **
REM ** H        : auxiliary variable             **
REM ** LL       : left margin                    **
REM ** MU       : major unit code number         **
REM ** MU$      : major unit code number         **
REM ** N        : auxiliary variable             **
REM ** N(x)     : number of units of a type      **
REM ** N$       : string variable for filename   **
REM ** P1       : pointer for graphic routine    **
REM ** PI       : mathematical constant          **
REM ** Q        : unit code number               **
REM ** ST       : auxiliary variable             **
REM ** U        : type of major unit             **
REM ** U(x)     : unit code number               **
REM ** U$       : auxiliary variable             **
REM ** U1$      : unit description               **
REM ** U2$      : additional information         **
REM ** W        : auxiliary variable             **
REM ** X        : auxiliary variable             **
REM ** Y        : auxiliary variable             **
REM ** ZF       : zoom factor for graph          **
REM ***********************************************
REM
KEY OFF:DIM U(40),N(40)
PI=3.141593
```

```
startscreen:
CLS:SCREEN 0
LOCATE 2,20:PRINT"***************************"
LOCATE 3,20:PRINT"***   Program TACMIX   ***"
LOCATE 4,20:PRINT"***      Options:      ***"
LOCATE 5,20:PRINT"***************************"
LOCATE 7,15:PRINT"- Create New Organisation .......(1)"
LOCATE 9,15:PRINT"- View Existing Organisation ....(2)"
LOCATE 11,15
PRINT"- Modify Existing Organisation ..(3)"
LOCATE 15,15
PRINT"- Terminate Program .............(9)"
LOCATE 24,20:INPUT"Enter Number of Choice : ",E
IF E<1 OR E>9 OR(E>3 AND E<9) THEN startscreen
CLS:SCREEN 2
ON E GOTO neworg,vieworg,modify,terminate,terminate,_
          terminate,terminate,terminate,terminate
GOTO startscreen
terminate:
SCREEN 0:CLS:END
REM ***************************************************
REM Branch Create a new Organization
REM
neworg:
P1=0:GOSUB pickunit:CLS:LL=60:X=1:Y=1
A=270:B=5:GOSUB tacsign
IF VAL(MU$)=0 THEN Q=1 ELSE IF VAL(MU$)<7_
                              THEN Q=11 ELSE Q=23
11:
FOR B=LL TO 161 STEP 60
  FOR A=0 TO 600 STEP 102
    GOSUB tacsign:Q=Q+1:IF Q>144 THEN F=1:_
          GOTO 13 ELSE 12
12:
  NEXT A
NEXT B
13:
LOCATE 23,1
PRINT"                                                    "
LOCATE 23,1
INPUT"enter unit code number  (ENTER for  next frame) :
",U$
IF U$=""  THEN 14 ELSE IF VAL(U$)<0 OR VAL(U$)>144 THEN
13
U(X)=VAL(U$)
LOCATE 23,1
PRINT"                                                    "
LOCATE 23,1
INPUT"enter number of these units (default=1) : ",U$
LOCATE 23,1
PRINT"                                                    "
```

```
U=VAL(U$):IF U>1 THEN N(X)=U ELSE N(X)=1
INCR X:LOCATE 23,1
INPUT"another unit from this frame (y/n) ";U$
IF U$="y" OR U$="Y" THEN 11
14:
IF Q<145 THEN CLS:LL=5:GOTO 11
DECR X:GOSUB sort
OPEN "o",1,N$
WRITE#1,VAL(MU$),X
FOR Y=1 TO X:WRITE#1,U(Y),N(Y):NEXT Y
CLOSE #1
GOTO startscreen
REM
******************************************************
REM Branch View Existing Organisation
REM
REM
vieworg:
P1=0:ZF=1/6:GOSUB pickunit:CLS
FOR Y=1 TO 40:U(Y)=0:N(Y)=0:NEXT Y
LOCATE 23,1
PRINT"                                                    "
OPEN "i",1,N$
INPUT#1,MU,C
FOR Y=1 TO C:INPUT#1,U(Y),N(Y):NEXT Y
CLOSE #1
A=270:B=5:Q=MU:W=1:P1=1:GOSUB tacsign:LL=65
15:
FOR B=LL TO 159 STEP 60
  FOR A=0 TO 600 STEP 125
    Q=U(W):E$=RIGHT$(STR$(N(W)),1)
    GOSUB tacsign:INCR W:IF W>C THEN 16
  NEXT A
NEXT B
16:
LOCATE 23,33:PRINT "Press any Key":E$=INPUT$(1)
CLS:IF U(W)>0 THEN LL=5:GOTO 15 ELSE GOTO startscreen
REM ******************************************************
REM Branch Modify an Organization
REM
modify:
P1=0:ZF=1/6:GOSUB pickunit:CLS
FOR Y=1 TO 40:U(Y)=0:N(Y)=0:NEXT Y
LOCATE 23,1
PRINT"                                                    "
```

```
18:
CLS:OPEN "i",1,N$
INPUT#1,MU,C
FOR Y=1 TO C:INPUT#1,U(Y),N(Y):NEXT Y
CLOSE #1
A=270:B=5:E$="1":Q=MU:W=1:P1=1:GOSUB tacsign:LL=65

19:
FOR B=LL TO 159 STEP 60
  FOR A=0 TO 600 STEP 125
     Q=U(W):E$=RIGHT$(STR$(N(W)),1):GOSUB tacsign
     INCR W:IF W>C THEN 110
  NEXT A
NEXT B

110:
LOCATE 23,33
PRINT "Any Modifications ?":E$=INPUT$(1)
IF E$="y" OR E$="Y" THEN 111 ELSE 119
LOCATE 23,5

111:
PRINT "Which Modification ? (0=erase,1=change number";
PRINT " of units,2=add a unit)":E$=INPUT$(1)
LOCATE 23,1
FOR Y=1 TO 70:PRINT " ";:NEXT Y:PRINT

112:
LOCATE 23,33:PRINT "Which Unit Number ";:INPUT U
E=VAL(E$)
IF E=0 THEN 113 ELSE IF E=1 THEN 116_
                        ELSE IF E=2 THEN 118 ELSE 112

113:
FOR Y=1 TO C:IF U(Y)=U THEN 114 ELSE NEXT Y:GOTO 18

114:
IF Y<C THEN_
   FOR X=Y TO C-1:U(X)=U(X+1):N(X)=N(X+1):NEXT X:_
       DECR C:GOTO 115
U(C)=0:N(C)=0:C=C-1

115:
KILL N$:OPEN "o",1,N$
WRITE#1,MU,C
FOR Y=1 TO C:WRITE#1,U(Y),N(Y):NEXT Y
CLOSE 1:GOTO 18
```

```
116:
LOCATE 23,5
FOR Y=1 TO 70:PRINT " ";:NEXT Y:PRINT
FOR Y=1 TO C
IF U(Y)=U THEN 117 ELSE NEXT Y
GOTO 18

117:
LOCATE 23,33
INPUT"Enter new number of units ",N(Y)
GOTO 115
118:
LOCATE 23,5
FOR Y=1 TO 70:PRINT " ";:NEXT Y:PRINT
LOCATE 23,33:INPUT"Enter number of new units ",N
C=C+1:U(C)=U:N(C)=N:X=C:GOSUB sort:GOTO 115

119:
CLS:IF U(W)>0 THEN LL=5:GOTO 19 ELSE GOTO startscreen
GOTO startscreen
SCREEN 0:CLS:END

REM **************************************************
REM
REM Pick Major Unit
REM
pickunit:
REM
Q=0:ST=102:ZF=1/5
FOR B=5 TO 161 STEP 60
  FOR A=0 TO 600 STEP ST
    GOSUB tacsign
    INCR Q
    IF Q>20 THEN 17
  NEXT A
NEXT B

17:
LOCATE 24,1
INPUT"enter major unit code number : ",MU$
Q=VAL(MU$)
IF Q=0 THEN N$="corps":RETURN
IF Q<=6 THEN N$="div"+MU$:RETURN
IF Q<=19 THEN N$="brig"+MU$:RETURN
PRINT "input error":GOTO 17
RETURN
REM
REM **************************************************
```

```
REM Subroutine Sort of numerical Variables
REM input arrays     : u(x),n(x)
REM input parameter : x
REM
sort:
REM
FOR A=1 TO X
  FOR B=1 TO X-1
    IF U(B)=U(B+1) AND U(B)<1000 THEN_
       FOR C=B TO X-1:U(C)=U(C+1):N(C)=N(C+1):NEXT C:_
       U(X)=9999:DECR X
    IF U(B)>U(B+1) THEN H=U(B+1):U(B+1)=U(B):_
       U(B)=H:H=N(B+1):N(B+1)=N(B):N(B)=H
  NEXT B
NEXT A
RETURN
REM ************************************************
```

```
REM ***  Subroutine to draw one Tactical Sign  ******
REM
REM necc. input: zoom factor           0 < ZF <= 1
REM ------------ horizl. start coord. 0 < A < 550
REM              vertcl. start coord. 0 < B < 150
REM              value for constant   PI=3.141593
REM              value for unit type  0 <= Q <= 144
REM
tacsign:
REM
IF Q>144 THEN RETURN
IF Q>121 THEN companies2
IF Q>100 THEN companies1
IF Q>40 THEN battalions
IF Q>20 THEN regiments
IF Q>10 THEN brigades
IF Q=0 THEN GOSUB 0:GOTO box
IF Q=1 THEN GOSUB 1:GOTO box
IF Q=2 THEN GOSUB 2:GOTO box
IF Q=3 THEN GOSUB 3:GOTO box
IF Q=4 THEN GOSUB 4:GOTO box
IF Q=5 THEN GOSUB 5:GOTO box
IF Q=6 THEN GOSUB 6:GOTO box
IF Q=7 THEN Q=Q+1
IF Q=8 THEN Q=Q+1
IF Q=9 THEN Q=Q+1
IF Q=10 THEN RETURN
brigades:
IF Q=11 THEN GOSUB 11:GOTO box
IF Q=12 THEN GOSUB 12:GOTO box
IF Q=13 THEN GOSUB 13:GOTO box
IF Q=14 THEN GOSUB 14:GOTO box
IF Q=15 THEN GOSUB 15:GOTO box
IF Q=16 THEN GOSUB 16:GOTO box
IF Q=17 THEN GOSUB 17:GOTO box
IF Q=18 THEN Q=Q+1
IF Q=19 THEN GOSUB 19:GOTO box
IF Q=20 THEN RETURN
regiments:
IF Q=21 THEN GOSUB 21:GOTO box
IF Q=22 THEN GOSUB 22:GOTO box
IF Q=23 THEN GOSUB 23:GOTO box
IF Q=24 THEN GOSUB 24:GOTO box
IF Q=25 THEN GOSUB 25:GOTO box
IF Q=26 THEN GOSUB 26:GOTO box
IF Q=27 THEN GOSUB 27:GOTO box
IF Q=28 THEN GOSUB 28:GOTO box
IF Q=29 THEN GOSUB 29:GOTO box
IF Q=30 THEN GOSUB 30:GOTO box
IF Q=31 THEN Q=Q+1
IF Q=32 THEN Q=Q+1
```

```
IF Q=33 THEN Q=Q+1
IF Q=34 THEN Q=Q+1
IF Q=35 THEN Q=Q+1
IF Q=36 THEN Q=Q+1
IF Q=37 THEN Q=Q+1
IF Q=38 THEN Q=Q+1
IF Q=39 THEN Q=Q+1
IF Q=40 THEN RETURN
battalions:
IF Q=41 THEN GOSUB 41:GOTO box
IF Q=42 THEN GOSUB 42:GOTO box
IF Q=43 THEN GOSUB 43:GOTO box
IF Q=44 THEN GOSUB 44:GOTO box
IF Q=45 THEN GOSUB 45:GOTO box
IF Q=46 THEN GOSUB 46:GOTO box
IF Q=47 THEN GOSUB 47:GOTO box
IF Q=48 THEN GOSUB 48:GOTO box
IF Q=49 THEN GOSUB 49:GOTO box
IF Q=50 THEN GOSUB 50:GOTO box
IF Q=51 THEN GOSUB 51:GOTO box
IF Q=52 THEN GOSUB 52:GOTO box
IF Q=53 THEN GOSUB 53:GOTO box
IF Q=54 THEN GOSUB 54:GOTO box
IF Q=55 THEN GOSUB 55:GOTO box
IF Q=56 THEN GOSUB 56:GOTO box
IF Q=57 THEN GOSUB 57:GOTO box
IF Q=58 THEN GOSUB 58:GOTO box
IF Q=59 THEN GOSUB 59:GOTO box
IF Q=60 THEN GOSUB 60:GOTO box
IF Q=61 THEN GOSUB 61:GOTO box
IF Q=62 THEN GOSUB 62:GOTO box
IF Q=63 THEN GOSUB 63:GOTO box
IF Q=64 THEN GOSUB 64:GOTO box
IF Q=65 THEN GOSUB 65:GOTO box
IF Q=66 THEN GOSUB 66:GOTO box
IF Q=67 THEN GOSUB 67:GOTO box
IF Q=68 THEN GOSUB 68:GOTO box
IF Q=69 THEN GOSUB 69:GOTO box
IF Q=70 THEN GOSUB 70:GOTO box
IF Q=71 THEN GOSUB 71:GOTO box
IF Q=72 THEN GOSUB 72:GOTO box
IF Q=73 THEN GOSUB 73:GOTO box
IF Q=74 THEN GOSUB 74:GOTO box
IF Q=75 THEN GOSUB 75:GOTO box
IF Q=76 THEN GOSUB 76:GOTO box
IF Q=77 THEN GOSUB 77:GOTO box
IF Q=78 THEN GOSUB 78:GOTO box
IF Q=79 THEN GOSUB 79:GOTO box
IF Q=80 THEN GOSUB 80:GOTO box
IF Q=81 THEN GOSUB 81:GOTO box
IF Q=82 THEN GOSUB 82:GOTO box
```

```
IF Q=83 THEN GOSUB 83:GOTO box
IF Q=84 THEN GOSUB 84:GOTO box
IF Q=85 THEN GOSUB 85:GOTO box
IF Q=86 THEN GOSUB 86:GOTO box
IF Q=87 THEN GOSUB 87:GOTO box
IF Q=88 THEN GOSUB 88:GOTO box
IF Q=89 THEN GOSUB 89:GOTO box
IF Q=90 THEN Q=Q+1
IF Q=91 THEN Q=Q+1
IF Q=92 THEN Q=Q+1
IF Q=93 THEN Q=Q+1
IF Q=94 THEN Q=Q+1
IF Q=95 THEN Q=Q+1
IF Q=96 THEN Q=Q+1
IF Q=97 THEN Q=Q+1
IF Q=98 THEN Q=Q+1
IF Q=99 THEN Q=Q+1
IF Q=100 THEN RETURN
companies1:
IF Q=101 THEN GOSUB 101:GOTO box
IF Q=102 THEN GOSUB 102:GOTO box
IF Q=103 THEN GOSUB 103:GOTO box
IF Q=104 THEN GOSUB 104:GOTO box
IF Q=105 THEN GOSUB 105:GOTO box
IF Q=106 THEN GOSUB 106:GOTO box
IF Q=107 THEN GOSUB 107:GOTO box
IF Q=108 THEN GOSUB 108:GOTO box
IF Q=109 THEN GOSUB 109:GOTO box
IF Q=110 THEN GOSUB 110:GOTO box
IF Q=111 THEN GOSUB 111:GOTO box
IF Q=112 THEN GOSUB 112:GOTO box
IF Q=113 THEN GOSUB 113:GOTO box
IF Q=114 THEN GOSUB 114:GOTO box
IF Q=115 THEN GOSUB 115:GOTO box
IF Q=116 THEN GOSUB 116:GOTO box
IF Q=117 THEN GOSUB 117:GOTO box
IF Q=118 THEN GOSUB 118:GOTO box
IF Q=119 THEN GOSUB 119:GOTO box
IF Q=120 THEN GOSUB 120:GOTO box
IF Q=121 THEN GOSUB 121:GOTO box
companies2:
IF Q=122 THEN GOSUB 122:GOTO box
IF Q=123 THEN GOSUB 123:GOTO box
IF Q=124 THEN GOSUB 124:GOTO box
IF Q=125 THEN GOSUB 125:GOTO box
IF Q=126 THEN GOSUB 126:GOTO box
IF Q=127 THEN GOSUB 127:GOTO box
IF Q=128 THEN GOSUB 128:GOTO box
IF Q=129 THEN GOSUB 129:GOTO box
IF Q=130 THEN GOSUB 130:GOTO box
IF Q=131 THEN GOSUB 131:GOTO box
```

```
IF Q=132 THEN GOSUB 132:GOTO box
IF Q=133 THEN GOSUB 133:GOTO box
IF Q=134 THEN GOSUB 134:GOTO box
IF Q=135 THEN GOSUB 135:GOTO box
IF Q=136 THEN GOSUB 136:GOTO box
IF Q=137 THEN GOSUB 137:GOTO box
IF Q=138 THEN GOSUB 138:GOTO box
IF Q=139 THEN GOSUB 139:GOTO box
IF Q=140 THEN GOSUB 140:GOTO box
IF Q=141 THEN GOSUB 141:GOTO box
IF Q=142 THEN GOSUB 142:GOTO box
IF Q=143 THEN GOSUB 143:GOTO box
IF Q=144 THEN GOSUB 144:GOTO box
RETURN
$segment
REM *****************************************************
box:
LINE (10*ZF+A,10*ZF+B)-(360*ZF+A,120*ZF+B), ,B
IF U=1 THEN LINE(155*ZF+A,1*ZF+B)-(175*ZF+A,8*ZF+B):_
            LINE(175*ZF+A,1*ZF+B)-(155*ZF+A,8*ZF+B):_
            LINE(195*ZF+A,1*ZF+B)-(215*ZF+A,8*ZF+B):_
            LINE(215*ZF+A,1*ZF+B)-(195*ZF+A,8*ZF+B)
IF U=2 THEN LINE(175*ZF+A,1*ZF+B)-(195*ZF+A,8*ZF+B):_
            LINE(195*ZF+A,1*ZF+B)-(175*ZF+A,8*ZF+B)
IF U=3 THEN LINE(165*ZF+A,1*ZF+B)-(165*ZF+A,7*ZF+B):_
            LINE(185*ZF+A,1*ZF+B)-(185*ZF+A,7*ZF+B):_
            LINE (205*ZF+A,1*ZF+B)-(205*ZF+A,7*ZF+B)
IF U=4 THEN LINE(175*ZF+A,1*ZF+B)-(175*ZF+A,7*ZF+B):_
            LINE (195*ZF+A,1*ZF+B)-(195*ZF+A,7*ZF+B)
IF U=5 THEN LINE(185*ZF+A,1*ZF+B)-(185*ZF+A,7*ZF+B)
IF P1=1 THEN LOCATE B/8+17*ZF,A/8+54*ZF
IF E$="1" THEN 120 ELSE PRINT E$
120:
LOCATE B/8+24*ZF,A/8+18*ZF:PRINT Q:RETURN
REM
*******************************************************
u4:
LINE(10*ZF+A,10*ZF+B)-(360*ZF+A,120*ZF+B)
LINE(10*ZF+A,120*ZF+B)-(360*ZF+A,10*ZF+B)
RETURN
REM
*******************************************************
```

```
0:
REM corps #0
U=0:U1$="CORPS":U2$=""
LINE(130*ZF+A,ZF+B)-(150*ZF+A,7*ZF+B)
LINE(150*ZF+A,ZF+B)-(130*ZF+A,7*ZF+B)
LINE(175*ZF+A,ZF+B)-(195*ZF+A,7*ZF+B)
LINE(195*ZF+A,ZF+B)-(175*ZF+A,7*ZF+B)
LINE(220*ZF+A,ZF+B)-(240*ZF+A,7*ZF+B)
LINE(220*ZF+A,7*ZF+B)-(240*ZF+A,ZF+B)
LINE (10*ZF+A,10*ZF+B)-(360*ZF+A,120*ZF+B), ,B
RETURN

1:
REM mechanized infantery division #1
U=1:U1$="MECHANIZED INFANTRY DIVISION":U2$=""
CIRCLE (90*ZF+A,65*ZF+B),48*ZF,1,90*PI/180,270*PI/180
CIRCLE(280*ZF+A,65*ZF+B),48*ZF,1,270*PI/180,90*PI/180
LINE (90*ZF+A,45*ZF+B)-(280*ZF+A,45*ZF+B)
LINE (90*ZF+A,85*ZF+B)-(280*ZF+A,85*ZF+B)
GOSUB u4
RETURN

2:
REM armored  division  #2
U=1:U1$="ARMORED DIVISION":U2$=""
CIRCLE (90*ZF+A,65*ZF+B),48*ZF,1,90*PI/180,270*PI/180
CIRCLE(280*ZF+A,65*ZF+B),48*ZF,1,270*PI/180,90*PI/180
LINE (90*ZF+A,45*ZF+B)-(280*ZF+A,45*ZF+B)
LINE (90*ZF+A,85*ZF+B)-(280*ZF+A,85*ZF+B)
LINE (10*ZF+A,10*ZF+B)-(360*ZF+A,120*ZF+B), ,B
RETURN

3:
REM mountaineer division #3
U1$="MOUNTAINEER DIVISION":U2$=""
GOSUB u4
LINE (10*ZF+A,10*ZF+B)-(360*ZF+A,120*ZF+B), ,B
LINE(155*ZF+A,120*ZF+B)-(185*ZF+A,100*ZF+B)
LINE(215*ZF+A,120*ZF+B)-(185*ZF+A,100*ZF+B)
PAINT(185*ZF+A,114*ZF+B)
RETURN

4:
REM airborne  division #4
U=1:U1$="AIRBORNE DIVISION":U2$=""
CIRCLE(130*ZF+A,120*ZF+B),50*ZF,1,15*PI/180,165*PI/180
CIRCLE(240*ZF+A,120*ZF+B),50*ZF,1,15*PI/180,165*PI/180
LINE(178*ZF+A,115*ZF+B)-(192*ZF+A,115*ZF+B)
GOSUB u4
RETURN
```

```
5:
REM infantry division #5
U=1:U1$="INFANTRY DIVISION":U2$=""
GOSUB u4
RETURN

6:
REM air assault division #6
U=1:U1$="AIR ASSAULT DIVISION":U2$=""
LINE(145*ZF+A,20*ZF+B)-(185*ZF+A,40*ZF+B)
LINE(225*ZF+A,20*ZF+B)-(185*ZF+A,40*ZF+B)
LINE(185*ZF+A,40*ZF+B)-(185*ZF+A,100*ZF+B)
LINE(145*ZF+A,100*ZF+B)-(225*ZF+A,100*ZF+B)
RETURN

11:
REM mechanized infantery brigade #11
U=2:U1$="MECHANIZED INFANTRY BRIGADE":U2$=""
CIRCLE (90*ZF+A,65*ZF+B),48*ZF,1,90*PI/180,270*PI/180
CIRCLE(280*ZF+A,65*ZF+B),48*ZF,1,270*PI/180,90*PI/180
LINE (90*ZF+A,45*ZF+B)-(280*ZF+A,45*ZF+B)
LINE (90*ZF+A,85*ZF+B)-(280*ZF+A,85*ZF+B)
GOSUB u4
RETURN

12:
REM armored  brigade   #12
U=2:U1$="ARMORED BRIGADE":U2$=""
CIRCLE (90*ZF+A,65*ZF+B),48*ZF,1,90*PI/180,270*PI/180
CIRCLE(280*ZF+A,65*ZF+B),48*ZF,1,270*PI/180,90*PI/180
LINE (90*ZF+A,45*ZF+B)-(280*ZF+A,45*ZF+B)
LINE (90*ZF+A,85*ZF+B)-(280*ZF+A,85*ZF+B)
RETURN

13:
REM mountaineer brigade #13
U=2:U1$="MOUNTAINEER BRIGADE":U2$=""
GOSUB u4
LINE (10*ZF+A,10*ZF+B)-(360*ZF+A,120*ZF+B), ,B
LINE(152*ZF+A,120*ZF+B)-(182*ZF+A,100*ZF+B)
LINE(212*ZF+A,120*ZF+B)-(182*ZF+A,100*ZF+B)
PAINT(165*ZF+A,114*ZF+B)
RETURN

14:
REM airborne brigade #14
U=2:U1$="AIRBORNE BRIGADE":U2$=""
CIRCLE(130*ZF+A,120*ZF+B),50*ZF,1,15*PI/180,165*PI/180
CIRCLE(240*ZF+A,120*ZF+B),50*ZF,1,15*PI/180,165*PI/180
LINE(178*ZF+A,115*ZF+B)-(192*ZF+A,115*ZF+B)
GOSUB u4
RETURN
```

```
15:
REM infantry brigade #15
U=2:U1$="INFANTRY BRIGADE":U2$=""
GOSUB u4
RETURN

16:
REM combat aviation brigade #16
U=2:U1$="COMBAT AVIATION BRIGADE":U2$=""
CIRCLE(100*ZF+A,65*ZF+B),35*ZF,1,56*PI/180,304*PI/180
CIRCLE(270*ZF+A,65*ZF+B),35*ZF,1,236*PI/180,124*PI/180
LINE(120*ZF+A,52*ZF+B)-(250*ZF+A,78*ZF+B)
LINE(120*ZF+A,78*ZF+B)-(250*ZF+A,52*ZF+B)
PAINT(100*ZF+A,65*ZF+B),1:PAINT(270*ZF+A,65*ZF+B),1
RETURN

17:
REM air cav cbt brig #17
U=2:U1$="AIR CAVALRY COMBAT BRIGADE":U2$=""
CIRCLE(100*ZF+A,65*ZF+B),35*ZF,1,56*PI/180,304*PI/180
CIRCLE(270*ZF+A,65*ZF+B),35*ZF,1,236*PI/180,124*PI/180
LINE(120*ZF+A,52*ZF+B)-(250*ZF+A,78*ZF+B)
LINE(120*ZF+A,78*ZF+B)-(250*ZF+A,52*ZF+B)
LINE(10*ZF+A,120*ZF+B)-(360*ZF+A,10*ZF+B)
PAINT(100*ZF+A,65*ZF+B),1:PAINT(270*ZF+A,65*ZF+B),1
RETURN

19:
REM  signal brigade #19
U=2:U1$="SIGNAL BRIGADE":U2$=""
LINE (10*ZF+A,10*ZF+B)-(185*ZF+A,80*ZF+B)
LINE(185*ZF+A,50*ZF+B)-(360*ZF+A,120*ZF+B)
LINE(185*ZF+A,80*ZF+B)-(185*ZF+A,50*ZF+B)
RETURN
```

```
21:
REM anti aircraft regiment (corps) #21
U1$="ANTI AIRCRAFT DEFENCE REGIMENT":U2$=""
U=3:CIRCLE
(207*ZF+A,50*ZF+B),140*ZF,1,138*PI/180,223*PI/180
CIRCLE
(186*ZF+A,175*ZF+B),220*ZF,1,39*PI/180,142*PI/180
CIRCLE (140*ZF+A,105*ZF+B),14*ZF,1,90*PI/180,270*PI/180
CIRCLE (220*ZF+A,105*ZF+B),14*ZF,1,270*PI/180,90*PI/180
LINE(140*ZF+A,98*ZF+B)-(220*ZF+A,98*ZF+B)
LINE (140*ZF+A,112*ZF+B)-(220*ZF+A,112*ZF+B)
LINE (76*ZF+A,45*ZF+B)-(206*ZF+A,45*ZF+B)
LINE (156*ZF+A,65*ZF+B)-(286*ZF+A,65*ZF+B)
LINE (206*ZF+A,45*ZF+B)-(156*ZF+A,65*ZF+B)
LINE (182*ZF+A,101*ZF+B)-(168*ZF+A,104*ZF+B)
LINE (182*ZF+A,101*ZF+B)-(196*ZF+A,104*ZF+B)
LINE (182*ZF+A,106*ZF+B)-(168*ZF+A,109*ZF+B)
LINE (182*ZF+A,106*ZF+B)-(196*ZF+A,109*ZF+B)
LINE(130*ZF+A,15*ZF+B)-(150*ZF+A,25*ZF+B)
LINE(150*ZF+A,15*ZF+B)-(130*ZF+A,25*ZF+B)
LINE(175*ZF+A,15*ZF+B)-(195*ZF+A,25*ZF+B)
LINE(195*ZF+A,15*ZF+B)-(175*ZF+A,25*ZF+B)
LINE(220*ZF+A,15*ZF+B)-(240*ZF+A,25*ZF+B)
LINE(220*ZF+A,25*ZF+B)-(240*ZF+A,15*ZF+B)
RETURN

22:
REM anti aircraft regiment (divisional) #22
U1$="ANTI AIRCRAFT DEFENCE REGIMENT":U2$=""
U=3:CIRCLE
(207*ZF+A,50*ZF+B),140*ZF,1,137*PI/180,223*PI/180
CIRCLE
(186*ZF+A,175*ZF+B),220*ZF,1,39*PI/180,142*PI/180
CIRCLE (140*ZF+A,105*ZF+B),14*ZF,1,90*PI/180,270*PI/180
CIRCLE (220*ZF+A,105*ZF+B),14*ZF,1,270*PI/180,90*PI/180
LINE(140*ZF+A,98*ZF+B)-(220*ZF+A,98*ZF+B)
LINE (140*ZF+A,112*ZF+B)-(220*ZF+A,112*ZF+B)
LINE (76*ZF+A,45*ZF+B)-(206*ZF+A,45*ZF+B)
LINE (156*ZF+A,65*ZF+B)-(286*ZF+A,65*ZF+B)
LINE (206*ZF+A,45*ZF+B)-(156*ZF+A,65*ZF+B)
CIRCLE(182*ZF+A,105*ZF+B),10*ZF,3
PAINT(186*ZF+A,105*ZF+B),1,3
LINE(150*ZF+A,15*ZF+B)-(170*ZF+A,25*ZF+B)
LINE(170*ZF+A,15*ZF+B)-(150*ZF+A,25*ZF+B)
LINE(200*ZF+A,15*ZF+B)-(220*ZF+A,25*ZF+B)
LINE(200*ZF+A,25*ZF+B)-(220*ZF+A,15*ZF+B)
RETURN
```

```
23:
REM aviation regiment # 23
U=3:U1$="AVIATION REGIMENT":U2$=""
CIRCLE(100*ZF+A,100*ZF+B),35*ZF,1,56*PI/180,304*PI/180
CIRCLE(270*ZF+A,100*ZF+B),35*ZF,1,236*PI/180,124*PI/180
LINE(120*ZF+A,87*ZF+B)-(250*ZF+A,113*ZF+B)
LINE(120*ZF+A,113*ZF+B)-(250*ZF+A,87*ZF+B)
PAINT(100*ZF+A,100*ZF+B),1
PAINT(270*ZF+A,100*ZF+B),1
CIRCLE(185*ZF+A,50*ZF+B),70*ZF
LINE (135*ZF+A,31*ZF+B)-(237*ZF+A,69*ZF+B)
LINE(136*ZF+A,69*ZF+B)-(235*ZF+A,31*ZF+B)
LINE (185*ZF+A,22*ZF+B)-(185*ZF+A,78*ZF+B)
LINE (115*ZF+A,50*ZF+B)-(255*ZF+A,50*ZF+B)
LINE (30*ZF+A,10*ZF+B)-(30*ZF+A,120*ZF+B)
RETURN

24:
REM aviation regiment #24
U=3:U1$="AVIATION REGIMENT":U2$=""
CIRCLE(100*ZF+A,100*ZF+B),35*ZF,1,56*PI/180,304*PI/180
CIRCLE(270*ZF+A,100*ZF+B),35*ZF,1,236*PI/180,124*PI/180
LINE(120*ZF+A,87*ZF+B)-(250*ZF+A,113*ZF+B)
LINE(120*ZF+A,113*ZF+B)-(250*ZF+A,87*ZF+B)
PAINT(100*ZF+A,100*ZF+B),1:PAINT(270*ZF+A,100*ZF+B),1
CIRCLE(180*ZF+A,50*ZF+B),70*ZF
LINE (130*ZF+A,31*ZF+B)-(232*ZF+A,69*ZF+B)
LINE(131*ZF+A,69*ZF+B)-(230*ZF+A,31*ZF+B)
LINE (180*ZF+A,22*ZF+B)-(180*ZF+A,78*ZF+B)
LINE (110*ZF+A,50*ZF+B)-(250*ZF+A,50*ZF+B)
RETURN

25:
REM combat aviation regiment #25
U=3:U1$="COMBAT AVIATION REGIMENT":U2$=""
CIRCLE(100*ZF+A,100*ZF+B),35*ZF,1,56*PI/180,304*PI/180
CIRCLE(270*ZF+A,100*ZF+B),35*ZF,1,236*PI/180,124*PI/180
LINE(120*ZF+A,87*ZF+B)-(250*ZF+A,113*ZF+B)
LINE(120*ZF+A,113*ZF+B)-(250*ZF+A,87*ZF+B)
LINE(10*ZF+A,120*ZF+B)-(180*ZF+A,10*ZF+B)
LINE(360*ZF+A,120*ZF+B)-(180*ZF+A,10*ZF+B)
PAINT(100*ZF+A,100*ZF+B),1:PAINT(270*ZF+A,100*ZF+B),1
LINE(145*ZF+A,60*ZF+B)-(185*ZF+A,40*ZF+B)
LINE(185*ZF+A,40*ZF+B)-(225*ZF+A,60*ZF+B)
LINE(145*ZF+A,80*ZF+B)-(185*ZF+A,60*ZF+B)
LINE(185*ZF+A,60*ZF+B)-(225*ZF+A,80*ZF+B)
RETURN
```

```
26:
REM armored cavalry regiment #26
U=3:U1$="ARMORED CAVALRY REGIMENT":U2$=""
CIRCLE (90*ZF+A,65*ZF+B),48*ZF,1,90*PI/180,270*PI/180
CIRCLE(280*ZF+A,65*ZF+B),48*ZF,1,270*PI/180,90*PI/180
LINE(90*ZF+A,45*ZF+B)-(280*ZF+A,45*ZF+B)
LINE(90*ZF+A,85*ZF+B)-(280*ZF+A,85*ZF+B)
LINE(10*ZF+A,120*ZF+B)-(360*ZF+A,10*ZF+B)
RETURN

27:
REM military intelligence regiment #27
U=3:U1$="MILITARY INTELLIGENCE REGIMENT":U2$=""
LINE(220*ZF+A,30*ZF+B)-(260*ZF+A,30*ZF+B)
LINE(220*ZF+A,100*ZF+B)-(260*ZF+A,100*ZF+B)
LINE(80*ZF+A,30*ZF+B)-(80*ZF+A,100*ZF+B)
LINE(160*ZF+A,30*ZF+B)-(160*ZF+A,100*ZF+B)
LINE(240*ZF+A,30*ZF+B)-(240*ZF+A,100*ZF+B)
LINE (80*ZF+A,30*ZF+B)-(120*ZF+A,60*ZF+B)
LINE(120*ZF+A,60*ZF+B)-(160*ZF+A,30*ZF+B)
RETURN

28:
REM field artillery regiment #28
U=3:U1$="FIELD ARTY REGIMENT":U2$=""
CIRCLE (185*ZF+A,65*ZF+B),30*ZF,3
PAINT(185*ZF+A,65*ZF+B),1,3
LINE(155*ZF+A,16*ZF+B)-(175*ZF+A,22*ZF+B)
LINE(175*ZF+A,16*ZF+B)-(155*ZF+A,22*ZF+B)
LINE(195*ZF+A,16*ZF+B)-(215*ZF+A,22*ZF+B)
LINE(215*ZF+A,16*ZF+B)-(195*ZF+A,22*ZF+B)
RETURN

29:
REM cav reg air atk #29
U=3:U1$="CAVALRY REGIMENT AIR ATTACK":U2$=""
CIRCLE(100*ZF+A,65*ZF+B),35*ZF,1,56*PI/180,304*PI/180
CIRCLE(270*ZF+A,65*ZF+B),35*ZF,1,236*PI/180,124*PI/180
LINE(120*ZF+A,52*ZF+B)-(250*ZF+A,78*ZF+B)
LINE(120*ZF+A,78*ZF+B)-(250*ZF+A,52*ZF+B)
LINE(10*ZF+A,120*ZF+B)-(360*ZF+A,10*ZF+B)
PAINT(100*ZF+A,65*ZF+B),1:PAINT(270*ZF+A,65*ZF+B),1
RETURN
```

```
30:
REM  aviation gp #30
U=0:U1$="AVIATION GROUP":U2$=""
CIRCLE(100*ZF+A,65*ZF+B),35*ZF,1,56*PI/180,304*PI/180
CIRCLE(270*ZF+A,65*ZF+B),35*ZF,1,236*PI/180,124*PI/180
LINE(120*ZF+A,52*ZF+B)-(250*ZF+A,78*ZF+B)
LINE(120*ZF+A,78*ZF+B)-(250*ZF+A,52*ZF+B)
PAINT(100*ZF+A,65*ZF+B),1:PAINT(270*ZF+A,65*ZF+B),1
LINE(140*ZF+A,25*ZF+B)-(185*ZF+A,55*ZF+B)
LINE(185*ZF+A,55*ZF+B)-(230*ZF+A,25*ZF+B)
LINE(185*ZF+A,55*ZF+B)-(185*ZF+A,95*ZF+B)
LINE(140*ZF+A,95*ZF+B)-(230*ZF+A,95*ZF+B)
RETURN

41:
REM motorized infantery (divisional) #41
U=4:U1$="MOTORIZED INFANRTY BATTALION":U2$=""
GOSUB u4
RETURN

42:
REM airborne battalion #42
U=4:U1$="AIRBORNE BATTALION":U2$=""
CIRCLE(130*ZF+A,120*ZF+B),50*ZF,1,15*PI/180,165*PI/180
CIRCLE(240*ZF+A,120*ZF+B),50*ZF,1,15*PI/180,165*PI/180
LINE(178*ZF+A,115*ZF+B)-(192*ZF+A,115*ZF+B)
GOSUB u4
RETURN

43:
REM mountaineer battalion #43
U=4:U1$="INFANTRY BATTALION":U2$="MOUNTAINEER DIVISION"
GOSUB u4
LINE (10*ZF+A,10*ZF+B)-(360*ZF+A,120*ZF+B), ,B
LINE(152*ZF+A,120*ZF+B)-(182*ZF+A,100*ZF+B)
LINE(212*ZF+A,120*ZF+B)-(182*ZF+A,100*ZF+B)
PAINT(165*ZF+A,114*ZF+B)
RETURN

44:
REM mechanized infantery battalion (armored brig) #44
U1$="MECHANIZED     INFANTRY    BATTALION":U2$="ARMORED
BRIGADE":U=4
CIRCLE (90*ZF+A,65*ZF+B),48*ZF,1,90*PI/180,270*PI/180
CIRCLE(280*ZF+A,65*ZF+B),48*ZF,1,270*PI/180,90*PI/180
LINE(90*ZF+A,45*ZF+B)-(280*ZF+A,45*ZF+B)
LINE(90*ZF+A,85*ZF+B)-(280*ZF+A,85*ZF+B)
GOSUB u4
RETURN
```

```
45:
REM mechanized infantery battalion (infant. brig) #45
U=4:U1$="MECHANIZED INFANTRY BATTALION"
U2$="MECHANIZED INFANTRY BRIGADE"
CIRCLE (90*ZF+A,65*ZF+B),48*ZF,1,90*PI/180,270*PI/180
CIRCLE(280*ZF+A,65*ZF+B),48*ZF,1,270*PI/180,90*PI/180
LINE(90*ZF+A,45*ZF+B)-(280*ZF+A,45*ZF+B)
LINE(90*ZF+A,85*ZF+B)-(280*ZF+A,85*ZF+B)
GOSUB u4
LINE(10*ZF+A,10*ZF+B)-(35*ZF+A,120*ZF+B),,BF
RETURN

46:
REM motorized infantery (divisional) #46
U=4:U1$="MOTORIZED INFANRTY BATTALION":U2$=""
LINE (165*ZF+A,90*ZF+B)-(205*ZF+A,110*ZF+B)
LINE (165*ZF+A,110*ZF+B)-(205*ZF+A,90*ZF+B)
GOSUB u4
RETURN

47:
REM motorized infantery (aaslt div) #47
U=4:U1$="MOTORIZED INFANRTY BATTALION":U2$=""
GOSUB u4
LINE(140*ZF+A,35*ZF+B)-(185*ZF+A,65*ZF+B)
LINE(185*ZF+A,65*ZF+B)-(230*ZF+A,35*ZF+B)
RETURN

48:
REM airborne infantry battalion #48
U=4:U1$="AIRBORNE INFANTRY BATTALION":U2$=""
CIRCLE(185*ZF+A,80*ZF+B),80*ZF
LINE(50*ZF+A,60*ZF+B)-(280*ZF+A,115*ZF+B),0,BF
LINE(10*ZF+A,10*ZF+B)-(127*ZF+A,47*ZF+B)
LINE(360*ZF+A,10*ZF+B)-(233*ZF+A,47*ZF+B)
LINE(10*ZF+A,120*ZF+B)-(127*ZF+A,73*ZF+B)
LINE(360*ZF+A,120*ZF+B)-(233*ZF+A,73*ZF+B)
LINE (122*ZF+A,60*ZF+B)-(185*ZF+A,90*ZF+B)
LINE(185*ZF+A,90*ZF+B)-(245*ZF+A,60*ZF+B)
LINE(122*ZF+A,60*ZF+B)-(245*ZF+A,60*ZF+B)
RETURN

49:
REM armored reconnaissance battalion  #49
U=4:U1$="ARMORED CAVALRY BATTALION ":U2$=""
CIRCLE (90*ZF+A,65*ZF+B),48*ZF,1,90*PI/180,270*PI/180
CIRCLE(280*ZF+A,65*ZF+B),48*ZF,1,270*PI/180,90*PI/180
LINE (90*ZF+A,45*ZF+B)-(280*ZF+A,45*ZF+B)
LINE (90*ZF+A,85*ZF+B)-(280*ZF+A,85*ZF+B)
LINE (10*ZF+A,120*ZF+B)-(360*ZF+A,10*ZF+B)
RETURN
```

```
50:
REM air cavalry squadron #50
U=4:U1$="AIR CAVALRY SQUADRON":U2$=""
CIRCLE(100*ZF+A,65*ZF+B),35*ZF,1,56*PI/180,304*PI/180
CIRCLE(270*ZF+A,65*ZF+B),35*ZF,1,236*PI/180,124*PI/180
LINE(120*ZF+A,52*ZF+B)-(250*ZF+A,78*ZF+B)
LINE(120*ZF+A,78*ZF+B)-(250*ZF+A,52*ZF+B)
LINE(10*ZF+A,120*ZF+B)-(360*ZF+A,10*ZF+B)
PAINT(100*ZF+A,65*ZF+B),1:PAINT(270*ZF+A,65*ZF+B),1
RETURN

51:
REM armored battalion (mech. inf. brig) #51
U1$="ARMORED BATTALION"
U2$="MECHANIZED INFANTRY BRIGADE"
CIRCLE (90*ZF+A,65*ZF+B),48*ZF,1,90*PI/180,270*PI/180
CIRCLE(280*ZF+A,65*ZF+B),48*ZF,1,270*PI/180,90*PI/180
LINE(90*ZF+A,45*ZF+B)-(280*ZF+A,45*ZF+B)
LINE(90*ZF+A,85*ZF+B)-(280*ZF+A,85*ZF+B)
RETURN

52:
REM armored  battalion (armored brig) #52
U=4:U1$="ARMORED BATTALION":U2$="ARMORED BRIGADE"
CIRCLE (90*ZF+A,65*ZF+B),48*ZF,1,90*PI/180,270*PI/180
CIRCLE(280*ZF+A,65*ZF+B),48*ZF,1,270*PI/180,90*PI/180
LINE (90*ZF+A,45*ZF+B)-(280*ZF+A,45*ZF+B)
LINE (90*ZF+A,85*ZF+B)-(280*ZF+A,85*ZF+B)
LINE (10*ZF+A,10*ZF+B)-(30*ZF+A,120*ZF+B),,BF
RETURN

53:
REM armored  battalion (airborne brig) #53
U1$="ARMORED BATTALION":U2$="AIRBORNE BRIGADE"
CIRCLE (90*ZF+A,65*ZF+B),48*ZF,1,90*PI/180,270*PI/180
CIRCLE(280*ZF+A,65*ZF+B),48*ZF,1,270*PI/180,90*PI/180
LINE (90*ZF+A,45*ZF+B)-(280*ZF+A,45*ZF+B)
LINE (90*ZF+A,85*ZF+B)-(280*ZF+A,85*ZF+B)
CIRCLE(130*ZF+A,120*ZF+B),50*ZF,1,15*PI/180,165*PI/180
CIRCLE(240*ZF+A,120*ZF+B),50*ZF,1,15*PI/180,165*PI/180
LINE(178*ZF+A,115*ZF+B)-(192*ZF+A,115*ZF+B)
RETURN
```

```
54:
REM attack helicopter battalion #54
U1$="ATTACK HELICOPTER BATTALION":U2$=""
CIRCLE (90*ZF+A,65*ZF+B),60*ZF,1,90*PI/180,270*PI/180
CIRCLE(280*ZF+A,65*ZF+B),60*ZF,1,270*PI/180,90*PI/180
LINE(90*ZF+A,40*ZF+B)-(280*ZF+A,40*ZF+B)
LINE(90*ZF+A,90*ZF+B)-(280*ZF+A,90*ZF+B)
CIRCLE(100*ZF+A,65*ZF+B),35*ZF,1,56*PI/180,304*PI/180
CIRCLE(270*ZF+A,65*ZF+B),35*ZF,1,236*PI/180,124*PI/180
LINE(120*ZF+A,52*ZF+B)-(250*ZF+A,78*ZF+B)
LINE(120*ZF+A,78*ZF+B)-(250*ZF+A,52*ZF+B)
PAINT(100*ZF+A,65*ZF+B),1:PAINT(270*ZF+A,65*ZF+B),1
RETURN

55:
REM field artillery battalion #55
U=4:U1$="FIELD ARTY BATTALION":U2$=""
CIRCLE (185*ZF+A,65*ZF+B),30*ZF,3
PAINT(185*ZF+A,65*ZF+B),1,3
RETURN

56:
REM field artilley battalion #56
U=4:U1$="FIELD ARTILLERY BATTALION":U2$=""
CIRCLE
(185*ZF+A,65*ZF+B),30*ZF,3:PAINT(185*ZF+A,65*ZF+B)
LINE (10*ZF+A,10*ZF+B)-(360*ZF+A,120*ZF+B)
LINE (10*ZF+A,120*ZF+B)-(360*ZF+A,10*ZF+B)
RETURN

57:
REM field artilley battalion #57
U=4:U1$="FIELD ARTILLERY BATTALION":U2$=""
CIRCLE (185*ZF+A,65*ZF+B),30*ZF,3
PAINT(185*ZF+A,65*ZF+B),1,3
LINE (10*ZF+A,10*ZF+B)-(360*ZF+A,120*ZF+B)
LINE (10*ZF+A,120*ZF+B)-(360*ZF+A,10*ZF+B)
CIRCLE(130*ZF+A,120*ZF+B),50*ZF,1,15*PI/180,165*PI/180
CIRCLE(240*ZF+A,120*ZF+B),50*ZF,1,15*PI/180,165*PI/180
LINE(178*ZF+A,115*ZF+B)-(192*ZF+A,115*ZF+B)
RETURN

58:
REM reconnaissance battalion #58
U=4:U1$="RECONNAISSANCE BATTALION ":U2$=""
CIRCLE (185*ZF+A,65*ZF+B),30*ZF,3
PAINT(185*ZF+A,65*ZF+B),1,3
LINE (10*ZF+A,120*ZF+B)-(360*ZF+A,10*ZF+B)
RETURN
```

```
59:
REM field arty battalion #59
U1$="FIELD ARTILLERY BATTALION":U2$=""
CIRCLE (90*ZF+A,65*ZF+B),48*ZF,1,90*PI/180,270*PI/180
CIRCLE(280*ZF+A,65*ZF+B),48*ZF,1,270*PI/180,90*PI/180
LINE(90*ZF+A,45*ZF+B)-(280*ZF+A,45*ZF+B)
LINE(90*ZF+A,85*ZF+B)-(280*ZF+A,85*ZF+B)
CIRCLE (180*ZF+A,65*ZF+B),20*ZF,3
PAINT (180*ZF+A,65*ZF+B),3,3
RETURN

60:
REM missile artillery battalion #60
U=4:U1$="MISSILE ARTY BATTALION":U2$=""
LINE(180*ZF+A,35*ZF+B)-(140*ZF+A,60*ZF+B)
LINE(180*ZF+A,35*ZF+B)-(220*ZF+A,60*ZF+B)
LINE(180*ZF+A,45*ZF+B)-(140*ZF+A,70*ZF+B)
LINE(180*ZF+A,45*ZF+B)-(220*ZF+A,70*ZF+B)
CIRCLE (180*ZF+A,80*ZF+B),25*ZF,3
PAINT(180*ZF+A,80*ZF+B),1,3
RETURN

61:
REM pershing battalion #61
U=4:U1$="MISSILE ARTY BATTALION":U2$="PERSHING"
CIRCLE(360*ZF+A,100*ZF+B),
       220*ZF,1,142.5*PI/180,180*PI/180
CIRCLE(5*ZF+A,100*ZF+B),220*ZF,1,0,37.5*PI/180
LINE(140*ZF+A,100*ZF+B)-(140*ZF+A,120*ZF+B)
LINE(225*ZF+A,100*ZF+B)-(225*ZF+A,120*ZF+B)
LINE(143*ZF+A,86*ZF+B)-(222*ZF+A,86*ZF+B)
LINE(140*ZF+A,106*ZF+B)-(225*ZF+A,106*ZF+B)
CIRCLE (180*ZF+A,96*ZF+B),9*ZF,3
PAINT(180*ZF+A,96*ZF+B),1,3
RETURN

62:
REM mountaineer artillery battalion #62
U=4:U1$="FIELD ARTILLERY BATTALION"
U2$="MOUNTAINEER DIVISION"
CIRCLE (185*ZF+A,65*ZF+B),30*ZF,3
PAINT(185*ZF+A,65*ZF+B),1,3
LINE (10*ZF+A,10*ZF+B)-(360*ZF+A,120*ZF+B), ,B
LINE(152*ZF+A,120*ZF+B)-(182*ZF+A,100*ZF+B)
LINE(212*ZF+A,120*ZF+B)-(182*ZF+A,100*ZF+B)
PAINT(165*ZF+A,114*ZF+B)
RETURN
```

```
63:
REM  nbc  battalion  #63
U=4:U1$="NBC BATTALION":U2$=""
CIRCLE (80*ZF+A,50*ZF+B),30*ZF,3
PAINT(80*ZF+A,50*ZF+B),1,3
CIRCLE (290*ZF+A,50*ZF+B),30*ZF,3
PAINT(290*ZF+A,50*ZF+B),1,3
LINE (290*ZF+A,100*ZF+B)-(98*ZF+A,40*ZF+B)
LINE (80*ZF+A,100*ZF+B)-(270*ZF+A,40*ZF+B)
RETURN

64:
REM combat aviation bn #64
U=4:U1$="COMBAT AVIATION BATTALION":U2$=""
CIRCLE(100*ZF+A,65*ZF+B),35*ZF,1,56*PI/180,304*PI/180
CIRCLE(270*ZF+A,65*ZF+B),35*ZF,1,236*PI/180,124*PI/180
LINE(120*ZF+A,52*ZF+B)-(250*ZF+A,78*ZF+B)
LINE(120*ZF+A,78*ZF+B)-(250*ZF+A,52*ZF+B)
PAINT(100*ZF+A,65*ZF+B),1:PAINT(270*ZF+A,65*ZF+B),1
RETURN

65:
REM military intelligence battalion #65
U=4:U1$="MILITARY INTELLIGENCE BATTALION":U2$=""
LINE(220*ZF+A,30*ZF+B)-(260*ZF+A,30*ZF+B)
LINE(220*ZF+A,100*ZF+B)-(260*ZF+A,100*ZF+B)
LINE(80*ZF+A,30*ZF+B)-(80*ZF+A,100*ZF+B)
LINE(160*ZF+A,30*ZF+B)-(160*ZF+A,100*ZF+B)
LINE(240*ZF+A,30*ZF+B)-(240*ZF+A,100*ZF+B)
LINE (80*ZF+A,30*ZF+B)-(120*ZF+A,60*ZF+B)
LINE(120*ZF+A,60*ZF+B)-(160*ZF+A,30*ZF+B)
RETURN

66:
REM medical battalion #66
U=4:U1$="MEDICAL BATTALION":U2$=""
LINE (10*ZF+A,65*ZF+B)-(360*ZF+A,65*ZF+B)
LINE (185*ZF+A,10*ZF+B)-(185*ZF+A,120*ZF+B)
RETURN

67:
REM  medical evacuation battalion #67
U=4:U1$="MEDICAL EVACUATION BATTALION":U2$=""
CIRCLE(180*ZF+A,65*ZF+B),70*ZF
LINE (130*ZF+A,46*ZF+B)-(232*ZF+A,84*ZF+B)
LINE(131*ZF+A,85*ZF+B)-(230*ZF+A,46*ZF+B)
LINE (185*ZF+A,10*ZF+B)-(185*ZF+A,120*ZF+B)
LINE (10*ZF+A,65*ZF+B)-(360*ZF+A,65*ZF+B)
LINE(10*ZF+A,10*ZF+B)-(40*ZF+A,120*ZF+B),,BF
RETURN
```

```
68:
REM  maintainance  battalion  #68
U=4:U1$="MAINTAINANCE BATTALION":U2$=""
CIRCLE (50*ZF+A,65*ZF+B),50*ZF,1,270*PI/180,90*PI/180
CIRCLE (320*ZF+A,65*ZF+B),50*ZF,1,90*PI/180,270*PI/180
LINE (100*ZF+A,65*ZF+B)-(270*ZF+A,65*ZF+B)
RETURN

69:
REM  maintainance  battalion  (electronic mn.) #69
U=4:U1$="MAINTAINANCE BATTALION"
U2$="(ELECTRONIC MAINTAINANCE)"
CIRCLE (50*ZF+A,85*ZF+B),50*ZF,1,270*PI/180,90*PI/180
CIRCLE (320*ZF+A,85*ZF+B),50*ZF,1,90*PI/180,270*PI/180
LINE (100*ZF+A,85*ZF+B)-(270*ZF+A,85*ZF+B)
LINE(140*ZF+A,30*ZF+B)-(140*ZF+A,50*ZF+B)
LINE (230*ZF+A,30*ZF+B)-(230*ZF+A,50*ZF+B)
LINE(230*ZF+A,30*ZF+B)-(140*ZF+A,40*ZF+B)
LINE(230*ZF+A,50*ZF+B)-(140*ZF+A,40*ZF+B)
PAINT(160*ZF+A,40*ZF+B)
LINE(57*ZF+A,40*ZF+B)-(310*ZF+A,40*ZF+B)
RETURN

70:
REM  supply battalion  #70
U=4:U1$="SUPPLY BATTALION":U2$=""
LINE (10*ZF+A,100*ZF+B)-(360*ZF+A,100*ZF+B)
RETURN

71:
REM  supply battalion for special weapons #71
U=4:U1$="SUPPLY BATTALION (SW)":U2$=""
LINE (10*ZF+A,100*ZF+B)-(360*ZF+A,100*ZF+B)
LINE(235*ZF+A,55*ZF+B)-(220*ZF+A,75*ZF+B)
CIRCLE(150*ZF+A,60*ZF+B),13*ZF,1,45*PI/180,270*PI/180
CIRCLE(150*ZF+A,70*ZF+B),13*ZF,1,225*PI/180,90*PI/180
LINE(185*ZF+A,55*ZF+B)-(200*ZF+A,75*ZF+B)
LINE(200*ZF+A,75*ZF+B)-(210*ZF+A,60*ZF+B)
LINE(210*ZF+A,60*ZF+B)-(220*ZF+A,75*ZF+B)
RETURN

72:
REM  transportation  battalion #72
U=4:U1$="TRANSPORTATION BATTALION":U2$=""
CIRCLE(180*ZF+A,65*ZF+B),70*ZF
LINE (130*ZF+A,46*ZF+B)-(232*ZF+A,84*ZF+B)
LINE(131*ZF+A,85*ZF+B)-(230*ZF+A,46*ZF+B)
LINE (180*ZF+A,37*ZF+B)-(180*ZF+A,93*ZF+B)
LINE (110*ZF+A,65*ZF+B)-(250*ZF+A,65*ZF+B)
RETURN
```

```
73:
REM military police  battalion #73
U=4:U1$="MILITARY POLICE BATTALION":U2$=""
CIRCLE(255*ZF+A,42*ZF+B),29*ZF,1,270*PI/180,90*PI/180
LINE(240*ZF+A,30*ZF+B)-(255*ZF+A,30*ZF+B)
LINE(240*ZF+A,54*ZF+B)-(255*ZF+A,54*ZF+B)
LINE(80*ZF+A,30*ZF+B)-(80*ZF+A,100*ZF+B)
LINE(160*ZF+A,30*ZF+B)-(160*ZF+A,100*ZF+B)
LINE(240*ZF+A,30*ZF+B)-(240*ZF+A,100*ZF+B)
LINE (80*ZF+A,30*ZF+B)-(120*ZF+A,60*ZF+B)
LINE(120*ZF+A,60*ZF+B)-(160*ZF+A,30*ZF+B)
RETURN

74:
REM ada battalion #74
U=4:U1$="ADA BATTALION":U2$="CORPS ADA COMMAND"
CIRCLE
(186*ZF+A,175*ZF+B),220*ZF,1,39*PI/180,142*PI/180
RETURN

75:
REM ada battalion #75
U=4:U1$="ADA BATTALION":U2$="AIRBORNE DIVISION"
CIRCLE
(186*ZF+A,175*ZF+B),220*ZF,1,39*PI/180,142*PI/180
CIRCLE(130*ZF+A,120*ZF+B),50*ZF,1,15*PI/180,165*PI/180
CIRCLE(240*ZF+A,120*ZF+B),50*ZF,1,15*PI/180,165*PI/180
LINE(178*ZF+A,115*ZF+B)-(192*ZF+A,115*ZF+B)
RETURN

76:
REM  ew battalion #76
U=4:U1$="EW BATTALION":U2$=""
LINE (10*ZF+A,120*ZF+B)-(185*ZF+A,50*ZF+B)
LINE(185*ZF+A,80*ZF+B)-(360*ZF+A,10*ZF+B)
LINE(185*ZF+A,80*ZF+B)-(185*ZF+A,50*ZF+B)
RETURN

77:
REM  signal battalion #77
U=4:U1$="SIGNAL BATTALION":U2$=""
LINE (10*ZF+A,10*ZF+B)-(185*ZF+A,80*ZF+B)
LINE(185*ZF+A,50*ZF+B)-(360*ZF+A,120*ZF+B)
LINE(185*ZF+A,80*ZF+B)-(185*ZF+A,50*ZF+B)
RETURN
```

```
78:
REM  signal construction battalion #78
U=4:U1$="SIGNAL CONSTRUCTION BATTALION":U2$=""
LINE (70*ZF+A,42*ZF+B)-(300*ZF+A,42*ZF+B)
LINE(70*ZF+A,53*ZF+B)-(300*ZF+A,53*ZF+B)
LINE(120*ZF+A,65*ZF+B)-(250*ZF+A,65*ZF+B)
LINE(182*ZF+A,65*ZF+B)-(182*ZF+A,91*ZF+B)
RETURN

79:
REM signal construction battalion #79
CIRCLE(185*ZF+A,B-20*ZF),160*ZF,1,230*PI/180,310*PI/180
U=4:U1$="SIGNAL CONSTRUCTION BATTALION":U2$=""
LINE (130*ZF+A,45*ZF+B)-(130*ZF+A,100*ZF+B)
LINE (225*ZF+A,45*ZF+B)-(225*ZF+A,85*ZF+B)
LINE (225*ZF+A,85*ZF+B)-(250*ZF+A,65*ZF+B)
LINE(250*ZF+A,65*ZF+B)-(250*ZF+A,100*ZF+B)
RETURN

80:
REM feld ersatz battailon #80
U=4:U1$="FELD ERSATZ BATTALION":U2$=""
LINE(50*ZF+A,100*ZF+B)-(50*ZF+A,30*ZF+B)
LINE(50*ZF+A,30*ZF+B)-(110*ZF+A,30*ZF+B)
LINE(50*ZF+A,65*ZF+B)-(100*ZF+A,65*ZF+B)
LINE(120*ZF+A,100*ZF+B)-(120*ZF+A,30*ZF+B)
LINE(120*ZF+A,30*ZF+B)-(180*ZF+A,30*ZF+B)
LINE(120*ZF+A,65*ZF+B)-(170*ZF+A,65*ZF+B)
LINE(120*ZF+A,100*ZF+B)-(180*ZF+A,100*ZF+B)
CIRCLE(225*ZF+A,77*ZF+B),25*ZF,1,0,PI
LINE(200*ZF+A,65*ZF+B)-(200*ZF+A,100*ZF+B)
CIRCLE(290*ZF+A,75*ZF+B),20*ZF,1,0,PI*3/2
CIRCLE(290*ZF+A,91*ZF+B),20*ZF,1,180*PI/180,90*PI/180
RETURN

81:
REM engineer battalion #81
U1$="ENGINEER BATTALION":U2$=""
LINE (90*ZF+A,55*ZF+B)-(280*ZF+A,55*ZF+B)
LINE (90*ZF+A,55*ZF+B)-(90*ZF+A,75*ZF+B)
LINE (180*ZF+A,55*ZF+B)-(180*ZF+A,75*ZF+B)
LINE (280*ZF+A,55*ZF+B)-(280*ZF+A,75*ZF+B)
RETURN
```

```
82:
REM armored engineer battalion #82
U1$="ARMORED ENGINEER BATTALION":U2$=""
CIRCLE (90*ZF+A,65*ZF+B),48*ZF,1,90*PI/180,270*PI/180
CIRCLE(280*ZF+A,65*ZF+B),48*ZF,1,270*PI/180,90*PI/180
LINE(90*ZF+A,45*ZF+B)-(280*ZF+A,45*ZF+B)
LINE(90*ZF+A,85*ZF+B)-(280*ZF+A,85*ZF+B)
LINE (90*ZF+A,55*ZF+B)-(280*ZF+A,55*ZF+B)
LINE (90*ZF+A,55*ZF+B)-(90*ZF+A,75*ZF+B)
LINE (180*ZF+A,55*ZF+B)-(180*ZF+A,75*ZF+B)
LINE (280*ZF+A,55*ZF+B)-(280*ZF+A,75*ZF+B)
RETURN

83:
REM engineer battalion #83
U=4:U1$="ENGINEERING FERRY BATTALION":U2$=""
CIRCLE(57*ZF+A,50*ZF+B),32*ZF,1,0,PI
CIRCLE(121*ZF+A,50*ZF+B),32*ZF,1,PI,0
CIRCLE(185*ZF+A,50*ZF+B),32*ZF,1,0,PI
CIRCLE(249*ZF+A,50*ZF+B),32*ZF,1,PI,0
CIRCLE(313*ZF+A,50*ZF+B),32*ZF,1,0,PI
LINE (80*ZF+A,80*ZF+B)-(290*ZF+A,80*ZF+B)
LINE (80*ZF+A,80*ZF+B)-(80*ZF+A,99*ZF+B)
LINE (180*ZF+A,80*ZF+B)-(180*ZF+A,99*ZF+B)
LINE (290*ZF+A,80*ZF+B)-(290*ZF+A,99*ZF+B)
RETURN

84:
REM engineer battalion (bridge layers) #84
U=4:U1$="ENGINEER BATTALION":U2$="BRIDGE LAYERS"
LINE (90*ZF+A,80*ZF+B)-(280*ZF+A,80*ZF+B)
LINE (90*ZF+A,80*ZF+B)-(90*ZF+A,99*ZF+B)
LINE (185*ZF+A,80*ZF+B)-(185*ZF+A,99*ZF+B)
LINE (280*ZF+A,80*ZF+B)-(280*ZF+A,99*ZF+B)
LINE (110*ZF+A,55*ZF+B)-(260*ZF+A,55*ZF+B)
LINE (110*ZF+A,40*ZF+B)-(260*ZF+A,40*ZF+B)
LINE (90*ZF+A,65*ZF+B)-(110*ZF+A,55*ZF+B)
LINE (260*ZF+A,40*ZF+B)-(280*ZF+A,30*ZF+B)
LINE (90*ZF+A,30*ZF+B)-(110*ZF+A,40*ZF+B)
LINE (260*ZF+A,55*ZF+B)-(280*ZF+A,65*ZF+B)
RETURN
```

```
85:
REM support battalion #85
U=4:U1$="SUPPORT BATTALION":U2$="AIRBORNE DIVISION"
CIRCLE(130*ZF+A,120*ZF+B),50*ZF,1,15*PI/180,165*PI/180
CIRCLE(240*ZF+A,120*ZF+B),50*ZF,1,15*PI/180,165*PI/180
LINE(178*ZF+A,115*ZF+B)-(192*ZF+A,115*ZF+B)
CIRCLE(150*ZF+A,60*ZF+B),13*ZF,1,45*PI/180,270*PI/180
CIRCLE(150*ZF+A,70*ZF+B),13*ZF,1,225*PI/180,90*PI/180
CIRCLE(180*ZF+A,70*ZF+B),13*ZF,1,225*PI/180,135*PI/180
LINE(172*ZF+A,66*ZF+B)-(172*ZF+A,85*ZF+B)
CIRCLE(213*ZF+A,70*ZF+B),13*ZF,1,180*PI/180,340*PI/180
LINE(200*ZF+A,55*ZF+B)-(200*ZF+A,70*ZF+B)
LINE(192*ZF+A,63*ZF+B)-(210*ZF+A,63*ZF+B)
RETURN

86:
REM support battalion #86.
U=4:U1$="SUPPORT BATTALION":U2$=""
CIRCLE(150*ZF+A,60*ZF+B),13*ZF,1,45*PI/180,270*PI/180
CIRCLE(150*ZF+A,70*ZF+B),13*ZF,1,225*PI/180,90*PI/180
CIRCLE(180*ZF+A,70*ZF+B),13*ZF,1,225*PI/180,135*PI/180
LINE(172*ZF+A,66*ZF+B)-(172*ZF+A,85*ZF+B)
CIRCLE(213*ZF+A,70*ZF+B),13*ZF,1,180*PI/180,340*PI/180
LINE(200*ZF+A,55*ZF+B)-(200*ZF+A,70*ZF+B)
LINE(192*ZF+A,63*ZF+B)-(210*ZF+A,63*ZF+B)
RETURN

87:
REM  supply and transportation  battalion #87
U=4:U1$="SUPPLY AND TRANSPORTATION BATTALION":U2$=""
CIRCLE(180*ZF+A,65*ZF+B),70*ZF
LINE (130*ZF+A,46*ZF+B)-(232*ZF+A,84*ZF+B)
LINE(131*ZF+A,85*ZF+B)-(230*ZF+A,46*ZF+B)
LINE (180*ZF+A,37*ZF+B)-(180*ZF+A,93*ZF+B)
LINE (110*ZF+A,65*ZF+B)-(250*ZF+A,65*ZF+B)
LINE (10*ZF+A,100*ZF+B)-(360*ZF+A,100*ZF+B)
RETURN

88:
REM field arty battalion (mech division) #88
U1$="FIELD ARTILLERY BATTALION":U2$="MECH. DIVISION"
CIRCLE (185*ZF+A,65*ZF+B),30*ZF,3
PAINT(185*ZF+A,65*ZF+B)
CIRCLE (90*ZF+A,65*ZF+B),48*ZF,1,90*PI/180,270*PI/180
CIRCLE(280*ZF+A,65*ZF+B),48*ZF,1,270*PI/180,90*PI/180
LINE(90*ZF+A,45*ZF+B)-(280*ZF+A,45*ZF+B)
LINE(90*ZF+A,85*ZF+B)-(280*ZF+A,85*ZF+B)
LINE(10*ZF+A,10*ZF+B)-(360*ZF+A,120*ZF+B)
LINE(10*ZF+A,120*ZF+B)-(360*ZF+A,10*ZF+B)
RETURN
```

```
89:
REM field arty battalion (AASLT division) #89
U1$="FIELD ARTILLERY BATTALION":U2$="AASLT DIVISION"
CIRCLE (185*ZF+A,65*ZF+B),30*ZF,3
PAINT(185*ZF+A,65*ZF+B)
LINE(10*ZF+A,10*ZF+B)-(360*ZF+A,120*ZF+B)
LINE(10*ZF+A,120*ZF+B)-(360*ZF+A,10*ZF+B)
LINE(145*ZF+A,20*ZF+B)-(185*ZF+A,40*ZF+B)
LINE(225*ZF+A,20*ZF+B)-(185*ZF+A,40*ZF+B)
LINE(185*ZF+A,40*ZF+B)-(185*ZF+A,100*ZF+B)
LINE(145*ZF+A,100*ZF+B)-(225*ZF+A,100*ZF+B)
RETURN

101:
REM military band #101
U=5:U1$="MILITARY BAND":U2$=""
CIRCLE(25*ZF+A,60*ZF+B),80*ZF,1,0,PI/2
CIRCLE(185*ZF+A,60*ZF+B),80*ZF,1,PI,0
CIRCLE(345*ZF+A,60*ZF+B),80*ZF,1,PI/2,PI
LINE(185*ZF+A,30*ZF+B)-(185*ZF+A,110*ZF+B)
LINE(140*ZF+A,30*ZF+B)-(140*ZF+A,70*ZF+B)
LINE(230*ZF+A,30*ZF+B)-(230*ZF+A,70*ZF+B)
LINE(100*ZF+A,50*ZF+B)-(270*ZF+A,50*ZF+B)
RETURN

102:
REM military police co #102
U=5:U1$="MILITARY POLICE COMPANY":U2$=""
CIRCLE(255*ZF+A,42*ZF+B),29*ZF,1,270*PI/180,90*PI/180
LINE(240*ZF+A,30*ZF+B)-(255*ZF+A,30*ZF+B)
LINE(240*ZF+A,54*ZF+B)-(255*ZF+A,54*ZF+B)
LINE(80*ZF+A,30*ZF+B)-(80*ZF+A,100*ZF+B)
LINE(160*ZF+A,30*ZF+B)-(160*ZF+A,100*ZF+B)
LINE(240*ZF+A,30*ZF+B)-(240*ZF+A,100*ZF+B)
LINE (80*ZF+A,30*ZF+B)-(120*ZF+A,60*ZF+B)
LINE(120*ZF+A,60*ZF+B)-(160*ZF+A,30*ZF+B)
RETURN

103:
REM chemical co #103
U=5:U1$="CHEMICAL COMPANY":U2$=""
CIRCLE (80*ZF+A,50*ZF+B),30*ZF,3
PAINT(80*ZF+A,50*ZF+B),1,3
CIRCLE (290*ZF+A,50*ZF+B),30*ZF,3
PAINT(290*ZF+A,50*ZF+B),1,3
LINE (290*ZF+A,100*ZF+B)-(98*ZF+A,40*ZF+B)
LINE (80*ZF+A,100*ZF+B)-(270*ZF+A,40*ZF+B)
RETURN
```

```
104:
REM cavalry troop #104
U=5:U1$="CAVALRY TROOP":U2$=""
CIRCLE (90*ZF+A,65*ZF+B),48*ZF,1,90*PI/180,270*PI/180
CIRCLE(280*ZF+A,65*ZF+B),48*ZF,1,270*PI/180,90*PI/180
LINE (90*ZF+A,45*ZF+B)-(280*ZF+A,45*ZF+B)
LINE(90*ZF+A,85*ZF+B)-(280*ZF+A,85*ZF+B)
LINE(10*ZF+A,120*ZF+B)-(360*ZF+A,10*ZF+B)
RETURN

105:
REM military intelligence company #105
U=5:U1$="MILITARY INTELLIGENCE COMPANY":U2$=""
LINE(220*ZF+A,30*ZF+B)-(260*ZF+A,30*ZF+B)
LINE(220*ZF+A,100*ZF+B)-(260*ZF+A,100*ZF+B)
LINE(80*ZF+A,30*ZF+B)-(80*ZF+A,100*ZF+B)
LINE(160*ZF+A,30*ZF+B)-(160*ZF+A,100*ZF+B)
LINE(240*ZF+A,30*ZF+B)-(240*ZF+A,100*ZF+B)
LINE (80*ZF+A,30*ZF+B)-(120*ZF+A,60*ZF+B)
LINE(120*ZF+A,60*ZF+B)-(160*ZF+A,30*ZF+B)
RETURN

106:
REM hhq corps command #106
U=5:U1$="HEADQUATERS AND HEADQUARTERS CO"
U2$="CORPS COMMAND"
LINE(10*ZF+A,10*ZF+B)-(360*ZF+A,30*ZF+B),1,BF
LINE(140*ZF+A,36*ZF+B)-(160*ZF+A,43*ZF+B)
LINE(160*ZF+A,36*ZF+B)-(140*ZF+A,43*ZF+B)
LINE(175*ZF+A,36*ZF+B)-(195*ZF+A,43*ZF+B)
LINE(195*ZF+A,36*ZF+B)-(175*ZF+A,43*ZF+B)
LINE(210*ZF+A,36*ZF+B)-(230*ZF+A,43*ZF+B)
LINE(210*ZF+A,43*ZF+B)-(230*ZF+A,36*ZF+B)
RETURN

107:
REM hhq corps maintainance command #107
U=5:U1$="HEADQUATERS AND HEADQUARTERS CO"
U2$="CORPS MAINTAINANCE COMMAND"
CIRCLE (50*ZF+A,65*ZF+B),50*ZF,1,270*PI/180,90*PI/180
CIRCLE (320*ZF+A,65*ZF+B),50*ZF,1,90*PI/180,270*PI/180
LINE (100*ZF+A,65*ZF+B)-(270*ZF+A,65*ZF+B)
LINE(10*ZF+A,10*ZF+B)-(360*ZF+A,25*ZF+B),1,BF
LINE(175*ZF+A,28*ZF+B)-(195*ZF+A,37*ZF+B)
LINE(195*ZF+A,28*ZF+B)-(175*ZF+A,37*ZF+B)
RETURN
```

```
108:
REM corps signal command #108
U=5:U1$="HEADQUATERS AND HEADQUARTERS CO"
U2$="CORPS SIGNAL COMMAND"
LINE (10*ZF+A,10*ZF+B)-(185*ZF+A,80*ZF+B)
LINE(185*ZF+A,50*ZF+B)-(360*ZF+A,120*ZF+B)
LINE(185*ZF+A,80*ZF+B)-(185*ZF+A,50*ZF+B)
LINE(10*ZF+A,10*ZF+B)-(360*ZF+A,25*ZF+B),1,BF
LINE(175*ZF+A,28*ZF+B)-(195*ZF+A,37*ZF+B)
LINE(195*ZF+A,28*ZF+B)-(175*ZF+A,37*ZF+B)
RETURN

109:
REM hhq arty command #109
U=5:U1$="HEADQUATERS AND HEADQUARTERS CO"
U2$="ARTILLERY COMMAND"
CIRCLE (185*ZF+A,65*ZF+B),30*ZF,3
PAINT(185*ZF+A,65*ZF+B),1,3
LINE(10*ZF+A,10*ZF+B)-(360*ZF+A,25*ZF+B),1,BF
LINE(175*ZF+A,28*ZF+B)-(195*ZF+A,37*ZF+B)
LINE(195*ZF+A,28*ZF+B)-(175*ZF+A,37*ZF+B)
RETURN

110:
REM hhq corps ada command #110
U=5:U1$="HEADQUATERS AND HEADQUARTERS CO"
U2$="CORPS ADA COMMAND"
CIRCLE
(186*ZF+A,175*ZF+B),220*ZF,1,39*PI/180,142*PI/180
LINE(10*ZF+A,10*ZF+B)-(360*ZF+A,25*ZF+B),1,BF
LINE(175*ZF+A,28*ZF+B)-(195*ZF+A,37*ZF+B)
LINE(195*ZF+A,28*ZF+B)-(175*ZF+A,37*ZF+B)
RETURN

111:
REM hhq corps engineer command #111
U=5:U1$="HEADQUATERS AND HEADQUARTERS CO"
U2$="CORPS ENGINEER COMMAND"
LINE (90*ZF+A,55*ZF+B)-(280*ZF+A,55*ZF+B)
LINE (90*ZF+A,55*ZF+B)-(90*ZF+A,75*ZF+B)
LINE (180*ZF+A,55*ZF+B)-(180*ZF+A,75*ZF+B)
LINE (280*ZF+A,55*ZF+B)-(280*ZF+A,75*ZF+B)
LINE(10*ZF+A,10*ZF+B)-(360*ZF+A,25*ZF+B),1,BF
LINE(175*ZF+A,28*ZF+B)-(195*ZF+A,37*ZF+B)
LINE(195*ZF+A,28*ZF+B)-(175*ZF+A,37*ZF+B)
RETURN
```

```
112:
REM hhq corps medical command #112
U=5:U1$="HEADQUATERS AND HEADQUARTERS CO"
U2$="CORPS MEDICAL COMMAND"
LINE(10*ZF+A,65*ZF+B)-(360*ZF+A,65*ZF+B)
LINE(185*ZF+A,50*ZF+B)-(185*ZF+A,120*ZF+B)
LINE(10*ZF+A,10*ZF+B)-(360*ZF+A,25*ZF+B),1,BF
LINE(175*ZF+A,28*ZF+B)-(195*ZF+A,37*ZF+B)
LINE(195*ZF+A,28*ZF+B)-(175*ZF+A,37*ZF+B)
RETURN

113:
REM hhq corps supply command #113
U=5:U1$="HEADQUATERS AND HEADQUARTERS CO"
U2$="CORPS SUPPLY COMMAND"
LINE (10*ZF+A,100*ZF+B)-(360*ZF+A,100*ZF+B)
LINE(10*ZF+A,10*ZF+B)-(360*ZF+A,25*ZF+B),1,BF
LINE(175*ZF+A,28*ZF+B)-(195*ZF+A,37*ZF+B)
LINE(195*ZF+A,28*ZF+B)-(175*ZF+A,37*ZF+B)
RETURN

114:
REM hhq aviation brigade #114
U=5:U1$="HEADQUATERS AND HEADQUARTERS CO"
U2$="AVIATION BRIGADE"
CIRCLE(100*ZF+A,65*ZF+B),35*ZF,1,56*PI/180,304*PI/180
CIRCLE(270*ZF+A,65*ZF+B),35*ZF,1,236*PI/180,124*PI/180
LINE(120*ZF+A,52*ZF+B)-(250*ZF+A,78*ZF+B)
LINE(120*ZF+A,78*ZF+B)-(250*ZF+A,52*ZF+B)
PAINT(100*ZF+A,65*ZF+B),1:PAINT(270*ZF+A,65*ZF+B),1
LINE(10*ZF+A,10*ZF+B)-(360*ZF+A,25*ZF+B),1,BF
LINE(175*ZF+A,28*ZF+B)-(195*ZF+A,37*ZF+B)
LINE(195*ZF+A,28*ZF+B)-(175*ZF+A,37*ZF+B)
RETURN

115:
REM hhq division command #115
U=5:U1$="HEADQUATERS AND HEADQUARTERS CO"
U2$="DIVISION COMMAND"
LINE(10*ZF+A,10*ZF+B)-(360*ZF+A,25*ZF+B),1,BF
LINE(150*ZF+A,36*ZF+B)-(170*ZF+A,42*ZF+B)
LINE(170*ZF+A,36*ZF+B)-(150*ZF+A,42*ZF+B)
LINE(200*ZF+A,36*ZF+B)-(220*ZF+A,42*ZF+B)
LINE(220*ZF+A,36*ZF+B)-(200*ZF+A,42*ZF+B)
RETURN
```

```
116:
REM hhq mountaineer brigade #116
U=5:U1$="HEADQUATERS AND HEADQUARTERS CO"
U2$="MOUNTAINEER BRIGADE"
LINE (10*ZF+A,10*ZF+B)-(360*ZF+A,120*ZF+B)
LINE (10*ZF+A,120*ZF+B)-(360*ZF+A,10*ZF+B)
LINE (10*ZF+A,10*ZF+B)-(360*ZF+A,120*ZF+B), ,B
LINE(10*ZF+A,10*ZF+B)-(360*ZF+A,25*ZF+B),1,BF
LINE(152*ZF+A,120*ZF+B)-(182*ZF+A,100*ZF+B)
LINE(212*ZF+A,120*ZF+B)-(182*ZF+A,100*ZF+B)
PAINT(165*ZF+A,114*ZF+B)
LINE(175*ZF+A,28*ZF+B)-(195*ZF+A,37*ZF+B)
LINE(195*ZF+A,28*ZF+B)-(175*ZF+A,37*ZF+B)
RETURN

117:
REM hhq airborne brigade #117
U=5:U1$="HEADQUATERS AND HEADQUARTERS CO"
U2$="AIRBORNE BRIGADE"
CIRCLE(130*ZF+A,120*ZF+B),50*ZF,1,15*PI/180,165*PI/180
CIRCLE(240*ZF+A,120*ZF+B),50*ZF,1,15*PI/180,165*PI/180
LINE(178*ZF+A,115*ZF+B)-(192*ZF+A,115*ZF+B)
LINE(10*ZF+A,10*ZF+B)-(360*ZF+A,25*ZF+B),1,BF
LINE(175*ZF+A,28*ZF+B)-(195*ZF+A,37*ZF+B)
LINE(195*ZF+A,28*ZF+B)-(175*ZF+A,37*ZF+B)
GOSUB u4
RETURN

118:
REM hhq mech brigade #118
U=5:U1$="HEADQUATERS AND HEADQUARTERS CO"
U2$="MECH BRIGADE"
CIRCLE (90*ZF+A,70*ZF+B),48*ZF,1,90*PI/180,270*PI/180
CIRCLE(280*ZF+A,70*ZF+B),48*ZF,1,270*PI/180,90*PI/180
LINE (90*ZF+A,50*ZF+B)-(280*ZF+A,50*ZF+B)
LINE (90*ZF+A,90*ZF+B)-(280*ZF+A,90*ZF+B)
GOSUB u4
LINE(10*ZF+A,10*ZF+B)-(360*ZF+A,25*ZF+B),1,BF
LINE(175*ZF+A,28*ZF+B)-(195*ZF+A,37*ZF+B)
LINE(195*ZF+A,28*ZF+B)-(175*ZF+A,37*ZF+B)
RETURN
```

```
119:
REM hhq  armd brigade #119
U=5:U1$="HEADQUATERS AND HEADQUARTERS CO"
U2$="ARMORED BRIGADE"
CIRCLE (90*ZF+A,65*ZF+B),48*ZF,1,90*PI/180,270*PI/180
CIRCLE(280*ZF+A,65*ZF+B),48*ZF,1,270*PI/180,90*PI/180
LINE (90*ZF+A,45*ZF+B)-(280*ZF+A,45*ZF+B)
LINE (90*ZF+A,85*ZF+B)-(280*ZF+A,85*ZF+B)
LINE(10*ZF+A,10*ZF+B)-(360*ZF+A,25*ZF+B),1,BF
LINE(175*ZF+A,28*ZF+B)-(195*ZF+A,37*ZF+B)
LINE(195*ZF+A,28*ZF+B)-(175*ZF+A,37*ZF+B)
RETURN

120:
REM hhq air cavalry combat brigade #120
U=5:U1$="HHQ AIR CAVALRY COMBAT BRIGADE":U2$=""
CIRCLE(100*ZF+A,65*ZF+B),35*ZF,1,56*PI/180,304*PI/180
CIRCLE(270*ZF+A,65*ZF+B),35*ZF,1,236*PI/180,124*PI/180
LINE(120*ZF+A,52*ZF+B)-(250*ZF+A,78*ZF+B)
LINE(120*ZF+A,78*ZF+B)-(250*ZF+A,52*ZF+B)
LINE(10*ZF+A,120*ZF+B)-(360*ZF+A,10*ZF+B)
PAINT(100*ZF+A,65*ZF+B),1:PAINT(270*ZF+A,65*ZF+B),1
LINE(10*ZF+A,10*ZF+B)-(360*ZF+A,30*ZF+B),1,BF
LINE(170*ZF+A,36*ZF+B)-(190*ZF+A,42*ZF+B)
LINE(190*ZF+A,36*ZF+B)-(170*ZF+A,42*ZF+B)
RETURN

121:
REM  ew company #121
U=5:U1$="EW COMPANY":U2$=""
LINE (10*ZF+A,120*ZF+B)-(185*ZF+A,50*ZF+B)
LINE(185*ZF+A,80*ZF+B)-(360*ZF+A,10*ZF+B)
LINE(185*ZF+A,80*ZF+B)-(185*ZF+A,50*ZF+B)
RETURN

122:
REM  signal company #122
U=5:U1$="SIGNAL COMPANY":U2$=""
LINE (10*ZF+A,10*ZF+B)-(185*ZF+A,80*ZF+B)
LINE(185*ZF+A,50*ZF+B)-(360*ZF+A,120*ZF+B)
LINE(185*ZF+A,80*ZF+B)-(185*ZF+A,50*ZF+B)
RETURN

123:
REM fernspaeh co #123
U=5:U1$="FERNSPAEH KOMPANIE":U2$=""
LINE(10*ZF+A,120*ZF+B)-(360*ZF+A,10*ZF+B)
LINE(50*ZF+A,65*ZF+B)-(320*ZF+A,65*ZF+B)
LINE(280*ZF+A,45*ZF+B)-(320*ZF+A,65*ZF+B)
LINE(280*ZF+A,85*ZF+B)-(320*ZF+A,65*ZF+B)
RETURN
```

```
124:
REM armored engineer company #124
U1$="ARMORED ENGINEER COMPANY":U2$=""
CIRCLE (90*ZF+A,65*ZF+B),48*ZF,1,90*PI/180,270*PI/180
CIRCLE(280*ZF+A,65*ZF+B),48*ZF,1,270*PI/180,90*PI/180
LINE(90*ZF+A,45*ZF+B)-(280*ZF+A,45*ZF+B)
LINE(90*ZF+A,85*ZF+B)-(280*ZF+A,85*ZF+B)
LINE (90*ZF+A,55*ZF+B)-(280*ZF+A,55*ZF+B)
LINE (90*ZF+A,55*ZF+B)-(90*ZF+A,75*ZF+B)
LINE (180*ZF+A,55*ZF+B)-(180*ZF+A,75*ZF+B)
LINE (280*ZF+A,55*ZF+B)-(280*ZF+A,75*ZF+B)
RETURN

125:
REM adjudant general #125
U=5:U1$="ADJUDANT GENERAL":U2$=""
LINE(70*ZF+A,100*ZF+B)-(120*ZF+A,30*ZF+B)
LINE(120*ZF+A,30*ZF+B)-(170*ZF+A,100*ZF+B)
LINE(95*ZF+A,65*ZF+B)-(145*ZF+A,65*ZF+B)
CIRCLE(240*ZF+A,85*ZF+B),40*ZF,1,PI,2*PI
CIRCLE(240*ZF+A,50*ZF+B),40*ZF,1,10/180*PI,PI
LINE(200*ZF+A,50*ZF+B)-(200*ZF+A,85*ZF+B)
LINE(280*ZF+A,65*ZF+B)-(280*ZF+A,85*ZF+B)
LINE(240*ZF+A,65*ZF+B)-(280*ZF+A,65*ZF+B)
RETURN

126:
REM engineer co #126
U=5:U1$="ENGINEER COMPANY":U2$=""
LINE (90*ZF+A,55*ZF+B)-(280*ZF+A,55*ZF+B)
LINE (90*ZF+A,55*ZF+B)-(90*ZF+A,75*ZF+B)
LINE (180*ZF+A,55*ZF+B)-(180*ZF+A,75*ZF+B)
LINE (280*ZF+A,55*ZF+B)-(280*ZF+A,75*ZF+B)
LINE (10*ZF+A,10*ZF+B)-(360*ZF+A,120*ZF+B), ,B
LINE(152*ZF+A,120*ZF+B)-(182*ZF+A,100*ZF+B)
LINE(212*ZF+A,120*ZF+B)-(182*ZF+A,100*ZF+B)
PAINT(165*ZF+A,114*ZF+B)
RETURN
```

```
127:
REM chemical co #127
U=5:U1$="CHEMICAL COMPANY":U2$=""
CIRCLE (80*ZF+A,50*ZF+B),30*ZF,3
PAINT(80*ZF+A,50*ZF+B),1,3
CIRCLE (290*ZF+A,50*ZF+B),30*ZF,3
PAINT(290*ZF+A,50*ZF+B),1,3
LINE (290*ZF+A,100*ZF+B)-(98*ZF+A,40*ZF+B)
LINE (80*ZF+A,100*ZF+B)-(270*ZF+A,40*ZF+B)
LINE (10*ZF+A,10*ZF+B)-(360*ZF+A,120*ZF+B), ,B
LINE(152*ZF+A,120*ZF+B)-(182*ZF+A,100*ZF+B)
LINE(212*ZF+A,120*ZF+B)-(182*ZF+A,100*ZF+B)
PAINT(165*ZF+A,114*ZF+B)
RETURN

128:
REM support co #128
U=5:U1$="SUPPORT COMPANY":U2$=""
LINE(10*ZF+A,100*ZF+B)-(360*ZF+A,120*ZF+B),1,BF
LINE(10*ZF+A,10*ZF+B)-(360*ZF+A,120*ZF+B), ,B
LINE(152*ZF+A,100*ZF+B)-(182*ZF+A,80*ZF+B)
LINE(212*ZF+A,100*ZF+B)-(182*ZF+A,80*ZF+B)
PAINT(165*ZF+A,94*ZF+B)
RETURN

129:
REM tragtier co #129
U=5:U1$="TRAGTIER KOMPANIE":U2$=""
LINE(70*ZF+A,55*ZF+B)-(300*ZF+A,55*ZF+B)
LINE(70*ZF+A,85*ZF+B)-(100*ZF+A,55*ZF+B)
LINE(100*ZF+A,55*ZF+B)-(130*ZF+A,85*ZF+B)
LINE(300*ZF+A,85*ZF+B)-(270*ZF+A,55*ZF+B)
LINE(270*ZF+A,55*ZF+B)-(240*ZF+A,85*ZF+B)
LINE(10*ZF+A,10*ZF+B)-(360*ZF+A,120*ZF+B), ,B
LINE(152*ZF+A,120*ZF+B)-(182*ZF+A,100*ZF+B)
LINE(212*ZF+A,120*ZF+B)-(182*ZF+A,100*ZF+B)
PAINT(165*ZF+A,114*ZF+B)
RETURN
```

```
130:
REM antitank co #130
U=5:U1$="ANTITANK COMPANY":U2$=""
LINE(10*ZF+A,120*ZF+B)-(185*ZF+A,10*ZF+B)
LINE(185*ZF+A,10*ZF+B)-(360*ZF+A,120*ZF+B)
LINE(145*ZF+A,60*ZF+B)-(185*ZF+A,40*ZF+B)
LINE(185*ZF+A,40*ZF+B)-(225*ZF+A,60*ZF+B)
LINE(145*ZF+A,80*ZF+B)-(185*ZF+A,60*ZF+B)
LINE(185*ZF+A,60*ZF+B)-(225*ZF+A,80*ZF+B)
LINE (10*ZF+A,10*ZF+B)-(360*ZF+A,120*ZF+B), ,B
LINE(152*ZF+A,120*ZF+B)-(182*ZF+A,100*ZF+B)
LINE(212*ZF+A,120*ZF+B)-(182*ZF+A,100*ZF+B)
PAINT(165*ZF+A,114*ZF+B)
RETURN

131:
REM airborne  mortar co #131
U=5:U1$="AIRBORNE MORTAR COMPANY":U2$=""
CIRCLE(130*ZF+A,120*ZF+B),50*ZF,1,15*PI/180,165*PI/180
CIRCLE(240*ZF+A,120*ZF+B),50*ZF,1,15*PI/180,165*PI/180
LINE(178*ZF+A,115*ZF+B)-(192*ZF+A,115*ZF+B)
CIRCLE(182*ZF+A,80*ZF+B),25*ZF
LINE(182*ZF+A,25*ZF+B)-(182*ZF+A,70*ZF+B)
LINE(152*ZF+A,50*ZF+B)-(212*ZF+A,50*ZF+B)
LINE(152*ZF+A,60*ZF+B)-(212*ZF+A,60*ZF+B)
LINE(152*ZF+A,35*ZF+B)-(182*ZF+A,25*ZF+B)
LINE(182*ZF+A,25*ZF+B)-(212*ZF+A,35*ZF+B)
RETURN

132:
REM airborne  engineer co #132
U=5:U1$="AIRBORNE ENGINEER COMPANY":U2$=""
CIRCLE(130*ZF+A,120*ZF+B),50*ZF,1,15*PI/180,165*PI/180
CIRCLE(240*ZF+A,120*ZF+B),50*ZF,1,15*PI/180,165*PI/180
LINE(178*ZF+A,115*ZF+B)-(192*ZF+A,115*ZF+B)
LINE (90*ZF+A,55*ZF+B)-(280*ZF+A,55*ZF+B)
LINE (90*ZF+A,55*ZF+B)-(90*ZF+A,75*ZF+B)
LINE (180*ZF+A,55*ZF+B)-(180*ZF+A,75*ZF+B)
LINE (280*ZF+A,55*ZF+B)-(280*ZF+A,75*ZF+B)
RETURN

133:
REM airborne medical co #133
U=5:U1$="AIRBORNE MEDICAL COMPANY":U2$=""
CIRCLE(130*ZF+A,120*ZF+B),50*ZF,1,15*PI/180,165*PI/180
CIRCLE(240*ZF+A,120*ZF+B),50*ZF,1,15*PI/180,165*PI/180
LINE(178*ZF+A,115*ZF+B)-(192*ZF+A,115*ZF+B)
LINE (10*ZF+A,65*ZF+B)-(360*ZF+A,65*ZF+B)
LINE(185*ZF+A,10*ZF+B)-(185*ZF+A,100*ZF+B)
RETURN
```

```
134:
REM airborne support co #134
U=5:U1$="AIRBORNE SUPPLY COMPANY":U2$=""
CIRCLE(130*ZF+A,90*ZF+B),50*ZF,1,15*PI/180,165*PI/180
CIRCLE(240*ZF+A,90*ZF+B),50*ZF,1,15*PI/180,165*PI/180
LINE(178*ZF+A,85*ZF+B)-(192*ZF+A,85*ZF+B)
LINE (10*ZF+A,100*ZF+B)-(360*ZF+A,120*ZF+B),1,BF
RETURN

135:
REM cavalry troop #135
U=5:U1$="CAVALRY TROOP":U2$=""
CIRCLE (90*ZF+A,65*ZF+B),48*ZF,1,90*PI/180,270*PI/180
CIRCLE(280*ZF+A,65*ZF+B),48*ZF,1,270*PI/180,90*PI/180
LINE(90*ZF+A,45*ZF+B)-(280*ZF+A,45*ZF+B)
LINE(90*ZF+A,85*ZF+B)-(280*ZF+A,85*ZF+B)
LINE(10*ZF+A,120*ZF+B)-(360*ZF+A,10*ZF+B)
CIRCLE(130*ZF+A,120*ZF+B),50*ZF,1,15*PI/180,165*PI/180
CIRCLE(240*ZF+A,120*ZF+B),50*ZF,1,15*PI/180,165*PI/180
LINE(178*ZF+A,115*ZF+B)-(192*ZF+A,115*ZF+B)
RETURN

136:
REM liaison squad #136
U=5:U1$="AVIATION LIAISON":U2$=""
CIRCLE(100*ZF+A,100*ZF+B),35*ZF,1,56*PI/180,304*PI/180
CIRCLE(270*ZF+A,100*ZF+B),35*ZF,1,236*PI/180,124*PI/180
LINE(120*ZF+A,87*ZF+B)-(250*ZF+A,113*ZF+B)
LINE(120*ZF+A,113*ZF+B)-(250*ZF+A,87*ZF+B)
PAINT(100*ZF+A,100*ZF+B),1:PAINT(270*ZF+A,100*ZF+B),1
LINE(140*ZF+A,75*ZF+B)-(230*ZF+A,75*ZF+B)
LINE(185*ZF+A,48*ZF+B)-(140*ZF+A,75*ZF+B)
LINE(185*ZF+A,48*ZF+B)-(230*ZF+A,75*ZF+B)
LINE(140*ZF+A,75*ZF+B)-(230*ZF+A,75*ZF+B)
LINE(185*ZF+A,48*ZF+B)-(140*ZF+A,75*ZF+B)
LINE(185*ZF+A,48*ZF+B)-(230*ZF+A,75*ZF+B)
CIRCLE(80*ZF+A,40*ZF+B),30*ZF
CIRCLE(290*ZF+A,40*ZF+B),30*ZF
LINE(110*ZF+A,40*ZF+B)-(260*ZF+A,40*ZF+B)
RETURN
```

```
137:
REM liaison squad #137
U=5:U1$="AVIATION LIAISON":U2$=""
CIRCLE(100*ZF+A,100*ZF+B),35*ZF,1,56*PI/180,304*PI/180
CIRCLE(270*ZF+A,100*ZF+B),35*ZF,1,236*PI/180,124*PI/180
LINE(120*ZF+A,87*ZF+B)-(250*ZF+A,113*ZF+B)
LINE(120*ZF+A,113*ZF+B)-(250*ZF+A,87*ZF+B)
PAINT(100*ZF+A,100*ZF+B),1:PAINT(270*ZF+A,100*ZF+B),1
LINE(140*ZF+A,75*ZF+B)-(230*ZF+A,75*ZF+B)
LINE(185*ZF+A,48*ZF+B)-(140*ZF+A,75*ZF+B)
LINE(185*ZF+A,48*ZF+B)-(230*ZF+A,75*ZF+B)
PAINT(185*ZF+A,60*ZF+B),1
CIRCLE(80*ZF+A,40*ZF+B),30*ZF
CIRCLE(290*ZF+A,40*ZF+B),30*ZF
LINE(110*ZF+A,40*ZF+B)-(260*ZF+A,40*ZF+B)
RETURN

138:
REM flugplatzkommandantur  #138
U=5:U1$="FIELD AIRPORT":U2$=""
CIRCLE(100*ZF+A,100*ZF+B),35*ZF,1,56*PI/180,304*PI/180
CIRCLE(270*ZF+A,100*ZF+B),35*ZF,1,236*PI/180,124*PI/180
LINE(120*ZF+A,87*ZF+B)-(250*ZF+A,113*ZF+B)
LINE(120*ZF+A,113*ZF+B)-(250*ZF+A,87*ZF+B)
PAINT(100*ZF+A,100*ZF+B),1:PAINT(270*ZF+A,100*ZF+B),1
LINE(182*ZF+A,15*ZF+B)-(122*ZF+A,35*ZF+B)
LINE(182*ZF+A,15*ZF+B)-(242*ZF+A,35*ZF+B)
LINE(150*ZF+A,40*ZF+B)-(150*ZF+A,80*ZF+B)
LINE(150*ZF+A,65*ZF+B)-(220*ZF+A,40*ZF+B)
LINE(162*ZF+A,60*ZF+B)-(220*ZF+A,80*ZF+B)
RETURN

139:
REM brigade maintenance co #139
U=5:U1$="MAINTENANCE COMPANY":U2$=""
CIRCLE (50*ZF+A,65*ZF+B),50*ZF,1,270*PI/180,90*PI/180
CIRCLE (320*ZF+A,65*ZF+B),50*ZF,1,90*PI/180,270*PI/180
LINE (100*ZF+A,65*ZF+B)-(270*ZF+A,65*ZF+B)
RETURN

140:
REM brigade supply co #140
U=5:U1$="SUPPLY COMPANY":U2$=""
LINE (10*ZF+A,100*ZF+B)-(360*ZF+A,100*ZF+B)
RETURN
```

```
141:
REM antitank co #141
U=5:U1$="ANTITANK COMPANY":U2$=""
LINE(10*ZF+A,120*ZF+B)-(185*ZF+A,10*ZF+B)
LINE(185*ZF+A,10*ZF+B)-(360*ZF+A,120*ZF+B)
LINE(145*ZF+A,60*ZF+B)-(185*ZF+A,40*ZF+B)
LINE(185*ZF+A,40*ZF+B)-(225*ZF+A,60*ZF+B)
LINE(145*ZF+A,80*ZF+B)-(185*ZF+A,60*ZF+B)
LINE(185*ZF+A,60*ZF+B)-(225*ZF+A,80*ZF+B)
RETURN

142:
REM engineer company #142
U=5:U1$="ENGINEER COMPANY":U2$=""
LINE (90*ZF+A,55*ZF+B)-(280*ZF+A,55*ZF+B)
LINE (90*ZF+A,55*ZF+B)-(90*ZF+A,75*ZF+B)
LINE (180*ZF+A,55*ZF+B)-(180*ZF+A,75*ZF+B)
LINE (280*ZF+A,55*ZF+B)-(280*ZF+A,75*ZF+B)
RETURN

143:
REM wehrgericht #143
U=0:U1$="wehrgericht":U2$=""
CIRCLE(240*ZF+A,85*ZF+B),40*ZF,1,PI,2*PI
CIRCLE(240*ZF+A,50*ZF+B),40*ZF,1,10/180*PI,PI
LINE(200*ZF+A,50*ZF+B)-(200*ZF+A,85*ZF+B)
LINE(280*ZF+A,65*ZF+B)-(280*ZF+A,85*ZF+B)
LINE(240*ZF+A,65*ZF+B)-(280*ZF+A,65*ZF+B)
LINE(50*ZF+A,33*ZF+B)-(80*ZF+A,103*ZF+B)
LINE(80*ZF+A,103*ZF+B)-(110*ZF+A,65*ZF+B)
LINE(110*ZF+A,65*ZF+B)-(140*ZF+A,103*ZF+B)
LINE(140*ZF+A,103*ZF+B)-(170*ZF+A,33*ZF+B)
RETURN

144:
REM hhq infantry brigade #144
U=5:U1$="HEADQUATERS AND HEADQUARTERS CO"
U2$="INFANTRY BRIGADE"
GOSUB u4
LINE(10*ZF+A,10*ZF+B)-(360*ZF+A,25*ZF+B),1,BF
LINE(170*ZF+A,36*ZF+B)-(190*ZF+A,42*ZF+B)
LINE(190*ZF+A,36*ZF+B)-(170*ZF+A,42*ZF+B)
RETURN
```

2. <u>Source Code Program INMILDAT</u>

```
REM program INMILDAT                               JUNE 1987
REM INMILDAT enables the user to define the distri-
REM bution of systems in a corps.
REM INMILDAT presents all available tactical signs and
REM prompts for the number of systems in the
REM respective unit.
DIM N(150),UC(150)
REM
REM *****************************************************
REM **              Variable Directory                 **
REM *****************************************************
REM ** A       : auxiliary variable for screen pos. **
REM ** B       : auxiliary variable for screen pos. **
REM ** E$      : auxiliary string variable            **
REM ** F1      : auxiliary variable                   **
REM ** N(x)    : number of systems in unit            **
REM ** P2      : auxiliary variable for screen pos. **
REM ** PI      : mathematical constant                **
REM ** Q       : unit code number                     **
REM ** U       : unit size code                       **
REM ** U1$     : unit description                     **
REM ** U2$     : additional description               **
REM ** UC(x)   : unit code number                     **
REM ** ZF      : zoom factor for graph                **
REM *****************************************************
REM
KEY OFF:SCREEN 2
A=140:B=10:F1=0:P2=1:PI=3.14153:X=0:ZF=1
FOR Q=21 TO 144
11:
  INCR X:UC(X)=Q
  CLS:GOSUB tacsign
  LOCATE 18,20:PRINT U1$
  LOCATE 19,20:PRINT U2$
  E$=""
  LOCATE 23,25
  INPUT "How many Systems in this Unit ";E$
  IF E$<>"" THEN N(X)=VAL(E$) ELSE DECR X
  LOCATE 24,31:INPUT "Entry ok ";E$
  LOCATE 22,20:PRINT "                                                "
  LOCATE 23,20:PRINT "                                                "
  IF E$="n" OR E$="N" THEN 11
NEXT Q
```

```
12:
CLS:LOCATE 5,25
INPUT "Do you want a Reserve ";E$
IF E$="N" OR E$="n" THEN 13
INCR X:LOCATE 7,25
INPUT "In per cent or absolute (p/a) ";E$
IF E$="P" OR E$="p" THEN UC(X)=198:LOCATE 10,15:_
   INPUT "Enter Percentage (e.g.: 25 ): ";N(X):_
   N(X)=N(X)/100:GOTO 14
IF E$="A" OR E$="a" THEN UC(X)=199:LOCATE 10,15:_
   INPUT "Enter Number : ";N(X):GOTO 14
GOTO 12

13:
INCR X:UC(X)=200:N(X)=99

14:
CLS:LOCATE 5,23
PRINT "To save Data, name File."
locate 7,20
PRINT "Default Filename is DISTRIBN."
LOCATE 11,10
PRINT "To accept default press ENTER key or";
INPUT " enter Filename ",F$
IF F$="" THEN F$="DISTRIBN"
OPEN "O",1,F$
WRITE#1,X
FOR A=1 TO X
   WRITE#1,UC(A),N(A)
NEXT A
CLOSE 1
CLS:LOCATE 20,30:PRINT "PROGRAM END"
END
REM **************************************************
```

```
REM *****  Subroutine to draw one Tactical Sign   *****
REM
REM necc. input: zoom factor            0 < ZF <= 1
REM ------------ horizl. start coord. 0 < A < 550
REM              vertcl. start coord. 0 < B < 150
REM              value for constant   PI=3.141593
REM              value for unit type  0 <= Q <= 144
REM
tacsign:
IF Q>144 THEN RETURN
IF Q>121 THEN companies2
IF Q>100 THEN companies1
IF Q>40 THEN battalions
IF Q=21 THEN GOSUB 21:GOTO box
IF Q=22 THEN GOSUB 22:GOTO box
IF Q=23 THEN GOSUB 23:GOTO box
IF Q=24 THEN GOSUB 24:GOTO box
IF Q=25 THEN GOSUB 25:GOTO box
IF Q=26 THEN GOSUB 26:GOTO box
IF Q=27 THEN GOSUB 27:GOTO box
IF Q=28 THEN GOSUB 28:GOTO box
IF Q=29 THEN GOSUB 29:GOTO box
IF Q=30 THEN GOSUB 30:GOTO box
IF Q=31 THEN INCR Q
IF Q=32 THEN INCR Q
IF Q=33 THEN INCR Q
IF Q=34 THEN INCR Q
IF Q=35 THEN INCR Q
IF Q=36 THEN INCR Q
IF Q=37 THEN INCR Q
IF Q=38 THEN INCR Q
IF Q=39 THEN INCR Q
IF Q=40 THEN INCR Q
battalions:
IF Q=41 THEN GOSUB 41:GOTO box
IF Q=42 THEN GOSUB 42:GOTO box
IF Q=43 THEN GOSUB 43:GOTO box
IF Q=44 THEN GOSUB 44:GOTO box
IF Q=45 THEN GOSUB 45:GOTO box
IF Q=46 THEN GOSUB 46:GOTO box
IF Q=47 THEN GOSUB 47:GOTO box
IF Q=48 THEN GOSUB 48:GOTO box
IF Q=49 THEN GOSUB 49:GOTO box
IF Q=50 THEN GOSUB 50:GOTO box
IF Q=51 THEN GOSUB 51:GOTO box
IF Q=52 THEN GOSUB 52:GOTO box
IF Q=53 THEN GOSUB 53:GOTO box
IF Q=54 THEN GOSUB 54:GOTO box
IF Q=55 THEN GOSUB 55:GOTO box
IF Q=56 THEN GOSUB 56:GOTO box
IF Q=57 THEN GOSUB 57:GOTO box
```

```
IF Q=58 THEN GOSUB 58:GOTO box
IF Q=59 THEN GOSUB 59:GOTO box
IF Q=60 THEN GOSUB 60:GOTO box
IF Q=61 THEN GOSUB 61:GOTO box
IF Q=62 THEN GOSUB 62:GOTO box
IF Q=63 THEN GOSUB 63:GOTO box
IF Q=64 THEN GOSUB 64:GOTO box
IF Q=65 THEN GOSUB 65:GOTO box
IF Q=66 THEN GOSUB 66:GOTO box
IF Q=67 THEN GOSUB 67:GOTO box
IF Q=68 THEN GOSUB 68:GOTO box
IF Q=69 THEN GOSUB 69:GOTO box
IF Q=70 THEN GOSUB 70:GOTO box
IF Q=71 THEN GOSUB 71:GOTO box
IF Q=72 THEN GOSUB 72:GOTO box
IF Q=73 THEN GOSUB 73:GOTO box
IF Q=74 THEN GOSUB 74:GOTO box
IF Q=75 THEN GOSUB 75:GOTO box
IF Q=76 THEN GOSUB 76:GOTO box
IF Q=77 THEN GOSUB 77:GOTO box
IF Q=78 THEN GOSUB 78:GOTO box
IF Q=79 THEN GOSUB 79:GOTO box
IF Q=80 THEN GOSUB 80:GOTO box
IF Q=81 THEN GOSUB 81:GOTO box
IF Q=82 THEN GOSUB 82:GOTO box
IF Q=83 THEN GOSUB 83:GOTO box
IF Q=84 THEN GOSUB 84:GOTO box
IF Q=85 THEN GOSUB 85:GOTO box
IF Q=86 THEN GOSUB 86:GOTO box
IF Q=87 THEN GOSUB 87:GOTO box
IF Q=88 THEN GOSUB 88:GOTO box
IF Q=89 THEN GOSUB 89:GOTO box
IF Q=90 THEN INCR Q
IF Q=91 THEN INCR Q
IF Q=92 THEN INCR Q
IF Q=93 THEN INCR Q
IF Q=94 THEN INCR Q
IF Q=95 THEN INCR Q
IF Q=96 THEN INCR Q
IF Q=97 THEN INCR Q
IF Q=98 THEN INCR Q
IF Q=99 THEN INCR Q
IF Q=100 THEN INCR Q
```

```
companies1:
IF Q=101 THEN GOSUB 101:GOTO box
IF Q=102 THEN GOSUB 102:GOTO box
IF Q=103 THEN GOSUB 103:GOTO box
IF Q=104 THEN GOSUB 104:GOTO box
IF Q=105 THEN GOSUB 105:GOTO box
IF Q=106 THEN GOSUB 106:GOTO box
IF Q=107 THEN GOSUB 107:GOTO box
IF Q=108 THEN GOSUB 108:GOTO box
IF Q=109 THEN GOSUB 109:GOTO box
IF Q=110 THEN GOSUB 110:GOTO box
IF Q=111 THEN GOSUB 111:GOTO box
IF Q=112 THEN GOSUB 112:GOTO box
IF Q=113 THEN GOSUB 113:GOTO box
IF Q=114 THEN GOSUB 114:GOTO box
IF Q=115 THEN GOSUB 115:GOTO box
IF Q=116 THEN GOSUB 116:GOTO box
IF Q=117 THEN GOSUB 117:GOTO box
IF Q=118 THEN GOSUB 118:GOTO box
IF Q=119 THEN GOSUB 119:GOTO box
IF Q=120 THEN GOSUB 120:GOTO box
IF Q=121 THEN GOSUB 121:GOTO box
companies2:
IF Q=122 THEN GOSUB 122:GOTO box
IF Q=123 THEN GOSUB 123:GOTO box
IF Q=124 THEN GOSUB 124:GOTO box
IF Q=125 THEN GOSUB 125:GOTO box
IF Q=126 THEN GOSUB 126:GOTO box
IF Q=127 THEN GOSUB 127:GOTO box
IF Q=128 THEN GOSUB 128:GOTO box
IF Q=129 THEN GOSUB 129:GOTO box
IF Q=130 THEN GOSUB 130:GOTO box
IF Q=131 THEN GOSUB 131:GOTO box
IF Q=132 THEN GOSUB 132:GOTO box
IF Q=133 THEN GOSUB 133:GOTO box
IF Q=134 THEN GOSUB 134:GOTO box
IF Q=135 THEN GOSUB 135:GOTO box
IF Q=136 THEN GOSUB 136:GOTO box
IF Q=137 THEN GOSUB 137:GOTO box
IF Q=138 THEN GOSUB 138:GOTO box
IF Q=139 THEN GOSUB 139:GOTO box
IF Q=140 THEN GOSUB 140:GOTO box
IF Q=141 THEN GOSUB 141:GOTO box
IF Q=142 THEN GOSUB 142:GOTO box
IF Q=143 THEN GOSUB 143:GOTO box
IF Q=144 THEN GOSUB 144:GOTO box
RETURN
$segment
REM ************************************************
```

```
box:
LINE (10*ZF+A,10*ZF+B)-(360*ZF+A,120*ZF+B), ,B
IF U=1 THEN LINE(155*ZF+A,1*ZF+B)-(175*ZF+A,8*ZF+B):_
            LINE(175*ZF+A,1*ZF+B)-(155*ZF+A,8*ZF+B):_
            LINE(195*ZF+A,1*ZF+B)-(215*ZF+A,8*ZF+B):_
            LINE(215*ZF+A,1*ZF+B)-(195*ZF+A,8*ZF+B)
IF U=2 THEN LINE(175*ZF+A,1*ZF+B)-(195*ZF+A,8*ZF+B):_
            LINE(195*ZF+A,1*ZF+B)-(175*ZF+A,8*ZF+B)
IF U=3 THEN LINE(165*ZF+A,1*ZF+B)-(165*ZF+A,7*ZF+B):_
            LINE(185*ZF+A,1*ZF+B)-(185*ZF+A,7*ZF+B):_
            LINE (205*ZF+A,1*ZF+B)-(205*ZF+A,7*ZF+B)
IF U=4 THEN LINE(175*ZF+A,1*ZF+B)-(175*ZF+A,7*ZF+B):_
            LINE (195*ZF+A,1*ZF+B)-(195*ZF+A,7*ZF+B)
IF U=5 THEN LINE(185*ZF+A,1*ZF+B)-(185*ZF+A,7*ZF+B)
IF P1=1 THEN LOCATE B/8+17*ZF,A/8+54*ZF
IF E$="1" THEN 120 ELSE PRINT E$
120:
RETURN
REM *******************************************************

u4:
LINE(10*ZF+A,10*ZF+B)-(360*ZF+A,120*ZF+B)
LINE(10*ZF+A,120*ZF+B)-(360*ZF+A,10*ZF+B)
RETURN
REM *******************************************************
```

```
21:
REM anti aircraft regiment (corps) #21
U1$="ANTI AIRCRAFT DEFENCE REGIMENT":U2$="":U=3
CIRCLE (207*ZF+A,50*ZF+B),_
       140*ZF,1,138*PI/180,223*PI/180
CIRCLE (186*ZF+A,175*ZF+B),_
       220*ZF,1,39*PI/180,142*PI/180
CIRCLE (140*ZF+A,105*ZF+B),_
       14*ZF,1,90*PI/180,270*PI/180
CIRCLE (220*ZF+A,105*ZF+B),_
       14*ZF,1,270*PI/180,90*PI/180
LINE(140*ZF+A,98*ZF+B)-(220*ZF+A,98*ZF+B)
LINE (140*ZF+A,112*ZF+B)-(220*ZF+A,112*ZF+B)
LINE (76*ZF+A,45*ZF+B)-(206*ZF+A,45*ZF+B)
LINE (156*ZF+A,65*ZF+B)-(286*ZF+A,65*ZF+B)
LINE (206*ZF+A,45*ZF+B)-(156*ZF+A,65*ZF+B)
LINE (182*ZF+A,101*ZF+B)-(168*ZF+A,104*ZF+B)
LINE (182*ZF+A,101*ZF+B)-(196*ZF+A,104*ZF+B)
LINE (182*ZF+A,106*ZF+B)-(168*ZF+A,109*ZF+B)
LINE (182*ZF+A,106*ZF+B)-(196*ZF+A,109*ZF+B)
LINE(130*ZF+A,15*ZF+B)-(150*ZF+A,25*ZF+B)
LINE(150*ZF+A,15*ZF+B)-(130*ZF+A,25*ZF+B)
LINE(175*ZF+A,15*ZF+B)-(195*ZF+A,25*ZF+B)
LINE(195*ZF+A,15*ZF+B)-(175*ZF+A,25*ZF+B)
LINE(220*ZF+A,15*ZF+B)-(240*ZF+A,25*ZF+B)
LINE(220*ZF+A,25*ZF+B)-(240*ZF+A,15*ZF+B)
RETURN

22:
REM anti aircraft regiment (divisional) #22
U1$="ANTI AIRCRAFT DEFENCE REGIMENT":U2$="":U=3
CIRCLE (207*ZF+A,50*ZF+B),_
       140*ZF,1,137*PI/180,223*PI/180
CIRCLE (186*ZF+A,175*ZF+B),_
       220*ZF,1,39*PI/180,142*PI/180
CIRCLE (140*ZF+A,105*ZF+B),_
       14*ZF,1,90*PI/180,270*PI/180
CIRCLE (220*ZF+A,105*ZF+B),_
       14*ZF,1,270*PI/180,90*PI/180
LINE(140*ZF+A,98*ZF+B)-(220*ZF+A,98*ZF+B)
LINE (140*ZF+A,112*ZF+B)-(220*ZF+A,112*ZF+B)
LINE (76*ZF+A,45*ZF+B)-(206*ZF+A,45*ZF+B)
LINE (156*ZF+A,65*ZF+B)-(286*ZF+A,65*ZF+B)
LINE (206*ZF+A,45*ZF+B)-(156*ZF+A,65*ZF+B)
CIRCLE(182*ZF+A,105*ZF+B),10*ZF,3
PAINT(186*ZF+A,105*ZF+B),1,3
LINE(150*ZF+A,15*ZF+B)-(170*ZF+A,25*ZF+B)
LINE(170*ZF+A,15*ZF+B)-(150*ZF+A,25*ZF+B)
LINE(200*ZF+A,15*ZF+B)-(220*ZF+A,25*ZF+B)
LINE(200*ZF+A,25*ZF+B)-(220*ZF+A,15*ZF+B)
RETURN
```

```
23:
REM aviation regiment # 23
U=3:U1$="AVIATION REGIMENT":U2$=""
CIRCLE(100*ZF+A,100*ZF+B),_
      35*ZF,1,56*PI/180,304*PI/180
CIRCLE(270*ZF+A,100*ZF+B),_
      35*ZF,1,236*PI/180,124*PI/180
LINE(120*ZF+A,87*ZF+B)-(250*ZF+A,113*ZF+B)
LINE(120*ZF+A,113*ZF+B)-(250*ZF+A,87*ZF+B)
PAINT(100*ZF+A,100*ZF+B),1
PAINT(270*ZF+A,100*ZF+B),1
CIRCLE(185*ZF+A,50*ZF+B),70*ZF
LINE (135*ZF+A,31*ZF+B)-(237*ZF+A,69*ZF+B)
LINE(136*ZF+A,69*ZF+B)-(235*ZF+A,31*ZF+B)
LINE (185*ZF+A,22*ZF+B)-(185*ZF+A,78*ZF+B)
LINE (115*ZF+A,50*ZF+B)-(255*ZF+A,50*ZF+B)
LINE (30*ZF+A,10*ZF+B)-(30*ZF+A,120*ZF+B)
RETURN

24:
REM aviation regiment #24
U=3:U1$="AVIATION REGIMENT":U2$=""
CIRCLE(100*ZF+A,100*ZF+B),_
      35*ZF,1,56*PI/180,304*PI/180
CIRCLE(270*ZF+A,100*ZF+B),_
      35*ZF,1,236*PI/180,124*PI/180
LINE(120*ZF+A,87*ZF+B)-(250*ZF+A,113*ZF+B)
LINE(120*ZF+A,113*ZF+B)-(250*ZF+A,87*ZF+B)
PAINT(100*ZF+A,100*ZF+B),1
PAINT(270*ZF+A,100*ZF+B),1
CIRCLE(180*ZF+A,50*ZF+B),70*ZF
LINE (130*ZF+A,31*ZF+B)-(232*ZF+A,69*ZF+B)
LINE(131*ZF+A,69*ZF+B)-(230*ZF+A,31*ZF+B)
LINE (180*ZF+A,22*ZF+B)-(180*ZF+A,78*ZF+B)
LINE (110*ZF+A,50*ZF+B)-(250*ZF+A,50*ZF+B)
RETURN
```

```
25:
REM combat aviation regiment #25
U=3:U1$="COMBAT AVIATION REGIMENT":U2$=""
CIRCLE(100*ZF+A,100*ZF+B),_
      35*ZF,1,56*PI/180,304*PI/180
CIRCLE(270*ZF+A,100*ZF+B),_
      35*ZF,1,236*PI/180,124*PI/180
LINE(120*ZF+A,87*ZF+B)-(250*ZF+A,113*ZF+B)
LINE(120*ZF+A,113*ZF+B)-(250*ZF+A,87*ZF+B)
LINE(10*ZF+A,120*ZF+B)-(180*ZF+A,10*ZF+B)
LINE(360*ZF+A,120*ZF+B)-(180*ZF+A,10*ZF+B)
PAINT(100*ZF+A,100*ZF+B),1
PAINT(270*ZF+A,100*ZF+B),1
LINE(145*ZF+A,60*ZF+B)-(185*ZF+A,40*ZF+B)
LINE(185*ZF+A,40*ZF+B)-(225*ZF+A,60*ZF+B)
LINE(145*ZF+A,80*ZF+B)-(185*ZF+A,60*ZF+B)
LINE(185*ZF+A,60*ZF+B)-(225*ZF+A,80*ZF+B)
RETURN

26:
REM armored cavalry regiment #26
U=3:U1$="ARMORED CAVALRY REGIMENT":U2$=""
CIRCLE (90*ZF+A,65*ZF+B),_
       48*ZF,1,90*PI/180,270*PI/180
CIRCLE(280*ZF+A,65*ZF+B),_
       48*ZF,1,270*PI/180,90*PI/180
LINE(90*ZF+A,45*ZF+B)-(280*ZF+A,45*ZF+B)
LINE(90*ZF+A,85*ZF+B)-(280*ZF+A,85*ZF+B)
LINE(10*ZF+A,120*ZF+B)-(360*ZF+A,10*ZF+B)
RETURN

27:
REM military intelligence regiment #27
U=3:U1$="MILITARY INTELLIGENCE REGIMENT":U2$=""
LINE(220*ZF+A,30*ZF+B)-(260*ZF+A,30*ZF+B)
LINE(220*ZF+A,100*ZF+B)-(260*ZF+A,100*ZF+B)
LINE(80*ZF+A,30*ZF+B)-(80*ZF+A,100*ZF+B)
LINE(160*ZF+A,30*ZF+B)-(160*ZF+A,100*ZF+B)
LINE(240*ZF+A,30*ZF+B)-(240*ZF+A,100*ZF+B)
LINE (80*ZF+A,30*ZF+B)-(120*ZF+A,60*ZF+B)
LINE(120*ZF+A,60*ZF+B)-(160*ZF+A,30*ZF+B)
RETURN
```

```
28:
REM field artillery regiment #28
U=3:U1$="FIELD ARTY REGIMENT":U2$=""
CIRCLE (185*ZF+A,65*ZF+B),30*ZF,3
PAINT(185*ZF+A,65*ZF+B),1,3
LINE(155*ZF+A,16*ZF+B)-(175*ZF+A,22*ZF+B)
LINE(175*ZF+A,16*ZF+B)-(155*ZF+A,22*ZF+B)
LINE(195*ZF+A,16*ZF+B)-(215*ZF+A,22*ZF+B)
LINE(215*ZF+A,16*ZF+B)-(195*ZF+A,22*ZF+B)
RETURN

29:
REM cav reg air atk #29
U=3:U1$="CAVALRY REGIMENT AIR ATTACK":U2$=""
CIRCLE(100*ZF+A,65*ZF+B),_
      35*ZF,1,56*PI/180,304*PI/180
CIRCLE(270*ZF+A,65*ZF+B),_
      35*ZF,1,236*PI/180,124*PI/180
LINE(120*ZF+A,52*ZF+B)-(250*ZF+A,78*ZF+B)
LINE(120*ZF+A,78*ZF+B)-(250*ZF+A,52*ZF+B)
LINE(10*ZF+A,120*ZF+B)-(360*ZF+A,10*ZF+B)
PAINT(100*ZF+A,65*ZF+B),1
PAINT(270*ZF+A,65*ZF+B),1
RETURN

30:
REM  aviation gp #30
U=0:U1$="AVIATION GROUP":U2$=""
CIRCLE(100*ZF+A,65*ZF+B),_
      35*ZF,1,56*PI/180,304*PI/180
CIRCLE(270*ZF+A,65*ZF+B),_
      35*ZF,1,236*PI/180,124*PI/180
LINE(120*ZF+A,52*ZF+B)-(250*ZF+A,78*ZF+B)
LINE(120*ZF+A,78*ZF+B)-(250*ZF+A,52*ZF+B)
PAINT(100*ZF+A,65*ZF+B),1
PAINT(270*ZF+A,65*ZF+B),1
LINE(140*ZF+A,25*ZF+B)-(185*ZF+A,55*ZF+B)
LINE(185*ZF+A,55*ZF+B)-(230*ZF+A,25*ZF+B)
LINE(185*ZF+A,55*ZF+B)-(185*ZF+A,95*ZF+B)
LINE(140*ZF+A,95*ZF+B)-(230*ZF+A,95*ZF+B)
RETURN

41:
REM motorized infantery (divisional) #41
U=4:U1$="MOTORIZED INFANRTY BATTALION":U2$=""
GOSUB u4
RETURN
```

```
42:
REM airborne battalion #42
U=4:U1$="AIRBORNE BATTALION":U2$=""
CIRCLE(130*ZF+A,120*ZF+B),_
      50*ZF,1,15*PI/180,165*PI/180
CIRCLE(240*ZF+A,120*ZF+B),_
      50*ZF,1,15*PI/180,165*PI/180
LINE(178*ZF+A,115*ZF+B)-(192*ZF+A,115*ZF+B)
GOSUB u4
RETURN

43:
REM mountaineer battalion #43
U=4:U1$="INFANTRY BATTALION"
U2$="MOUNTAINEER DIVISION"
GOSUB u4
LINE (10*ZF+A,10*ZF+B)-(360*ZF+A,120*ZF+B), ,B
LINE(152*ZF+A,120*ZF+B)-(182*ZF+A,100*ZF+B)
LINE(212*ZF+A,120*ZF+B)-(182*ZF+A,100*ZF+B)
PAINT(165*ZF+A,114*ZF+B)
RETURN

44:
REM mechanized infantery battalion (armored brig) #44
U=4:U1$="MECHANIZED INFANTRY BATTALION"
U2$="ARMORED BRIGADE"
CIRCLE (90*ZF+A,65*ZF+B),_
      48*ZF,1,90*PI/180,270*PI/180
CIRCLE(280*ZF+A,65*ZF+B),_
      48*ZF,1,270*PI/180,90*PI/180
LINE(90*ZF+A,45*ZF+B)-(280*ZF+A,45*ZF+B)
LINE(90*ZF+A,85*ZF+B)-(280*ZF+A,85*ZF+B)
GOSUB u4
RETURN

45:
REM mechanized infantery battalion (infant. brig) #45
U=4:U1$="MECHANIZED INFANTRY BATTALION"
U2$="MECHANIZED INFANTRY BRIGADE"
CIRCLE (90*ZF+A,65*ZF+B),_
      48*ZF,1,90*PI/180,270*PI/180
CIRCLE(280*ZF+A,65*ZF+B),_
      48*ZF,1,270*PI/180,90*PI/180
LINE(90*ZF+A,45*ZF+B)-(280*ZF+A,45*ZF+B)
LINE(90*ZF+A,85*ZF+B)-(280*ZF+A,85*ZF+B)
GOSUB u4
LINE(10*ZF+A,10*ZF+B)-(35*ZF+A,120*ZF+B),,BF
RETURN
```

```
46:
REM motorized infantery (divisional) #46
U=4:U1$="MOTORIZED INFANRTY BATTALION":U2$=""
LINE (165*ZF+A,90*ZF+B)-(205*ZF+A,110*ZF+B)
LINE (165*ZF+A,110*ZF+B)-(205*ZF+A,90*ZF+B)
GOSUB u4
RETURN

47:
REM motorized infantery (aaslt div) #47
U=4:U1$="MOTORIZED INFANRTY BATTALION":U2$=""
GOSUB u4
LINE(140*ZF+A,35*ZF+B)-(185*ZF+A,65*ZF+B)
LINE(185*ZF+A,65*ZF+B)-(230*ZF+A,35*ZF+B)
RETURN

48:
REM airborne infantry battalion #48
U=4:U1$="AIRBORNE INFANTRY BATTALION":U2$=""
CIRCLE(185*ZF+A,80*ZF+B),80*ZF
LINE(50*ZF+A,60*ZF+B)-(280*ZF+A,115*ZF+B),0,BF
LINE(10*ZF+A,10*ZF+B)-(127*ZF+A,47*ZF+B)
LINE(360*ZF+A,10*ZF+B)-(233*ZF+A,47*ZF+B)
LINE(10*ZF+A,120*ZF+B)-(127*ZF+A,73*ZF+B)
LINE(360*ZF+A,120*ZF+B)-(233*ZF+A,73*ZF+B)
LINE (122*ZF+A,60*ZF+B)-(185*ZF+A,90*ZF+B)
LINE(185*ZF+A,90*ZF+B)-(245*ZF+A,60*ZF+B)
LINE(122*ZF+A,60*ZF+B)-(245*ZF+A,60*ZF+B)
RETURN

49:
REM armored reconnaissance battalion  #49
U=4:U1$="ARMORED CAVALRY BATTALION ":U2$=""
CIRCLE (90*ZF+A,65*ZF+B),_
       48*ZF,1,90*PI/180,270*PI/180
CIRCLE(280*ZF+A,65*ZF+B),_
       48*ZF,1,270*PI/180,90*PI/180
LINE (90*ZF+A,45*ZF+B)-(280*ZF+A,45*ZF+B)
LINE (90*ZF+A,85*ZF+B)-(280*ZF+A,85*ZF+B)
LINE (10*ZF+A,120*ZF+B)-(360*ZF+A,10*ZF+B)
RETURN
```

```
50:
REM air cavalry squadron #50
U=4:U1$="AIR CAVALRY SQUADRON":U2$=""
CIRCLE(100*ZF+A,65*ZF+B),_
      35*ZF,1,56*PI/180,304*PI/180
CIRCLE(270*ZF+A,65*ZF+B),_
      35*ZF,1,236*PI/180,124*PI/180
LINE(120*ZF+A,52*ZF+B)-(250*ZF+A,78*ZF+B)
LINE(120*ZF+A,78*ZF+B)-(250*ZF+A,52*ZF+B)
LINE(10*ZF+A,120*ZF+B)-(360*ZF+A,10*ZF+B)
PAINT(100*ZF+A,65*ZF+B),1
PAINT(270*ZF+A,65*ZF+B),1
RETURN

51:
REM armored battalion (mech. inf. brig) #51
U=4:U1$="ARMORED BATTALION"
U2$="MECHANIZED INFANTRY BRIGADE"
CIRCLE (90*ZF+A,65*ZF+B),_
       48*ZF,1,90*PI/180,270*PI/180
CIRCLE(280*ZF+A,65*ZF+B),_
       48*ZF,1,270*PI/180,90*PI/180
LINE(90*ZF+A,45*ZF+B)-(280*ZF+A,45*ZF+B)
LINE(90*ZF+A,85*ZF+B)-(280*ZF+A,85*ZF+B)
RETURN

52:
REM armored  battalion (armored brig) #52
U=4:U1$="ARMORED BATTALION":U2$="ARMORED BRIGADE"
CIRCLE (90*ZF+A,65*ZF+B),_
       48*ZF,1,90*PI/180,270*PI/180
CIRCLE(280*ZF+A,65*ZF+B),_
       48*ZF,1,270*PI/180,90*PI/180
LINE (90*ZF+A,45*ZF+B)-(280*ZF+A,45*ZF+B)
LINE (90*ZF+A,85*ZF+B)-(280*ZF+A,85*ZF+B)
LINE (10*ZF+A,10*ZF+B)-(30*ZF+A,120*ZF+B),,BF
RETURN
```

```
53:
REM armored  battalion (airborne brig) #53
U=4:U1$="ARMORED BATTALION":U2$="AIRBORNE BRIGADE"
CIRCLE (90*ZF+A,65*ZF+B),_
       48*ZF,1,90*PI/180,270*PI/180
CIRCLE(280*ZF+A,65*ZF+B),_
       48*ZF,1,270*PI/180,90*PI/180
LINE (90*ZF+A,45*ZF+B)-(280*ZF+A,45*ZF+B)
LINE (90*ZF+A,85*ZF+B)-(280*ZF+A,85*ZF+B)
CIRCLE(130*ZF+A,120*ZF+B),_
      50*ZF,1,15*PI/180,165*PI/180
CIRCLE(240*ZF+A,120*ZF+B),_
      50*ZF,1,15*PI/180,165*PI/180
LINE(178*ZF+A,115*ZF+B)-(192*ZF+A,115*ZF+B)
RETURN

54:
REM attack helicopter battalion #54
U=4:U1$="ATTACK HELICOPTER BATTALION":U2$=""
CIRCLE (90*ZF+A,65*ZF+B),_
       60*ZF,1,90*PI/180,270*PI/180
CIRCLE(280*ZF+A,65*ZF+B),_
       60*ZF,1,270*PI/180,90*PI/180
LINE(90*ZF+A,40*ZF+B)-(280*ZF+A,40*ZF+B)
LINE(90*ZF+A,90*ZF+B)-(280*ZF+A,90*ZF+B)
CIRCLE(100*ZF+A,65*ZF+B),_
      35*ZF,1,56*PI/180,304*PI/180
CIRCLE(270*ZF+A,65*ZF+B),_
      35*ZF,1,236*PI/180,124*PI/180
LINE(120*ZF+A,52*ZF+B)-(250*ZF+A,78*ZF+B)
LINE(120*ZF+A,78*ZF+B)-(250*ZF+A,52*ZF+B)
PAINT(100*ZF+A,65*ZF+B),1
PAINT(270*ZF+A,65*ZF+B),1
RETURN

55:
REM field artillery battalion #55
U=4:U1$="FIELD ARTY BATTALION":U2$=""
CIRCLE (185*ZF+A,65*ZF+B),30*ZF,3
PAINT(185*ZF+A,65*ZF+B),1,3
RETURN

56:
REM field artilley battalion #56
U=4:U1$="FIELD ARTILLERY BATTALION":U2$=""
CIRCLE (185*ZF+A,65*ZF+B),30*ZF,3
PAINT(185*ZF+A,65*ZF+B)
LINE (10*ZF+A,10*ZF+B)-(360*ZF+A,120*ZF+B)
LINE (10*ZF+A,120*ZF+B)-(360*ZF+A,10*ZF+B)
RETURN
```

```
57:
REM field artilley battalion #57
U=4:U1$="FIELD ARTILLERY BATTALION":U2$=""
CIRCLE (185*ZF+A,65*ZF+B),30*ZF,3
PAINT(185*ZF+A,65*ZF+B),1,3
LINE (10*ZF+A,10*ZF+B)-(360*ZF+A,120*ZF+B)
LINE (10*ZF+A,120*ZF+B)-(360*ZF+A,10*ZF+B)
CIRCLE(130*ZF+A,120*ZF+B),50*ZF,1,15*PI/180,165*PI/180
CIRCLE(240*ZF+A,120*ZF+B),50*ZF,1,15*PI/180,165*PI/180
LINE(178*ZF+A,115*ZF+B)-(192*ZF+A,115*ZF+B)
RETURN

58:
REM reconnaissance battalion #58
U=4:U1$="RECONNAISSANCE BATTALION ":U2$=""
CIRCLE (185*ZF+A,65*ZF+B),30*ZF,3
PAINT(185*ZF+A,65*ZF+B),1,3
LINE (10*ZF+A,120*ZF+B)-(360*ZF+A,10*ZF+B)
RETURN

59:
REM field arty battalion #59
U=4:U1$="FIELD ARTILLERY BATTALION":U2$=""
CIRCLE (90*ZF+A,65*ZF+B),48*ZF,1,90*PI/180,270*PI/180
CIRCLE(280*ZF+A,65*ZF+B),48*ZF,1,270*PI/180,90*PI/180
LINE(90*ZF+A,45*ZF+B)-(280*ZF+A,45*ZF+B)
LINE(90*ZF+A,85*ZF+B)-(280*ZF+A,85*ZF+B)
CIRCLE (180*ZF+A,65*ZF+B),20*ZF,3
PAINT (180*ZF+A,65*ZF+B),3,3
RETURN

60:
REM missile artillery battalion #60
U=4:U1$="MISSILE ARTY BATTALION":U2$=""
LINE(180*ZF+A,35*ZF+B)-(140*ZF+A,60*ZF+B)
LINE(180*ZF+A,35*ZF+B)-(220*ZF+A,60*ZF+B)
LINE(180*ZF+A,45*ZF+B)-(140*ZF+A,70*ZF+B)
LINE(180*ZF+A,45*ZF+B)-(220*ZF+A,70*ZF+B)
CIRCLE (180*ZF+A,80*ZF+B),25*ZF,3
PAINT(180*ZF+A,80*ZF+B),1,3
RETURN
```

```
61:
REM pershing battalion #61
U=4:U1$="MISSILE ARTY BATTALION":U2$="PERSHING"
CIRCLE(360*ZF+A,100*ZF+B),_
      220*ZF,1,142.5*PI/180,180*PI/180
CIRCLE(5*ZF+A,100*ZF+B),220*ZF,1,0,37.5*PI/180
LINE(140*ZF+A,100*ZF+B)-(140*ZF+A,120*ZF+B)
LINE(225*ZF+A,100*ZF+B)-(225*ZF+A,120*ZF+B)
LINE(143*ZF+A,86*ZF+B)-(222*ZF+A,86*ZF+B)
LINE(140*ZF+A,106*ZF+B)-(225*ZF+A,106*ZF+B)
CIRCLE (180*ZF+A,96*ZF+B),9*ZF,3
PAINT(180*ZF+A,96*ZF+B),1,3
RETURN

62:
REM mountaineer artillery battalion #62
U=4:U1$="FIELD ARTILLERY BATTALION"
U2$="MOUNTAINEER DIVISION"
CIRCLE (185*ZF+A,65*ZF+B),30*ZF,3
PAINT(185*ZF+A,65*ZF+B),1,3
LINE (10*ZF+A,10*ZF+B)-(360*ZF+A,120*ZF+B), ,B
LINE(152*ZF+A,120*ZF+B)-(182*ZF+A,100*ZF+B)
LINE(212*ZF+A,120*ZF+B)-(182*ZF+A,100*ZF+B)
PAINT(165*ZF+A,114*ZF+B)
RETURN

63:
REM  nbc  battalion  #63
U=4:U1$="NBC BATTALION":U2$=""
CIRCLE (80*ZF+A,50*ZF+B),30*ZF,3
PAINT(80*ZF+A,50*ZF+B),1,3
CIRCLE (290*ZF+A,50*ZF+B),30*ZF,3
PAINT(290*ZF+A,50*ZF+B),1,3
LINE (290*ZF+A,100*ZF+B)-(98*ZF+A,40*ZF+B)
LINE (80*ZF+A,100*ZF+B)-(270*ZF+A,40*ZF+B)
RETURN

64:
REM combat aviation bn #64
U=4:U1$="COMBAT AVIATION BATTALION":U2$=""
CIRCLE(100*ZF+A,65*ZF+B),_
      35*ZF,1,56*PI/180,304*PI/180
CIRCLE(270*ZF+A,65*ZF+B),_
      35*ZF,1,236*PI/180,124*PI/180
LINE(120*ZF+A,52*ZF+B)-(250*ZF+A,78*ZF+B)
LINE(120*ZF+A,78*ZF+B)-(250*ZF+A,52*ZF+B)
PAINT(100*ZF+A,65*ZF+B),1
PAINT(270*ZF+A,65*ZF+B),1
RETURN
```

```
65:
REM military intelligence battalion #65
U=4:U1$="MILITARY INTELLIGENCE BATTALION":U2$=""
LINE(220*ZF+A,30*ZF+B)-(260*ZF+A,30*ZF+B)
LINE(220*ZF+A,100*ZF+B)-(260*ZF+A,100*ZF+B)
LINE(80*ZF+A,30*ZF+B)-(80*ZF+A,100*ZF+B)
LINE(160*ZF+A,30*ZF+B)-(160*ZF+A,100*ZF+B)
LINE(240*ZF+A,30*ZF+B)-(240*ZF+A,100*ZF+B)
LINE (80*ZF+A,30*ZF+B)-(120*ZF+A,60*ZF+B)
LINE(120*ZF+A,60*ZF+B)-(160*ZF+A,30*ZF+B)
RETURN

66:
REM medical battalion #66
U=4:U1$="MEDICAL BATTALION":U2$=""
LINE (10*ZF+A,65*ZF+B)-(360*ZF+A,65*ZF+B)
LINE (185*ZF+A,10*ZF+B)-(185*ZF+A,120*ZF+B)
RETURN

67:
REM  medical evacuation battalion #67
U=4:U1$="MEDICAL EVACUATION BATTALION":U2$=""
CIRCLE(180*ZF+A,65*ZF+B),70*ZF
LINE (130*ZF+A,46*ZF+B)-(232*ZF+A,84*ZF+B)
LINE(131*ZF+A,85*ZF+B)-(230*ZF+A,46*ZF+B)
LINE (185*ZF+A,10*ZF+B)-(185*ZF+A,120*ZF+B)
LINE (10*ZF+A,65*ZF+B)-(360*ZF+A,65*ZF+B)
LINE(10*ZF+A,10*ZF+B)-(40*ZF+A,120*ZF+B),,BF
RETURN

68:
REM  maintainance  battalion  #68
U=4:U1$="MAINTAINANCE BATTALION":U2$=""
CIRCLE (50*ZF+A,65*ZF+B),_
       50*ZF,1,270*PI/180,90*PI/180
CIRCLE (320*ZF+A,65*ZF+B),_
       50*ZF,1,90*PI/180,270*PI/180
LINE (100*ZF+A,65*ZF+B)-(270*ZF+A,65*ZF+B)
RETURN
```

```
69:
REM  maintainance  battalion  (electronic mn.) #69
U=4:U1$="MAINTAINANCE BATTALION"
U2$="(ELECTRONIC MAINTAINANCE)"
CIRCLE (50*ZF+A,85*ZF+B),_
       50*ZF,1,270*PI/180,90*PI/180
CIRCLE (320*ZF+A,85*ZF+B),_
       50*ZF,1,90*PI/180,270*PI/180
LINE (100*ZF+A,85*ZF+B)-(270*ZF+A,85*ZF+B)
LINE(140*ZF+A,30*ZF+B)-(140*ZF+A,50*ZF+B)
LINE (230*ZF+A,30*ZF+B)-(230*ZF+A,50*ZF+B)
LINE(230*ZF+A,30*ZF+B)-(140*ZF+A,40*ZF+B)
LINE(230*ZF+A,50*ZF+B)-(140*ZF+A,40*ZF+B)
PAINT(160*ZF+A,40*ZF+B)
LINE(57*ZF+A,40*ZF+B)-(310*ZF+A,40*ZF+B)
RETURN

70:
REM  supply battalion  #70
U=4:U1$="SUPPLY BATTALION":U2$=""
LINE (10*ZF+A,100*ZF+B)-(360*ZF+A,100*ZF+B)
RETURN

71:
REM  supply battalion for special weapons #71
U=4:U1$="SUPPLY BATTALION (SW)":U2$=""
LINE (10*ZF+A,100*ZF+B)-(360*ZF+A,100*ZF+B)
LINE(235*ZF+A,55*ZF+B)-(220*ZF+A,75*ZF+B)
CIRCLE(150*ZF+A,60*ZF+B),_
      13*ZF,1,45*PI/180,270*PI/180
CIRCLE(150*ZF+A,70*ZF+B),_
      13*ZF,1,225*PI/180,90*PI/180
LINE(185*ZF+A,55*ZF+B)-(200*ZF+A,75*ZF+B)
LINE(200*ZF+A,75*ZF+B)-(210*ZF+A,60*ZF+B)
LINE(210*ZF+A,60*ZF+B)-(220*ZF+A,75*ZF+B)
RETURN

72:
REM  transportation  battalion #72
U=4:U1$="TRANSPORTATION BATTALION":U2$=""
CIRCLE(180*ZF+A,65*ZF+B),70*ZF
LINE (130*ZF+A,46*ZF+B)-(232*ZF+A,84*ZF+B)
LINE(131*ZF+A,85*ZF+B)-(230*ZF+A,46*ZF+B)
LINE (180*ZF+A,37*ZF+B)-(180*ZF+A,93*ZF+B)
LINE (110*ZF+A,65*ZF+B)-(250*ZF+A,65*ZF+B)
RETURN
```

```
73:
REM military police  battalion #73
U=4:U1$="MILITARY POLICE BATTALION":U2$=""
CIRCLE(255*ZF+A,42*ZF+B),_
      29*ZF,1,270*PI/180,90*PI/180
LINE(240*ZF+A,30*ZF+B)-(255*ZF+A,30*ZF+B)
LINE(240*ZF+A,54*ZF+B)-(255*ZF+A,54*ZF+B)
LINE(80*ZF+A,30*ZF+B)-(80*ZF+A,100*ZF+B)
LINE(160*ZF+A,30*ZF+B)-(160*ZF+A,100*ZF+B)
LINE(240*ZF+A,30*ZF+B)-(240*ZF+A,100*ZF+B)
LINE (80*ZF+A,30*ZF+B)-(120*ZF+A,60*ZF+B)
LINE(120*ZF+A,60*ZF+B)-(160*ZF+A,30*ZF+B)
RETURN

74:
REM ada battalion #74
U=4:U1$="ADA BATTALION":U2$="CORPS ADA COMMAND"
CIRCLE (186*ZF+A,175*ZF+B),_
       220*ZF,1,39*PI/180,142*PI/180
RETURN

75:
REM ada battalion #75
U=4:U1$="ADA BATTALION":U2$="AIRBORNE DIVISION"
CIRCLE (186*ZF+A,175*ZF+B),_
       220*ZF,1,39*PI/180,142*PI/180
CIRCLE(130*ZF+A,120*ZF+B),_
       50*ZF,1,15*PI/180,165*PI/180
CIRCLE(240*ZF+A,120*ZF+B),_
       50*ZF,1,15*PI/180,165*PI/180
LINE(178*ZF+A,115*ZF+B)-(192*ZF+A,115*ZF+B)
RETURN

76:
REM  ew battalion #76
U=4:U1$="EW BATTALION":U2$=""
LINE (10*ZF+A,120*ZF+B)-(185*ZF+A,50*ZF+B)
LINE(185*ZF+A,80*ZF+B)-(360*ZF+A,10*ZF+B)
LINE(185*ZF+A,80*ZF+B)-(185*ZF+A,50*ZF+B)
RETURN

77:
REM  signal battalion #77
U=4:U1$="SIGNAL BATTALION":U2$=""
LINE (10*ZF+A,10*ZF+B)-(185*ZF+A,80*ZF+B)
LINE(185*ZF+A,50*ZF+B)-(360*ZF+A,120*ZF+B)
LINE(185*ZF+A,80*ZF+B)-(185*ZF+A,50*ZF+B)
RETURN
```

```
78:
REM  signal construction battalion #78
U=4:U1$="SIGNAL CONSTRUCTION BATTALION":U2$=""
LINE (70*ZF+A,42*ZF+B)-(300*ZF+A,42*ZF+B)
LINE(70*ZF+A,53*ZF+B)-(300*ZF+A,53*ZF+B)
LINE(120*ZF+A,65*ZF+B)-(250*ZF+A,65*ZF+B)
LINE(182*ZF+A,65*ZF+B)-(182*ZF+A,91*ZF+B)
RETURN

79:
REM signal construction battalion #79
U=4:U1$="SIGNAL CONSTRUCTION BATTALION":U2$=""
CIRCLE(185*ZF+A,B-20*ZF),_
      160*ZF,1,230*PI/180,310*PI/180
LINE (130*ZF+A,45*ZF+B)-(130*ZF+A,100*ZF+B)
LINE (225*ZF+A,45*ZF+B)-(225*ZF+A,85*ZF+B)
LINE (225*ZF+A,85*ZF+B)-(250*ZF+A,65*ZF+B)
LINE(250*ZF+A,65*ZF+B)-(250*ZF+A,100*ZF+B)
RETURN

80:
REM feld ersatz battailon #80
U=4:U1$="FELD ERSATZ BATTALION":U2$=""
LINE(50*ZF+A,100*ZF+B)-(50*ZF+A,30*ZF+B)
LINE(50*ZF+A,30*ZF+B)-(110*ZF+A,30*ZF+B)
LINE(50*ZF+A,65*ZF+B)-(100*ZF+A,65*ZF+B)
LINE(120*ZF+A,100*ZF+B)-(120*ZF+A,30*ZF+B)
LINE(120*ZF+A,30*ZF+B)-(180*ZF+A,30*ZF+B)
LINE(120*ZF+A,65*ZF+B)-(170*ZF+A,65*ZF+B)
LINE(120*ZF+A,100*ZF+B)-(180*ZF+A,100*ZF+B)
CIRCLE(225*ZF+A,77*ZF+B),25*ZF,1,0,PI
LINE(200*ZF+A,65*ZF+B)-(200*ZF+A,100*ZF+B)
CIRCLE(290*ZF+A,75*ZF+B),20*ZF,1,0,PI*3/2
CIRCLE(290*ZF+A,91*ZF+B),_
      20*ZF,1,180*PI/180,90*PI/180
RETURN

81:
REM engineer battalion #81
U=4:U1$="ENGINEER BATTALION":U2$=""
LINE (90*ZF+A,55*ZF+B)-(280*ZF+A,55*ZF+B)
LINE (90*ZF+A,55*ZF+B)-(90*ZF+A,75*ZF+B)
LINE (180*ZF+A,55*ZF+B)-(180*ZF+A,75*ZF+B)
LINE (280*ZF+A,55*ZF+B)-(280*ZF+A,75*ZF+B)
RETURN
```

```
82:
REM armored engineer battalion #82
U=4:U1$="ARMORED ENGINEER BATTALION":U2$=""
CIRCLE (90*ZF+A,65*ZF+B),_
       48*ZF,1,90*PI/180,270*PI/180
CIRCLE (280*ZF+A,65*ZF+B),_
       48*ZF,1,270*PI/180,90*PI/180
LINE(90*ZF+A,45*ZF+B)-(280*ZF+A,45*ZF+B)
LINE(90*ZF+A,85*ZF+B)-(280*ZF+A,85*ZF+B)
LINE (90*ZF+A,55*ZF+B)-(280*ZF+A,55*ZF+B)
LINE (90*ZF+A,55*ZF+B)-(90*ZF+A,75*ZF+B)
LINE (180*ZF+A,55*ZF+B)-(180*ZF+A,75*ZF+B)
LINE (280*ZF+A,55*ZF+B)-(280*ZF+A,75*ZF+B)
RETURN

83:
REM engineer battalion #83
U=4:U1$="ENGINEERING FERRY BATTALION":U2$=""
CIRCLE(57*ZF+A,50*ZF+B),32*ZF,1,0,PI
CIRCLE(121*ZF+A,50*ZF+B),32*ZF,1,PI,0
CIRCLE(185*ZF+A,50*ZF+B),32*ZF,1,0,PI
CIRCLE(249*ZF+A,50*ZF+B),32*ZF,1,PI,0
CIRCLE(313*ZF+A,50*ZF+B),32*ZF,1,0,PI
LINE (80*ZF+A,80*ZF+B)-(290*ZF+A,80*ZF+B)
LINE (80*ZF+A,80*ZF+B)-(80*ZF+A,99*ZF+B)
LINE (180*ZF+A,80*ZF+B)-(180*ZF+A,99*ZF+B)
LINE (290*ZF+A,80*ZF+B)-(290*ZF+A,99*ZF+B)
RETURN

84:
REM engineer battalion (bridge layers) #84
U=4:U1$="ENGINEER BATTALION":U2$="BRIDGE LAYERS"
LINE (90*ZF+A,80*ZF+B)-(280*ZF+A,80*ZF+B)
LINE (90*ZF+A,80*ZF+B)-(90*ZF+A,99*ZF+B)
LINE (185*ZF+A,80*ZF+B)-(185*ZF+A,99*ZF+B)
LINE (280*ZF+A,80*ZF+B)-(280*ZF+A,99*ZF+B)
LINE (110*ZF+A,55*ZF+B)-(260*ZF+A,55*ZF+B)
LINE (110*ZF+A,40*ZF+B)-(260*ZF+A,40*ZF+B)
LINE (90*ZF+A,65*ZF+B)-(110*ZF+A,55*ZF+B)
LINE (260*ZF+A,40*ZF+B)-(280*ZF+A,30*ZF+B)
LINE (90*ZF+A,30*ZF+B)-(110*ZF+A,40*ZF+B)
LINE (260*ZF+A,55*ZF+B)-(280*ZF+A,65*ZF+B)
RETURN
```

```
85:
REM support battalion #85
U=4:U1$="SUPPORT BATTALION":U2$="AIRBORNE DIVISION"
CIRCLE(130*ZF+A,120*ZF+B),_
      50*ZF,1,15*PI/180,165*PI/180
CIRCLE(240*ZF+A,120*ZF+B),_
      50*ZF,1,15*PI/180,165*PI/180
LINE(178*ZF+A,115*ZF+B)-(192*ZF+A,115*ZF+B)
CIRCLE(150*ZF+A,60*ZF+B),_
      13*ZF,1,45*PI/180,270*PI/180
CIRCLE(150*ZF+A,70*ZF+B),_
      13*ZF,1,225*PI/180,90*PI/180
CIRCLE(180*ZF+A,70*ZF+B),_
      13*ZF,1,225*PI/180,135*PI/180
LINE(172*ZF+A,66*ZF+B)-(172*ZF+A,85*ZF+B)
CIRCLE(213*ZF+A,70*ZF+B),_
      13*ZF,1,180*PI/180,340*PI/180
LINE(200*ZF+A,55*ZF+B)-(200*ZF+A,70*ZF+B)
LINE(192*ZF+A,63*ZF+B)-(210*ZF+A,63*ZF+B)
RETURN

86:
REM support battalion #86
U=4:U1$="SUPPORT BATTALION":U2$=""
CIRCLE(150*ZF+A,60*ZF+B),_
      13*ZF,1,45*PI/180,270*PI/180
CIRCLE(150*ZF+A,70*ZF+B),_
      13*ZF,1,225*PI/180,90*PI/180
CIRCLE(180*ZF+A,70*ZF+B),_
      13*ZF,1,225*PI/180,135*PI/180
LINE(172*ZF+A,66*ZF+B)-(172*ZF+A,85*ZF+B)
CIRCLE(213*ZF+A,70*ZF+B),_
      13*ZF,1,180*PI/180,340*PI/180
LINE(200*ZF+A,55*ZF+B)-(200*ZF+A,70*ZF+B)
LINE(192*ZF+A,63*ZF+B)-(210*ZF+A,63*ZF+B)
RETURN

87:
REM  supply and transportation  battalion #87
U=4:U1$="SUPPLY AND TRANSPORTATION BATTALION":U2$=""
CIRCLE(180*ZF+A,65*ZF+B),70*ZF
LINE (130*ZF+A,46*ZF+B)-(232*ZF+A,84*ZF+B)
LINE(131*ZF+A,85*ZF+B)-(230*ZF+A,46*ZF+B)
LINE (180*ZF+A,37*ZF+B)-(180*ZF+A,93*ZF+B)
LINE (110*ZF+A,65*ZF+B)-(250*ZF+A,65*ZF+B)
LINE (10*ZF+A,100*ZF+B)-(360*ZF+A,100*ZF+B)
RETURN
```

```
88:
REM field arty battalion (mech division) #88
U1$="FIELD ARTILLERY BATTALION":U2$="MECH. DIVISION"
U=4:CIRCLE (185*ZF+A,65*ZF+B),30*ZF,3
PAINT(185*ZF+A,65*ZF+B)
CIRCLE(90*ZF+A,65*ZF+B),_
      48*ZF,1,90*PI/180,270*PI/180
CIRCLE(280*ZF+A,65*ZF+B),_
      48*ZF,1,270*PI/180,90*PI/180
LINE(90*ZF+A,45*ZF+B)-(280*ZF+A,45*ZF+B)
LINE(90*ZF+A,85*ZF+B)-(280*ZF+A,85*ZF+B)
LINE(10*ZF+A,10*ZF+B)-(360*ZF+A,120*ZF+B)
LINE(10*ZF+A,120*ZF+B)-(360*ZF+A,10*ZF+B)
RETURN

89:
REM field arty battalion (AASLT division) #89
U1$="FIELD ARTILLERY BATTALION":U2$="AASLT DIVISION"
U=4:CIRCLE (185*ZF+A,65*ZF+B),30*ZF,3
PAINT(185*ZF+A,65*ZF+B)
LINE(10*ZF+A,10*ZF+B)-(360*ZF+A,120*ZF+B)
LINE(10*ZF+A,120*ZF+B)-(360*ZF+A,10*ZF+B)
LINE(145*ZF+A,20*ZF+B)-(185*ZF+A,40*ZF+B)
LINE(225*ZF+A,20*ZF+B)-(185*ZF+A,40*ZF+B)
LINE(185*ZF+A,40*ZF+B)-(185*ZF+A,100*ZF+B)
LINE(145*ZF+A,100*ZF+B)-(225*ZF+A,100*ZF+B)
RETURN

101:
REM military band #101
U=5:U1$="MILITARY BAND":U2$=""
CIRCLE(25*ZF+A,60*ZF+B),80*ZF,1,0,PI/2
CIRCLE(185*ZF+A,60*ZF+B),80*ZF,1,PI,0
CIRCLE(345*ZF+A,60*ZF+B),80*ZF,1,PI/2,PI
LINE(185*ZF+A,30*ZF+B)-(185*ZF+A,110*ZF+B)
LINE(140*ZF+A,30*ZF+B)-(140*ZF+A,70*ZF+B)
LINE(230*ZF+A,30*ZF+B)-(230*ZF+A,70*ZF+B)
LINE(100*ZF+A,50*ZF+B)-(270*ZF+A,50*ZF+B)
RETURN
```

```
102:
REM military police co #102
U=5:U1$="MILITARY POLICE COMPANY":U2$=""
CIRCLE(255*ZF+A,42*ZF+B),_
      29*ZF,1,270*PI/180,90*PI/180
LINE(240*ZF+A,30*ZF+B)-(255*ZF+A,30*ZF+B)
LINE(240*ZF+A,54*ZF+B)-(255*ZF+A,54*ZF+B)
LINE(80*ZF+A,30*ZF+B)-(80*ZF+A,100*ZF+B)
LINE(160*ZF+A,30*ZF+B)-(160*ZF+A,100*ZF+B)
LINE(240*ZF+A,30*ZF+B)-(240*ZF+A,100*ZF+B)
LINE (80*ZF+A,30*ZF+B)-(120*ZF+A,60*ZF+B)
LINE(120*ZF+A,60*ZF+B)-(160*ZF+A,30*ZF+B)
RETURN

103:
REM chemical co #103
U=5:U1$="CHEMICAL COMPANY":U2$=""
CIRCLE (80*ZF+A,50*ZF+B),30*ZF,3
PAINT(80*ZF+A,50*ZF+B),1,3
CIRCLE (290*ZF+A,50*ZF+B),30*ZF,3
PAINT(290*ZF+A,50*ZF+B),1,3
LINE (290*ZF+A,100*ZF+B)-(98*ZF+A,40*ZF+B)
LINE (80*ZF+A,100*ZF+B)-(270*ZF+A,40*ZF+B)
RETURN

104:
REM cavalry troop #104
U=5:U1$="CAVALRY TROOP":U2$=""
CIRCLE(90*ZF+A,65*ZF+B),_
      48*ZF,1,90*PI/180,270*PI/180
CIRCLE(280*ZF+A,65*ZF+B),_
      48*ZF,1,270*PI/180,90*PI/180
LINE (90*ZF+A,45*ZF+B)-(280*ZF+A,45*ZF+B)
LINE(90*ZF+A,85*ZF+B)-(280*ZF+A,85*ZF+B)
LINE(10*ZF+A,120*ZF+B)-(360*ZF+A,10*ZF+B)
RETURN

105:
REM military intelligence company #105
U=5:U1$="MILITARY INTELLIGENCE COMPANY":U2$=""
LINE(220*ZF+A,30*ZF+B)-(260*ZF+A,30*ZF+B)
LINE(220*ZF+A,100*ZF+B)-(260*ZF+A,100*ZF+B)
LINE(80*ZF+A,30*ZF+B)-(80*ZF+A,100*ZF+B)
LINE(160*ZF+A,30*ZF+B)-(160*ZF+A,100*ZF+B)
LINE(240*ZF+A,30*ZF+B)-(240*ZF+A,100*ZF+B)
LINE (80*ZF+A,30*ZF+B)-(120*ZF+A,60*ZF+B)
LINE(120*ZF+A,60*ZF+B)-(160*ZF+A,30*ZF+B)
RETURN
```

```
106:
REM hhq corps command #106
U=5:U1$="HEADQUATERS AND HEADQUARTERS CO"
U2$="CORPS COMMAND"
LINE(10*ZF+A,10*ZF+B)-(360*ZF+A,30*ZF+B),1,BF
LINE(140*ZF+A,36*ZF+B)-(160*ZF+A,43*ZF+B)
LINE(160*ZF+A,36*ZF+B)-(140*ZF+A,43*ZF+B)
LINE(175*ZF+A,36*ZF+B)-(195*ZF+A,43*ZF+B)
LINE(195*ZF+A,36*ZF+B)-(175*ZF+A,43*ZF+B)
LINE(210*ZF+A,36*ZF+B)-(230*ZF+A,43*ZF+B)
LINE(210*ZF+A,43*ZF+B)-(230*ZF+A,36*ZF+B)
RETURN

107:
REM hhq corps maintainance command #107
U=5:U1$="HEADQUATERS AND HEADQUARTERS CO"
U2$="CORPS MAINTAINANCE COMMAND"
CIRCLE(50*ZF+A,65*ZF+B),_
      50*ZF,1,270*PI/180,90*PI/180
CIRCLE(320*ZF+A,65*ZF+B),_
      50*ZF,1,90*PI/180,270*PI/180
LINE (100*ZF+A,65*ZF+B)-(270*ZF+A,65*ZF+B)
LINE(10*ZF+A,10*ZF+B)-(360*ZF+A,25*ZF+B),1,BF
LINE(175*ZF+A,28*ZF+B)-(195*ZF+A,37*ZF+B)
LINE(195*ZF+A,28*ZF+B)-(175*ZF+A,37*ZF+B)
RETURN

108:
REM corps signal command #108
U=5:U1$="HEADQUATERS AND HEADQUARTERS CO"
U2$="CORPS SIGNAL COMMAND"
LINE (10*ZF+A,10*ZF+B)-(185*ZF+A,80*ZF+B)
LINE(185*ZF+A,50*ZF+B)-(360*ZF+A,120*ZF+B)
LINE(185*ZF+A,80*ZF+B)-(185*ZF+A,50*ZF+B)
LINE(10*ZF+A,10*ZF+B)-(360*ZF+A,25*ZF+B),1,BF
LINE(175*ZF+A,28*ZF+B)-(195*ZF+A,37*ZF+B)
LINE(195*ZF+A,28*ZF+B)-(175*ZF+A,37*ZF+B)
RETURN

109:
REM hhq arty command #109
U=5:U1$="HEADQUATERS AND HEADQUARTERS CO"
U2$="ARTILLERY COMMAND"
CIRCLE (185*ZF+A,65*ZF+B),30*ZF,3
PAINT(185*ZF+A,65*ZF+B),1,3
LINE(10*ZF+A,10*ZF+B)-(360*ZF+A,25*ZF+B),1,BF
LINE(175*ZF+A,28*ZF+B)-(195*ZF+A,37*ZF+B)
LINE(195*ZF+A,28*ZF+B)-(175*ZF+A,37*ZF+B)
RETURN
```

```
110:
REM hhq corps ada command #110
U=5:U1$="HEADQUATERS AND HEADQUARTERS CO"
U2$="CORPS ADA COMMAND"
CIRCLE(186*ZF+A,175*ZF+B),_
      220*ZF,1,39*PI/180,142*PI/180
LINE(10*ZF+A,10*ZF+B)-(360*ZF+A,25*ZF+B),1,BF
LINE(175*ZF+A,28*ZF+B)-(195*ZF+A,37*ZF+B)
LINE(195*ZF+A,28*ZF+B)-(175*ZF+A,37*ZF+B)
RETURN

111:
REM hhq corps engineer command #111
U=5:U1$="HEADQUATERS AND HEADQUARTERS CO"
U2$="CORPS ENGINEER COMMAND"
LINE (90*ZF+A,55*ZF+B)-(280*ZF+A,55*ZF+B)
LINE (90*ZF+A,55*ZF+B)-(90*ZF+A,75*ZF+B)
LINE (180*ZF+A,55*ZF+B)-(180*ZF+A,75*ZF+B)
LINE (280*ZF+A,55*ZF+B)-(280*ZF+A,75*ZF+B)
LINE(10*ZF+A,10*ZF+B)-(360*ZF+A,25*ZF+B),1,BF
LINE(175*ZF+A,28*ZF+B)-(195*ZF+A,37*ZF+B)
LINE(195*ZF+A,28*ZF+B)-(175*ZF+A,37*ZF+B)
RETURN

112:
REM hhq corps medical command #112
U=5:U1$="HEADQUATERS AND HEADQUARTERS CO"
U2$="CORPS MEDICAL COMMAND"
LINE(10*ZF+A,65*ZF+B)-(360*ZF+A,65*ZF+B)
LINE(185*ZF+A,50*ZF+B)-(185*ZF+A,120*ZF+B)
LINE(10*ZF+A,10*ZF+B)-(360*ZF+A,25*ZF+B),1,BF
LINE(175*ZF+A,28*ZF+B)-(195*ZF+A,37*ZF+B)
LINE(195*ZF+A,28*ZF+B)-(175*ZF+A,37*ZF+B)
RETURN

113:
REM hhq corps supply command #113
U=5:U1$="HEADQUATERS AND HEADQUARTERS CO"
U2$="CORPS SUPPLY COMMAND"
LINE (10*ZF+A,100*ZF+B)-(360*ZF+A,100*ZF+B)
LINE(10*ZF+A,10*ZF+B)-(360*ZF+A,25*ZF+B),1,BF
LINE(175*ZF+A,28*ZF+B)-(195*ZF+A,37*ZF+B)
LINE(195*ZF+A,28*ZF+B)-(175*ZF+A,37*ZF+B)
RETURN
```

```
114:
REM hhq aviation brigade #114
U=5:U1$="HEADQUATERS AND HEADQUARTERS CO"
U2$="AVIATION BRIGADE"
CIRCLE(100*ZF+A,65*ZF+B),_
      35*ZF,1,56*PI/180,304*PI/180
CIRCLE(270*ZF+A,65*ZF+B),_
      35*ZF,1,236*PI/180,124*PI/180
LINE(120*ZF+A,52*ZF+B)-(250*ZF+A,78*ZF+B)
LINE(120*ZF+A,78*ZF+B)-(250*ZF+A,52*ZF+B)
PAINT(100*ZF+A,65*ZF+B),1
PAINT(270*ZF+A,65*ZF+B),1
LINE(10*ZF+A,10*ZF+B)-(360*ZF+A,25*ZF+B),1,BF
LINE(175*ZF+A,28*ZF+B)-(195*ZF+A,37*ZF+B)
LINE(195*ZF+A,28*ZF+B)-(175*ZF+A,37*ZF+B)
RETURN

115:
REM hhq division command #115
U=5:U1$="HEADQUATERS AND HEADQUARTERS CO"
U2$="DIVISION COMMAND"
LINE(10*ZF+A,10*ZF+B)-(360*ZF+A,25*ZF+B),1,BF
LINE(150*ZF+A,36*ZF+B)-(170*ZF+A,42*ZF+B)
LINE(170*ZF+A,36*ZF+B)-(150*ZF+A,42*ZF+B)
LINE(200*ZF+A,36*ZF+B)-(220*ZF+A,42*ZF+B)
LINE(220*ZF+A,36*ZF+B)-(200*ZF+A,42*ZF+B)
RETURN

116:
REM hhq mountaineer brigade #116
U=5:U1$="HEADQUATERS AND HEADQUARTERS CO"
U2$="MOUNTAINEER BRIGADE"
LINE (10*ZF+A,10*ZF+B)-(360*ZF+A,120*ZF+B)
LINE (10*ZF+A,120*ZF+B)-(360*ZF+A,10*ZF+B)
LINE (10*ZF+A,10*ZF+B)-(360*ZF+A,120*ZF+B),,B
LINE(10*ZF+A,10*ZF+B)-(360*ZF+A,25*ZF+B),1,BF
LINE(152*ZF+A,120*ZF+B)-(182*ZF+A,100*ZF+B)
LINE(212*ZF+A,120*ZF+B)-(182*ZF+A,100*ZF+B)
PAINT(165*ZF+A,114*ZF+B)
LINE(175*ZF+A,28*ZF+B)-(195*ZF+A,37*ZF+B)
LINE(195*ZF+A,28*ZF+B)-(175*ZF+A,37*ZF+B)
RETURN
```

```
117:
REM hhq airborne brigade #117
U=5:U1$="HEADQUATERS AND HEADQUARTERS CO"
U2$="AIRBORNE BRIGADE"
CIRCLE(130*ZF+A,120*ZF+B),_
      50*ZF,1,15*PI/180,165*PI/180
CIRCLE(240*ZF+A,120*ZF+B),_
      50*ZF,1,15*PI/180,165*PI/180
LINE(178*ZF+A,115*ZF+B)-(192*ZF+A,115*ZF+B)
LINE(10*ZF+A,10*ZF+B)-(360*ZF+A,25*ZF+B),1,BF
LINE(175*ZF+A,28*ZF+B)-(195*ZF+A,37*ZF+B)
LINE(195*ZF+A,28*ZF+B)-(175*ZF+A,37*ZF+B)
GOSUB u4
RETURN

118:
REM hhq mech brigade #118
U=5:U1$="HEADQUATERS AND HEADQUARTERS CO"
U2$="MECH BRIGADE"
CIRCLE(90*ZF+A,70*ZF+B),_
      48*ZF,1,90*PI/180,270*PI/180
CIRCLE(280*ZF+A,70*ZF+B),_
      48*ZF,1,270*PI/180,90*PI/180
LINE (90*ZF+A,50*ZF+B)-(280*ZF+A,50*ZF+B)
LINE (90*ZF+A,90*ZF+B)-(280*ZF+A,90*ZF+B)
GOSUB u4
LINE(10*ZF+A,10*ZF+B)-(360*ZF+A,25*ZF+B),1,BF
LINE(175*ZF+A,28*ZF+B)-(195*ZF+A,37*ZF+B)
LINE(195*ZF+A,28*ZF+B)-(175*ZF+A,37*ZF+B)
RETURN

119:
REM hhq  armd brigade #119
U=5:U1$="HEADQUATERS AND HEADQUARTERS CO"
U2$="ARMORED BRIGADE"
CIRCLE(90*ZF+A,65*ZF+B),_
      48*ZF,1,90*PI/180,270*PI/180
CIRCLE(280*ZF+A,65*ZF+B),_
      48*ZF,1,270*PI/180,90*PI/180
LINE (90*ZF+A,45*ZF+B)-(280*ZF+A,45*ZF+B)
LINE (90*ZF+A,85*ZF+B)-(280*ZF+A,85*ZF+B)
LINE(10*ZF+A,10*ZF+B)-(360*ZF+A,25*ZF+B),1,BF
LINE(175*ZF+A,28*ZF+B)-(195*ZF+A,37*ZF+B)
LINE(195*ZF+A,28*ZF+B)-(175*ZF+A,37*ZF+B)
RETURN
```

```
120:
REM hhq air cavalry combat brigade #120
U=5:U1$="HHQ AIR CAVALRY COMBAT BRIGADE":U2$=""
CIRCLE(100*ZF+A,65*ZF+B),_
      35*ZF,1,56*PI/180,304*PI/180
CIRCLE(270*ZF+A,65*ZF+B),_
      35*ZF,1,236*PI/180,124*PI/180
LINE(120*ZF+A,52*ZF+B)-(250*ZF+A,78*ZF+B)
LINE(120*ZF+A,78*ZF+B)-(250*ZF+A,52*ZF+B)
LINE(10*ZF+A,120*ZF+B)-(360*ZF+A,10*ZF+B)
PAINT(100*ZF+A,65*ZF+B),1
PAINT(270*ZF+A,65*ZF+B),1
LINE(10*ZF+A,10*ZF+B)-(360*ZF+A,30*ZF+B),1,BF
LINE(170*ZF+A,36*ZF+B)-(190*ZF+A,42*ZF+B)
LINE(190*ZF+A,36*ZF+B)-(170*ZF+A,42*ZF+B)
RETURN

121:
REM  ew company #121
U=5:U1$="EW COMPANY":U2$=""
LINE (10*ZF+A,120*ZF+B)-(185*ZF+A,50*ZF+B)
LINE(185*ZF+A,80*ZF+B)-(360*ZF+A,10*ZF+B)
LINE(185*ZF+A,80*ZF+B)-(185*ZF+A,50*ZF+B)
RETURN

122:
REM  signal company #122
U=5:U1$="SIGNAL COMPANY":U2$=""
LINE (10*ZF+A,10*ZF+B)-(185*ZF+A,80*ZF+B)
LINE(185*ZF+A,50*ZF+B)-(360*ZF+A,120*ZF+B)
LINE(185*ZF+A,80*ZF+B)-(185*ZF+A,50*ZF+B)
RETURN

123:
REM fernspaeh co #123
U=5:U1$="FERNSPAEH KOMPANIE":U2$=""
LINE(10*ZF+A,120*ZF+B)-(360*ZF+A,10*ZF+B)
LINE(50*ZF+A,65*ZF+B)-(320*ZF+A,65*ZF+B)
LINE(280*ZF+A,45*ZF+B)-(320*ZF+A,65*ZF+B)
LINE(280*ZF+A,85*ZF+B)-(320*ZF+A,65*ZF+B)
RETURN
```

```
124:
REM armored engineer company #124
U1$="ARMORED ENGINEER COMPANY":U2$=""
CIRCLE(90*ZF+A,65*ZF+B),_
      48*ZF,1,90*PI/180,270*PI/180
CIRCLE(280*ZF+A,65*ZF+B),_
      48*ZF,1,270*PI/180,90*PI/180
LINE(90*ZF+A,45*ZF+B)-(280*ZF+A,45*ZF+B)
LINE(90*ZF+A,85*ZF+B)-(280*ZF+A,85*ZF+B)
LINE (90*ZF+A,55*ZF+B)-(280*ZF+A,55*ZF+B)
LINE (90*ZF+A,55*ZF+B)-(90*ZF+A,75*ZF+B)
LINE (180*ZF+A,55*ZF+B)-(180*ZF+A,75*ZF+B)
LINE (280*ZF+A,55*ZF+B)-(280*ZF+A,75*ZF+B)
RETURN

125:
REM adjudant general #125.
U=5:U1$="ADJUDANT GENERAL":U2$=""
LINE(70*ZF+A,100*ZF+B)-(120*ZF+A,30*ZF+B)
LINE(120*ZF+A,30*ZF+B)-(170*ZF+A,100*ZF+B)
LINE(95*ZF+A,65*ZF+B)-(145*ZF+A,65*ZF+B)
CIRCLE(240*ZF+A,85*ZF+B),40*ZF,1,PI,2*PI
CIRCLE(240*ZF+A,50*ZF+B),40*ZF,1,10/180*PI,PI
LINE(200*ZF+A,50*ZF+B)-(200*ZF+A,85*ZF+B)
LINE(280*ZF+A,65*ZF+B)-(280*ZF+A,85*ZF+B)
LINE(240*ZF+A,65*ZF+B)-(280*ZF+A,65*ZF+B)
RETURN

126:
REM engineer co #126
U=5:U1$="ENGINEER COMPANY":U2$=""
LINE (90*ZF+A,55*ZF+B)-(280*ZF+A,55*ZF+B)
LINE (90*ZF+A,55*ZF+B)-(90*ZF+A,75*ZF+B)
LINE (180*ZF+A,55*ZF+B)-(180*ZF+A,75*ZF+B)
LINE (280*ZF+A,55*ZF+B)-(280*ZF+A,75*ZF+B)
LINE (10*ZF+A,10*ZF+B)-(360*ZF+A,120*ZF+B),,B
LINE(152*ZF+A,120*ZF+B)-(182*ZF+A,100*ZF+B)
LINE(212*ZF+A,120*ZF+B)-(182*ZF+A,100*ZF+B)
PAINT(165*ZF+A,114*ZF+B)
RETURN
```

```
127:
REM chemical co #127
U=5:U1$="CHEMICAL COMPANY":U2$=""
CIRCLE (80*ZF+A,50*ZF+B),30*ZF,3
PAINT(80*ZF+A,50*ZF+B),1,3
CIRCLE (290*ZF+A,50*ZF+B),30*ZF,3
PAINT(290*ZF+A,50*ZF+B),1,3
LINE (290*ZF+A,100*ZF+B)-(98*ZF+A,40*ZF+B)
LINE (80*ZF+A,100*ZF+B)-(270*ZF+A,40*ZF+B)
LINE (10*ZF+A,10*ZF+B)-(360*ZF+A,120*ZF+B),,B
LINE(152*ZF+A,120*ZF+B)-(182*ZF+A,100*ZF+B)
LINE(212*ZF+A,120*ZF+B)-(182*ZF+A,100*ZF+B)
PAINT(165*ZF+A,114*ZF+B)
RETURN

128:
REM support co #128
U=5:U1$="SUPPORT COMPANY":U2$=""
LINE(10*ZF+A,100*ZF+B)-(360*ZF+A,120*ZF+B),1,BF
LINE(10*ZF+A,10*ZF+B)-(360*ZF+A,120*ZF+B),,B
LINE(152*ZF+A,100*ZF+B)-(182*ZF+A,80*ZF+B)
LINE(212*ZF+A,100*ZF+B)-(182*ZF+A,80*ZF+B)
PAINT(165*ZF+A,94*ZF+B)
RETURN

129:
REM tragtier co #129
U=5:U1$="TRAGTIER KOMPANIE":U2$=""
LINE(70*ZF+A,55*ZF+B)-(300*ZF+A,55*ZF+B)
LINE(70*ZF+A,85*ZF+B)-(100*ZF+A,55*ZF+B)
LINE(100*ZF+A,55*ZF+B)-(130*ZF+A,85*ZF+B)
LINE(300*ZF+A,85*ZF+B)-(270*ZF+A,55*ZF+B)
LINE(270*ZF+A,55*ZF+B)-(240*ZF+A,85*ZF+B)
LINE(10*ZF+A,10*ZF+B)-(360*ZF+A,120*ZF+B),,B
LINE(152*ZF+A,120*ZF+B)-(182*ZF+A,100*ZF+B)
LINE(212*ZF+A,120*ZF+B)-(182*ZF+A,100*ZF+B)
PAINT(165*ZF+A,114*ZF+B)
RETURN
```

```
130:
REM antitank co #130
U=5:U1$="ANTITANK COMPANY":U2$=""
LINE(10*ZF+A,120*ZF+B)-(185*ZF+A,10*ZF+B)
LINE(185*ZF+A,10*ZF+B)-(360*ZF+A,120*ZF+B)
LINE(145*ZF+A,60*ZF+B)-(185*ZF+A,40*ZF+B)
LINE(185*ZF+A,40*ZF+B)-(225*ZF+A,60*ZF+B)
LINE(145*ZF+A,80*ZF+B)-(185*ZF+A,60*ZF+B)
LINE(185*ZF+A,60*ZF+B)-(225*ZF+A,80*ZF+B)
LINE (10*ZF+A,10*ZF+B)-(360*ZF+A,120*ZF+B),,B
LINE(152*ZF+A,120*ZF+B)-(182*ZF+A,100*ZF+B)
LINE(212*ZF+A,120*ZF+B)-(182*ZF+A,100*ZF+B)
PAINT(165*ZF+A,114*ZF+B)
RETURN

131:
REM airborne  mortar co #131
U=5:U1$="AIRBORNE MORTAR COMPANY":U2$=""
CIRCLE(130*ZF+A,120*ZF+B),_
      50*ZF,1,15*PI/180,165*PI/180
CIRCLE(240*ZF+A,120*ZF+B),_
      50*ZF,1,15*PI/180,165*PI/180
LINE(178*ZF+A,115*ZF+B)-(192*ZF+A,115*ZF+B)
CIRCLE(182*ZF+A,80*ZF+B),25*ZF
LINE(182*ZF+A,25*ZF+B)-(182*ZF+A,70*ZF+B)
LINE(152*ZF+A,50*ZF+B)-(212*ZF+A,50*ZF+B)
LINE(152*ZF+A,60*ZF+B)-(212*ZF+A,60*ZF+B)
LINE(152*ZF+A,35*ZF+B)-(182*ZF+A,25*ZF+B)
LINE(182*ZF+A,25*ZF+B)-(212*ZF+A,35*ZF+B)
RETURN

132:
REM airborne  engineer co #132
U=5:U1$="AIRBORNE ENGINEER COMPANY":U2$=""
CIRCLE(130*ZF+A,120*ZF+B),_
      50*ZF,1,15*PI/180,165*PI/180
CIRCLE(240*ZF+A,120*ZF+B),_
      50*ZF,1,15*PI/180,165*PI/180
LINE(178*ZF+A,115*ZF+B)-(192*ZF+A,115*ZF+B)
LINE (90*ZF+A,55*ZF+B)-(280*ZF+A,55*ZF+B)
LINE (90*ZF+A,55*ZF+B)-(90*ZF+A,75*ZF+B)
LINE (180*ZF+A,55*ZF+B)-(180*ZF+A,75*ZF+B)
LINE (280*ZF+A,55*ZF+B)-(280*ZF+A,75*ZF+B)
RETURN
```

```
133:
REM airborne medical co #133
U=5:U1$="AIRBORNE MEDICAL COMPANY":U2$=""
CIRCLE(130*ZF+A,120*ZF+B),_
      50*ZF,1,15*PI/180,165*PI/180
CIRCLE(240*ZF+A,120*ZF+B),_
      50*ZF,1,15*PI/180,165*PI/180
LINE(178*ZF+A,115*ZF+B)-(192*ZF+A,115*ZF+B)
LINE (10*ZF+A,65*ZF+B)-(360*ZF+A,65*ZF+B)
LINE(185*ZF+A,10*ZF+B)-(185*ZF+A,100*ZF+B)
RETURN

134:
REM airborne support co #134
U=5:U1$="AIRBORNE SUPPLY COMPANY":U2$=""
CIRCLE(130*ZF+A,90*ZF+B),_
      50*ZF,1,15*PI/180,165*PI/180
CIRCLE(240*ZF+A,90*ZF+B),_
      50*ZF,1,15*PI/180,165*PI/180
LINE(178*ZF+A,85*ZF+B)-(192*ZF+A,85*ZF+B)
LINE (10*ZF+A,100*ZF+B)-(360*ZF+A,120*ZF+B),1,BF
RETURN

135:
REM cavalry troop #135
U=5:U1$="CAVALRY TROOP":U2$=""
CIRCLE(90*ZF+A,65*ZF+B),_
      48*ZF,1,90*PI/180,270*PI/180
CIRCLE(280*ZF+A,65*ZF+B),_
      48*ZF,1,270*PI/180,90*PI/180
LINE(90*ZF+A,45*ZF+B)-(280*ZF+A,45*ZF+B)
LINE(90*ZF+A,85*ZF+B)-(280*ZF+A,85*ZF+B)
LINE(10*ZF+A,120*ZF+B)-(360*ZF+A,10*ZF+B)
CIRCLE(130*ZF+A,120*ZF+B),_
      50*ZF,1,15*PI/180,165*PI/180
CIRCLE(240*ZF+A,120*ZF+B),_
      50*ZF,1,15*PI/180,165*PI/180
LINE(178*ZF+A,115*ZF+B)-(192*ZF+A,115*ZF+B)
RETURN
```

```
136:
REM liaison squad #136
U=5:U1$="AVIATION LIAISON":U2$=""
CIRCLE(100*ZF+A,100*ZF+B),_
      35*ZF,1,56*PI/180,304*PI/180
CIRCLE(270*ZF+A,100*ZF+B),_
      35*ZF,1,236*PI/180,124*PI/180
LINE(120*ZF+A,87*ZF+B)-(250*ZF+A,113*ZF+B)
LINE(120*ZF+A,113*ZF+B)-(250*ZF+A,87*ZF+B)
PAINT(100*ZF+A,100*ZF+B),1
PAINT(270*ZF+A,100*ZF+B),1
LINE(140*ZF+A,75*ZF+B)-(230*ZF+A,75*ZF+B)
LINE(185*ZF+A,48*ZF+B)-(140*ZF+A,75*ZF+B)
LINE(185*ZF+A,48*ZF+B)-(230*ZF+A,75*ZF+B)
LINE(140*ZF+A,75*ZF+B)-(230*ZF+A,75*ZF+B)
LINE(185*ZF+A,48*ZF+B)-(140*ZF+A,75*ZF+B)
LINE(185*ZF+A,48*ZF+B)-(230*ZF+A,75*ZF+B)
CIRCLE(80*ZF+A,40*ZF+B),30*ZF
CIRCLE(290*ZF+A,40*ZF+B),30*ZF
LINE(110*ZF+A,40*ZF+B)-(260*ZF+A,40*ZF+B)
RETURN

137:
REM liaison squad #137
U=5:U1$="AVIATION LIAISON":U2$=""
CIRCLE(100*ZF+A,100*ZF+B),_
      35*ZF,1,56*PI/180,304*PI/180
CIRCLE(270*ZF+A,100*ZF+B),_
      35*ZF,1,236*PI/180,124*PI/180
LINE(120*ZF+A,87*ZF+B)-(250*ZF+A,113*ZF+B)
LINE(120*ZF+A,113*ZF+B)-(250*ZF+A,87*ZF+B)
PAINT(100*ZF+A,100*ZF+B),1
PAINT(270*ZF+A,100*ZF+B),1
LINE(140*ZF+A,75*ZF+B)-(230*ZF+A,75*ZF+B)
LINE(185*ZF+A,48*ZF+B)-(140*ZF+A,75*ZF+B)
LINE(185*ZF+A,48*ZF+B)-(230*ZF+A,75*ZF+B)
PAINT(185*ZF+A,60*ZF+B),1
CIRCLE(80*ZF+A,40*ZF+B),30*ZF
CIRCLE(290*ZF+A,40*ZF+B),30*ZF
LINE(110*ZF+A,40*ZF+B)-(260*ZF+A,40*ZF+B)
RETURN
```

```
138:
REM flugplatzkommandantur  #138
U=5:U1$="FIELD AIRPORT":U2$=""
CIRCLE(100*ZF+A,100*ZF+B),_
      35*ZF,1,56*PI/180,304*PI/180
CIRCLE(270*ZF+A,100*ZF+B),_
      35*ZF,1,236*PI/180,124*PI/180
LINE(120*ZF+A,87*ZF+B)-(250*ZF+A,113*ZF+B)
LINE(120*ZF+A,113*ZF+B)-(250*ZF+A,87*ZF+B)
PAINT(100*ZF+A,100*ZF+B),1
PAINT(270*ZF+A,100*ZF+B),1
LINE(182*ZF+A,15*ZF+B)-(122*ZF+A,35*ZF+B)
LINE(182*ZF+A,15*ZF+B)-(242*ZF+A,35*ZF+B)
LINE(150*ZF+A,40*ZF+B)-(150*ZF+A,80*ZF+B)
LINE(150*ZF+A,65*ZF+B)-(220*ZF+A,40*ZF+B)
LINE(162*ZF+A,60*ZF+B)-(220*ZF+A,80*ZF+B)
RETURN

139:
REM brigade maintenance co #139
U=5:U1$="MAINTAINANCE COMPANY":U2$=""
CIRCLE(50*ZF+A,65*ZF+B),_
      50*ZF,1,270*PI/180,90*PI/180
CIRCLE(320*ZF+A,65*ZF+B),_
      50*ZF,1,90*PI/180,270*PI/180
LINE (100*ZF+A,65*ZF+B)-(270*ZF+A,65*ZF+B)
RETURN

140:
REM brigade supply co #140
U=5:U1$="SUPPLY COMPANY":U2$=""
LINE (10*ZF+A,100*ZF+B)-(360*ZF+A,100*ZF+B)
RETURN

141:
REM antitank co #141
U=5:U1$="ANTITANK COMPANY":U2$=""
LINE(10*ZF+A,120*ZF+B)-(185*ZF+A,10*ZF+B)
LINE(185*ZF+A,10*ZF+B)-(360*ZF+A,120*ZF+B)
LINE(145*ZF+A,60*ZF+B)-(185*ZF+A,40*ZF+B)
LINE(185*ZF+A,40*ZF+B)-(225*ZF+A,60*ZF+B)
LINE(145*ZF+A,80*ZF+B)-(185*ZF+A,60*ZF+B)
LINE(185*ZF+A,60*ZF+B)-(225*ZF+A,80*ZF+B)
RETURN
```

```
142:
REM engineer company #142
U=5:U1$="ENGINEER COMPANY":U2$=""
LINE (90*ZF+A,55*ZF+B)-(280*ZF+A,55*ZF+B)
LINE (90*ZF+A,55*ZF+B)-(90*ZF+A,75*ZF+B)
LINE (180*ZF+A,55*ZF+B)-(180*ZF+A,75*ZF+B)
LINE (280*ZF+A,55*ZF+B)-(280*ZF+A,75*ZF+B)
RETURN

143:
REM wehrgericht #143
U=0:U1$="wehrgericht":U2$=""
CIRCLE(240*ZF+A,85*ZF+B),40*ZF,1,PI,2*PI
CIRCLE(240*ZF+A,50*ZF+B),40*ZF,1,10/180*PI,PI
LINE(200*ZF+A,50*ZF+B)-(200*ZF+A,85*ZF+B)
LINE(280*ZF+A,65*ZF+B)-(280*ZF+A,85*ZF+B)
LINE(240*ZF+A,65*ZF+B)-(280*ZF+A,65*ZF+B)
LINE(50*ZF+A,33*ZF+B)-(80*ZF+A,103*ZF+B)
LINE(80*ZF+A,103*ZF+B)-(110*ZF+A,65*ZF+B)
LINE(110*ZF+A,65*ZF+B)-(140*ZF+A,103*ZF+B)
LINE(140*ZF+A,103*ZF+B)-(170*ZF+A,33*ZF+B)
RETURN

144:
REM hhq infantry brigade #144
U=5:U1$="HEADQUATERS AND HEADQUARTERS CO"
U2$="INFANTRY BRIGADE"
GOSUB u4
LINE(10*ZF+A,10*ZF+B)-(360*ZF+A,25*ZF+B),1,BF
LINE(170*ZF+A,36*ZF+B)-(190*ZF+A,42*ZF+B)
LINE(190*ZF+A,36*ZF+B)-(170*ZF+A,42*ZF+B)
RETURN
```

3. Source Code Program INSYSDAT

```
REM program INSYSDAT                          JUNE 1987
REM
REM input data for arbitrary item within arbitrary
REM system required information: stock number, item
REM description, number per superset, unit price,MTBF,
REM replacement level, repair level, MTTR, item
REM redundancy, location in overall system.
REM
REM **************************************************
REM **             Variable Directory               **
REM **************************************************
REM ** A$(x)      : auxiliary string variable       **
REM ** Ax$        : field variable for rand. acc. file*
REM ** E$         : auxiliary string variable       **
REM ** F          : failure flag                    **
REM ** FZ         : data set counter                **
REM ** I$         : item description                **
REM ** ML         : repair level                    **
REM ** MT         : repair time                     **
REM ** MTBF       : mean time between failures      **
REM ** NAM$       : string variable for item file   **
REM ** NC         : number of items                 **
REM ** NN$        : national stock number           **
REM ** P          : item price                      **
REM ** REDUND     : redundancy                      **
REM ** RL         : replacement level               **
REM ** NN$        : national stock number           **
REM ** TYPE       : item type                       **
REM ** X          : auxiliary variable              **
REM **************************************************
```

```
REM
REM                 ****************************
REM                 *  Start of Main Program   *
REM                 ****************************
REM
DIM A$(10)
A$(0)="No Repair":A$(2)="Organ. Maintainance"
A$(3)="Direct / General Support":A$(4)="Depot"
start:
CLS:F=0
LOCATE 1,1
PRINT "Item Type (1=System,2=Component,3...8=Subcomp,";
INPUT "9=Part)";TYPE
IF (TYPE<1 OR TYPE >9) THEN start
LOCATE 2,1:INPUT "National Stock Number :";NN$
GOSUB dupl:IF F=3 OR F=4 THEN start
CLS:LOCATE 1,1:PRINT "National Stock Number :";NN$
LOCATE 3,1:INPUT "Item Description :";I$
LOCATE 4,1:INPUT "Unit Price in US Dollars : $";P
IF TYPE=1 THEN cont1
LOCATE 5,1:INPUT "How many of these Items  : ";NC
LOCATE 6,1:INPUT "MTBF ( assumed )  in Days: ";MTBF
LOCATE 7,1
PRINT "Replacement Level ";
INPUT "(2=Org Mn,3=DS/GS,4=Depot) : ";RL
IF TYPE<9 THEN LOCATE 8,1:_
PRINT "Repair Level ";
INPUT "(0=no,2=Org Mn,3=DS/GS,4=Depot) : ";ML
IF TYPE=9 THEN
LOCATE 8,1:ML=0
PRINT "Repair Level (0=no,2=Org";
PRINT " Mn,3=DS/GS,4=Depot) : ";ML
END IF
IF (ML<RL AND ML>0) THEN F=1:_
  GOSUB errflag:GOTO start
LOCATE 9,1
INPUT "MTTR (Repair/Replace) in Hours: ";MTTR
REM LOCATE 10,1:_
REM INPUT "Redundancy                   : ";redund
REDUND=1                        :REM default

cont1:
CLS
PRINT "Type                           : ";
IF TYPE=1 THEN PRINT"System":GOTO cont2
IF TYPE=2 THEN PRINT"Component":GOTO cont2
IF TYPE=9 THEN PRINT"Part":GOTO cont2
PRINT"Subcomponent"
```

```
cont2:
PRINT "National Stock Number    : ";NN$
PRINT "Item Description         : ";I$
PRINT "Unit Price in US Dollars :$";
PRINT USING " ######.##";P
IF TYPE=1 THEN cont3
PRINT "Number of items          : ";NC
PRINT "MTBF ( assumed )         : ";MTBF;"days"
PRINT "Replacement Level        : ";A$(RL)
PRINT "Repair Level             : ";A$(ML)
PRINT "MTTR (replace)           : ";
PRINT USING "####.#";MTTR;:PRINT " hours"
rem PRINT "Redundancy                   : ";REDUND

cont3:
LOCATE 24,1
PRINT "  *******    Entries correct ? (y/n)";
INPUT "     *******";E$
IF (E$="N" OR E$="n") THEN GOTO start
IF (E$="Y" OR E$="y") THEN GOSUB subwrite:GOTO cont4
GOTO cont3

cont4:
CLS:LOCATE 15,15:INPUT "another entry ?";E$
IF (E$="y" OR E$="Y") THEN start
CLS:END
REM
REM             ***************************
REM             *   End of Main Program   *
REM             ***************************
REM *****************************************************
```

```
REM
REM subroutine write data to random access file
REM
subwrite:
REM
OPEN "R",1,NAM$,60
FIELD#1,1 AS A1$,2 AS A2$,16 AS A3$,20 AS A4$,_
        8 AS A5$,5 AS A6$,1 AS A7$,1 AS A8$,_
        5 AS A9$,1 AS AR$
LSET A1$=RIGHT$(STR$(TYPE),1)
LSET A2$=RIGHT$(STR$(NC),2)
LSET A3$=NN$
LSET A4$=I$
LSET A5$=RIGHT$(STR$(P),8)
LSET A6$=RIGHT$(STR$(MTBF),8)
LSET A7$=RIGHT$(STR$(RL),1)
LSET A8$=RIGHT$(STR$(ML),1)
LSET A9$=RIGHT$(STR$(MTTR),5)
LSET AR$=RIGHT$(STR$(REDUND),1)
PUT#1,FZ
CLOSE #1
RETURN
REM
REM **************************************************
REM
errflag:
REM soubroutine errorhandling
REM
CLS:LOCATE 15,15
IF F=1 THEN_
  PRINT "input error: repair level invalid":GOTO cont5
IF F=3 THEN_
  PRINT "input error: item already stored":GOTO cont5
IF F=4 THEN_
  PRINT "input error: number of items exceeds maximum"
cont5:
LOCATE 20,18:PRINT "press any key to continue"
E$=INPUT$(1)
RETURN
REM
REM **************************************************
```

```
REM
REM subroutine check if item is already stored
REM
dupl:
REM
IF TYPE>1 THEN LOCATE 5,1:_
PRINT "belongs to what item ";
INPUT "(next level stocknumber): ";NAM$
LOCATE 5,1:FOR X=1 TO 78:PRINT " ";:NEXT:PRINT
IF TYPE=1 THEN NAM$=NN$
NAM$=LEFT$(NAM$,4)+RIGHT$(NAM$,4)+"."+MID$(NAM$,9,3)
OPEN "R",1,NAM$,60:FZ=0
FIELD#1,1 AS A1$,2 AS A2$,16 AS A3$,20 AS A4$,_
        8 AS A5$,5 AS A6$,1 AS A7$,1 AS A8$,_
        5 AS A9$,1 AS AR$
FOR X=1 TO 100
   GET#1,X
   IF MID$(A3$,5,1)="-"THEN FZ=FZ+1:_
      IF A3$=NN$ THEN F=3:GOSUB errflag:GOTO cont6
NEXT X
cont6:
FZ=FZ+1:IF FZ>95 THEN F=4:GOSUB errflag
CLOSE#1:RETURN
REM
REM ***************************************************
```

4. Source Code Program ESTIMATE

```
REM program ESTIMATE                        AUGUST 1987
REM This program simulates SAMPSIZE failures for a
REM maximum of 500 systems in a selectable major
REM unit simultaneously. It requires access to the
REM major unit data, the system distribution data and
REM the system data.
REM Output is the expected failure distribution within
REM the selected major unit in terms of consumed items,
REM used maintenance capabilities and system downtimes.
REM If the number of systems is greater than 2000,
REM the model is scaled.
REM
REM ***************************************************
REM **                Variable Directory             **
REM ***************************************************
REM ** A             : auxiliary variable            **
REM ** A$            : auxiliary stringvarible       **
REM ** Ax$           : fieldvar.'s for random acc. file **
REM ** ADMIN(a)      : admin del. time level a  distr. b**
REM ** AVAIL         : average system availability   **
REM ** B             : auxiliary variable            **
REM ** B$(x)         : fieldvariables for file "001" **
REM ** C             : auxiliary variable            **
REM ** C$(x)         : fieldvariables for file "004" **
REM ** COST          : cost                          **
REM ** COST(x)       : cost spent on system x        **
REM ** COUNT         : counting variable up to SAMPSIZE **
REM ** CPROB(x)      : cumulative failure prob. of item x*
REM ** D             : auxiliary variable            **
REM ** D1            : auxiliary variable            **
REM ** DAYHOURS      : workhours per day             **
REM ** DELSD         : delay std. dev. in % of mean  **
REM ** DIST(a,b)     : distribution of delay times   **
REM ** DOF           : degr. of freedom (GAMMA distrib.)**
REM ** DOWN          : system downtime               **
REM ** DOWN(x)       : sum of delay times on level b **
REM ** E             : storage for time random variables**
REM ** E$            : auxiliary stringvariable      **
REM ** ELTIME        : elapsed time for sim in weeks **
REM ** EX            : current exponential r.n.      **
REM ** EX(x)         : failure time system x         **
REM ** F             : error flag                    **
REM ** F(x)          : auxiliary array               **
REM ** F$            : auxiliary stringvariable      **
REM ** F1            : flag variable                 **
REM ** FH$           : system description            **
REM ** H             : auxiliary variable            **
REM ** H(x)          : auxiliary pointer             **
```

```
REM ** H$(x)       : pointers within item files         **
REM ** HEADS(x)    : manpower requ. level a item b       **
REM ** I           : counting var. for # of failures     **
REM ** I$          : auxiliary stringvariable            **
REM ** IT(x)       : # of systems per major unit x       **
REM ** ITEM$       : fieldvariable for file "001"        **
REM ** J$          : fieldvariable for file "001         **
REM ** L           : auxiliary variable                  **
REM ** LABOR       : labor, used to bring system up      **
REM ** LABOR(x)    : workhours used on system x          **
REM ** LAMBDA      : failure rate of item                **
REM ** LAMBDA(x)   : failure rate of subitem             **
REM ** LAMBDAS     : failure rate of overall system      **
REM ** LEVEL       : pointer for maintenance level       **
REM ** LL          : lower limits for uniform distrib.**
REM ** M           : auxiliary variable                  **
REM ** MTTIME(x)   : repairtime for item x               **
REM ** MTTR        : mean time to repair                 **
REM ** MU(a,b)     : mean of distribution                **
REM ** MULT        : scaling factor for simulation       **
REM ** N           : counting var. for # of systems      **
REM ** N$          : auxiliary variable                  **
REM ** Nx          : normally distr. random numbers      **
REM ** NAM$        : filename of current item file       **
REM ** NI(x)       : # of systems in major unit x        **
REM ** NO(x)       : system number                       **
REM ** NULL$       : auxiliary string variable           **
REM ** O$          : left margin for print out           **
REM ** P           : auxiliary pointer                   **
REM ** P1          : ptr to next active time             **
REM ** PC(x)       : ptr to next active time             **
REM ** PE.         : pointer to next entry               **
REM ** PERNUM      : number of 4 week periods            **
REM ** PI          : trigonometrical constant            **
REM ** POINTER(x)  : pointer to next failing system      **
REM ** PROB(x)     : probability of failure for item Y**
REM ** PS          : pointer to starting entry           **
REM ** Q           : auxiliary variable                  **
REM ** Q$          : fieldvariable for file "001"        **
REM ** REPDIST     : distribution of repairtimes         **
REM ** REPLCLVL    : replacement level for items         **
REM ** REPSD       : rep.time std. dev. in % of mean     **
REM ** RES         : result of r. n. generator           **
REM ** RN          : uniform r. n. to determine subitem*
REM ** S$          : system stock number                 **
REM ** SAMPSIZE    : number of samples to be computed **
REM ** SDOWN(x)    : downtime for system x               **
REM ** SGLSSCST    : cost of one unit of system          **
REM ** SIGMA       : standard deviation                  **
REM ** SPARECST    : cost of all spare stocked parts     **
REM ** SPENT       : amount of money,spent on a system**
```

```
REM ** ST           : intermediate pointer                 **
REM ** START        : pointer to next failing system       **
REM ** STOCK$       : fieldvariable for file "001"         **
REM ** SYS$         : fieldvariable for file "004"         **
REM ** SYSCOST      : systems+spares costs in major unit*
REM ** SYSNUM       : # of systems in the major unit       **
REM ** SYSWKHRS     : weekly # of system operat. hours **
REM ** SYSTEMS(x)   : up to 500 systems                    **
REM ** T(x)         : auxiliary array                      **
REM ** TA(b)        : admin. delay level b                 **
REM ** TC(x)        : # of sys in m.u. w/o lower echelon*
REM ** TCOST        : total cost for all comsumed parts**
REM ** TDOWN        : total down time for all systems   **
REM ** TRANSP(a)    : transp. del.time level a distr. b**
REM ** TS(x)        : # of sys in m.u. w lower echelon **
REM ** TT(x)        : transp. delay level b                **
REM ** U            : auxiliary variable                   **
REM ** Ux           : uniformly distributed r. n.          **
REM ** U$(x)        : major unit type                      **
REM ** UL           : upper limits for uniform distr.      **
REM ** UN(x)        : major unit type                      **
REM ** UP           : auxiliary variable                   **
REM ** V            : auxiliary variable                   **
REM ** WEEKDAYS     : number of workingdays per week       **
REM ** X            : auxiliary variable                   **
REM ** X(x)         : auxiliary array                      **
REM ** Y            : auxiliary variable                   **
REM ** Z            : control variable                     **
REM ** Z(x)         : auxiliary array                      **
REM ** ZIF          : counter for file "001"               **
REM *******************************************************
```

```
REM
REM                    ##########################
REM                    # Start of Main Program #
REM                    ##########################
REM
CLS
DIM ADMIN(3),B$(10),C$(10),COST(502),DIST(5,5)
DIM DOWN(5),EX(502),F(150),H(10),H$(9),HEADS(5),IT(20)
DIM LABOR(502),LAMBDA(100),MTTIME(5),MU(5,5)
DIM NI(150),NO(502),PC(502),POINTER(500),PROB(100)
DIM SDOWN(502),SYSTEMS(500),T(10),TA(4),TC(21)
DIM TRANSP(3),TS(21),TT(4),U$(20),UN(150),X(10)
DIM Z(150)
DATA 4,6,8
DATA 3,7,16
DATA "user / org maint.","     DS / GS      "
DATA "Depot / Industry "," no more entries ","uniform"
DATA "normal","exponential","Gamma","lognormal"
DATA "fixed","transportation times (one way)"
DATA "admin time delays","brig11","brig12","brig13"
DATA "brig14","brig15","brig16","brig17","brig19"
DATA "div1","div2","div3","div4","div5","div6","corps"
DATA "MECH. BRIG","ARM'D BRIG","MOUNT.BRIG"
DATA "AIRB. BRIG","INF.  BRIG","AVIAT.BRIG"
DATA "ACC.  BRIG","SIGN. BRIG","MECH. DIV "
DATA "DIV. UNITS","ARM'D. DIV","DIV. UNITS"
DATA "MOUNT.DIV ","DIV. UNITS","AIRB.  DIV"
DATA "DIV. UNITS","INF.  DIV.","DIV. UNITS"
DATA "AASLT  DIV","DIV. UNITS","CORPS      "
DATA "CORPS UNITS"
KILL"00?"              :REM erase all old temporary files
RANDOMIZE TIMER        :REM seed random number generator
NULL$="0":O$="          ":PAGE=1
BEEP:LOCATE 10,5
F=0:ON ERROR GOTO errhandler
f1:
PRINT "Enter total number of failures to simulate";
INPUT " ( <= 1000 ): ",SAMPSIZE
IF SAMPSIZE<=0 OR SAMPSIZE>1000 THEN_
  F=1:GOSUB errhandler
A=2:FOR X=1 TO 3:READ MU(A,X):NEXT X
A=1:FOR X=1 TO 3:READ MU(A,X):NEXT X
FOR Y=1 TO 10:READ B$(Y):NEXT Y
CLS:BEEP:LOCATE 10,30:PRINT "adjust paper in printer"
LOCATE 15,30:PRINT "press a key to continue"
l10:
E$=INPUT$(1):IF E$="" THEN l10 ELSE CLS
LPRINT "                              PAGE ";PAGE
INCR PAGE:LPRINT:LPRINT
LPRINT O$;"Number of Failures to be generated : ";
LPRINT SAMPSIZE:LPRINT:LPRINT
```

```
LPRINT O$;_
"Distributions and Parameters for Transportational"
LPRINT O$;_
"=================================================="
LPRINT O$;"        and Administrative Delays"
LPRINT O$;"        =============================="
LPRINT:LPRINT
CLS:LOCATE 10,20
PRINT "Enter desired value for the";
PRINT " standard deviation of"
LOCATE 12,20
PRINT " the delay distributions";
PRINT " in per cent of the mean"
LOCATE 20,20
PRINT "      the default value is 10 per cent"
LOCATE 22,20
PRINT "   to accept press.ENTER,";
INPUT " else type percentage ",E
IF E>0 THEN DELSD=E/100 ELSE DELSD=0.1
A=1:GOSUB dlay   :REM determine transp. delay times
INCR A:GOSUB dlay:REM determine admin. delay times
GOSUB repdist    :REM determine repair time distrib.
LPRINT CHR$(12)

f2:
CLS:BEEP:LOCATE 10,20
INPUT "Enter number of workhours per day ",DAYHOURS
IF DAYHOURS<=0 OR DAYHOURS>24 THEN_
  F=2:GOSUB errhandler

f3:
LOCATE 13,20
INPUT "Enter number of workdays per week ",WEEKDAYS
IF WEEKDAYS<=0 OR WEEKDAYS>7 THEN F=3:GOSUB errhandler
SYSWKHRS=WEEKDAYS*DAYHOURS
OPEN "A",6,"006"
WRITE#6,"WEEKDAYS=",WEEKDAYS,"DAILY HOURS=",DAYHOURS
CLOSE 6

f4:
CLS:BEEP:LOCATE 10,10
PRINT"Enter the System Stock";_
" Number (e.g. 1234-12-123-1234)"
LOCATE 13,45:INPUT" ",S$
IF LEN(S$)<>16 OR MID$(S$,5,1)<>"-"_
   OR MID$(S$,8,1)<>"-" OR MID$(S$,12,1)<>"-"_
   THEN F=4:GOSUB errhandler
C=2
```

```
REM
REM COMPUTE LAMBDA-SYSTEM (LAMBDAS)
REM
CLS:LOCATE 10,10
PRINT "Compute Lambda for overall System"
NAM$=LEFT$(S$,4)+RIGHT$(S$,4)+"."+MID$(S$,9,3)
10 OPEN "i",5,NAM$:CLOSE 5
OPEN "r",5,NAM$,60  :REM read random access data files
FIELD#5,1 AS A1$,2 AS A2$,16 AS A3$,20 AS A4$,_
        8 AS A5$,5 AS A6$,1 AS A7$,1 AS A8$,_
        5 AS A9$,1 AS AR$
GET 5,1:A=VAL(A6$):FH$=A4$:SGLSSCST=VAL(A5$)
IF A=0 THEN 10 ELSE LAMBDAS=A:CLOSE 5:GOTO 101

10:
GET 5,C
WHILE MID$(A3$,5,1)="-"
  LAMBDA(C)=1/VAL(A6$)     :REM individual failure rate

  C=C+1:GET#5,C
WEND
C=C-1:LAMBDAS=0
FOR Y=2 TO C
   LAMBDAS=LAMBDAS+LAMBDA(Y)
NEXT Y:CLOSE#5:C=2
REM
REM GET System Dislocation within Major Unit
REM

101:
CLS:BEEP:LOCATE 8,23
PRINT "System Dislocation within Major Units"
LOCATE 12,20
PRINT "Input filename for system distribution file"
LOCATE 14,26
PRINT "Default filename is DISTRIBN"
LOCATE 20,20
PRINT "To accept default press ENTER or enter name"
LOCATE 22,30:INPUT " ",F$
IF F$="" THEN F$="DISTRIBN"
F1=0
FOR X=1 TO 150
   Z(X)=0
NEXT X
CLS
OPEN "I",5,F$
INPUT#5,A$:X=VAL(A$)
```

```
FOR A=1 TO X
  INPUT#5,UN(A),NI(A)
  IF UN(A)=198 THEN F1=1
  IF UN(A)=199 THEN F1=2
NEXT A
CLOSE#5
FOR A=0 TO 20
   TS(A)=0
NEXT A
FOR A=1 TO 8
   READ F$:D=VAL(RIGHT$(F$,2)):GOSUB compsys
NEXT A
FOR A=1 TO 6
   READ F$:D=VAL(RIGHT$(F$,1)):GOSUB compsys
NEXT A
READ F$:D=0:GOSUB compsys
LPRINT "                              PAGE ";PAGE
INCR PAGE
LPRINT:LPRINT
F$="Dislocation of "+FH$+"within Corps"
LPRINT O$;O$;F$
B=LEN(F$):NN$=""
FOR A=1 TO B:NN$=NN$+"=":NEXT A
LPRINT O$;O$;NN$:LPRINT:LPRINT
LPRINT O$;O$;"Major Unit    # of systems"
LPRINT O$;O$;"---------------------------":LPRINT
CLS:LOCATE 2,10
PRINT "==========================================="
LOCATE 3,10
PRINT "== Menu :  CHOICE OF MAJOR UNIT TYPE        =="
LOCATE 4,10
PRINT "==========================================="
LOCATE 5,10
PRINT "==   Customer      # systems     selection  =="
LOCATE 6,10
PRINT "==  --------------------------------------  =="
M=1
FOR A=11 TO 17
  READ F$:IF Z(A)>0 THEN LPRINT O$;O$;F$,:_
        LPRINT USING "######";TS(A)
  IF Z(A)=0 THEN 11
  IT(M)=TS(A)
  LOCATE M+6,10
  PRINT"==                                          =="
  LOCATE M+6,15
  PRINT F$:LOCATE M+6,30:PRINT "(";TS(A);")"
  LOCATE M+6,45:PRINT ": ";M:U$(M)=F$:INCR M
11:
NEXT A
A=19:READ F$
```

```
IF Z(A)>0 THEN LPRINT O$;O$;F$,:_
        LPRINT USING "######";TS(A)
IF Z(A)=0 THEN la1
IT(M)=TS(A)
LOCATE M+6,10
PRINT "==                                               =="
LOCATE M+6,15
PRINT F$:LOCATE M+6,30:PRINT "(";TS(A);")"
LOCATE M+6,45:PRINT ": ";M:U$(M)=F$:INCR M
la1:
FOR A=1 TO 6
  READ F$,NN$
  IF Z(A)=0 THEN 12
  LPRINT O$;O$;F$,
  LPRINT USING "######";TS(A):LPRINT O$;O$;NN$,
  LPRINT USING "######";TC(A)
  LOCATE M+6,10
  PRINT"==                                              =="
  LOCATE M+6,15
  PRINT F$:LOCATE M+6,30:PRINT "(";TS(A);")"
  IT(M)=TS(A)
  LOCATE M+6,45:PRINT ": ";M:U$(M)=F$:INCR M
  LOCATE M+6,10
  PRINT"==                                              =="
  LOCATE M+6,15
  PRINT NN$:LOCATE M+6,30:PRINT "(";TC(A);")"
  IT(M)=TC(A)
  LOCATE M+6,45:PRINT ": ";M:U$(M)=NN$:INCR M
12:
NEXT A
READ F$,NN$
LPRINT O$;O$;F$,:LPRINT USING "######";TS(0)
LPRINT O$;O$;NN$,:LPRINT USING"######";TC(0)
LOCATE M+6,10
PRINT "==                                               =="
LOCATE M+6,15:PRINT F$
LOCATE M+6,30:PRINT "(";TS(0);")"
IT(M)=TS(0):LOCATE M+6,45:PRINT ": ";M:U$(M)=F$:INCR M
LOCATE M+6,10
PRINT "==                                               =="
LOCATE M+6,15:PRINT NN$
LOCATE M+6,30:PRINT "(";TC(0);")"
IT(M)=TC(0):LOCATE M+6,45:PRINT ": ";M:U$(M)=NN$:INCR M
F$="Reserve":Y=0
IF F1=1 THEN Y=INT(NI(X)*TS(0)+.5):_
  LPRINT O$;O$;F$,:LPRINT USING "######";Y
IF F1=1 THEN LOCATE M+6,10:_
  PRINT"==                                              =="
IF F1=1 THEN LOCATE M+6,15:PRINT F$:LOCATE M+6,30:_
  PRINT "(";Y;")":IT(M)=Y:_
  LOCATE M+6,45:PRINT ": ";M:U$(M)=F$:M=M+1
```

```
IF F1=2 THEN LPRINT O$;O$;F$,:_
  LPRINT USING "######";NI(X)
IF F1=2 THEN LOCATE M+6,10:_
  PRINT"==                                                    =="
IF F1=2 THEN LOCATE M+6,15:PRINT F$:_
  LOCATE M+6,30:PRINT "(";NI(X);")":_
  IT(M)=NI(X):LOCATE M+6,45:PRINT ":  ";M:U$(M)=F$:M=M+1
LPRINT O$;O$;"------------------"
F$="Total"
IF F1=1 THEN LPRINT O$;O$;F$,:_
  LPRINT USING "######";Y+TS(0)
IF F1=2 THEN LPRINT O$;O$;F$,:_
  LPRINT USING "######";NI(X)+TS(0)
LOCATE M+6,10
PRINT "=================================================="
REM
REM  GET MAJOR UNIT DATA
REM
13:
MIN=1000
FOR X=1 TO M-1
   IF MIN>IT(X) THEN MIN=IT(X)
NEXT X
IF SAMPSIZE<MIN THEN F=5:GOSUB errhandler
BEEP:LOCATE 24,20:INPUT "Enter choice : ",E
IF E<1 OR E>M-1 OR IT(E)>SAMPSIZE THEN_
LOCATE 24,20:PRINT "                           ":GOTO 13
LPRINT:LPRINT:LPRINT:LPRINT
LPRINT O$;O$;"choice :";E,:_
IF RIGHT$(U$(E),5)="UNITS" THEN_
   MU1$=U$(E):MU2$=U$(E-1):_
   LPRINT U$(E),U$(E-1):LPRINT_
   ELSE MU1$=U$(E):MU2$="?":LPRINT U$(E):LPRINT
ADDIT=0:MULT=1:I=IT(E)
WHILE I+ADDIT>500       :REM scaling of number
  MULT=MULT*3:I=INT(IT(E)/MULT):ADDIT=IT(E)-MULT*I
WEND:C=2
OPEN "o",5,"005"
WRITE#5,"number of systems:",I
WRITE#5,"number of simulations:",SAMPSIZE
REM
REM Generation of MTBF's
REM
CLS:LOCATE 11,27
PRINT "Subroutine FAILURE TIME Generation      "
PS=1
FOR N=1 TO I
   MU=1/LAMBDAS
   GOSUB expon
   GOSUB waittime
NEXT N
```

```
FOR N=1 TO I:WRITE#5,NO(N),EX(N),PC(N):NEXT N
CLOSE 5
REM At this point for up to 500 systems {NO(x)} the
REM MTBF's {EX(x)} and the linking pointers {PC(X)} are
REM predetermined and stored.
REM Now starts the simulation core. For SAMPSIZE trials
REM the program picks the most current MTBF,
REM generates a failure down to the part level and
REM computes all times and costs involved.
REM After having done this the program generates a new
REM MTBF for the respective system, gets this into the
REM LINKED LISTING and goes to the next startpointer.
REM At  the end  of this  part for SAMPSIZE simulations
REM the number of needed components,  subcomponents and
REM parts is known.
REM For each system the ratio of up and downtime,
REM the amount of maintenance manhours for each support
REM level and the respective administrative and/or
REM transportation delay times will be computed.
SYSNUM=I*MULT:CT=1
OPEN "r",1,"001",80
FIELD#1,2 AS J$,16 AS STOCK$,18 AS ITEM$,3 AS B$(1),_
      3 AS B$(2),3 AS B$(3),3 AS B$(4),3 AS B$(5),_
      3 AS B$(6),3 AS B$(7),3 AS B$(8),4 AS B$(9),_
      1 AS Q$,8 AS Q1$,7 AS L1$
OPEN "r",3,"003",80
FIELD#3,4 AS S1$,3 AS RL$,3 AS I1$,16 AS N1$,_
        11 AS RT$,11 AS TA$,11 AS TT$,_
        11 AS MTTF$,10 as L$
OPEN "r",4,"004",30
FIELD#4,3 AS C$(1),3 AS C$(2),3 AS C$(3),3 AS C$(4),_
        3 AS C$(5),3 AS C$(6),3 AS C$(7),3 AS C$(8),_
        6 AS SYS$
CLS:LOCATE 10,28:PRINT " Iteration for Simulation";
FOR COUNT=1 TO SAMPSIZE: REM for SAMPSIZE trials
   N$=S$:LOCATE 10,20:PRINT COUNT;
   IF COUNT=1 THEN PRINT "st";
   IF COUNT=2 THEN PRINT "nd";
   IF COUNT=3 THEN PRINT "rd";
   IF COUNT>3 THEN PRINT "th";
   UP=EX(PS):P=PC(PS)
   D1=SDOWN(PS):SPENT=COST(PS)
   LABOR=LABOR(PS)
REM
REM Generate Failure down to Part Level
REM
   C=2:GOSUB failure:PUT 4,COUNT
   FOR X=1 TO 8:H(X)=VAL(C$(X)):NEXT X
   LEVEL=1
```

```
14:
   NAM$=LEFT$(N$,4)+RIGHT$(N$,4)+"."+MID$(N$,9,3)

   OPEN "r",6,NAM$,60          :REM get item data
   FIELD#6,1 AS A1$,2 AS A2$,16 AS A3$,20 AS A4$,_
           8 AS A5$,5 AS A6$,1 AS A7$,1 AS A8$,_
           5 AS A9$,1 AS AR$
   GET#6,H(LEVEL)
   MU=VAL(A9$):COST=VAL(A5$):N$=A3$
   I=VAL(A1$):REPLCLVL=VAL(A7$)
   CLOSE 6
   LSET L$=STR$(MU)
   SIGMA=REPSD*MU
   IF REPDIST=1 THEN GOSUB uniform:GOTO 14a
   IF REPDIST=2 THEN
      GOSUB normal
      IF RES<0 THEN RES=0
      GOTO 14a
   END IF
   IF REPDIST=3 THEN GOSUB expon:GOTO 14a
   IF REPDIST=4 THEN GOSUB gamma:GOTO 14a
   IF REPDIST=5 THEN GOSUB lognormal:GOTO 14a
   IF REPDIST=6 THEN GOSUB fixed

14a:
   MTTR=RES :REM random variable repair time in hrs
   IF LEVEL=1 THEN GOSUB dtime:_
                   DOWN=DOWN+MTTR:TDOWN=TDOWN+DOWN*MULT

   MTTIME(REPLCLVL)=MTTIME(REPLCLVL)+MTTR
   IF I<9 THEN
     MTBF=0:LEVEL=LEVEL+1
     LSET S1$=SYS$
     LSET RL$=STR$(REPLCLVL)
     LSET I1$=STR$(I)
     LSET N1$=N$
     LSET RT$=STR$(MTTR)
     LSET TA$=STR$(TA(REPLCLVL-1))
     LSET TT$=STR$(TT(REPLCLVL-1))
     LSET MTTF$=STR$(MTBF)
     PUT#3,CT
     INCR CT:GOTO 14
   END IF
   UP=UP+DOWN                  :REM time system up again
   MU=1/LAMBDAS
   GOSUB expon
   MTBF=RES*DAYHOURS
   LSET S1$=SYS$
   LSET RL$=STR$(REPLCLVL)
   LSET I1$=STR$(I)
   LSET N1$=N$
```

```
   LSET RT$=STR$(MTTR)
   LSET TA$=STR$(TA(REPLCLVL-1))
   LSET TT$=STR$(TT(REPLCLVL-1))
   LSET MTTF$=STR$(MTBF)
   PUT#3,CT
   INCR CT
   EX=MTBF+UP                        :REM new failure time
GOSUB upll
NEXT COUNT
REM At this point for up to 500 systems a total of
REM SAMPSIZE failures are stored in "005". The failure
REM parameters are stored in "004". The manpower
REM requirements for each level are stored in "006".
ELTIME=UP/SYSWKHRS:PERNUM=ELTIME/4
OPEN "a",6,"006"
WRITE #6,"Scaling Factor=",MULT
WRITE#6,"Summary of item, manpower-requirement"
CLS:LOCATE 11,20
PRINT "Subroutine Summation of consumed Items  "
LPRINT CHR$(12)
LPRINT "                                    PAGE ";PAGE
INCR PAGE:LPRINT:LPRINT
LPRINT_
"          Summary of Items and Manhours for";_
SAMPSIZE;;"Simulations"
LPRINT_
"         =========================";_
"=========================="
C$(2)="Org. Maintenance"
C$(3)="DS/GS Maintenance"
C$(4)="Depot Maintenance"
FOR X=2 TO 4
  C=1:LPRINT:LPRINT:IF X>2 THEN LPRINT CHR$(12):_
  LPRINT "                                    PAGE ";:_
  LPRINT PAGE:INCR PAGE
  LPRINT:LPRINT
  LPRINT "               ";CHR$(27);CHR$(45);CHR$(1);_
  "For Level :";C$(X);CHR$(27);CHR$(45);CHR$(0);:LPRINT
  LPRINT:LPRINT:LPRINT O$;_
  "  stock number      Item Description     ";_
  "     # used"
  LPRINT O$;"=====================================";_
  "=================="
  GET#1,C
  Q=VAL(Q$)
  WHILE MID$(STOCK$,5,1)="-"
    IF Q=X THEN LPRINT O$;STOCK$,ITEM$,VAL(B$(9))
    INCR C:GET#1,C:Q=VAL(Q$)
  WEND
  LPRINT:LPRINT:LPRINT
```

```
  LPRINT O$;_
  "total consumed manhours on this level       : ";
  LPRINT USING "######.##  hrs.";MTTIME(X)
  LPRINT:LPRINT:LPRINT
  LPRINT O$;_
  "average manhrs within 4 weeks on this level : ";
  LPRINT USING "######.##  hrs.";MTTIME(X)/PERNUM
  Y=MTTIME(X)/(PERNUM*DAYHOURS*WEEKDAYS*4)
REM
REM include at least 1 person if work to do
REM
  IF Y=0 THEN HEADS(X)=0
  IF Y>0 AND Y<=1 THEN HEADS(X)=MULT
  IF Y>1 THEN HEADS(X)=INT(Y+0.5)*MULT
  WRITE #6,X,HEADS(X)
  LPRINT:LPRINT:LPRINT
  LPRINT O$;
  LPRINT "average rounded number";
  LPRINT " of mainten. personal : ";_
  HEADS(X);
  IF HEADS(X)>1 THEN_
     LPRINT "repairmen" ELSE LPRINT "repairman"
NEXT X
LPRINT CHR$(12)
LPRINT "                             PAGE ";:_
LPRINT PAGE:INCR PAGE:LPRINT:LPRINT
LPRINT O$;O$;"Additional Statistics"
LPRINT O$;O$;"====================="
LPRINT:LPRINT:LPRINT:LPRINT
C=1:WRITE#6,"4 week stock levels for items":GET 1,C
WHILE MID$(STOCK$,5,1)="-"
   IF VAL(J$)=9 THEN_
      TCOST=TCOST+VAL(B$(9))*VAL(Q1$)
   N=INT(VAL(B$(9))/PERNUM+.5)
   IF N<1 THEN N=1
   SPARECST=SPARECST+N*VAL(Q1$)
   WRITE#6,J$,STOCK$,ITEM$,N,L1$:INCR C:GET 1,C
WEND
LPRINT:LPRINT:LPRINT
LPRINT O$;
LPRINT using"total elapsed time for ####";SAMPSIZE;
LPRINT USING " simulations : ########.## weeks";ELTIME
LPRINT:LPRINT:LPRINT
LPRINT O$;"total number of 4 week periods for ";
LPRINT USING "#### simulations :";SAMPSIZE;
LPRINT USING " ###.##";PERNUM
LPRINT:LPRINT:LPRINT
LPRINT O$;"average system down time    : ";
LPRINT USING "#####.## Days";(TDOWN/SYSNUM)/DAYHOURS
LPRINT:LPRINT:LPRINT
AVAIL=(1-(TDOWN/SYSNUM)/UP)*100
```

```
LPRINT O$;"average system availability : ";
LPRINT USING "##.## per cent";AVAIL
LPRINT:LPRINT:LPRINT
LPRINT O$;"average maintenance cost per system : ";
LPRINT USING "$$#########.##";TCOST/SYSNUM
LPRINT:LPRINT:LPRINT
SYSCOST=SYSNUM*SGLSSCST+SPARECST
LPRINT O$;"initial system cost for ";
IF MU2$="?" THEN LPRINT MU1$;_
   ELSE LPRINT MU1$;" ";MU2$;
LPRINT ": ";:LPRINT USING "$$#########.##";SYSCOST
LPRINT O$;"(systems and spare parts for 4 weeks)
LPRINT:LPRINT:LPRINT
LPRINT O$;"cost of additional 4 week supplies   : ";
LPRINT USING "$$#########.##";TCOST/PERNUM
LPRINT:LPRINT:LPRINT
LPRINT CHR$(12):CLOSE #6
CLS:LOCATE 10,10
PRINT USING "elapsed time for #### simulations :";_
  SAMPSIZE;
PRINT USING " ######.## weeks";ELTIME
LOCATE 20,34:PRINT "PROGRAM END"
CLOSE:END
REM
REM                  #########################
REM                  #  End of Main Program  #
REM                  #########################
REM
REM **************************************************
```

```
REM
REM Subroutine ENTER ESTIMATED DELAY TIMES
REM
dlay:
REM
READ A$
15:
CLS:LOCATE 10,10
PRINT"Enter required information about the ";A$;":"
PRINT
LOCATE 12,15:PRINT "- on the ";B$(1);" level   ( 1 )"
LOCATE 13,15:PRINT "- on the ";B$(2);" level   ( 2 )"
LOCATE 14,15:PRINT "- on the ";B$(3);" level   ( 3 )"
LOCATE 17,15:PRINT "-        ";B$(4);"         ( 9 )"
16:
INPUT B
IF B<1 OR B>3 AND B<>9 THEN 16 ELSE_
   IF B=9 THEN CLS:GOTO 17_
   ELSE GOSUB deldist:GOTO 15
17:
RETURN
REM
REM ***************************************************
```

```
REM
REM Subroutine ENTER ESTIMATED ADMIN AND TRANSPORT
REM            DELAY DISTRIBUTIONS
REM
deldist:
REM
E$="                    "
18:
CLS:LOCATE 8,30:PRINT A$:LOCATE 10,20
PRINT "Enter desired distribution for the time delays:"
LOCATE 12,20:PRINT B$(5):LOCATE 12,40:PRINT "( 1 )"
LOCATE 13,20:PRINT B$(6):LOCATE 13,40:PRINT "( 2 )"
LOCATE 14,20:PRINT B$(7):LOCATE 14,40:PRINT "( 3 )"
LOCATE 15,20:PRINT B$(8):LOCATE 15,40:PRINT "( 4 )"
LOCATE 16,20:PRINT B$(9):LOCATE 16,40:PRINT "( 5 )"
LOCATE 17,20:PRINT B$(10):LOCATE 17,40:PRINT "( 6 )"
LOCATE 21,26
PRINT "default distribution is NORMAL"
LOCATE 22,20
PRINT "to accept default press ENTER";
INPUT " else type number ",E
IF E>0 THEN DIST(A,B)=E ELSE DIST(A,B)=2
19:
CLS:LOCATE 8,20:PRINT A$
PRINT:PRINT "     On the ";B$(B);_
   " level, enter for the ";_
   B$(DIST(A,B)+4);" distribution":PRINT
   PRINT E$;"the average (mean) value in hours"
   PRINT E$;"the default value is:";
   PRINT MU(A,B)
LOCATE 22,20
PRINT "to accept default press ENTER";
INPUT " else type value ",E
IF E>0 THEN MU(A,B)=E
SIGMA=DELSD*MU(A,B)
LPRINT O$;
IF A=1 THEN_
   LPRINT "transportation delays - level  ";B$(B)
IF A=2 THEN_
   LPRINT "administrative delays - level  ";B$(B)
LPRINT O$;"    distribution ";B$(4+DIST(A,B))
LPRINT O$;
IF DIST(A,B)=1 THEN
   LL=MU(A,B)-SQR(3*SIGMA^2)
   UL=MU(A,B)+SQR(3*SIGMA^2)
   LPRINT USING "Lower Limit= ###.###    ";LL,
   LPRINT USING "Upper Limit= ###.###";UL
END IF
```

```
IF DIST(A,B)=2 THEN
   LPRINT USING "Mean  = ###.###    ";MU(A,B),
   LPRINT USING "Sigma = ###.###";SIGMA
END IF
IF DIST(A,B)=3 THEN
   LPRINT USING "Lambda= ###.###";1/MU(A,B)
END IF
IF DIST(A,B)=4 THEN
   DOF=INT((MU(A,B)/SIGMA)^2)+1
   LPRINT USING "Lambda= ###.###    ";DOF/MU(A,B),
   LPRINT USING "DOF= ###";DOF
END IF
IF DIST(A,B)=5 THEN
   LPRINT USING "Mean  = ###.###    ";MU(A,B),
   LPRINT USING "Sigma = ###.###";SIGMA
END IF
IF DIST(A,B)=6 THEN LPRINT "value= ";MU(A,B)
LPRINT
RETURN
REM
REM ***********************************************
```

```
REM
REM Subroutine ENTER REPAIRTIME DISTRIBUTION
REM
REM the default distribution for the repairtimes is
REM lognormal with mu given by the manufacturer and
REM a sigma of 10 % of the mean
REM
repdist:
REM
CLS:LOCATE 10,20
PRINT "Enter desired distribution for the repair time:"
LOCATE 12,20:PRINT B$(5):LOCATE 12,40:PRINT "( 1 )"
LOCATE 13,20:PRINT B$(6):LOCATE 13,40:PRINT "( 2 )"
LOCATE 14,20:PRINT B$(7):LOCATE 14,40:PRINT "( 3 )"
LOCATE 15,20:PRINT B$(8):LOCATE 15,40:PRINT "( 4 )"
LOCATE 16,20:PRINT B$(9):LOCATE 16,40:PRINT "( 5 )"
LOCATE 17,20:PRINT B$(10):LOCATE 17,40:PRINT "( 6 )"
LOCATE 20,20
PRINT "the default distribution is LOGNORMAL"
REPDIST=5
LOCATE 22,20
INPUT "to accept press ENTER, else type number ",E
IF E>0 THEN REPDIST=E
CLS:LOCATE 10,20
PRINT "Enter desired value for the standard deviation"
LOCATE 12,21
PRINT "the default value is 10 per cent of the mean"
REPSD=0.1
LOCATE 22,20
INPUT "to accept press ENTER, else type percentage ",E
IF E>0 THEN REPSD=E/100
RETURN
REM
REM ****************************************************
```

```
REM
REM Subroutine FAILURE
REM
failure:
REM
NAM$=LEFT$(S$,4)+RIGHT$(S$,4)+"."+MID$(S$,9,3)
FOR X=1 TO 8:H$(X)="0":NEXT X:REM reset stringvariables
X=1
111:
OPEN "r",6,NAM$,60  :REM read random access data files
FIELD#6,1 AS A1$,2 AS A2$,16 AS A3$,20 AS A4$,_
        8 AS A5$,5 AS A6$,1 AS A7$,1 AS A8$,_
        5 AS A9$,1 AS AR$
GET#6,C
WHILE MID$(A3$,5,1)="-"
   LAMBDA(C)=1/VAL(A6$)    :REM individual failure rate

   INCR C:GET#6,C
WEND:DECR C:LAMBDA=0
FOR Y=2 TO C
  LAMBDA=LAMBDA+LAMBDA(Y)
NEXT Y
CPROB=0:RN=RND
FOR Y=2 TO C               :REM for all related items
   PROB(Y)=LAMBDA(Y)/LAMBDA:REM comp. prob. of failure
   CPROB=CPROB+PROB(Y)     :REM compute cumul. prob.s
   IF RN<=CPROB THEN H$(X)=STR$(Y):_
   GOTO 112   :REM pick item
NEXT Y
112:
GET#6,Y:A$=A3$:I$=A4$:A=VAL(A1$)
CLOSE #6:GOSUB upd1:REM get data set
IF A=9 THEN GOTO 113       :REM leave if a part
REM prepare next filename
NAM$=LEFT$(A$,4)+RIGHT$(A$,4)+"."+MID$(A$,9,3)
C=1:INCR X:GOTO 111        :REM next subitem iteration
113:
FOR X=1 TO 8:LSET C$(X)=H$(X):NEXT X
LSET SYS$=STR$(PS)
RETURN
REM
REM ******************************************************
```

```
REM
REM Subroutine UPDATE "001"
REM
upd1:
REM
ZIF=1
l14:
ON ERROR GOTO l16:GET#1,ZIF
ON ERROR GOTO 0
IF MID$(STOCK$,5,1)="-" THEN l15 ELSE l16
l15:
IF STOCK$=A$ THEN H$(9)=STR$(VAL(B$(9))+MULT):GOTO l17
INCR ZIF:GOTO l14
l16:
H$(9)=STR$(MULT)
l17:
LSET J$=A1$:LSET STOCK$=A$:LSET ITEM$=I$:LSET Q$=A7$
FOR Q=1 TO 9:LSET B$(Q)=H$(Q):NEXT Q:LSET Q1$=A5$
LSET L1$=A6$:PUT#1,ZIF:RETURN
REM
REM ***************************************************
```

```
REM
REM Subroutine COMPUTE # OF SYSTEMS FOR MAJOR UNIT D
REM
REM Inputs:     F$      : file name
REM             D       : major unit number
REM             UN(c)   : unit number
REM             NI(c)   : number of these units
REM
compsys:
REM
FOR B=0 TO 20:F(B)=0:NEXT B
OPEN"i",5,F$
INPUT#5,B,NS
FOR B=1 TO NS:INPUT#5,U,V
  IF D<10 AND U<21 THEN Z(10*D+U)=Z(10*D+U)+1
  IF U<21 THEN F(U)=V:Z(U)=Z(U)+1:_
              TS(D)=TS(D)+NI(U+100)*F(U):GOTO 118
  FOR C=1 TO X
    IF U=UN(C) THEN TS(D)=TS(D)+NI(C)*V:_
                    TC(D)=TC(D)+NI(C)*V
  NEXT C
118:
NEXT B
NI(D+100)=TS(D)
CLOSE#5
RETURN
REM
REM ******************************************************
```

```
REM
REM Subroutine EXPON. DISTRIBUTED WAITING TIMES (MTBF)
REM
waittime:
REM
EX=RES*DAYHOURS
P=PS:H=PS:Z=1           :REM set variables
REM
REM  VALUE INTO SEQUENCE (LINKED LISTING)
REM
IF N=1 THEN PE=N:P=N+1:GOTO endsub
IF EX>EX(PE) THEN PC(PE)=N:PE=N:P=N+1:GOTO endsub
IF EX<EX(PS) THEN P=PS:PS=N:GOTO endsub
119:
H=P:P=PC(P)
IF EX>EX(P) THEN 119
PC(H)=N
endsub:
NO(N)=N:EX(N)=EX:PC(N)=P
PC(PE)=N+1
RETURN
REM
REM ****************************************************
```

```
REM
REM Subroutine UNIFORM DISTRIBUTION
REM
REM required inputs  : MU,SIGMA
REM available output : RES
REM
uniform:
REM
LL=MU-SQR(3*SIGMA^2)
UL=MU+SQR(3*SIGMA^2)
RES=(INT((LL+(UL-LL)*RND+.5)*100))/100
RETURN
REM
REM ****************************************************
```

```
REM
REM Subroutine NORMAL DISTRIBUTION
REM Box-Mueller Method
REM
REM required inputs  : MU,SIGMA
REM available output : RES
REM
normal:
REM
U1=RND:U2=RND:PI=3.141592654
N1=COS(2*PI*U2)*SQR(-2*LOG(U1))
N2=SIN(2*PI*U2)*SQR(-2*LOG(U1))
RES=RND
IF RES>.5 THEN RES=MU+N2*SIGMA ELSE RES=MU+N1*SIGMA
RETURN
REM
REM **************************************************


REM
REM Subroutine EXPONENTIAL DISTRIBUTION
REM
REM required input   : MU
REM available output : RES
REM
expon:
REM
RES=-MU*LOG(RND)
RETURN
REM
REM **************************************************
```

```
REM
REM Subroutine GAMMA DISTRIBUTION
REM
REM required input   : MU,SIGMA
REM available output : RES
REM
gamma:
REM
DOF=INT((MU/SIGMA)^2)+1
REM change DELSD to 20%,
REM if DOF too large (DELSD too small)
IF DOF>26 THEN DOF=26
RES=1
FOR X= 1 TO DOF
   RES=RES*RND
NEXT X
RES=-(MU)*LOG(RES)
RETURN
REM
REM **************************************************
```

```
REM
REM Subroutine LOGNORMAL DISTRIBUTION
REM
REM required input   : MU,SIGMA
REM available output: RES
REM
lognormal:
REM
MUN=MU:SIGMAN=SIGMA
REM transform lognorm parameters to normal
MU=2*LOG(MUN)-0.5*LOG(SIGMAN^2+MUN^2)
SIGMA=SQR(LOG(SIGMAN^2+MUN^2)-2*LOG(MUN))
REM generate N(MU,SIGMA)
GOSUB normal
REM generate lognormal
RES=EXP(RES)
RETURN
REM
REM **************************************************
```

```
REM
REM Subroutine FIXED DISTRIBUTION
REM
REM required input   : MU
REM available output : RES
REM
fixed:
REM
RES=MU
RETURN
REM
REM ******************************************************
```

```
REM
REM Subroutine UPDATE LINKED LIST
REM
upll:
REM
P=PC(PS):P1=PS
IF EX<EX(P) THEN GOTO subend ELSE PS=P
IF EX>=EX(PE) THEN PC(PE)=P1:PE=P1:P=P1:GOTO subend
121:
H=P:P=PC(P)
IF EX>EX(P) THEN GOTO 121
PC(H)=P1
subend:
EX(P1)=EX:PC(P1)=P:COST(P1)=COST:LABOR(P1)=LABOR
SDOWN(P1)=DOWN+D1
RETURN
REM
REM ******************************************************
```

```
REM
REM Subroutine GENERATE DOWNTIMES FOR SYSTEM AND ITEMS
REM
dtime:
REM
FOR B=1 TO 3:A=1
  WHILE A<3
   MU=MU(A,B):SIGMA=DELSD*MU
   IF DIST(A,B)=1 THEN GOSUB uniform:GOTO 123
   IF DIST(A,B)=2 THEN
     GOSUB normal
     IF RES<0 THEN RES=0
     GOTO 123
   END IF
   IF DIST(A,B)=3 THEN GOSUB expon:GOTO 123
   IF DIST(A,B)=4 THEN GOSUB gamma:GOTO 123
   IF DIST(A,B)=5 THEN GOSUB lognormal:GOTO 123
   IF DIST(A,B)=6 THEN GOSUB fixed
123:
    IF A=1 THEN TT(B)=RES ELSE TA(B)=RES
  A=A+1:WEND
DOWN(B)=2*TT(B)+TA(B)    :REM total of time delays
NEXT B
DOWN=DOWN(1)
RETURN
REM
REM ***************************************************
```

```
REM
REM Subroutine ERROR HANDLER
REM
errhandler:
REM
IF F=1 THEN F=0:LOCATE 10,20:_
            PRINT "Iterations out of Range":_
            GOSUB keyp:GOTO f1
IF F=2 THEN F=0:LOCATE 10,25:_
            PRINT "Hours out of Range":_
            GOSUB keyp:GOTO f2
IF F=3 THEN F=0:LOCATE 10,25:_
            PRINT "Days out of Range":_
            GOSUB keyp:GOTO f3
IF F=4 THEN F=0:LOCATE 10,25:_
            PRINT "Wrong Input Format":_
            GOSUB keyp:GOTO f4
IF F=5 THEN F=0:LOCATE 10,25:_
            PRINT "Sample Size too small":_
            GOSUB keyp:GOTO f1
IF ERL=10 THEN LOCATE 10,20:_
   PRINT "No File with this Filename: ";NAM$:_
   GOSUB keyp:RESUME f4
IF ERL=20 THEN LOCATE 10,20:_
   PRINT "No File with this Filename: ";F$:_
   GOSUB keyp:RESUME 101
LOCATE 10,20:PRINT "File Error":_
GOSUB keyp:RESUME
RETURN
keyp:
LOCATE 15,25:PRINT "Press any Key"
F$=INPUT$(1)
RETURN
REM
REM **************************************************
```

5. Source Code Program QBEHAVE

```
REM program QBEHAVE                              AUGUST 1987
REM
REM This program recomputes the queueing behavior with
REM the numbers computed in the program ESTIMATE.
REM Given are the four weeks planning data for stocks
REM and manpower requirements. These figures are taken
REM to establish the environment, in which the systems
REM are failing.
REM The program accesses the data in datafiles "003",
REM "005" and "006", computes the arrival times at the
REM maintemance facilities, checks for availability of
REM parts and manpower and processes the systems
REM through the server queues.
REM Output is not only the statistics for the specified
REM number of servers but also a sensitivity analysis
REM with server numbers varying and the consumption
REM behavior with the given supply stocks.
REM The results can be used to propose a structure for
REM the logistics organization needed to support the
REM new systems.
REM
REM
REM  **************************************************
REM  **              Variable Directory              **
REM  **************************************************
REM  ** A         : auxiliary variable               **
REM  ** A(x,y)    : array for OrgMn data             **
REM  ** Ax$       : fieldvariable for sysfile$       **
REM  ** ARRTIME(x):arrival time at OrgMn             **
REM  ** AST       : average shipping time            **
REM  ** B         : auxiliary variable               **
REM  ** B(x,y)    : array for DS/GS data             **
REM  ** BUSY(x)   : time server x is busy            **
REM  ** C(x,y)    : array for DEPOT data             **
REM  ** CHOICE%   : program mode                     **
REM  ** COUNT%    : number of spare part transactions **
REM  ** CURR%(x)  : current stock of item x          **
REM  ** D$(x)     : dummy string variable            **
REM  ** D1        : auxiliary variable               **
REM  ** DAYHOURS  : hours per day, system is operatnl.**
REM  ** DP$       : filename for depot data          **
REM  ** DS$       : filename for ds/gs data          **
REM  ** F%        : busy flag for server             **
REM  ** FTIME     : time of failure                  **
REM  ** H%        : auxiliary variable               **
REM  ** Hx%       : auxiliary variable               **
REM  ** HISENS%   : upper limit for sensitivity anal. **
REM  ** I         : auxiliary variable               **
```

```
REM  ** I%          : auxiliary variable                    **
REM  ** Ix          : arrival time at level x               **
REM  ** I1$         : string variable file "003"            **
REM  ** IC%         : counting variable                     **
REM  ** INTx        : arrival time at level x               **
REM  ** ITEM%       : dataset number for itemfiles          **
REM  ** ITEM$       : stock number                          **
REM  ** ITEMNO%     : number of different items             **
REM  ** J$          : auxiliary string variable             **
REM  ** L$          : string variable file "003"            **
REM  ** LAMBDA      : failure rate of item                  **
REM  ** LAMBDA$     : string variable file "003"            **
REM  ** LEVEL$(x): maintenance level                        **
REM  ** LITTLESx%: minimum stock level before reorder**
REM  ** LOWSENSx%: lower limit of sensitivity anal.  **
REM  ** LT          : auxiliary variable                    **
REM  ** Mx          : repair time level x                   **
REM  ** MANPWR(x): available servers at level x        **
REM  ** MAX%        : max. number of customers in the Q **
REM  ** MIN%(x)     : minimum stock level for item x    **
REM  ** MTTF$       : string variable file "003"            **
REM  ** MTTR        : mean time to repair                   **
REM  ** MULT        : scaling factor if > 500 systems    **
REM  ** N%          : counting variable                     **
REM  ** NPTR%(x) : pointer to next data set for system*
REM  ** NSN$        : string variable file "003"            **
REM  ** OM$         : filename for org mn data              **
REM  ** OUTx%       : Q outtime                             **
REM  ** P%          : auxiliary variable                    **
REM  ** Px%         : pointer in linked listing             **
REM  ** Px(y)       : stores y items with x informations**
REM  ** PA$         : filename for item data                **
REM  ** PAGE%       : page counter for output               **
REM  ** PDEPOT(x,y): start-,endptr in Q server y Depot*
REM  ** PDSGS(x,y): start-,endptr in Q server y DS/GS *
REM  ** PEx%        : ptr for last element                  **
REM  ** PERx        : duration for simulation               **
REM  ** PORGMN(x,y): start-,endptr in Q server y OrgMn*
REM  ** PS%         : auxiliary variable                    **
REM  ** PSx%        : ptr for startelement                  **
REM  ** PSYS(x)     : pointer for system in repair          **
REM  ** PTR%(x)     : pointer to data set in "003"          **
REM  ** Qx%         : number of customers in the system **
REM  ** Q1x%        : number of waiting customers           **
REM  ** QDx%        : cumulative waiting times              **
REM  ** R           : stockout risk factor                  **
REM  ** R%          : current replacement level             **
REM  ** REPLCLVL%: replacement level out of data file**
REM  ** RL$         : string variable file "003"            **
REM  ** RL%         : repair level                          **
REM  ** RT          : repair time                           **
```

```
REM  ** RT$       : string variable file "003"         **
REM  ** S%        : auxiliary variable                 **
REM  ** S$        : system stock number                **
REM  ** Sx%       : number of servers level x          **
REM  ** S1$       : string variable file "003"         **
REM  ** SAMPSIZE%: number of simulation iterations     **
REM  ** SPTR%(x) : pointer for next systemepair        **
REM  ** STOCK(x) : number of items stocked             **
REM  ** SYS       : system number                      **
REM  ** SYS(n)    : system number array                **
REM  ** SYSDOWN(x): total downtime                     **
REM  ** SYSFILE$ : name of system data file            **
REM  ** SYSNUM%   : number of systems in major unit    **
REM  ** Tx%(y)    : spare part information             **
REM  ** TA        : admin delay time                   **
REM  ** TAx       : admin delay time                   **
REM  ** TA$       : string variable file "003"         **
REM  ** TT        : transp delay time                  **
REM  ** TTx       : transp delay time                  **
REM  ** WAITTIME : waiting time for item x             **
REM  ** WAITTTLx : total waiting time level x          **
REM  ** WEEK      : total workhours per week           **
REM  ** WEEKDAYS : days per week, system is operatnl.**
REM  ** X         : auxiliary variable                 **
REM  ** X%        : auxiliary variable                 **
REM  ** Y%        : auxiliary variable                 **
REM  ** Z         : auxiliary variable                 **
REM  *****************************************************
```

```
REM
REM                    ##########################
REM                    # START OF MAIN PROGRAM #
REM                    ##########################
REM
DIM A(8,1000),ARRTIME(500),B(11,1000)
DIM BUSY2(50),BUSY3(50),BUSY4(50),C(11,1000)
DIM CURR%(1000),FTIME(500),ITEM$(1000),ITEM%(1000)
DIM LITTLES%(1000),MANPWR%(3),MIN%(1000),NPTR%(500)
DIM P1(6000),P2%(6000),P3%(6000)
DIM PDEPOT(2,50),PDSGS(2,50),PORGMN(2,50),PTR%(500)
DIM SPTR%(500),STOCK%(1000),SYSDOWN(500)
DIM T1(500),T2%(500)
REM ----------------------------------------
REM A(1,N%) = QUEUE-INTIME  \
REM A(2,N%) = QUEUE-OUTTIME  |
REM A(3,N%) = POINTER FOR 003 \  ORG
REM A(4,N%) = SPARE PART 1    /  MN
REM A(5,N%) = SPARE PART 2   |
REM A(6,N%) = SPARE PART 3   |
REM A(7,N%) = SYSTEM NUMBER /
REM ----------------------------------------
REM B(1,N%) = FAILTIME      \
REM B(2,N%) = ARRIVALTIME    |
REM B(3,N%) = Q-INTIME       |
REM B(4,N%) = Q-OUTTIME      |
REM B(5,N%) = POINTER        |
REM B(6,N%) = PART 1          \  DS/GS
REM B(7,N%) = PART 2          /  DEPOT [C(X,Y)]
REM B(8,N%) = PART 3         |
REM B(9,N%) = REPAIRTIME     |
REM B(10,N%)= ADMIN DELAY    |
REM B(11,N%)= TRANSP DELAY  /
REM ----------------------------------------
CLS
CLS:OPEN "1",6,"006"
INPUT#6,D$(1),WEEKDAYS,D$(2),DAYHOURS
INPUT#6,D$(1),MULT
INPUT#6,D$(1)
REM
REM read manpower data for the three maint. levels
REM
FOR X%=1 TO 3
   INPUT#6,A,MANPWR%(X%)
NEXT X%
INPUT#6,D$(1)
```

```
REM
REM read items to stock and the number of these
REM
X%=1
INPUT#6,J$,ITEM$(X%),D$(2),STOCK%(X%),LAMBDA$
LAMBDA=1/VAL(LAMBDA$)
REM
REM compute restock level for parts
REM
IF VAL(J$)=9 THEN GOSUB poisson
CURR%(X%)=STOCK%(X%)
MIN%(X%)=STOCK%(X%)
WHILE NOT EOF(6)
  INCR X%
  INPUT#6,J$,ITEM$(X%),D$(2),STOCK%(X%),LAMBDA$
  LAMBDA=1/VAL(LAMBDA$)
  IF VAL(J$)=9 THEN GOSUB poisson
  CURR%(X%)=STOCK%(X%)
  MIN%(X%)=STOCK%(X%)
WEND
CLOSE#6
ITEMNO%=X%  :REM total number of different spares
REM
REM determine program mode
REM
GOSUB choice
REM
REM enter input variables SENSLIM, AST and R
REM
GOSUB varin
IF CHOICE%=0 THEN
  S2%=MANPWR%(1)
  S3%=MANPWR%(2)
  S4%=MANPWR%(3)
  GOTO prepdat
END IF
REM
REM set sensitivity bounds for server numbers
REM
LOWSENS2%=INT((1-SENSLIM)*MANPWR%(1))
HISENS2%=INT((1+SENSLIM)*MANPWR%(1)+0.5)
LOWSENS3%=INT((1-SENSLIM)*MANPWR%(2))
HISENS3%=INT((1+SENSLIM)*MANPWR%(2)+0.5)
LOWSENS4%=INT((1-SENSLIM)*MANPWR%(3))
HISENS4%=INT((1+SENSLIM)*MANPWR%(3)+0.5)
IF LOWSENS2%<1 THEN LOWSENS2%=1
IF LOWSENS3%<1 THEN LOWSENS3%=1
IF LOWSENS4%<1 THEN LOWSENS4%=1
IF HISENS2%=MANPWR%(1) THEN INCR HISENS2%
IF HISENS3%=MANPWR%(2) THEN INCR HISENS3%
IF HISENS4%=MANPWR%(3) THEN INCR HISENS4%
```

```
prepdat:
REM
REM prepare data file
REM
OPEN "R",3,"003",80
FIELD#3,4 AS S1$,3 AS RL$,3 AS I1$,16 AS NSN$,_
   11 AS RT$,11 AS TA$,11 AS TT$,11 AS MTTF$,10 AS L$
LPRINT "          Program Start Time is : ";DATE$,TIME$
PAGE%=1
REM
REM loops for sensitivity analysis
REM
IF CHOICE%=0 THEN slp
FOR S4%=LOWSENS4% TO HISENS4%
  FOR S3%=LOWSENS3% TO HISENS3%
    FOR S2%=LOWSENS2% TO HISENS2%
slp:
REM
REM read scaling parameters and failure times
REM
      OPEN "I",5,"005"
      INPUT#5,A$,SYSNUM%
      INPUT#5,A$,SAMPSIZE%
      FOR X%=1 TO SYSNUM%
        INPUT#5,A,FTIME(X%),B
      NEXT X%
      CLOSE#5
REM
REM reset pointers and variables
REM
      FOR X%=1 TO 2:FOR Y%=1 TO 50
        BUSY2(Y%)=0
        BUSY3(Y%)=0
        BUSY4(Y%)=0
        PORGMN(X%,Y%)=0
        PDSGS(X%,Y%)=0
        PDEPOT(X%,Y%)=0
      NEXT Y%:NEXT X%
      FOR X%=1 TO SYSNUM%
        ARRTIME(X%)=0
        PTR%(X%)=0
        NPTR%(X%)=0
        SPTR%(X%)=0
        SYSDOWN(X%)=0
      NEXT X%
      FOR N%=1 TO SAMPSIZE%
        FOR X%=1 TO 8
          A(X%,N%)=0
          B(X%,X%)=0
          C(X%,N%)=0
        NEXT X%
```

```
      FOR X%=9 TO 11
        B(X%,N%)=0
        C(X%,N%)=0
      NEXT X%
    NEXT N%
    MAX2%=0:MAX3%=0:MAX4%=0
    QD2%=0:QD3%=0:QD4%=0
    PER2=0:PER3=0:PER4=0
    WAITTTL2=0:WAITTTL3=0:WAITTTL4=0
    LPRINT
    LPRINT "        Queuing Behavior for Varying";_
          " Numbers of Servers ";:_
    LPRINT USING " PAGE ##";PAGE%:INCR PAGE%
    LPRINT "        ==================================";_
          "====================":LPRINT
    GOSUB sernum
REM
REM read the arrival times at OrgMn
REM
    PS1%=1:PE%=1
    IC%=1:F%=1:RL%=1
    WHILE RL%>0
      GET#3,IC%
      GOSUB getdata
      IF I%=9 THEN F%=1
      INCR IC%
    WEND
    FOR SYS%=1 TO SYSNUM%
      IF ARRTIME(SYS%)=0 THEN ARRTIME(SYS%)=999999
    NEXT SYS%
REM
REM sort data on level OrgMn
REM
    PS%=1:PE%=1
    FOR X%=1 TO SYSNUM%
      GOSUB orgsort
    NEXT X%:PS2%=1
REM
REM repair items on level Org Mn
REM
    LOCATE 20,35:PRINT "Level Org Mn"
    Q2%=0:N%=1:SYS%=PS%

    WHILE N%<=SAMPSIZE% AND ARRTIME(SYS%)<999999
      LOCATE 24,31:PRINT "Service Number  ";N%;
      A(3,N%)=PTR%(SYS%)
      A(7,N%)=SYS%
      INCR Q2%
```

```
REM
REM acquire data for repair
REM
        IC%=PTR%(SYS%):M2=0
        GET#3,IC%
        GOSUB gdata
        INCR IC%
        WHILE VAL(I1$)<9
          GET#3,IC%
          GOSUB gdata
          INCR IC%
        WEND
        INT2=ARRTIME(SYS%):I2=INT2
        FTIME2=FTIME(SYS%)
server2:
        A=10000000:Q12%=0
        FOR X%=1 TO S2% :REM check, if server available

          IF PORGMN(2,X%)<=INT2 THEN
            PORGMN(1,X%)=INT2
            PORGMN(2,X%)=INT2+M2
            DECR Q2%:GOTO iserve2
          ELSEIF PORGMN(2,X%)<A THEN
            A=PORGMN(2,X%)
          END IF
        NEXT X%
        WAITTIME=A-INT2
        WAITTTL2=WAITTTL2+WAITTIME
        INT2=A
        P%=SPTR%(SYS%):INCR QD2%:INCR Q12%
REM
REM check other arrival times for queue size
REM
        WHILE P%<=SAMPSIZE% AND ARRTIME(P%)<999999
          IF ARRTIME(P%)<(INT2+M2) THEN_
            INCR Q12%:P%=SPTR%(P%) ELSE maxq2
        WEND
maxq2:
        IF Q12%>MAX2% THEN MAX2%=Q12% :REM max Q length

        GOTO server2

iserve2:
        IF PER2<PORGMN(2,X%) THEN PER2=PORGMN(2,X%)
        BUSY2(X%)=BUSY2(X%)+M2
        A(1,N%)=PORGMN(1,X%):A(2,N%)=PORGMN(2,X%)
        SYSDOWN(SYS%)=SYSDOWN(SYS%)+_
                      PORGMN(2,X%)+TA2/2+TT2-FTIME2
        IF B(6,N%)=0 THEN GOTO fol
        B(1,N%)=A(2,N%)
        FTIME(SYS%)=A(2,N%)+TA2/2+TT2+MTTF
```

```
fol:
REM
REM check, if another failure for this system
REM
        IF N%=SAMPSIZE% THEN p3
        IF NPTR%(SYS%)>0 THEN
          IC%=NPTR%(SYS%):PTR%(SYS%)=0:NPTR%(SYS%)=0
          F%=1
          WHILE VAL(RL$)>0
            IF NPTR%(SYS%)>0 THEN n2
            GET#3,IC%
            IF VAL(S1$)=SYS% THEN GOSUB getdata
            IF VAL(I1$)=9 THEN F%=1
            INCR IC%
          WEND
n2:
          PS%=SPTR%(SYS%):X%=SYS%:GOSUB orgsort
        ELSE
          PS%=SPTR%(SYS%):GOTO n1
        END IF
n1:
        SYS%=PS%:INCR N%
      WEND
p3:
REM
REM sort data on level DS/GS
REM
      FOR N%=1 TO SAMPSIZE%
        IF B(1,N%)>0 THEN_
          B(2,N%)=B(1,N%)+B(10,N%)/2+B(11,N%)_
        ELSE B(2,N%)=999999
      NEXT N%
      PS%=1:PE%=1
      FOR X%=1 TO SAMPSIZE%
        GOSUB dssort
      NEXT X%:PS3%=PS%
REM
REM repair items on level DS/GS
REM
      LOCATE 20,35:PRINT "Level DS/GS "
      N%=PS%:Q3%=0:SYS%=1
      LOCATE 24,31
      PRINT "                              ";
      WHILE N%<=SAMPSIZE% AND B(1,N%)>0
        INCR Q3%
        INT3=B(2,N%):I3=INT3
        M3=B(9,N%):TA3=B(10,N%):TT3=B(11,N%)
```

```
server3:
        C=10000000:Q13%=0
        FOR Y%=1 TO S3% :REM check, if server available

          IF PDSGS(2,Y%)<=INT3 THEN
            PDSGS(1,Y%)=INT3
            PDSGS(2,Y%)=INT3+M3
            DECR Q3%:GOTO iserve3
          ELSEIF PDSGS(2,Y%)<C THEN
            C=PDSGS(2,Y%)
          END IF
        NEXT Y%
        WAITTIME=C-INT3
        WAITTTL3=WAITTTL3+WAITTIME
        INT3=C
        P%=B(5,N%):INCR QD3%:INCR Q13%
REM
REM check other arrival times for queue size
REM
        WHILE P%<=SAMPSIZE% AND B(2,P%)<999999
          IF B(2,P%)<(INT3+M3) THEN_
            INCR Q13%:P%=B(5,P%) ELSE maxq3
        WEND
maxq3:
        IF Q13%>MAX3% THEN MAX3%=Q13% :REM max Q length

        GOTO server3
iserve3:
        IF PER3<PDSGS(2,Y%) THEN PER3=PDSGS(2,Y%)
        BUSY3(Y%)=BUSY3(Y%)+M3
        B(3,N%)=PDSGS(1,Y%):B(4,N%)=PDSGS(2,Y%)
        IF C(6,N%)>0 THEN C(1,N%)=B(4,N%)
        N%=B(5,N%):INCR SYS%
      WEND




REM
REM sort data on level DEPOT
REM
      FOR N%=1 TO SAMPSIZE%
        IF C(1,N%)>0 THEN_
          C(2,N%)=C(1,N%)+C(10,N%)/2+C(11,N%)_
        ELSE C(2,N%)=999999
      NEXT N%
      PS%=1:PE%=1
      FOR X%=1 TO SAMPSIZE%
        GOSUB dpsort
      NEXT X%:PS4%=PS%
```

```
REM
REM repair items on level DEPOT
REM
      LOCATE 20,35:PRINT "Level DEPOT "
      N%=PS%:Q4%=0:SYS%=1
      WHILE N%<=SAMPSIZE% AND C(1,N%)>0
        INCR Q4%
        INT4=C(2,N%):I4=INT4
        M4=C(9,N%):TA4=C(10,N%):TT4=C(11,N%)
server4:
        E=10000000:Q14%=0
        FOR Z%=1 TO S4%:REM check, if server available
          IF PDEPOT(2,Z%)<=INT4 THEN
            PDEPOT(1,Z%)=INT4
            PDEPOT(2,Z%)=INT4+M4
            DECR Q4%:GOTO iserve4
          ELSEIF PDEPOT(2,Z%)<E THEN
            E=PDEPOT(2,Z%)
          END IF
        NEXT Z%
        WAITTIME=E-INT4
        WAITTTL4=WAITTTL4+WAITTIME
        INT4=E
        P%=C(5,N%):INCR QD4%:INCR Q14%
REM
REM check other arrival times for queue size
REM
        WHILE P%<=SAMPSIZE% AND C(2,P%)<999999
          IF C(2,P%)<(INT4+M4) THEN_
            INCR Q14%:P%=C(5,P%) ELSE maxq4
        WEND
maxq4:
        IF Q14%>MAX4% THEN MAX4%=Q14% :REM max Q length

        GOTO server4


iserve4:
        IF PER4<PDEPOT(2,Z%) THEN PER4=PDEPOT(2,Z%)
        BUSY4(Z%)=BUSY4(Z%)+M4
        C(3,N%)=PDEPOT(1,Z%):C(4,N%)=PDEPOT(2,Z%)
        N%=C(5,N%):INCR SYS%
      WEND
rt4:
      WEEK=WEEKDAYS*DAYHOURS
      PERNUM=PER2/(4*WEEK)
      O$="          "
      LPRINT:LPRINT o$;"LEVEL ORG MN :":LPRINT
      LPRINT "          no of customers     = ";:_
      LPRINT USING "######";SYSNUM%
```

```
LPRINT "         no of failures      = ";:_
LPRINT USING "######";SAMPSIZE%
LPRINT "         no of servers       = ";:_
LPRINT USING "######";S2%
LPRINT "         no of delays        = ";:_
LPRINT USING "######";QD2%
LPRINT "         max queue length    = ";:_
LPRINT USING "######";_
       MAX2%;:LPRINT " customers"
LPRINT "         total waiting time = ";:_
LPRINT USING "########.##";_
       WAITTTL2;:LPRINT " hours":LPRINT
LPRINT O$;
LPRINT "average  waiting time per ";_
       " waiting customer = ";
IF QD2%=0 THEN
   LPRINT "     0.00 hours"
ELSE
   LPRINT using "######.## hours";WAITTTL2/QD2%
END IF
LPRINT O$;
LPRINT "expected waiting time per";_
   " entering customer = ";:_
LPRINT using "######.## hours";WAITTTL2/SAMPSIZE%

LPRINT
FOR X=1 TO S2%
  LPRINT O$;
  LPRINT "percentage server ";
  LPRINT USING "## is busy = ##.# %";_
         x,BUSY2(X)/PER2*100
NEXT X
LPRINT:LPRINT O$;"LEVEL DS/GS :":LPRINT
LPRINT "         no of servers       = ";
LPRINT USING "######";S3%

LPRINT "         no of delays        = ";
LPRINT USING "######";QD3%
LPRINT "         max queue length    = ";:_
LPRINT using "###### customers";MAX3%
LPRINT "         total waiting time = ";:_
LPRINT using "########.##  hours";WAITTTL3
LPRINT:LPRINT O$;
LPRINT "average  waiting time per  waiting ";
IF QD3%=0 THEN
  LPRINT " customer =      0.00 hours"
ELSE
  LPRINT using "customer = ######.## hours";_
         WAITTTL3/QD3%
END IF
```

```
LPRINT O$;
LPRINT "expected waiting time per";
LPRINT " entering customer = ";
LPRINT using "######.## hours";WAITTTL3/SAMPSIZE%

LPRINT
FOR X=1 TO S3%
  LPRINT O$;
  IF PER3=0 THEN_
    LPRINT "percentage server ";:_
    LPRINT USING "## is busy =  0   %";X_
  ELSE_
    LPRINT "percentage server ";:_
    LPRINT USING "## is busy = ##.# %";_
           x,BUSY3(X)/PER3*100
NEXT X
LPRINT:LPRINT O$;"LEVEL DEPOT :":LPRINT
LPRINT "        no of servers       = ";
LPRINT USING "######";S4%
LPRINT "        no of delays        = ";
LPRINT USING "######";QD4%
LPRINT "        max queue length    = ";
LPRINT using "###### customers";MAX4%
LPRINT "        total waiting time = ";
LPRINT USING "########.##  hours";WAITTTL4
LPRINT:LPRINT O$;
LPRINT "average  waiting time per waiting";_
       "  customer = ";
IF QD4%=0 THEN
  LPRINT "      0.00 hours"
ELSE
  LPRINT USING "######.## hours";WAITTTL4/QD4%
END IF


LPRINT O$;
LPRINT "expected waiting time per entering";_
       " customer = ";
LPRINT USING "######.## hours";WAITTTL4/SAMPSIZE%

LPRINT
FOR X=1 TO S4%
  LPRINT O$;
  IF PER4=0 THEN_
    LPRINT USING "percentage server ##";X;:_
    LPRINT " is busy =  0.0 %"_
  ELSE_
    LPRINT USING "percentage server ##";X;:_
    LPRINT USING " is busy = ##.# %";_
           BUSY4(X)/PER4*100
NEXT X
```

```
      LPRINT::LPRINT O$;
      LPRINT "Total elapsed time for ";
      LPRINT USING "##### failures = ###.## weeks";_
            SAMPSIZE%,PER2/WEEK:LPRINT
      LPRINT O$;
      LPRINT"Total number of 4 week periods for ";:_
      LPRINT USING "#### failures";SAMPSIZE%;
      LPRINT USING " = ######.#";PERNUM
      LPRINT CHR$(12)
REM
REM order spare part consumption
REM
      CLS:LOCATE 15,22
      PRINT "Ordering of Part Consumption Times"
      FOR X%=1 TO ITEMNO%
        CURR%(X%)=0
        MIN%(X%)=0
      NEXT X%
      FOR X%=1 TO 500
        T1(X%)=0:T2%(X%)=0
      NEXT X%
      COUNT%=0
      FOR N%=1 TO SAMPSIZE%
        IF A(4,N%)=0 THEN nds
        FF=-MULT
        D1=A(1,N%):A=A(4,N%)
        GOSUB linkparts
        IF ITEM%(A(4,N%))=9 THEN GOTO nextfail
        IF A(5,N%)=0 THEN Z=A(4,N%):GOTO nextds
        FF=-MULT
        D1=A(1,N%):A=A(5,N%):GOSUB linkparts
        FF=MULT
        D1=A(2,N%)+A(8,N%)/2:A=A(4,N%)
        GOSUB linkparts
        IF ITEM%(A(5,N%))=9 THEN GOTO nextfail
        IF A(6,N%)=0 THEN Z=A(5,N%):GOTO nextds
        FF=-MULT
        D1=A(1,N%):A=A(6,N%):GOSUB linkparts
        FF=MULT
        D1=A(2,N%)+A(8,N%)/2:A=A(5,N%)
        GOSUB linkparts
        IF ITEM%(A(6,N%))=9 THEN GOTO nextfail
        Z=A(6,N%)
nextds:
        TA=B(10,N%):TT=B(11,N%)
        FF=MULT
        D1=B(4,N%)+TA/2+TT:A=Z:GOSUB linkparts
nds:
        TA=B(10,N%):TT=B(11,N%)
        IF B(6,N%)=0 THEN ndp
        FF=-MULT
```

```
        D1=B(3,N%):A=B(6,N%)
        GOSUB linkparts
        IF ITEM%(B(6,N%))=9 THEN GOTO nextfail
        IF B(7,N%)=0 THEN Z=B(6,N%):GOTO nextdp
        FF=-MULT
        D1=B(3,N%):A=B(7,N%):GOSUB linkparts
        FF=MULT
        D1=B(4,N%)+TA/2:A=B(6,N%)
        GOSUB linkparts
        IF ITEM%(B(7,N%))=9 THEN GOTO nextfail
        IF B(8,N%)=0 THEN Z=B(7,N%):GOTO nextdp
        FF=-MULT
        D1=B(3,N%):A=B(8,N%):GOSUB linkparts
        FF=MULT
        D1=B(4,N%)+TA/2:A=B(7,N%):GOSUB linkparts
        IF ITEM%(B(8,N%))=9 THEN GOTO nextfail
        Z=B(8,N%)
nextdp:
        TA=C(10,N%):TT=C(11,N%)
        FF=MULT
        D1=C(4,N%)+TA/2+TT:A=Z
        GOSUB linkparts
ndp:
        TA=C(10,N%):TT=C(11,N%)
        IF C(6,N%)=0 THEN nextfail
        FF=-MULT
        D1=C(3,N%):A=C(6,N%)
        GOSUB linkparts
        IF ITEM%(C(6,N%))=9 THEN GOTO nextfail
        IF C(7,N%)=0 THEN nextfail
        FF=-MULT
        D1=C(3,N%):A=C(7,N%):GOSUB linkparts
        FF=MULT
        D1=C(4,N%)+TA/2:A=C(6,N%)
        GOSUB linkparts
        IF ITEM%(C(7,N%))=9 THEN GOTO nextfail
        IF C(8,N%)=0 THEN nextfail
        FF=-MULT
        D1=C(3,N%):A=C(8,N%):GOSUB linkparts
        FF=MULT
        D1=C(4,N%)+TA/2:A=C(7,N%)
        GOSUB linkparts
nextfail:
      NEXT N%
REM
REM compute stock levels
REM
      FOR X%=1 TO ITEMNO%
        CURR%(X%)=STOCK%(X%)
      NEXT X%
      OUT1%=0:OUT2%=1
```

```
     FOR X%=1 TO COUNT%
sl1:
        IF T1(OUT2%)=0 THEN sl2
        IF OUT2%>OUT1% THEN sl2
        IF T1(OUT2%)<=P1(X%) THEN
          CURR%=CURR%(T2%(OUT2%))
          CURR%=CURR%+T3%(OUT2%)
          CURR%(T2%(OUT2%))=CURR%
          INCR OUT2%
          GOTO sl1
        END IF
sl2:
        CURR%(P2%(X%))=CURR%(P2%(X%))+P3%(X%)
        IF CURR%(P2%(X%))=LITTLES%(P2%(X%)) THEN
          IF ITEM%(P2%(X%))=9 THEN
           INCR OUT1%
           T1(OUT1%)=P1(X%)+AST
           T2%(OUT1%)=P2%(X%)
           T3%(OUT1%)=STOCK%(P2%(X%))-LITTLES%(P2%(X%))

          END IF
        END IF
        IF CURR%(P2%(X%))<MIN%(P2%(X%)) THEN_
          MIN%(P2%(X%))=CURR%(P2%(X%))
     NEXT X%
     FOR X%=1 TO ITEMNO%
        IF MIN%(X%)<0 THEN_
          IF ITEM%(X%)<9 THEN MIN%(X%)=0
     NEXT X%
     LPRINT O$;_
     "Summary of Item Consumption and Stock Levels";
     LPRINT USING "     PAGE ##";PAGE%:INCR PAGE%
     LPRINT O$;_
     "============================================="
     LPRINT:LPRINT
     LPRINT O$;O$;_
     "STOCK NO.   INITIAL STOCK";_
     "   FINAL STOCK   MIN. STOCK"
    LPRINT O$;"    _________";_
     "_______________________________________________"

    LPRINT
    FOR N%=1 TO ITEMNO%
        LPRINT USING "        ####  ";N%;
        LPRINT ITEM$(N%);:LPRINT USING"     ####";_
               STOCK%(N%);
        LPRINT USING "           ####          ####";_
               CURR%(N%),MIN%(N%)
        MIN%(N%)=STOCK%(N%)
    NEXT N%
    CLS:LPRINT CHR$(12)
```

```
REM
REM close sensitivity loops
REM
     IF CHOICE%=0 THEN pend
     IF BUSY2(1)/PER2*100<10 THEN EXIT FOR
    NEXT S2%
    IF BUSY3(1)/PER3*100<10 THEN EXIT FOR
  NEXT S3%
  IF BUSY4(1)/PER4*100<10 THEN EXIT FOR
NEXT S4%
pend:
LPRINT:LPRINT
LPRINT "            Program End Time is : ";TIME$
LOCATE 20,30:PRINT "*******************"
LOCATE 21,30:PRINT "**  Program End  **"
LOCATE 22,30:PRINT "*******************"
LPRINT CHR$(12)
END
REM                 #########################
REM                 #  END OF MAIN PROGRAM  #
REM                 #########################
REM
REM ***************************************************




REM
REM Subroutine CHOOSE PROGRAM MODE
REM
choice:
REM
CLS:LOCATE 7,25
PRINT "**************************"
LOCATE 8,25
PRINT "**   Program Mode Menu  **"
LOCATE 9,25
PRINT "**************************"
LOCATE 13,16
PRINT "( 0 ) ... Check Queuing Behavior without"
LOCATE 14,16
PRINT "          Sensitivity Analysis (faster)"
LOCATE 16,16
PRINT "( 1 ) ... Check Queuing Behavior with"
LOCATE 17,16
PRINT "          Sensitivity Analysis"
LOCATE 19,12
PRINT "  ============================================"
LOCATE 22,29
PRINT "Default Value is 0."
```

```
itr:
LOCATE 23,18
PRINT "To accept press ENTER,";
INPUT " else enter value  : ",CHOICE%
IF CHOICE%<0 OR CHOICE%>1 THEN
  LOCATE 23,61
  PRINT "                "
  GOTO itr
END IF
RETURN
REM
REM *****************************************************


REM
REM Subroutine READ INITIAL DATA
REM
getdata:
REM
SYS%=VAL(S1$)
RL%=VAL(RL$)
I%=VAL(I1$)
TA=VAL(TA$)
TT=VAL(TT$)
IF RL%=2 THEN
  IF PTR%(SYS%)=0 THEN PTR%(SYS%)=IC%:F%=0:_
     ARRTIME(SYS%)=FTIME(SYS%)+TA/2+TT
  IF F%=1 AND NPTR%(SYS%)=0 THEN_
   IF IC%>PTR%(SYS%) THEN NPTR%(SYS%)=IC%
END IF
IF RL%=3 THEN_
  IF ARRTIME(SYS%)=0 THEN B(1,N%)=FTIME(SYS%)
IF RL%=4 THEN_
  IF ARRTIME(SYS%)=0 THEN C(1,N%)=FTIME(SYS%)
RETURN
REM
REM *****************************************************
```

```
REM
REM Subroutine SORT ARRIVAL TIMES ON LEVEL ORG MN
REM
orgsort:
REM
IF ARRTIME(X%)>=ARRTIME(PE%) THEN_
   SPTR%(PE%)=X%:SPTR%(X%)=IC%:PE%=X%:GOTO endsort
IF ARRTIME(X%)<ARRTIME(PS%) THEN_
   SPTR%(X%)=PS%:PS%=X%:INCR SPTR%(PE%):GOTO endsort
P%=PS%
iorg:
H%=P%:P%=SPTR%(P%)
IF ARRTIME(X%)>ARRTIME(P%) THEN iorg
SPTR%(X%)=P%:SPTR%(H%)=X%
endsort:
RETURN
REM
REM ***********************************************************


REM
REM Subroutine SORT ARRIVAL TIMES ON LEVEL DS/GS
REM
dssort:
REM
IF B(2,X%)>=B(2,PE%) THEN_
   B(5,PE%)=X%:B(5,X%)=IC%:PE%=X%:GOTO enddssort
IF B(2,X%)<B(2,PS%) THEN_
   B(5,X%)=PS%:PS%=X%:INCR B(5,PE%):GOTO enddssort
P%=PS%
ids:
H%=P%:P%=B(5,P%)
IF B(2,X%)>B(2,P%) THEN ids
B(5,X%)=P%:B(5,H%)=X%
enddssort:
RETURN
REM
REM ***********************************************************
```

```
REM
REM Subroutine SORT ARRIVAL TIMES ON LEVEL DEPOT
REM
dpsort:
REM
IF C(2,X%)>=C(2,PE%) THEN_
   C(5,PE%)=X%:C(5,X%)=IC%:PE%=X%:GOTO enddpsort
IF C(2,X%)<C(2,PS%) THEN_
   C(5,X%)=PS%:PS%=X%:INCR C(5,PE%):GOTO enddpsort
P%=PS%
idp:
H%=P%:P%=C(5,P%)
IF C(2,X%)>C(2,P%) THEN idp
C(5,X%)=P%:C(5,H%)=X%
enddpsort:
RETURN
REM
REM ***************************************************


REM
REM Subroutine DETERMINE TYPE OF FAILED ITEM
REM
partno:
REM
FOR P%=1 TO ITEMNO%
  IF NSN$=ITEM$(P%) THEN ITEM%(P%)=VAL(I1$):EXIT FOR
NEXT P%
RETURN
REM
REM ***************************************************


REM
REM Subroutine CHECK IF ITEM AVAILABLE
REM
partcheck:
REM
FOR X%=1 TO ITEMNO%              :REM check which part
  IF ITEM$(X%)=D$(1) THEN RETURN
NEXT X%
RETURN
REM
REM ***************************************************
```

```
REM
REM Subroutine READ INDIVIDUAL FAILURE DATA
REM
gdata:
REM
SYS%=VAL(S1$)
RL%=VAL(RL$)
I%=VAL(I1$)
RT=VAL(RT$)
TA2=VAL(TA$)
TT2=VAL(TT$)
IF RL%=2 THEN
  M2=M2+RT
  A(8,N%)=TA2
  GOSUB partno
  IF A(4,N%)=0 THEN A(4,N%)=P%:GOTO e1
  IF A(5,N%)=0 THEN A(5,N%)=P%:GOTO e1
  A(6,N%)=P%:GOTO e1
END IF
IF RL%=3 THEN
  B(9,N%)=B(9,N%)+RT
  B(10,N%)=TA
  B(11,N%)=TT
  GOSUB partno
  IF B(6,N%)=0 THEN B(6,N%)=P%:GOTO e1
  IF B(7,N%)=0 THEN B(7,N%)=P%:GOTO e1
  B(8,N%)=P%:GOTO e1
END IF
IF RL%=4 THEN
  C(9,N%)=C(9,N%)+RT
  C(10,N%)=TA
  C(11,N%)=TT
  GOSUB partno
  IF C(6,N%)=0 THEN C(6,N%)=P%:GOTO e1
  IF C(7,N%)=0 THEN C(7,N%)=P%:GOTO e1
  C(8,N%)=P%
END IF
e1:
IF I%=9 THEN MTTF=VAL(MTTF$)
RETURN
REM
REM ********************************************************
```

```
REM
REM Subroutine VARIABLE INPUT
REM
varin:
REM
IF CHOICE%=0 THEN srf
CLS:LOCATE 10,15:SENSLIM=0.1
PRINT "Enter sensitivity analysis bounds";
PRINT " in per cent"
LOCATE 16,15
PRINT "             The default value is 10."
LOCATE 22,17
PRINT "To accept default press ENTER,";
INPUT " else type value ",RF
IF RF>0 THEN SENSLIM=RF/100
srf:
CLS:LOCATE 10,20
PRINT "Enter the stockout risk factor r"
LOCATE 12,20
PRINT "       ( 0.0 < r < 1.0 )"
LOCATE 16,20
PRINT "    The default value is 0.9."
LOCATE 22,15
PRINT "To accept default press ENTER,";
INPUT " else type value",RF
IF RF>0 THEN R=RF ELSE R=0.9
CLS:LOCATE 10,10
PRINT "Enter ave. shipping time";
PRINT " for ordered items in days"
LOCATE 16,10
PRINT "             The default value is 5 days."
LOCATE 22,12
PRINT "To accept default press ENTER,";
INPUT " else type value",RF
IF RF>0 THEN AST=RF ELSE AST=5
RETURN
REM
REM ***************************************************
```

```
REM
REM Subroutine POISSON DISTRIB.: MINIMUM STOCK LEVEL
REM
poisson:
REM
K=0
LT=LAMBDA*AST
E=EXP(-LT)
CUM=E
WHILE CUM<R
  INCR K
  CUM=CUM+((LT^K)/K)*E
WEND
LITTLES%(X)=K
RETURN
REM
REM ****************************************************
```

```
REM
REM Subroutine PRINT NUMBER OF SERVERS
REM
sernum:
REM
CLS:LOCATE 6,20
PRINT "number of servers for level Depot  : ";
PRINT USING "###";S4%
LOCATE 8,20
PRINT "number of servers for level DS/GS  : ";
PRINT USING "###";S3%
LOCATE 10,20
PRINT "number of servers for level Org Mn : ";
PRINT USING "###";S2%
LOCATE 15,27:PRINT "Compute Queueing Behavior"
RETURN
REM
REM ****************************************************
```

```
REM
REM Subroutine ORDER PARTS CONSUMPTION
REM
linkparts:
REM
C%=COUNT%:INCR COUNT%
IF COUNT%=1 THEN_
   P1(COUNT%)=D1:P2%(COUNT%)=A:P3%(COUNT%)=FF:GOTO lp3
IF D1>=P1(C%) THEN_
   P1(COUNT%)=D1:P2%(COUNT%)=A:P3%(COUNT%)=FF:GOTO lp3
lp1:
IF D1>=P1(C%-1) THEN
lp2:
   H1=P1(C%):H2%=P2%(C%):H3%=P3%(C%)
   P1(C%)=D1:P2%(C%)=A:P3%(C%)=FF
   D1=H1:A=H2%:FF=H3%:INCR C%
   IF C%<=COUNT% THEN lp2 ELSE lp3
ELSE
   DECR C%:GOTO lp1
END IF
lp3:
RETURN
REM
REM ******************************************************
```

LIST OF REFERENCES

1. Opus VII Manual, 3rd ed., Systecon AB, Stockholm, 1978

2. Larson, H. L., Introduction To Probability Theory And Statistical Inference, John Wiley & Sons, 1982.

3. Morgan, B. J. T., Elements Of Simulation, Chapman And Hall, 1984.

4. Barr, D. R. and Zehna, P. W., Probability, p. 162, Brooks/Cole Publishing Company, 1971

5. Ross, S. M., Introduction To Probability Models, pp. 168-183, Academic Press, 1984

6. Borland International Inc., Turbo Basic, 1987

7. Combined Arms And Services Staff School, E709, Organization Of The Army In The Field, Fort Leavenworth, Kansas, November 1983

8. Deutscher Bundeswehr Kalender, Grundwerk, Walhalla & Praetoria Verlag, 1985

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22304-6145	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, California 93943-5002	2
3.	Defense Logistics Studies Information Exchange U.S. Army Logistics Management Center Fort Lee, Virginia 23801	1
4.	Chief of Naval Operations (OP-04) Navy Department Washington, D.C. 20350-2000	2
5.	Chief of Naval Operations (OP-91) Navy Department Washington, D.C. 20350-2000	2
6.	Dokumentationszentrum der Bundeswehr Friedrich Ebert Allee 34 5300 Bonn 1 Federal Republic of Germany	2
7.	Bundesminister der Verteidigung StAL FueH VII Postfach 1328 5300 Bonn 1 Federal Republic of Germany	2
8.	Dr. F. Russel Richards Evaluation Technology Inc. 2150 Garden Road, Suite B3 Monterey, California 93940	1
9.	Assoc. Professor Alvin F. Andrus Code 55As Naval Postgraduate School Monterey, California 93943	1

10. Major Dipl. Ing. Robert Bund 1
SMC 1677
Naval Postgraduate School
Monterey, California 93943

11. Hptm Dipl. Ing. Rainer Seth 4
Schule Technische Truppe 1 und
Fachschule des Heeres fuer Technik
Lehrgruppe E, XIV. Inspektion
Birkengangstr. 140
5180 Eschweiler
Federal Republic of Germany

Thesis
S41857 Seth
c.1 Evaluation of multi-level support structure requirements for new weapon systems.