GPO PRICE $ ______

CFSTI PRICE(S) $ ______

Hard copy (HC) $4.00

Microfiche (MF) 1.00

ff 653 July 65

# TECHNICAL REPORT NO. 6

# A COMPUTER PROGRAM FOR THE SIMULATION OF FAILURE — RESPONSIVE SYSTEMS

Contract NASw - 572 April 1966

**Reference WGD - 38521**

N66 26880

| | |
|---|---|
| (ACCESSION NUMBER) | (THRU) |
| 131 (PAGES) | 1 (CODE) |
| CR-75277 (NASA CR OR TMX OR AD NUMBER) | 08 (CATEGORY) |

FACILITY FORM 602


Westinghouse Defense and Space Center
Surface Division

P. O. BOX 1897 BALTIMORE, MD. 21203

Technical Report No. 6

# A COMPUTER PROGRAM FOR THE SIMULATION OF FAILURE-RESPONSIVE SYSTEMS

Contract NASw-572
Reference WGD-38521

by
J. M. Hannigan

April 1966

MDE 6566

## TABLE OF CONTENTS



## APPENDICES

# I. INTRODUCTION

When analyzing the relaibility of complex electronic systems, it is usually desirable to obtain exact mathematical expressions for the reliability. This is especially true in the analysis of redundant systems, since the main goal of redundancy is increased reliability. As the complexity of redundant systems is increased, however, there is a rapid increase in the number of calculations required to determine the exact reliability expressions.

As described in an earlier report[1], failure responsive systems are multiple-line redundant systems with the added capability to partially reorganize their redundant subsystems in response to the random occurrence of internal failure patterns. Because of the added reorganizational capability, the analytical reliability expressions for these systems become extremely complex. Furthermore, a completely different reliability expression must be derived for each different reorganizational strategy class. In general, then, it is virtually impossible to analyze failure responsive systems by deriving exact mathematical reliability equations.

Because of these restrictions, a generalized computer simulation program was developed to facilitate a Monte Carlo approach to the reliability analysis problem. The simulation program may be employed in the analysis of failure responsive systems, multiple line systems, or other modularly redundant systems.

The purpose of this report is to make the simulation program available to those performing research on similar redundant systems. The report contains a complete description of the program, including input data formats, output information, a descriptive list of program variables, detailed flow diagrams, and a Fortran IV program listing. This report assumes a basic familiarity with the FORTRAN IV language and with the failure responsive system reorganization procedures.

---

[1] "Analysis and Development of Failure Responsive System Organizations" by C. G. Masters, Special Technical Report No. 5 of Contract NASw-572, December, 1964.

## II. GENERAL DESCRIPTION OF THE SIMULATION PROGRAM

The simulation program is capable of simulating any number of complete missions of a failure responsive or similar redundant system. It can also simulate any number of different system organizations in a given computer run by the addition of an input data deck for each system to be simulated. The program is composed of an executive routine, MAIN, and ten subroutine subprograms, all written in FORTRAN IV. A general flow diagram of the program is shown in figure 1.

### A. The System Representation Matrix

The program begins simulation by setting up a three dimensional matrix representation of the system. The matrix is shown diagrammatically in figure 2. The first dimension of the matrix represents the order of redundancy, and the second dimension, the number of stages in the system. Each element of the "front" plane of the matrix therefore represents a single subsystem and each column represents one stage of the system. The third dimension, KTEMP, is used to store all the information necessary to specify the instantaneous operating state of each subsystem.

Referring to this third dimension, the first two data storage locations for each subsystem contain the limits of an interval of numbers located between zero and one. The size of the interval is determined by the ratio of the failure rate of the associated subsystem, to the sum of failure rates of all of the subsystems. Each subsystem has a different portion of the zero-to-one range, and the entire range is allocated to the system. Failures are generated by drawing a random number from a set uniformly distributed between zero and one, and designating as failed the subsystem within whose interval the number falls.

The third storage location for each subsystem in the system representation matrix stores an identification (ID) number assigned to that subsystem. The fourth location stores a "failure order" number, which indicates, if non-zero, that the subsystem has failed, plus the order of failures in each stage.

The remaining locations in the matrix contain the "spare lists," or lists of identification numbers which specify which subsystems will be used as spares for each subsystem, if it fails, as well as the order in which the "spares" will be selected.

### B. The Simulation Procedure

The simulation proceeds by generating a random number, and designating the corresponding subsystem as failed. The program then determines whether or not the system is still operational, according to the operational criteria which were read in as data. For

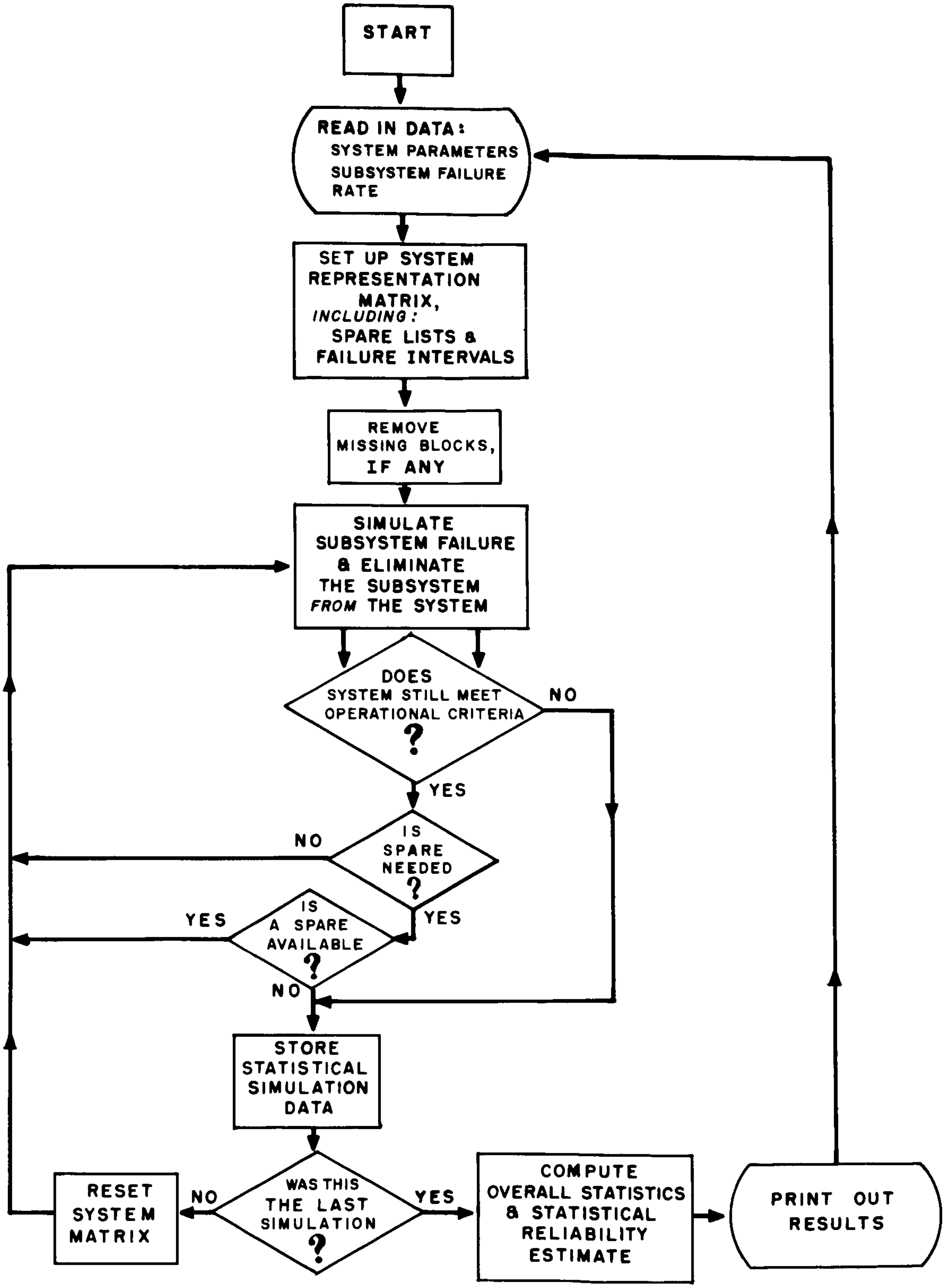



Figure 1. Summary Flow Diagram of Simulation Program

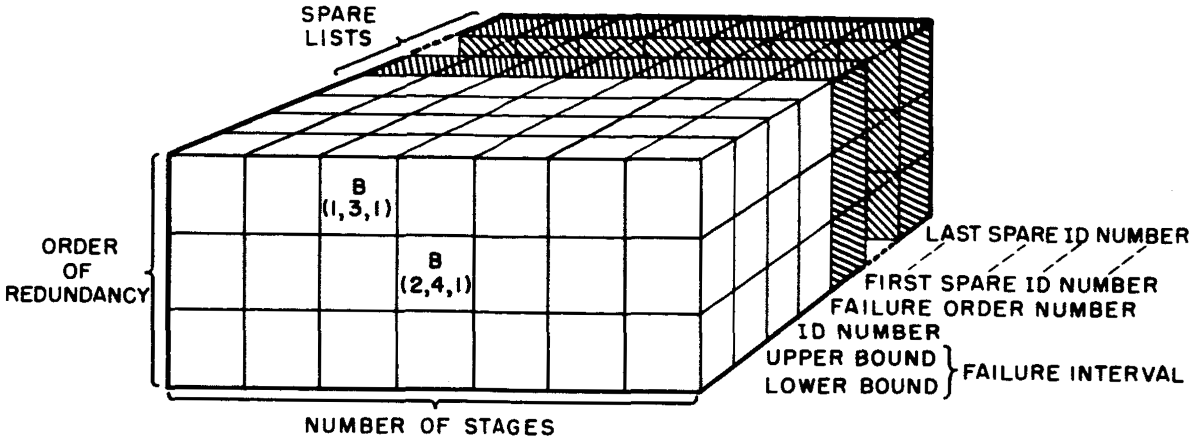



Figure 2. Diagram of System Representation Matrix

example, if the system is operational, the program determines whether the stage requires a spare subsystem from another stage in order to perform a "vote." If not, another failure is generated somewhere in the system. If the stage does require a spare, the program scans the spare list of the failed subsystem for a replacement from another stage. If none is available, the simulation terminates and a new mission simulation is begun. If a spare is available, it is moved to the new location and another simulated failure is generated.

The program contains a provision for generating a time-to-failure when a subsystem has failed. This is done by drawing a number from an exponentially distributed set of numbers. There is also an option for printing either the simulation results only, or a running synopsis of the response of the system, from time zero to failure.

C. Compilation of Results

After a predesignated number of simulations (usually 500-1000) of a system, the program computes an estimate of the reliability of the system using the results of all simulations. There is an option for computing either the 90% reliability time, or the entire reliability curve, from time = 0 to 5000 Hrs.

When the computations are completed, the program returns to its starting point and reads in the data deck specifying the next system to be simulated. If no data are remaining (the last system has been simulated), execution of the program is terminated by the attempt to read through an end-of-file. By this method, any number of systems may be simulated sequentially, simply by the addition of a data deck for each separate system.

## III. DETAILED DESCRIPTIONS OF PROGRAM ROUTINES

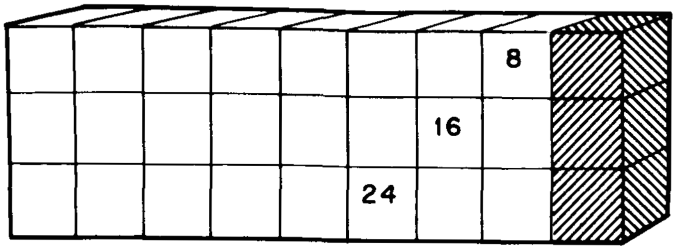
This section contains a detailed description of the operation of the MAIN executive program, followed by descriptions of all of the subroutines, in alphabetical order.

### A. The MAIN Program

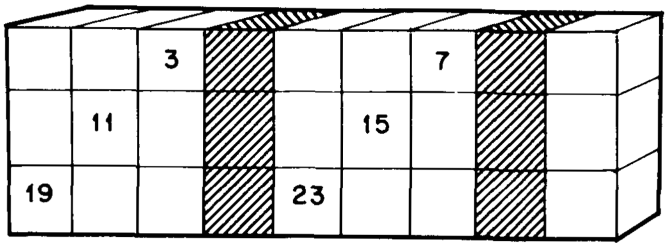
This is the executive routine which sets up the system representation matrix and controls the simulation by performing the majority of subroutine calling, regulating the number of simulations, and computing the statistical results of the simulations. The operation of MAIN begins by setting up the system representation matrix. The program reads in the data which specify the system to be simulated. (Note: Most of this data is printed out, to provide a permanent record, at the beginning of the printout, of what system has been simulated.) The program then calls subroutine ALOCAT, which sets up the limits of the "failure" interval, by which random numbers simulate failures of specific subsystems. Subroutine DET is called, to assign an ID number to each subsystem.

The final operation in setting up the system matrix is the assignment of spare lists to all subsystems in the system. NRL, the Non-Repetitive List flag, controls the type of list to be set up. If NRL = 0, MAIN calls subroutine LIST, which reads in the lists for the subsystems in the last stage, and reproduces the pattern of these lists for the remainder of the subsystems. For example, figure 3(a) illustrates diagrammatically a simple list which might be read in for the last stage in a nine-stage system. The subsystems had previously been numbered consecutively from left to right, with ID numbers 1 through 9 on the top row, 10 through 18 on the second row, and 19 through 27 on the bottom row. The shaded subsystems in the last stage have the read-in spare list consisting of subsystems 8, 16, and 24, in that order. Figure 3(b) illustrates two of the reproduced spare list patterns. The second last stage has spares 7, 15, and 23, and the fourth stage has spares 3, 11, and 19. The spare list pattern would be reproduced in this manner for each of the remaining stages in the system. When NRL = 1, MAIN calls subroutine RANLIS, which generates a random spare list for each subsystem. If NRL = 2, MAIN reads in a separate list for each subsystem from input data cards.

Before the actual simulation, the program removes any "missing blocks" from the matrix. These are subsystems which are considered either failed, or simply not a part of the system at time = 0. This provision has been included in the program to provide the investigator with a wide range of system configurations different from the simple rectangular matrix. (For example, a system whose stages contain unequal orders of redundancy may be simulated with this option.) The program removes the subsystems by calling subroutine




(a) READ-IN LIST




(b) TWO OF THE REPRODUCED PATTERNS

Figure 3. Example of List Pattern Reproduction

FAIL. This subroutine designates the subsystems as failed by setting their "failure order numbers" to non-zero values. FAIL then sets the subsystems' failure intervals to zero to prevent future failures of the same subsystem. The MAIN program then subtracts the failure rates of the missing blocks from the sum of the failure rates of the whole system, and computes the average failure rate for the modified system. The list of missing blocks is printed out.

To begin the simulation, MAIN sets up a duplicate matrix, C, which is used to reset the system representation matrix, B, before the next simulation. Subroutine FAIL is called to randomly generate a subsystem failure. If the failure has not caused system failure, MAIN determines if the stage that experienced the failure requires a spare to continue operation. If it does not, a new failure is generated. If it does, either subroutine SWITCH or subroutine SBACK is called to check the failed subsystem's spare list for an available spare. The determination of the proper subroutine to be called here is explained in the sections on subroutine SWITCH and subroutine SBACK. In either case, if none of the spares is available, the system is failed. If one is found, the simulation continues with the generation of another failure.

After each simulation, MAIN computes the total number of subsystem failures absorbed before system failure, and the mean time between subsystem failures. Then the system representation matrix B is reset to its initial condition and another simulation is begun. The number of simulations to be performed on a system is specified by the input data constant, NR. At the end of this specified number, the mean-time-to-system failure and the average number of subsystem failures absorbed are computed.

Using the results of all of the simulations of a system, the MAIN program computes the estimate of the system reliability, which provides a single-valued statistical estimate of system performance by which different systems may be compared. There are two options for computing the reliability. The first, which is the single-valued reliability estimate, involves the statistical estimation of the time at which the system reliability falls below 0.90. The second option is actually an extension of the first - the program computes the estimated reliability for every 100 hours between zero and 5000 hours.

1. Method of Computing the Statistical Reliability Estimate

From the results of all of the simulations, the MAIN program first computes a histogram of the frequency of systems failing with X failures absorbed, where X varies between zero and the maximum number of subsystem failures absorbed in a simulation. Each frequency is divided by the total number of simulations, resulting in a histogram of the percentage of systems failing with X subsystem failures. Each of the frequencies represents the probability that the system will fail with exactly X subsystem failures. The frequencies are then accumulated as X varies between zero and its maximum value, resulting in a list of the percentages, F(x), of systems failing with X or less subsystem failures. Each of the values is therefore the observed conditional probability that the system has failed, given that X of its subsystems have failed.

Now, the probability that exactly X subsystems have failed at time t is given by the binominal equation:

$$P(X, t) = \binom{N}{X} \left(1-e^{-\lambda t}\right)^{X} \left(e^{-\lambda t}\right)^{N-X},$$

where N represents the total number of subsystems, and $\lambda$, the average subsystem failure rate. The two conditional probabilities F(X) and P(X, t) can be combined to produce the statistical estimate of the reliability at time t,

$$R(t) = 1 - \sum_{X=0}^{N} F(X)\, P(X, t).$$

The combinational factorial $\binom{N}{X}$ is computed for each value of X, by subroutine COMFAC.

To compute the time of 0.90 reliability, the program uses a method of trial and error, computing the reliability for some time t, and readjusting the value of time for the next trial, based on the reliability estimate obtained with the first time. This process is repeated until the reliability estimate obtained is within the interval 0.90 ±.001.

The value of time used for the first trial is chosen so that the number of trials needed to complete the estimate is relatively small. The first estimate is obtained in the following manner. As mentioned above, the program contains a subroutine, TIME, which computes the time-to-failure for each subsystem failure. During the first 25 simulations, this subroutine is called after each subsystem failure is generated. This subroutine effectively draws a random number from an exponentially distributed set of numbers, based on the assumption that all subsystems have a constant failure rate. The program totals all the times between subsystem failures, obtaining, at the end of a simulation, the total system time to failure. The MAIN program stores in ascending order, the five shortest times of the first 25 simulations. Then the longest of these five, called T1, is used to compute the first approximation to the 0.90 reliability time. The actual time used in the first trial is 0.7 (T1). Subroutine TIME is not used by the program after the first 25 simulations of each system.

2. Results Printout

There are two options for the amount of printout obtainable from the simulations. The "No Sample Format" flag, NSF, specifies which option is used.

a. No Sample Format Option (NSF=1)

First, a listing of most of the input data variables and constants is printed to provide a permanent record with the results identifying the type of system simulated. Next, the portion of the spare lists which was read in as data is printed, as a further identifying record. If a "missing block" list was used, it is printed. The histogram of the number of systems failing with X subsystem failures, computed after all simulations of the system, is printed in tabular form. After each of the trials in the reliability estimation procedure, the value of time used and the probability of system failure (which is 1 - Reliability) are printed, until the 0.10 value is reached. The average number of subsystem failures per simulation is printed below the reliability computation results, followed by a final listing of the system's 0.90 reliability time.

b. Sample Format Option (NSF=0)

This printout option is used mainly as a check on the operation of the program. An extremely large amount of printout results from this option whenever more than a few simulations of a system are requested. All of the information printed under the first option just described is also printed with this option. In addition, the program prints a complete list of the contents of the system representation matrix before the simulation is begun. The first item to be printed is a list of the limits of the "failure interval" for every subsystem, followed by the ID numbers for all of the subsystems. The complete spare list for every subsystem is printed after the lists are stored in the system matrix. During the simulation, the MAIN program prints every subsystem failure which occurs, listing the location in the matrix of the failed subsystem, its ID number, the generated random number which designated the subsystem as failed, the random number specifying the time since the previous failure, and the time itself. The last two quantities will not apply after the first 25 simulations of a given system, since subroutine TIME is not used.

After the occurrence of each system failure, MAIN prints the average time between subsystem failures, the total system time-to-failure, and the total number of subsystem failures. Again, the times do not apply after the first 25 simulations of each system.

If the system fails because no spare is available for a stage in need of a spare, subroutine SWITCH prints a list of all ID numbers in the system, in the order in which they are located in the system representation matrix. This list shows where each subsystem is located in the system.

Examples of the printout obtained from both options are shown in Appendix B.

3. Program Size Limitations

The number of simulations which can be performed for each system is restricted only by the computer memory size. Results from each of the simulations are stored in memory until all simulations of a system have been completed. The results of all of the simulations are used to compute the system performance statistics. Therefore the amount of memory needed to store all individual simulation results is proportional to the number of simulations performed (NR).

The main restriction imposed by memory space is the size of the system itself. The system representation matrix, B, and its duplicate matrix C, utilize the largest portion of the computer memory when large systems are simulated. The program, listed in Appendix C, was run on a UNIVAC 1107 computer with a 32K word memory. The dimension statements at the beginning of the MAIN program provided sufficient memory for the operation of the program. Storage is provided for a twenty stage, order five system, with eleven spares per subsystem.

The computer memory size imposes no restriction on the number of systems simulated during each program run, because a new input data deck is read in for each system, cancelling the input data for the previous system.

In general, for systems much larger than the size mentioned above, and for significantly more than 500 to 1000 simulations per system, a memory larger than the 32K memory may be required.

B. Subroutine ALOCAT

Random failures are generated in the system by selecting a number from a uniformly distributed set of random numbers between zero and one. The number thus generated is used to find the "failed" subsystem by searching through the system representation matrix for the location containing an interval of numbers in which this generated number falls.

The function of subroutine ALOCAT is to assign to each subsystem the interval of numbers, the size of which is proportional to the subsystem's failure rate. The interval, F, for subsystem j, whose failure rate is $\lambda j$, is given by the equation:

$$F = \frac{\lambda_j}{\sum_{i=1}^{x} \lambda_i} ,$$

where X is the total number of subsystems.

When the interval is computed for each subsystem, the limits of the interval are stored in locations B (I, J, 1) and B (I, J, 2). These two locations for each subsystem are examined when a random number is generated, until the "failed" subsystem is located.

Subroutine ALOCAT is called only by the MAIN program.

C. Subroutine COMFAC

This subroutine computes the combinational factorial,

$$FACTL = \binom{X}{Y} = \frac{X!}{Y!\,(X-Y)\,!} .$$

This factorial is used, together with the simulation results, to obtain the statistical reliability estimate. The actual variables used in the above formula are $\binom{IPROD}{LL}$, where IPROD is the total number of subsystems operating at simulated time zero, and LL is a dummy variable representing the number of subsystem failures absorbed at system failure. LL is incremented in the MAIN program from 0 to the maximum number of subsystem failures absorbed, and the combinational factorial is computed for each value of LL.

Subroutine COMFAC is called only by the MAIN program.

D. Subroutine DET

In order to identify each subsystem and to provide a means for listing different subsystems on spare lists, each subsystem carries an identification (ID) number. The function of subroutine DET is to assign an ID number to each subsystem, and store this number in the location B (I, J, 3) corresponding to the subsystem's location. For example, the subsystem in the second row of the fifth stage in matrix B, would have its ID number stored in location B (2, 5, 3).

The subsystems are numbered consecutively, starting at row one, stage one, and proceeding across row one to stage N, then to row 2, stage 1, etc.

Subroutine DET is called only by the MAIN program.

E. Subroutine FAIL

This subroutine generates the random failures throughout the system. Using a starting index, or argument, which has been read in as data, the subroutine calls UDRNRT, a FORTRAN library function for generating random numbers between zero and one. The program then searches through the system representation matrix until it locates the subsystem which has had a failure interval or range of numbers assigned to it, within which the generated random number falls. This is the subsystem which has "failed." If the program is unable to find such a subsystem, it (the subsystem) has either failed previously or is one of the "missing blocks." In this case, a new random number is generated and the search is repeated. When the "failed" subsystem is found, its failure interval is set to zero, and the program then counts the number of remaining subsystems in that stage. If the number is less than the value of input variable, MIN, the minimum number allowed before spare switching is necessary, a flag, S, is set. This will indicate to the MAIN program that a spare is needed for the stage.

The next quantity computed is the "failure order number" of the failed subsystem. Every subsystem in the system is assigned a failure order number, initially zero, stored in location B (I, J, 4). These numbers identify the order in which failures occur in each stage, by the following method. After the first failure in a given stage, the program stores a one in location B (I, J, 4) corresponding to the failed subsystem. After the second failure in the same stage, a two is stored in the corresponding B (I, J, 4), and so forth. (The "I" of this location is necessarily different from that of the failure location immediately preceding the current failure.) When a stage requires a spare, the subsystem which is replaced is not the most recently failed subsystem in the stage, but the previous failure. The failure order numbers are the means of locating this previous failure.

Subroutine FAIL also checks the number of subsystems remaining in a stage to determine whether there are any subsystems remaining in the stage. If not, the system has failed and the "system failure" flag, SYSFA, is set to one. If there is only one remaining subsystem and MIN $\geq 3$, the system has failed. This is because MIN would be as large as three only in a system which could not tolerate the loss of a correct output signal, even for a single clock pulse. But the output would be temporarily incorrect if a stage were down to only one subsystem, even though a spare would be switched in to correct the output. Control of the program is then transferred to MAIN, which uses the information generated.

Subsystem FAIL is called only by the MAIN program.

F. Subroutine LIST

This subroutine reads in the spare list for a subsystem in stage N, and duplicates the pattern of spares for the rest of the system. There are two options for reading in the spare lists, controlled by MM, the common column spare list flag. If MM is zero, each of the subsystems in a column, or stage, has exactly the same spare list as the others, so a list is read in for one subsystem in stage N, and duplicated for the others. If MM equals one, a separate spare list is read in for each subsystem in stage N. For both values of MM, the spare list pattern is reproduced by the program for every subsystem in each row of the system matrix.

Subroutine LIST is called only if NRL=0, and is called only by the MAIN program.

G. Subroutine RANLIS

This subroutine generates random spare lists by calling the FORTRAN function UDRNRT (IRL)[2] which generates random numbers between zero and one. Each number is multiplied by the total number of subsystems, then increased by one, and changed to fixed point. This results in a number which is one of the subsystem ID numbers. If input variable, NOS, equals zero, a subsystem can have subsystems from its own stage on its spare list. If NOS=1, it cannot.

[2] "IRL" is any odd number.

The subroutine generates identical spare lists for all subsystems in a stage. Thus, for an N stage system, N different spare lists will be generated.

Subroutine RANLIS is called only if NRL=1, and is called only by the MAIN program.

H. Subroutine SBACK

This subroutine performs an operation called "spring-back," which in effect, calls a spare to a failed stage in order to "vote" the failed subsystem out of the system. Instead of remaining there after the subsystem is eliminated, however, the spare returns to its original location.

Subroutine SBACK is called by the MAIN program when subroutine FAIL indicates that a spare is needed in a certain stage. SBACK finds the previously failed subsystem in the stage, and scans its spare list. The subroutine contains a provision to "rescan" the spare list under weaker spare movement restraints if no available spare is found during the first scan. The scan and rescan procedure operates as follows.

An input data constant, called NOP1, specifies the number of operating subsystems which must remain in a spare's stage after a spare is taken from the stage. The program locates the first subsystem on the spare list and determines the number of remaining subsystems if the spare were switched. If it is less than NOP1, the next spare on the failed subsystem's list is examined, etc. If the program searches the entire spare list in this manner and finds no available spare, a new input data constant, NOP2, is brought into use. The program goes through the spare list once again, using NOP2 in exactly the same manner as NOP1. It is usually given a lower value. This allows the program to switch a spare from a stage with fewer operating blocks, if no spare were found available during the first scan of the spare list. (If NOP2 is equal to, or greater than, NOP1, the program does not rescan the spare list.)

If no available spare is found during either scan of a spare list, the system is failed. In this case, the system failure flag, SYSFA, is set to one, and control is returned to MAIN.

When the program does find an available spare, the spare is not actually transferred to the failed stage, but it is assumed that it has moved, that the failed subsystem has been eliminated from the voter inputs, and that the spare has returned to its original location as it would in an actual system utilizing a "spring-back" capability.

In the switching of spares, the MAIN program has a choice of two options, controlled by an input data constant, ISBAC. The first option is to call subroutine SBACK, in which case the spare is not actually switched, but simply located. This option is used when

ISBAC=1. If ISBAC=0, the second option is taken, that of calling subroutine SWITCH. In this case the spare would be moved to the failed subsystem's location and left there.

Thus, subroutine SBACK is called only when ISBAC=1, and it is called only by the MAIN program.

I. Subroutine SEARCH

This subroutine is called by subroutine SWITCH, to revise the spare lists after a spare has been switched to a new location. SEARCH performs one of two options, the choice of which is controlled by the dummy variable NREP. When NREP=0, the subroutine searches through all of the spare lists and removes ID number FAILB from every list on which it appears. When NREP=1, the subroutine removes FAILB and replaces it with ID number SPARE. FAILB is used as a dummy variable which may represent either a failed subsystem or the spare which has just replaced it in the system. The latter case would be used if each spare may replace a subsystem only once, in which case it is removed from all spare lists as soon as it has replaced a failed subsystem. SPARE represents the ID number of the spare which has just repaired a failed stage. It replaces the failed subsystem ID on all spare lists when it is desired that spares take over the repair capability of the subsystems which they are replacing.

Subroutine SEARCH may also be called by the MAIN program, if there are any missing blocks to be removed from the system before simulation is initiated. Since these subsystems are not in the system, they must be removed from the spare lists. MAIN calls SEARCH, with FAILB equal to BML(KK), the ID number of a missing block, and SPARE and NREP equal to zero. These three arguments indicate to subroutine SEARCH that ID number BML(KK) is to be removed from all spare lists on which it appears.

Subroutine SEARCH is called only by the MAIN program and subroutine SWITCH.

J. Subroutine SWITCH

This subroutine performs the search for, and switching of spare subsystems to "repair" failed stages. After switching a spare, the subroutine handles the necessary transfer of information within the system representation matrix.

SWITCH performs the search for a spare in exactly the same manner as that of subroutine SBACK. The details of this procedure can be found above, in the section on subroutine SBACK. After finding a spare, the program determines what alterations should be made to the spare lists which contain ID numbers of the failed subsystem and the spare which is replacing it. The type of alteration is controlled by the input data constant ISW2, which can have one of three values. If ISW2=0, the spare's ID number is left on the lists, and only the failed subsystem's ID is removed. When ISW2=1, both the spare and the failed subsystem are removed. If ISW2=2, the spare's ID is removed from the lists, and the failed subsystem's ID is replaced by the spare's ID. SWITCH calls subroutine SEARCH to make the actual spare list changes, using as arguments the necessary ID numbers and values for NREP. This latter variable specifies whether SEARCH should perform a removal or a replacement operation.

After revising the spare lists, SWITCH transfers the spare's information to its new location in the system representation matrix. The first four locations of the spare's information block are always transferred. These locations contain: the two limits of the failure interval computed in subroutine ALOCAT, the spare's ID number, and the spare's "failure order number." The latter location must, of course, be zero, since the spare must be an operating subsystem.

The program next alters the spare list of the spare's new location, if necessary. The input data constant ISW1 specifies which of three operations must be performed. If ISW1=0, the original spare list of the failed subsystem is retained at the location when the spare information is transferred to that location. When ISW1=1, the spare's own spare list is transferred to the new location, and the original list discarded. If ISW1=2, the spare list of the spare's new location is deleted entirely.

Next, subroutine SWITCH returns to the original location of the spare and sets its failure range, which was computed in ALOCAT, to zero. The program then computes the failure order number for the spare's original location, since, as far as that stage is concerned, the spare is essentially failed. SWITCH now returns control to the MAIN program.

As explained in the section on subroutine SBACK, subroutine SWITCH is called only when ISBAC=0; and it is called only by the MAIN program.

K. Subroutine TIME

This subroutine computes the estimated time to failure of a failed subsystem, by using a method of random number generation. Each subsystem is assumed to have a constant failure rate, $\lambda$ . Thus, the probability of continuous operation of all subsystems from time $t=0$ is given by the expression,

$$R(t) = e^{-\sum_{i=1}^{N} \lambda_i t} ,$$

where N is the total number of subsystems in the system. Therefore the probability of the first failure occurring by time t is given by

$$P(1) = 1 - e^{-\sum_{i=1}^{N} \lambda_i t} \; .$$

In order to simulate random failures corresponding to this exponential distribution, the numbers must be chosen from an exponentially distributed set of numbers. However, it is known that a set of random numbers taken from a population uniformly distributed between zero and one may be transformed to a similar set of random numbers belonging to any other distribution. This is what is done in subroutine TIME. The FORTRAN random number generation function UDRNRT (IR) is used to draw a number at random from a set uniformly distributed between zero and one. This number, RNT, is transformed to correspond to an exponentially distributed set by the equation

$$T = \mathrm{Ln(RNT)/(-Y)},$$

where

$$Y = \sum_{i=1}^{N} \lambda_i \; .$$

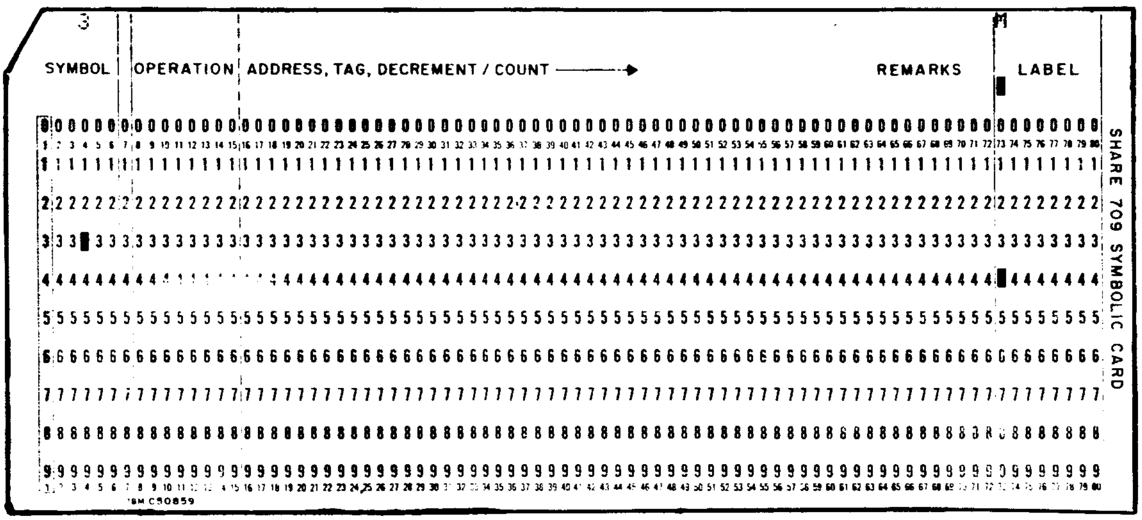
This gives the time to the failure of the first subsystem, with failure rate $\lambda_j$. Similarly, the time from the first to the second failure is generated by subtracting $\lambda_j$ from Y and repeating the procedure.

Subroutine TIME is called only by the MAIN program, after a failure has been generated by subroutine FAIL.

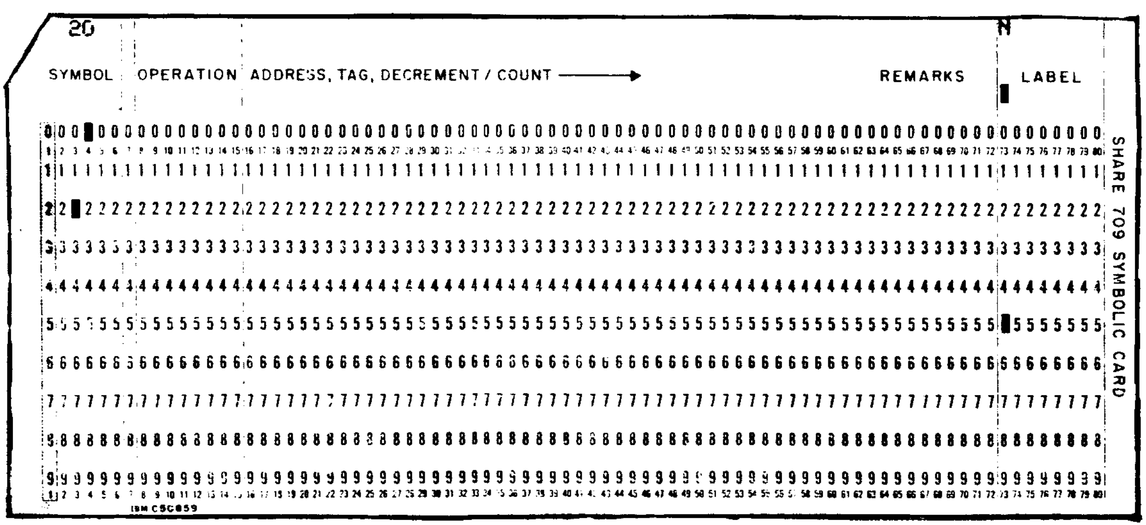
## IV. INPUT DATA CARD FORMAT

This section describes the formats of all of the input data cards, in the order in which they must appear in the data deck.

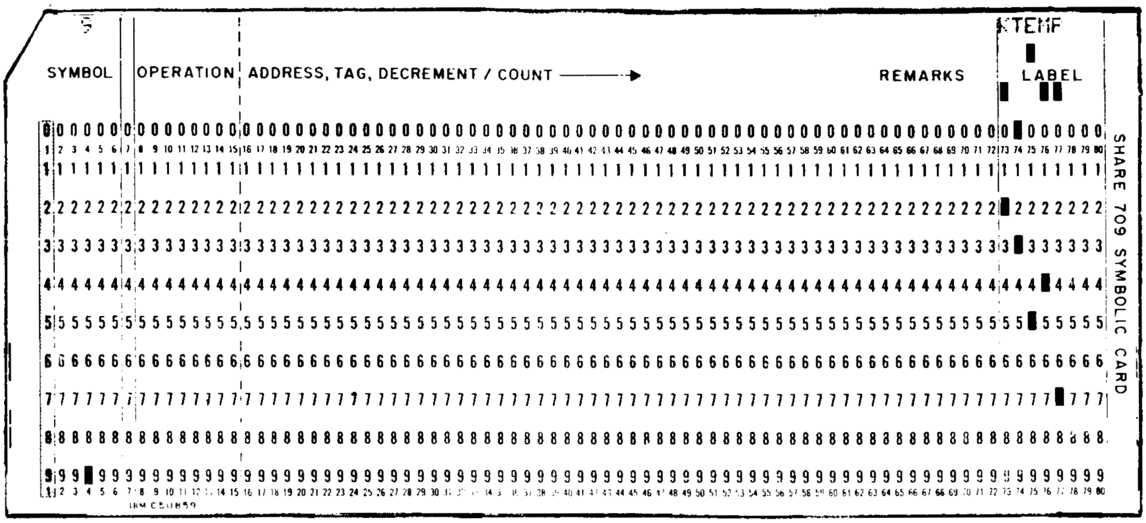
M - Format (I4), right justified. M is equal to, in general, the order of redundancy of the system being simulated. However, for systems containing fractional orders of redundancy or those in which voters or other peripheral circuitry are considered to be separate stages, M might not be equal to the order of redundancy.



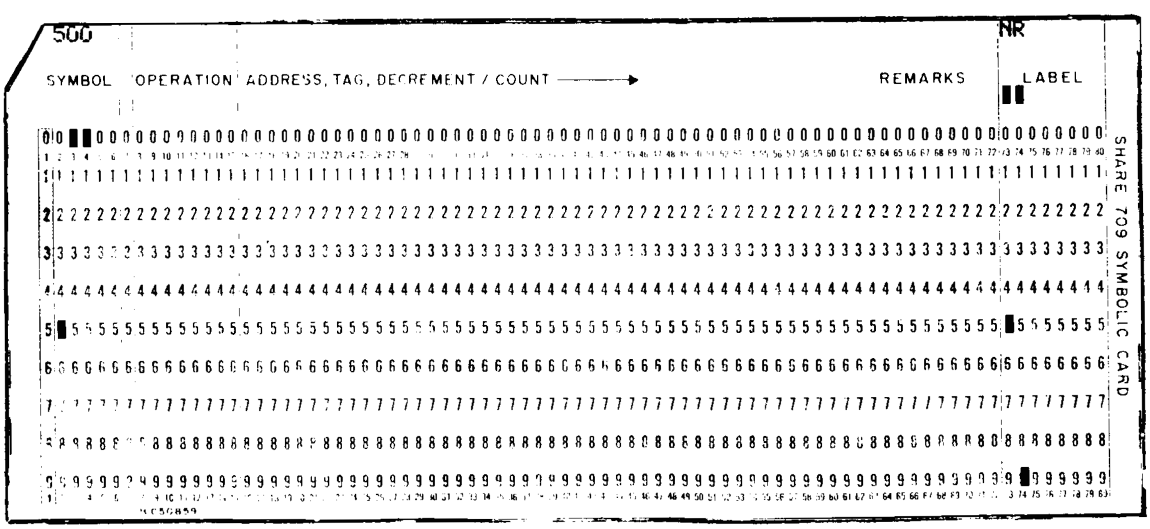

N - Format (I4), right justified. N is equal to the number of stages in the simulated system.



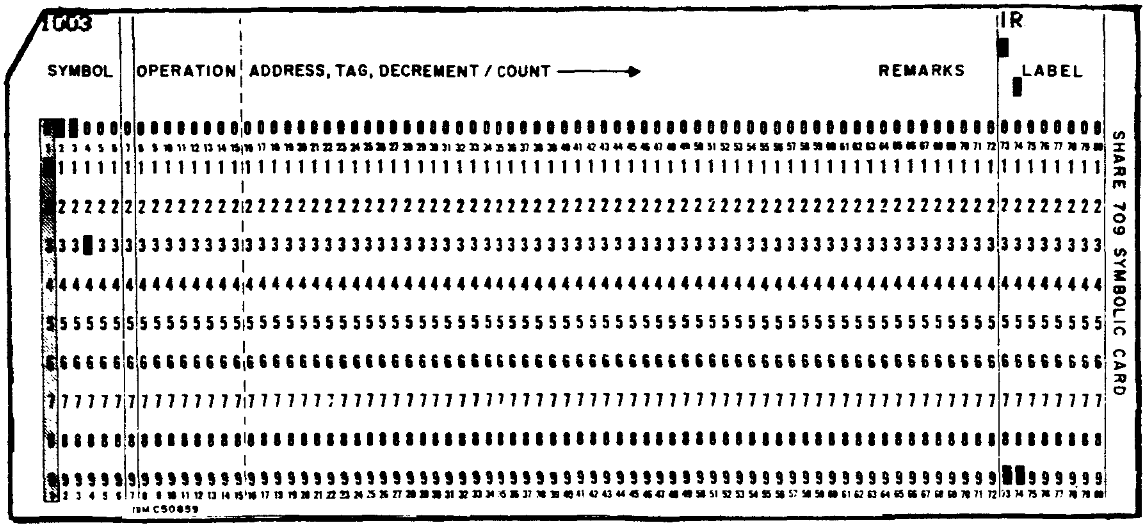
KTEMP - Format (I4), right justified. KTEMP is the third dimension of the system representation matrix B. Its value is equal to the total number of "pieces" of information to be stored for each subsystem. KTEMP = 4 + Length of the longest spare list.



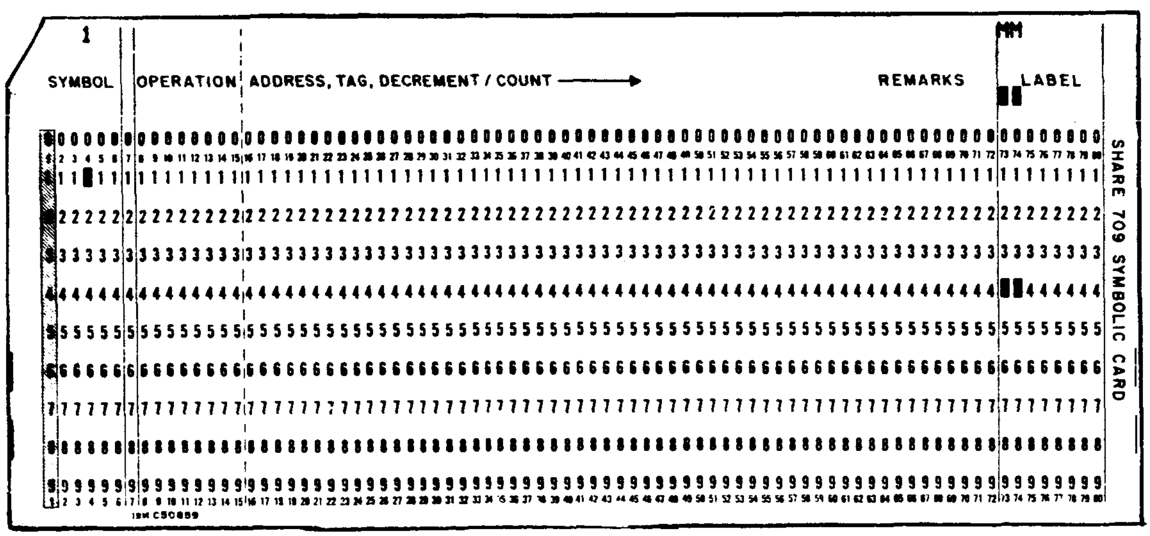

NR - Format (I4), right justified. This is the number of runs, or simulations, to be performed on the system.



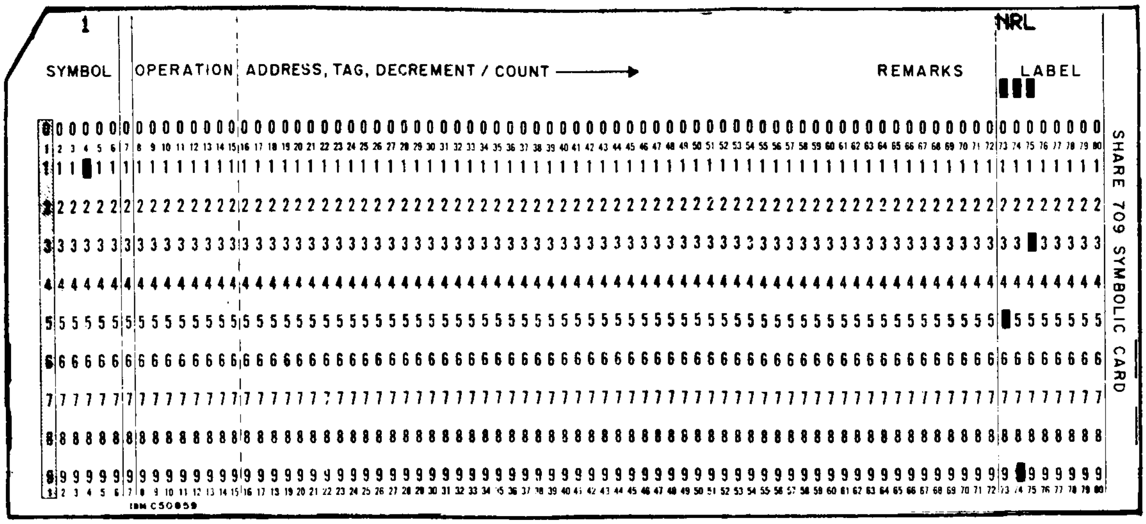
IR - Format (I4), right justified. IR is a starting number for the random number generating routine UDRNRT (IR). It must be any odd number.



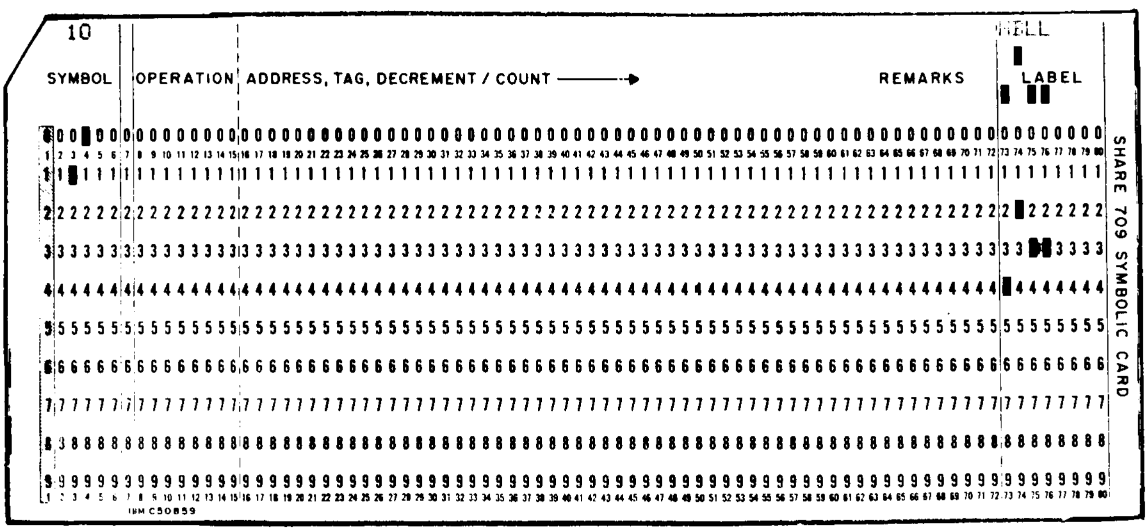

MM - Format (I4), right justified. MM is the common column spare list flag, which specifies whether or not the spares in a stage share a common spare list. MM can be either 0 or 1. The value of MM is used by the program only if NRL = 0.



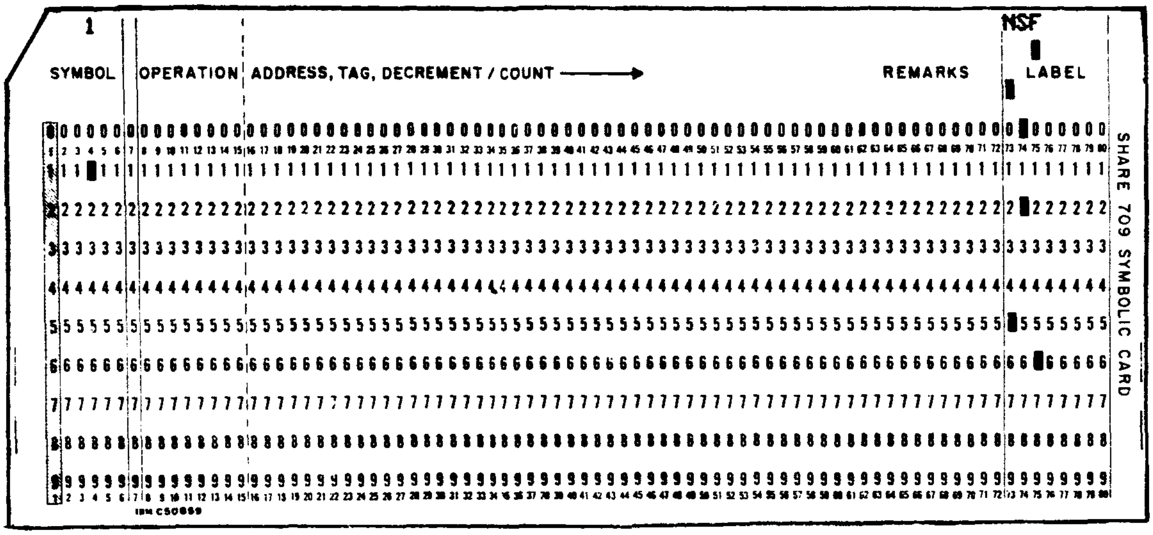
NRL - Format (I4), right justified. This is the non-repetitive list flag, which specifies whether the spare list will be read in by subroutine LIST (NRL = 0), generated by subroutine RANLIS (NRL = 1), or read in by the MAIN program (NRL = 2).



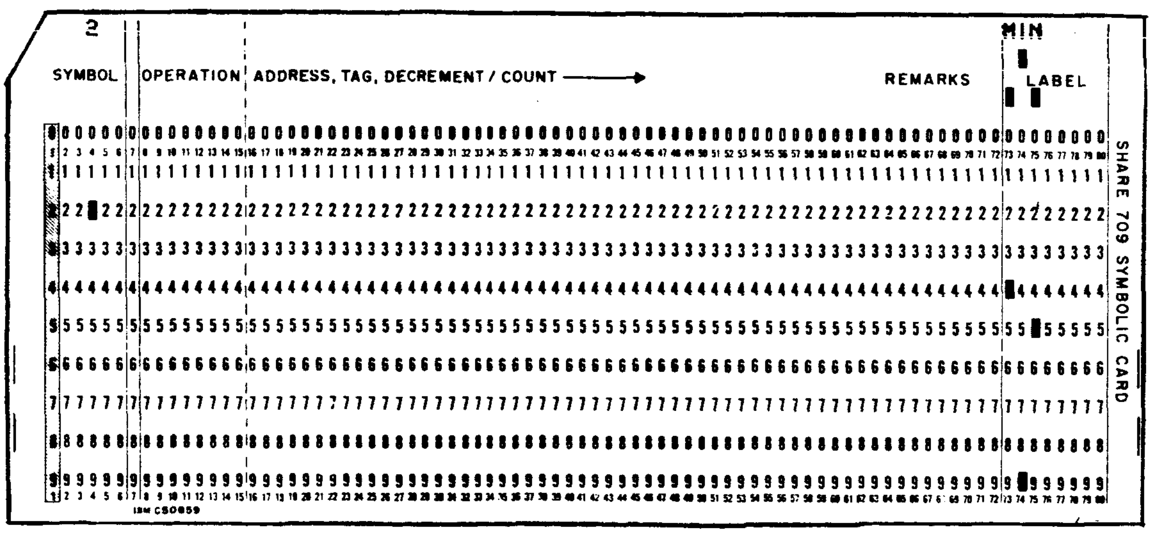

MBLL - Format (I4), right justified. MBLL is the missing block list length, which specifies the number of subsystems which are either failed or not included in the system at the start of the simulations.



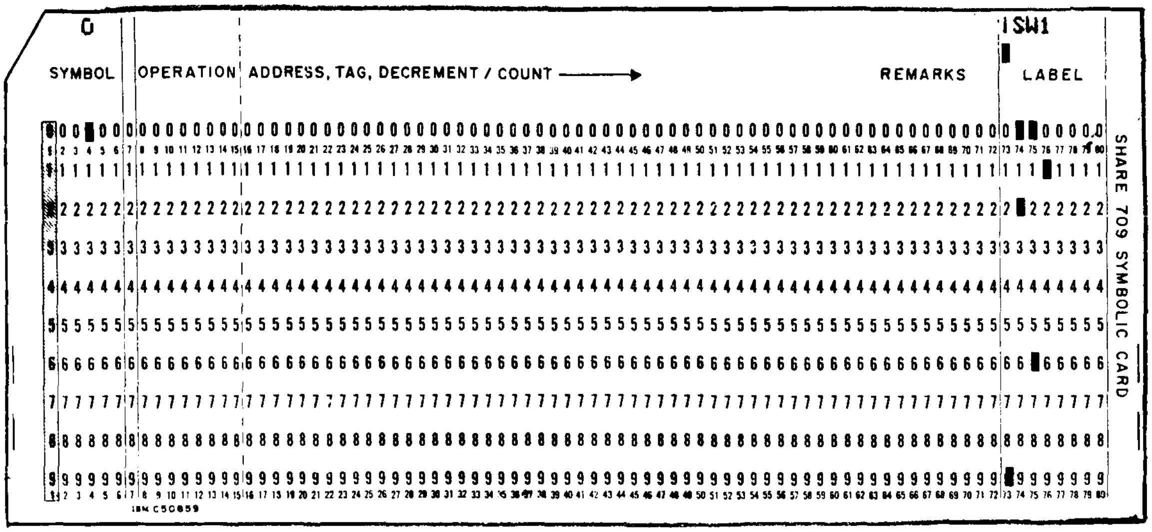
NSF - Format (I4), right-justified. NSF is the "No Sample Format" flag, which specifies whether the printout will be a complete listing of the system's failure response during every simulation (NSF = 0), or simply a listing of the statistical results of the simulations (NSF = 1).



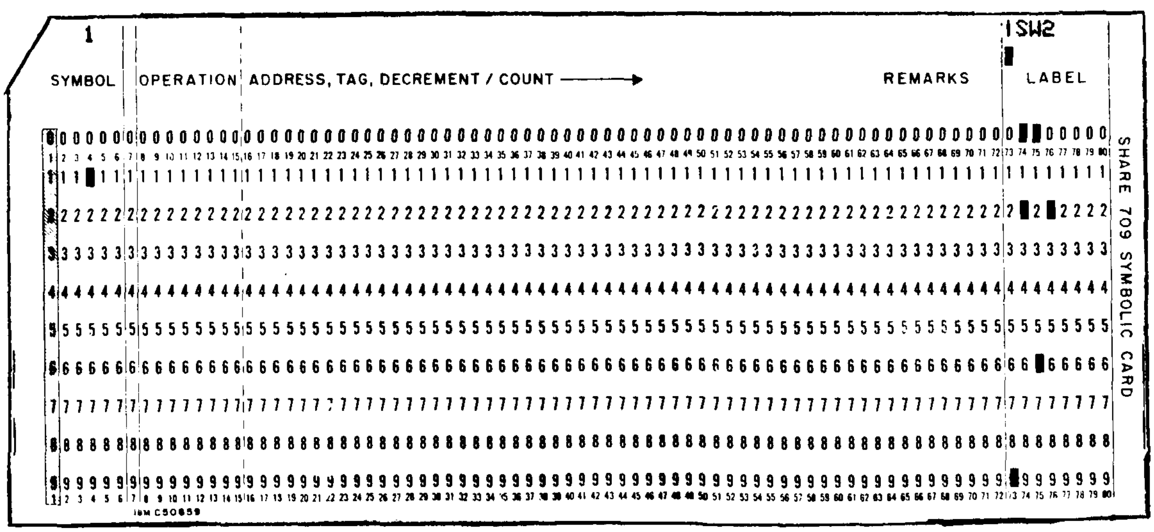

MIN - Format (I4), right-justified. This is the minimum number of operating subsystems required per stage, such that no spare switching, or reorganization, is necessary anywhere in the simulated system.



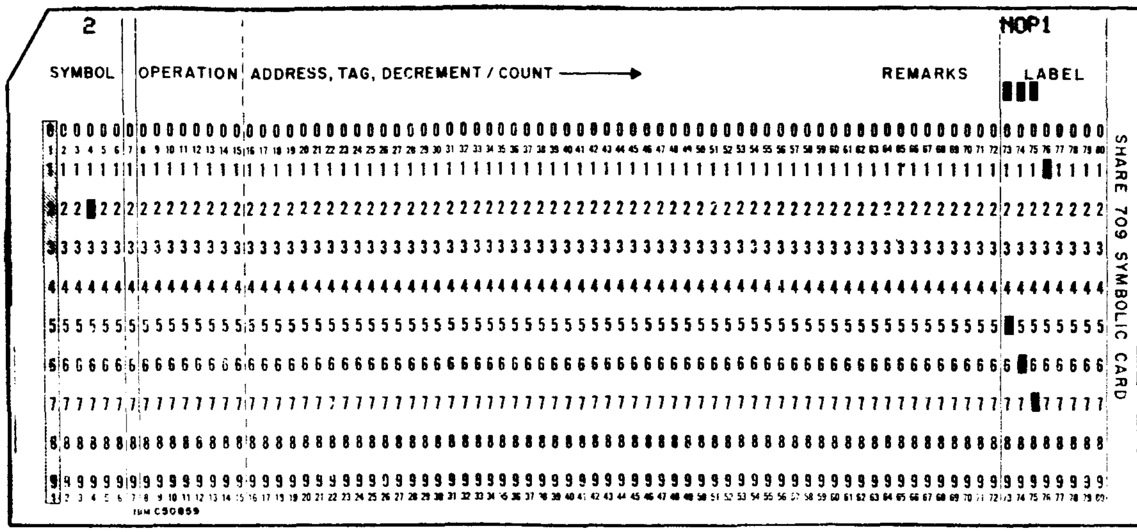
ISW1 - Format (I4), right-justified. This constant determines what the program does to the spare list of a failed location after that location obtains a spare. ISW1 may have one of three values, 0, 1, or 2, the determination of which is explained in the list of program variables and constants.



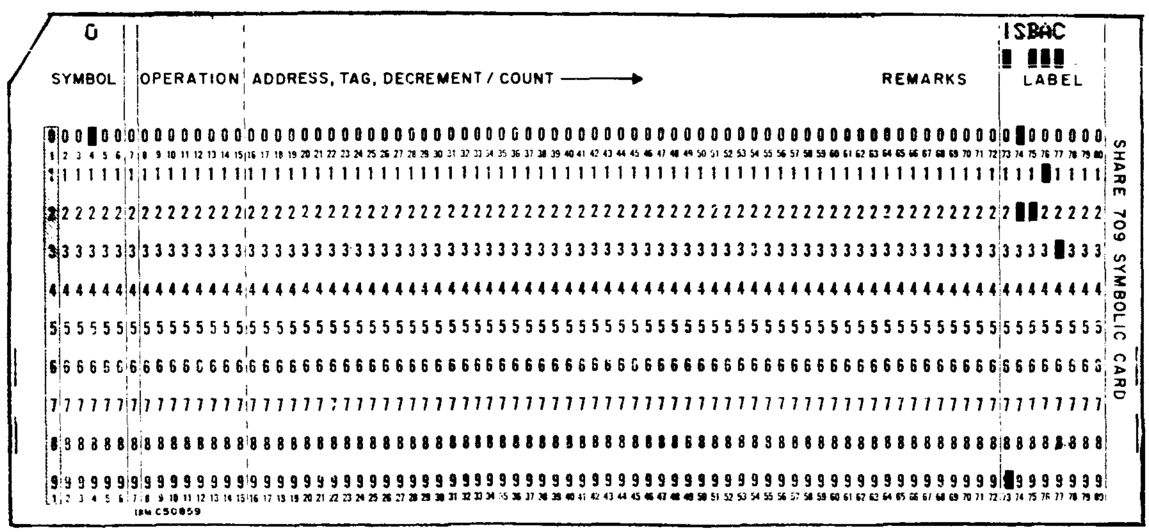

ISW2 - Format (I4), right-justified. This constant determines what the program does to the spare lists which contain the ID numbers of a failed subsystem and the spare which replaces it. The three permissable values for ISW2 are explained in the list of program variables and constants.



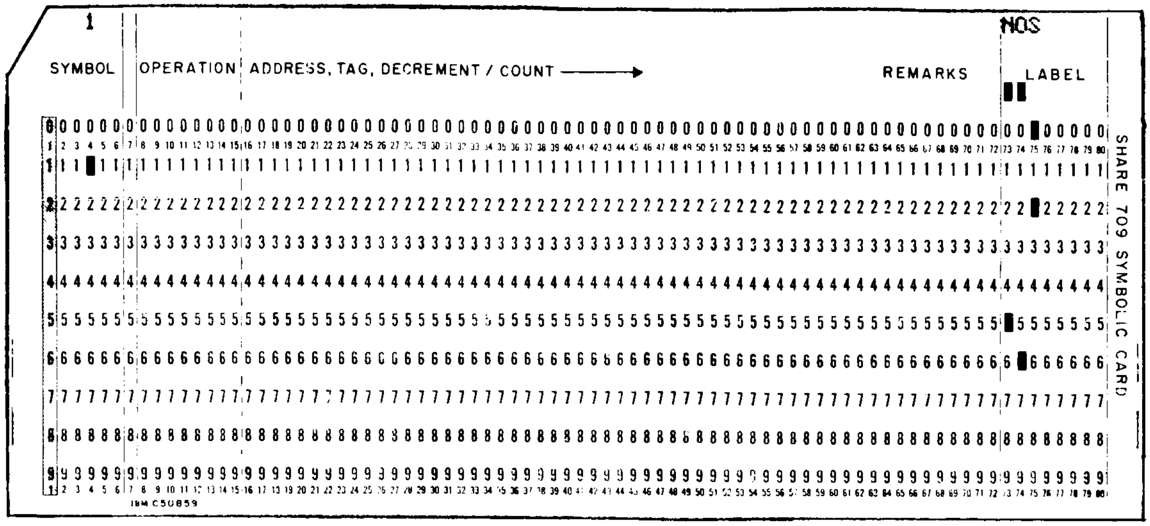
NOP1 - Format (I4), right-justified. NOP1 is used during the first "scan" of a failed location's spare list. It is the minimum number of operating subsystems which can be left in a stage if a spare is taken from that stage.



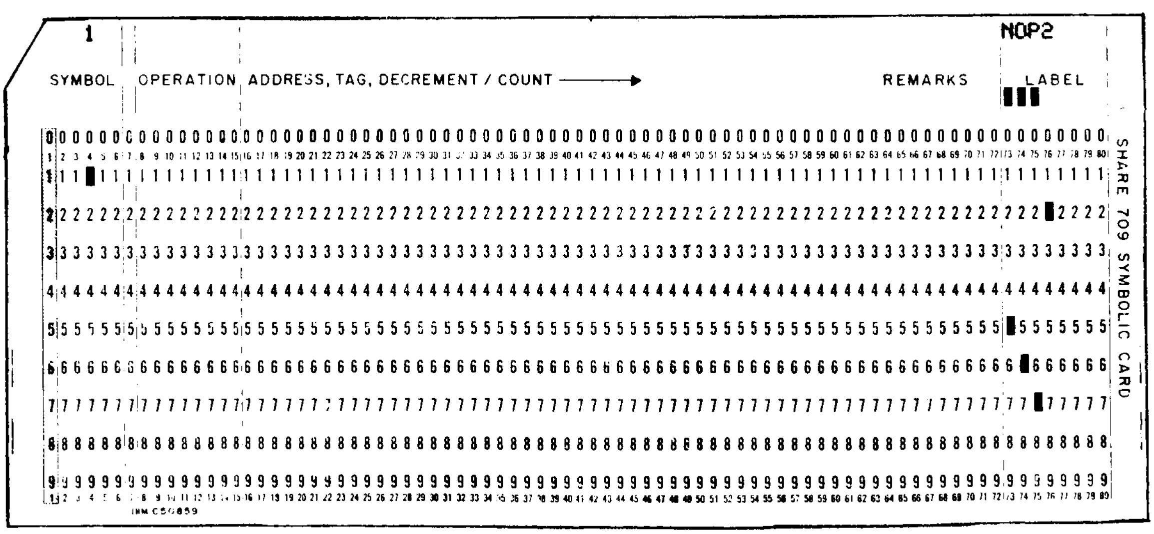

ISBAC - Format (I4), right-justified. This constant determines if available spares are switched to different locations as needed (ISBAC = 0), or if they are simply found, but left where they are (ISBAC = 1).



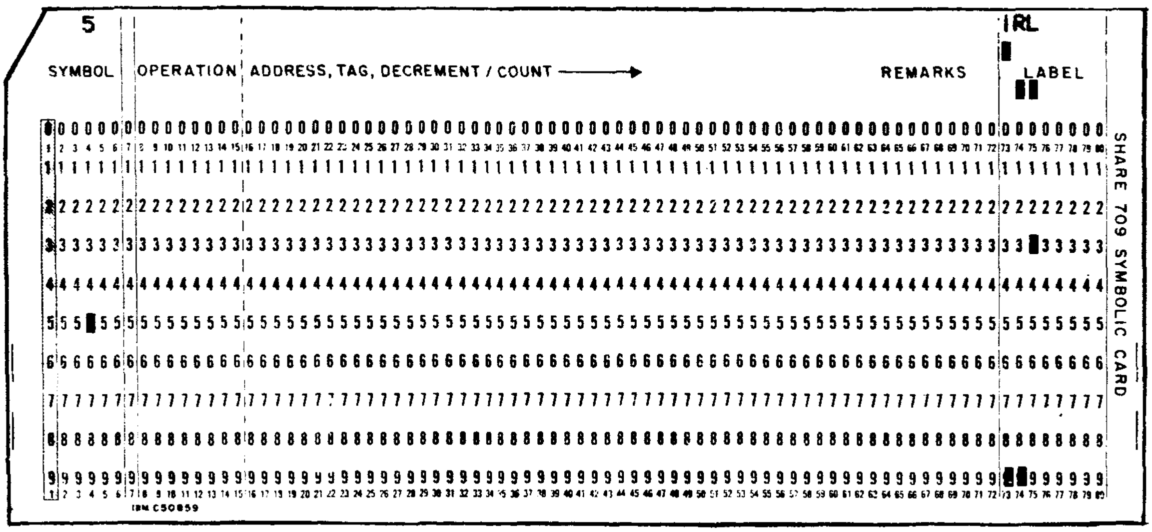
NOS - Format (I4), right-justified. This is the "No Own-Stage Spare" flag used in the random spare list generation routine, RANLIS. NOS determines whether a subsystem may have a subsystem from its own stage on its spare list (NOS = 0), or may not (NOS = 1). The value of NOS is used by the program only if NRL = 1.



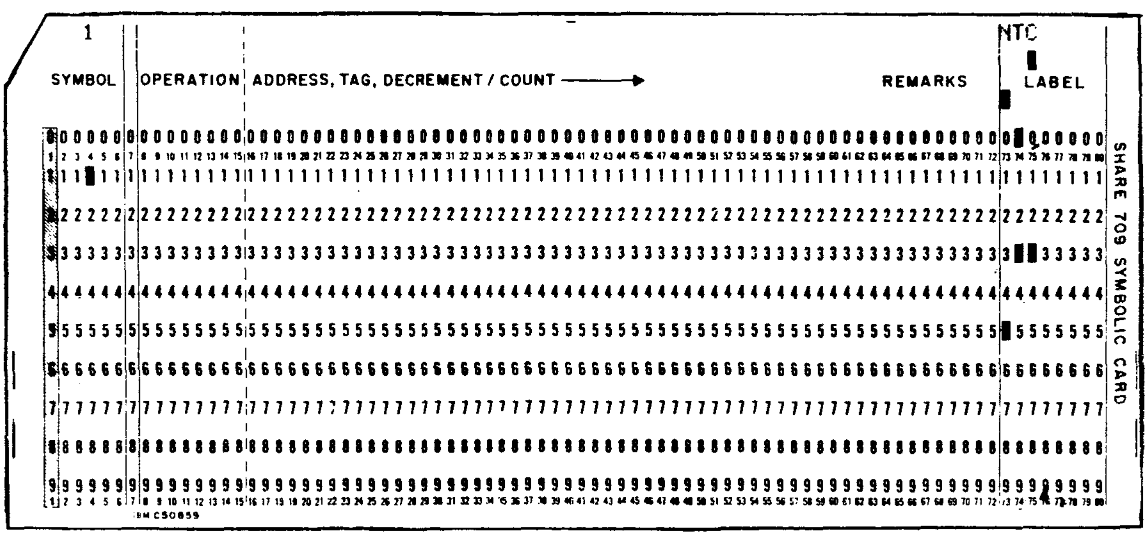

NOP 2 - Format (I4), right-justified. NOP2 is used during a "rescan" of a failed location's spare list. The list is rescanned only if no spares were available with the NOP1 criterion during the first scan. NOP2 is the minimum number of operating subsystems which can be left in a stage if a spare is taken from that stage during a rescan.



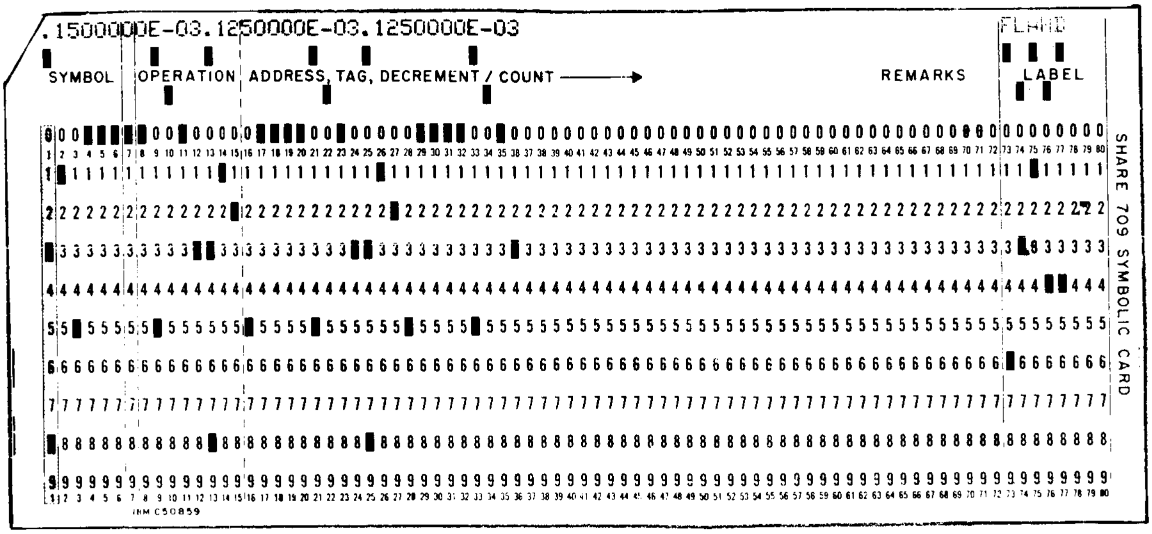
IRL - Format (I4), right-justified. This is the starting index for the random spare list generating routine UDRNRT (IRL), used in subroutine RANLIS. IRL must be any odd number.



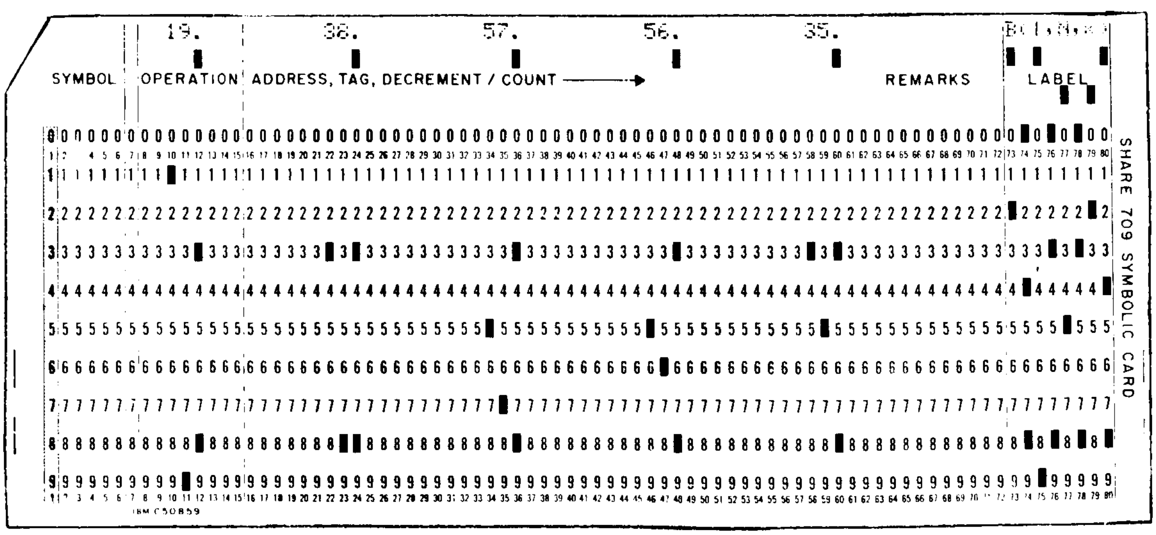

NTC - Format (I4), right-justified. This is the "No-Time-Curve" flag, which determines whether the program will compute and print out the Reliability-vs-Time curve, over Time from 0 to 5000 units (NTC = 0), or just the 90% Reliability Time or "Useful Life" (NTC = 1).



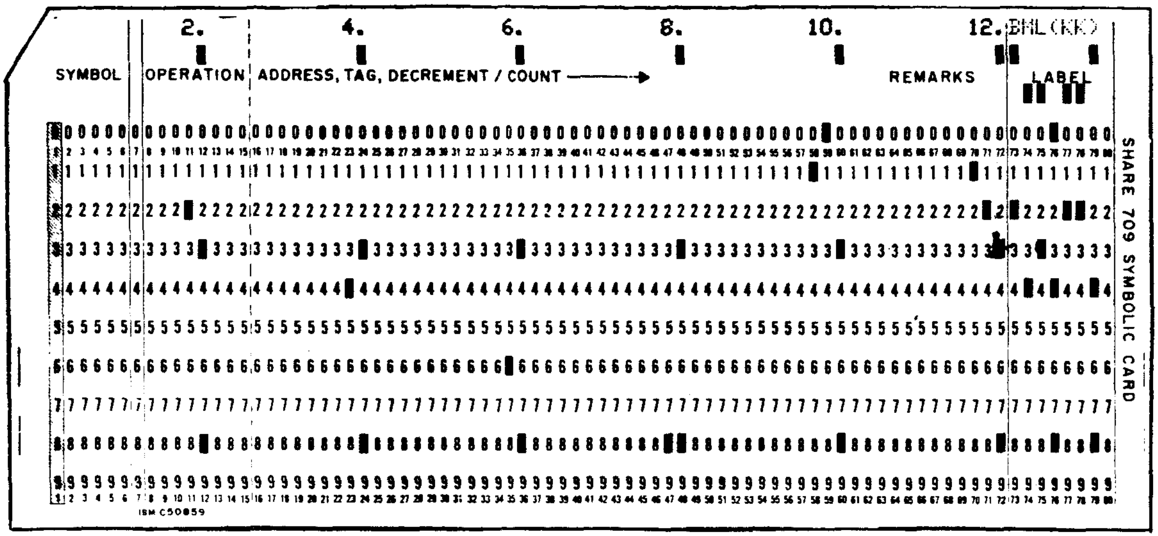
FLAMD(I), I = 1, M. Format (5E12.6), including decimal points. FLAMD(I) is the failure rate of the subsystems in the $I^{th}$ row of the system representation matrix, together with their associated circuitry.




B(I, N, K), K = 5, KTEMP. Format (6 F12.0), including decimal points. This is the spare list for the subsystem in row I, column N of the system representation matrix. The values read in are the ID numbers of the subsystems which will be called, in the order in which they appear on the list, to replace the subsystem.



BML(KK), KK = 1, MBLL. Format (6 F12.0), including decimal points. This is the list of "missing blocks"; that is, ID numbers of subsystems which are to be considered by the program as either failed or not included in the system at simulated time zero.



## V. LIST OF PROGRAM VARIABLES AND CONSTANTS

AMEAN - Non-subscripted variable-equal to the mean-time-to-failure of a given system, computed from all of the simulations of that system.

AUGT - Non-subscripted variable-equal to the average time between subsystem failures in a given system.

AVLAMD - Non-subscripted variable-equal to the sum of the failure rates of all subsystems in a given system.

B - Subscripted variable matrix which stores a mathematical model of the system being simulated, together with information specifying the instantaneous "state" of the system.

BML - Subscripted variables, equal to the identification numbers of the "missing blocks", which are subsystems considered either failed or not included in the system at time = 0.

C - Subscripted variable - matrix which stores the initial condition of matrix B during the simulation. This information is used to reset matrix B before another simulation of the same system is initiated.

CON - Non-subscripted variable - Average of the failure rates of all subsystems which are operational at time = 0.

CNT - Non-subscripted variable-equal to the total number of operational subsystems at the beginning of a simulation.

FAILB - Non-subscripted variable-equal to the identification number of the most recently failed subsystem.

FAL - Non-subscripted variable - value denotes the "failure order number" of the second most recently failed subsystem in the stage of the most recent failure. If a "repair" is necessary, this subsystem is replaced, not the most recent failure.

FAVGF - Non-subscripted variable-equal to the average number of subsystems failing per system simulation, computed from all of the simulations of a given system.

FLAMD - Subscripted constants - equal to the failure rates of the subsystems, plus those of their associated switching circuitry. All subsystems in the same row have the same failure rate.

IR - Non-subscripted variable - indexing number for the random number routine UDRNRT (IR). IR is initially an odd integer starting number for the routine.

IRL - Non-subscripted variable - starting index for the random spare list generating routine UDRNRT (IRL). IRL is initially an odd integer.

IRR - Non-subscripted variable - used to update IR after every simulation. This provides the generation of a different failure pattern for each simulation.

ISBAC - Non-subscripted constant - Determines whether or not available spares are switched. If ISBAC equals 0, spares are switched to new locations as needed. If it is 1, the program determines the availability of the spares, but does not switch them.

ISW1 - Non-subscripted constant, which determines what the program does to the spare list of a "repaired" location. The constant may have one of three values:

0 - Retain spare list of failed location

1 - Substitute spare list of spare's old location

2 - Delete spare list completely.

ISW2 - Non-subscripted constant, which determines what the program does to the spare lists which contain either a failed subsystem or the spare which replaced it. The constant may have one of three values.

0 - Leave spare's ID number on lists, and remove failed subsystem's.

1 - Remove spare's ID number from lists, and remove failed subsystem's.

2 - Remove spare's ID number from lists, and replace failed subsystem's ID with spare's ID number.

IW - Non-subscripted variable - equal to the instantaneous number of failed subsystems in a given system during simulation.

IWF - Subscripted variable - stores the total number of failed subsystems for each simulation of a particular system.

IWFT - Non-subscripted variable - stores the cumulative total number of failed subsystems for all simulations of a particular system.

KTEMP - Non-subscripted constant - equal to one dimension of matrix B, which stores the information specifying the instantaneous state of each subsystem, such as ID numbers, spare lists, failure order numbers, etc. Information for each subsystem is stored in locations B(I, J, K), where K = 1, . . . . KTEMP, and I and J are fixed.

M - Non-subscripted constant - equal to the order of redundancy of the system being simulated. M is one dimension of matrix B.

MBLL - Non-subscripted constant - Missing block list length; i. e. , the number of subsystems which are considered either failed or not included in the simulated system at $t = 0$.

MIN - Non-subscripted constant - equal to the minimum number of operating subsystems required in each stage of a particular system, so that no spares are required anywhere in the system.

MM - Non-subscripted constant - Common column spare list flag, which specifies one of two conditions:

MM = 0 - all subsystems have the same spare list pattern, but none has a spare list identical to that of another subsystem.

MM = 1 - all subsystems have the same spare list pattern, and all subsystems in a stage have identical (common) spare lists.

MR - Non-subscripted variable - equal to the instantaneous total number of simulations which have been performed on a given system.

N - Non-subscripted constant - equal to the number of stages in the simulated system. N is one dimension of matrix B, which stores the system state information.

NI - Non-subscripted variable - equal to the "row" in the system in which a subsystem to be replaced is located. (Row refers to the row in matrix B).

NJ - Non-subscripted variable - equal to the stage in the system in which a subsystem to be replaced is located. (A stage is represented by a column in matrix B).

NOP1 - Non-subscripted constant - Minimum number of operating subsystems which can be left in a stage if a spare is taken from that stage. NOP1 is used during the first "scan" of a failed subsystem's spare list.

NOP2 - Non-subscripted constant - Minimum number of operating subsystems which can be left in a stage if a spare were taken from that stage, during a rescan of a failed subsystem's spare list. NOP2 is generally lower than NOP1. (Rescan is only performed if no spares are available during the first scan).

NOS - Non-subscripted constant - "No own-stage spare" flag, which is used in the random spare list generation routine RANLIS. NOS may have one of two values:

0 - Any subsystem may have a subsystem from its own stage on its spare list.

1 - No subsystem may have a subsystem from its own stage on its spare list.

NR - Non-subscripted constant - equal to the number of simulations, or runs, which the program is directed to perform on a given system.

NRL - Non-subscripted constant - Non-repetitive list flag, which specifies to the program the method of spare list generation to be used. NRL may have one of three values:

0 - A number of spare lists are read in and their patterns reproduced for the remainder of the lists, by subroutine LIST.

1 - Completely random spare lists are generated by subroutine RANLIS.

2 - A separate, non-repetitive spare list is read in for each subsystem, by the MAIN program.

NSF - Non-subscripted constant - "No sample format" flag, which controls the amount of printout obtained from the simulations. NSF may have one of two values:

0 - The contents of the entire system representation matrix B at the beginning of the simulation are printed out, plus the entire sequence of failures and repairs, and results of the simulation.

1 - A brief listing of simulation results is printed.

In both cases, the input data specifying the system to be simulated is printed.

NTC - Non-subscripted constant - No-time-curve flag, which specifies the amount of statistics to be compiled from the simulation results. NTC may have one of two values:

0 - The program computes and prints out the complete reliability-vs.-time curve for values of time from 0 to 5000 hours, in 100 hour intervals. The values are computed from the statistical results of all of the simulations.

1 - The program computes and prints out only the "useful life" of the system.

OPRT - Non-subscripted variable - equal to the running total of the system operating times during the first 25 simulations of a given system.

P - Subscripted variable - Before the overall simulation results are computed, P equals a histogram of the frequency of systems failing with exactly x subsystem failures. During overall results computation, P is converted to a list of the conditional probabilities that the system has failed, given that there are x subsystem failures.

PFAT - Non-subscripted variable - equal to the probability that the simulated system has failed at some time, t. (PFAT = f(t)). PFAT is computed from the statistical results of all simulations of a given system.

RNT - Non-subscripted variable - The random number in subroutine TIME, chosen from a uniformly distributed set of random numbers, and used to generate the time to failure of each failed subsystem.

S - Non-subscripted variable - generated by subroutine FAIL, and indicates, if non-zero, that the most recently generated failure has created a need for a spare.

SUM - Non-subscripted variable - equal to the sum of the failure rates of all subsystems operating at the start of a simulation.

SUMT - Subscripted variable - equal to the running totals of the operating times of a particular system for every simulation of the system.

SYSFA - Non-subscripted variable - indicates to the program, if non-zero, that the system being simulated has failed.

T - Subscripted variable - A matrix used to store, in ascending order, the running times of the five shortest-lived of the first twenty-five simulations. The largest of the five is used in computing the first approximation to the "useful life" of the simulated system.

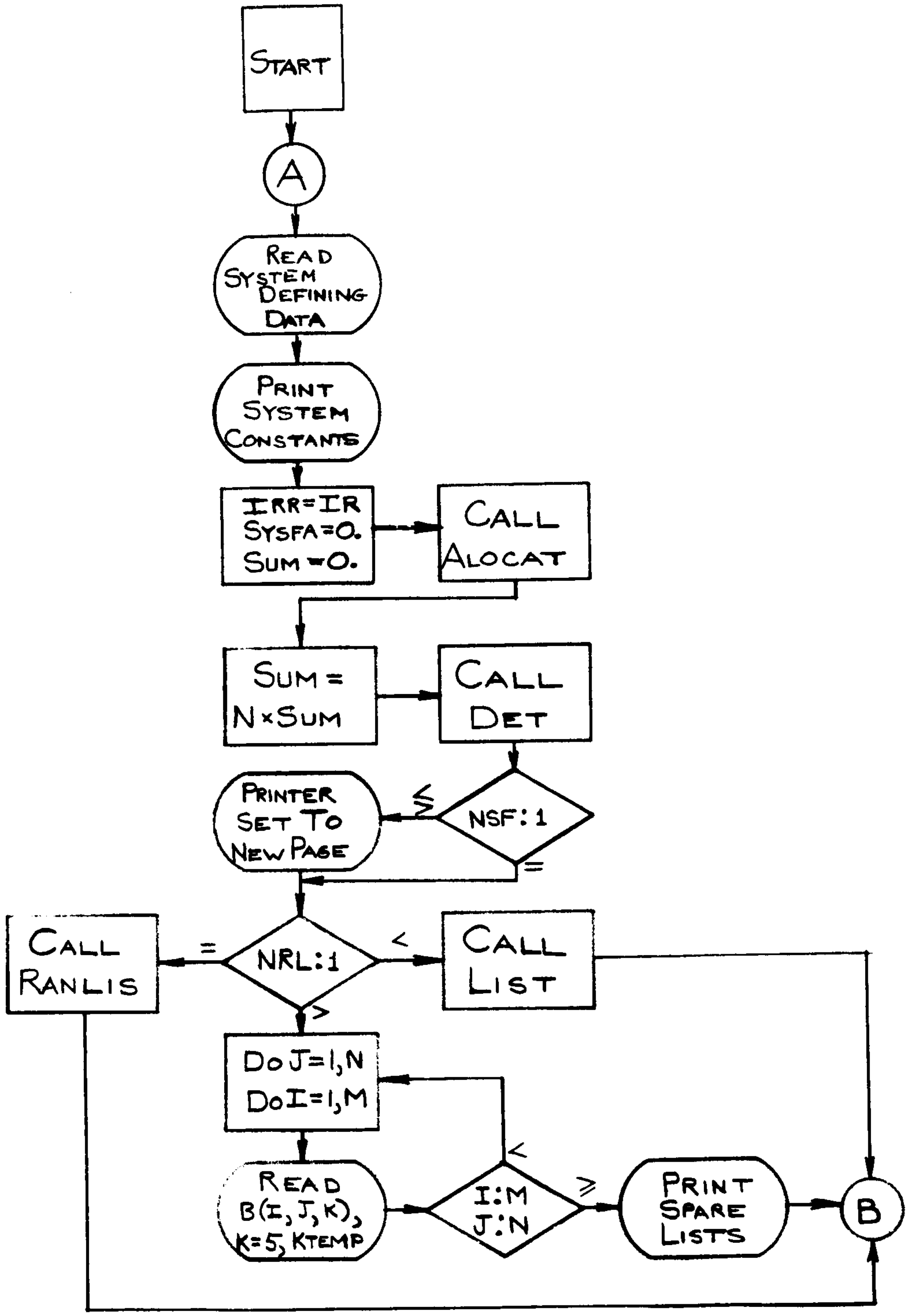
TI - Non-subscripted variable - the time used in the equations for computing the useful life or the reliability curve of the simulated system.

TT - Subscripted variable - Equal to the times between the subsystem failures for a given system simulation.

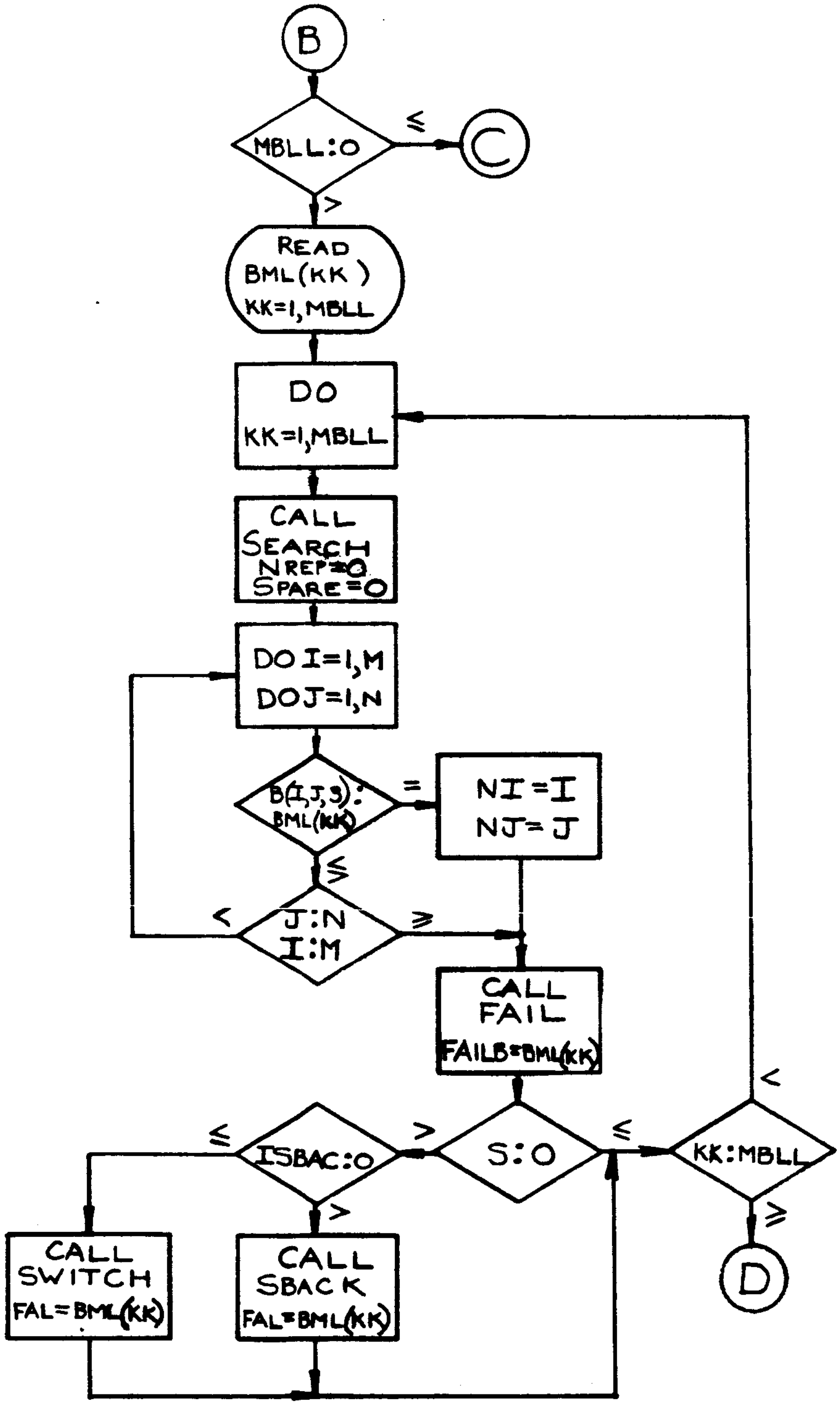
X - Non-subscripted variable - a random number generated in subroutine FAIL, which designates the subsystem which has failed. X is chosen from a uniformly distributed set of random numbers between zero and one.

# APPENDIX A
# DETAILED FLOW DIAGRAMS

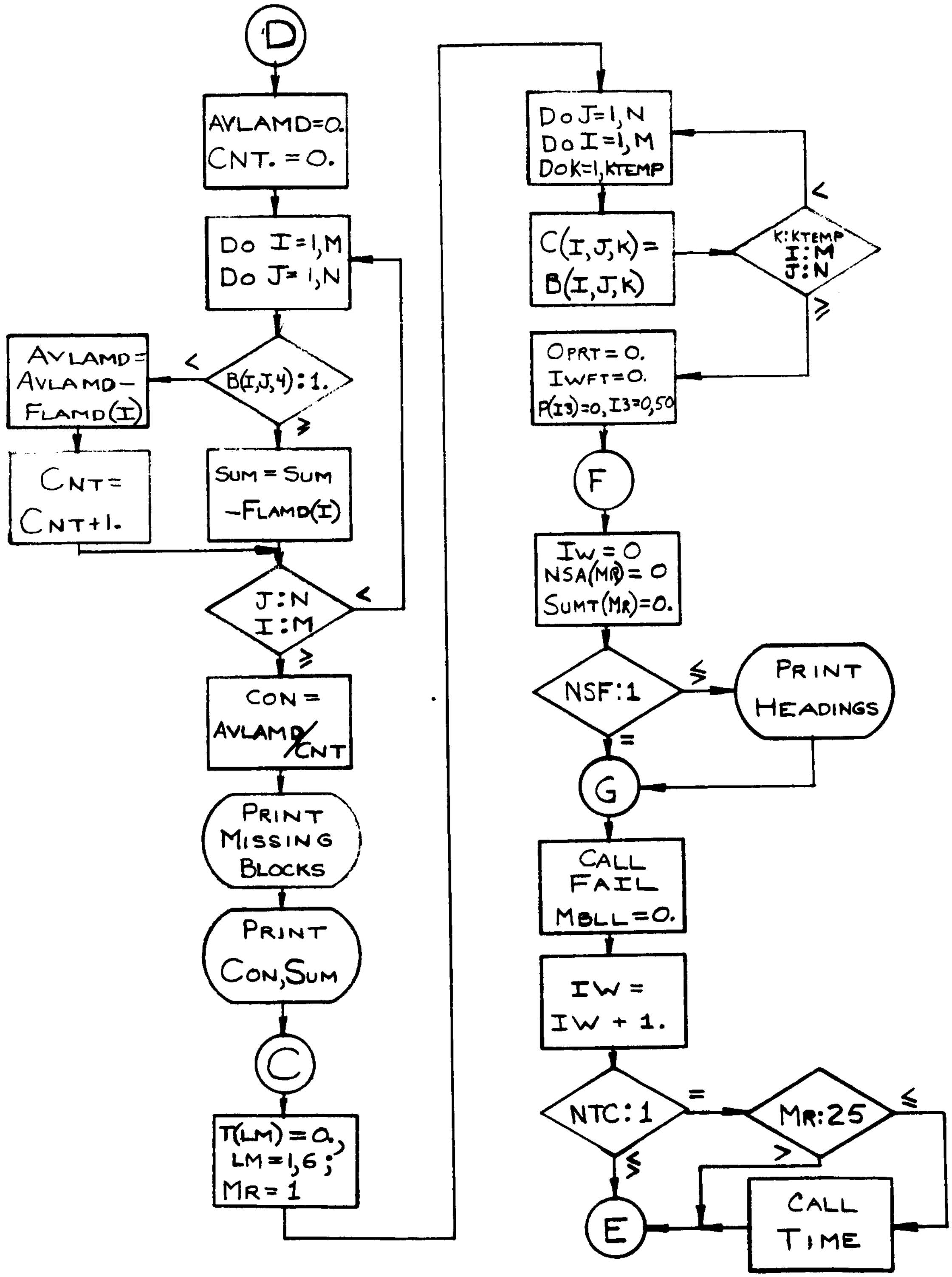
This section contains, first, a detailed flow diagram of the MAIN executive program, followed by flow diagrams of each of the ten subroutines, in alphabetical order.



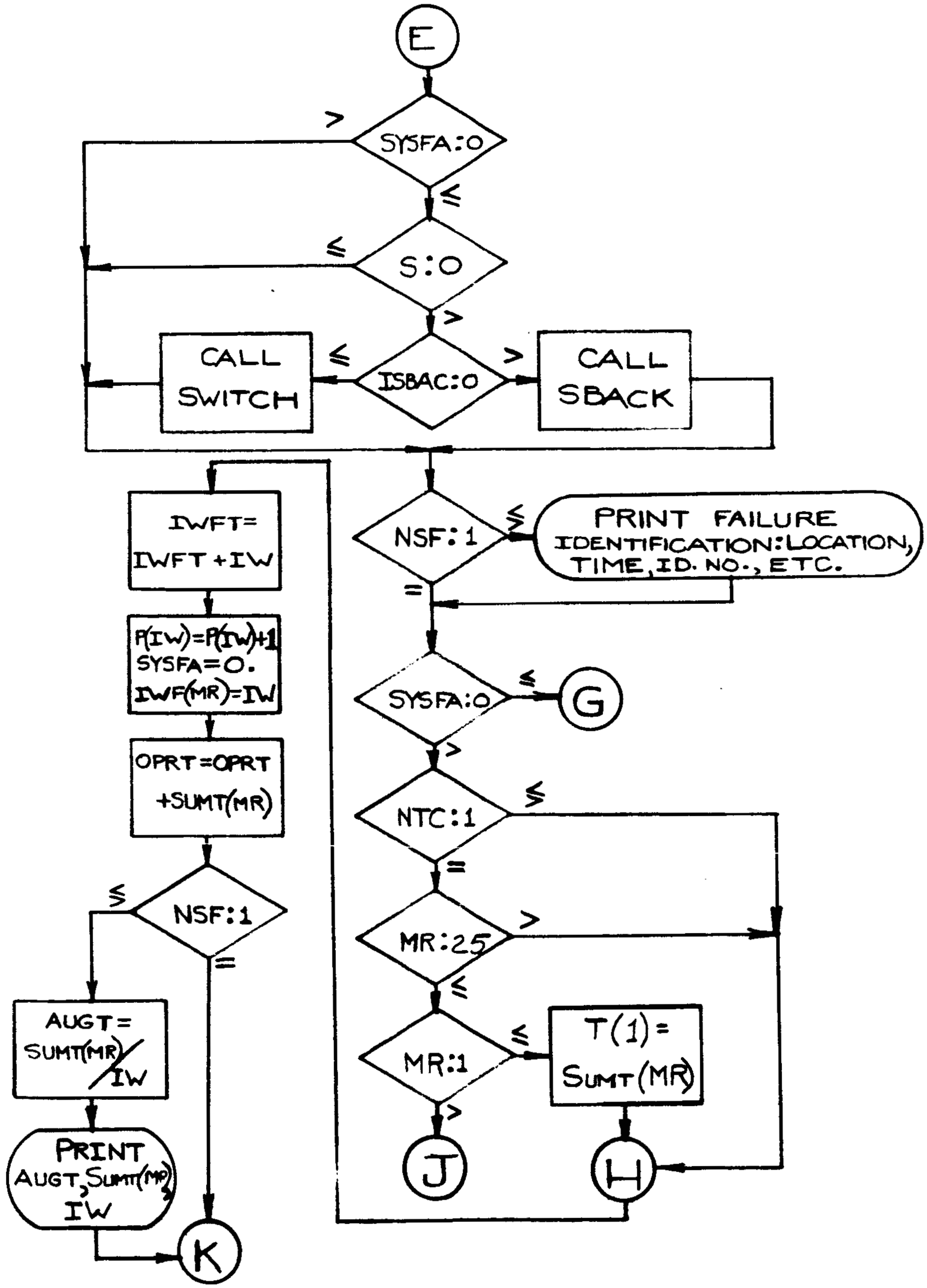

MAIN PROGRAM



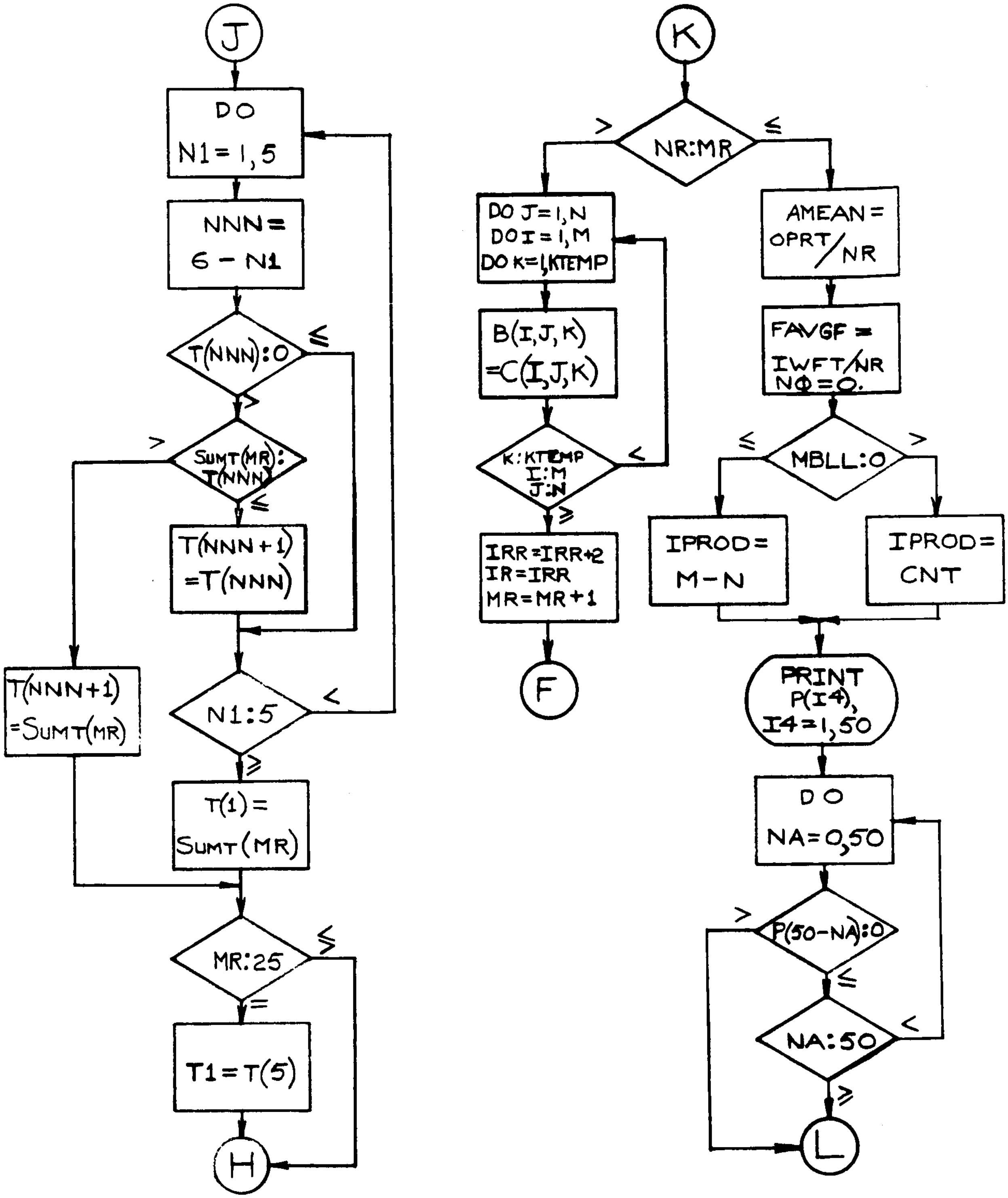

MAIN PROGRAM (CONT.)



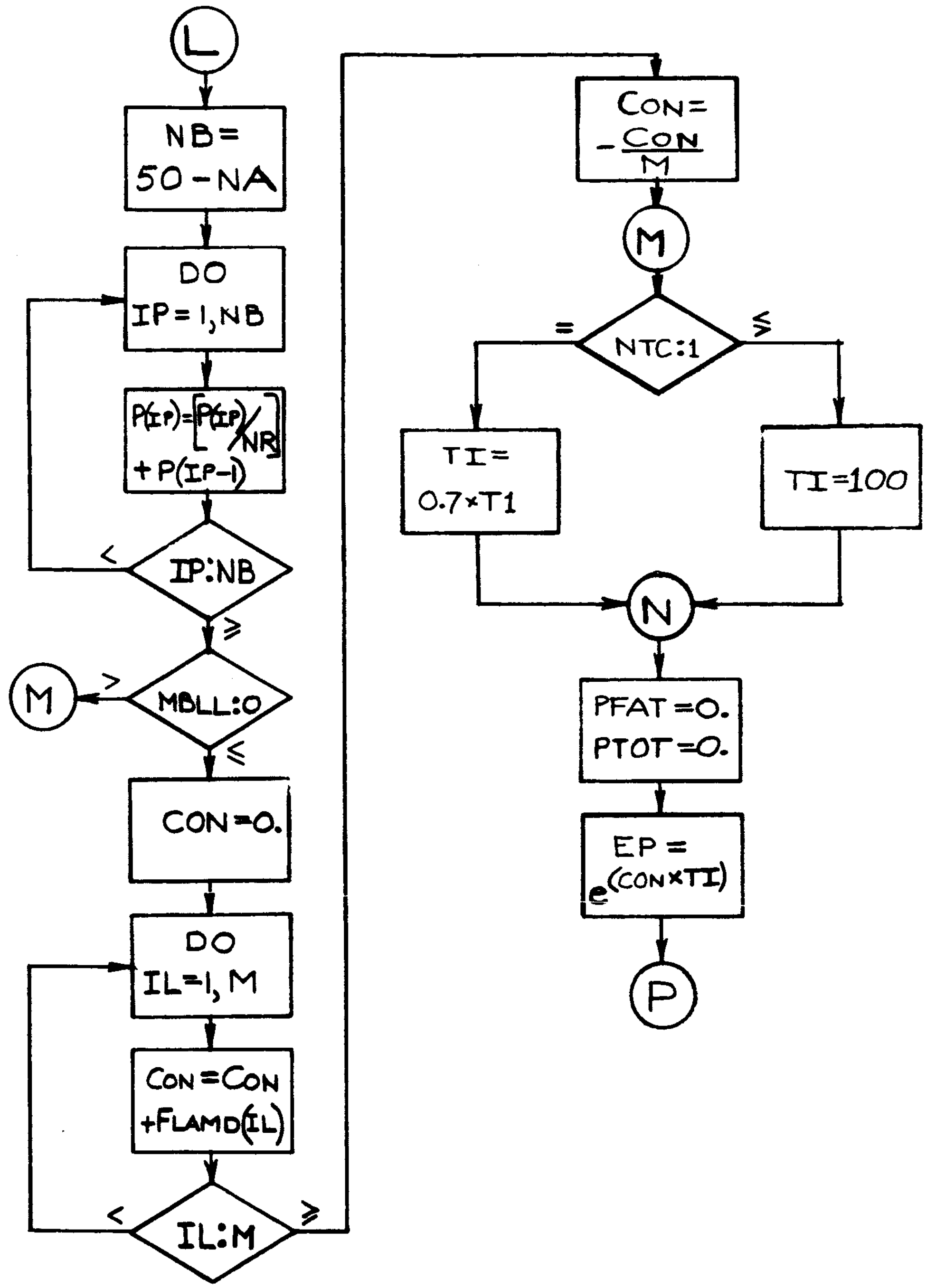

MAIN PROGRAM (CONT.)



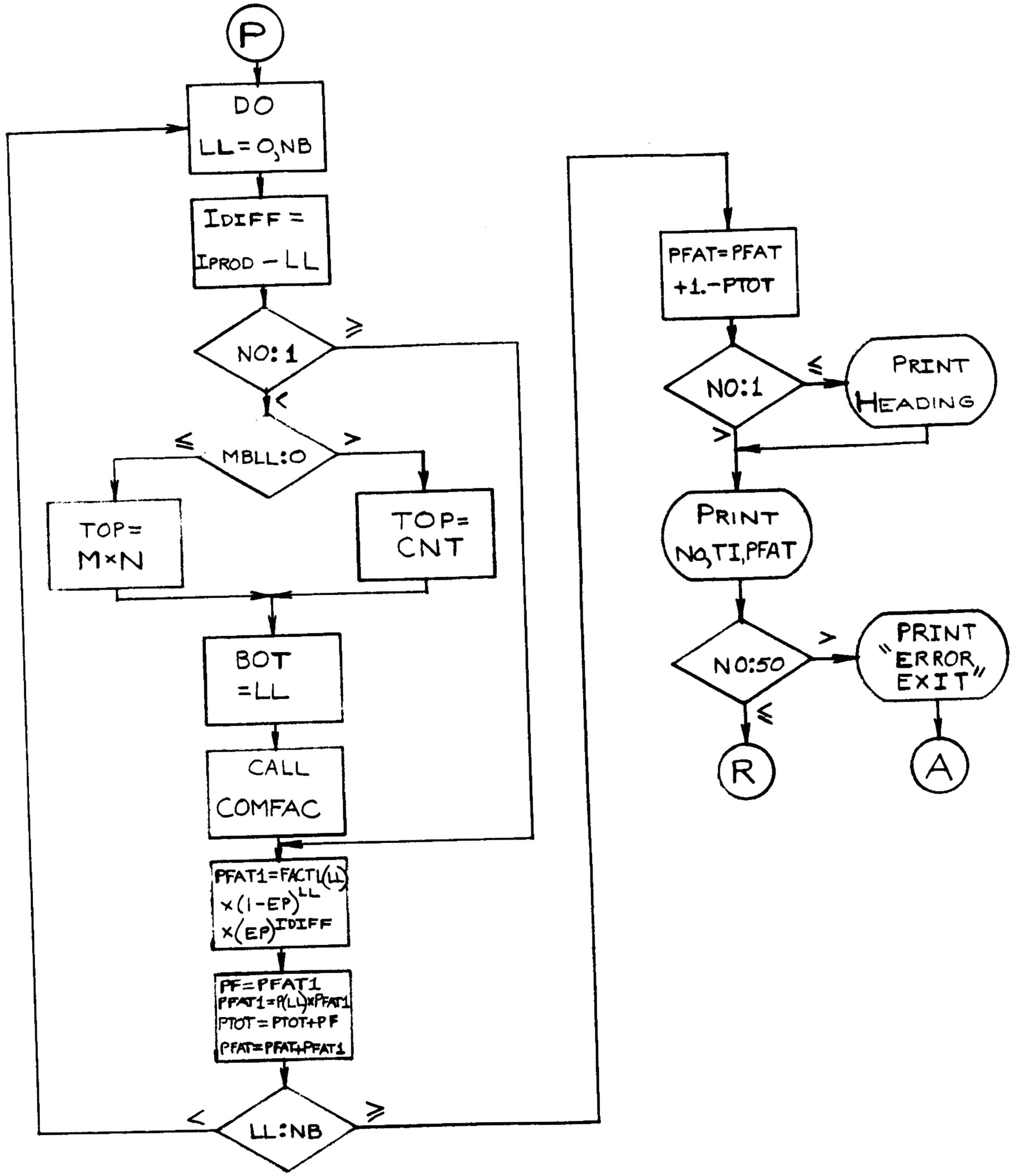

MAIN PROGRAM (CONT.)



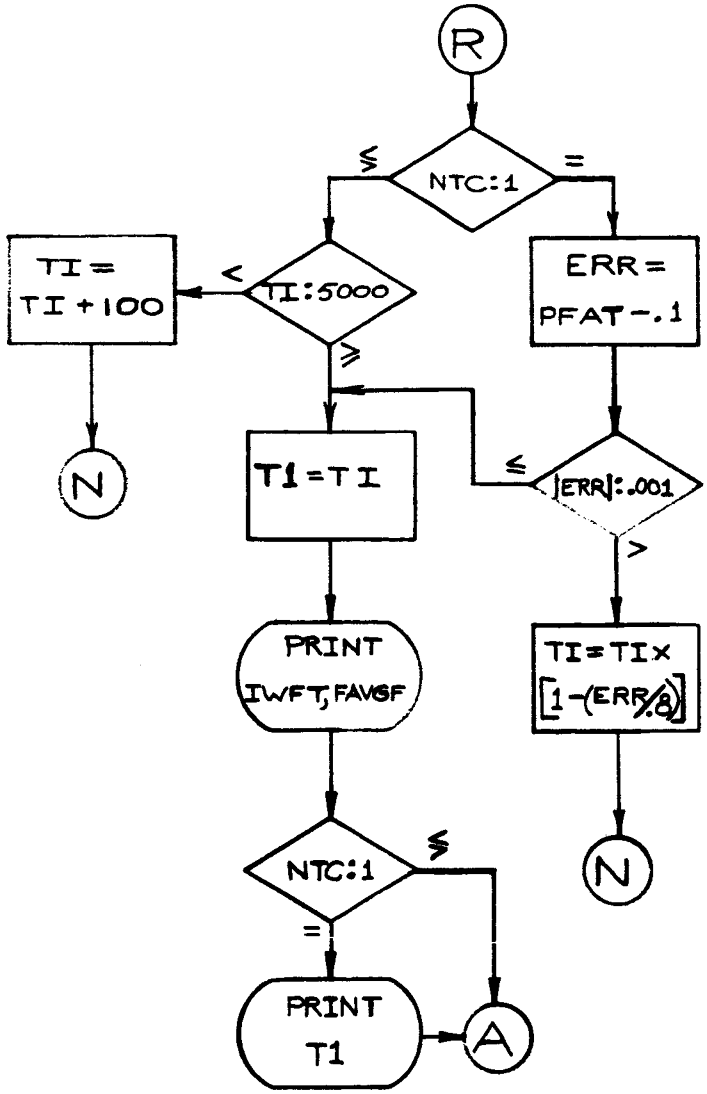

MAIN PROGRAM (CONT.)



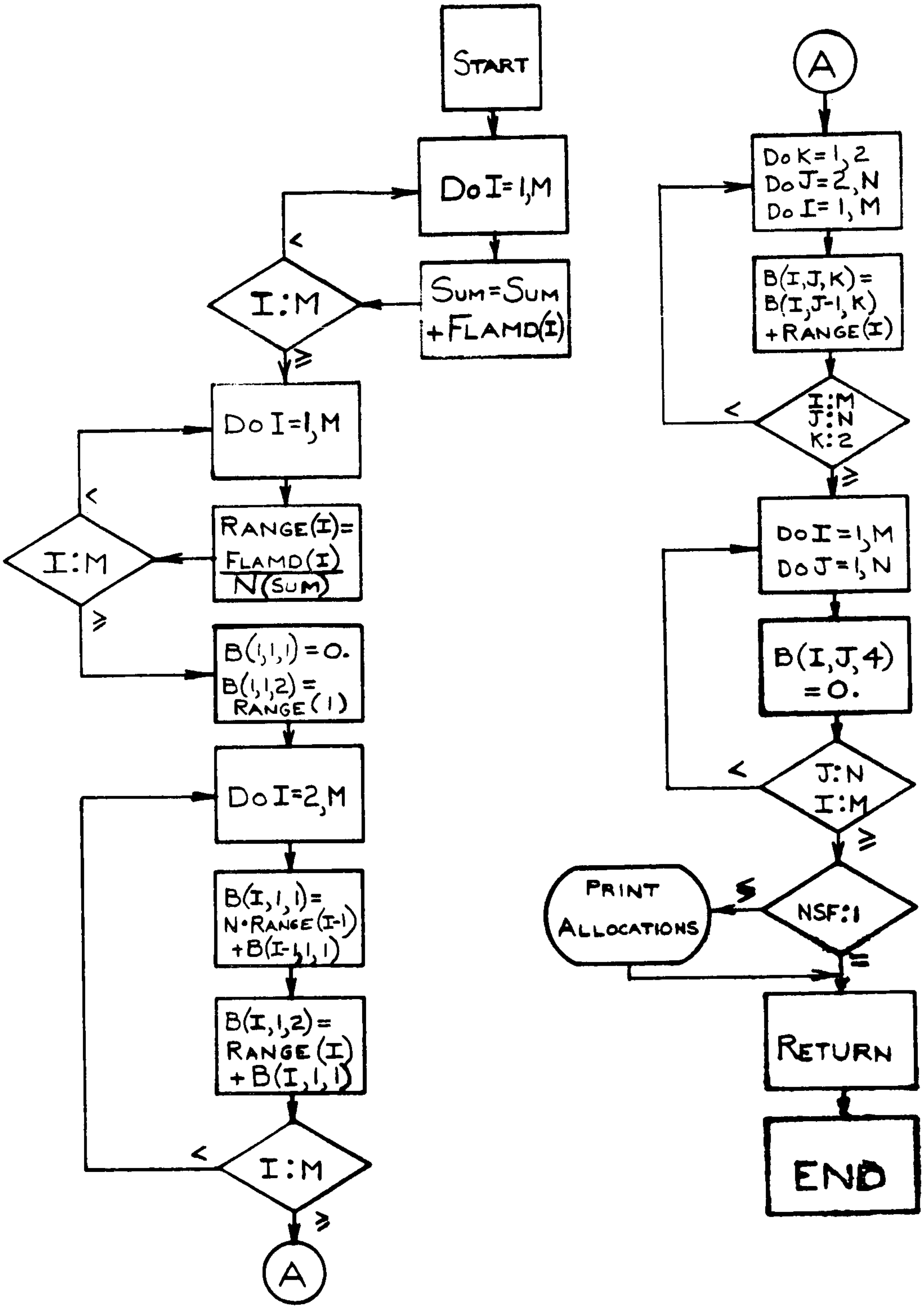

MAIN PROGRAM (CONT.)



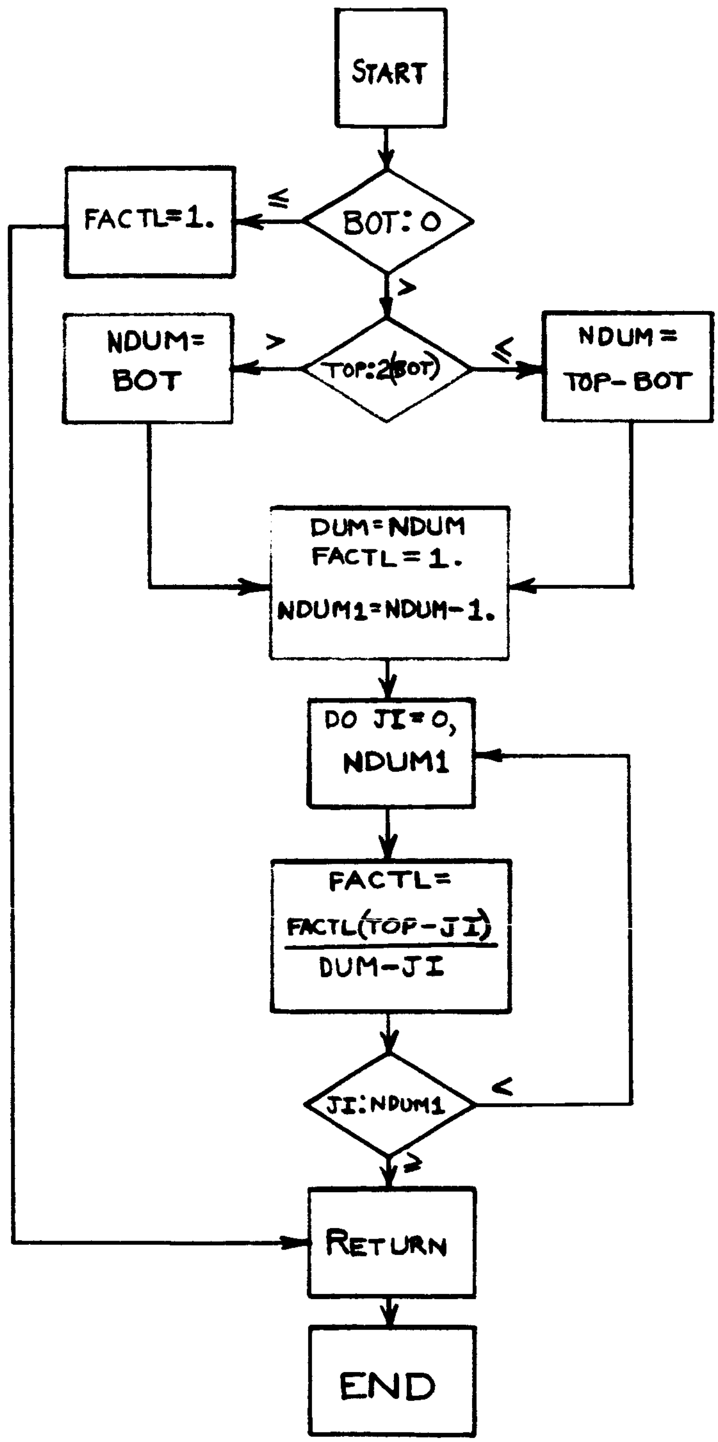

MAIN PROGRAM (CONT.)



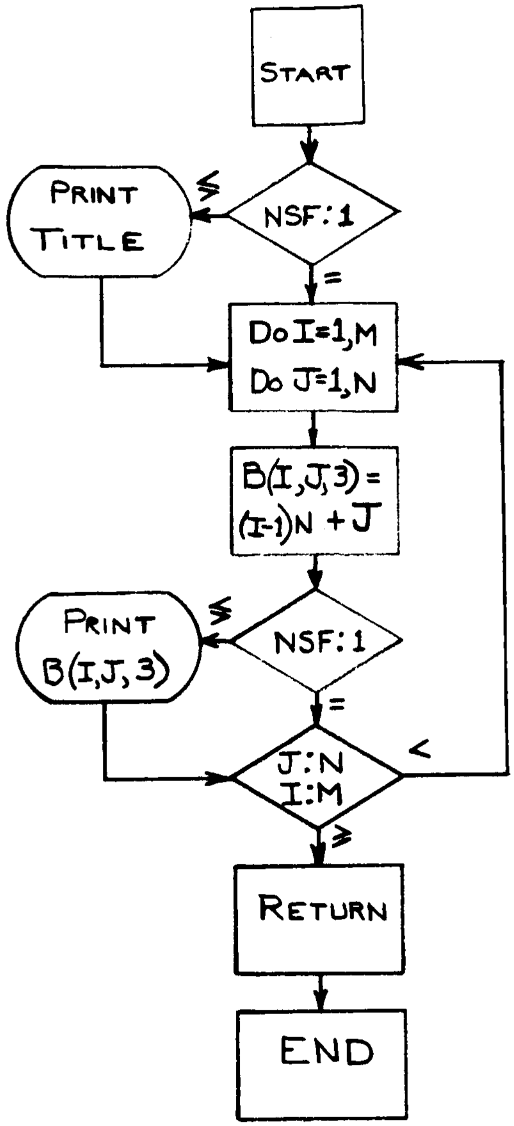

MAIN PROGRAM (CONT.)



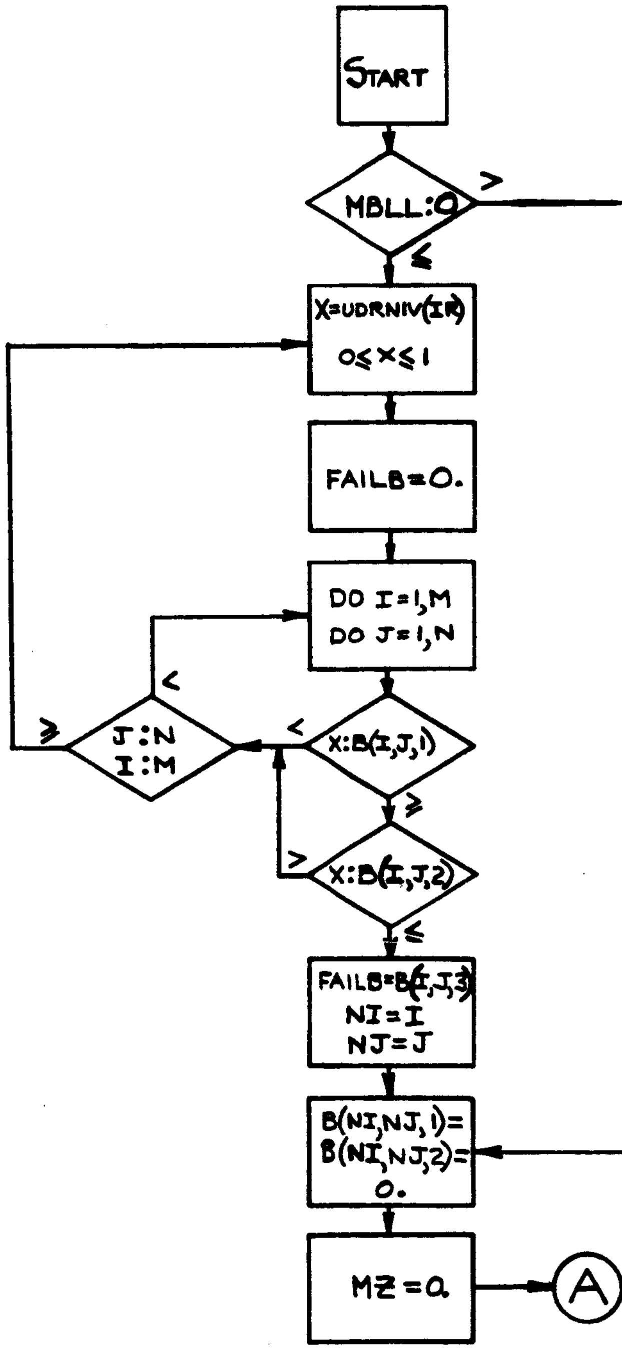

SUBROUTINE ALOCAT



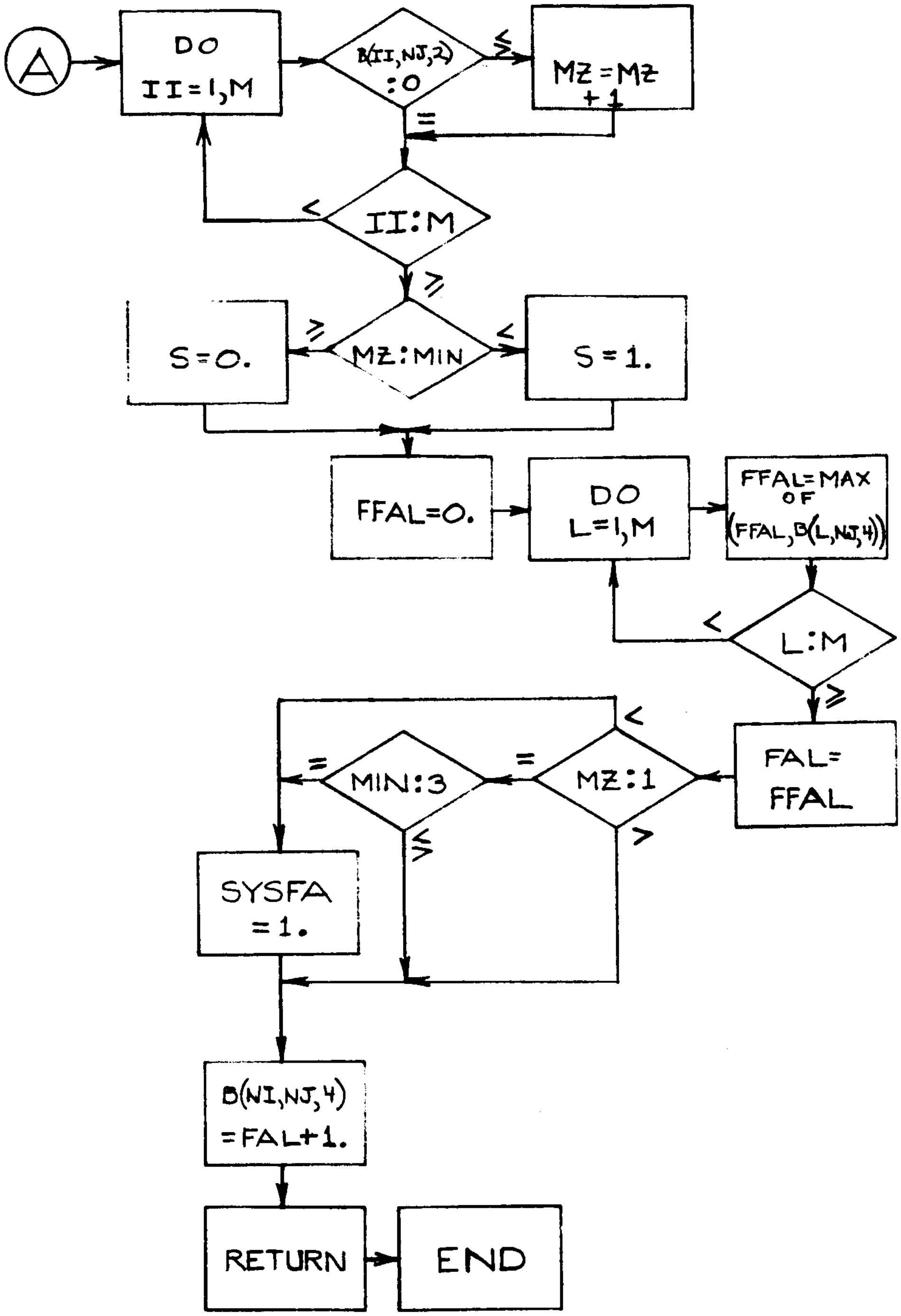

SUBROUTINE COMFAC



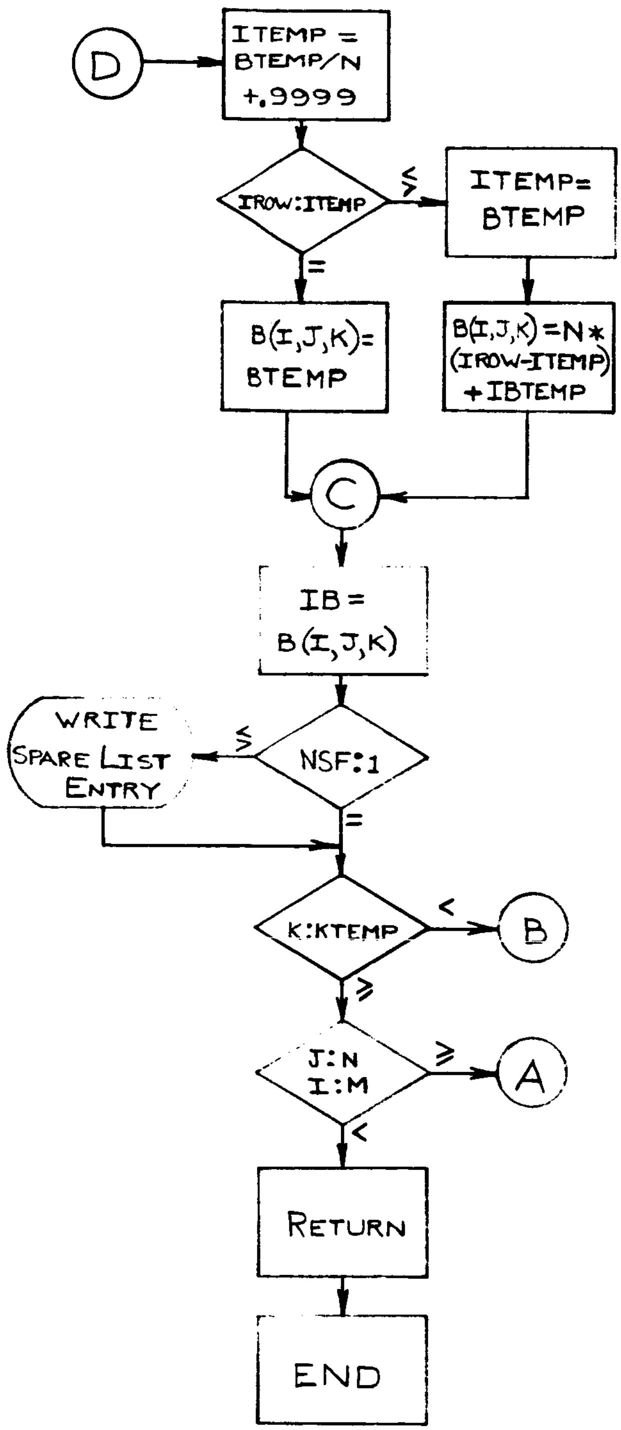

SUBROUTINE DET



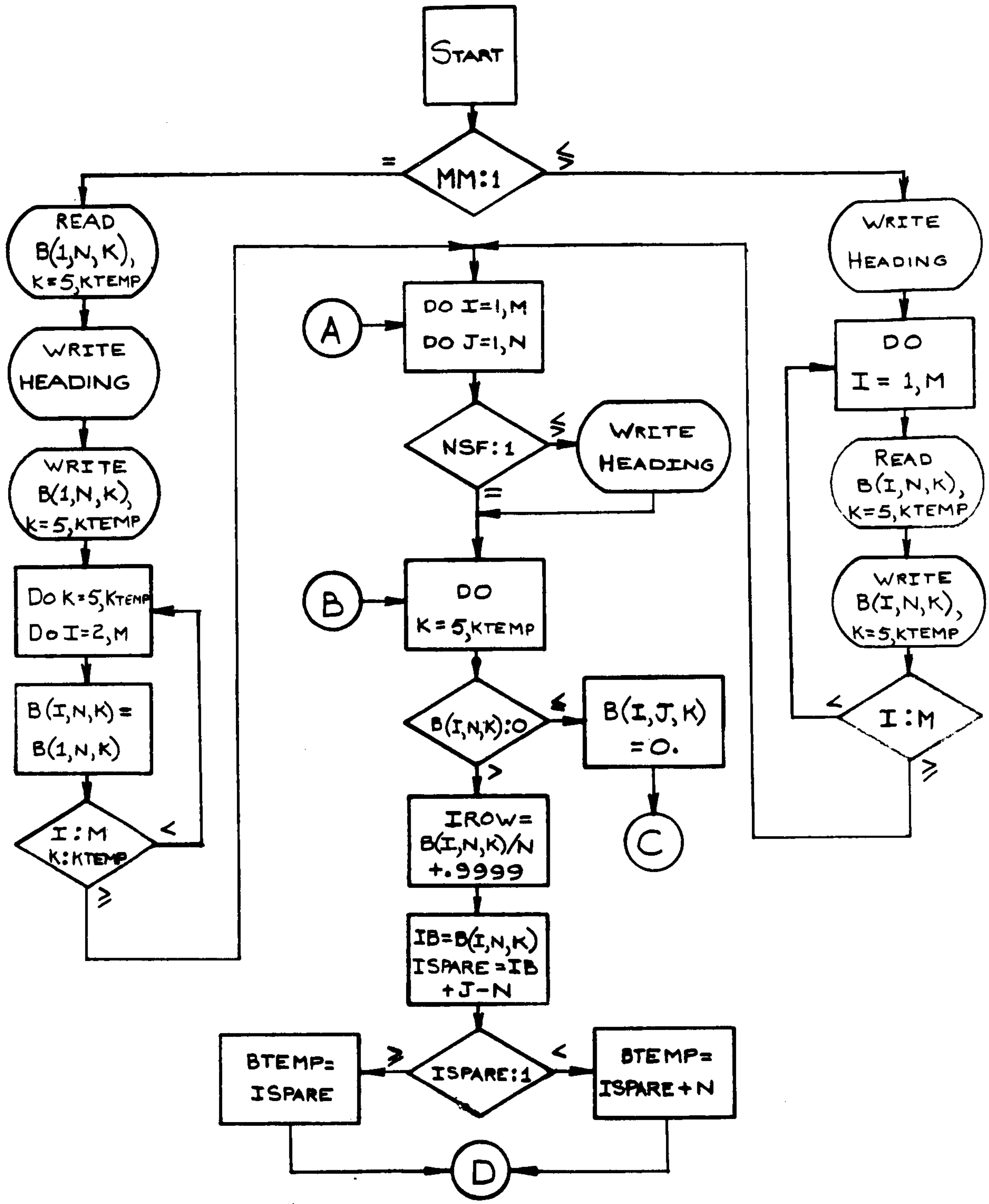

SUBROUTINE FAIL



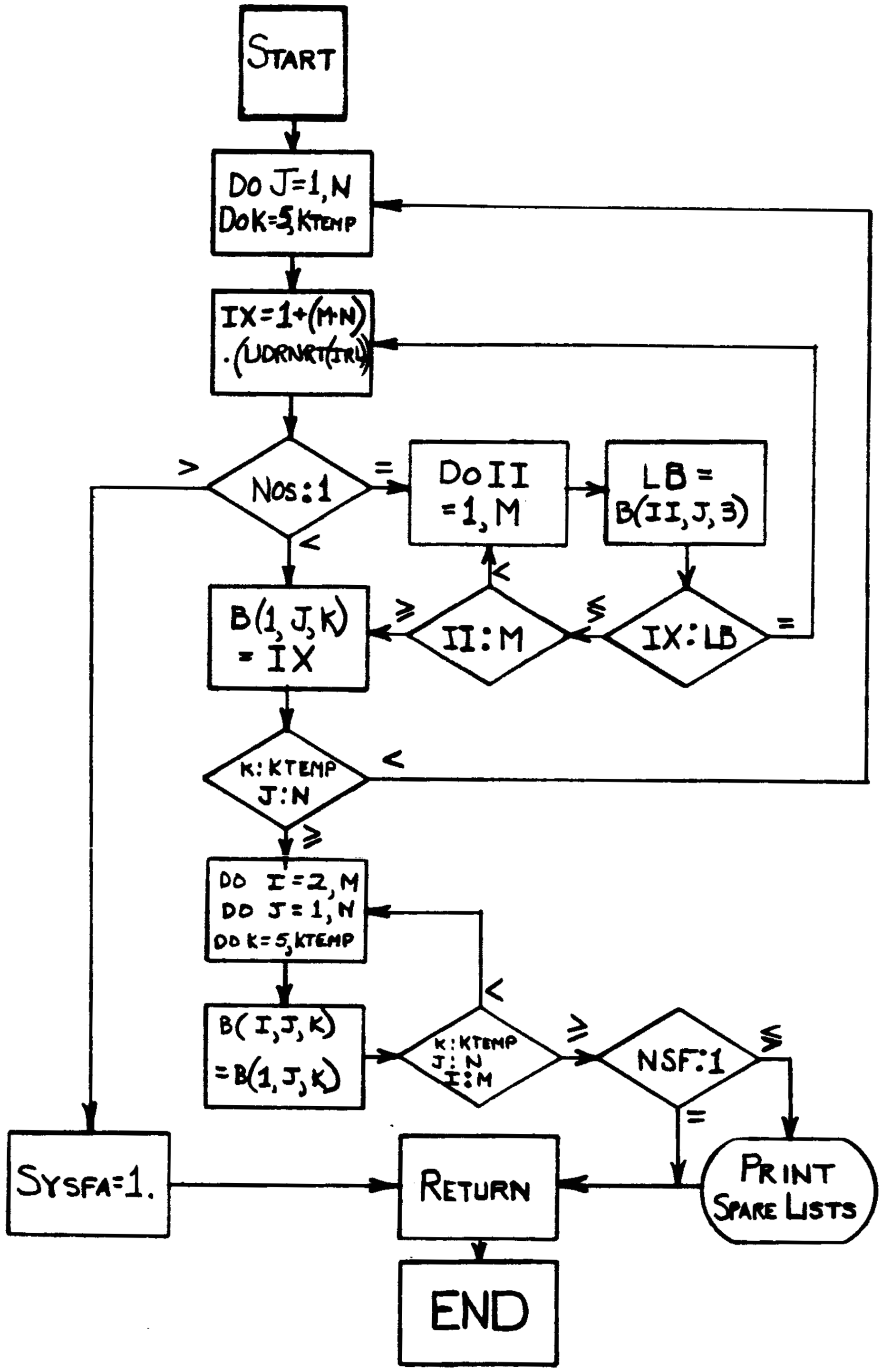

SUBROUTINE FAIL (CONT.)



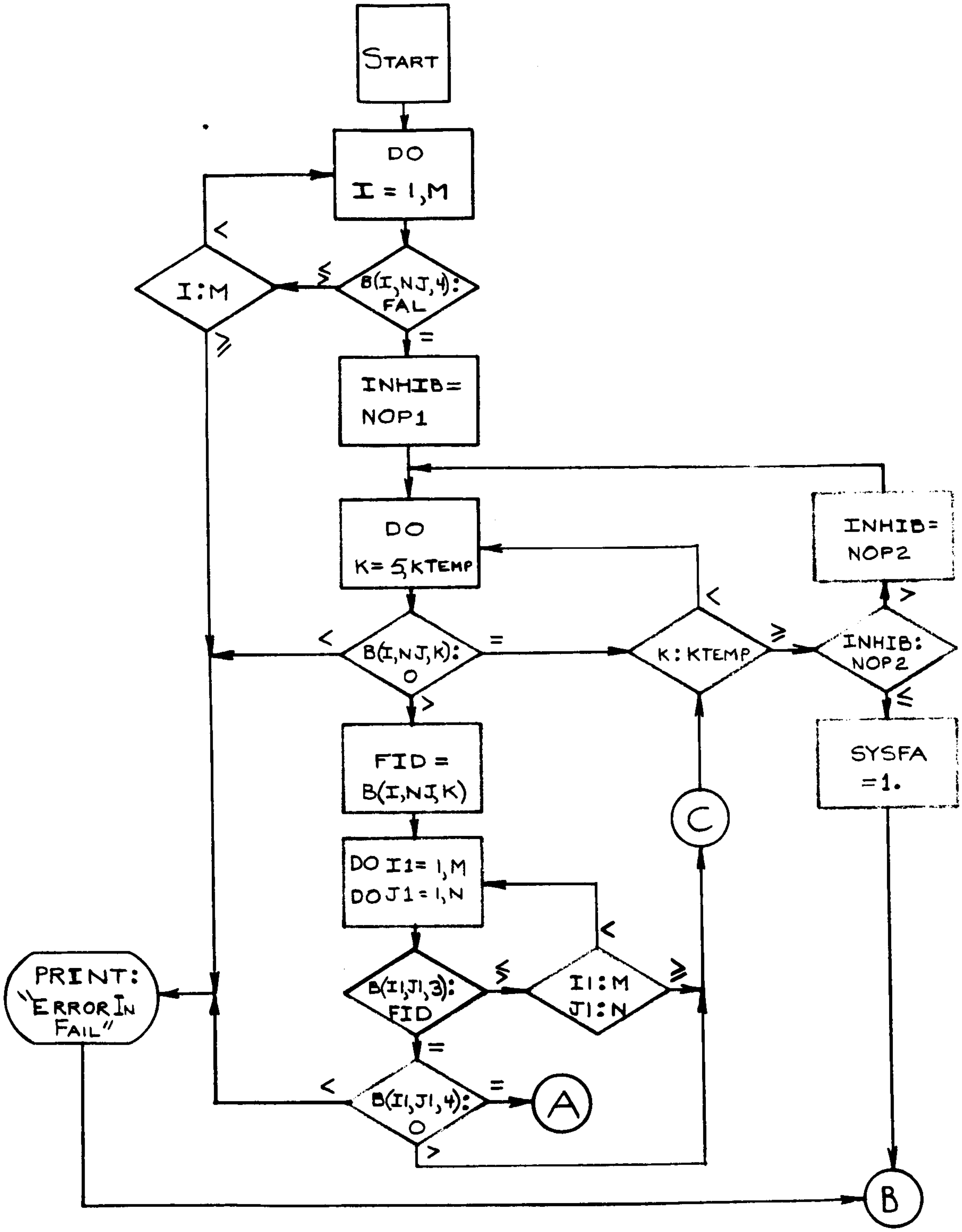

SUBROUTINE LIST (CONT)



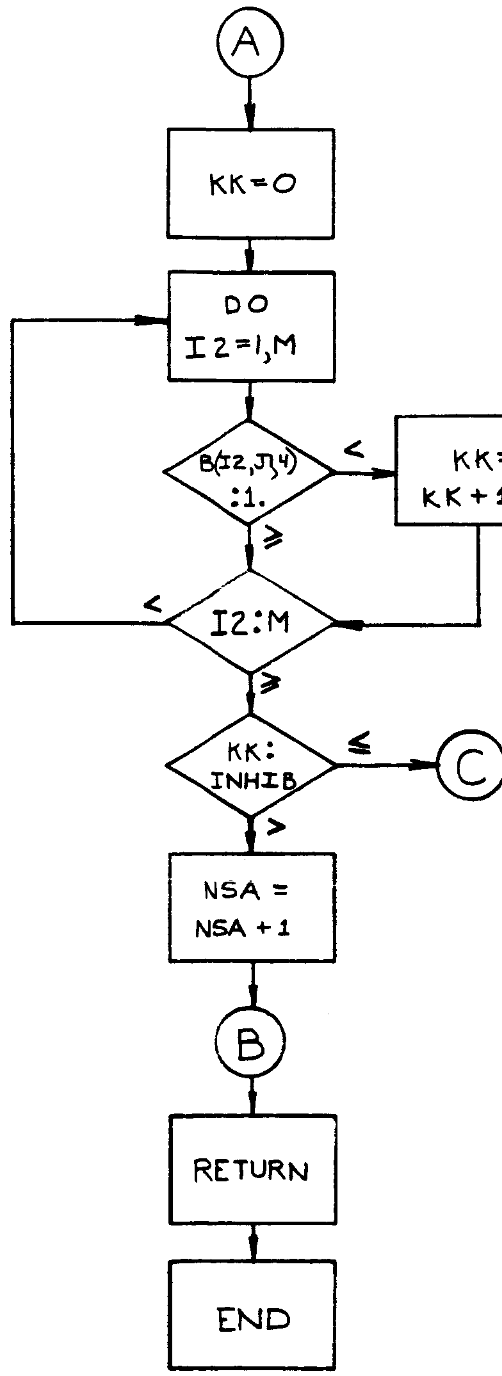

SUBROUTINE LIST



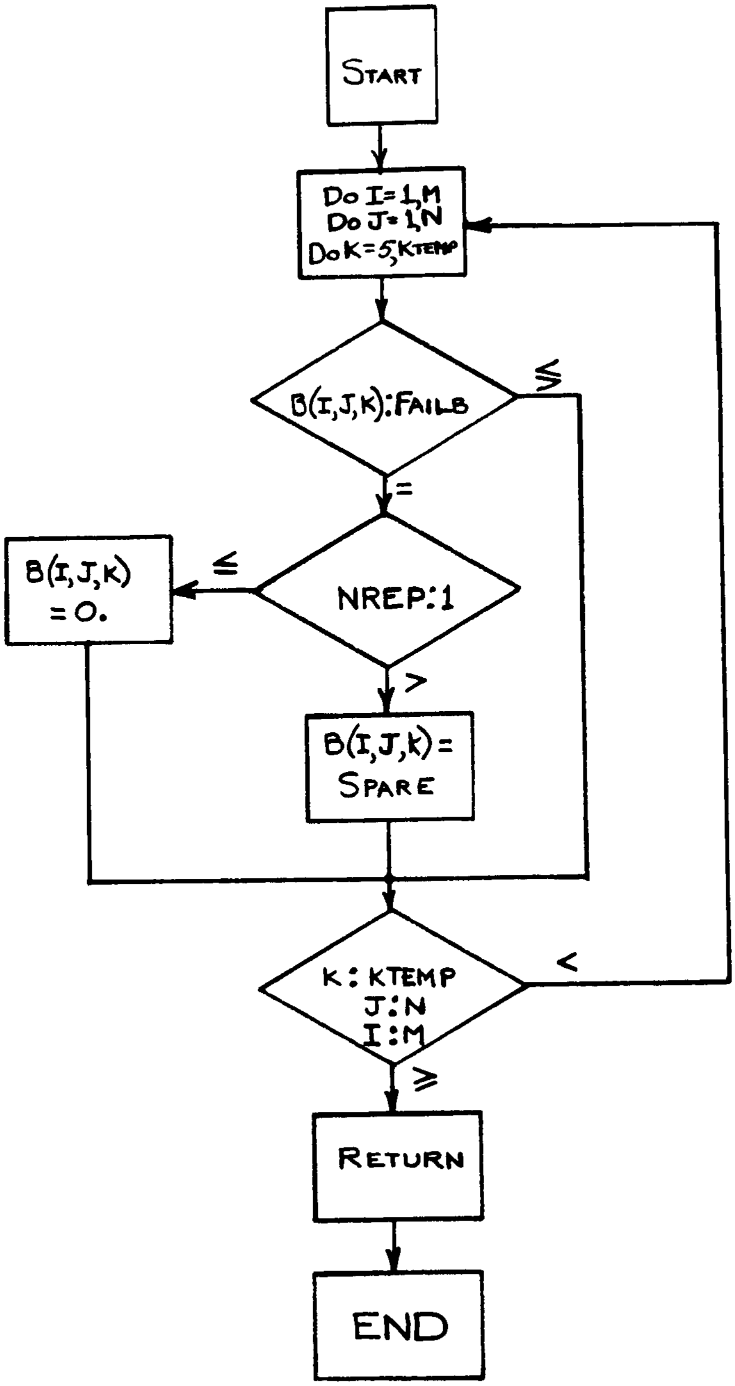

SUBROUTINE RANLIS



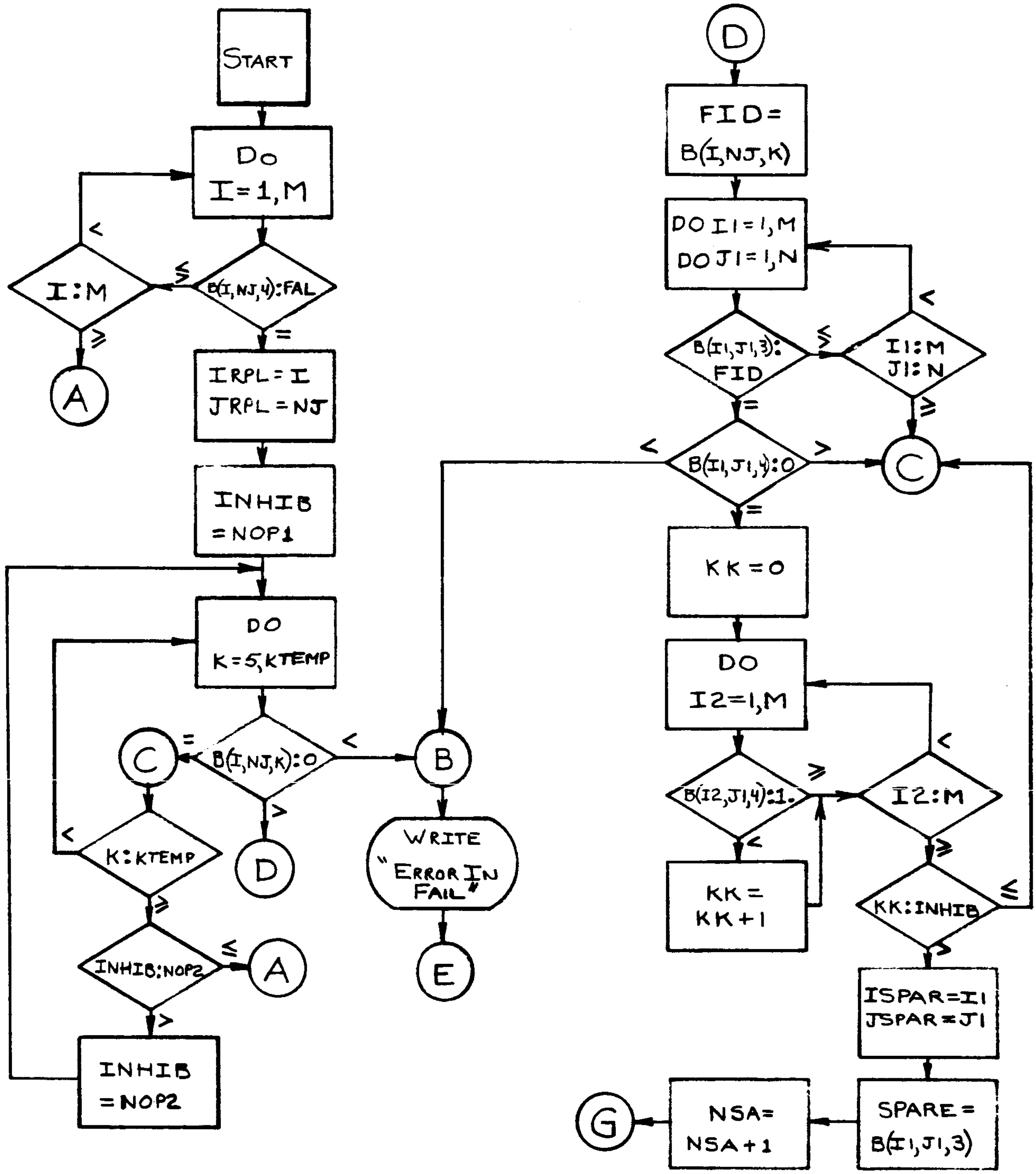

Subroutine Sback



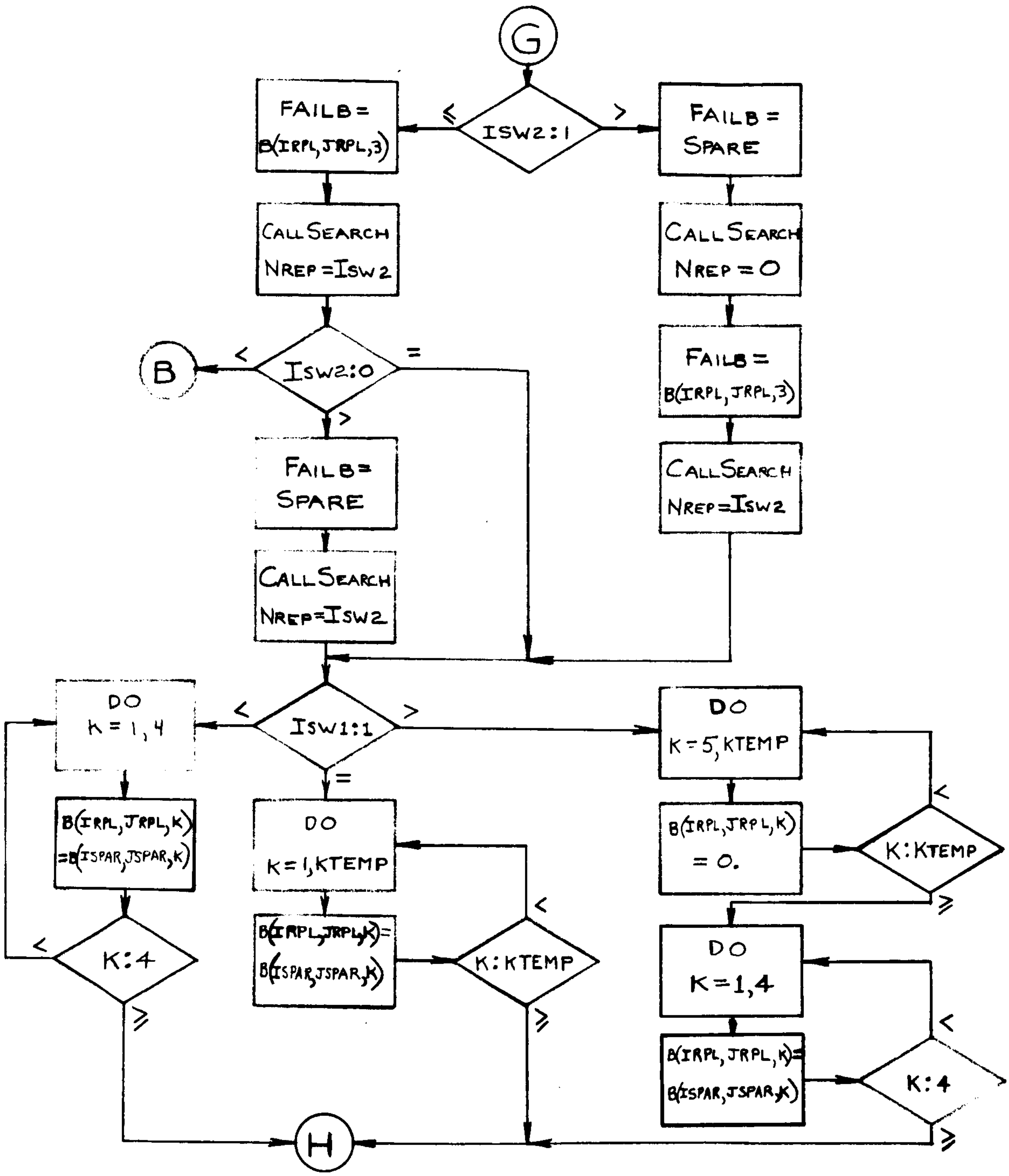

SUBROUTINE SBACK (CONT.)



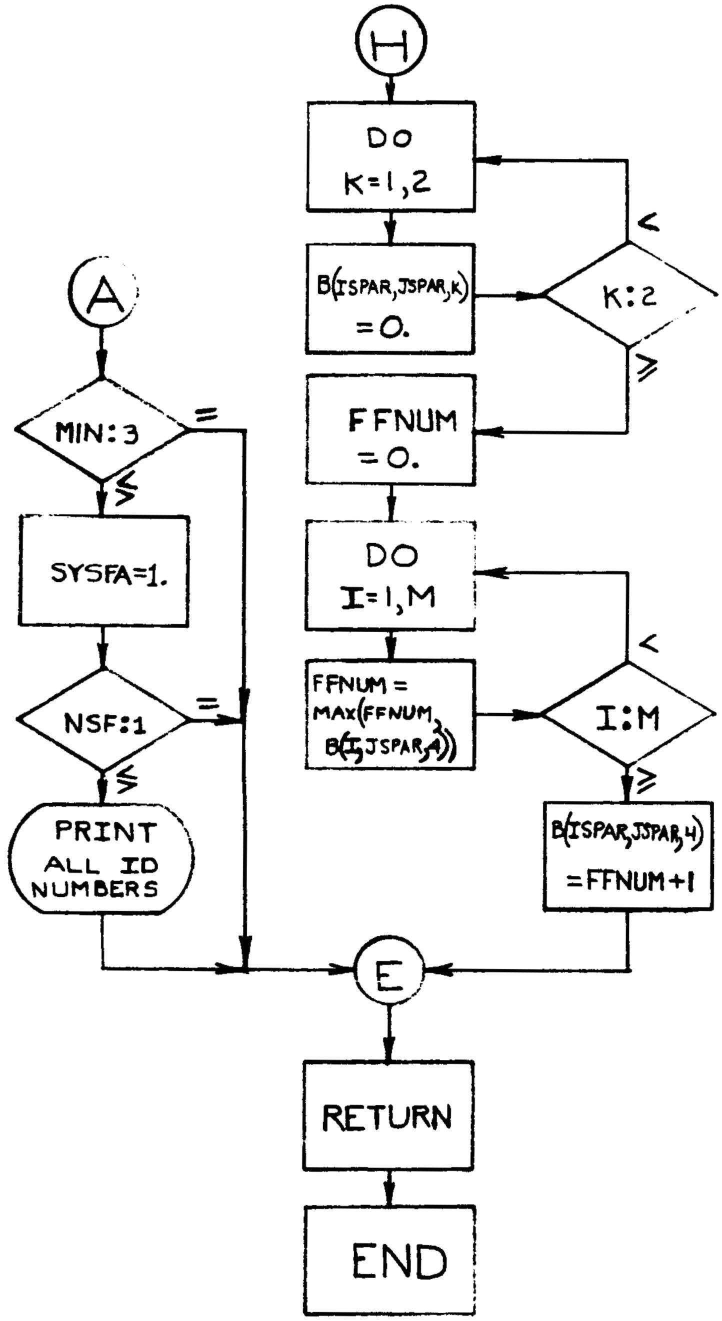

SUBROUTINE SEARCH



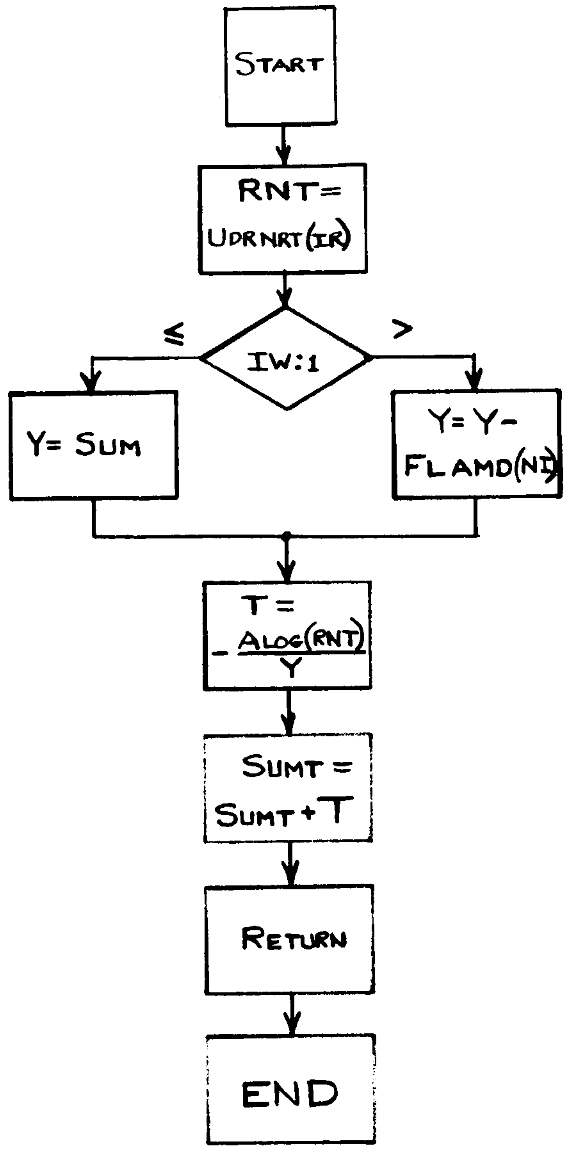

SUBROUTINE SWITCH




SUBROUTINE SWITCH (CONT.)




SUBROUTINE SWITCH (CONT.)




SUBROUTINE TIME

# APPENDIX B

## EXAMPLES OF THE PRINTOUT OPTIONS

This section contains examples of the printout obtained with the two printout options: NSF = 0, no-sample-format; and NSF = 1, sample-format. The no-sample-format printout is the complete printout obtained from 500 simulation of a typical failure responsive system. The sample format printout was obtained from only 5 simulations of a typical system, so that the form of the printout could be shown without producing an inordinately large amount of printout. Therefore the results obtained for this system are not particularly meaningful.

```
FAILURE RESPONSIVE SYSTEM ORGANIZATION

STRATEGY DESCRIPTION


CONTROL CONSTANTS

LAMDA=  .150000-03
LAMDA=  .125000-03
LAMDA=  .125000-03
    M=   3
    N=  20
KTEMP=  10
   NR= 500    (NUMBER OF RUNS)
   IR=1003    (RANDOM NUMBER SET)
   MM=   1    (COMMON SPARE LIST FLAG)
  NRL=   0    (NON REPETITIVE LIST FLAG)
  MIN=   2    (OPR. BLOCKS REQ. TO RESOLVE ERROR)
 MBLL=   0    (MISSING BLOCK LIST LENGTH)
  NSF=   1    (NO SAMPLE FORMAT FLAG)
 ISW1=   0
 ISW2=   1
 NOP1=   2
 NOP2=   1
ISBAC=   0
  NOS=   1
  NTC=   1
SPARE LIST
        19.         38.         57.         16.         35.         54.
```

| -X- | FREQUENCY OF FAILURES WITH X FAILED SUBSYSTEMS |
|---|---|
| 1 | 0. |
| 2 | 0. |
| 3 | 0. |
| 4 | 0. |
| 5 | 0. |
| 6 | 0. |
| 7 | 0. |
| 8 | 0. |
| 9 | 0. |
| 10 | 0. |
| 11 | 0. |
| 12 | 0. |
| 13 | 0. |
| 14 | 3. |
| 15 | 0. |
| 16 | 1. |
| 17 | 7. |
| 18 | 7. |
| 19 | 16. |
| 20 | 12. |
| 21 | 22. |
| 22 | 30. |
| 23 | 56. |
| 24 | 38. |
| 25 | 55. |
| 26 | 71. |
| 27 | 53. |
| 28 | 52. |
| 29 | 34. |
| 30 | 26. |
| 31 | 15. |
| 32 | 1. |
| 33 | 1. |
| 34 | 0. |
| 35 | 0. |
| 36 | 0. |
| 37 | 0. |
| 38 | 0. |
| 39 | 0. |
| 40 | 0. |
| 41 | 0. |
| 42 | 0. |
| 43 | 0. |
| 44 | 0. |
| 45 | 0. |
| 46 | 0. |
| 47 | 0. |
| 48 | 0. |
| 49 | 0. |
| 50 | 0. |

| COMPUTATION | TIME | PROB. OF FAILURE |
|---|---|---|
| 1 | 2062.6 | .201783-01 |
| 2 | 2268.4 | .359277-01 |
| 3 | 2450.1 | .561257-01 |
| 4 | 2584.5 | .755035-01 |
| 5 | 2663.6 | .888326-01 |
| 6 | 2700.8 | .956030-01 |
| 7 | 2715.7 | .983978-01 |
| 8 | 2721.1 | .994349-01 |

AVG. NO. OF FAILURES= 25.15

10 PERCENT FAILED AT 2721.1

```
FAILURE RESPONSIVE SYSTEM ORGANIZATION

STRATEGY DESCRIPTION


CONTROL CONSTANTS

LAMDA=  .150000-03
LAMDA=  .125000-03
LAMDA=  .125000-03
    M=    3
    N=   20
KTEMP=   10
   NR=    5     (NUMBER OF RUNS)
   IR=1003      (RANDOM NUMBER SET)
   SM=    1     (COMMON SPARE LIST FLAG)
  NRL=    0     (NON REPETITIVE LIST FLAG)
  MIN=    2     (OPR. BLOCKS REQ. TO RESOLVE ERROR
 MBLL=    0     (MISSING BLOCK LIST LENGTH)
  NSF=    0     (NO SAMPLE FORMAT FLAG)
 ISW1=    0
 ISW2=    1
 NOP1=    2
 NOP2=    1
ISBAC=    0
  NOS=    1
  NTC=    1
```

NSF = 0

LLOCATIONS

```
( 1 , 1 , 1 )=     .000000
( 1 , 1 , 2 )=     .018750
( 1 , 2 , 1 )=     .018750
( 1 , 2 , 2 )=     .037500
( 1 , 3 , 1 )=     .037500
( 1 , 3 , 2 )=     .056250
( 1 , 4 , 1 )=     .056250
( 1 , 4 , 2 )=     .075000
( 1 , 5 , 1 )=     .075000
( 1 , 5 , 2 )=     .093750
( 1 , 6 , 1 )=     .093750
( 1 , 6 , 2 )=     .112500
( 1 , 7 , 1 )=     .112500
( 1 , 7 , 2 )=     .131250
( 1 , 8 , 1 )=     .131250
( 1 , 8 , 2 )=     .150000
( 1 , 9 , 1 )=     .150000
( 1 , 9 , 2 )=     .168750
( 1 ,10 , 1 )=     .168750
( 1 ,10 , 2 )=     .187500
( 1 ,11 , 1 )=     .187500
( 1 ,11 , 2 )=     .206250
( 1 ,12 , 1 )=     .206250
( 1 ,12 , 2 )=     .225000
( 1 ,13 , 1 )=     .225000
( 1 ,13 , 2 )=     .243750
( 1 ,14 , 1 )=     .243750
( 1 ,14 , 2 )=     .262500
( 1 ,15 , 1 )=     .262500
( 1 ,15 , 2 )=     .281250
( 1 ,16 , 1 )=     .281250
( 1 ,16 , 2 )=     .300000
( 1 ,17 , 1 )=     .300000
( 1 ,17 , 2 )=     .318750
( 1 ,18 , 1 )=     .318750
( 1 ,18 , 2 )=     .337500
( 1 ,19 , 1 )=     .337500
( 1 ,19 , 2 )=     .356250
( 1 ,20 , 1 )=     .356250
( 1 ,20 , 2 )=     .375000
( 2 , 1 , 1 )=     .375000
( 2 , 1 , 2 )=     .390625
```

| | |
|---|---|
| ( 2 , 2 , 1 )= | .390625 |
| ( 2 , 2 , 2 )= | .406250 |
| ( 2 , 3 , 1 )= | .406250 |
| ( 2 , 3 , 2 )= | .421875 |
| ( 2 , 4 , 1 )= | .421875 |
| ( 2 , 4 , 2 )= | .437500 |
| ( 2 , 5 , 1 )= | .437500 |
| ( 2 , 5 , 2 )= | .453125 |
| ( 2 , 6 , 1 )= | .453125 |
| ( 2 , 6 , 2 )= | .468750 |
| ( 2 , 7 , 1 )= | .468750 |
| ( 2 , 7 , 2 )= | .484375 |
| ( 2 , 8 , 1 )= | .484375 |
| ( 2 , 8 , 2 )= | .500000 |
| ( 2 , 9 , 1 )= | .500000 |
| ( 2 , 9 , 2 )= | .515625 |
| ( 2 ,10 , 1 )= | .515625 |
| ( 2 ,10 , 2 )= | .531250 |
| ( 2 ,11 , 1 )= | .531250 |
| ( 2 ,11 , 2 )= | .546875 |
| ( 2 ,12 , 1 )= | .546875 |
| ( 2 ,12 , 2 )= | .562500 |
| ( 2 ,13 , 1 )= | .562500 |
| ( 2 ,13 , 2 )= | .578125 |
| ( 2 ,14 , 1 )= | .578125 |
| ( 2 ,14 , 2 )= | .593750 |
| ( 2 ,15 , 1 )= | .593750 |
| ( 2 ,15 , 2 )= | .609375 |
| ( 2 ,16 , 1 )= | .609375 |
| ( 2 ,16 , 2 )= | .625000 |
| ( 2 ,17 , 1 )= | .625000 |
| ( 2 ,17 , 2 )= | .640625 |
| ( 2 ,18 , 1 )= | .640625 |
| ( 2 ,18 , 2 )= | .656250 |
| ( 2 ,19 , 1 )= | .656250 |
| ( 2 ,19 , 2 )= | .671875 |
| ( 2 ,20 , 1 )= | .671875 |
| ( 2 ,20 , 2 )= | .687500 |
| ( 3 , 1 , 1 )= | .687500 |
| ( 3 , 1 , 2 )= | .703125 |
| ( 3 , 2 , 1 )= | .703125 |
| ( 3 , 2 , 2 )= | .718750 |
| ( 3 , 3 , 1 )= | .718750 |

```
( 3 , 3 , 2 )=     .734375
( 3 , 4 , 1 )=     .734375
( 3 , 4 , 2 )=     .750000
( 3 , 5 , 1 )=     .750000
( 3 , 5 , 2 )=     .765625
( 3 , 6 , 1 )=     .765625
( 3 , 6 , 2 )=     .781250
( 3 , 7 , 1 )=     .781250
( 3 , 7 , 2 )=     .796875
( 3 , 8 , 1 )=     .796875
( 3 , 8 , 2 )=     .812500
( 3 , 9 , 1 )=     .812500
( 3 , 9 , 2 )=     .828125
( 3 ,10 , 1 )=     .828125
( 3 ,10 , 2 )=     .843750
( 3 ,11 , 1 )=     .843750
( 3 ,11 , 2 )=     .859375
( 3 ,12 , 1 )=     .859375
( 3 ,12 , 2 )=     .875000
( 3 ,13 , 1 )=     .875000
( 3 ,13 , 2 )=     .890625
( 3 ,14 , 1 )=     .890625
( 3 ,14 , 2 )=     .906250
( 3 ,15 , 1 )=     .906250
( 3 ,15 , 2 )=     .921875
( 3 ,16 , 1 )=     .921875
( 3 ,16 , 2 )=     .937500
( 3 ,17 , 1 )=     .937500
( 3 ,17 , 2 )=     .953125
( 3 ,18 , 1 )=     .953125
( 3 ,18 , 2 )=     .968750
( 3 ,19 , 1 )=     .968750
( 3 ,19 , 2 )=     .984375
( 3 ,20 , 1 )=     .984375
( 3 ,20 , 2 )=    1.000000
```

```
ID NUMBERS
( 1 , 1 ,3)=   1.
( 1 , 2 ,3)=   2.
( 1 , 3 ,3)=   3.
( 1 , 4 ,3)=   4.
( 1 , 5 ,3)=   5.
( 1 , 6 ,3)=   6.
( 1 , 7 ,3)=   7.
( 1 , 8 ,3)=   8.
( 1 , 9 ,3)=   9.
( 1 ,10 ,3)=  10.
( 1 ,11 ,3)=  11.
( 1 ,12 ,3)=  12.
( 1 ,13 ,3)=  13.
( 1 ,14 ,3)=  14.
( 1 ,15 ,3)=  15.
( 1 ,16 ,3)=  16.
( 1 ,17 ,3)=  17.
( 1 ,18 ,3)=  18.
( 1 ,19 ,3)=  19.
( 1 ,20 ,3)=  20.
( 2 , 1 ,3)=  21.
( 2 , 2 ,3)=  22.
( 2 , 3 ,3)=  23.
( 2 , 4 ,3)=  24.
( 2 , 5 ,3)=  25.
( 2 , 6 ,3)=  26.
( 2 , 7 ,3)=  27.
( 2 , 8 ,3)=  28.
( 2 , 9 ,3)=  29.
( 2 ,10 ,3)=  30.
( 2 ,11 ,3)=  31.
( 2 ,12 ,3)=  32.
( 2 ,13 ,3)=  33.
( 2 ,14 ,3)=  34.
( 2 ,15 ,3)=  35.
( 2 ,16 ,3)=  36.
( 2 ,17 ,3)=  37.
( 2 ,18 ,3)=  38.
( 2 ,19 ,3)=  39.
( 2 ,20 ,3)=  40.
( 3 , 1 ,3)=  41.
( 3 , 2 ,3)=  42.
```

```
( 3 , 3 ,3)=  43.
( 3 , 4 ,3)=  44.
( 3 , 5 ,3)=  45.
( 3 , 6 ,3)=  46.
( 3 , 7 ,3)=  47.
( 3 , 8 ,3)=  48.
( 3 , 9 ,3)=  49.
( 3 ,10 ,3)=  50.
( 3 ,11 ,3)=  51.
( 3 ,12 ,3)=  52.
( 3 ,13 ,3)=  53.
( 3 ,14 ,3)=  54.
( 3 ,15 ,3)=  55.
( 3 ,16 ,3)=  56.
( 3 ,17 ,3)=  57.
( 3 ,18 ,3)=  58.
( 3 ,19 ,3)=  59.
( 3 ,20 ,3)=  60.
```

SPARE LIST
19. 38. 57. 16. 35. 54.

| B( 1 , 1 ) | K |
|---|---|
| 20 | 5 |
| 39 | 6 |
| 58 | 7 |
| 17 | 8 |
| 36 | 9 |
| 55 | 10 |

| B( 1 , 2 ) | K |
|---|---|
| 1 | 5 |
| 40 | 6 |
| 59 | 7 |
| 18 | 8 |
| 37 | 9 |
| 56 | 10 |

| B( 1 , 3 ) | K |
|---|---|
| 2 | 5 |
| 21 | 6 |
| 60 | 7 |
| 19 | 8 |
| 38 | 9 |
| 57 | 10 |

| B( 1 , 4 ) | K |
|---|---|
| 3 | 5 |
| 22 | 6 |
| 41 | 7 |
| 20 | 8 |
| 39 | 9 |
| 58 | 10 |

| B( 1 , 5 ) | K |
|---|---|
| 4 | 5 |
| 23 | 6 |
| 42 | 7 |
| 1 | 8 |
| 40 | 9 |
| 59 | 10 |

| | |
|---|---|
| B( 1 , 6 ) | K |
| 5 | 5 |
| 24 | 6 |
| 43 | 7 |
| 2 | 8 |
| 21 | 9 |
| 60 | 10 |
| B( 1 , 7 ) | K |
| 6 | 5 |
| 25 | 6 |
| 44 | 7 |
| 3 | 8 |
| 22 | 9 |
| 41 | 10 |
| B( 1 , 8 ) | K |
| 7 | 5 |
| 26 | 6 |
| 45 | 7 |
| 4 | 8 |
| 23 | 9 |
| 42 | 10 |
| B( 1 , 9 ) | K |
| 8 | 5 |
| 27 | 6 |
| 46 | 7 |
| 5 | 8 |
| 24 | 9 |
| 43 | 10 |
| B( 1 ,10 ) | K |
| 9 | 5 |
| 28 | 6 |
| 47 | 7 |
| 6 | 8 |
| 25 | 9 |
| 44 | 10 |
| B( 1 ,11 ) | K |
| 10 | 5 |

| | |
|---|---|
| 29 | 6 |
| 48 | 7 |
| 7 | 8 |
| 26 | 9 |
| 45 | 10 |
| B( 1 ,12 ) | K |
| 11 | 5 |
| 30 | 6 |
| 49 | 7 |
| 8 | 8 |
| 27 | 9 |
| 46 | 10 |
| B( 1 ,13 ) | K |
| 12 | 5 |
| 31 | 6 |
| 50 | 7 |
| 9 | 8 |
| 28 | 9 |
| 47 | 10 |
| B( 1 ,14 ) | K |
| 13 | 5 |
| 32 | 6 |
| 51 | 7 |
| 10 | 8 |
| 29 | 9 |
| 48 | 10 |
| B( 1 ,15 ) | K |
| 14 | 5 |
| 33 | 6 |
| 52 | 7 |
| 11 | 8 |
| 30 | 9 |
| 49 | 10 |
| B( 1 ,16 ) | K |
| 15 | 5 |
| 34 | 6 |
| 53 | 7 |
| 12 | 8 |

| | |
|---|---|
| 31 | 9 |
| 50 | 10 |
| B( 1 ,17 ) | K |
| 16 | 5 |
| 35 | 6 |
| 54 | 7 |
| 13 | 8 |
| 32 | 9 |
| 51 | 10 |
| B( 1 ,18 ) | K |
| 17 | 5 |
| 36 | 6 |
| 55 | 7 |
| 14 | 8 |
| 33 | 9 |
| 52 | 10 |
| B( 1 ,19 ) | K |
| 18 | 5 |
| 37 | 6 |
| 56 | 7 |
| 15 | 8 |
| 34 | 9 |
| 53 | 10 |
| B( 1 ,20 ) | K |
| 19 | 5 |
| 38 | 6 |
| 57 | 7 |
| 16 | 8 |
| 35 | 9 |
| 54 | 10 |
| B( 2 , 1 ) | K |
| 20 | 5 |
| 39 | 6 |
| 58 | 7 |
| 17 | 8 |
| 36 | 9 |
| 55 | 10 |

| B( 2 , 2 ) | K |
|---|---|
| 1 | 5 |
| 40 | 6 |
| 59 | 7 |
| 18 | 8 |
| 37 | 9 |
| 56 | 10 |

| B( 2 , 3 ) | K |
|---|---|
| 2 | 5 |
| 21 | 6 |
| 60 | 7 |
| 19 | 8 |
| 38 | 9 |
| 57 | 10 |

| B( 2 , 4 ) | K |
|---|---|
| 3 | 5 |
| 22 | 6 |
| 41 | 7 |
| 20 | 8 |
| 39 | 9 |
| 58 | 10 |

| B( 2 , 5 ) | K |
|---|---|
| 4 | 5 |
| 23 | 6 |
| 42 | 7 |
| 1 | 8 |
| 40 | 9 |
| 59 | 10 |

| B( 2 , 6 ) | K |
|---|---|
| 5 | 5 |
| 24 | 6 |
| 43 | 7 |
| 2 | 8 |
| 21 | 9 |
| 60 | 10 |

| B( 2 , 7 ) | K |
|---|---|
| 6 | 5 |
| 25 | 6 |

| | |
|---|---|
| 44 | 7 |
| 3 | 8 |
| 22 | 9 |
| 41 | 10 |
| B( 2 , 8 ) | K |
| 7 | 5 |
| 26 | 6 |
| 45 | 7 |
| 4 | 8 |
| 23 | 9 |
| 42 | 10 |
| B( 2 , 9 ) | K |
| 8 | 5 |
| 27 | 6 |
| 46 | 7 |
| 5 | 8 |
| 24 | 9 |
| 43 | 10 |
| B( 2 ,10 ) | K |
| 9 | 5 |
| 28 | 6 |
| 47 | 7 |
| 6 | 8 |
| 25 | 9 |
| 44 | 10 |
| B( 2 ,11 ) | K |
| 10 | 5 |
| 29 | 6 |
| 48 | 7 |
| 7 | 8 |
| 26 | 9 |
| 45 | 10 |
| B( 2 ,12 ) | K |
| 11 | 5 |
| 30 | 6 |
| 49 | 7 |
| 8 | 8 |
| 27 | 9 |

| | |
|---|---|
| 46 | 10 |
| B( 2 ,13 ) | K |
| 12 | 5 |
| 31 | 6 |
| 50 | 7 |
| 9 | 8 |
| 28 | 9 |
| 47 | 10 |
| B( 2 ,14 ) | K |
| 13 | 5 |
| 32 | 6 |
| 51 | 7 |
| 10 | 8 |
| 29 | 9 |
| 48 | 10 |
| B( 2 ,15 ) | K |
| 14 | 5 |
| 33 | 6 |
| 52 | 7 |
| 11 | 8 |
| 30 | 9 |
| 49 | 10 |
| B( 2 ,16 ) | K |
| 15 | 5 |
| 34 | 6 |
| 53 | 7 |
| 12 | 8 |
| 31 | 9 |
| 50 | 10 |
| B( 2 ,17 ) | K |
| 16 | 5 |
| 35 | 6 |
| 54 | 7 |
| 13 | 8 |
| 32 | 9 |
| 51 | 10 |
| B( 2 ,18 ) | K |

| | |
|---|---|
| 17 | 5 |
| 36 | 6 |
| 55 | 7 |
| 14 | 8 |
| 33 | 9 |
| 52 | 10 |
| B( 2 ,19 ) | K |
| 18 | 5 |
| 37 | 6 |
| 56 | 7 |
| 15 | 8 |
| 34 | 9 |
| 53 | 10 |
| B( 2 ,20 ) | K |
| 19 | 5 |
| 38 | 6 |
| 57 | 7 |
| 16 | 8 |
| 35 | 9 |
| 54 | 10 |
| B( 3 , 1 ) | K |
| 20 | 5 |
| 39 | 6 |
| 58 | 7 |
| 17 | 8 |
| 36 | 9 |
| 55 | 10 |
| B( 3 , 2 ) | K |
| 1 | 5 |
| 40 | 6 |
| 59 | 7 |
| 18 | 8 |
| 37 | 9 |
| 56 | 10 |
| B( 3 , 3 ) | K |
| 2 | 5 |
| 21 | 6 |
| 60 | 7 |

| | |
|---|---|
| 19 | 8 |
| 38 | 9 |
| 57 | 10 |

| B( 3 , 4 ) | K |
|---|---|
| 3 | 5 |
| 22 | 6 |
| 41 | 7 |
| 20 | 8 |
| 39 | 9 |
| 58 | 10 |

| B( 3 , 5 ) | K |
|---|---|
| 4 | 5 |
| 23 | 6 |
| 42 | 7 |
| 1 | 8 |
| 40 | 9 |
| 59 | 10 |

| B( 3 , 6 ) | K |
|---|---|
| 5 | 5 |
| 24 | 6 |
| 43 | 7 |
| 2 | 8 |
| 21 | 9 |
| 60 | 10 |

| B( 3 , 7 ) | K |
|---|---|
| 6 | 5 |
| 25 | 6 |
| 44 | 7 |
| 3 | 8 |
| 22 | 9 |
| 41 | 10 |

| B( 3 , 8 ) | K |
|---|---|
| 7 | 5 |
| 26 | 6 |
| 45 | 7 |
| 4 | 8 |
| 23 | 9 |
| 42 | 10 |

| B( 3 , 9 ) | K |
|---|---|
| 8 | 5 |
| 27 | 6 |
| 46 | 7 |
| 5 | 8 |
| 24 | 9 |
| 43 | 10 |

| B( 3 ,10 ) | K |
|---|---|
| 9 | 5 |
| 28 | 6 |
| 47 | 7 |
| 6 | 8 |
| 25 | 9 |
| 44 | 10 |

| B( 3 ,11 ) | K |
|---|---|
| 10 | 5 |
| 29 | 6 |
| 48 | 7 |
| 7 | 8 |
| 26 | 9 |
| 45 | 10 |

| B( 3 ,12 ) | K |
|---|---|
| 11 | 5 |
| 30 | 6 |
| 49 | 7 |
| 8 | 8 |
| 27 | 9 |
| 46 | 10 |

| B( 3 ,13 ) | K |
|---|---|
| 12 | 5 |
| 31 | 6 |
| 50 | 7 |
| 9 | 8 |
| 28 | 9 |
| 47 | 10 |

| B( 3 ,14 ) | K |
|---|---|
| 13 | 5 |

```
        32                        6
        51                        7
        10                        8
        29                        9
        48                       10
-------------------------------------
     B( 3 ,15 )                   K
        14                        5
        33                        6
        52                        7
        11                        8
        30                        9
        49                       10
-------------------------------------
     B( 3 ,16 )                   K
        15                        5
        34                        6
        53                        7
        12                        8
        31                        9
        50                       10
-------------------------------------
     B( 3 ,17 )                   K
        16                        5
        35                        6
        54                        7
        13                        8
        32                        9
        51                       10
-------------------------------------
     B( 3 ,18 )                   K
        17                        5
        36                        6
        55                        7
        14                        8
        33                        9
        52                       10
-------------------------------------
     B( 3 ,19 )                   K
        18                        5
        37                        6
        56                        7
        15                        8
        34                        9
        53                       10
-------------------------------------
     B( 3 ,20 )                   K
        19                        5
        38                        6
        57                        7
        16                        8
        35                        9
        54                       10
```

SEQUENCE OF FAILURES

| RUN NO. | FAILURE I.D. | | | FAILURE RAN. NO. | TIMERAN. NO. | HOURS BETWEEN FAILURES | FAILURE NO. |
|---|---|---|---|---|---|---|---|
| 1 | | | | | | | |
| | 2 | 17 | 37. | .633718 | .118438 | 266.7 | 1 |
| | 1 | 19 | 19. | .345467 | .169745 | 225.9 | 2 |
| | 1 | 12 | 12. | .222605 | .001718 | 826.8 | 3 |
| | 1 | 18 | 18. | .327796 | .125551 | 274.8 | 4 |
| | 3 | 4 | 44. | .739773 | .720780 | 44.1 | 5 |
| | 2 | 1 | 21. | .377195 | .761279 | 37.4 | 6 |
| | 2 | 10 | 30. | .528732 | .601157 | 70.9 | 7 |
| | 1 | 20 | 20. | .363486 | .838925 | 25.0 | 8 |
| | 3 | 17 | 57. | .949500 | .111992 | 317.3 | 9 |
| | 2 | 14 | 34. | .583859 | .350528 | 154.7 | 10 |
| | 3 | 10 | 50. | .835967 | .212533 | 232.9 | 11 |
| | 1 | 6 | 6. | .100626 | .640186 | 68.6 | 12 |
| | 3 | 1 | 41. | .690897 | .023837 | 586.1 | 13 |
| | 1 | 17 | 17. | .317138 | .748940 | 46.4 | 14 |
| | 3 | 12 | 52. | .871381 | .946510 | 9.0 | 15 |
| | 1 | 4 | 4. | .071656 | .285400 | 210.7 | 16 |
| | 3 | 2 | 42. | .704080 | .349497 | 180.5 | 17 |
| | 2 | 5 | 25. | .450313 | .053273 | 514.4 | 18 |
| | 3 | 4 | 3. | .049669 | .449307 | 143.5 | 19 |
| | 2 | 8 | 28. | .489722 | .697681 | 66.1 | 20 |
| | 2 | 11 | 31. | .539377 | .438329 | 154.9 | 21 |
| | 1 | 15 | 15. | .266802 | .701153 | 68.6 | 22 |
| | 1 | 5 | 5. | .086711 | .377772 | 193.7 | 23 |
| | 1 | 10 | 10. | .182648 | .897724 | 22.1 | 24 |

| | | | | | | |
|---|---|---|---|---|---|---|
| 1. | 55. | 41. | | | | |
| 2. | 22. | 42. | | | | |
| 3. | 23. | 43. | | | | |
| 22. | 24. | 3. | | | | |
| 5. | 23. | 45. | | | | |
| 6. | 26. | 46. | | | | |
| 7. | 27. | 47. | | | | |
| 8. | 28. | 48. | | | | |
| 9. | 29. | 49. | | | | |
| 10. | 9. | 47. | | | | |
| 7. | 31. | 51. | | | | |
| 11. | 32. | 52. | | | | |
| 13. | 33. | 53. | | | | |
| 14. | 34. | 54. | | | | |
| 15. | 35. | 14. | | | | |
| 16. | 36. | 56. | | | | |
| 17. | 16. | 13. | | | | |
| 18. | 38. | 58. | | | | |
| 19. | 39. | 59. | | | | |
| 20. | 40. | 60. | | | | |
| | 3 10 | 47. | .788687 | .411827 | 186.8 | 25 |
| AVG. TIME= | 197.1 | | | TOTAL OPERATING TIME= | 4928.1 | TOTAL FAILURES= 25 |

SEQUENCE OF FAILURES

| RUN NO. | FAILURE I.D. | | | FAILURE RAN. NO. | TIMERAN. NO. | HOURS BETWEEN FAILURES | FAILURE NO. |
|---|---|---|---|---|---|---|---|
| 2 | | | | | | | |
| | 3 | 2 | 42. | .704772 | .917278 | 10.8 | 1 |
| | 3 | 3 | 43. | .729007 | .305677 | 150.5 | 2 |
| | 1 | 9 | 9. | .157247 | .823257 | 25.2 | 3 |
| | 2 | 17 | 37. | .629546 | .666180 | 53.4 | 4 |
| | 2 | 3 | 23. | .411238 | .108060 | 297.7 | 5 |
| | 2 | 10 | 30. | .525505 | .137673 | 269.8 | 6 |
| | 2 | 4 | 24. | .431131 | .239656 | 197.7 | 7 |
| | 2 | 9 | 29. | .509774 | .594337 | 73.3 | 8 |
| | 3 | 20 | 60. | .986289 | .269743 | 187.9 | 9 |
| | 1 | 12 | 12. | .218124 | .640359 | 65.3 | 10 |
| | 1 | 18 | 18. | .335142 | .641670 | 66.5 | 11 |
| | 2 | 18 | 38. | .642202 | .897694 | 16.5 | 12 |
| | 1 | 10 | 10. | .182985 | .047219 | 477.0 | 13 |
| | 1 | 17 | 17. | .311996 | .790978 | 37.5 | 14 |
| | 3 | 16 | 56. | .923952 | .620822 | 77.8 | 15 |
| | 3 | 6 | 46. | .767436 | .296354 | 202.7 | 16 |
| | 2 | 13 | 33. | .570303 | .794444 | 39.2 | 17 |
| | 1 | 14 | 14. | .253839 | .846704 | 29.1 | 18 |
| | 1 | 3 | 3. | .044407 | .424644 | 153.6 | 19 |
| | 1 | 11 | 11. | .199954 | .374048 | 181.3 | 20 |
| | 2 | 2 | 22. | .397896 | .117721 | 403.7 | 21 |
| | 2 | 6 | 26. | .462135 | .456796 | 151.4 | 22 |
| | 3 | 3 | 21. | .389833 | .021073 | 764.3 | 23 |
| | 2 | 11 | 31. | .535054 | .780075 | 50.4 | 24 |
| | 2 | 14 | 51. | .858530 | .339127 | 225.3 | 25 |

```
          1.        21.        41.
          2.        22.         1.
          2.        19.        21.
          4.        24.        44.
          5.        25.        45.
          6.        26.         5.
          7.        27.        47.
          8.        28.        48.
          8.        29.        49.
         10.        47.        50.
         48.        31.        45.
         12.        32.        52.
         13.        33.        53.
         13.        51.        54.
         15.        35.        55.
         16.        34.        56.
         17.        35.        57.
         36.        38.        58.
         19.        39.        59.
         20.        40.        60.
                1   6     6.          .110097          .518007             141.5          26
AVG. TIME=   167.3                                     TOTAL OPERATING TIME= 4349.2   TOTAL FAILURES= 26
```

SEQUENCE OF FAILURES

| RUN NO. | FAILURE I.D. | | | FAILURE RAN. NO. | TIMERAN. NO. | HOURS BETWEEN FAILURES | FAILURE NO. |
|---|---|---|---|---|---|---|---|
| 3 | | | | | | | |
| | 3 | 6 | 46. | .775827 | .716119 | 41.7 | 1 |
| | 1 | 7 | 7. | .112547 | .441609 | 104.1 | 2 |
| | 1 | 5 | 5. | .091888 | .644796 | 57.0 | 3 |
| | 3 | 16 | 56. | .931297 | .206809 | 208.0 | 4 |
| | 2 | 8 | 28. | .495339 | .673814 | 53.0 | 5 |
| | 2 | 9 | 29. | .514066 | .125521 | 283.3 | 6 |
| | 3 | 17 | 57. | .950581 | .216715 | 212.4 | 7 |
| | 2 | 18 | 38. | .656062 | .349748 | 148.5 | 8 |
| | 1 | 2 | 2. | .023077 | .427494 | 122.7 | 9 |
| | 3 | 11 | 51. | .852389 | .930191 | 10.6 | 10 |
| | 3 | 10 | 50. | .834316 | .070808 | 396.7 | 11 |
| | 1 | 10 | 10. | .183779 | .155202 | 285.5 | 12 |
| | 3 | 19 | 59. | .979369 | .571691 | 87.4 | 13 |
| | 1 | 18 | 18. | .334449 | .527542 | 102.3 | 14 |
| | 3 | 14 | 54. | .891060 | .331011 | 180.5 | 15 |
| | 3 | 10 | 44. | .744110 | .530766 | 105.6 | 16 |
| | 2 | 19 | 39. | .663490 | .646197 | 74.3 | 17 |
| | 1 | 4 | 4. | .068951 | .303488 | 208.3 | 18 |
| | 3 | 8 | 48. | .803598 | .343912 | 190.6 | 19 |
| | 3 | 18 | 58. | .962578 | .035540 | 609.5 | 20 |
| | 1 | 14 | 14. | .258375 | .596774 | 96.9 | 21 |
| | 3 | 1 | 41. | .696065 | .485940 | 138.8 | 22 |
| | 1 | 16 | 16. | .288551 | .140857 | 388.1 | 23 |
| | 1 | 20 | 20. | .368954 | .862225 | 30.3 | 24 |
| | 2 | 14 | 34. | .591543 | .873795 | 28.3 | 25 |
| | 2 | 5 | 25. | .449660 | .775613 | 54.6 | 26 |
| | 3 | 20 | 60. | .990237 | .918953 | 18.7 | 27 |

```
                1.        21.        41.
                2.        22.        42.
                3.        23.        43.
                4.        24.         3.
               23.        25.        45.
                6.        26.        46.
                7.        27.        47.
                8.        26.        48.
                9.        29.        49.
                9.        30.        44.
               11.        31.        51.
               12.        32.        52.
               13.        33.        53.
               32.        34.        13.
               15.        35.        55.
               16.        36.        15.
               17.        37.        57.
               52.        55.        58.
               19.        39.        53.
               19.        40.        60.
                       1  14    32.        .547922        .128546                    468.9                    28
AVG. TIME=     168.1                                      TOTAL OPERATING TIME= 4706.7    TOTAL FAILURES= 28
```

SEQUENCE OF FAILURES

| RUN NO. | FAILURE I.D. | | | FAILURE RAN. NO. | TIMERAN. NO. | HOURS BETWEEN FAILURES | FAILURE NO. |
|---|---|---|---|---|---|---|---|
| 4 | | | | | | | |
| | 3 | 11 | 51. | .846881 | .514959 | 83.0 | 1 |
| | 2 | 8 | 28. | .496088 | .577541 | 69.7 | 2 |
| | 1 | 2 | 2. | .026529 | .466334 | 98.8 | 3 |
| | 1 | 13 | 13. | .233047 | .747438 | 38.4 | 4 |
| | 3 | 5 | 45. | .754168 | .882619 | 16.8 | 5 |
| | 3 | 9 | 49. | .822124 | .890460 | 15.8 | 6 |
| | 2 | 19 | 39. | .661506 | .024494 | 515.2 | 7 |
| | 3 | 8 | 48. | .802351 | .105160 | 318.3 | 8 |
| | 1 | 4 | 4. | .059866 | .585244 | 77.4 | 9 |
| | 1 | 12 | 12. | .220022 | .333490 | 162.1 | 10 |
| | 3 | 3 | 43. | .725355 | .412710 | 133.1 | 11 |
| | 3 | 6 | 46. | .775753 | .096163 | 358.9 | 12 |
| | 3 | 17 | 57. | .950804 | .743088 | 46.4 | 13 |
| | 3 | 20 | 60. | .991141 | .484597 | 115.4 | 14 |
| | 1 | 8 | 8. | .137579 | .364587 | 164.7 | 15 |
| | 2 | 12 | 32. | .559545 | .996041 | .7 | 16 |
| | 2 | 13 | 33. | .567598 | .812531 | 35.3 | 17 |
| | 1 | 19 | 19. | .353356 | .841119 | 30.2 | 18 |
| | 1 | 10 | 10. | .174593 | .701247 | 63.7 | 19 |
| | 1 | 13 | 31. | .546321 | .584554 | 99.0 | 20 |
| | 1 | 15 | 15. | .274408 | .509745 | 127.7 | 21 |
| | 2 | 8 | 7. | .120306 | .891882 | 22.2 | 22 |
| | 1 | 12 | 30. | .529614 | .202855 | 319.1 | 23 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | 1. | 21. | | 41. | | | |
| | 2. | 22. | | 42. | | | |
| | 3. | 23. | | 43. | | | |
| | 4. | 24. | | 44. | | | |
| | 5. | 25. | | 45. | | | |
| | 6. | 26. | | 46. | | | |
| | 7. | 27. | | 47. | | | |
| | 23. | 7. | | 26. | | | |
| | 9. | 29. | | 49. | | | |
| | 10. | 9. | | 50. | | | |
| | 11. | 31. | | 51. | | | |
| | 30. | 27. | | 52. | | | |
| | 31. | 50. | | 53. | | | |
| | 14. | 34. | | 54. | | | |
| | 15. | 35. | | 55. | | | |
| | 16. | 36. | | 56. | | | |
| | 17. | 37. | | 57. | | | |
| | 18. | 38. | | 58. | | | |
| | 19. | 18. | | 59. | | | |
| | 20. | 40. | | 60. | | | |
| | 2 | 12 | 27. | | .469908 | .843959 | 34.8 | 24 |
| AVG. TIME= | 122.8 | | | | | TOTAL OPERATING TIME= 2946.6 | | TOTAL FAILURES= 24 |

SEQUENCE OF FAILURES

| RUN NO. | FAILURE I.D. | FAILURE RAN. NO. | TIMERAN. NO. | HOURS BETWEEN FAILURES | FAILURE NO. |
|---|---|---|---|---|---|
| 5 | | | | | |
| | 3 15 55. | .917935 | .313799 | 144.9 | 1 |
| | 3 13 53. | .879628 | .713472 | 42.9 | 2 |
| | 3 18 58. | .961171 | .287873 | 160.7 | 3 |
| | 2 11 31. | .534798 | .288068 | 163.2 | 4 |
| | 2 4 24. | .425634 | .269899 | 174.6 | 5 |
| | 3 19 59. | .970433 | .266853 | 179.1 | 6 |
| | 2 9 29. | .514302 | .372430 | 136.2 | 7 |
| | 3 10 50. | .832273 | .948639 | 7.4 | 8 |
| | 3 12 52. | .860571 | .096655 | 333.8 | 9 |
| | 3 4 44. | .742995 | .120918 | 307.3 | 10 |
| | 3 16 56. | .929083 | .266932 | 195.7 | 11 |
| | 2 19 39. | .670218 | .572137 | 84.3 | 12 |
| | 2 16 36. | .620635 | .567158 | 87.2 | 13 |
| | 1 5 5. | .091223 | .638184 | 70.7 | 14 |
| | 2 1 21. | .390561 | .744393 | 47.4 | 15 |
| | 1 13 13. | .236469 | .455599 | 129.4 | 16 |
| | 1 19 19. | .345884 | .066245 | 458.1 | 17 |
| | 2 19 18. | .321575 | .903115 | 17.6 | 18 |
| | 2 7 27. | .474842 | .017806 | 709.8 | 19 |
| | 2 12 32. | .549567 | .451894 | 143.1 | 20 |
| | 1 2 2. | .027027 | .495868 | 129.9 | 21 |
| | 2 20 40. | .677883 | .852752 | 30.2 | 22 |
| | 3 17 57. | .952061 | .642907 | 85.8 | 23 |
| | 2 17 16. | .283885 | .036756 | 657.4 | 24 |

```
                 1.       21.       41.
                 2.       22.       42.
                 3.       23.       43.
                 4.        3.       44.
                 5.       25.       45.
                 6.       26.       46.
                 7.       27.       47.
                 8.       23.       48.
                 9.       29.       49.
                10.       30.       50.
                11.       31.       51.
                12.       32.       46.
                13.       33.       28.
                14.       34.       54.
                15.       35.       55.
                16.       36.       34.
                17.       16.       54.
                18.       38.       58.
                15.       18.       37.
                20.       40.       60.
                       1  19    15.         .275596            .096375                    479.9                   25
AVG. TIME=       199.1                                         TOTAL OPERATING TIME= 4976.7   TOTAL FAILURES= 25
```

| -X- | FREQUENCY OF FAILURES WITH X FAILED SUBSYSTEMS |
|---|---|
| 1 | 0. |
| 2 | 0. |
| 3 | 0. |
| 4 | 0. |
| 5 | 0. |
| 6 | 0. |
| 7 | 0. |
| 8 | 0. |
| 9 | 0. |
| 10 | 0. |
| 11 | 0. |
| 12 | 0. |
| 13 | 0. |
| 14 | 0. |
| 15 | 0. |
| 16 | 0. |
| 17 | 0. |
| 18 | 0. |
| 19 | 0. |
| 20 | 0. |
| 21 | 0. |
| 22 | 0. |
| 23 | 0. |
| 24 | 1. |
| 25 | 2. |
| 26 | 1. |
| 27 | 0. |
| 28 | 1. |
| 29 | 0. |
| 30 | 0. |
| 31 | 0. |
| 32 | 0. |
| 33 | 0. |
| 34 | 0. |
| 35 | 0. |
| 36 | 0. |
| 37 | 0. |
| 38 | 0. |
| 39 | 0. |
| 40 | 0. |

| | |
|---|---|
| 41 | 0. |
| 42 | 0. |
| 43 | 0. |
| 44 | 0. |
| 45 | 0. |
| 46 | 0. |
| 47 | 0. |
| 48 | 0. |
| 49 | 0. |
| 50 | 0. |

| COMPUTATION | TIME | PROB. OF FAILURE |
|---|---|---|
| 1 | 1904.8 | .664029-03 |
| 2 | 2141.3 | .292749-02 |
| 3 | 2401.1 | .107953-01 |
| 4 | 2668.8 | .313202-01 |
| 5 | 2898.0 | .650688-01 |
| 6 | 3024.5 | .917853-01 |
| 7 | 3055.6 | .992891-01 |

AVG. NO. OF FAILURES= 25.60

10 PERCENT FAILED AT 3055.6

APPENDIX C

FORTRAN IV PROGRAM LISTING

```
@TIP   FOR  MAIN
COMPILATION BY UNIVAC 1107 FORTRAN-IV DATED     FEB. 10,1965 F4003
THIS COMPILATION WAS DONE ON 20 OCT 65 AT 16:10:59

   MAIN PROGRAM          ENTRY POINT 000000

   STORAGE USED (BLOCK, NAME, LENGTH)

         0001    *CODE    001742
         0000    *DATA    006170
         0002    *BLANK   002741


   EXTERNAL REFERENCES (BLOCK, NAME)

         0003    ALOCAT
         0004    DET
         0005    LIST
         0006    RANLIS
         0007    SEARCH
         0010    FAIL
         0011    SWITCH
         0012    SBACK
         0013    TIME
         0014    DEXP
         0015    COMFAC
         0016    NRDU$
         0017    NI01$
         0020    NI02$
         0021    NWDU$
         0022    NEXP9$
         0023    NSTOP$
         0024    NCDP$


   STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001    001703 104L      0001    000341 13L      0001    000060 134G      0001    000711 14L      0001
    0001    000776 17L       0001    001234 19L      0001    000565 2L        0001    001100 200L     0001
    0001    000760 22L       0001    000423 23L      0001    000363 262G      0001    000364 265G     0001
```

```
0001     000370 271G      0001     000514 28L       0001     000406 302G      0001     000407 305G      0001
0001     001223 306L      0001     001220 309L      0001     001227 310L      0001     000413 311G      0001
0001     000444 331G      0001     000473 335G      0001     000476 340G      0001     001343 35L       0001
0001     000625 375G      0001     001152 40L       0000     005567 400F      0000     005570 401F      0000
0000     005575 403F      0000     005600 404F      0000     005603 405F      0000     005612 406F      0000
0000     005631 408F      0000     005641 409F      0000     005643 413F      0000     005652 414F      0000
0001     000662 415G      0000     005702 416F      0000     005703 417F      0000     005711 418F      0000
0000     005737 420F      0000     005751 421F      0000     005762 422F      0000     005772 423F      0000
0000     006000 425F      0000     006003 427F      0000     006006 428F      0000     006011 429F      0000
0000     006026 431F      0000     006032 432F      0000     006035 433F      0001     000733 433G      0000
0000     006042 435F      0001     001127 44L       0000     006046 441F      0001     000741 441G      0000
0001     000742 444G      0000     006077 445F      0000     006110 446F      0001     000743 447G      0000
0000     006116 452F      0000     006120 454F      0000     006123 456F      0001     001275 46L       0001
0001     001423 502L      0001     001047 51L       0001     001471 53L       0001     000644 550L      0001
0001     000641 555L      0001     001200 563G      0001     001541 568L      0001     001543 569L      0001
0001     001374 572L      0001     001714 58L       0001     001661 583L      0001     000313 61L       0001
0001     001323 634G      0001     001324 637G      0001     001636 66L       0001     001401 666G      0001
0001     001555 68L       0001     001434 705G      0001     001447 717G      0001     000327 74L       0001
0001     000000 93L       0001     000601 94L       0001     001112 96L       0000 R 005545 AMEAN     0000 F
0000 R 005524 AVLAMD    0002 R 000000 B         0000 R 002734 BML       0000 R 005564 BOT       0000 F
0000 R 005525 CNT       0000 R 005526 CON       0000 D 005400 CONN      0014 R 000000 DEXP      0000 D
0000 R 005566 ERR       0000 D 005222 FACTL     0000 R 005535 FAILB     0000 R 005521 FAL       0000 F
0000 R 005546 FIWFT     0002 R 002734 FLAMD     0000 R 005544 FLONR     0000 R 005557 FM        0000 F
0000 I 005507 I         0000 I 005562 IDIFF     0000 I 005556 IL        0000 I 005555 IP        0000 I
0000 I 005471 IR        0000 I 005505 IRL       0000 I 005510 IRR       0000 I 005502 ISBAC     0000 I
0000 I 005500 ISW2      0000 I 005534 IW        0000 I 004230 IWF       0000 I 005532 IWFT      0000 I
0000 I 005552 I4        0000 I 005514 J         0000 I 005515 K         0000 I 005516 KK        0000 I
0000 I 005561 LL        0000 I 005527 LM        0000 I 005465 M         0000 I 005474 MBLL      0000 I
0000 I 005472 MM        0000 I 005530 MR        0000 I 005466 N         0000 I 005553 NA        0000 I
0000 I 005517 NI        0000 I 005520 NJ        0000 I 005540 NNN       0000 I 005550 NO        0000 I
0000 I 005504 NOP2      0000 I 005503 NOS       0000 I 005470 NR        0000 I 005473 NRL       0000 I
0000 I 005506 NTC       0000 I 005537 N1        0000 R 005531 OPRT      0000 R 005402 P         0000 F
0000 D 005370 PFAT      0000 D 005372 PFAT1     0000 D 005374 PTOT      0000 R 005536 RNT       0000 F
0000 R 005512 SUM       0000 R 003244 SUMT      0000 R 005511 SYSFA     0000 R 005214 T         0000 F
0000 R 005563 TOP       0000 R 003100 TT        0000 R 005541 T1        0000 R 005542 W         0000 F

00100      1.   CMAIN
00101      2.         DIMENSION C(5,20,15),BML(100),TT(100),
00101      3.        CSUMT(500),IWF(500),T(6)
00103      4.         DOUBLE PRECISION FACTL,PFAT,PFAT1,PTOT,EP,CONN
```

```
00104          DIMENSION B(5,20,15),FLAMD(5),P(51),FACTL(51)
00105          COMMON B, FLAMD
00106       93 READ(5,400) M,N,KTEMP,NR,IR,MM,NPL,MRLL,NSF,MIN,ISW1,ISW2,NOP1,
00106         CISBAC,NOS,NOP2,IRL,NTC
00132          READ(5,401)(FLAMD(I),I=1,M)
00140          WRITE (6,429)
00142          WRITE (6,430)
00144          WRITE (6,431)
00146          WRITE(6,404)(FLAMD(I),I=1,M)
00154          WRITE(6,402)M
00157          WRITE(6,403)N
00162          WRITE (6,419)KTEMP
00165          WRITE(6,405)NR
00170          WRITE(6,406)IP
00173          WRITE(6,407)MM
00176          WRITE(6,408)NRL
00201          WRITE(6,420)MIN
00204          WRITE(6,421)MRLL
00207          WRITE(6,422)NSF
00212          WRITE(6,424)ISW1
00215          WRITE(6,425)ISW2
00220          WRITE(6,427)NOP1
00223          WRITE(6,432) NOP2
00226          WRITE(6,428)ISBAC
00231          WRITE (6,423) NOS
00234          WRITE(6,454)NTC
00237          IRR=IR
00240          SYSFA=0.
00241          SUM=0.0
00242          CALL ALOCAT(M,N,KTEMP,FN,SUM,NSF)
00243          SUM=FN*SUM
00244          CALL DET    (M,N,KTEMP,NSF)
00245          IF(NSF-1) 60,61,60
00250       60 WRITE(6,433)
00252       61 IF(NRL-1)12,74,13
00255       12 CALL LIST(MM,M,N,KTEMP,NSF)
00256          GO TO 23
00257       74 CALL RANLIS(M,N,KTEMP,NOS,IRL,SYSFA,NSF)
00260          GO TO 23
00261       13 DO 24 J=1,N
00264          DO 24 I=1,M
00267       24 READ(5,409)(B(I,J,K),K=5,KTEMP)
00277          WRITE(6,450)
```

```
00301            DO 25 J=1,N
00304            DO 25 I=1,M
00307       25   WRITE(6,409)(B(I,J,K),K=5,KTEMP)
00317      23    IF(MBLL)14,14,16
00322      16    READ(5,409)(BML(KK),KK=1,MBLL)
00330            DO 94 KK=1, MBLL
00330     C      REMOVE MISSING BLOCKS FROM SPARE LISTS
00333            CALL SEARCH(BML(KK),0.,0,M,N,KTEMP)
00334            DO 27 I=1,M
00337            DO 27 J=1,N
00342            IF(B(I,J,3)-BML(KK))27,26,27
00345      26    NI=I
00346            NJ=J
00347            GO TO 28
00350      27    CONTINUE
00353      28    CALL FAIL (M,N,KTEMP,IR,MBLL,NJ,NI,FAL,BML(KK),MIN,X,S,SYSFA)
00354            IF(S)94,94,20
00357      20    IF(ISBAC)1,1,2
00362      1     CALL SWITCH(M,N,KTEMP,NJ,ISW1,ISW2,BML(KK),SYSFA,NOP1,
00362           1NOP2,NSF,MIN)
00363            GO TO 94
00364      2     CALL SBACK(M,N,KTEMP,NJ,BML(KK),SYSFA,NOP1,NOP2)
00365       94   CONTINUE
00367            AVLAMD=0.
00370            CNT=0.
00371            DO 550 I=1,M
00374            DO 550 J=1,N
00377            IF(B(I,J,4)-1.)551,555,555
00402      551   AVLAMD=AVLAMD-FLAMD(I)
00403            CNT=CNT + 1.
00404            GO TO 550
00405      555   SUM=SUM-FLAMD(I)
00406      550   CONTINUE
00411            CON=AVLAMD/CNT
00412            WRITE(6,434)
00414            DO 81 KK=1,MBLL
00417       81   WRITE(6,435) KK,BML(KK)
00424            WRITE(6,404)CON
00427            WRITE(6,404)SUM
00432      14    DO 82 LM=1,6
00435      82    T(LM)=0.
00435     C      SET UP DUPLICATE MATRIX C(I,J,K)
00437            MR=1
```

```
00440          DO 15 J=1,N
00443          DO 15 I=1,M
00446          DO 15 K=1,KTEMP
00451     15   C(I,J,K)=P(I,J,K)
00455          OPRT=0.
00456          IWFT=0
00457          DO 29 I3=0,50
00462     29   P(I3)=0.
00464     22   IW=0
00465          SUMT(MR)=0.
00466          IF(NSF-1)42,17,42
00471     42   WRITE(6,414)
00473          WRITE(6,415)
00475     17   CALL FAIL(M,N,KTEMP,IR,    0,NJ,NI,FAL,FAILB,MIN,X,S,SYSFA)
00476          IW=IW+1
00477          IF(NTC-1)51,54,51
00502     54   IF(MR-25)50,50,51
00505     50   CALL TIME(IR,FN,SUM,IW,RNT,TT(IW),SUMT(MR),NI)
00506     51   IF(SYSFA)97,97,96
00511     97   IF(S)96,96,98
00514     98   IF(ISBAC)100,100,200
00517    100   CALL SWITCH(M,N,KTEMP,NJ,ISW1,ISW2,FAL,SYSFA,NOP1,NOP2,NSF,MIN)
00520          GO TO 96
00521    200   CALL SBACK(M,N,KTEMP,NJ,FAL,SYSFA,NOP1,NOP2)
00522     96   IF(NSF-1) 91,40,91
00525     91   IF(IW-1)43,43,44
00530     43   WRITE(6,416)MR
00533     44   WRITE(6,417)  NI,NJ,FAILB,X,RNT,TT(IW),IW
00544     40   IF(SYSFA)17,17,52
00547     52   IF(NTC-1)19,59,19
00552     59   IF(MR-25)303,303,19
00555    303   IF(MR-1)304,304,305
00560    304   T(1)=SUMT(MR)
00561          GO TO 19
00561   C      LIST FIRST  5 OF 25 IN ORDER
00562    305   DO 306 N1=1,5
00565          NNN=6-N1
00566          IF(T(NNN ))306,306,307
00571    307   IF(SUMT(MR)-T(NNN ))309,309,308
00574    308   T(NNN+1)=SUMT(MR)
00575          GO TO 310
00576    309   T(NNN+1)=T(NNN )
00577    306   CONTINUE
```

```
00601          T(1)=SUMT(MR)
00602     310  IF(MR-25)19,312,19
00605     312  T1=T(5)
00605    C     COMPUTE INDIVIDUALRUN STATISTICS
00606      19  IWFT=IWFT+IW
00607          W=IW
00610          P(IW)=P(IW)+1.
00611          SYSFA=0.
00612          IWF(MR)=IW
00613          OPRT=OPRT+SUMT(MR)
00614          IF(NSF-1) 45,46,45
00617      45  AUGT=SUMT(MR)/W
00620          WRITE(6,418)(AUGT,SUMT(MR),IW)
00625      46  IF(NR-MR)35,35,37
00625    C     RESET SYSTEM
00630      37  DO 36 J=1,N
00633          DO 36 I=1,M
00636          DO 36 K=1,KTEMP
00641      36  B(I,J,K)=C(I,J,K)
00645          IRR=IRR+2
00646          IR=IRR
00647          MR=MR+1
00650          GO TO 22
00650    C     COMPUTE OVERALL STATISTICS
00651      35  FLONR=NR
00652          AMEAN=OPRT/25.
00653          FIWFT=IWFT
00654          FAVGF=FIWFT/FLONR
00655          NO=0
00656          IF(MBLL)570,570,571
00661     570  IPROD=M*N
00662          GO TO 572
00663     571  IPROD=CNT
00664     572  WRITE(6,444)((I4,P(I4)),I4=1,50)
00673          DO 500 NA=0,50
00676          IF(P(50-NA)) 500,500,502
00701     500  CONTINUE
00703     502  NB=50-NA
00704          DO 102 IP=1,NB
00707          P(IP)=(P(IP)/FLONR)+P(IP-1)
00710     102  CONTINUE
00712          IF(MBLL)552,552,553
00715     552  CON=0.
```

```
00716         DO 105 IL=1,N
00721   105   CON=CON+FLAMD(IL)
00723         FN=N
00724         CON=(-CON)/FN
00725   553   IF(NTC-1)202,201,202
00730   201   TI=0.7*TI
00731         GO TO 53
00732   202   TI=100
00733   53    PFAT=0.
00734         PTOT=0.
00735         CONN=CON*TI
00736         EP=DEXP(CONN)
00737         DO 56 LL=0,NB
00742         IDIFF=IPROD-LL
00743         IF(NO-1)67,68 ,68
00746   67    IF(MBLL)567,567,568
00751   567   TOP=M*N
00752         GO TO 569
00753   568   TOP=CNT
00754   569   BOT=LL
00754  C      COMPUTE COMBINATIONAL FACTORIAL
00755         CALL COMFAC(TOP,BOT,FACTL(LL))
00755  C      COMPUTE PROBABILITY OF FAILURE AT TI
00756   68    PFAT1=FACTL(LL)*((1.-EP)**LL)*(EP**IDIFF)
00757         PF=PFAT1
00760         PFAT1=P(LL)*PFAT1
00761         PTOT=PTOT+PF
00762         PFAT=PFAT+PFAT1
00763   56    CONTINUE
00765         PFAT=PFAT+1.-PTOT
00766         NO=NO+1
00767         IF(NO-1)65,65,66
00772   65    WRITE(6,445)
00774   66    WRITE(6,446)NO,TI,PFAT
01001         IF(NO-50)583,583,993
01004   993   WRITE(6,456)
01006         GO TO 93
01007   583   IF(NTC-1)104,103,104
01012   103   ERR=PFAT-.1
01013         IF(ABS(ERR)-.001)58,58,57
01016   57    TI=TI*(1.-(ERR/.8))
01017         GO TO 53
01020   104   IF(TI-5000.)105,58,58
```

```
01023       105  TI=TI+100.
01024            GO TO 53
01025       58   T1=TI
01026            WRITE(6,413)FAVGF
01031            IF(NTC-1) 93,203,93
01034       203  WRITE(6,441)T1
01037            GO TO 93
01040       400  FORMAT(I4)
01041       401  FORMAT(5E12.6)
01042       402  FORMAT(7H      N=,I4)
01043       403  FORMAT(7H      N=,I4)
01044       404  FORMAT(7H LAMDA=,E12.6)
01045       405  FORMAT(7H     NR=,I4,20H    (NUMBER OF RUNS))
01046       406  FORMAT(7H     IR=,I4,23H    (RANDOM NUMBER SET))
01047       407  FORMAT(7H     MM=,I4,28H    (COMMON SPARE LIST FLAG))
01050       408  FORMAT(7H    NRL=,I4,30H    (NON REPETITIVE LIST FLAG))
01051       409  FORMAT(6F12.0)
01051     C      409 FORMAT IS A SPARE LIST READ IN FORMAT
01052       413  FORMAT(28H        AVG. NO. OF FAILURES=,F7.2)
01053       414  FORMAT(21H1SEQUENCE OF FAILURES//)
01054       415  FORMAT(106H RUN NO.     FAILURE I.D.     FAILURE RAN. NO.     TIME
01054           CRAN. NO.     HOURS BETWEEN FAILURES     FAILURE NO.)
01055       416  FORMAT (I5)
01056       417  FORMAT(I15,I4,F6.0,F16.6,F19.6,F21.1,I21)
01057       418  FORMAT(11H AVG. TIME=,F8.1,55H                                  TO
01057           CTAL OPERATING TIME=,F7.1,18H   TOTAL FAILURES=,I3)
01060       419  FORMAT(7H KTEMP=,I4)
01061        420 FORMAT(7H    MIN=,I4,39H    (OPR. BLOCKS REQ. TO RESOLVE ERROR))
01062       421  FORMAT (7H   MBLL=,I4,31H    (MISSING BLOCK LIST LENGTH))
01063       422  FORMAT(7H    NSF=,I4,27H    (NO SAMPLE FORMAT FLAG))
01064       423  FORMAT (7H    NOS=,I4)
01065       424  FORMAT (7H   ISW1=,I4)
01066       425  FORMAT (7H   ISW2=,I4)
01067       427  FORMAT (7H   NOP1=,I4)
01070       428  FORMAT (7H ISBAC=,I4)
01071       429  FORMAT(39H1FAILURE RESPONSIVE SYSTEM ORGANIZATION/)
01072       430  FORMAT(21H STRATEGY DESCRIPTION //)
01073       431  FORMAT(18H CONTROL CONSTANTS/)
01074       432  FORMAT(7H   NOP2=,I4)
01075       433  FORMAT(1H1)
01076       434  FORMAT(15H1MISSING BLOCKS//)
01077       435  FORMAT(5H BVL(,I2,3H )=,F6.0)
01100       441  FORMAT(21H 10 PERCENT FAILED AT,F8.1)
```

```
01101   263.    444  FORMAT(10H]        -X-,11X,27H FREQUENCY OF FAILURES WITH/44H
01101   264.       C                     X FAILED SUBSYSTEMS//(I9,F28.0))
01102   265.    445  FORMAT(44H1COMPUTATION     TIME          PROB. OF FAILURE/)
01103   266.    446  FORMAT( I7,F15.1,D20.6)
01104   267.    450  FORMAT(11H SPARE LIST)
01105   268.    452  FORMAT(D30.6)
01106   269.    454  FORMAT(7H    NTC=,I4)
01107   270.    456  FORMAT(11H ERROR EXIT)
01110   271.         STOP
01111   272.         END

       END OF LISTING.      0 *DIAGNOSTIC* MESSAGE(S).
PHASE 1 TIME =   3 SEC.
PHASE 2 TIME =   1 SEC.
PHASE 3 TIME =   3 SEC.
PHASE 4 TIME =   1 SEC.
PHASE 5 TIME =   2 SEC.
PHASE 6 TIME =   4 SEC.

TOTAL COMPILATION TIME =  14  SEC
```

```
WTIP   FOR  ALOCAT
COMPILATION BY UNIVAC 1107 FORTRAN-IV DATED     FEB. 10,1965 F4003
THIS COMPILATION WAS DONE ON 19 OCT 65 AT 17:06:37

   SUBROUTINE ALOCAT     ENTRY POINT 000170

   STORAGE USED (BLOCK, NAME, LENGTH)

        0001    *CODE    000221
        0000    *DATA    000055
        0002    *BLANK   002741

   EXTERNAL REFERENCES (BLOCK, NAME)

        0003    NWDU$
        0004    NI02$
        0005    NI01$

   STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0000    000010 103F      0000   000017 104F     0001    000031 107G     0001    000042 115G     0001
    0001    000066 132G      0001   000067 135G     0001    000071 140G     0001    000103 147G     0001
    0001    000122 165G      0001   000125 170G     0001    000130 173G     0001    000155 8L       0002 R
    0002 R  002734 FLAMD     0000 I 000005 I        0000 I  000007 J        0000 I  000006 K        0000 R

00100     1.   C
00101     2.          SUBROUTINE ALOCAT(M,N,KTEMP,FN,SUM,NSF)
00103     3.          DIMENSION RANGE(5)
00104     4.          DIMENSION B(5,20,15),FLAMD(5)
00105     5.          COMMON B, FLAMD
00106     6.          DO 1 I=1,M
00111     7.   1      SUM=SUM + FLAND(I)
00113     8.          FN=N
00114     9.          DO 2 I=1,M
00117    10.   2      RANGE(I)=FLAMD(I)/(FN*SUM)
```

```
00121     11.          B(1,1,1)=0.
00122     12.          B(1,1,2)=RANGE(1)
00123     13.          DO 3 I=2,N
00126     14.          B(I,1,1)=(F*RANGE(I-1))+B(I-1,1,1)
00127     15.     3    B(I,1,2)=RANGE(I)+B(I,1,1)
00131     16.          DO 4 K=1,2
00134     17.          DO 4 J=2,N
00137     18.          DO 4 I=1,N
00142     19.     4    B(I,J,K)=B(I,J-1,K)+RANGE(I)
00146     20.          DO 5 I=1,N
00151     21.          DO 5 J=1,N
00154     22.     5    B(I,J,4)=0.
00157     23.          IF(USE-1)6,8,6
00162     24.     6    WRITE(6,104)
00164     25.          DO 7 I=1,N
00167     26.          DO 7 J=1,N
00172     27.          DO 7 K=1,2
00175     28.     7    WRITE(6,103)I,J,K,B(I,J,K)
00206     29.     8    RETURN
00207     30.   103    FORMAT(3H B(,I2,2H ,,I2,2H ,,I2,3H )=,F12.6)
00210     31.   104    FORMAT(12H1ALLOCATIONS)
00211     32.          END

      END OF LISTING.       0 *DIAGNOSTIC* MESSAGE(S).
PHASE 1 TIME =   0 SEC.
PHASE 2 TIME =   0 SEC.
PHASE 3 TIME =   1 SEC.
PHASE 4 TIME =   1 SEC.
PHASE 5 TIME =   1 SEC.
PHASE 6 TIME =   0 SEC.

TOTAL COMPILATION TIME =   3  SEC
```

```
@TIP   FOR  COMFAC
COMPILATION BY UNIVAC 1107 FORTRAN-IV DATED     FEB. 10,1965 F4003
THIS COMPILATION WAS DONE ON 19 OCT 65 AT 17:06:41

   SUBROUTINE COMFAC    ENTRY POINT 000067

   STORAGE USED (BLOCK, NAME, LENGTH)

         0001   *CODE   000107
         0000   *DATA   000017
         0002   *BLANK  000000

   EXTERNAL REFERENCES (BLOCK, NAME)

         0003   NCDPS

   STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001   000046 123G       0001   000025 2L        0001   000032 3L        0001   000010 6L        0000 F
    0000 R 000001 DUM        0000 I 000003 JI        0000 I 000000 NDUM      0000 I 000002 NDUM1

00100      1.   C
00101      2.         SUBROUTINE COMFAC(TOP,BCT,FACTL)
00103      3.         DOUBLE PRECISION FACTL
00104      4.         IF(BOT)5,5,6
00107      5.    5    FACTL=1.
00110      6.         RETURN
00111      7.    6    IF(TOP-2.*BOT)1,1,2
00114      8.    1    NDUM=TOP-BOT
00115      9.         GO TO 3
00116     10.    2    NDUM=BOT
00117     11.    3    DUM=NDUM
00120     12.         FACTL=1.
00121     13.         NDUM1=NDUM-1
00122     14.         DO 4 JI=0,NDUM1
```

```
00125     15.          AJI=JI
00126     16.     4    FACTL=(FACTL*(TCP-AJI))/(F[illegible]M-AJI)
00130     17.          RETURN
00131     18.          END

         END OF LISTING.      0 *DIAGNOSTIC* MESSAGE(S).
PHASE 1 TIME =   0 SEC.
PHASE 2 TIME =   1 SEC.
PHASE 3 TIME =   0 SEC.
PHASE 4 TIME =   0 SEC.
PHASE 5 TIME =   1 SEC.
PHASE 6 TIME =   0 SEC.

TOTAL COMPILATION TIME =   2  SEC
```

```
WTIP    FOR  DET
COMPILATION BY UNIVAC 1107 FORTRAN-IV DATED      FEB. 10,1965 F4003
THIS COMPILATION WAS DONE ON 19 OCT 65 AT 17:05:44

   SUBROUTINE DET        ENTRY POINT 000071

   STORAGE USED (BLOCK, NAME, LENGTH)

         0001   *CODE   000106
         0000   *DATA   000031
         0002   *BLANK  002741


   EXTERNAL REFERENCES (BLOCK, NAME)

         0003   NWDU$
         0004   NI02$
         0005   NI01$


   STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001    000054 100L        0001    000024 113G        0001    000027 116G        0000    000002 890F        0001
    0000    000005 893F        0002 R  000000 B           0002    002734 FLAMD       0000 I  000000 I           0000 I


00100      1.   C
00101      2.          SUBROUTINE DET    (M,N,KTEMP,NSF)
00101      3.   C      NUMBERS STEP ACROSS ROWS FIRST
00103      4.          DIMENSION B(5,20,15),FLAMD(5)
00104      5.          COMMON B, FLAMD
00105      6.          IF(NSF-1) 892,891,392
00110      7.    892   WRITE(6,890)
00112      8.    891   DO 100 I=1,M
00115      9.          DO 100 J=1,N
00120     10.          B(I,J,3)=(I-1)*N+J
00121     11.          IF(NSF-1)101,100,101
00124     12.    101   WRITE(6,893) I,J,B(I,J,3)
```

```
00131     13.    190  CONTINUE
00134     14.         RETURN
00135     15.    690  FORMAT(1H1ID NUMBERS)
00136     16.    693  FORMAT(3H B(,I2,2 ,,I2,5H ,3)=,F5.0)
00137     17.         END

          END OF LISTING.      0 *DIAGNOSTIC* MESSAGE(S).
PHASE 1 TIME =    0 SEC.
PHASE 2 TIME =    0 SEC.
PHASE 3 TIME =    1 SEC.
PHASE 4 TIME =    0 SEC.
PHASE 5 TIME =    1 SEC.
PHASE 6 TIME =    0 SEC.

TOTAL COMPILATION TIME =   2  SEC
```

```
WTIP   FOR  FAIL
COMPILATION BY UNIVAC 1107 FORTRAN-IV DATED      FEB. 10,1965 F4003
THIS COMPILATION WAS DONE ON 19 OCT 65 AT 17:06:47


   SUBROUTINE FAIL       ENTRY POINT 000154

   STORAGE USED (BLOCK, NAME, LENGTH)

          0001    *CODE    000225
          0000    *DATA    000027
          0002    *BLANK   002741


   EXTERNAL REFERENCES (BLOCK, NAME)

          0003    UDRNRT


   STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

     0001     000025 113G       0001     000030 116G       0001     000075 12L       0001     000106 14L       0001
     0001     000116 162G       0001     000046 5L         0001     000107 51L       0001     000015 6L        0001
     0001     000135 8L         0001     000137 9L         0002 R 000000 B         0000 R 000004 FFAL      0002
     0000 I 000000 I           0000 I 000003 I1          0000 I 000001 J         0000 I 000005 L         0000 I
     0003 R 000000 UDRNRT


00100       1.    C
00101       2.             SUBROUTINE FAIL(M,N,KTEMP,IR,MBLL,NJ,NI,FAL,FAILB,MIN,X,S,SYSFA)
00103       3.             DIMENSION B(5,20,15),FLAMD(5)
00104       4.             COMMON B, FLAMD
00105       5.             IF(MBLL)6,6,7
00110       6.      6      X=UDRNRT(IR)
00111       7.             FAILB=0.
00112       8.             DO 5 I=1,[illegible]
00115       9.             DO 5 J=1,[illegible]
0012[illegible]     10.             IF(X-B(I,J,1))5,2,2
0012[illegible]     11.      2      IF(X-B(I,J,2))5,3,5
```

```
00126      3   FAILB=B(I,J,3)
00127          NI=I
00130          NJ=J
00131          GO TO 7
00132      6   CONTINUE
00135          GO TO 6
00136      7   B(NI,NJ,1)=0.
00137          B(NI,NJ,2)=0.
00137   C      TEST TO SEE IF REPAIR IS NECESSARY
00140     50   MZ=0
00141          DO 12 II=1,M
00144          IF(B(II,NJ,2))11,12,11
00144   C      MZ=NO. OF OPERATING SUBSYSTEMS
00147     11   MZ=MZ + 1
00150     12   CONTINUE
00152          IF(MZ-MIN)13,14,14
00155     13   S=1.
00156          GO TO 51
00157     14   S=0.
00160     51   FFAL=0.
00161          DO 4 L=1,M
00164      4   FFAL=AMAX1(FFAL,B(L,NJ,4))
00166          FAL=FFAL
00167          IF(MZ-1)8,15,9
00172     15   IF(MIN-3)9,8,9
00175      8   SYSFA=1.
00176      9   B(NI,NJ,4)=FAL+1.
00177          RETURN
00200          END

        END OF LISTING.     0 *DIAGNOSTIC* MESSAGE(S).
PHASE 1 TIME =   0 SEC.
PHASE 2 TIME =   1 SEC.
PHASE 3 TIME =   0 SEC.
PHASE 4 TIME =   1 SEC.
PHASE 5 TIME =   0 SEC.
PHASE 6 TIME =   1 SEC.

TOTAL COMPILATION TIME =   3  SEC
```

```
WTIP   FOR  LIST
COMPILATION BY UNIVAC 1107 FORTRAN-IV DATED      FEB. 10,1965 F4003
THIS COMPILATION WAS DONE ON 19 OCT 65 AT 17:06:51


   SUBROUTINE LIST       ENTRY POINT 000352

   STORAGE USED (BLOCK, NAME, LENGTH)

         0001    *CODE    000402
         0000    *DATA    000110
         0002    *BLANK   002741


   EXTERNAL REFERENCES (BLOCK, NAME)

         0003    NRDU$
         0004    NI01$
         0005    NI02$
         0006    NWDU$


   STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0000    000012 102F       0000    000014 103F       0000    000023 104F       0000    000027 105F       0000
    0001    000247 11L        0001    000057 112G       0001    000075 122G       0001    000105 127G       0001
    0001    000106 132G       0001    000300 14L        0001    000126 143G       0001    000132 147G       0001
    0001    000155 163G       0001    000160 166G       0001    000115 2L         0001    000202 202G       0001
    0001    000210 31L        0001    000242 5L         0001    000320 6L         0001    000200 9L         0002 R
    0000 R  000007 BTEMP      0002    002730 FLAMD      0000 R  000003 FN         0000 I  000001 I          0000 I
    0000 I  000011 IBTEMP     0000 I  000004 IROW       0000 I  000006 ISPARE     0000 I  000010 ITEMP      0000 I
    0000 I  000000 K


00100      1.    C
00101      2.           SUBROUTINE LIST(MM,M,N,XTEMP,NSF)
00103      3.           DIMENSION B(5,20,15),FLAMD(5)
00104      4.           COMMON B, FLAMD
00105      5.           IF(MM-1)2,1,2
```

```
00111        1    READ(5,102)( (I,J,K),K=5,KTEMP)
00115             WRITE(6,102)
00120             WRITE(6,102)((I,J,K),K=5, TEMP)
00126             DO 7 K=5,KTEMP
00131             DO 7 I=2,
00134        7    B(I,J,K)=B(1,J,K)
00137             GO TO 3
00140        2    WRITE(6,103)
00142             DO 20 I=1,
00145             READ(5,102)(B(I,J,K),K=5,KTEMP)
00153       20    WRITE(6,102)(B(I,J,K),K=5,KTEMP)
00162        3    DO 6 I=1,
00165             DO 6 J=1,
00170             IF(ISF-1)8,9,8
00173        8    WRITE(6,105)
00175             WRITE(6,103)I,J
00201        9    DO 6 K=5,KTEMP
00204             IF(B(I,J,K))30,30,31
00207       30    B(I,J,K)=0.
00210             GO TO 14
00211       31    F=
00212             IROW=(B(I,J,K)/F )+.999
00213             IB=B(I,J, )
00214             ISPACE=IB+J-
00215             IF(ISPACE-1)5,4,4
00220        4    BTEMP=ISPACE
00221             GO TO 11
00222        5    BTEMP=ISPACE+
00223       11    ITEMP=(BTEMP/F )+.9999
00224             IF(IROW-ITEMP) 12,13,12
00227       12    BTEMP=BTEMP
00230             B(I,J,K)=((IROW-ITEMP)*  )+BTEMP
00231             GO TO 1
00232       13    B(I,J,K )=BTEMP
00233       14    IB=B(I,J,K )
00234             IF(ISF-1)10,6,10
00237       10    WRITE(6,104)IB,K
00243        6    CONTINUE
00247             RETURN
00250      102    FORMAT(6F12.0)
00251      103    FORMAT(1H ,30X,3  (,I2,2   ,,I2,2  ),10X,2  K)
00252      104    FORMAT(1H ,32X,I4,15X,I4)
00253      105    FORMAT(1H ,25X,3  ------------------------------)
```

```
00254     49.      100 FORMAT(11H SPARE LIST)
00255     50.          END

        END OF LISTING.      0 *DIAGNOSTIC* MESSAGE(S).
PHASE 1 TIME =   1 SEC.
PHASE 2 TIME =   0 SEC.
PHASE 3 TIME =   1 SEC.
PHASE 4 TIME =   0 SEC.
PHASE 5 TIME =   1 SEC.
PHASE 6 TIME =   1 SEC.

TOTAL COMPILATION TIME =   4  SEC
```

```
WTIP   FOR  RANLIS
COMPILATION BY UNIVAC 1107 FORTRAN-IV DATED      FEB. 10,1965 F4003
THIS COMPILATION WAS DONE ON 19 OCT 65 AT 17:06:56

   SUBROUTINE RANLIS     ENTRY POINT 000224

   STORAGE USED (BLOCK, NAME, LENGTH)

        0001    *CODE    000255
        0000    *DATA    000071
        0002    *BLANK   002741

   EXTERNAL REFERENCES (BLOCK, NAME)

        0003    UDRNRT
        0004    NWDU$
        0005    NI02$
        0006    NI01$

   STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0000     000010 103F        0000     000017 104F        0000     000023 105F        0001     000026 106G        0001
    0001     000076 122G        0001     000122 137G        0001     000124 142G        0001     000125 145G        0001
    0001     000146 162G        0001     000165 173G        0001     000052 701L        0001     000207 705L        0001
    0001     000111 711L        0002 R 000000 B            0002     002734 FLAMD       0000 R 000003 FM          0000 R
    0000 I 000007 I            0000 I 000005 I1           0000 I 000004 IX          0000 I 000000 J           0000 I
    0000 I 000006 LB           0003 R 000000 UDRNRT

00100      1.    C
00101      2.          SUBROUTINE RANLIS(M,N,KTEMP,NOS,IRL,SYSFA,NSF)
00103      3.          DIMENSION R(5,20,15),FLAMD(5)
00104      4.          COMMON B,FLAMD
00104      5.    C     ROUTINE FOR GENERATING RANDOM SPARE LIST
00105      6.          DO 704 J=1,N
00110      7.          DO 704 K=5,KTEMP
```

```
00113     8.          FN=N
00114     9.          FM=M
00115    10.     701  IX=FN*FM*UPRNRT(IRL)  +1.
00116    11.          IF(NOS-1)711,712,705
00121    12.     712  DO 707 II=1,M
00124    13.          LB=B(II,J,3)
00125    14.          IF(IX-LB)707,701,707
00130    15.     707  CONTINUE
00132    16.     711  B(1,J,K)=IX
00133    17.     704  CONTINUE
00136    18.          DO 713 I=2,M
00141    19.          DO 713 J=1,N
00144    20.          DO 713 K=5,KTEMP
00147    21.     713  B(I,J,K)=B(1,J,K)
00153    22.          IF(NSF-1)709,706,709
00156    23.     709  DO 710 I=1,M
00161    24.          DO 710 J=1,N
00164    25.          WRITE (6,105)
00166    26.          WRITE (6,103)I,J
00172    27.          DO 710 K=5,KTEMP
00175    28.     710  WRITE(6,104)B(I,J,K),K
00204    29.          GO TO 706
00205    30.     705  SYSFA=1.
00206    31.     706  RETURN
00207    32.     103  FORMAT(1H ,30X,3H B(,I2,2H ,,I2,2H ),10X,2H K)
00210    33.     104  FORMAT(1H ,32X,F4.0,15X,I2)
00211    34.     105  FORMAT(1H ,25X,30H ------------------------------)
00212    35.          END

        END OF LISTING.      0 *DIAGNOSTIC* MESSAGE(S).
PHASE 1 TIME =   1 SEC.
PHASE 2 TIME =   0 SEC.
PHASE 3 TIME =   1 SEC.
PHASE 4 TIME =   0 SEC.
PHASE 5 TIME =   1 SEC.
PHASE 6 TIME =   0 SEC.

TOTAL COMPILATION TIME =   3  SEC
```

C-24

```
@TIP    FOR   SPACK
COMPILATION BY UNIVAC 1107 FORTRAN-IV DATED      FEB. 10,1965 F4P03
THIS COMPILATION WAS DONE ON 19 OCT 65 AT 17:07:09

   SUBROUTINE SPACK      ENTRY POINT 000170

   STORAGE USED (BLOCK, NAME, LENGTH)

         0001    *CODE    000230
         0000    *DATA    000055
         0002    *BLANK   002741


   EXTERNAL REFERENCES (BLOCK, NAME)

         0003    NWDU$
         0004    NIO2$


   STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001    000020 106G        0001    000041 117G        0001    000051 126G        0001    000060 131G        0001
    0001    000036 902L        0001    000125 943L        0001    000111 940L        0001    000120 950L        0001
    0001    000140 952L        0000    000010 953F        0001    000135 956L        0001    000034 982L        0001
    0002 R  000000 B           0000 R  000003 FIO         0002    002734 FLAMD       0000 I  000000 I           0000 I
    0000 I  000004 I1          0000 I  000007 I2          0000 I  000005 J1          0000 I  000002 K           0000 I


00100     1.   C
00101     2.          SUBROUTINE SBACK(M,N,KTEMP,NJ,FAL,SYSFA,NOP1,NOP2)
00103     3.          DIMENSION B(5,20,15),FLAMD(5)
00104     4.          COMMON B,FLAMD
00104     5.   C      PROGRAM TO SPRING BACK SPARE
00105     6.    901   DO 956 I=1,M
00105     7.   C      LOOK FOR PREVIOUS FAILURE
00110     8.          IF(B(I,NJ,4)-FAL)956,981,956
00110     9.   C      BEGIN SCANNING THE SPARE LIST TO BE USED
00113    10.    981   INHIB=NOP1
```

```
00114     11.            GO TO 982
00115     12.      982   INHIB=NOP2
00116     13.      902   DO 903 K=5,KTEMP
00121     14.            IF(B(I,NJ,K))952,903,905
00124     15.      905   FID=B(I,NJ,K)
00124     16.     C      SEARCH FOR THE SPARE WITH ID NUMBER=FID
00125     17.            DO 950 I1=1,M
00130     18.            DO 950 J1=1,N
00133     19.            IF(B(I1,J1,3)-FID)950,908,950
00133     20.     C      IF THE SPARE IS UNAVAILABLE,OR HAS FAILED,KEEP LOOKING
00136     21.      908   IF(B(I1,J1,4))952,930,903
00141     22.      930   KK=0
00142     23.            DO 940 I2=1,M
00145     24.            IF(B(I2,J1,4)-1.)931,940,940
00150     25.      931   KK=KK+1
00151     26.      940   CONTINUE
00153     27.            IF(KK-INHIB)903,903,999
00156     28.      950   CONTINUE
00161     29.      903   CONTINUE
00163     30.      932   IF(INHIB-NOP2)951,951,982
00166     31.      956   CONTINUE
00170     32.      952   WRITE(6,953)
00172     33.            GO TO 999
00173     34.      951   SYSFA=1.
00174     35.            GO TO 999
00175     36.      999   RETURN
00176     37.      953   FORMAT(14H ERROR IN FAIL)
00177     38.            END

        END OF LISTING.      0 *DIAGNOSTIC* MESSAGE(S).
PHASE 1 TIME =   1 SEC.
PHASE 2 TIME =   0 SEC.
PHASE 3 TIME =   1 SEC.
PHASE 4 TIME =   0 SEC.
PHASE 5 TIME =   1 SEC.
PHASE 6 TIME =   0 SEC.

TOTAL COMPILATION TIME =   3  SEC
```

```
@TIP   FOR  SEARCH
COMPILATION BY UNIVAC 1107 FORTRAN-IV DATED      FEB. 10,1965 F4003
THIS COMPILATION WAS DONE ON 19 OCT 65 AT 17:07:04

   SUBROUTINE SEARCH     ENTRY POINT 000053

   STORAGE USED (BLOCK, NAME, LENGTH)

         0001    *CODE    000074
         0000    *DATA    000024
         0002    *BLANK   002741

   STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001    000022 106G      0001    000023 111G      0001    000024 114G      0001    000037 3L      0001
    0002 R  000000 B         0002    002734 FLAMD     0000 I  000000 I         0000 I  000001 J       0000 I

00100      1.    C
00101      2.           SUBROUTINE SEARCH(FAILR,SPARE,NREP,M,N,KTEMP)
00103      3.           DIMENSION B(5,20,15),FLAMD(5)
00104      4.           COMMON B,FLAMD
00105      5.           DO 4 I=1,M
00110      6.           DO 4 J=1,N
00113      7.           DO 4 K=5,KTEMP
00116      8.           IF(B(I,J,K)-FAILR)4,1,4
00121      9.     1     IF(NREP-1)3,3,2
00124     10.     2     B(I,J,K)=SPARE
00125     11.           GO TO 4
00126     12.     3     B(I,J,K)=0.
00127     13.     4     CONTINUE
00133     14.           RETURN
00134     15.           END
       END OF LISTING.      0 *DIAGNOSTIC* MESSAGE(S).
PHASE 1 TIME =   1 SEC.
PHASE 2 TIME =   0 SEC.
PHASE 3 TIME =   0 SEC.
PHASE 4 TIME =   1 SEC.
PHASE 5 TIME =   0 SEC.
PHASE 6 TIME =   1 SEC.

TOTAL COMPILATION TIME =   3  SEC
```

```
MTIP   FOR  SWITCH
COMPILATION BY UNIVAC 1107 FORTRAN-IV DATED      FEB. 10,1965 F4003
THIS COMPILATION WAS DONE ON 19 OCT 65 AT 17:07:07

   SUBROUTINE SWITCH     ENTRY POINT 000470

   STORAGE USED (BLOCK, NAME, LENGTH)

         0001    *CODE    000565
         0000    *DATA    000100
         0002    *BLANK   002741


   EXTERNAL REFERENCES (BLOCK, NAME)

         0003    SEARCH
         0004    NWDU$
         0005    NIO2$
         0006    NIO1$


   STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001     000020 106G         0001     000044 126G         0001     000054 130G         0001     000063 133G         0001
    0001     000257 222G         0001     000302 230G         0001     000331 236G         0001     000340 243G         0001
    0001     000371 256G         0001     000431 276G         0001     000435 302G         0001     000134 503L         0001
    0001     000207 513L         0001     000236 514L         0001     000263 516L         0001     000306 518L         0001
    0001     000343 521L         0000     000026 533F         0001     000114 540L         0001     000127 550L         0001
    0001     000444 552L         0000     000020 553F         0000     000024 554F         0001     000144 556L         0000
    0001     000041 580L         0001     000037 582L         0002 R 000000 B             0000 R 000015 FAILR         0000 R
    0000 R 000005 FID           0002     002734 FLAMR        0000 I 000000 I             0000 I 000003 INHIB         0000 I
    0000 I 000012 ISPAR         0000 I 000006 I1            0000 I 000011 I2            0000 I 000017 J             0000 I
    0000 I 000013 JSPAR         0000 I 000007 J1            0000 I 000004 K             0000 I 000010 KK            0000 R


00100      1.    C
00101      2.          SUBROUTINE SWITCH(M,N,KTEMP,NJ,ISW1,ISW2,FAL,SYSFA,NOP1,NOP2,
00101      3.         CNSF,MIN)
```

```
00103              DIMENSION B(5,20,15),FLAMD(5)
00104              COMMON B, FLAMD
00104       C      LOOK FOR PREVIOUS FAILURE
00105         591  DO 556 I=1,M
00110              IF(B(I,NJ,4)-FAL) 556,502,556
00113         502  IRPL=I
00114              JRPL=NJ
00114       C      BEGIN SCANNING THE SPARE LIST TO BE USED
00115         581  INHIB=NOP1
00116              GO TO 580
00117         582  INHIB=NOP2
00120         580  DO 503 K=5,KTEMP
00123         504  IF(B(I,NJ,K))552,503,505
00126         505  FID=B(I,NJ,K)
00126       C      SEARCH FOR THE SPARE WITH ID NUMBER=FID
00127              DO 550 I1=1,M
00132              DO 550 J1=1,N
00135              IF(B(I1,J1,3)-FID)550,508,550
00140         508  IF(B(I1,J1,4))552,530,503
00140       C      IF THE SPARE HAS FAILED OR IS UNAVAILABLE, KEEP LOOKING
00143         530  KK=0
00144              DO 540 I2=1,M
00147              IF(B(I2,J1,4)-1.)531,540,540
00152         531  KK=KK+1
00153         540  CONTINUE
00155              IF(KK-INHIB)503,503,509
00160         509  ISPAR=I1
00161              JSPAR=J1
00162              SPARE=B(I1,J1,3)
00163              GO TO 510
00164         550  CONTINUE
00167         503  CONTINUE
00171              IF(INHIB-NOP2)551,551,582
00174         556  CONTINUE
00176              GO TO 551
00177         510  IF(ISW2-1)511,511,513
00202         511  FAILB=B(IRPL,JRPL,3)
00203              CALL SEARCH(FAILB,SPARE,ISW2,M,N,KTEMP)
00204              IF(ISW2)552,514,512
00207         512  FAILB=SPARE
00210              CALL SEARCH(FAILB,SPARE,ISW2,M,N,KTEMP)
00211              GO TO 514
00212         513  FAILB=SPARE
```

```
00213            CALL SEARCH(FAILR,SPARE,1,[illegible],N,KTEMP)
00214            FAILR=B(IRPL,JRPL,3)
00215            CALL SEARCH(FAILR,SPARE,IS[illegible]2,[illegible],N,KTEMP)
00216     514    IF(ISW1-1)515,516,518
00221     515    DO 570 K=1,4
00224     570    B(IRPL,JRPL,K)=B(ISPAR,JSPAR,K)
00226            GO TO 521
00227     516    DO 517 K=1,KTEMP
00232     517    B(IRPL,JRPL,K)=B(ISPAR,JSPAR,K)
00234            GO TO 521
00235     518    DO 519 K=5,KTEMP
00240     519    B(IRPL,JRPL,K)=0.
00242            DO 560 K=1,4
00245     560    B(IRPL,JRPL,K)=B(ISPAR,JSPAR,K)
00247     521    DO 522 K=1,2
00252     522    B(ISPAR,JSPAR,K)=0.
00254     590    FFNUM=0.
00255            DO 523 I=1,M
00260      523   FFNUM=AMAX1(FFNUM,B(I,JSPAR,4))
00262            B(ISPAR,JSPAR,4)=FFNUM+1.
00263            GO TO 520
00264     551    IF(MIN-3)600,520,600
00267     600    SYSFA=1.
00270            IF(NSF-1)534,520,534
00273      534   WRITE(6,533)
00275            DO 585 J=1,N
00300      585   WRITE(6,554) (B(I,J,3),I=1,M)
00307            GO TO 520
00310     552    WRITE(6,553)
00312     553    FORMAT(14H ERROR IN FAIL)
00313      554   FORMAT(5F10.0)
00314      533   FORMAT(1H1)
00315      561   FORMAT(5H FAL=,F6.2)
00316     520    RETURN
00317            END

       END OF LISTING.      0 *DIAGNOSTIC* MESSAGE(S).
PHASE 1 TIME =   1 SEC.
PHASE 2 TIME =   1 SEC.
PHASE 3 TIME =   1 SEC.
PHASE 4 TIME =   1 SEC.
PHASE 5 TIME =   1 SEC.
PHASE 6 TIME =   1 SEC.

TOTAL COMPILATION TIME =   6  SEC
```

```
TI   FOR  TIME
COMPILATION BY UNIVAC 1107 FORTRAN-IV MATEL      FEB. 10,1965 F1003
THIS COMPILATION WAS DONE ON 19 OCT 65 AT 17:07:14

   SUBROUTINE TIME       ENTRY POINT 000031

   STORAGE USED (BLOCK, NAME, LENGTH)

         0001    *CODE    000054
         0000    *DATA    000007
         0002    *BLANK   002741


   EXTERNAL REFERENCES (BLOCK, NAME)

         0003    UDRNRT
         0004    ALOG


   STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001   000014 806L       0001   000017 807L       0004 R 000000 ALOG       0002   000000 B          0002 R
    0003 R 000000 UDRNRT     0000 R 000000 Y


00100       1.    C
00101       2.          SUBROUTINE TIME (IR,FN,SUM,IW,RNT,T,SUMT,NI)
00103       3.          DIMENSION B(5,20,15),FLAMD(5)
00104       4.          COMMON B,FLAMD
00105       5.          RNT=UDRNRT(IR)
00106       6.          IF(IW-1) 805,805,806
00111       7.     805  Y=SUM
00112       8.          GO TO 807
00113       9.      806 Y=Y-FLAMD(NI)
00114      10.     807  T=ALOG(RNT)/(-Y)
00115      11.          SUMT=SUMT+T
00116      12.          RETURN
00117      13.          END
```

```
     END OF LISTING.      0 *DIAGNOSTIC* MESSAGE(S).
PHASE  1 TIME =    0 SEC.
PHASE  2 TIME =    0 SEC.
PHASE  3 TIME =    1 SEC.
PHASE  4 TIME =    0 SEC.
PHASE  5 TIME =    1 SEC.
PHASE  6 TIME =    0 SEC.

TOTAL COMPILATION TIME =    2  SEC
```