
Retrospective Theses and Dissertations

1982

Simulation of Naval Training Pipelines

Roger H. Werner
University of Central Florida

 Part of the [Engineering Commons](#)

Find similar works at: <https://stars.library.ucf.edu/rtd>

University of Central Florida Libraries <http://library.ucf.edu>

This Masters Thesis (Open Access) is brought to you for free and open access by STARS. It has been accepted for inclusion in Retrospective Theses and Dissertations by an authorized administrator of STARS. For more information, please contact STARS@ucf.edu.

STARS Citation

Werner, Roger H., "Simulation of Naval Training Pipelines" (1982). *Retrospective Theses and Dissertations*. 663.

<https://stars.library.ucf.edu/rtd/663>

SIMULATION OF NAVAL TRAINING PIPELINES

BY

ROGER H. WERNER

B.S., Tennessee Technological University, 1975

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science
in the Graduate Studies Program of the College of Engineering
University of Central Florida
Orlando, Florida

Summer Term
1982

ABSTRACT

Flow of enlisted Navy men through basic training, BEE school and "A" school was simulated using SLAM. Queuing for courses, pass, setback and failure of students, and the limitation of resources in the BEE schools were modeled for variable numbers of "A" courses and enlisted student ratings. A travel delay between schools was modeled as a direct step function of the distance involved.

For each course modeled, data included course duration, interconvening time between classes, minimum and maximum class sizes, pass, failure and setback rates, and the time, measured from the beginning of the simulation, when the first class is to convene. For each rating modeled, data included course sequence information (variable number of courses and school sequences), the number of that rating expected to report to basic training for the entire Navy each year for up to five years, the fractions of that total which report to each of three bases for basic training, the fractions which report each month, and the fractions which report each day of the week. Fleet returnees are modeled to enter the system as an across-the-board percentage of all basic training graduates.

Output for each course included average, maximum and current queue lengths; standard deviation of the queue length; average waiting time in the queue for the course; total numbers of students

who had started, failed, passed, and been setback in the course; and the number who were under instruction at the time the report was written. Output for each rating included a list of the "A" courses taken, numbers of regular recruits and fleet returnees who had completed training, and average times required to complete training, with optional histograms.

A preliminary check was made for bottleneck situations before the simulation was started.

ACKNOWLEDGEMENTS

Much credit is due to Lucy Morse and Thin Dong for their tireless efforts in reducing boxes of data to the form used by this program and to Enrique Daboin, who did much of the initial work and structuring of the program. In addition to Lucy, Dong and Enrique, I would also like to thank my advisor and committee director, Dr. G. E. Whitehouse, for his supervision and frequent consultations throughout this project, Dr. Darrell G. Linton for his helpful proofreading and comments (especially in the examples section), Morris Middleton, Dr. Bill Rankine, and Gary Hodak for their help in describing and defining the model, Jean O'Reilly for her patient assistance with what was thought to be a bug but turned out not to be, Sabra and Sam for their friendly assistance with their respective computer systems, and Mary Garner, Tom Peeples, Chris Maukonen and Bill Embach for their assistance with the printing of the final document.

This research was funded under a contract with the Training Analysis and Evaluation Group (TAEG) at the Naval Training Center, Orlando, Florida.

TABLE OF CONTENTS

INTRODUCTION	1
------------------------	---

PART 1. USER'S MANUAL

1.1 MODEL DESCRIPTION	4
---------------------------------	---

- 1.1.1 Problem Definition
- 1.1.2 Abstract of Solution
 - 1.1.2.1 Data
 - 1.1.2.2 Program logic
- 1.1.3 Timing

1.2 USE OF MODEL	12
----------------------------	----

- 1.2.1 Data Entry
 - 1.2.1.1 SLAM statement cards
 - 1.2.1.2 Simulation specifications cards
 - 1.2.1.3 Student type data cards
 - 1.2.1.4 Course data cards
 - 1.2.1.5 Main program
- 1.2.2 Report Interpretation

PART 2. TECHNICAL MANUAL

2.1 INTRODUCTION TO SIMULATION	38
--	----

- 2.1.1 Entities and Attributes
- 2.1.2 Events
- 2.1.3 Resources

2.2 DOCUMENTATION OF SUBPROGRAMS	46
--	----

- 2.2.1 Subroutine INTLC
- 2.2.2 Subroutine EVENT
- 2.2.3 Subroutine RCRT
- 2.2.4 Subroutine TRAVEL
 - 2.2.4.1 Function TRTIME
 - 2.2.4.2 Subroutine FLERN

2.2.5	Subroutine ROUTE	
2.2.6	Network Portion: BEE chair acquisition	
2.2.7	Subroutine TDST	
2.2.8	Subroutine CLASS	
2.2.9	Subroutine CLASSE	
2.2.10	Subroutine STATS	
2.2.11	Subroutine FAILUR	
2.2.12	Subroutine REPORT	
2.2.13	Subroutine STATUS	
2.2.14	Subroutine ERRORA	
2.2.15	Subroutine PRECHK	
2.2.15.1	Subroutine BEECHK	
2.2.15.2	Subroutine ASCHK	
2.3	STUDENT FLOW PATHS	75
2.3.1	Regular Type Students	
2.3.2	A-Before-BEE Students	
2.3.3	AFUN Students	
2.3.4	Examples	
2.4	TIMING CONSIDERATIONS	97
APPENDICIES		
A.	SAMPLE DATA	101
B.	PROGRAM LISTING	108
C.	SAMPLE OUTPUT	137
D.	EXPANDING STUDENT TYPE AND COURSE NUMBER LIMITATIONS	149
E.	TIME AND SPACE SAVING IDEAS	151
F.	ERROR MESSAGES	154
G.	SLAM MODIFICATIONS	156
GLOSSARY OF PROGRAM VARIABLES		158
LIST OF NAVY ABBREVIATIONS		161
REFERENCES		164
INDEX		165

LIST OF TABLES

1. Program counters	47
2. CPU time (seconds)	54
3. Authorized travel time via POV (actual days)	61
4. COMMON statements and arrays affected by changing limits . . .	150
5. SLAM array sizes	152

LIST OF FIGURES

1. General routes of students	5
2. Flowchart of subroutine TRAVEL	56
3. Flowchart of function TRTIME	63
4. Flowchart of subroutine CLASSE	67
5. Student paths through the Model	76
6. Sample output for one rating of regular type students	79
7. Yearly fluctuation in input to basic training	83
8. Weekly fluctuation in input to basic training	83
9. Histograms of time-in-system for a single rating	87
10. Sample output for one rating of A-before-BEE students	88
11. Sample output for one rating of AFUN students	93

INTRODUCTION

In Naval training "pipelines" it has not been uncommon for long queues of students to develop, waiting to take various courses. While waiting for their classes to convene, the men are frequently assigned to duties which, though necessary (eg. watch-standing, building and grounds maintenance), are not commensurate with their current training or pay grades. A program has been written to model student flow and to be used as an aid for making strategic management decisions to help minimize queue lengths. Given the frequencies of arrivals of various types (ET, EM, GSM, ect.) of USN recruits, their course sequences, the course convening schedules, failure and setback rates, this program models the flow of students in electronics fields from their arrivals to basic training and through their training pipelines, with prioritized queuing and travel time between bases.

The SLAM (simulation language for alternate modeling)* simulation package has been used. It has many useful routines for modeling queues, resources, and scheduling and processing of events. Pritsker and Pegden* is an excellent simulation text and reference manual for SLAM. The model was developed on the VAX 11/780 in the Experimental Computer Simulation Laboratory at the Naval Training

*A. Alan B. Pritsker and Claude Dennis Pegden, Introduction to Simulation and SLAM (New York: Wiley and Sons, 1979).

Center, Orlando. The system source code required 56 K bytes of memory for the model and 258 1/2 K for the SLAM processor. Both parts are written in FORTRAN. Some effort was made by part of the research team to translate the SLAM processor into BASIC.

The "User's Manual" for the program is in two parts. The first, "Model Description," describes WHAT the program does. The second, "Use of the Model," explains the data entry and reports. The "Technical Manual" is intended to serve as an aid for maintenance and program modification. It should be studied thoroughly before any program modification is attempted. It will explain each subroutine in detail including flowcharts, definitions of the variables and explanations of parts of the code. Three examples are discussed in the "Student flow paths" section of the Technical Manual. Sample data, program and output listings are provided in appendices A, B and C. A glossary of the major program variables is given at the end.

PART 1
USER'S MANUAL

SECTION 1.1

MODEL DESCRIPTION

1.1.1 Problem Definition

Students are classified in three ways. The finest distinction is by "ratings." "Rating" is the Navy's term for occupational specialty. A student might be rated as an "Electrician's Mate" (EM), "Electronics Technician" (ET), etc. Several ratings are listed in the "List of Military Abbreviations" at the end of this report. For this analysis, these ratings, or "student types," are grouped into three broad categories: Regular types, A-before-BEE types, and AFUN types (described below). A single rating can be in only one of these broad categories. Every rating will consist of "regular recruits," who are just entering the Navy and training for the first time, and "fleet returnees," who have completed at least one tour of duty or are changing ratings. Every student modeled will have a rating, and be either a regular recruit or a fleet returnee. Every rating will be either a regular type, an A-before-BEE type or an AFUN type.

All "regular recruits" begin training with basic training. After successful completion of basic training: "regular type" students will take a sequence of courses at Basic Electricity and Electronics (BEE) school followed by a sequence of courses at "A" school. "A-before-BEE" students will take their "A" school sequences before the BEE sequence. "AFUN" students will take one lockstep

course, called "AFUN" (aviation fundamentals), then the BEE sequence, and then the "A" sequence. The general category for a student and the exact sequences of courses he takes are specified by his rating. Figure 1 portrays the general routes students can take.

Basic training and "A" courses fall into a category called "lockstep." Lockstep courses form classes* on a regular schedule, and each lockstep class endures for a fixed number of days. Basic training courses differ from "A" courses in that there is no limit to the number of basic training classes ("companies") which can form on a given day.

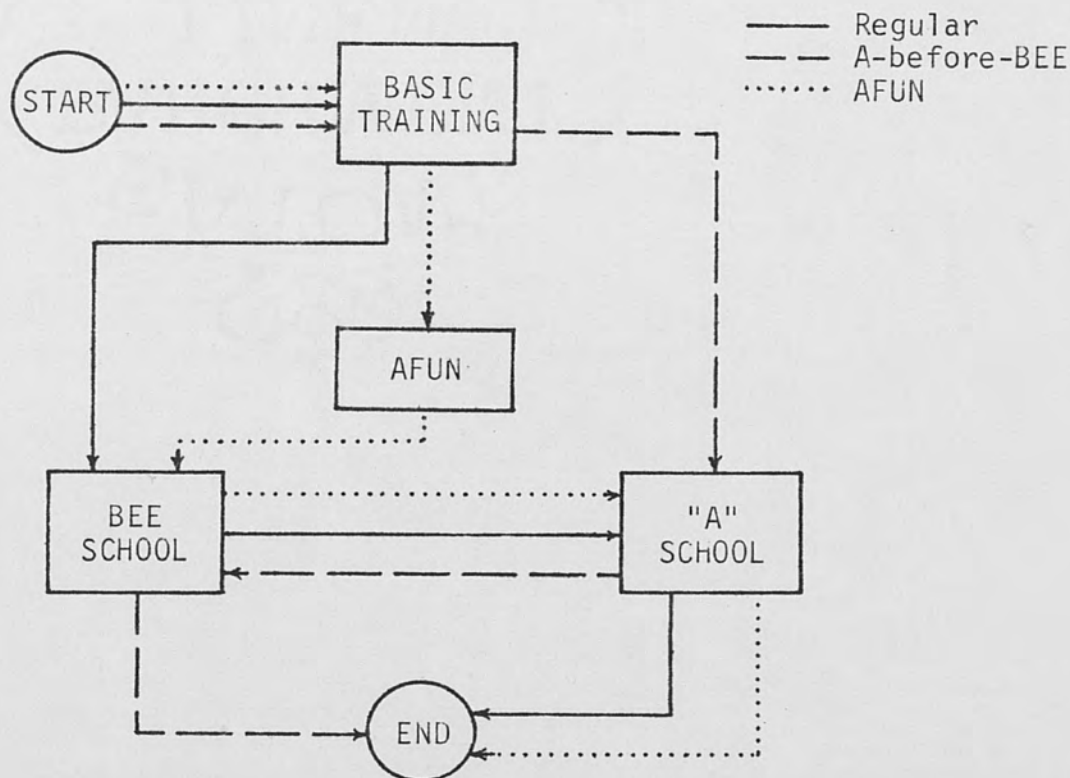


Figure 1. General routes of students.

BEE courses differ from basic training and "A" courses in that they are "self-paced," i.e., the student progresses through the courses material at his own pace. He does not wait for a class convening; he waits in line for a study carrel ("chair") to be vacated. When he graduates from or fails the course, his chair is made available for the next waiting student.

Fleet returnees are modeled to enter the system following basic training. From this point they will flow through the rest of their respective pipelines just as their regular recruit counterparts, except that fleet returnees have head-of-line privileges over regular recruits in the BEE queues. (Such a priority could easily be implemented for all courses, but it is currently used only for BEE.) Fleet returnees do not take basic training.

1.1.2 Abstract of Solution

1.1.2.1 Data

In order to run the simulation, some sort of "driver" is necessary to generate students of the various ratings and start them in the system. The driver for this program requires for each rating modeled: 1) the number of recruits of that rating expected to arrive

*Definitions: Throughout this report the term "course" is used to refer to an organized body of information/instruction. The term "class" will refer to a group of persons which meet at a certain time to take the course. "School" refers to a place where an entire sequence of courses ("A" or BEE) is taught for a given type of student. "Base" or "place" refers to a major Naval installation. Bases modeled include Orlando, Great Lakes, San Diego, Memphis, Corry Station, Fleaswtrac, New London, Charleston, Mare Island, Port Hueneme, and Gulf Port.

at Recruit Training Commands (RTCs) for the entire Navy for each year modeled (up to 5 years); 2) the fraction of the total yearly of that rating input expected to arrive each calendar month (average = $1/12$); 3) the fractions of total input of that rating expected to come on each day of the week (average = $1/5$); and 4) the fractions of that rating which go to each of three bases (Orlando, Great Lakes, San Diego, -- average = $1/3$) for their basic training. The driver data is read from student data cards and the mechanism is encoded in subroutine RCRT. It should not be difficult to install different drivers if considered necessary.

For each type of student (rating) modeled, in addition to the driver data, student data cards specify the number of BEE courses to be taken (up to 3) and the "A" course sequence (up to ten "A" courses). If a student is of one of the AFUN types, the AFUN course is listed as the first of the "A" sequence, with the true "A" courses following. (AFUN students are thereby limited to 9 true "A" courses.) Whether a student falls into the regular type, A-before-BEE or AFUN broad category is also specified in the data cards for his rating.

For each course modeled, the user must specify (1) the number of working days between class convenings, (2) the duration of a class, (3) the maximum and (4) minimum numbers of students allowed in a class, and (5) attrition and (6) setback rates. Having the time between class convenings be less than the course duration implies the running of multiple tracks -- or at least overlapping classes. Having the course duration be shorter than the interconvening time implies

breaks for the staff between classes. The three RTCs (at Orlando, Great Lakes and San Diego) are not required to hold classes on the same schedules.

A student's prognosis in a course will be either "pass," "fail," or "setback." Failures leave the system. Setbacks occur when, due to extenuating circumstances, a student is allowed to finish with the next class.

1.1.2.2 Program logic

Each simulation day, at 1200, the driver generates the specified numbers of each type of student to arrive at the three RTCs for that day and places them in the queues for basic training. Recruits will wait in these queues for the formation of basic training classes (companies).

Every day basic training queues are checked, and basic training companies are formed. Student prognoses and completion times are determined, and pass/failure events are scheduled at that time.

Upon graduation from basic training, a given percentage (input datum FLER) of basic training graduates will be cloned to simulate the arrivals of fleet returnees to the system. Graduates and clones are routed to their next schools ("A" or BEE). Regular recruits will be delayed by some travel time (which may be 0). Fleet returnees are assumed to enter the system at the school to which they are being sent at this point, and will not undergo travel.

The BEE sequence consists of one to three consecutive courses. After arrival to BEE school, a student is filed in the queue for a chair for his first BEE course. All students take at least one BEE course. After getting a chair, a student's prognosis is determined and his course completion/failure event is scheduled.* Upon completion of a BEE course, the student will free his chair and get in line for the next course (if he is to take more), travel to the base where his "A" school is taught (regular and AFUN types), or leave the simulation (A-before-BEE types). His chair is made available to the next waiting student.

At "A" school a student takes a sequence of up to ten predetermined "A" courses. Upon arrival to "A" school, a student is filed in the queue for the first "A" course specified. On class-convening days queues are checked and classes are formed. For each student started in a class, his prognosis in that course is determined at the time he starts, and his graduation or failure event is scheduled at that time. Upon successful completion of a course, a student gets in line for the next course in his sequence until he has completed his entire "A" sequence. Upon completion of "A" school, statistics are gathered for regular and AFUN types, and A-before-BEE

*Several distributions are available with SLAM for modeling randomness. Among these the normal, exponential, and Erlang distributions are considered as being possibly appropriate for modeling BEE course durations. Parameters might be functions of the course, student rating, and even an individual student intelligence attribute. Currently, BEE time is rather primitively arbitrated to be the constant entered in the data, 10% greater if he is "setback," or 40% less if he is to fail (see subroutine EVENT, variable BEETIM).

types are sent to BEE school.

Students of different ratings may take the same "A" course, i.e., pipelines may intersect, not only in "A" courses, but also in BEE and basic training courses.

If a student fails a course, he is dropped from the simulation, and no statistics are taken other than a failure count by course.

If he is setback, his time of graduation from that course is simply delayed by the time between class convenings for that course, i.e., he graduates with the next class. It is assumed that all setbacks will graduate with the next class. The only exception to this is with basic training, where classes convene every day. Basic training setbacks are delayed 5 days (1 week).

Travel time is modeled between basic training, BEE school and "A" school (subroutine TRTIME). It is assumed that there will be no travel within course sequences, i.e., the entire BEE sequence is taken at one place, and the entire "A" course sequence is taken at one place. Travel occurs only when there is a change of school. If the next school is at the same base, the travel time is currently set equal to 0, though it may be reasonable to model some interschool delay for the process of changing commands.

Students wait in queues for all courses. To enter a queue for a course a student must have successfully completed all prerequisites, and he must also be at the base where the course is taught (He enters the queue after travel). Within queues there are two general priority considerations. The major consideration is the student's origin

(regular recruit, fleet returnee, foreign, etc.). The minor consideration is based on rating. Fleet returnees have priority over regular recruits. Within the same large category priority is based on rating, as specified in the data. Currently only regular recruits and fleet returnees are modeled, and all ratings are given the same relative priorities.

Upon completion of the training pipeline, statistics are collected by student type to determine average time in the system.

1.1.3 Timing

It is assumed that the following events, when scheduled at the same time, will occur in the order listed: course completions, arrival to a school after travel and entry into a queue, class convenings, printing of reports. Classes are modeled to convene at the beginnings of days. Recruits are modeled to arrive at 1200 each working day. For more discussion of the sequencing of synchronous events, see the section on "timing considerations" at the end of the technical manual.

SECTION 1.2

USE OF MODEL

1.2.1 Data Entry

Four groups of data cards are required. These will be termed the "SLAM statement" cards, "simulation specifications" cards, "student data" cards and "course data" cards. The cards are described below and a sample listing of a full set of data cards is given in Appendix A.

1.2.1.1 SLAM statement cards

These cards give certain control parameters and describe the network portion of the model. They are commonly considered to contain simulation control macrostatements (starting in column 1) and network macrostatements (starting in column 7), but they are actually data cards, read and interpreted by the SLAM processor. The potential user is referred to Pritsker and Pegden (1979) as a text and manual for SLAM. Much of the following description is paraphrased from chapter 5 of their book.

All SLAM statements must end with a semicolon (;). Anything coming after the semicolon will be ignored (may serve as a comment). Omission of any field in any SLAM statement will cause its default value to be assumed. Delimiters separating fields include commas (,), slashes (/) and the semicolon. Formatting is very free. The major

restrictions are the columns in which the card begins. Spaces are allowed and will be ignored when reading the card unless they come in an alphanumeric field. Certain special symbols [, / () + - * ' ;] are not allowed in alphanumeric fields (Pritsker and Pegden, 1979, p. 121). Very little modification of the SLAM statement cards should be necessary. Five user modifications are allowed:

(1) The GEN card must be the first of the data cards. It specifies several output options. The general form of the GEN statement is as follows:

```
GEN,NAME,PROJECT,MONTH/DAY/YEAR,NNRNS,ILIST,IECHO,IXQT,IPIRH,ISMRY;
```

The word GEN must start in column 1.

NAME and PROJECT are the analyst's and project's names. These may be up to 20 characters each and may include blanks. The default values of both are blanks.

MONTH/DAY/YEAR are integers specifying the date. The default values are 1/1/2001.

NNRNS is the number of runs = 1. Only one run is permitted with this model because the simulation specifications, student and course data cards must be read during each run. The default value of NNRNS is 1.

ILIST ('Y' or 'N'). If 'Y,' a listing of the SLAM input statements including any error messages will be printed prior to the run. The default value for ILIST is 'Y'.

IECHO ('Y' or 'N'). If 'Y,' a SLAM echo report is printed prior to the run. SLAM echo reports include listings of the options chosen, the parameters specified in the LIMITS card, all queues with their respective priority rules, items on which statistics are to be collected (see STAT cards) and pseudorandom number stream seeds. The default value of IECHO is 'Y'.

IXQT ('Y' or 'N'). If 'Y,' execution will be attempted if no SLAM input errors are detected. The default value is 'Y'.

IPIRH ('Y' or 'N'). If 'Y,' the heading "INTERMEDIATE RESULTS" will be printed prior to the simulation run. The default value is 'Y'.

ISMRY ('Y' or 'N'). If 'Y,' a SLAM summary report is printed following the run. Several types of reports are possible with this system: SLAM summary reports, Model reports, Status reports and Error reports. The Model reports are custom tailored to the pipeline problem and contain much of the information contained in SLAM summary reports. Model reports will be discussed in the Report Interpretation section. Status reports, also called "snapshots," describe the current state of the system, fully describing queues, classes and transients with student listings. Since Status reports list one line for each student currently in the simulation, they can be very long. Error reports attempt to describe the state of the model when error conditions are encountered. These too tend to be very long. Status and Error reports are discussed in more detail in the Technical Manual.

SLAM summary reports include several statistics on queues (maximum, minimum, average and current lengths and standard deviations of the lengths) and resource (BEE seat) utilization (average, maximum, current utilization, standard deviation). SLAM summary reports will always include all histograms specified. These can be very long. One can use the SLAM MONTR statement to cause SLAM summary reports to be printed periodically. The user is referred to the SLAM text for information about this statement. For more information about SLAM summary reports see Pritsker and Pegden (1979), pp. 157-61. The default value of ISMRY is 'Y'.

Omission of the GEN card will cause all default values to be assumed.

(2) The LIMITS card must be the second card in the deck. The general form is as follows:

```
LIMITS,MFIL,MATR,MNTRY;
```

The word LIMITS must begin in column 1.

MFIL is the maximum number of files (queues) to be used. In this case it is the number of courses to be modeled, counting the three basic training courses, twelve BEE courses and all actual and dummy "A" courses + 1. Currently this number is limited to ≤ 99 . The default value of MFIL is 1.

MATR is the number of regular attributes per entity. It should always

be the integer 2 (for this model). See the technical manual for discussion of entities and attributes. The default value of MATR is 0.

MNTRY is the maximum number of concurrent entries expected during the simulation. An entry may be a modeled student or a scheduled event (class convening or report printing). The modeled students make up the vast majority of the entries. Beware that students may pile up at "bottlenecks" making the number of concurrent entries very large for long simulations. The program keeps track of the maximum number of entries and prints this number at the ends of Model reports. The default value of MNTRY is 0.

(3) Unless otherwise specified in a PRIORITY statement (SLAM statement card), all students wait in queues in FIFO (first-in-first-out, or first-come-first-served) order. To change this priority rule a PRIORITY statement is required. The general form is as follows:

```
PRIORITY/IFILE,ranking[/repeats];
```

Priority rules for several files (queues) may be specified by one PRIORITY statement.

The word PRIORITY must begin in column 1.

IFILE is the number of the file (course number, as ordered in the course data cards) the specification applies to.

All BEE rankings for this model are "HVF(2)," which means "high value

first" based on attribute 2. Priority levels are stored in attribute 2 of each entity (student). The higher the value of attribute 2, the higher his priority. He gets in line ahead of all other students with lower values of attribute 2. Priority assignments are discussed under student data cards. Other priority rankings made available by SLAM are FIFO (first-in-first-out), LIFO (last-in-last-out) and LVF(N) (low-value-first based on attribute N). The default ranking is FIFO.

A PRIORITY specification of HVF(2) or LVF(2) for the highest file (file MFIL) must be made (see subroutine RCRT, randomizer). In order to fulfill the timing considerations, mentioned in the Model Description and explained in the Technical Manual, a PRIORITY specification of "NCLNR,HVF(3)" must be made.

(4) Two STAT statement cards must be included for each type of student modeled (ET, EM, GSM, etc.): one for regular recruits and one for fleet returnees of each type. The form is as follows:

```
STAT,ICLCT,ID, NCEL/HLOW/HWID;
```

The word STAT must begin in column 1.

ICLCT (integer) is the statistic type number. Each student type has two STAT cards. Student type numbers (NS) are assigned in the order the student data cards are read. The first STAT card for student type NS will apply to regular recruits of that type and

have $ICLCT = 2 * NS - 1$. The second STAT card will apply to fleet returnees of that type and have $ICLCT = 2 * NS$. ICLCT has no default value.

ID is a name for the student type. It may include up to 20 alphanumeric characters including blanks, but must not include any of the special characters $[,/()+-.*';]$. This name is used only in SLAM summary reports and for labeling histograms. If one uses names such as ET, EM, GSM,... one must remember to change these cards whenever one changes the student data cards. The student data cards have their own fields for specifying alphanumeric student type names for use in the Model reports. The default value of ID is blanks.

NCEL (positive integer constant) is the number of cells desired in a histogram. The number of cells actually listed will be NCEL plus 2: one cell for the range below the histogram range and one one for the range above. The total number of cells for all histograms, counting these extra 2 for each histogram, is currently limited to 500. See Appendix D for directions to increase this limit. The default value of NCEL is 0 (no histogram).

HLOW (real constant) is the upper limit of the first cell of the histogram.

HWID (positive real constant) is the width of each cell.

The STAT statement is necessary in order to set up arrays for the collection of time-in-system statistics for the various types of students. These statistics include maximum, average, minimum and standard deviation. Only the average is printed in Model reports. SLAM summary reports print the other values. The extended form of the STAT statement allows for the plotting of histograms. Histograms more completely describe the distributions of the data, showing the difference between the average value and the most probable value. They also allow one to approximate the median and mode. The histograms may serve as discrete probability functions for the time in the system. The cumulative functions are also plotted. Histogram printings can be very long. Each histogram printed has NCEL + 11 lines (this includes heading and axis labeling). Using SLAM subroutine PRNTH(ICLCT), it is possible to get printings of single histograms. SLAM summary reports automatically print all histograms specified. Setting the flag IHST = 1 in the second simulation specifications card will cause all histograms to be printed in Model reports.

The SLAM statements between the NETWORK and INIT statements begin in column 7 and must not be altered.

(5) The INIT statement card specifies the beginning and ending times for the simulation. THIS IS THE ONLY CARD THAT MAKES THE SIMULATION STOP. The INIT statement is of form:

INIT,TTBEG,TTFIN;

The word INIT must begin in column 1.

TTBEG (real constant) is the beginning time for the simulation. The default value = 0.

TTFIN (real constant) is the stopping time for the simulation. The model is currently limited to 5 years (1300 working days) due to the dimension of the yearly demand array, YD (see student data card type 3).^{*} All desired reports should be scheduled at or before this time (see simulation specifications cards). THE DEFAULT VALUE FOR TTFIN IS UNLIMITED.

The timestep is the working day. Each year is modeled to have $52 \times 5 = 260$ working days. Holidays were not considered because it had been the programmer's experience that the Navy made up for holidays by working extra before and after. The year could be modeled to have any number of days with some modification to the driver (recruit generator) coding.

The SLAM input cards end with the FIN; card.

^{*}Other FORMATS must be changed (in INTLC) for the reading and echo-printing of student data if one wishes to model more than five years.

1.2.1.2 Simulation specifications cards

Three simulation specifications cards are currently used. The are of fixed FORMATS.

(1) Simulation specifications card 1:

ORL, GL, SD, MEM, COR, FLEA, NLON, CHS, MARI, PORH, GULP
FORMAT (16I5)

These variables represent the numbers of "A" courses taught at Orlando, Great Lakes, San Diego, Memphis, Corry Station, Fleaswtrac, New London, Charleston, Mare Island, Port Hueneme and Gulf Port, respectively. They are all integers.

It is recommended that as many as possible "dummy" courses be included at each place in order to facilitate easy additions without renumbering all the courses. (See NASCHD in student data discussion). The program is currently limited to the modeling of 83 distinct "A" courses. Appendix D describes how this limit can be increased.

(2) Simulation specifications card 2:

IECHOICASE IHSTFACTR FSTR TBR FSTS TBS FLER TBE

This card is included only for convenience. It indicates where the items of simulation specifications card 3 are to be placed. This card may be eliminated if one deletes the slash and comma (/,) at the beginning of FORMAT statement 9013 in subroutine INTLC.

(3) Simulation specifications card 3:

IECHO, ICASE, IHST, FACTOR, FSTR, TBR, FSTS, TBS, FLER, TBE
 FORMAT(3I5, 13F5.0)*

Columns 1-5: IECHO. If IECHO is not equal to zero, the data will be echo-printed with headings and error messages, and a preliminary data analysis, which searches for any obvious bottlenecks, is performed before any simulation takes place. This is highly recommended for any new data set, and is handy for reference when analyzing output. A timing study showed that this echo-print and preliminary analysis required about 5 seconds of CPU time on the VAX 11/780.

Columns 6-10: ICASE. If ICASE = 0, all bases will be modeled. If ICASE equals an integer between 1 and 11 (inclusive), it refers to one of the eleven bases, as ordered above. All students other than those going to that place for A-school are eliminated when their times come to begin their "A" sequences, unless they are of the A-before-BEE type -- none of these types are eliminated based on ICASE. It is intended for ICASE = N to have the effect of modeling only base N. Of course all bases which send students to base N must be modeled for prerequisite courses. A-before-BEE students must be retained to fairly model competition for chairs

*The F5.0 format specification allows the user to place the decimal point wherever he sees fit within the 5-space data field. If no decimal point is entered, it is assumed to come after the fifth space.

at all BEE schools. The reason for modeling only one base at a time is to try to hold down the number of concurrent students in the simulation.

Columns 11-15: IHIST. If IHST = 1, histograms of student training time will be printed in Model reports. Histograms can be quite lengthy and are a regular part of SLAM summary reports.

Columns 16-20: FACTOR. FACTOR is a decimal expression indicating what fraction of the total recruits specified in the driver data are actually to be modeled. Class sizes are scaled and rounded accordingly. Class sizes and student counts are descaled back to full scale for the output reports. This does induce some error. (We cannot model a fraction of a BEE chair.) Quantities which are scaled/descaled include: minimum and maximum class sizes, numbers of students generated by the driver, numbers of starts, failures, setbacks and graduates for each course, current class size (number under instruction) for each course, number of recruit arrivals, fleet returnees and transients counters, current, maximum and average queue lengths and queue length standard deviations. Quantities which are NOT scaled include: ORL, GL, SD, MEM, COR, etc., all times, FLER (described below), numbers of courses taken, course sequence data, class durations and convening schedules, course failure and setback rates, waiting times, times in the system, maximum number of concurrent entries.

Columns 21-25: FSTR (Time until first Model Report) and Columns

26-30: TBR (Time between Model Reports). The first Model report will come FSTR simulated time units (working days) from the beginning of the simulation and successive reports will be spaced TBR time units apart. Model Reports give current accountings of the numbers of recruits and fleet returnees to arrive to the system, numbers of starts, failures, setbacks, under instruction and graduates for each course, queue length information (maximum, current, average, standard deviation), numbers of students to finish their various pipelines and the average time it took them to finish by rating (divided into regular recruit and fleet returnee subgroups). Each report will print two lines for each course modeled and one line for each student type modeled plus about fifty additional lines for headings and miscellaneous information (not counting histograms). Each report required about 2 seconds of CPU time on the VAX. If IFST = 1, two histograms will be printed for each student type. The length of a single histogram is NCEL + 11 lines, where NCEL is the number of cells specified in the STAT card.

Columns 31-35: FSTS (Time until first Snapshot) and Columns 36-40: TBS (Time Between Snapshots). FSTS and TBS works the same way as FSTR and TBR, but they pertain to Status Reports (snapshots). Status reports describe the current state of the system. The reports include for each course listings of who is waiting in the queue and when they arrived, and who is currently under instruction and when they are expected to finish (graduate of

fail). Transient information includes who the transients are (rating, arrival time to the system), where they are going (base), for what (BEE or A school), and when they are expected to get there. Since one line is printed for every student in the simulation these reports can be quite lengthy and one must exercise discretion when calling for a status report. Status reports do not descale information.

Columns 41-45: FLER. Fleet returnees are created as flat percentage of the basic training graduates. FLER is this percentage. For every basic training graduate there is a probability, FLER, of a clone being created at that time to simulate a fleet returnee entering the system. The only distinction made between fleet returnees and regular recruits is the priority for queue positions.

Columns 46-50: TBE (Time Before deliberate Error reports). Error reports are intended for use only with debugging and program modification. Error reports can be quite long, and one may wish to edit the error report-writing routine (subroutine ERRORA) before using it. It is described in the technical manual. An error report contains a deliberate FORTRAN error (SQRT(-1.)) which will stop the simulation and provoke system traceback and diagnostics. Unless one wishes an error report, a large number (99999) is recommended for TBE.

1.2.1.3 Student type data cards

For each student type, three cards are required:

(1) Student data card type 1:

```
NO, STYP1(I), STYP2(I), BE(I), (NROUTE(I, J), J=1, 11),  
  (NASCHD(I, J), J=1, 10), PR(I)  
FORMAT(I3, 2A4, I3, 11I2, 10I3, F5.0)
```

Columns 1-3: NO is a user assigned number for the student type. it is not used by the program, but it is intended to help the user to keep track of the student data cards and keep them in order. The program will assign student type numbers consecutively as it reads the student data cards. Leaving NO blank or zero will flag the end of the student data cards.

Columns 4-11: STYP1(I),STYP2(I). "I" is the program's counter for student types. STYP1(I) and STYP2(I) represent an alphanumeric name for the Ith student type (ET, EM, GSM, etc.) which will be used in the Model and Status reports. This name can be anything, with one exception: If the student type is in the AFUN category STYP2(I) must be 'AFUN'.

Columns 12-14: BE(I) tells how many BEE courses a student type is to take. It must be 1, 2, 3, -1, -2 or -3. A negative sign indicates that this type of students is to take its "A" course sequence before the BEE sequence. All students will take at least one BEE course. (BE(I) should be positive for AFUN students.)

Columns 15-36: (NROUTE(I,J),J=1,11). This is an array of eleven

2-digit integers. $NROUTE(I,J)$ tells the number of the base to which student type I will go for BEE if he is now ready to take the BEE sequence and is at base J. The bases are numbered as ordered as in the section describing simulations specifications card 1. All $NROUTE$ entries must be 01, 02, 03 or 04 (zeros may be left blank) because BEE is taught only at Orlando, Great Lakes, San Diego and Memphis.

Columns 37-66: $(NASCHD(I,J), J=1,10)$. This is a sequence of up to ten 3-digit integers indicating the "A" course sequence for student type I. AFUN and "A" courses are numbered 16 through 98 (described under course data cards). Any student type may take up to ten "A" courses (including AFUN). The identifying numbers of the courses taken are entered in sequential order from left to right. The remaining spaces should be left blank. For AFUN types, the AFUN course is entered as $NASCHD(I,1)$ and the true "A" courses are listed as $NASCHD(I,2), \dots$.

Columns 67-71: $PR(NS)$ is the priority level of student type NS relative to the other student types. Students wait in queues in High Value First (HVF) order by this priority. Fleet returnees have 100.0 added to their priorities when they are cloned (subroutine FLERN).

(2) Student data card type 2:

```
(FM(I,J),J=1,12)
FORMAT(10X,12F5.0)
```

Columns 1-10: unused. One may enter student type identification numbers here.

Columns 11-80: (FM(I,J),J=1,12). These are twelve 5-digit decimal numbers indicating the fractions of type I students who arrive during each of the twelve months of the year. The twelve numbers should average 1/12 for each student type.

(3) Student data card type 3:

```
(YD(I,J),J=1,5),(DOWF(I,J),J=1,5),(BF(I,J),J=1,3)
FORMAT(5X,13F5.0)
```

Columns 1-5: unused. Student type identification number recommended.

Columns 6-30: (YD(I,J),J=1,5). YD(I,J) is the total yearly demand (input to the Navy) of student type I for year J. Up to five consecutive years may be modeled.

Columns 31-55: (DOWF(I,J),J=1,5) (Day Of Week Factor). DOWF(I,J) tells what fraction of the type I students are expected to arrive on the Jth workday of the week. These should average 1/5 for each student type. Students reporting over the weekend should be included in Friday's fraction.

Columns 56-70: (BF(I,J),J=1,3) (Base Factor). BF(I,J) is the decimal fraction of all type I recruits to enter the Navy who report to

base J for basic training. Base J will be Orlando, Great Lakes or San Diego. For each student type the BF entries should average 1/3.

The three data cards are entered for each student type until all student types have been described. A blank card following a type 3 student data card will flag the end of the student data. Currently the program is limited to a maximum of 49 student types. Appendix D describes how to extend this limit.

1.2.1.4 Course data cards

Only one data card is required to describe each course. Cards for basic training courses at Orlando, Great Lakes and San Diego must come first, in that order, followed by the cards for BEE1, BEE2 and BEE3 at Orlando, BEE1, BEE2 and BEE3 at Great Lakes, three BEE's at San Diego and three BEE's at Memphis. Then come all the AFUN and "A" course cards grouped by place: First ORL cards for the "A" courses taught at Orlando, then GL cards for the "A" courses taught at Great Lakes, SD cards for the "A" courses at San Diego, MEM cards for all the AFUN and "A" courses taught at Memphis,* and so forth for Corry Station, Fleaswtrac, New London, Charleston, Mare Island, Port Hueneme, Gulf Port in that order. It is recommended that as many as possible "dummy" courses be included to allow easy additions later on without having to renumber. Dummy courses must be counted in ORL, GL,

*AFUN courses are actually only taught at Memphis, but they could be included with "A" courses at any base.

SD, etc. and have OPD entries (described below). No more than 98 courses, including the 3 basic trainings, 12 BEE's and all dummies are currently allowed (see Appendix D for expansion).

The form of the course data card is:

```
NO,CNAM1(I),CNAM2(I),FAIL(I),SBACK(I),MINS(I),MAXS(I),ADV(I),
  FIRSTA(I),OPD(I)
FORMAT(I3,2A4,4F5.0,3F5.0)
```

Columns 1-3: NO is the course identification number. The program will not use this number, but will assign its own numbers consecutively as the data is read. The first three courses will be numbered 1, 2 and 3 and used for basic training courses at Orlando, Great Lakes and San Diego. The next three will be numbered 4, 5 and 6 and used for BEE1, BEE2 and BEE3 at Orlando. The next three will be used for the BEE's at Great Lakes, the next three for BEE's at San Diego and three more for BEE's at Memphis. AFUN and "A" courses must follow and will be numbered starting at 16 and going up to a maximum of 98.

Columns 4-11: CNAM1(I),CNAM2(I). Name of course I (CDP 6015, RTC ORL, BEE1 ORL, unused, etc.).

Columns 12-16: FAIL(I) is the failure rate for course I. It is entered as a percentage. If 25% of the students fail a given course, the entry could be 00025 or 25.00 (zeros may be left blank).

Columns 17-21: SBACK(I) is the setback rate for course I, entered as a percentage.

Columns 22-26: MINS(I) is the minimum class size entered as an integer (right justified) or real number.

Columns 27-31: MAXS(I) is the maximum class size. This number should be greater than zero. For basic training companies the number should be scaled to reflect the size of the company which represents the types of students modeled. For a BEE course it is the number of study carrels available for that course (multiplied by the number of shifts). Entering MAXS(I) = 0 for dummy courses will cause any students inadvertently routed to those courses to be trapped in permanent queues. It will also cause a warning message to be printed.

Columns 32-36: ADV(I) is the course duration in working days. It may be a floating point number.

Columns 37-41: FIRSTA(I) is the number of working days from the beginning of the simulation until the first convening of course I.

Columns 42-46: OPD(I) is the number of days between convenings of course I. $OPD(I) < ADV(I)$ implies the operation of multiple tracks. $OPD(I) > ADV(I)$ implies breaks for the teaching staff between one class and the next. Dummy courses should have large values for OPD (99999), since they will be scheduled as will any other course.

Exactly LASTA course data cards are read, where $LASTA = 15 + ORL + GL + SD + \dots + PORH + GULP$.

1.2.1.5 Main program

In addition to the data cards the user must provide dimensions for the large arrays, NSET/QSET and NAVY, and numbers for the I/O devices. These are specified in the main program. A sample listing of a main program is provided in Appendix B.

NSET and QSET represent the integer and real forms of the same array. The dimension of this array is specified in the main program:

```
DIMENSION NSET(65000)
COMMON QSET(65000)
EQUIVALENCE (NSET(1),QSET(1))
NNSET = 65000
```

Their dimensions must be the same and NNSET must be set equal to that number. The arrays hold information about the entities and events to be processed, coding of the SLAM input statements, and data for any tables or histograms to be printed. The dimension should be greater than 6 times the maximum number of entries expected. Failure to make the dimension large enough will result in a SLAM error message (see Appendix E). A special insert into the SLAM processor will cause the printing of a Model report and a SLAM summary report before aborting (see Appendix F) in the event of student overflow.

The array NAVY holds only auxiliary attributes of entities (students) and should be dimensioned to 4 times the maximum number of concurrent entities (students) expected during the simulation. NAVY is dimensioned in the NAVAL2 COMMON block. NDNNAVY must also be set equal to the dimension of NAVY.

NPRNT is the device to which all output is written. SLAM statement cards are read from device NCRDR. All other data cards are read from device NCRD2.

1.2.2 Report Interpretation

A sample output is presented in Appendix C. If IECHO = 'Y' on the GEN card, output will begin with a listing of the SLAM statement cards. Any other options specified on the GEN card will then follow. If IECHO = 1 on simulation specifications card 1, an echo-print of the data with headings will then follow. The fractions in the driver data (Base factors, month factors and day-of-week factors) are shortened in order to make them all fit onto one line. The values in the computer, though, will be the original data. In the listing of the "A" course sequences, the last column indicates how many courses the computer counted in the respective sequences. Data for each course indicate failure and setback rates as percentages, the maximum and minimum class sizes (without scaling), the course length, time before first convening and time between convenings in working days.

Error messages will immediately follow obviously erroneous data. If the data is echoed these messages will come in the listing. Otherwise they will appear alone.

The WHR array specifies the base number as a function of course number. It is derived from values of ORL, GL, SD, etc. For example, with the output in Appendix C, course 52 is taught at base 4 (found in column 2 of row 6).

The preliminary data analyzer computes the average yearly student input and determines, based on course sequences, failure rates and fleet-returnee input, the expected steady state output. For each course the expected input and maximum capacity are also determined. (See the technical manual for further description of the pre-execution analyzer.)

At the head of each Model report are printed the current simulation time, scaling factor and fleet returnee fraction. The current time is the simulation time in working days. The scaling factor is shown only to give some idea of the degree of error due to scaling. All counts (except "maximum number of concurrent entries") are descaled for reports. The fleet returnee fraction is not the fraction of all students which are fleet returnees. It is the fraction of basic training graduates which are cloned (at the time of graduation) to simulate fleet returnee input (subroutines TRAVEL and FLERN).

The first section of a Model report gives queue length statistics. A current queue length close to the maximum can imply two things. If the queue length is near the minimum class size, they may just be waiting for the next class to open. If the queue length far exceeds maximum class size this is a sign of a probable bottleneck in the pipeline. Waiting times can be long due to infrequent class convenings, large minimum class sizes and a lack of students to take the course, or small maximum class sizes and an overabundance of students (a bottleneck situation). In the first case a bright student

may get ahead of his contemporaries only to have to wait for them to catch up later to help form a class.

Waiting time statistics are only taken on students leaving queues; hence there may be queues indicated, but if nobody has left the queue we have no idea what the average wait will be, and the average waiting time indicated will be zero.

The second part of the Model report lists many useful numbers indicating the flow of students through the courses. For all courses the number who start should equal the sum of those who have passed, those who have failed and those who are under instruction. Of the total number of setbacks some are still under instruction and some have finished the course before the report was written. Setback counts are useless for making head counts. If one models one type of student alone the number of passes from one course will equal the sum of the number of starts and the number awaiting instruction for the next course plus any transients on their way to that course. For the whole simulation, with any number of student types, the total number of arrivals to the system (regular recruits and fleet returnees) will equal the sum of those still in the system (under instruction, awaiting instruction, and transients) and those who have left the system (failures and those who have finished their training):

$$\begin{aligned} &\text{total number of recruit arrivals} + \text{total number of fleet} \\ &\text{returnees} = \text{total under instruction} + \text{total awaiting} \\ &\text{instruction} + \text{total failures} + \text{total finishing training} + \\ &\text{total number of transients.} \end{aligned}$$

The third part of the Model report lists for each student type the number who have completed their pipeline and the average time it took them to do it. These numbers are split into regular recruit and fleet returnee columns. Basic training time is included for the recruits; it is not included for the fleet returnees. One must not misinterpret the average time as being the most probable time. A few very long samples can push the average far above the most probable or median times. One must also not fall into the trap of using a most probable time for an expected time.

Next are some accounting numbers used to complete the head count described above. All counts are descaled except for the "MAXIMUM NUMBER OF CONCURRENT ENTRIES." This last figure is useful for picking reasonable dimensions for NSET/QSET and NAVY.

If printing of histograms is specified they will come last.

Three walk-through examples are given in the "Student Flow Paths" section of the Technical Manual.

PART 2
TECHNICAL MANUAL

SECTION 2.1

INTRODUCTION TO SIMULATION

A familiarity with SLAM or some related simulation language may be essential to understanding this program. I will however attempt to explain the program as thoroughly as possible assuming little or no simulation experience on the part of the reader. It is assumed that the reader is familiar with the material in the User's Manual.

2.1.1 Entities and Attributes

The students are modeled as "entities" which flow through a network of queues and activities (courses, travel). Each student has two "regular" attributes and four "auxiliary" attributes. These attributes describe the student. Whenever he is processed his attributes are checked to determine what to do with him next. His regular attributes will be in the array ATRIB. Auxiliary attributes for all students are stored in the large real array, NAVY.

Regular attribute 2 (ATRIB(2)) holds the student's priority for queuing. Currently all queuing for RTC and "A" classes is FIFO (first in first out). Queuing for BEE courses is "HVF(2)," which means students queue first based on the values of their attributes 2 (High Value First) and then FIFO for those with equal values of ATRIB(2). ATRIB(2) is the only distinction between fleet returnees and regular students of the same type (rating). For regular recruits $0 \leq$

ATTRIB(2) < 100. For fleet returnees, $100 \leq \text{ATTRIB}(2) < 200$.

ATTRIB(1) holds a pointer which designates where, in NAVY, the student's auxiliary attributes are stored. Each student has his own unique pointer and set of auxiliary attributes. The variable NGET is frequently assigned the value of ATTRIB(1) within a subroutine. NAVY(NGET+1) will hold auxiliary attribute 1; NAVY(NGET+2) holds auxiliary attribute 2; etc.

Auxiliary attribute 1 (NAVY(NGET+1)) holds the time the student arrived to the system. For a regular recruit this will be the time he was created at an RTC base. For a fleet returnee it will be the time he was cloned from some recruit graduate. This time is used to determine his total training time when the student completes training. Training times are gathered to compute average training time by type (rating).

Auxiliary attribute 2 (NAVY(NGET + 2)) holds the student type number. This number is usually assigned to the variable NS within subroutines. Student types are numbered consecutively in the order in which the student data are read.

Auxiliary attribute 3 (NAVY(NGET + 3)) plays two roles. When a student is taking a BEE course it holds the file number (course number) of the course he is taking. When a student is in an A-school it holds the ordinal number (1st, 2nd, 3rd, etc.) of the "A" course which he is taking in his sequence. As an ordinal number, auxiliary attribute 3 is frequently assigned to the variable NA.

Auxiliary attribute 4 (NAVY(NGET + 4)) is used to determine waiting time in individual queues. When a student arrives at a queue it is set equal to the current simulation time (TNOW). When he leaves the queue the queue waiting time is computed. This is used to compute average waiting times by queue (course).

ATRIB(1) (regular attribute) has another function in addition to being the auxiliary attribute pointer. When a student begins A course, whether he will pass, fail or be setback is determined probabilistically (subroutine FAILUR). If he is to fail, he will eventually be dropped from the model. In order to save memory space, when it is determined that a student will fail, his auxiliary attributes are removed, making their space in NAVY available for a new student's auxiliary attributes. The student is tagged as a failure by setting ATRIB(1) to a negative number. Making this negative number the course number ($-1 * \text{course no.}$) will tell us later, when the student actually fails, which course he has failed.

The other uses for ATRIB(1) are to indicate student type in the randomizer in subroutine RCRT; and to indicate the course number in class convening events (see CLASS).

When one considers all the attributes, he will find very few, if any, students in the simulation who appear exactly the same. A summary listing of the attributes and their meanings is given in the glossary.

One might ask why we have auxiliary attributes. Why not make all the attributes regular attributes? The main reason is that considerable space savings can be realized by dropping auxiliary attributes as soon as it is determined that a student will fail a course. Another reason is the waste of regular attribute space when one schedules an "event not associated with an entity" (see discussion of events).

2.1.2 Events

The system at hand is characterized by discrete events. Examples of such events are the convenings of classes, a student's graduation or failure, and his arrival at a destination after traveling. If one knows in advance when an event will take place it can be scheduled and encoded on an "event calendar." The SLAM subroutine SCHDL(KEVENT, DTIME, A) schedules an event KEVENT to occur after a delay time DTIME. The argument A is the array holding the regular attributes of the event or entity associated with the event. Subroutine SCHDL loads the current values of the attributes, A, the event code, KEVENT, and time, TNOW+DTIME, onto the calendar file. The calendar is a large file of events which will be processed in chronological order by the SLAM processor. When an event is to be processed, the attributes are loaded into ATRIB and subroutine EVENT is called with the argument KEVENT. Subroutine EVENT then either processes the event or calls another subroutine which processes it. Events scheduled for the same future time will be processed FIFO

unless otherwise specified. (See priority cards in the Data section and Timing Considerations at the end of the Technical Manual.)

Consider two types of events: "Events associated with entities" and "events not associated with entities. Events associated with entities include individual student graduations and failures from courses. The event of a student finishing a course is scheduled when he is entered into the course "activity" (This activity was initiated by a class convening event for lockstep courses or by securing a chair for BEE courses). Since students do not necessarily graduate together, each graduation or failure is scheduled separately. When such an event is scheduled the student's attributes are loaded onto the calendar. He will reside there, forgotten by the rest of the world, until his event comes up. He will then be removed from the calendar and processed. This is called an "event associated with an entity" because the essential information about the student (entity) is stored with the event.

Examples of events not associated with entities include convenings of lockstep classes and report-writing events. These events are scheduled to occur regardless of the existence of entities, and no entity is tied to such an event.

All event and queue entries are stored in NSET/QSET. Each entry has two array elements for regular attributes, an element for an event code and an element for an event time. In the case of events not associated with entities, the spaces for regular attributes is frequently wasted, but need not always be. They can be used to store

arguments to the event-processing subroutines. In the case of class convenings ATRIB(1) is used to hold the course number (see event 6, CLASS).

In the case of entities waiting in files other than the calendar file the array elements of event code and event time will usually be wasted.*

This difference between the file space necessary for the storage of events associated with entities, events not associated with entities without arguments and entities waiting in queues may prompt one to use auxiliary attributes. It should be remembered, though, that every entity in a SLAM simulation will have an entry in NSET/QSET either as an event or a queue entry, regardless of the use of auxiliary attributes.

2.1.3 Resources

In many systems the availability of resources determines how fast or whether something gets done. For example, when you take your car into the shop to have some work done, both a bay and a mechanic must be available before the work can begin. In such a case a simple job may take hours to get done because you must first wait for both a bay and a mechanic to become free. In this example the bay and the mechanic are examples of two different types of resources. When there

*If one places entities in the file and removes them WITHOUT USING THE NETWORK, one MAY be able to use these spaces for some special attributes. This would be an advanced procedure and currently is entirely theoretical.

is competition for the resources entities wait in line for the resources to become available.

In this model BEE courses are unique in that no regular classes convene. The courses are "self-paced." A student works at his own pace reading the text, doing the exercises and taking the tests until he has finished. Before the work can begin, though, a study carrel (chair) must be available. Students of all types wait in queues for chairs for the different BEE courses. Once a student has acquired a chair he keeps it until he either passes or fails the course. Then the chair is made available the next waiting student.

In this program there are 12 different types of resources: chairs for BEE1 at Orlando, chairs for BEE2 at Orlando, chairs for BEE3 at Orlando, chairs for BEE1 at Great Lakes, chairs for BEE2 at Great Lakes, and so forth for three BEE's each at Orlando, Great Lakes, San Diego and Memphis.

It should be obvious by now that it would be very useful to have a general system for handling queues, resources and an event calendar. This is what the SLAM processor does (actually there are many more things SLAM can do). SLAM also makes available many useful statistics. These statistics include average, maximum, minimum and standard deviation of queue lengths and resource utilizations with respect to time, and unweighted average, maximum and standard deviation of any other sample quantities one desires statistics on (time in system in this program). Following is an attempt to describe

the Model and the functions and workings of the subroutines written to simulate it.

When the main program calls subroutine SLAM the SLAM statement cards are read, the SLAM statistical arrays are cleared, the filing system is initialized and subroutine INTLC is called. Events are processed until (1) no more events remain (or entities waiting in queues), or (2) TTFIN has been reached.

SECTION 2.2

DOCUMENTATION OF SUBPROGRAMS

Frequent reference to the program listing, in Appendix B, will be helpful for following the descriptions below.

2.2.1 Subroutine INTLC

Subroutine INTLC contains all user-written* initialization. INTLC reads the simulation specifications cards, student data cards and course data cards and checks them for certain obvious errors. Certain initializations and adjustments are made, and the first events are scheduled. Some initialization was done in the BLOCK DATA section at the beginning of program execution. Counters are defined in Table 1.

A call to the SLAM subroutine SETAA initializes the auxiliary attribute array NAVY. SETAA sets up pointers to the next available spaces throughout NAVY. NSTUD is the number of auxiliary attributes per entity (4). MSTUD is a pointer to the next available space in NAVY (updated every time auxiliary attributes are added to or removed from the array). STUD is a small array into which an entity's auxiliary attributes can be downloaded from NAVY (by SLAM subroutine COPAA or GETAA) or from which the auxiliary attributes are loaded into

*The term, "user-written," is used to distinguish parts of the Model from the SLAM processor, provided by Pritsker and Associates.

NAVY (SLAM subroutine PUTAA). STUD must have dimension equal to NSTUD. NAVY must have dimension equal to NDNVY for the SETAA call.

TABLE 1
PROGRAM COUNTERS

COUNTER	INC	DEC
WTIME(K) contains the sum of all waiting times for course K.	CLASS(K) Events 14-25	
YYY(K) = the number of students to start course K.	CLASS(K) Events 14-25	
YYF(K) = the number of students to successfully complete course K.	TRAVEL	
FLK(K) = the number of failures for course K.	CLASSE(K) TRAVEL	
BACK(K) = the number of students setback in course K.	CLASSE(K) FAILUR	
NCLSZ(K) = the number of students currently under instruction in course K.	CLASS(K) Events 14-25	EVENT
IFAT = number of fatal errors detected in data cards	FAT	
KKK = number of recruit arrivals to the system	RCRT	
IFLER = number of fleet returnee arrivals to the system	FLERN	
MAXAUX = highest auxiliary attribute pointer (It gives a measure of maximum number of concurrent nonfailing students.)	RCRT FLERN	
MAXSYS = highest pointer to next available space in NSET/QSET (A measure of maximum number of concurrent entries.)	RCRT FLERN	
INTRAN = number in transit.	TRTIME FLERN TRAVEL	ROUTE TDST

NST is the file number for the first "A" course (Beyond all RTC and BEE files. NST1 is the number of the last BEE file.

The simulation specifications cards are read.

The student data cards are read. All percentages are converted to decimal fractions. "A" courses in each sequence are counted. This count is stored in NASCHD(NS,11) and is later used to determine whether a student has finished his "A" course sequence. Student yearly demands (YD(NS,i)) are multiplied by the scaling factor (FACTOR). The day-of-week factors (DOWF(NS,i)) are adjusted (see discussion on the driver in subroutine RCRT). The student types are counted.

The course data is read and checked for errors. Setback and failure rates are converted to decimal fractions. The fractions are converted to a cumulative form of distribution to facilitate using a uniform random sample between 0 and 1 later on for determining whether a student will pass, fail or be setback. Maximum and minimum class sizes are adjusted by multiplying by the scaling factor.

The BEE seats were set up as resources with initial capacities (numbers of available resource units) of zero by the SLAM input statements. SLAM subroutine ALTER(IR,IU) alters the capacity of resource IR by IU units. The BEE capacities are set equal to the scaled maximum "class sizes" (MAXS(K), scaled and rounded off to a whole number) entered for the BEE courses.

The quantities ORL, GL, SD, MEM, COR, FLEA, NLON, CHS, MARI, PORH, and GULP are redefined to equal the highest course numbers of courses taught at those places. Array WHR is then defined. WHR(K) holds the base number where course K is taught. Array WHR is used as

a fast way to determine where a student is based on the course he has just finished. This saves using another attribute or having to go through a cascade of IF statements every time a location is needed. Precaution is taken against zero "A" courses being taught at a given base. With some compilers a DO loop will be executed at least once regardless of the values of the indexes.

If no fatal errors have been detected the first convenings of all lockstep courses are scheduled (event 6, CLASS(K) where K = ATRIB(1) specifies the course number); the first reports are scheduled (events 1, 2 and 3); and the first day of recruit arrivals is scheduled (event 4). Each of these events reschedules itself when it is processed.

Subroutine WHERE makes the actual assignments to the array WHR. It seemed more efficient to have one DO loop in a subroutine than to have 11 separate DO loops.

Subroutine FAT counts fatal errors detected and writes error messages. Times between regularly scheduled events (report writings and lockstep class convenings) must be greater than zero. Otherwise the simulation would never get past time zero. It would run endlessly scheduling and processing events zero time units apart (an endless loop situation).

2.2.2 Subroutine EVENT

Subroutine EVENT(J) calls user-defined and scheduled events. Events 1, 2 and 3 print reports. Event 4 (RCRT) generates each day's recruit inputs. Event 5 is not used. Event 6 calls class convenings. ATRIB(1) for event 6 specified the course number. Event 7 (ERRORA) calls a comprehensive error report. Event 8 (ROUTE) puts a student in line for the next BEE course. It is scheduled to follow any travel time directly preceding BEE school and between BEE courses. Event 9 (TDST) puts a student in line for the "A" course in his sequence specified by the ordinal number in auxiliary attribute 3.

Events 14-25 are called from the network. Each of these events represents a student securing a chair for a BEE course. The number-of-starts and under-instruction counters (YYY(K) and NCLSZE(K)) are incremented. Waiting time is collected; the course duration (BEETIM) is determined, and the course completion event is scheduled.

Events numbered greater than LIMCOU (LIMCOU = the file number for the last course modeled or 25, whichever is greater) represent course completions. For all course completion events, the under instruction counter (NCLSZE(K)) is decremented. Those finishing "A" courses are processed by CLASSE(J). Those finishing BEE must free their chairs. All coming out of basic training and BEE are processed by subroutine TRAVEL.

Since the event number is used to determine tiebreaking (see "Timing Considerations") in scheduling, one must consider this aspect when changing event numbers or introducing new events.

2.2.3 Subroutine RCRT

Subroutine RCRT generates new recruit arrivals to the three recruit training commands (Orlando, Great Lakes and San Diego) and puts them in line for basic training.

SLAM subroutine PUTAA puts the auxiliary attributes in array STUD into NAVY and updates the next-available-space pointer MSTUD. MSTUD must be saved before the attributes are put away. SLAM subroutine FILEM(K, ATRIB) puts a student with attributes ATRIB into queue K.

For each type of student data must be supplied to tell how many come per year, to what base they report and when they come. The daily input for a type NS student is computed from the formula:

$$\text{daily input at base L for student type NS} = YD(NS,K) * FM(NS,M) * DOWF(NS,ND) * BF(NS,L)$$

where $YD(NS,K)$ is the total yearly input to the Navy (all bases) of type NS students for year K; $FM(NS,M)$ is the fraction of all type NS students to arrive for the year who report during month M; $DOWF(NS,ND)$ is the fraction of all type NS students who report on the ND'th day of the week (the portion reporting over the weekend should be added to Friday's share); and $BF(NS,L)$ is the fraction who report to base L.* The month factors should average 1/12 and add to 1.00 for any single student type. The base factors should average 1/3. The

*When this report was written available data indicated clear peaks of input on Wednesdays and during the summer months.

day-of-week factors were input to average $1/5$ (according to the definition); however, since there are more than 5 days in a month, $1/5$ of the monthly input ($YD(NS,K)*FM(NS,M)$) does not occur every day. Some adjustment must be made if the data just defined is to be compatible with the formula above. Assuming a year has $52 \times 5 = 260$ working days, the average month has $260/12$ working days. It was necessary in INTLC to adjust the day-of-week factors (DOWF) from values averaging $1/5$ to values averaging $12/260$.

$$\begin{array}{ll} \text{IF} & 12/260 = X \cdot 1/5 \\ \text{THEN} & X = (12/260)(5/1) = 3/13. \end{array}$$

All DOWF data were therefore multiplied by $3/13$. DOWF data averaging $1/5$ was called for as input because it is easier to grasp the meaning of the $1/5$ average than the $12/260$ average.

Randomizer

One might at first think it appropriate to every day go through each base and student type in order, generate the number of students of that type to arrive that day at that base, and put them in the queue. A hidden flaw to this procedure is that, even though the entire day's input occurs at the same time and all student types may have the same priorities, the first student types (according to the order in the student data cards) will be put in the queue ahead of the others. They will be removed from the queue in the same order, and graduations will be scheduled in that order. After graduations they

will get in queues for their next courses in the order of graduation. The only perturbations to this order will be due to setbacks, different travel times, different BEE times, and different course sequences.

The result is that of those types with the same priority specifications, the student types whose cards come first in the data are given a head start over those who come after them.

To eliminate the prioritizing effect of the order of the student data cards, subroutine RCRT has a "randomizer." Each day, for each base, subroutine RCRT first determines how many recruits of each type will arrive. These new recruits are then randomly ordered in a scratch file (NCLNR-1). After all students for the base for that day have been generated and randomized they are transferred to the appropriate RTC queues.

Subroutine RCRT keeps track of the maximum number of concurrent entries (all students and scheduled events) by monitoring the pointer, MFA, to the next available space in NSET/QSET. The maximum value of MFA is held in variable MAXSYS. The maximum number of concurrent nonfailing students is monitored through the pointer, MSTUD, to the next available space in NAVY. The maximum value of this pointer is stored in MAXAUX. Actually MAXSYS will be 6 times the maximum number of entries, and MAXAUX will be 4 times the maximum number of concurrent nonfailing students. The maximum number of students will be more than that implied by MAXAUX because the auxiliary attributes of failures are dropped when they start the course they are going to

fail. For head-counting reasons, and because in BEE a student occupies a chair until he fails (preventing others from using it), the student is not actually dropped from the simulation until he gets out of the course. These numbers will be printed in reports to help one set reasonable sizes for NSET/QSET and NAVY.

TABLE 2
CPU TIME (SECONDS)

SIMULATED TIME (days)	SCALING FACTOR			
	1/20	1/10	1/5	1/3
50	10.7	10.72	16.69	25.51
100	18.54	28.99	68.96	153.29
150	29.59	59.96	160.63	529.40
200	46.97	108.47	322.28	1614.20
250	68.55	170.80	524.85	3454.10
300	88.63	227.79	703.64	5515.63
350	110.03	283.58	879.66	7687.70
400	130.22	340.19	1026.98	9945.36
450	153.94	406.71	1216.84	12789.34
500	181.89	482.70	1435.24	16618.83

The WRITE statement which goes with FORMAT 9000 is useful for monitoring the day-to-day ebb and flow of population. When population is maximum, it will be just after the creation of a new student(s) -- either at the end of a recruit generation event, or after the cloning of a basic training graduate. Recall that ATRIB(1) holds the pointer to where the auxiliary attributes were stored in NAVY. NAVY fills from the bottom up with vacated spaces being filled before going higher in the array. Hence, the maximum value of ATRIB(1) in this listing will be MAXAUX and a measure of maximum number of concurrent

students (not counting failures). If the output is directed to the screen of a CRT while the program is being executed, one can visually monitor the processing time for a simulated day's events as the system fills and drains. This data may be useful for predicting run times for long simulations with many students. A tabulation of simulated time versus processing time for a typical run is given in Table 2. Table 2 was taken from data generated using a call to a system subroutine which returned CPU time.

2.2.4 Subroutine TRAVEL(J)

All students leaving BEE and basic training courses are processed by subroutine TRAVEL. The argument, J, represents the number of the course just finished (passed or failed). TRAVEL is called from events 101-115. A flowchart of TRAVEL is shown in Figure 2. TRAVEL will first check to see whether the student passed his course.

If a student is a failure his ATRIB(1) will be negative. Failures are counted and dropped from the system. Returning from subroutine TRAVEL without refiling the student, either in a queue or as a scheduled event, will cause the student to be lost from the simulation. It is important to remove the auxiliary attributes from NAVY before dropping the student. His auxiliary attributes were pulled when he started the course when it was determined he would fail (subroutine FAILUR).

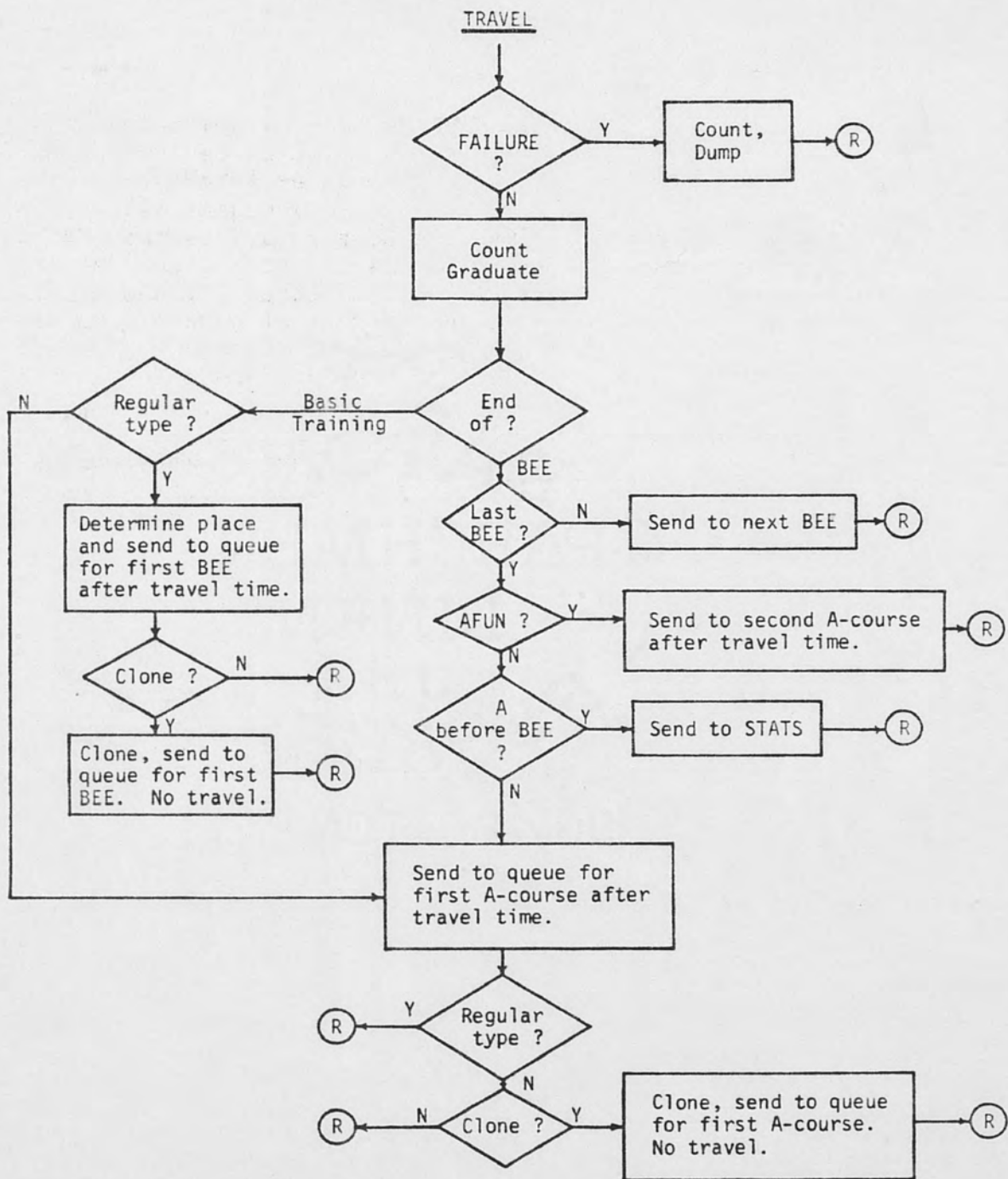


Figure 2. Flowchart of subroutine TRAVEL.

If the student has successfully completed the course (ATTRIB (1) is not negative) the graduate counter for that course (YYF(J)) is incremented. Class size, NCLSZE(J), was already decremented in subroutine EVENT. His auxiliary attributes are then accessed and his type (NS) is determined. The argument J tells what course he has just finished. If $J \leq 3$ he has just finished a basic training course. If $J > 3$ he has just finished a BEE course.

The following outline structure is useful for describing the organization of subroutine TRAVEL.

A. A Basic training graduate will either go to his first BEE course (regular type) or first "'A' course" (A-before-BEE and AFUN types). A-before-BEE types are flagged by $BE(NS) < 0$. AFUN types are distinguished by the second part of their name (STYP2(NS)) being 'AFUN'.

1. For a regular type we next determine to what base he will go for BEE. It should be one which teaches the courses he needs, and it should be reasonably close. This information is supplied in the data and stored in array NROUTE. $NROUTE(NS, J)$ is the base to which a type NS student will go for BEE if he is at base J when he is ready, in his training, to take BEE. For basic training courses the course number is the same as the number of the base where it is taught. The student will be scheduled to arrive at the next place after a travel time $TRTIME(K, KBEE)$, where K is the current base and KBEE is the base where he is going for BEE. At that time he will be processed by event 8, ROUTE. ROUTE sends the student to the queue for the NEXT course

(consecutively according to course numbers). At each base K, the BEE courses are numbered $K*3 + 1$, $K*3 + 2$, $K*3 + 3$ (4,5,6 at Orlando; 7,8,9 at Great Lakes; and 10,11,12 at San Diego). In ROUTE auxiliary attribute 3 is checked to see what BEE course the student has just taken. When sending him there for his first BEE course auxiliary attribute 3 is loaded with $K*3 + 0$ (as if a course number for a BEE0 course) to trick subroutine ROUTE. All ROUTE does is add 1 to this number and send the student to that queue.

Back in TRAVEL, setting the student up for his first BEE course and scheduling ROUTE, function DRAND is used to provide a random number (DRAND(3) provides a random number from random number stream 3) between 0.0 and 1.0 for determining whether to clone the student to simulate a fleet returnee entering the system. FLER is the fraction of basic training graduates which will be cloned. Subroutine FLERN creates the clone with his own set of auxiliary attributes and a higher priority for the BEE queues. FLERN then sends the clone the same way as the recruit graduate.

2. If a student has just finished basic training and he is an A-before-BEE type or an AFUN type his next course will be the first in his "A" course sequence (NASCHD(NS,1)). The AFUN course before BEE for AFUN students is treated as the first "A" course. (The actual "A" courses make up the rest of the NASCHD sequence.) Before getting in the queue for this first lockstep course a student must first be at the base where it is taught. That base is determined (WHR(NASCHD(NS,1))) at the statement labeled 11 in TRAVEL. Statement

11 heads an all-purpose section for sending a student to his first "A" course. Regular type students come this way after finishing their BEE sequences. For those just finishing basic training the base number is J. For those finishing BEE the base number is not J; it is $(J-1)/3$. In both cases, before coming to statement 11 the base number is determined and assigned to variable K. Travel time between where he is and where he is going is obtained. Event 9 (TDST) is scheduled to occur after this time (DELAY). TDST then will put the student in the queue for the "A" course in his sequence as indicated by the ordinal number stored in auxiliary attribute 3.

The A-before-BEE and AFUN types are just finishing basic training at this point. A fraction FLER of them will be cloned to simulate fleet returnee inputs.

One might wonder why all the cloning could not have been done by a single call to FLERN instead of at the two places in TRAVEL. Cloning in subroutine EVENT was not viable because the clones must not be counted as basic training graduates, and only graduates should be cloned. Also, fleet returnees are modeled to enter the system upon arrival to BEE or A school -- not before travel to that place. FLERN sends fleet returnees directly to the queues without any travel delay.

B. If a student has just finished a BEE course we then determine where he is (what base) and which BEE course he has just finished (1st, 2nd or 3rd). We then check to see if this is his last BEE course. Has he taken all the BEE's he is supposed to? $IABS(BE(NS))$ indicates how many BEE courses type NS students are to

take.

1. If it is not his last BEE course he is routed to the next one (ROUTE). It is assumed that each student takes all his BEE courses at the same base.

2. If it was his last BEE course he will take one of three routes based on his type:

a. An A-before-BEE type has finished his training pipeline at this point. He will be sent to subroutine STATS, where his auxiliary attributes will be pulled; his total time in the system will be determined, and he will be dropped from the system.

b. An AFUN type is ready to start the lockstep courses (true "A" courses) following BEE, entered in the data as all "A" courses following the first. First there is a check to make certain that there are more courses in the "A" course list ($NASCHD(NS,11) > 1$). If so, he is scheduled after whatever travel DELAY for TDST, which puts him in the queue for the "A" course specified by the ordinal number in auxiliary attribute 3. If no more courses are indicated, then he has finished his pipeline, just as the A-before-BEE types have.

c. A regular type student finishing BEE is routed to his first "A" course (at statement 11). Note that only AFUN and A-before-BEE types can be cloned at this point.

Between schools on the same base we can model either no delay or some very small delay (see function TRTIME). There may be some delay associated with changing commands.

2.2.4.1 Function TRTIME

Function TRTIME(I,J) returns the travel time from place I to place J. If I = J then it is the delay time between graduation from one school and getting in the first queue in another school at the same base. For changes of base 85% of all students are assumed to fly, making the travel times 1 day in these cases. It is assumed that all other changes of station are made via privately-owned vehicle (POV). POV travel times are listed in Table 3. The basic rule is 1 day for every 300 miles, plus one additional day only if the remainder, after dividing by 300, is greater than 150 miles. For the first day the minimum distance is only 50 miles.

TABLE 3
AUTHORIZED TRAVEL TIME VIA POV (ACTUAL DAYS)

ORLANDO	0	4	8	3	2	8	4	1	9	8	2
GRT LKS	4	0	7	2	3	7	3	3	7	7	3
SANDIEGO	8	7	0	6	7	0	10	8	2	1	6
MEMPHIS	3	2	6	0	1	6	4	2	7	6	1
CORRY	2	3	7	1	0	7	4	2	9	7	1
FLEASWTR	8	7	0	6	7	0	10	8	2	1	6
N LONDON	4	3	10	4	4	10	0	3	10	10	5
CHASTON	1	3	8	2	2	8	3	0	9	8	2
MARISLND	9	7	2	7	9	2	10	9	0	1	8
PORHUEN	8	7	1	6	7	1	10	8	1	0	7
GULFPORT	2	3	6	1	1	6	5	2	8	7	0
	ORL	GL	SD	MEM	COR	FLEA	NLON	CHAS	MARI	PORH	GULP

SOURCE: U.S. Department of Defense, Official Table of Distances, Continental United States, Alaska, Hawaii, Canada, Canal Zone, Central America, Mexico, Puerto Rico, Army Regulation No. 55-60 / Navy Publication P-2471 / Air Force Regulation No. 177-135 (Washington D. C.: 1 December 1979).

WEEKEND REMOVAL: Since the timestep of the simulation is the working day, and travel will take place on weekends, a routine is necessary for removing weekend days from TRTIME so that it will only reflect the number of working days of travel. First the number of working days remaining in the week (WOD) is determined:

$$WOD = 5.0 - TNOW + 5 * INT(TNOW/5)$$

This could come out to 5 either at the very beginning or very end of the work week. Travel time is determined at the ends of courses. Since it is far more likely that a course would end at the end of the week than at the beginning, whenever $WOD = 5$, it is reset to zero.

Now, if the originally-computed travel time is shorter than the remaining workdays, it is left alone. If it is longer than the remaining workdays but does not extend past the weekend, it is truncated to the number of remaining workdays. If it extends past the weekend, 2 days are subtracted from TRTIME, and the 5 workdays for the next week are added to WOD. The tests are repeated. A flowchart of the routine is shown in Figure 3.

The transient counter, INTRAN, is incremented in TRTIME and FLERN.

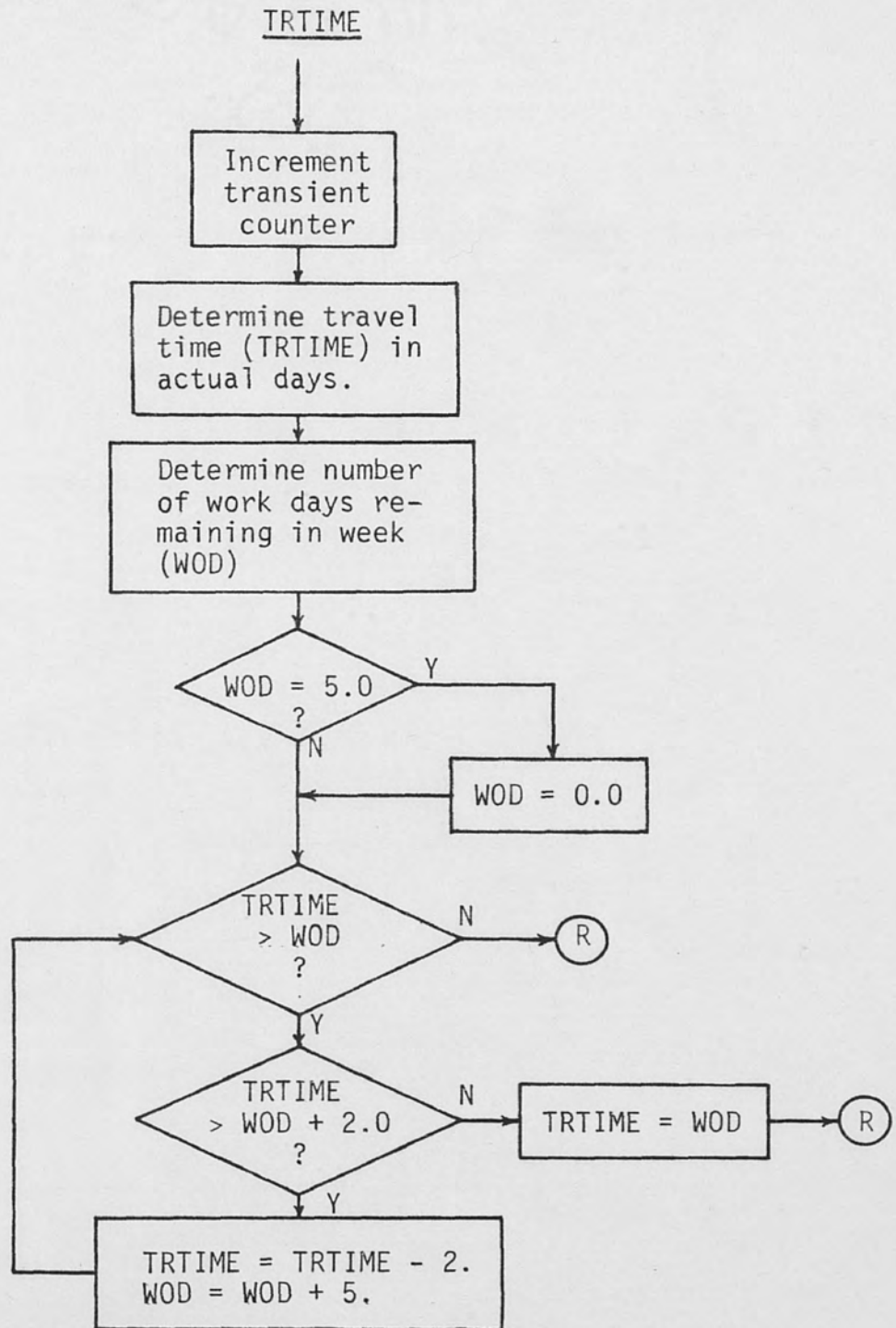


Figure 3. Flowchart of function TRTIME.

2.2.4.2 Subroutine FLERN(J)

Subroutine FLERN clones basic training graduates to simulate fleet returnees entering the system. The recruit's auxiliary attributes are copied. Auxiliary attribute 1 (STUD(1)) is reset to the current simulation time, TNOW. The fleet returnee's priority in BEE queues is higher than that of a regular recruit, so ATTRIB(2) is incremented by 100. The pointer, MSTUD, to where the auxiliary attributes will be placed in NAVY must be saved in ATTRIB(1) before the auxiliary attributes are put away (SLAM subroutine PUTAA). The transient and fleet returnee counters (INTRAN and IFLERN) are incremented, and MAXAUX and MAXSYS are adjusted if necessary. The new arrival is sent to the queue for his first BEE (ROUTE) or "A" course (TDST) according to the calling argument, J.

2.2.5 Subroutine ROUTE

Subroutine ROUTE sends a student to the queue for the next BEE course. This queue is determined by adding 1 to auxiliary attribute 3. SLAM subroutine ENTER effects the immediate transfer of the entity from a user-written subrouine to the indicated ENTER node in the network portion of the model.

INTRAN is decremented.

2.2.6 Network Portion: BEE Chair Acquisition

To leave a BEE queue a student must get a chair (study carrel) for his BEE course. The queues waiting for the resources are modeled by the "AWAIT" SLAM statements. A student entity gets into the network and to the AWAIT node through an ENTER node. (Use of SLAM subroutine ENTER and ENTER nodes assures the processing of network nodes as "current events" -- see Timing Considerations.) A student will not leave an AWAIT queue until the indicated resource has become available and is assigned to him. At this time an event (14-25) will be called which determines pass/failure and schedules course completion event. The same event collects waiting times and increments number-of-starts (YYY(K)) and under-instruction (NCLSZE(K)) counters.

The network portion is described by the SLAM statement cards in the data deck between the "NETWORK;" and "END;" cards.

2.2.7 Subroutine TDST

Subroutine TDST puts a student in line for the "A" course in his sequence indicated by its ordinal number, found in auxiliary attribute 3 (NAVY(NGET+3)).

INTRAN is decremented.

2.2.8 Subroutine CLASS(K)

Subroutine CLASS(K) handles the opening of a class for course K. First the next convening date is scheduled.

CLASS next checks the queue for the course. If the number of students waiting is equal to or greater than the minimum number of students allowed for a class for that course, a class is formed with up to the maximum number of students allowed. The students are removed from the queue. For each student, whether he will pass, fail or be setback is determined, and his graduation or failure event is scheduled (Event K+LIMCOU, CLASSE(K)).

For basic training the time between class convenings (OPD(K)) is 1 day. Since it is unlikely that one would be setback for only one day, the setback delay is arbitrated at 5 days for basic training.

Queue waiting times are collected. The number-of-starts (YYY(K)), and under-instruction (NCLSZE(K)) counters are incremented. If it is a basic training class, more companies are formed until less than the minimum number required to form a company remain in the queue.

2.2.9 Subroutine CLASSE(K)

Subroutine CLASSE(K) processes a student finishing "A" course K. If he is a failure he is counted and dropped. If he has passed, his next destination must be determined. The ordinal number of the course he just finished is stored in auxiliary attribute 3 (NAVY(NGET+3)). The total number of "A" courses in his sequence is

stored in NASCHD(NS,11).

1. If it was the first in his "A" course sequence (NASCHD) AND he is an AFUN type, he will go to BEE school next (statement 20). The same procedure as was used in subroutine TRAVEL is used to set him up for BEE school.

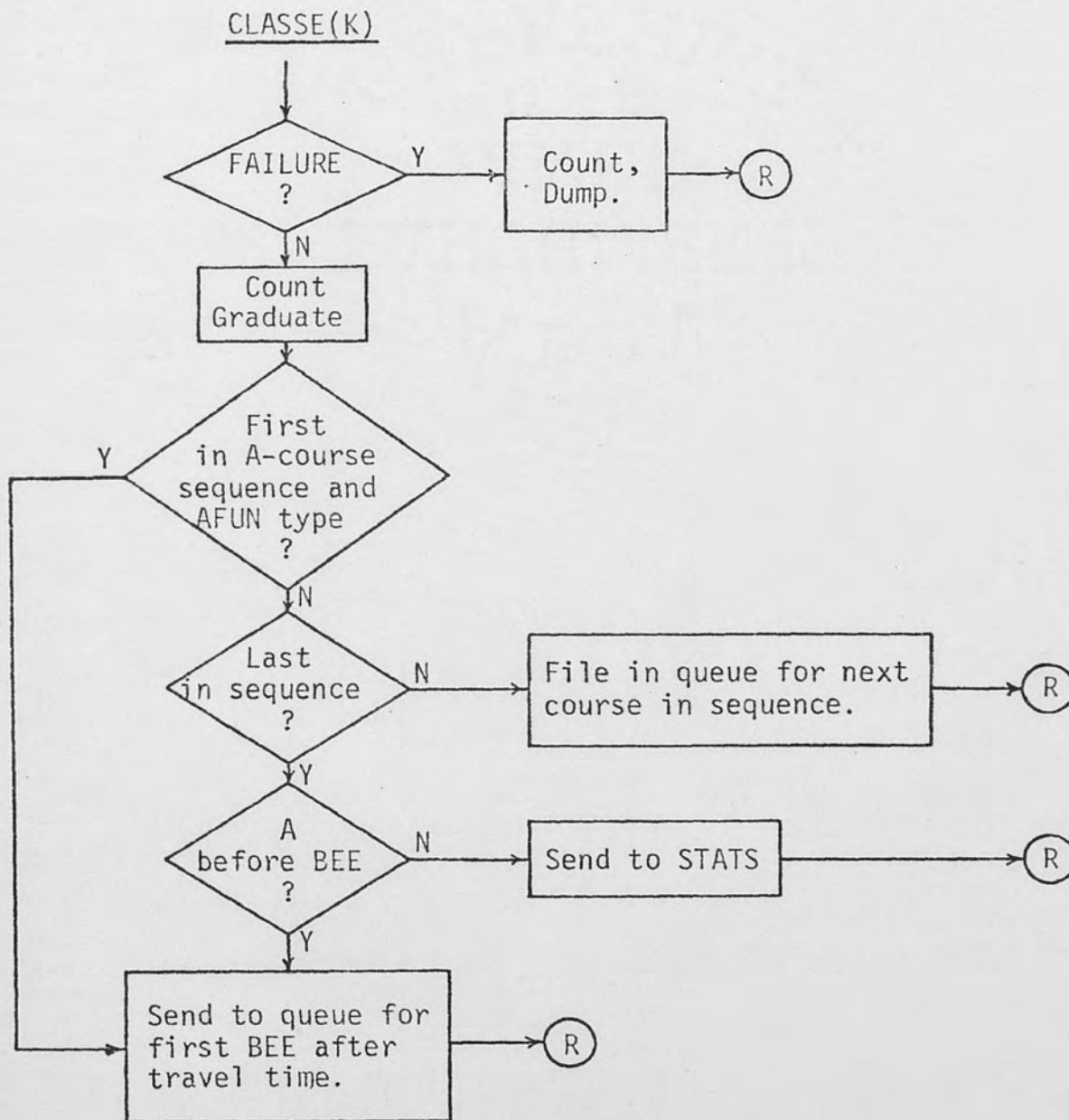


Figure 4. Flowchart of subroutine CLASSE.

2. If it was the last in his "A" course sequence AND:
 - a. he is an A-before-BEE type, then he goes next to BEE school (statement 20).
 - b. he is a regular type, then he is finished (sent to STATS).
3. If he is in none of the above categories he is placed in the queue for the next "A" course in his sequence. Auxiliary attribute 3 (NAVY(NGET+3)) is adjusted accordingly.

Under-instruction counters (NCLSZE(K)) were adjusted in subroutine EVENT. A flowchart of CLASSE is shown in Figure 4.

2.2.10 Subroutine STATS

Subroutine STATS collects time-in-system statistics by student type. SLAM subroutine COLCT(TSYS,NS) accepts an observation, TSYS, of type, NS. The SLAM processor will keep track of the average, minimum, maximum, and standard deviation of each type of observation. The student is then destroyed. SLAM subroutine GETAA removes auxiliary attributes from NAVY and updates the next-available-space pointer (MSTUD).

2.2.11 Subroutine FAILUR(K, RN)

Subroutine FAILUR is called whenever a student begins any course (K). It determines whether he will pass, fail or be setback and returns this information via argument RN. If the student is to fail, his auxiliary attributes are removed (SLAM subroutine GETAA) and regular attribute 1 (ATRIB(1)) is set equal to -K. Any operations requiring auxiliary attributes must be completed before calling

FAILUR.

Setbacks are counted (BACK(K)).

2.2.12 Subroutine REPORT

Subroutine REPORT first reschedules itself TBR time units hence. All calendar and queue entries are apparently made by SLAM subroutine FFILE. If one inserts "CALL REP2" at statement 10 in SLAM subroutine FFILE (see Appendix G) a Model report will be printed before aborting in the event of student overflow (to many entries for NSET/QSET). It is important that "ENTRY REP2" come after the SCHDL call to avoid an endless loop situation in the event of an overflow.

FFAVG(K) returns the average length of queue K. FFSTD(K) returns the standard deviation of the length of queue K. MMAXQ(K) returns the maximum length of queue K. NQ(K) is the current length of queue K. All queue length information and student counts are descaled. Failure and setback rates were converted to decimal fractions in INTLC. The percentages are computed and printed in the course data section. The student type data section should be transparent. SLAM subroutine PRNTH(0) prints all histograms. Sample outputs are given in section 2.3.4 and in appencix C.

2.2.13 Subroutine STATUS

Subroutine STATUS describes the current state of the system. Students currently under instruction are located by searching the calendar file for course completion events. Students awaiting instruction are found in the respective queues. Students in transit

will be destined for event 8 or 9, for those going to BEE schools and A-schools, respectively.

SLAM function MMFE(K) returns the pointer in NSET/QSET to the first entry in file (queue) K. It will be zero if there are no entries. NSUCR(L) returns the pointer to the successor of the entry with pointer L. If there is no successor NSUCR returns a zero. NCLNR is the SLAM variable for the calendar file number. PLACE(12) = 'BEE' and PLACE(13) = 'ASCH'. These elements of PLACE are used only here to designate the transient's destination school. 'ASCH' for AFUN types may indicate either the AFUN course before BEE, or the actual A-school. (Since AFUN, BEE and "A" school are all taught at Memphis for AFUN students, "ASCH" is most likely to indicate travel to AFUN.)

2.2.14 Subroutine ERRORA

Subroutine ERRORA is called for various error conditions detected during execution. The argument, IERROR, designates where the error was detected and where the call was made. The array, LABEL, indicates where these places are in the program.

Adjusted quantities such as fail, setback rates, DOWF and scaled quantities are not deadjusted in ERRORA. The general idea is to describe things exactly as they were at the time the error was detected. The statements after CALL REP2 and CALL SUMRY print the calendar file and the integer and real versions of NSET/QSET. These reports can be very long.

At the end of the error report is a deliberate FORTRAN error (SQRT(-1.)) to provoke system traceback routines.

2.2.15 Subroutine PRECHK

Subroutine PRECHK is called if IECHO \neq 0 on simulation specifications card 1. PRECHK performs a simple preliminary analysis of the data to search for any obvious potential bottlenecks.

For each student type PRECHK determines the average yearly input expected. Then, using course sequences, failure rates and the fleet returnee input fraction (FLER), PRECHK determines the average yearly feed rate of that type of student to each course. The array WTIME, which has not yet been used to store any waiting times, is used for a temporary scratch file to hold these feed rates to each course. This facilitates adding up the total feed rates (of all types of students) for mixed courses, such as basic training, BEE and one "A" course (CDP 8563, taken by GSEs and GSMs).

For each course the maximum yearly capacity (students per year) is determined, using maximum class size and the number of classes held per year (course duration for BEE). The feed rate and capacity are compared and obvious bottlenecks are noted in the output.

2.2.15.1 Subroutine BEECHK

Subroutine BEECHK(NS,L,X30) performs the analysis for student type NS, with feed rate of X30 to BEE school, through all BEE courses slated for that type at base L.

2.2.15.2 Subroutine ASCHK

Subroutine ASCHK(NS,NA,X) performs the analysis for student type NS with current feed rate of X, through the NAth course in his "A" sequence.

The exact procedure is probably best described using an example:

Suppose a given student type is to take two "A" courses, Course A followed by course B. For simplicity let us first assume that this student type is the only type which takes these courses and that 1000 students per year reach course A. Course A convenes 50 times per year with a maximum class size of 10 students. Course B convenes 50 times per year with a maximum class size of 12 students. Both courses have failure rates of 20%. Setback rates are ignored since all setbacks are assumed to eventually pass, and they are not considered in the model when filling classes.

If 1000 students per year start course A and 20% fail, then course B will be fed at the rate of 800 students per year. If 800 per year started course B and 20% failed then 640 per year would finish the A-B sequence. However, course A has a maximum yearly capacity of 500 students and course B a maximum capacity of 600. The true maximum feed rate of course B would be the lesser of: 1) the feed rate to A times the graduation rate ($1000 \times 0.80 = 800$); or 2) the maximum capacity of A times the graduation rate; ($500 \times 0.80 = 400$ students per year).

Under the stated conditions course A will represent a bottleneck and course B will not, but only because of the restricted flow from course A. The "potential bottleneck" at B is said to be "masked" by the "visible bottleneck" at A. When the maximum class size or convening frequency of A is increased to accommodate more than $600/0.80 = 750$ students per year (feed), its graduate frequency ($750 \times 0.80 = 600$ students per year) will exceed B's capacity and B will become a visible bottleneck.

Subroutine PRECHK calculates the student flow rate coming out of one course and feeding the next based only on the feed rate and failure rate. It does not consider the restriction a course capacity might impose on student flow. This way the routine spots all POTENTIAL bottlenecks for a given data set. One will not be increasing the capacity of one class only to later discover the downstream bottlenecks previously masked by the upstream restriction(s). The figures listed under "FEED RATE" in the predata analysis report give the minimum course capacities required to accommodate student input based on given failure rates. It is not recommended that course capacities be set at these numbers because, due to the random nature of student input (the average yearly input rate is not uniform throughout the year), long queues will develop during peak seasons.

Another effect of calculating student flow rate this way is that the student output rate, listed under "OUT" in the student section, is optimistic. It is reasonable only if student flow is not

restricted anywhere by course capacity (no bottlenecks).

SECTION 2.3

STUDENT FLOW PATHS

Figure 5 is presented as an aid to following student flow through the program. Decisions made in subroutines TRAVEL and CLASSE are based on the student type and the point in his training pipeline.

2.3.1 Regular Type Students

A regular type student is generated in RCRT and placed in an RTC queue. He will remain in the queue until removed by a class convening event (CLASS). At that time it is determined whether he will pass, fail or be setback (FAILUR), and his completion of basic training is scheduled (event 101, 102 or 103 -- TRAVEL). When the event occurs, subroutine EVENT will send him to TRAVEL. In TRAVEL, if he is a failure he is counted and dropped. If he passed, TRAVEL determines travel time (TRTIME) from where he is to where his BEE school is and sends him there to take BEE1. SCHDL puts his arrival at BEE school on the calendar. When he arrives (event 8) he is put in the queue (ROUTE). Students in BEE queues wait for BEE chairs to become available (SLAM AWAIT node in the network portion). When the student gets his chair (one of events 14-25), his course prognosis is determined (FAILUR) and his course completion/failure event is scheduled (one of events 104-115). Upon passing or failing BEE the student frees his chair (SLAM subroutine FREE) and he is sent to

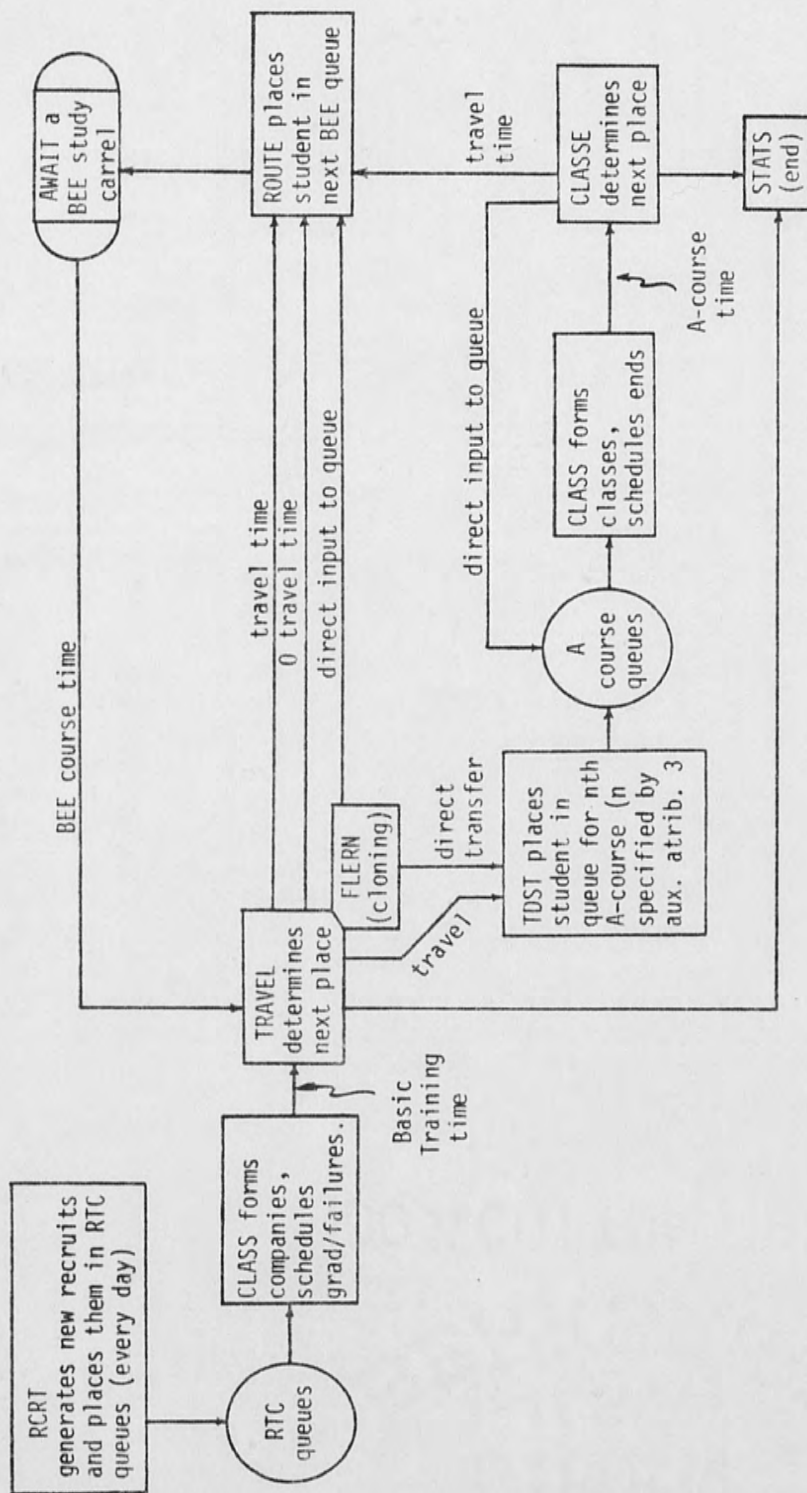


Figure 5. Student paths through the Model.

TRAVEL. He continues to take successive BEE courses until he has completed his BEE requirements. Then TRAVEL determines travel time and sends him to the queue for his first "A" course (TDST). (Actual place is not really modeled. The program simply keeps track of where the various courses are taught and models delays between schools at different places.) When he arrives at "A" school (Event 9), he is put in the queue for the first course in his "A" course sequence (TDST). He remains in the queue until removed by a class-convening event (CLASS). When he is put into a class, his prognosis is determined (FAILUR) and his graduation or failure event is scheduled (Event LIMCOU+K, CLASSE). Upon completion of an "A" course he is put into the queue for the next "A" course the sequence, if there are more, or sent to STATS, if it was the last one. In STATS time-in-system statistics are taken and he is dropped from the system.

2.3.2 A-Before-BEE Students

An A-before-BEE type student flows like a regular type except:

- a. After basic training TRAVEL sends him to A-school (Event 9, TDST, auxiliary attribute 3 = 1.).
- b. After A-school CLASSE sends him to BEE school (Event 8, ROUTE).
- c. After BEE school TRAVEL sends him to STATS.

2.3.3 AFUN Students

An AFUN student type flows like the A-before-BEE type except:

a. After finishing his first "A" course (AFUN), CLASSE sends him to BEE school.

b. After finishing BEE school TRAVEL sends him to A-school for the second course in his "A" sequence (Event 9, TDST, auxiliary attribute 3 = 2.)

c. After finishing his "A" course sequence CLASSE sends him to STATS.

2.3.4 Examples

If one models only one type of student at a time, it is possible to verify the student flow path from the output.* Samples of such outputs are presented and described below:

2.3.4.1 Regular type students

An output with an echo-print of data and a preliminary data analysis is shown in Figure 6 for a regular type student. "A" courses are only taught at the Memphis RTC for this example.; The scaling factor (FACTOR) is 1.0, which means that one student in the simulation represents one in real life. If the scaling factor had been other than 1, there would still have been a 1:1 correlation in the reports, but only a scaled number of students would actually have been

*Much of this section has also been included in "A Pipeline Analysis of US Navy 'A' School Training Using SLAM," an article written for the Journal of Computers in Industrial Engineering.

ORLANDO = 0
GREAT LAKES = 0
SAN DIEGO = 0
MEMPHIS = 5
CORY STATION =
FLEASWTRAC = 0
NEW LONDON = 0
CHARLESTON = 0
MARK ISLAND = 0
PORT HUENEME = 0
GULF PORT = 0

```

ICASE = 0
SCALING FACTOR = 1.00
TIME UNTIL FIRST REPORT = 520.0
TIME BETWEEN REPORTS = 50.0
TIME UNTIL FIRST SHAPSHOT = 99100.0
TIME BETWEEN SHAPSHOTS = 99999.0
FLUET RETURNS = 10.000

```

[illegible]

Negative "BE" indicates that A school comes before BE.

	TOTAL YEARLY DEMAND	BASE FACTORS	MONTH FACTORS
1	100	100	100
2	100	100	100
3	100	100	100
4	100	100	100
5	100	100	100
6	100	100	100
7	100	100	100
8	100	100	100
9	100	100	100
10	100	100	100
11	100	100	100
12	100	100	100
13	100	100	100
14	100	100	100
15	100	100	100
16	100	100	100
17	100	100	100
18	100	100	100
19	100	100	100
20	100	100	100
21	100	100	100
22	100	100	100
23	100	100	100
24	100	100	100
25	100	100	100
26	100	100	100
27	100	100	100
28	100	100	100
29	100	100	100
30	100	100	100
31	100	100	100
32	100	100	100
33	100	100	100
34	100	100	100
35	100	100	100
36	100	100	100
37	100	100	100
38	100	100	100
39	100	100	100
40	100	100	100
41	100	100	100
42	100	100	100
43	100	100	100
44	100	100	100
45	100	100	100
46	100	100	100
47	100	100	100
48	100	100	100
49	100	100	100
50	100	100	100
51	100	100	100
52	100	100	100
53	100	100	100
54	100	100	100
55	100	100	100
56	100	100	100
57	100	100	100
58	100	100	100
59	100	100	100
60	100	100	100
61	100	100	100
62	100	100	100
63	100	100	100
64	100	100	100
65	100	100	100
66	100	100	100
67	100	100	100
68	100	100	100
69	100	100	100
70	100	100	100
71	100	100	100
72	100	100	100
73	100	100	100
74	100	100	100
75	100	100	100
76	100	100	100
77	100	100	100
78	100	100	100
79	100	100	100
80	100	100	100
81	100	100	100
82	100	100	100
83	100	100	100
84	100	100	100
85	100	100	100
86	100	100	100
87	100	100	100
88	100	100	100
89	100	100	100
90	100	100	100
91	100	100	100
92	100	100	100
93	100	100	100
94	100	100	100
95	100	100	100
96	100	100	100
97	100		

TOTAL YEARLY DEMAND	BASE FACTORS	MONTH FACTORS	DAY-OF-WEEK FACTORS
1,000, 1000, 1000, 1000, 1000,	0.59 0.28 0.13	0.09 0.07 0.07 0.06 0.07 0.11 0.10 0.11 0.09 0.07 0.06	0.18 0.20 0.24 0.20 0.18

		COURSE DATA:																	
1	2	CNAME	FAIL BACK		MIN	MAX	LENGTH	FRST	DET										
			(%)	(%)															
1	1	RTC ORL	10.00	10.00	5.00	5.00	40.0	0.0	1.0										
2	2	ATC GL	10.00	10.00	5.00	5.00	40.0	0.0	1.0										
3	3	RTC SD	10.00	10.00	5.00	5.00	40.0	0.0	1.0										
4	4	BEL1 ORL	10.00	10.00	0.00	53.00	25.0	0.0	0.0										
5	5	BEL2 ORL	10.00	10.00	0.00	53.00	25.0	0.0	0.0										
6	6	BEL3 ORL	10.00	10.00	0.00	53.00	25.0	0.0	0.0										
7	7	BEL1 GL	10.00	10.00	0.00	53.00	25.0	0.0	0.0										
8	8	BEL2 GL	10.00	10.00	0.00	53.00	25.0	0.0	0.0										
9	9	BEL3 GL	10.00	10.00	0.00	53.00	25.0	0.0	0.0										
10	10	BEL1 SD	10.00	10.00	0.00	53.00	25.0	0.0	0.0										
11	11	BEL2 SD	10.00	10.00	0.00	53.00	25.0	0.0	0.0										
12	12	BEL3 SD	10.00	10.00	0.00	53.00	25.0	0.0	0.0										
13	13	BEL1 MEM	10.00	10.00	0.00	53.00	25.0	0.0	0.0										
14	14	BEL2 MEM	10.00	10.00	0.00	53.00	25.0	0.0	0.0										
15	15	BEL3 MEM	10.00	10.00	0.00	53.00	25.0	0.0	0.0										
16	16	CDP 6093	3.00	2.00	1.00	22.00	29.0	0.0	10.0										
17	17	(UNUSCD)	100.00	0.00	0.00	0.00	99999.0	99999.0	99999.0										
18	18	CDP 6070	5.00	40.00	45.00	47.00	33.0	0.0	5.0										
19	19	CDP 605X	0.00	0.00	5.00	10.00	33.0	0.0	5.0										
20	20	CDP 603V	15.00	40.00	5.00	10.00	40.0	0.0	5.0										

Figure 6(a). Sample output for one rating of regular type students.

PHR-EXECUTION DATA ANALYSIS: Fleet returns = 10.00% of basic training graduates.

STEADY STATE SIMPLIFIED RATES (if no bottlenecks)
STUDENT TYPE In Out (students per year)
1 SAAP ke 1000.0 647.5

COURSE MAXIMUM CAPACITIES AND FEED RATES (assuming no previous bottlenecks restricting feed):

COURSE	CAPACITY	FEED RATE	W/FALL	GMAN RATE	(students per year)
1 R1C OHL UNLIMITED	587.0	10.0	529.3		
2 R1C GL UNLIMITED	279.0	10.0	251.1		
3 R1C SD UNLIMITED	134.0	10.0	120.6		
4 R1E1 OHL UNLIMITED	581.1	10.0	496.1	****	
5 R1E2 OHL UNLIMITED	521.0	10.0	470.7		
6 R1E3 OHL UNLIMITED	364.0	0.0	0.0		
7 R1E1 GL UNLIMITED	551.2	276.2	248.6		
8 R1E2 GL UNLIMITED	551.2	248.6	223.7		
9 R1E3 GL UNLIMITED	364.0	0.0	0.0		
10 R1E1 SD UNLIMITED	551.2	132.7	119.4		
11 R1E2 SD UNLIMITED	551.2	119.4	107.3		
12 R1E3 SD UNLIMITED	364.0	0.0	0.0		
13 R1E1 ME4 UNLIMITED	551.2	0.0	0.0		
14 R1E2 ME4 UNLIMITED	551.2	0.0	0.0		
15 R1E3 ME4 UNLIMITED	364.0	0.0	0.0		
16 CUP 6093 UNLIMITED	572.0	0.0	3.0	0.0	
17 (UNUSED)	0.0	0.0	100.0	0.0	
18 CUP 6070 UNLIMITED	2444.0	801.9	5.0	761.8	****
19 CUP 6051 UNLIMITED	520.0	647.5	0.0	520.0	****
20 CUP 6031 UNLIMITED	520.0	761.8	15.0	442.0	****

**** feed rate exceeds capacity.

TIME = 520.0000 SCALING FACTOR = 1.00 FLEET RETURNS = 10.00% OF RECRUIT GRADUATES

		AVERAGE QUEUE	STD. DEV.	MAXQ	CURRENT QUEUE	AVERAGE WAITING TIME
1 R1C OHL	OHL	3.10	1.88	8	2	1.37
2 R1C GL	GL	2.53	1.54	6	1	2.43
3 R1C SD	SD	2.18	1.31	5	4	4.46
4 R1E1 OHL	OHL	21.23	20.34	69	65	9.91
5 R1E2 OHL	OHL	0.00	0.00	1	0	0.00
6 R1E3 OHL	OHL	0.00	0.00	0	0	0.00
7 R1E1 GL	GL	0.00	0.00	1	0	0.00
8 R1E2 GL	GL	0.00	0.00	1	0	0.00
9 R1E3 GL	GL	0.00	0.00	0	0	0.00
10 R1E1 SD	SD	0.00	0.00	1	0	0.00
11 R1E2 SD	SD	0.00	0.00	1	0	0.00
12 R1E3 SD	SD	0.00	0.00	0	0	0.00
13 R1E1 ME4	ME4	0.00	0.00	0	0	0.00
14 R1E2 ME4	ME4	0.00	0.00	0	0	0.00
15 R1E3 ME4	ME4	0.00	0.00	0	0	0.00
16 CUP 6093	ME4	0.00	0.00	0	0	0.00
17 (UNUSED)	ME4	0.00	0.00	0	0	0.00
18 CUP 6070	ME4	19.42	15.53	61	26	8.43
19 CUP 6051	ME4	0.59	1.36	16	1	0.54
20 CUP 6031	ME4	108.31	102.60	313	303	53.93
TOTALS:				402	402	

Figure 6(b). Sample output for one rating of regular type students.

Figure 6(c). Sample output for one rating of regular type students.

simulated. The simulation is being run for 2 years (520 working days). Ten percent of recruit graduates will be cloned upon graduation to represent fleet returnees entering the system.

Under "STUDENT DATA:" we see that this type of student will take two BEE courses ($BE = 2$). NROUTE tells us where students who finish basic training or A-school at the 11 places modeled will go for BEE. Since basic training is only taught at Orlando, Great Lakes and San Diego, the remaining 8 columns could have been left blank for this particular type of student (for regular types, BEE will always follow basic training; it will never follow A-school). After taking BEE this type of student will take courses 18, 20 and 19 in that order. Courses are assigned numbers in the order they appear in the Course Data cards. Between the list of "A" courses and "BEE PRI" the number 3 tells us that three "A" courses have been specified for this rating. His priority for BEE is 0.0. For a possible five years to be modeled, 1000 students per year will report to the Navy for basic training. Of these, 59% will report to Orlando, 28% to Great Lakes, and 13% to San Diego. Of the new recruits, 9% will report in January, 7% in February, 7% in March, 6% in April, 7% in May, 11% in June, ... 7% in November and 6% in December. Eighteen percent will come on Mondays, 20% on Tuesdays, 24% on Wednesdays, 20% on Thursdays and 18% on Fridays and weekends. Figures 7 and 8 depict the yearly and weekly input cycles. Yearly and weekly input cycles.

The "COURSE DATA:" indicates the course name (CNAME), failure and setback rates (FAIL and SBACK), minimum and maximum class sizes (MIN and MAX), course duration in working days (LENGTH), time from the beginning of simulation until first scheduled class convening (FRST), and time between class convenings (BET). For basic training and BEE courses, MIN and MAX are scaled to reflect the number who will actually be this type of student. For BEE courses, MAX is the number of chairs available, LENGTH is the average course duration, and FRST and BET are irrelevant since there are no actual class convenings (due to the self-paced nature of BEE).

Array WHR tells us the base numbers where the courses are taught.

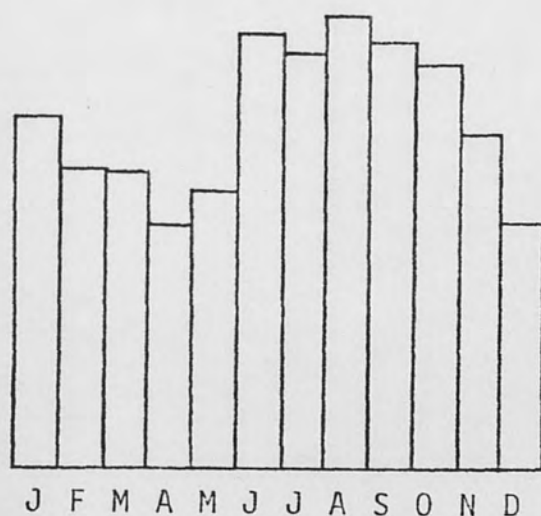


Figure 7. Yearly input fluctuation.

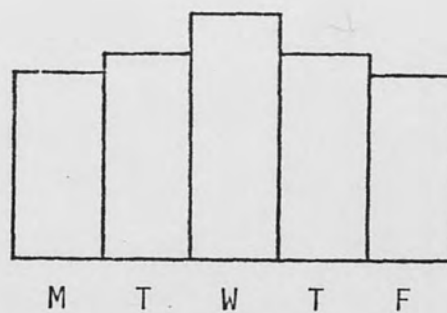


Figure 8. Weekly input fluctuation.

The "PRE-EXECUTION DATA ANALYSIS:," shown in Figure 6(b), alerts the analyst to all potential bottlenecks. The bottleneck at course 19 will be masked by the one at course 20 (the student takes "A" courses in the order 18, 20, 19). Only 520 students per year can be admitted to course 20. Of those, the data specifies that 15% will fail. This leaves a maximum feed rate of $520 \times 0.85 = 442$ students per year who will reach course 19. The bottleneck at BEE1 ORL will not mask the one at 20 because the "A" school is fed from the outputs of all the BEE schools, and these combined outputs are sufficient to choke course 20. The projected steady state student output of 647.5 students per year cannot be realized because of the bottlenecks.

At time 520 days, the first Model Report is printed. A total of 1972 recruits have been generated (bottom of Figure 6(c)). This is very close to the 1000 students/year specified. Referring to Figure 6(c), of the 1972 recruits, 1175 have started basic training at Orlando, 540 at Great Lakes and 250 at San Diego. Seven ($2 + 1 + 4$) are waiting for basic training. Of the 1175 to start basic training at Orlando, 992 have passed, 113 have failed and 70 are still under instruction. Of the 123 setbacks, some have passed, and some are still under instruction.

All basic training graduates will remain at the same base for BEE. A total of $992 + 448 + 214 = 1654$ students have passed basic training. Of these, 155 have been cloned to make fleet returnees (bottom of Figure 6(c)). This makes $155 + 1654 = 1809$ who have either started BEE1 (the first of the two required BEE courses) or are

waiting for it (1019/65 in Orlando, 488/0 in Great Lakes, 237/0 in San Diego). Note the bottleneck predicted for BEE1 in Orlando. In Figure 6(b), the current queue length for BEE1 at Orlando is very close to the maximum length, indicating a growing queue and a bottleneck situation. Those who finish BEE1 go directly to BEE2 at the same base.

Of the $719 + 360 + 164 = 1243$ students who have passed BEE2 at the three bases, 1217 have begun their first "A" course (18), and 26 are waiting for it. Of the 1053 who have finished 18, 750 have started 20, and 303 are waiting for it -- in harmony with the prediction of a bottleneck at this point. Of the 564 who have finished 20, 563 have begun 19, and only 1 is waiting, in harmony with the prediction that the potential bottleneck at this point would not be apparent here. The one student waiting alone is insufficient to form a class of minimum size (specified to be 5 students).

The 505 who have finished 19 have finished their training. Of these, 448 were regular recruits with an average training time of 258.28 working days (1 year = 260 working days), and 57 were fleet returnees with an average time of 213.19 days. One might expect the average time for fleet returnees to be shorter in general than the time for regular recruits because fleet returnees do not have to take basic training over, and they have higher queuing priority for BEE school. This is probably true, but will not always be indicated in the simulation. Sometimes a significant number of regular recruits will finish training with uncommonly short times because they got

through before severe queuing problems developed at the bottlenecks. No fleet returnees enter the system until they are cloned from graduating recruits, hence they will not have as much chance at the short queues. Although they have head-of-the line privileges for BEE, no such privilege has been specified for "A" courses. (Such a priority for "A" courses could easily be added.) Although rare, runs have indicated longer times for fleet returnees of some types.

Of the total 2127 students to enter the system (1972 regular, 155 fleet returnees), 505 have finished training (i.e., finished 19), 662 have left by failing, 558 are currently under instruction and, 402 are awaiting instruction. The maximum number of concurrent entries was 980. This includes scheduled events as well as students.

Histograms of the student time-in-system for both regular recruits and fleet returnees are shown in Figure 9. Note the differences between the mean, median and mode of each.

2.3.4.2 A-before-BEE students

A sample A-before-BEE student output is given in Figure 10. The student data used is the same, except that BE is negative to indicate that this student goes to A-school before BEE school. The preliminary data analysis indicates a different set of bottlenecks. Since all students take their "A" courses first, they are all in Memphis when they finish A-school and are ready to start BEE school. (Array WHR indicates that course 19, the last in the "A" sequence, is taught at base 4, Memphis). NROUTE indicates that students of this

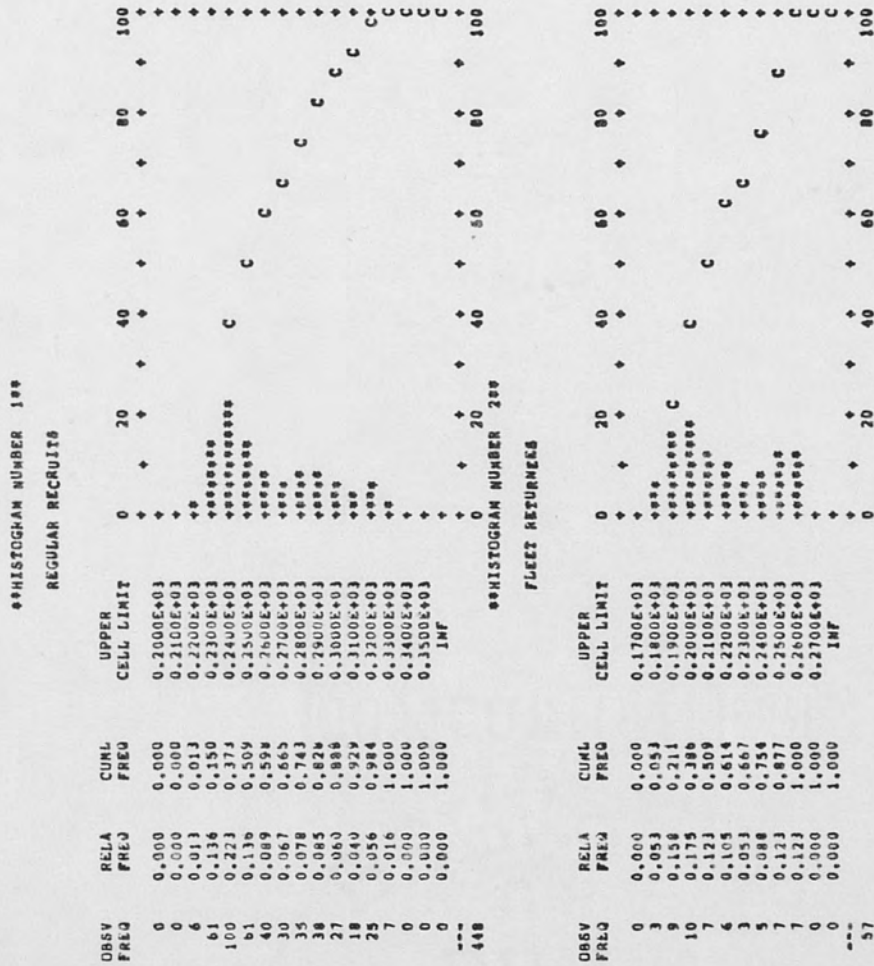


Figure 9. Histograms of time-in-system for a single rating.


```

ICASE = 0
SCALING FACTOR = 1.00
TIME UNTIL FIRST REPORT = 520.0
TIME BETWEEN REPORTS = 50.0
TIME UNTIL FIRST SHAPSHOT = 99100.0
TIME BETWEEN SHAPSHOTS = 8898.0
PKEY RETURNS = 10.000

```

STUDENT DATA:

TYPE	BE	ROUTE	A COURSE	SEC PRI
1	2	1	2	0
1	2	2	3	0
1	2	3	1	0
1	2	3	2	0
1	2	3	3	0
1	2	3	4	0
1	2	3	5	0
1	2	3	6	0
1	2	3	7	0
1	2	3	8	0
1	2	3	9	0
1	2	3	10	0
1	2	3	11	0
1	2	3	12	0
1	2	3	13	0
1	2	3	14	0
1	2	3	15	0
1	2	3	16	0
1	2	3	17	0
1	2	3	18	0
1	2	3	19	0
1	2	3	20	0
1	2	3	21	0
1	2	3	22	0
1	2	3	23	0
1	2	3	24	0
1	2	3	25	0
1	2	3	26	0
1	2	3	27	0
1	2	3	28	0
1	2	3	29	0
1	2	3	30	0
1	2	3	31	0
1	2	3	32	0
1	2	3	33	0
1	2	3	34	0
1	2	3	35	0
1	2	3	36	0
1	2	3	37	0
1	2	3	38	0
1	2	3	39	0
1	2	3	40	0
1	2	3	41	0
1	2	3	42	0
1	2	3	43	0
1	2	3	44	0
1	2	3	45	0
1	2	3	46	0
1	2	3	47	0
1	2	3	48	0
1	2	3	49	0
1	2	3	50	0
1	2	3	51	0
1	2	3	52	0
1	2	3	53	0
1	2	3	54	0
1	2	3	55	0
1	2	3	56	0
1	2	3	57	0
1	2	3	58	0
1	2	3	59	0
1	2	3	60	0
1	2	3	61	0
1	2	3	62	0
1	2	3	63	0
1	2	3	64	0
1	2	3	65	0
1	2	3	66	0
1	2	3	67	0
1	2	3	68	0
1	2	3	69	0
1	2	3	70	0
1	2	3	71	0
1	2	3	72	0
1	2	3	73	0
1	2	3	74	0
1	2	3	75	0
1	2	3	76	0
1	2	3	77	0
1	2	3	78	0
1	2	3	79	0
1	2	3	80	0
1	2	3	81	0
1	2	3	82	0
1	2	3	83	0
1	2	3	84	0
1	2	3	85	0
1	2	3	86	0
1	2	3	87	0
1	2	3	88	0
1	2	3	89	0
1	2	3	90	0
1	2	3	91	0
1	2	3	92	0

'338 eiofoq sams toochs y taut setepuf #38. oatiobem

TOTAL YEARLY DEMAND	BASE FACTORS	MONTH FACTORS	DAY-OF-WEEK FACTORS
1000-1000-1000-1000-1000-	0.59 0.28 0.13	0.09 0.07 0.07 0.06 0.07	0.11 0.10 0.11 0.10 0.09 0.07 0.06
0.18 0.20 0.24 0.30 0.18			

CNAME	COURSE DATA		MAX	LENGTH	FIRST	BET
	FAIL SECK	(\$)				
1 1 MTC ORL	10.00	5.00	5.00	40.0	0.0	1.0
2 2 MTC GL	10.00	5.00	5.00	40.0	0.0	1.0
3 3 MTC SD	10.00	5.00	5.00	40.0	0.0	1.0
4 4 MTC ORL	10.00	0.00	53.00	25.0	0.0	0.0
5 5 SEET ORL	10.00	0.00	53.00	25.0	0.0	0.0
6 6 SEET ORL	10.00	0.00	21.00	15.0	0.0	0.0
7 7 SEET GL	10.00	0.00	53.00	25.0	0.0	0.0
8 8 SEET GL	10.00	0.00	53.00	25.0	0.0	0.0
9 9 SEET SD	10.00	0.00	21.00	15.0	0.0	0.0
10 10 SEET GL	10.00	0.00	53.00	25.0	0.0	0.0
11 11 SEET SD	10.00	0.00	53.00	25.0	0.0	0.0
12 12 SEET SD	10.00	0.00	21.00	15.0	0.0	0.0
13 13 SEET MEM	10.00	0.00	53.00	25.0	0.0	0.0
14 14 SEET MEM	10.00	0.00	53.00	25.0	0.0	0.0
15 15 SEET MEM	10.00	0.00	21.00	15.0	0.0	0.0
16 16 COP SD	2.00	1.00	22.00	25.0	0.0	10.0
17 17 COP SD	10.00	0.00	0.00	9999.0	9999.0	9999.0
18 18 COP SD	5.00	4.00	47.00	33.0	0.0	5.0
19 19 COP SD	0.00	5.00	10.00	33.0	0.0	5.0
20 20 COP SD	15.00	40.00	5.00	10.00	40.0	5.0

[illegible]

Figure 10. Sample output for one rating of A-before-BEE students.

PRE-EXECUTION DATA ANALYSIS: Fleet returns = 10.0% of basic training graduates.

STEADY STATE STUDENT I/O RATES (if no bottlenecks)
STUDENT TYPE 1, 01F (students per year)
1 SAMP ACZ 1000.0 647.5

COURSE MAXIMUM CAPACITIES AND FEED RATES (assuming no previous bottlenecks restricting feed):

COURSE	CAPACITY	FEED RATE	WALL	GRAD RATE	(students per year)
1 RTC ORL UNLIMITED	587.0	10.0	528.3		
2 RTC GL UNLIMITED	279.0	10.0	251.1		
3 RTC SD UNLIMITED	134.0	10.0	120.8		
4 BEE1 ORL UNLIMITED	551.2	10.0	496.1	****	
5 BEE2 ORL UNLIMITED	719.5	10.0	496.1	****	
6 BEE3 ORL UNLIMITED	364.0	0.0	0.0		
7 BEE1 GL UNLIMITED	551.2	0.0	0.0		
8 BEE2 GL UNLIMITED	551.2	0.0	0.0		
9 BEE3 GL UNLIMITED	364.0	0.0	0.0		
10 BEE1 SD UNLIMITED	551.2	0.0	0.0		
11 BEE2 SD UNLIMITED	551.2	0.0	0.0		
12 BEE3 SD UNLIMITED	364.0	0.0	0.0		
13 BEE1 MEM UNLIMITED	551.2	0.0	0.0		
14 BEE2 MEM UNLIMITED	551.2	0.0	0.0		
15 BEE3 MEM UNLIMITED	364.0	0.0	0.0		
16 CDP 6093 UNLIMITED	572.0	0.0	3.0		
17 (UNUSED)	0.0	0.0	100.0		
18 CDP 6070 UNLIMITED	2444.0	990.0	5.0	940.5	****
19 CDP 6051 UNLIMITED	520.0	799.4	0.0	520.0	****
20 CDP 603V UNLIMITED	520.0	940.5	15.0	442.0	****

**** feed rate exceeds capacity.

TIME = 520.0000 SCALING FACTOR = 1.00 FLEET RETURNS = 10.0% OF RECRUIT GRADUATES

		AVERAGE QUEUE	STD. DEV.	CURRENT QUEUE	AVERAGE WAITING TIME
1	RTC ORL	3.10	1.88	8	2
2	RTC GL	2.53	1.54	6	1
3	RTC SD	2.18	1.31	5	4
4	BEE1 ORL	0.00	0.04	1	0
5	BEE2 ORL	0.00	0.00	1	0
6	BEE3 ORL	0.00	0.00	0	0
7	BEE1 GL	0.00	0.00	0	0
8	BEE2 GL	0.00	0.00	0	0
9	BEE3 GL	0.00	0.00	0	0
10	BEE1 SD	0.00	0.00	0	0
11	BEE2 SD	0.00	0.00	0	0
12	BEE3 SD	0.00	0.00	0	0
13	BEE1 MEM	0.00	0.00	0	0
14	BEE2 MEM	0.00	0.00	0	0
15	BEE3 MEM	0.00	0.00	0	0
16	CDP 6093	0.00	0.00	0	0
17	(UNUSED)	0.00	0.00	0	0
18	CDP 6070	22.22	15.00	71	1
19	CDP 6051	0.63	1.00	15	0
20	CDP 603V	265.75	217.00	668	648
TOTALS:					92.03
					656

Figure 10(b). Sample output for one rating of A-before-BEE students.

TIME = 520.0000

SPECIFICATIONS										RESULTS			
COURSE LENGTH		BET	MIN	MAX	FAIL	SETBACK	NO. START	NO. PASS	NO. NO. FAIL	NO. NO. BACK	UNDER INSTR	AWAY INSTR	
1	WTC ORL	40.0	1.0	5.0	5.0	10.0	10.0	1175	973	130	115	72	2
2	ATC GL	40.0	1.0	5.0	5.0	10.0	10.0	540	446	58	48	36	1
3	ATC SD	40.0	1.0	5.0	5.0	10.0	10.0	250	211	24	22	15	4
4	BEE1 ORL	25.0	0.0	0.0	53.0	10.0	10.0	591	503	40	0	48	0
5	BEE2 ORL	25.0	0.0	0.0	53.0	10.0	10.0	503	439	33	0	31	0
6	BEE3 ORL	15.0	0.0	0.0	21.0	10.0	10.0	0	0	0	0	0	0
7	BEE1 GL	25.0	0.0	0.0	53.0	10.0	10.0	0	0	0	0	0	0
8	BEE2 GL	25.0	0.0	0.0	53.0	10.0	10.0	0	0	0	0	0	0
9	BEE3 GL	15.0	0.0	0.0	21.0	10.0	10.0	0	0	0	0	0	0
10	BEE1 SD	25.0	0.0	0.0	53.0	10.0	10.0	0	0	0	0	0	0
11	BEE2 SD	25.0	0.0	0.0	53.0	10.0	10.0	0	0	0	0	0	0
12	BEE3 SD	15.0	0.0	0.0	21.0	10.0	10.0	0	0	0	0	0	0
13	BEE1 MEM	25.0	0.0	0.0	53.0	10.0	10.0	0	0	0	0	0	0
14	BEE2 MEM	25.0	0.0	0.0	53.0	10.0	10.0	0	0	0	0	0	0
15	BEE3 MEM	15.0	0.0	0.0	21.0	10.0	10.0	0	0	0	0	0	0
16	CDP 6093 MEM	29.0	10.0	11.0	22.0	3.0	2.0	0	0	0	0	0	0
17	(UNUSED)	MEM	999999.0	99999.0	0.0	0.0	100.0	0	0	0	0	0	0
18	CDP 6070 MEM	33.0	5.0	45.0	47.0	3.0	30.0	1779	1818	73	844	188	1
19	CDP 6053 MEM	33.0	5.0	5.0	10.0	0.0	0.0	850	591	0	0	59	0
20	CUP 603V MEM	40.0	5.0	10.0	15.0	40.0	40.0	870	650	135	343	85	648
										483	834	656	
TOTALS:										483	834	656	

TIME = 520.0000

	NO. FINISHING PIPELINE		AVERAGE TIME		A COURSES	
	REG	FUER	REG	FUER		
1	SAMP	A+E	406	33	273-51	225.30
			----	----		
			406	33		
TOTALS:						

TOTAL NUMBER OF RECRUIT ARRIVALS TO SYSTEM = 1972
TOTAL FLEEING RETUENERS = 551
TOTAL IN TRANSIT = 1
MAXIMUM NUMBER OF CONCURRENT ENTRIES = 1210 (NOT DECALED)

Figure 10(c). Sample output for one rating of A-before-BEE students.

type who find themselves at place 4 when the time comes to go to BEE school will go to place 1 (Orlando) for BEE. Hence, everybody goes to Orlando for BEE, and as one might expect there is a hideous bottleneck predicted for BEE1 at Orlando. A bottleneck is also predicted for BEE2, but this will be masked by the bottleneck at BEE1. In fact, the bottleneck at course 20 is the only one which will not be masked. It hides the bottlenecks at course 19 and at both BEEs. Maximum student output will be limited to the 520/year maximum input to course 20 times the pass rates for all subsequent courses:

$$520 \times 0.85 \times 1.00 \times 0.90 \times 0.90 = 358.02 \text{ students per year.}$$

As expected, the report written at time = 520 indicates a bottleneck only for course 20. Of the 1972 regular recruits who reported to the RTCs, 7 are waiting, and 1965 have started basic training. Of the 1630 who graduated, 151 were cloned, making 1781 ready for A-school. Of these 1781, 1779 have begun the first course (18), 1 is waiting, and 1 is in transit to the school. Of the 1518 to finish 18, 870 have started 20, and 648 are waiting. Of the 650 to finish 20,, all have started 19. Of the 591 to finish 19, all have started BEE1 in Orlando. Of the 503 to finish BEE1, all have started BEE2. Of the 439 to finish training (BEE2), 406 were regular recruits, and 33 were fleet returnees.

2.3.4.3 AFUN students

A sample output of an AFUN type is given in Figure 11. Again the same student data is used, except that he is tagged as an AFUN type by listing the second part of his name as "AFUN." Although actual AFUN types only take one BEE course, this sample takes two. Bottlenecks are again predicted for Orlando. This time they should be visible in the Report, since the first "A" course will be sending $990 \times 0.95 = 940.5$ students per year to BEE1 in Orlando. BEE1 in Orlando will hide the bottlenecks at BEE2, 20 and 19. The Report at time = 520 shows the queueing problem only at BEE1 in Orlando.

Of the 1646 students who have finished basic training, 154 have been cloned, making 1800 who have been sent to the first "A" course (representing AFUN). Of those, 1782 have started the course, 17 are waiting, and 1 is in transit. Of the 1543 who have finished 18, 957 have started BEE1 at Orlando, and 586 are waiting for it. Of the 681 to finish at BEE school, 677 have started "A" course 20, and 4 are waiting. Of the 509 to finish 20, 507 have started 19, and 2 are waiting. Of the 452 to complete their training, 392 were regular recruits, and 60 were fleet returnees.

Such headcounts as the ones described above are only feasible when modeling only one student type, because of intersections of pipelines in basic training and BEE school and occasionally A-school. When mixing categories of student types (regular, A-before-BEE and AFUN), the number of starts and number waiting for BEE1s will not

```

ICASE = 0
SCALING FACTOR = 1.00
TIME UNTIL FIRST REPORT = 520.0
TIME BETWEEN REPORTS = 50.0
TIME UNTIL FIRST SHAPSHOT = 99100.0
TIME BETWEEN SHAPSHOTS = 88888.0
PLAT RETURNS = 10.000

```

STUDENT DATA

TITLE	RE	THROUGH	A COURSE	SEP	PR
1	1	2	1	0	0
1	2	3	2	0	0
1	3	4	3	0	0
1	4	5	4	0	0
1	5	6	5	0	0
1	6	7	6	0	0
1	7	8	7	0	0
1	8	9	8	0	0
1	9	10	9	0	0
1	10	11	10	0	0
1	11	12	11	0	0
1	12	13	12	0	0
1	13	14	13	0	0
1	14	15	14	0	0
1	15	16	15	0	0
1	16	17	16	0	0
1	17	18	17	0	0
1	18	19	18	0	0
1	19	20	19	0	0
1	20	21	20	0	0
1	21	22	21	0	0
1	22	23	22	0	0
1	23	24	23	0	0
1	24	25	24	0	0
1	25	26	25	0	0
1	26	27	26	0	0
1	27	28	27	0	0
1	28	29	28	0	0
1	29	30	29	0	0
1	30	31	30	0	0
1	31	32	31	0	0
1	32	33	32	0	0
1	33	34	33	0	0
1	34	35	34	0	0
1	35	36	35	0	0
1	36	37	36	0	0
1	37	38	37	0	0
1	38	39	38	0	0
1	39	40	39	0	0
1	40	41	40	0	0
1	41	42	41	0	0
1	42	43	42	0	0
1	43	44	43	0	0
1	44	45	44	0	0
1	45	46	45	0	0
1	46	47	46	0	0
1	47	48	47	0	0
1	48	49	48	0	0
1	49	50	49	0	0
1	50	51	50	0	0
1	51	52	51	0	0
1	52	53	52	0	0
1	53	54	53	0	0
1	54	55	54	0	0
1	55	56	55	0	0
1	56	57	56	0	0
1	57	58	57	0	0
1	58	59	58	0	0
1	59	60	59	0	0
1	60	61	60	0	0
1	61	62	61	0	0
1	62	63	62	0	0
1	63	64	63	0	0
1	64	65	64	0	0
1	65	66	65	0	0
1	66	67	66	0	0
1	67	68	67	0	0
1	68	69	68	0	0
1	69	70	69	0	0
1	70	71	70	0	0
1	71	72	71	0	0
1	72	73	72	0	0
1	73	74	73	0	0
1	74	75	74	0	0
1	75	76	75	0	0
1	76	77	76	0	0
1	77	78	77	0	0
1	78	79	78	0	0

negative "SE" indicates that a school comes before SE.

TOTAL YEARLY DEMAND	BASE FACTORS	MONTH FACTORS	DAY-OF-WEEK FACTORS
1,000,100,100,100,1000,	0.59,0.28,0.13	0.09,0.07,0.07,0.06,0.67	0.11,0.10,0.11,0.10,0.09,0.07,0.06
1,000,100,100,100,100,1000,			0.18,0.20,0.24,0.26,0.20,0.18

COURSE DATA:

	CHAVE	FAIL BACK	MIN	MAX	LENGTH	FIRST	DEPT
1	1	RTC ORU	10.00	10.00	5.00	40.0	0.0
2	2	RTC GL	10.00	10.00	5.00	40.0	0.0
3	3	RTC SD	10.00	10.00	5.00	40.0	0.0
4	4	BE1 ORU	10.00	10.00	0.00	53.0	25.0
5	5	BE2 ORU	10.00	10.00	0.00	53.0	0.0
6	6	BE3 ORU	10.00	10.00	0.00	21.0	0.0
7	7	BE1 GL	10.00	10.00	0.00	53.0	25.0
8	8	BE2 GL	10.00	10.00	0.00	53.0	25.0
9	9	BE3 GL	10.00	10.00	0.00	21.0	0.0
10	10	BE1 SD	10.00	10.00	0.00	53.0	25.0
11	11	BE2 SD	10.00	10.00	0.00	53.0	25.0
12	12	BE3 SD	10.00	10.00	0.00	21.0	0.0
13	13	BE1 MEM	10.00	10.00	0.00	53.0	25.0
14	14	BE2 MEM	10.00	10.00	0.00	53.0	25.0
15	15	BE3 MEM	10.00	10.00	0.00	21.0	0.0
16	16	CDP 6093	3.00	3.00	0.00	21.0	0.0
17	17	UNUSU100.00	0.00	0.00	11.00	23.0	29.0
18	18	CDP 6070	5.00	30.00	0.00	9999.0	9999.0
19	19	CDP 6054	0.00	0.00	45.00	47.00	33.0
20	20	CDP 603V	15.00	40.00	5.00	10.00	33.0
					5.00	10.00	40.0
							5.0

[illegible]

Figure 11(a). Sample output for one rating of AFJN students.

PPE-EXECUTION DATA ANALYSIS: Fleet returnees = 10.00% of basic training graduates.

STEADY STATE STUDENT I/U RATES (if no bottlenecks)
 STUDENT TYPE 14 UNIT (students per year)
 1 SAMPAFVN 1000.0 647.5

COURSE MAXIMUM CAPACITIES AND FEED RATES (assuming no previous bottlenecks restricting feed):

COURSE	CAPACITY	FEED RATE	SPALL	CHAD RATE	(students per year)
1 RTC ORL UNLIMITED	587.0	10.0	528.3		
2 RTC GL UNLIMITED	279.0	10.0	251.1		
3 RTC SD UNLIMITED	134.0	10.0	120.6		
4 REE1 ORL	551.2	940.5	10.0	496.1	****
5 REE2 ORL	551.2	845.5	10.0	496.1	****
6 REE3 ORL	364.0	0.0	10.0	0.0	
7 REE1 GL	551.2	0.0	10.0	0.0	
8 REE2 GL	551.2	0.0	10.0	0.0	
9 REE3 GL	364.0	0.0	10.0	0.0	
10 REE1 SD	551.2	0.0	10.0	0.0	
11 REE2 SD	551.2	0.0	10.0	0.0	
12 REE3 SD	364.0	0.0	10.0	0.0	
13 REE1 MEM	551.2	0.0	10.0	0.0	
14 REE2 MEM	551.2	0.0	10.0	0.0	
15 REE3 MEM	364.0	0.0	10.0	0.0	
16 CUP 6093	572.0	0.0	3.0	0.0	
17 (UNUSED)	0.0	0.0	100.0	0.0	
18 CUP 6070	2444.0	990.0	5.0	940.5	
19 CUP 6051	520.0	647.5	0.0	520.0	****
20 CUP 603V	520.0	761.8	15.0	442.0	****

**** feed rate exceeds capacity.

TIME = 520.0000 SCALING FACTOR = 1.00 FLEET RETURNES = 10.00% OF RECRUIT GRADUATES

	AVERAGE QUEUE	SID. DEV.	MAXO QUEUE	CURRENT QUEUE	AVERAGE WAITING TIME
1 RTC ORL	3.10	1.88	8	2	1.37
2 RTC GL	2.53	1.54	6	1	2.43
3 RTC SD	2.18	1.51	5	4	4.46
4 REE1 ORL	220.31	184.47	586	586	70.47
5 REE2 ORL	0.00	0.00	1	0	0.00
6 REE3 ORL	0.00	0.00	0	0	0.00
7 REE1 GL	0.00	0.00	0	0	0.00
8 REE2 GL	0.00	0.00	0	0	0.00
9 REE3 GL	0.00	0.00	0	0	0.00
10 REE1 SD	0.00	0.00	0	0	0.00
11 REE2 SD	0.00	0.00	0	0	0.00
12 REE3 SD	0.00	0.00	0	0	0.00
13 REE1 MEM	0.00	0.00	0	0	0.00
14 REE2 MEM	0.00	0.00	0	0	0.00
15 REE3 MEM	0.00	0.00	0	0	0.00
16 CUP 6033	0.00	0.00	0	0	0.00
17 (UNUSED)	0.00	0.00	0	0	0.00
18 CUP 6070	23.59	15.24	65	17	6.88
19 CUP 6051	0.47	1.22	16	2	0.48
20 CUP 603V	2.04	2.60	16	4	1.56
TOTALS:				616	

Figure 11(b). Sample output for one rating of AFUN students.

TIME = 520.0000										SCALING FACTOR = 1.00										FLEET RETURNEES = 10.00% OF RECRUIT GRADUATES									
SPECIFICATIONS										RESULTS										UNDER AWAIT									
COURSE										NO.										NO.									
LENGTH	SET	MIN	MAX	FAIL	SETBACK	START	PASS	FAIL	SETBACK	START	PASS	FAIL	SETBACK	INSTR	INSTR														
1 RHC ORL	40.0	1.0	5.0	5.0	10.0	10.0	1175	990	115	119	70	2																	
2 RHC GL	40.0	1.0	5.0	5.0	10.0	10.0	540	456	59	51	35	1																	
3 RHC SD	40.0	1.0	5.0	5.0	10.0	10.0	250	210	24	23	16	4																	
4 REE1 ORL	25.0	0.0	0.0	53.0	10.0	10.0	957	806	98	0	53	596																	
5 REE2 ORL	25.0	0.0	0.0	53.0	10.0	10.0	806	681	80	0	45	0																	
6 REE3 ORL	15.0	0.0	0.0	21.0	10.0	10.0	0	0	0	0	0	0																	
7 REE1 GL	25.0	0.0	0.0	53.0	10.0	10.0	0	0	0	0	0	0																	
8 REE2 GL	25.0	0.0	0.0	53.0	10.0	10.0	0	0	0	0	0	0																	
9 REE3 GL	15.0	0.0	0.0	21.0	10.0	10.0	0	0	0	0	0	0																	
10 REE1 SD	25.0	0.0	0.0	53.0	10.0	10.0	0	0	0	0	0	0																	
11 REE2 SD	25.0	0.0	0.0	53.0	10.0	10.0	0	0	0	0	0	0																	
12 REE3 SD	25.0	0.0	0.0	21.0	10.0	10.0	0	0	0	0	0	0																	
13 REE1 MEM	25.0	0.0	0.0	53.0	10.0	10.0	0	0	0	0	0	0																	
14 REE2 MEM	25.0	0.0	0.0	53.0	10.0	10.0	0	0	0	0	0	0																	
15 REE3 MEM	25.0	0.0	0.0	21.0	10.0	10.0	0	0	0	0	0	0																	
16 CDP 603V	MEM	29.0	10.0	22.0	3.0	2.0	0	0	0	0	0	0																	
17 (UNUSED)	MEM	9999.099999	0	0	0	0	0	0	0	0	0	0																	
18 CDP 6070	MEM	33.0	5.0	45.0	47.0	5.0	30.0	1782	1543	80	546	189	17																
19 CDP 605Y	MEM	33.0	5.0	5.0	10.0	0.0	0.0	507	452	0	0	55	2																
20 CDP 603V	MEM	40.0	5.0	5.0	10.0	15.0	40.0	677	509	95	290	73	4																
TOTALS:										881										806									

TIME = 520.0000										SCALING FACTOR = 1.00										FLEET RETURNEES = 10.00% OF RECRUIT GRADUATES									
NO.										A COURSES																			
FINISHING PIPELINE										A COURSES																			
REG	FLEET	REG	FLEET	REG	FLEET	REG	FLEET	REG	FLEET	REG	FLEET	REG	FLEET	REG	FLEET	REG	FLEET	REG	FLEET	REG	FLEET	REG	FLEET	REG	FLEET	REG	FLEET	REG	FLEET
1	SAMP - AFUN	392	60	268.03	173.22	CDP 6070,	CDP 603V,	CDP 603Y,																					
TOTALS:										392										60									

TOTAL NUMBER OF RECRUIT ARRIVALS TO SYSTEM = 1972																													
TOTAL FLEET RETURNEES = 154																													
TOTAL IN TRANSIT = 1																													
MAXIMUM NUMBER OF CONCURRENT ENTRIES = 1142 (NOT DESCALED)																													

TIME = 520.0000										SCALING FACTOR = 1.00										FLEET RETURNEES = 10.00% OF RECRUIT GRADUATES									
FINISHING PIPELINE										A COURSES:																			
NO.	REG	FLER	REG	FLER	REG	FLER	REG	FLER	REG	NO.	NO.	NO.	NO.	NO.	NO.	NO.	NO.	NO.	NO.	NO.	NO.	NO.	NO.	NO.	NO.	NO.	NO.	NO.	NO.
1	SAMP - AFUN	392	50	268.03	173.22	CDP 6070, CDP 603V, CDP 603Y,																							
TOTALS:										392										50									

TOTAL NUMBER OF RECRUIT ARRIVALS TO SYSTEM = 1972
 TOTAL FLEET RETURNEES = 154
 TOTAL IN TRANSIT = 1
 MAXIMUM NUMBER OF CONCURRENT ENTRIES = 1142 (NOT DESCALED)

Figure 11(c). Sample output for one rating of AFUN students.

equal the number of RTC graduates plus fleet returnees generated. This is because some of the basic training graduates and fleet returnees go to A-school first, and some of the BEE students have just come from A-school.

One may be wondering at this time why the results for basic training were not the same for all three examples. All the specifications were the same. The reason is that the random number samples used to decide pass/setback/failure are all taken from same random number stream for all courses. After basic training, students go their separate ways, and the separate paths disturb differently the same stream of numbers which is used to determine prognoses in basic training. One can avoid this perturbation by assigning a separate random number stream to determine pass/failure for each course. The choice of the stream would be made in subroutine FAILUR. This was not considered a worthwhile endeavor, and hence has not been included.

The above runs required 7640 page faults, CPU time = 3:42.76; Peak working set size: 287 pages; Peak virtual size: 615 pages. One page = 512 bytes. A page fault represents the fetching of 250 pages of stored program or data to working memory.

SECTION 2.4

TIMING CONSIDERATIONS

Suppose the following two events are scheduled for the exact same time: 1) A student finishes a course and gets in line for the next one; 2) A class for the next course convenes if enough students are waiting.

One might contend that in reality these two events cannot happen at exactly the same time. This is true. It is also difficult to simulate in such a way that no two events are ever SCHEDULED to occur at the same time. In reality the first event will probably occur on a Friday afternoon, and the second will occur on a Monday morning. Since the simulation does not model weekends, the end of a Friday and the beginning of a Monday are simulated as the same time. A travel time of one real day, adjusted to 0 simulated time might also be involved.

It should be clear that the order of processing of "simultaneous" events such as these can make a big difference in the output. Whether or not the student gets into the queue before the class convening event occurs can make the difference of his having to wait for the next class convening. Suppose the minimum class size is 20 and there are already 19 in the queue, this being the 20th student. If event 1 is processed before event 2, a class forms and they all

start the course at that time. If event 2 is processed first, there are not enough students waiting when the queue is checked, and ALL students in the queue (as well as our 20th student when he gets there 0 time units hence) have to wait for the next class convening. For some courses this is over 100 working days.

How can one control the order of processing of events scheduled at the same time? This problem is addressed by Pritsker and Pegden (1979, p. 141). An event is categorized by the SLAM processor at the time of its scheduling as being either "current" or "future." "Current" events are scheduled for the current time (after delay = 0), and are processed in LIFO (last-in-first-out) order. "Future" events are scheduled for some future time (delay > 0) and are processed in FIFO (first-in-first-out) order unless otherwise specified in a PRIORITY statement.

It seemed reasonable that simultaneous events should be processed in the following order:

```

course ending events (CLASSE and TRAVEL, numbered > LIMCOU);
arrivals to schools after travel (ROUTE, TDST, events 8
and 9);
class convenings (CLASS, event 6);
report writings (events 1, 2, 3).

```

This numbering scheme makes it convenient to break ties based on the event code using a high-value-first (HVF) rule. Event codes are stored in NSET/QSET in the position for regular attribute 3, hence the HVF(3) PRIORITY specification for the calendar file (NCLNR) in the SLAM input statements.

Recruit arrivals are all scheduled to occur at midday (1200 -- $\text{time} = n + 0.5$) and hence will not coincide with the events above which will usually occur at integral times.

Events 14-25 are scheduled by the network and coded differently than events scheduled by the subroutines. However, they are all current events, and hence will be processed immediately (LIFO).

Note: When assigning event numbers or tiebreaking priority rules based on event numbers, bear in mind that the event codes for events CALLED BY THE NETWORK are stored as large NEGATIVE numbers in the SLAM filing system. This is how SLAM tells whether to call subroutine EVENT or process a node. The negative number tells where the coding describing the node is stored in NSET/QSET. All network nodes are coded on the calendar and processed in this manner. The processing of an EVENT node will effect the calling of subroutine EVENT with the proper code. Since all network nodes (in this simulation) represent current events, they will be processed LIFO and the negative event codes and the calendar priority rule will not pose a problem for these events.

APPENDICIES

APPENDIX A

SAMPLE DATA

Column

1 7

```

GEN,ROGER WERNER,TMS MODEL,01/09/82,,N,N,,N,N;
;GEN,ROGER WERNER,TMS MODEL,01/09/82,,N,N,,N,Y; SUMMARY
;GEN,ROGER WERNER,TMS MODEL,01/09/82,,N,N; SUMMARY, ECHO
;GEN,ROGER WERNER,TMS MODEL,01/09/82,,Y,N,,N,N; LIST
;GEN,ROGER WERNER,TMS MODEL,01/09/82,,Y,Y,,N,N; LIST,ECHO
;GEN,ROGER WERNER,TMS MODEL,01/09/82,,Y,N,,N,Y; SUMMARY, LIST
LIMITS,99,2,3000;
PRIORITY/04,HVF(2)/05,HVF(2)/06,HVF(2);
PRIORITY/07,HVF(2)/08,HVF(2)/09,HVF(2);
PRIORITY/10,HVF(2)/11,HVF(2)/12,HVF(2);
PRIORITY/13,HVF(2)/14,HVF(2)/15,HVF(2)/99,HVF(2)/NCLNR,HVF(3);
STAT, 1,STUD TYPE 1, 20/50./10.;
STAT, 2,STUD TYPE 1 FLER, 20/50./10.;
STAT, 3,STUD TYPE 2, 20/50./10.;
STAT, 4,STUD TYPE 2 FLER, 20/50./10.;
STAT, 5,STUD TYPE 3, 20/50./10.;
STAT, 6,STUD TYPE 3 FLER, 20/50./10.;
STAT, 7,STUD TYPE 4, 20/50./10.;
STAT, 8,STUD TYPE 4 FLER, 20/50./10.;
STAT, 9,STUD TYPE 5, 20/50./10.;
STAT,10,STUD TYPE 5 FLER, 20/50./10.;
STAT,11,STUD TYPE 6, 20/50./10.;
STAT,12,STUD TYPE 6 FLER, 20/50./10.;
STAT,13,STUD TYPE 7, 20/50./10.;
STAT,14,STUD TYPE 7 FLER, 20/50./10.;
STAT,15,STUD TYPE 8, 20/50./10.;
STAT,16,STUD TYPE 8 FLER, 20/50./10.;
STAT,17,STUD TYPE 9, 20/50./10.;
STAT,18,STUD TYPE 9 FLER, 20/50./10.;
STAT,19,STUD TYPE 10, 20/50./10.;
STAT,20,STUD TYPE 10 FLER, 20/50./10.;
STAT,21,STUD TYPE 11, 20/50./10.;
STAT,22,STUD TYPE 11 FLER, 20/50./10.;
STAT,23,STUD TYPE 12, 20/50./10.;
STAT,24,STUD TYPE 12 FLER, 20/50./10.;
STAT,25,STUD TYPE 13, 20/50./10.;
STAT,26,STUD TYPE 13 FLER, 20/50./10.;
STAT,27,STUD TYPE 14, 20/50./10.;
STAT,28,STUD TYPE 14 FLER, 20/50./10.;
STAT,29,STUD TYPE 15, 20/50./10.;

```

```

STAT,30,STUD TYPE 15  FLER, 20/50./10.;
STAT,31,STUD TYPE 16, 20/50./10.;
STAT,32,STUD TYPE 16  FLER, 20/50./10.;
STAT,33,STUD TYPE 17, 20/50./10.;
STAT,34,STUD TYPE 17  FLER, 20/50./10.;
STAT,35,STUD TYPE 18, 20/50./10.;
STAT,36,STUD TYPE 18  FLER, 20/50./10.;
STAT,37,STUD TYPE 19, 20/50./10.;
STAT,38,STUD TYPE 19  FLER, 20/50./10.;
STAT,39,STUD TYPE 20, 20/50./10.;
STAT,40,STUD TYPE 20  FLER, 20/50./10.;
STAT,41,STUD TYPE 21, 20/50./10.;
STAT,42,STUD TYPE 21  FLER, 20/50./10.;
STAT,43,STUD TYPE 22, 20/50./10.;
STAT,44,STUD TYPE 22  FLER, 20/50./10.;
STAT,45,STUD TYPE 23, 20/50./10.;
STAT,46,STUD TYPE 23  FLER, 20/50./10.;
STAT,47,STUD TYPE 24, 20/50./10.;
STAT,48,STUD TYPE 24  FLER, 20/50./10.;
STAT,49,STUD TYPE 25, 20/50./10.;
STAT,50,STUD TYPE 25  FLER, 20/50./10.;
STAT,51,STUD TYPE 26, 20/50./10.;
STAT,52,STUD TYPE 26  FLER, 20/50./10.;
STAT,53,STUD TYPE 27, 20/50./10.;
STAT,54,STUD TYPE 27  FLER, 20/50./10.;

```

NETWORK;

```

RESOURCE/BEE1 ORL(00),04/BEE2 ORL(00),05/BEE3 ORL(00),06;
RESOURCE/BEE1 GL(00),07/BEE2 GL(00),08/BEE3 GL(00),09;
RESOURCE/BEE1 SD(00),10/BEE2 SD(00),11/BEE3 SD(00),12;
RESOURCE/BEE1 MEM(00),13/BEE2 MEM(00),14/BEE3 MEM(00),15;
ENTER,4,1;
AWAIT(4),BEE1 ORL;
EVENT, 14;
TERM;
ENTER,5,1;
AWAIT(5),BEE2 ORL;
EVENT, 15;
TERM;
ENTER,6,1;
AWAIT(6),BEE3 ORL;
EVENT, 16;
TERM;
ENTER,7,1;
AWAIT(7),BEE1 GL;
EVENT, 17;
TERM;
ENTER,8,1;
AWAIT(8),BEE2 GL;
EVENT, 18;

```

```

TERM;
ENTER,9,1;
AWAIT(9),BEE3  GL;
EVENT, 19;
TERM;
ENTER,10,1;
AWAIT(10),BEE1  SD;
EVENT, 20;
TERM;
ENTER,11,1;
AWAIT(11),BEE2  SD;
EVENT, 21;
TERM;
ENTER,12,1;
AWAIT(12),BEE3  SD;
EVENT, 22;
TERM;
ENTER,13,1;
AWAIT(13),BEE1 MEM;
EVENT, 23;
TERM;
ENTER,14,1;
AWAIT(14),BEE2 MEM;
EVENT, 24;
TERM;
ENTER,15,1;
AWAIT(15),BEE3 MEM;
EVENT, 25;
TERM;
END;
INIT,0,520;
FIN;
  2   21   4   17   15   13   4   2   2   2   1
IECHOICASE IHSTFACTR FSTR TBR FSTS TBS FLER TBE
  1   0   0.0500 520 509910088888 1099999
1CE      1 1 2 3 1 2 3 1 2 3 1 2 96
      .0858.0713.0707.0597.0686.1073.1015.1103.1042.0891.0715.0599
  1 344 344 344 344 344.1776.2006.2414.2007.1797.5870.2790.134
2CTM      2 1 2 3 1 2 3 1 2 3 1 2 60 61 62 63
      .0858.0713.0707.0597.0686.1073.1015.1103.1042.0891.0715.0599
  2 424 424 424 424 424.1776.2006.2414.2007.1797.8400.0590.101
3DS      2 1 2 3 1 2 3 1 2 3 1 2 94
      .0858.0713.0707.0597.0686.1073.1015.1103.1042.0891.0715.0599
  3 634 634 634 634 634.1776.2006.2414.2007.1797.3450.2480.407
4EM      2 1 2 3 1 2 3 1 2 3 1 2 18
      .0858.0713.0707.0597.0686.1073.1015.1103.1042.0891.0715.0599
  4 2608 2608 2608 2608 2608.1776.2006.2414.2007.1797.2160.4840.300
8ET AEF    3 1 2 3 1 2 3 1 2 3 1 2 20
      .0858.0713.0707.0597.0686.1073.1015.1103.1042.0891.0715.0599

```


8	640	640	640	640	640	1776.2006.2414.2007.1797.3110.5180.171
9ET NF	3	1	2	3	1 2 3 1 2 3 1 2 21	
						.0858.0713.0707.0597.0686.1073.1015.1103.1042.0891.0715.0599
9	601	601	601	601	601	1776.2006.2414.2007.1797.2060.4580.336
10ET	3	1	2	3	1 2 3 1 2 3 1 2 22	
						.0858.0713.0707.0597.0686.1073.1015.1103.1042.0891.0715.0599
10	122	122	122	122	122	1776.2006.2414.2007.1797.2370.6600.103
12EW	2	1	2	3	1 2 3 1 2 3 1 2 68 69 70 71 72 73	
						.0858.0713.0707.0597.0686.1073.1015.1103.1042.0891.0715.0599
12	549	549	549	549	549	1776.2006.2414.2007.1797.8070.1040.089
13FTG	2	1	2	3	1 2 3 1 2 3 1 2 26 27	
						.0858.0713.0707.0597.0686.1073.1015.1103.1042.0891.0715.0599
13	731	731	731	731	731	1776.2006.2414.2007.1797.1860.5570.257
14FTG SS	2	1	2	3	1 2 3 1 2 3 1 2 88 89	
						.0858.0713.0707.0597.0686.1073.1015.1103.1042.0891.0715.0599
14	255	255	255	255	255	1776.2006.2414.2007.1797.2200.3060.474
15FTM	2	1	2	3	1 2 3 1 2 3 1 2 28 29	
						.0858.0713.0707.0597.0686.1073.1015.1103.1042.0891.0715.0599
15	664	664	664	664	664	1776.2006.2414.2007.1797.2080.4890.303
16GMG	1	1	2	3	1 2 3 1 2 3 1 2 30	
						.0858.0713.0707.0597.0686.1073.1015.1103.1042.0891.0715.0599
16	499	499	499	499	499	1776.2006.2414.2007.1797.1680.5510.281
17GMM	1	1	2	3	1 2 3 1 2 3 1 2 31	
						.0858.0713.0707.0597.0686.1073.1015.1103.1042.0891.0715.0599
17	395	395	395	395	395	1776.2006.2414.2007.1797.2050.5520.243
18GMT ASRO	1	1	2	3	1 2 3 1 2 3 1 2 32	
						.0858.0713.0707.0597.0686.1073.1015.1103.1042.0891.0715.0599
18	226	226	266	266	266	1776.2006.2414.2007.1797.0880.6020.310
19GSE	-2	1	2	3	1 2 3 1 2 3 1 2 33 34 35	
						.0858.0713.0707.0597.0686.1073.1015.1103.1042.0891.0715.0599
19	118	118	118	118	118	1776.2006.2414.2007.1797.2370.4320.331
20GSM	2	1	2	3	1 2 3 1 2 3 1 2 36 34 37	
						.0858.0713.0707.0597.0686.1073.1015.1103.1042.0891.0715.0599
20	436	436	436	436	436	1776.2006.2414.2007.1797.1310.7250.144
21IC	2	1	2	3	1 2 3 1 2 3 1 2 39	
						.0858.0713.0707.0597.0686.1073.1015.1103.1042.0891.0715.0599
21	1484	1484	1484	1484	1484	1776.2006.2414.2007.1797.2200.1990.581
26RM SS	1	3	3	3	3 3 3 3 3 3 3 3 3 41	
						.0858.0713.0707.0597.0686.1073.1015.1103.1042.0891.0715.0599
26	236	236	236	236	236	1776.2006.2414.2007.1797 1.000
28STG	-2	3	3	3	3 3 3 3 3 3 3 3 3 75 76	
						.0858.0713.0707.0597.0686.1073.1015.1103.1042.0891.0715.0599
28	576	576	576	576	576	1776.2006.2414.2007.1797 1.000
29STS	2	3	3	3	3 3 3 3 3 3 3 3 3 90 82 83	
						.0858.0713.0707.0597.0686.1073.1015.1103.1042.0891.0715.0599
29	501	501	501	501	501	1776.2006.2414.2007.1797 1.000
30TM	1	1	2	3	1 2 3 1 2 3 1 2 16	
						.0858.0713.0707.0597.0686.1073.1015.1103.1042.0891.0715.0599
30	468	468	468	468	468	1776.2006.2414.2007.1797.7930.1300.077

32AE AFUN 1 4 4 4 4 4 4 4 4 4 4 4 43 44
 .0858.0713.0707.0597.0686.1073.1015.1103.1042.0891.0715.0599
 32 3292 3292 3292 3292 3292.1776.2006.2414.2007.1797.3333.3333.3333
 33AQ AFUN 1 4 4 4 4 4 4 4 4 4 4 4 45 46 47
 .0858.0713.0707.0597.0686.1073.1015.1103.1042.0891.0715.0599
 33 714 714 714 714 714.1776.2006.2414.2007.1797.3333.3333.3333
 34ASE AFUN 1 4 4 4 4 4 4 4 4 4 4 4 48 49
 .0858.0713.0707.0597.0686.1073.1015.1103.1042.0891.0715.0599
 34 316 316 316 316 316.1776.2006.2414.2007.1797.3333.3333.3333
 35AT AFUN 1 4 4 4 4 4 4 4 4 4 4 4 50 51 52
 .0858.0713.0707.0597.0686.1073.1015.1103.1042.0891.0715.0599
 35 1981 1981 1981 1981 1981.1776.2006.2414.2007.1797.3333.3333.3333
 36AX AFUN 1 4 4 4 4 4 4 4 4 4 4 4 53 54 55
 .0858.0713.0707.0597.0686.1073.1015.1103.1042.0891.0715.0599
 36 430 430 430 430 430.1776.2006.2414.2007.1797.3333.3333.3333
 38TD AFUN 1 4 4 4 4 4 4 4 4 4 4 4 56 57 58
 .0858.0713.0707.0597.0686.1073.1015.1103.1042.0891.0715.0599
 38 479 479 479 479 479.1776.2006.2414.2007.1797.3333.3333.3333

1RTC	ORL	10	10	20	20	40	0	1
2RTC	GL	10	10	20	20	40	0	1
3RTC	SD	10	10	20	20	40	0	1
4BEE1	ORL	10	10	0	700	25		
5BEE2	ORL	10	10	0	700	25		
6BEE3	ORL	10	10	0	320	15		
7BEE1	GL	10	10	0	700	25		
8BEE2	GL	10	10	0	700	25		
9BEE3	GL	10	10	0	320	15		
10BEE1	SD	10	10	0	700	25		
11BEE2	SD	10	10	0	700	25		
12BEE3	SD	10	10	0	320	15		
13BEE1	MEM	10	10	0	700	25		
14BEE2	MEM	10	10	0	700	25		
15BEE3	MEM	10	10	0	320	15		
16CDP	6093	3.	2.	11	22	29.	0.	10.
17(UNUSED)		100	0	0	.00199999999999999999			
18CDP	6070	5.	30.	45	47	33.	0.	5.
19CDP	605Y	0.	0.	5	10	33.	0.	5.
20CDP	603V	15.	40.	5	10	140.	0.	5.
21CDP	604E	15.	40.	26	27	100.	0.	5.
22CDP	603X	15.	40.	1	4	140.	0.	5.
23CDP	603W	15.	40.	1	2	140.	0.	5.
24CDP	603Z	15.	40.	1	2	95.	0.	5.
25CDP	604F	15.	40.	3	4	45.	0.	5.
26CDP	6377	9.	15.	11	18	55.	0.	5.
27CDP	6376	4.	1.	17	23	60.	0.	10.
28CDP	6027	14.	15.	9	15	55.	0.	5.
29CDP	6108	4.	1.	15	21	60.	0.	10.
30CDP	6400	10.	10.	6	13	60.	0.	5.

MADE-UP MIN,MAX
 MADE-UP MIN,MAX

31CDP 6115	10.	10.	4	8	60.	0.	5.	
32CDP 6378	10.	10.	5	6	60.	0.	5.	
33CDP 606C	1.	1.	5	10	16.	0.	5.	MADE-UP DATA
34CDP 8563	1.	1.	27	28	20.	0.	20.	OPD VARIES.
35CDP 8565	1.	1.	5	10	30.	0.	10.	MADE-UP DATA
36CDP 606B	10.	0.	9	10	20.	0.	5.	
37CDP 8564	1.	1.	17	20	27.	0.	20.	OPD VARIES.
38(UNUSED)	100	0	0	.00199999999999999999				
39CDP 6073	5.	21.	5	10	40.	0.	5.	MADE-UP DATA
40CDP 605X	5.	21.	5	10	40.	0.	5.	MADE-UP DATA
41CDP 606D	15.	6.	6	7	35.	0.	5.	
42(UNUSED)	100	0	0	.00199999999999999999				
43CDP 6218	5.	7.	31	32	10	0	5	
44CDP 6515	7	25	8	16	55	0	3	FIRSTA VARIES.
45CDP 6220	5	7	14	18	10	0	5	
46CDP 6240	10	0	11	13	91	0	5	
47CDP 6245	10	25	4	5	100	0	5	FIRSTA VARIES.
48CDP 6215	5	7	5	8	10	0	5	
49CDP 6530	5	20	11	12	46	1	10	
50CDP 6219	5	7	41	43	10	0	5	
51CDP 6239	10	0	33	36	91	0	5	
52CDP 6244	10	25	10	11	100	0	5	FIRSTA VARIES.
53CDP 6221	5	7	8	9	10	0	5	
54CDP 6241	10	0	7	8	91	0	5	
55CDP 6246	10	25	1	2	100	0	5	FIRSTA VARIES.
56CDP 6222	5	7	8	11	10	0	5	
57CDP 6242	10	0	5	9	69	0	5	
58CDP 6521	5	12	6	7	29	0	5	
59(UNUSED)	100	0	0	.00199999999999999999				
60CDP 605A	8.	28.	7	9	50.	0.	5.	
61CDP 605B	3.	9.	6	8	15.	0.	5.	
62CDP 605C	2.	15.	6	8	20.	0.	5.	
63CDP 6161	15.	25.	5	7	75.	0.	5.	
64CDP 604A	5.	20.	1	2	50.	0.	5.	
65CDP 604B	2.	15.	1	2	20.	0.	5.	
66CDP 604C	1.	5.	7	8	15.	0.	20.	
67CDP 604D	1.	6.	1	2	20.	0.	5.	
68CDP 602A	5.	12.	10	13	15.	0.	5.	
69CDP 602B	10.	28.	9	13	50.	0.	5.	
70CDP 602C	1.	10.	8	11	10.	0.	5.	
71CDP 602D	2.	10.	8	11	20.	0.	5.	
72CDP 603A	3.	15.	4	6	25.	0.	5.	
73CDP 603B	1.	15.	4	5	20.	0.	5.	
74(UNUSED)	100	0	0	.00199999999999999999				
75CDP 6015	10.	4.	5	10	30.	0.	5.	MADE-UP DATA
76CDP 6195	1.	0.	5	10	15.	0.	5.	MADE-UP DATA
77CDP 6196	1.	0.	5	10	10.	0.	5.	MADE-UP DATA
78CDP 6198	1.	0.	5	10	15.	0.	5.	MADE-UP DATA
79CDP 6118	2.	2.	5	10	35.	0.	5.	MADE-UP DATA

80CDP 541U	0.	0.	5	10	15.	0.	5.	MADE-UP DATA
81CDP 4573	1.	0.	5	10	20.	0.	5.	MADE-UP DATA
82CDP 6172	20.	15.	5	10	40.	0.	5.	MADE-UP DATA
83CDP 6402	10.	10.	5	10	20.	0.	5.	MADE-UP DATA
84CDP 6401	0.	0.	5	10	20.	0.	5.	MADE-UP DATA
85CDP 6460	10.	10.	5	10	20.	0.	5.	MADE-UP DATA
86CDP 600A	10.	10.	5	10	30.	0.	5.	MADE-UP DATA
87(UNUSED)	100	0	0	.001999999999999999				
88CDP 5200	25.	15.	86	89	29.	0.	5.	
89CDP 6337	10.	0.	11	12	60.	0.	15.	
90CDP 6342	6.	10.	5	10	29.	0.	5.	MADE-UP DATA
91(UNUSED)	100	0	0	.001999999999999999				
92CDP 6041	12.	15.	12	13	80.	0.	20.	
93(UNUSED)	100	0	0	.001999999999999999				
94CDP 6131	13.	25.	14	19	110.	0.	5.	
95(UNUSED)	100	0	0	.001999999999999999				
96CDP 6079	4.	3.	18	31	40.	0.	20.	
97(UNUSED)	100	0	0	.001999999999999999				
98CDP 6289	4.	3.	20	22	40.	0.	20.	
99(UNUSED)	100	0	0	.001999999999999999				

APPENDIX B

PROGRAM LISTING

C

C MAIN

```

CHARACTER PLACE*4,CNAM1*4,CNAM2*4,STYP1*4,STYP2*4
INTEGER WHR,COR,FLEA,CHS,PORH,GULP,BE,ORL,GL,SD,MEM,YYY,FLK,BACK
INTEGER YYF, CPUTIME
REAL NAVY,MINS,MAXS
COMMON/SCOM1/ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,NCLNR,
1 NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TEXT,TNOW,XX(100)
COMMON/NAVAL0/ORL,GL,SD,MEM,COR,FLEA,NLON,CHS,MARI,PORH,GULP,NCRD2
COMMON/NAVAL1/IECHO,ICASE,MM,LASTA,KKK,NST,NST1,MSTUD,NSTUD,NDNAVY
COMMON/NAVAL2/FLER,TBR,TBS,FACTOR,STUD(4),NAVY(12000)
COMMON/NAVAL3/BE(49),NASCHD(49,11),YD(49,5),NROUTE(49,11),IFAT
COMMON/NAVAL4/FM(49,12),DOWF(49,5),BF(49,3),PR(49)
COMMON/NAVAL5/IMET,WHR(98),NCLSIZE(98),YYY(98),YYF(98),LIMCOU
COMMON/NAVAL6/INTRAN,FLK(98),BACK(98),MAXSYS,MAXAUX,IFLER,IHST
COMMON/NAVAL7/FIRSTA(98),OPD(98),ADV(98),FAIL(98),SBACK(98)
COMMON/NAVAL8/MINS(98),MAXS(98),WTIME(98)
COMMON/NAVAL9/PLACE(13),CNAM1(98),CNAM2(98),STYP1(49),STYP2(49)

```

CNAV10

CNAV11

CNAV12

CNAV13

```

DIMENSION NSET(25000)
COMMON QSET(25000)
EQUIVALENCE (NSET(1),QSET(1))

```

C IMET = CPUTIME(0) (Initiation of CPU time function)

```

NNSET=25000
NDNAVY = 12000
NCRDR=5
NCRD2=5
NPRNT=6

```

C NTAPE must be defined to meet SLAM processor requirements.

```

NTAPE=7
CALL SLAM
STOP
END

```

BLOCK DATA

```

INTEGER WHR,YD,YYY,FLK,BACK
INTEGER YYF
REAL MINS,MAXS

```

```

CHARACTER PLACE*4,CNAM1*4,CNAM2*4,STYP1*4,STYP2*4
COMMON/NAVAL5/IMET,WHR(98),NCLSIZE(98),YYY(98),YYF(98),LIMCOU
COMMON/NAVAL6/INTRAN,FLK(98),BACK(98),MAXSYS,MAXAUX,IFLER,IHST

```



```

COMMON/NAVAL8/MINS(98),MAXS(98),WTIME(98)
COMMON/NAVAL9/PLACE(13),CNAM1(98),CNAM2(98),STYP1(49),STYP2(49)
CNAV10
CNAV11
CNAV12
CNAV13
  DATA WTIME/98*0./,YYY/98*0/,FLK/98*0/,BACK/98*0/NCLSZE/98*0/
  DATA WHR/1,2,3,3*1,3*2,3*3,3*4,83*0/
  DATA PLACE/' ORL',' GL',' SD',' MEM',' COR',' FLEA',' NLON',
1    'CHAS',' MARI',' PORH',' GULP',' BEE',' ASCH'/
  DATA YYF/98*0/
  END

  SUBROUTINE INTLC
  CHARACTER PLACE*4,CNAM1*4,CNAM2*4,STYP1*4,STYP2*4
  INTEGER WHR,COR,FLEA,CHS,PORH,GULP,BE,ORL,GL,SD,MEM,YYY,FLK,BACK
  INTEGER YYF
  REAL NAVY,MINS,MAXS
  COMMON/SCOM1/ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,NCLNR,
1 NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TEXT,TNOW,XX(100)
  COMMON/XCOM7/ MEQT,MHIST,MCELS,MCLCT,MSTAT,MATRB,MFELS,MPLT,
1MVARP,MSTRM,MACT,MNODE,MITYP,HIVAL,MENTR,MRSC,MGAT,MSCND
  COMMON/NAVAL0/ORL,GL,SD,MEM,COR,FLEA,NLON,CHS,MARI,PORH,GULP,NCRD2
  COMMON/NAVAL1/IECHO,ICASE,MM,LASTA,KKK,NST,NST1,MSTUD,NSTUD,NDNAVY
  COMMON/NAVAL2/FLER,TBR,TBS,FACTOR,STUD(4),NAVY(12000)
  COMMON/NAVAL3/BE(49),NASCHD(49,11),YD(49,5),NROUTE(49,11),IFAT
  COMMON/NAVAL4/FM(49,12),DOWF(49,5),BF(49,3),PR(49)
  COMMON/NAVAL5/IMET,WHR(98),NCLSZE(98),YYY(98),YYF(98),LIMCOU
  COMMON/NAVAL6/INTRAN,FLK(98),BACK(98),MAXSYS,MAXAUX,IFLER,IHST
  COMMON/NAVAL7/FIRSTA(98),OPD(98),ADV(98),FAIL(98),SBACK(98)
  COMMON/NAVAL8/MINS(98),MAXS(98),WTIME(98)
  COMMON/NAVAL9/PLACE(13),CNAM1(98),CNAM2(98),STYP1(49),STYP2(49)
CNAV10
CNAV11
CNAV12
CNAV13
  IFAT = 0
  NST = 16
  NST1 = NST - 1
  KKK=0
  IFLER = 0
  MAXAUX = 0
  MAXSYS = 0
  INTRAN = 0
  NSTUD = 4
  CALL SETAA(NSTUD,NAVY,MSTUD,NDNAVY)
  IF(IECHO.NE.0) WRITE(NPRNT,9016)NAVY
  READ(NCRD2,9002)ORL,GL,SD,MEM,COR,FLEA,NLON,CHS,MARI,PORH,GULP
  LASTA=NST1+ORL+GL+SD+MEM+COR+FLEA+NLON+CHS+MARI+PORH+GULP

```

```

LIMCOU = MAX0(LASTA,25)
IF (LASTA.GT.MFILS-2) CALL FAT(9)
READ(NCRD2,9013)IECHO,ICASE,IHST,FACTOR,FSTR,TBR,FSTS,TBS,
1  FLER,TBE
IF(TBR.LE.0.) CALL FAT(1)
IF(TBS.LE.0.) CALL FAT(2)
IF(FACTOR.LE.0.) CALL FAT(11)
IF(IECHO.NE.0) WRITE(NPRNT,9003)ORL,GL,SD,MEM,COR,FLEA,NLON,
1  CHS,MARI,PORH,GULP,ICASE,FACTOR,FSTR,TBR,FSTS,TBS,FLER
FLER = FLER / 100.
C READ STUDENT DATA:
IF(IECHO.NE.0) WRITE (NPRNT, 9008)
MCLCT1 = MCLCT + 1
DO 20 I = 1, 1000
IF(I.GT.MCLCT1) CALL FAT(7)
READ(NCRD2,9004)NO,STYP1(I),STYP2(I),BE(I),
1  (NROUTE(I,J),J=1,11),(NASCHD(I,J),J=1,10),PR(I)
IF (NO.EQ.0) GO TO 21
IBE = IABS(BE(I))
IF(IBE.LT.1.OR.IBE.GT.3) CALL FAT(3)
DO 3 J = 1, 11
IBE = NROUTE(I,J)
IF(IBE.LT.1.OR.IBE.GT.4) CALL FAT(4)
3 CONTINUE
DO 10 J = 1, 10
NASH = NASCHD(I,J)
IF (NASH.EQ.0) GO TO 11
IF(NASH.LT.NST.OR.NASH.GT.99) CALL FAT(5)
10 CONTINUE
11 NASCHD(I,11) = J - 1
IF(IECHO.NE.0) WRITE(NPRNT,9005)I,NO,STYP1(I),STYP2(I),BE(I),
1  (NROUTE(I,J),J=1,11),(NASCHD(I,J),J=1,11),PR(I)
READ(NCRD2,9010)(FM(I,J),J=1,12)
READ(NCRD2,9012)(YD(I,J),J=1,5),(DOWF(I,J),J=1,5),(BF(I,J),J=1,3)
20 CONTINUE
21 MM = I - 1
IF(MM.GT.MCLCT) CALL FAT(7)
IF(IECHO.NE.0) WRITE (NPRNT,9014)
DO 9 I = 1, MM
IF(IECHO.NE.0) WRITE (NPRNT, 9011) I, (YD(I,J), J = 1, 5),
1  (BF(I,J),J=1,3),(FM(I, J), J = 1, 12), (DOWF(I, J), J = 1, 5)
DOWS = 0
DO 4 J = 1, 5
YD(I,J) = YD(I,J) * FACTOR
IF(YD(I,J).LE.0) WRITE (NPRNT,9020) I, J
DOWS = DOWS + DOWF(I,J)
4 DOWF(I,J) = DOWF(I,J) * 3/13.
IF(ABS(DOWS-1.).GT.0.01) WRITE (NPRNT, 9021) I
DOWS = 0.

```

```

DO 5 J = 1, 12
5 DOWS = DOWS + FM(I,J)
  IF(ABS(DOWS-1).GT.0.01) WRITE (NPRNT, 9022) I
  DOWS = 0.
DO 6 J = 1, 3
6 DOWS = DOWS + BF(I,J)
  IF(ABS(DOWS-1).GT.0.01) WRITE(NPRNT, 9023) I
9 CONTINUE
C READ COURSE DATA:
  IF(IECHO.NE.0) WRITE (NPRNT, 9009)
  DO 30 I = 1, LASTA
    READ(NCRD2, 9006) NO, CNAM1(I),CNAM2(I),FAIL(I),SBACK(I),
1      MINS(I), MAXS(I), ADV(I), FIRSTA(I), OPD(I)
    IF(IECHO.NE.0)WRITE(NPRNT,9007)I,NO,CNAM1(I),CNAM2(I),FAIL(I),
1      SBACK(I), MINS(I), MAXS(I), ADV(I), FIRSTA(I), OPD(I)
    IF(OPD(I).LE.0..AND.(I.LE.3.OR.I.GE.NST)) CALL FAT(6)
    IF(MINS(K).GT.MAXS(K)) CALL FAT(10)
    SBACK(I) = (SBACK(I) + FAIL(I)) / 100.
    FAIL(I) = FAIL(I) / 100.
    MINS(I) = MINS(I) * FACTOR
    MAXS(I) = MAXS(I) * FACTOR
    IF(MAXS(I).LE.0) WRITE (NPRNT, 9024) I
30 CONTINUE
  DO 40 I = 4, NST1
    MAXS(I) = FLOAT(INT(MAXS(I) + 0.5))
    J = IFIX(MAXS(I))
40 CALL ALTER(I-3, J)
    ORL=ORL + NST1
    GL=ORL+GL
    SD=GL+SD
    MEM = MEM + SD
    COR = COR + MEM
    FLEA = FLEA + COR
    NLON = NLON + FLEA
    CHS = CHS + NLON
    MARI = MARI + CHS
    PORH = PORH + MARI
    GULP = GULP + PORH
    I = NST
    IF(ORL.NE.NST1) CALL WHERE(I,ORL,1)
    IF(GL.NE.ORL) CALL WHERE(I,GL, 2)
    IF(SD.NE.GL) CALL WHERE(I,SD, 3)
    IF(MEM.NE.SD) CALL WHERE(I,MEM,4)
    IF(COR.NE.MEM) CALL WHERE(I,COR,5)
    IF(FLEA.NE.COR) CALL WHERE(I,FLEA,6)
    IF(NLON.NE.FLEA) CALL WHERE(I,NLON,7)
    IF(CHS.NE.NLON) CALL WHERE(I,CHS,8)
    IF(MARI.NE.CHS) CALL WHERE(I,MARI,9)
    IF(PORH.NE.MARI) CALL WHERE(I,PORH,10)

```

```

      IF(GULP.NE.PORH) CALL WHERE(I,GULP,11)
      IF(IECHO.NE.0) WRITE (NPRNT,9015) WHR
      IF(IECHO.NE.0) CALL PRECHK
      IF(IFAT.GT.0) GO TO 8000
C  SCHEDULE FIRST LOCKSTEP CLASSES:
      DO 69 K = 1, 3
      ATRIB(1) = FLOAT(K)
69  CALL SCHDL(6,FIRSTA(K),ATRIB)
      DO 70 K = NST, LASTA
      ATRIB(1) = FLOAT(K)
70  CALL SCHDL(6, FIRSTA(K), ATRIB)
      CALL SCHDL(1,FSTR,ATRIB)
      CALL SCHDL(2,FSTS,ATRIB)
      CALL SCHDL(3, TBE, ATRIB)
      CALL SCHDL(4,0.5,ATRIB)
      RETURN
8000 WRITE (NPRNT, 9025) IFAT
      STOP
9000 FORMAT (16I5)
9001 FORMAT (//,1X'DATA:',8X,'A COURSE SEQUENCES BY STUDENT TYPE:',/,
1      (1X, I5, 5X, 10I5, I8))
9002 FORMAT (16I5)
9003 FORMAT(///,' ORLANDO =',I3,/, ' GREAT LAKES =',
1      I3,/, ' SAN DIEGO =',I3,/,
2      ' MEMPHIS =',I3,/, ' CORRY STATION =',I3,/, ' FLEASWTRAC =',I3,
3/, ' NEW LONDON =',I3,/, ' CHARLESTON =',I3,/, ' MARE ISLAND =',I3,/,
4' PORT HUENEME =',I3,/, ' GULF PORT =',I3,/, ' ICASE =',I2,/, ' SCAL
5ING FACTOR =',F5.2,/, ' TIME UNTIL FIRST REPORT =',F8.1,/, ' TIME BE
6TWEEN REPORTS =',F8.1,/, ' TIME UNTIL FIRST SNAPSHOT =',F8.1,/,
7 ' TIME BETWEEN SNAPSHOTS =',F8.1,/, ' FLEET RETURNEES =',F6.2,'%')
9004 FORMAT (I3,2A4,I3,11I2,10I3,F5.0)
9005 FORMAT (1X,2I3,2X,2A4,I5,2X,11I3,2X,10I3,I5,F8.1)
9006 FORMAT (I3, 2A4, 7F5.0)
9007 FORMAT (1X, 2I3, 2X, 2A4, 2F6.2, 2F7.2, 3F8.1)
9008 FORMAT(//,20X,'STUDENT DATA:',//,9X,'TYPE BE NROUTE',
1      11X, 7X,10X,'A COURSES:',27X,'BEE PRI')
9009 FORMAT(///,20X,'COURSE DATA:',/,10X,'CNAME FAIL SBACK',3X,
1      'MIN MAX LENGTH FRST BET',/,19X,'(%) (%)')
9010 FORMAT (10X, 12F5.0)
9011 FORMAT (1X, I2, 1X, 5F5.0, 1X, 3F5.2, 1X, 12F5.2, 1X, 5F5.2)
9012 FORMAT (5X, 5F5.0, 5F5.0, 3F5.0)
9013 FORMAT (/,3I5, 12F5.0)
9014 FORMAT (/,5X,'Negative "BE" indicates that A school comes before B
1EE.',//, 5X,'TOTAL YEARLY DEMAND',7X,
2      'BASE FACTORS MONTH FACTORS',48X,'DAY-OF-WEEK FACTORS')
9015 FORMAT (1X,'WHR:',/,(1X,10I3))
9016 FORMAT (///,1X,'NAVY:',/,(10X,20F6.0))
9020 FORMAT (' ***** WARNING: No arrivals of student type',I3,
1      ' during year',I5,' after scaling.')

```



```

9021 FORMAT (' ***** WARNING: Day-Of-Week factors (DOWF) for stu
1dent type',I3,' do not add to 1.000.')
9022 FORMAT (' ***** WARNING: Month factors (FM) for student typ
1e',I3,' do not add to 1.000.')
9023 FORMAT (' ***** WARNING: Base factors (BF) for student type
1',I3,' do not add to 1.000.')
9024 FORMAT (' ***** WARNING: Maximum number of students per cla
1ss = 0 after scaling for course',I3)
9025 FORMAT (///,20x,I5,' FATAL ERRORS DETECTED.',/,24x,'SIMULATION NOT
1 ATTEMPTED')
END

```

```

SUBROUTINE WHERE(I, N, J)
INTEGER WHR
COMMON/NAVAL5/IMET,WHR(98),NCLSIZE(98),YYY(98),YYF(98),LIMCOU
DO 10 I = 1, N
10 WHR(I) = J
I = N + 1
RETURN
END

```

```

SUBROUTINE FAT(I)
INTEGER BE
COMMON/NAVAL3/BE(49),NASCHD(49,11),YD(49,5),NROUTE(49,11),IFAT
COMMON/SCOM1/ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,NCLNR,
1 NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TEXT,TNOW,XX(100)
COMMON/XCOM7/ MEQT,MHIST,MCELS,MCLCT,MSTAT,MATRB,MFELS,MPLT,
1MVARP,MSTRM,MACT,MNODE,MITYP,HIVAL,MENTR,MRSC,MGAT,MSCND
COMMON/NAVAL1/IECHO,ICASE,MM,LASTA,KKK,NST,NST1,MSTUD,NSTUD,NDNAVY
IFAT = IFAT + 1
GO TO (10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110), I
10 WRITE (NPRNT, 9001)
RETURN
20 WRITE (NPRNT, 9002)
RETURN
30 WRITE (NPRNT, 9003)
RETURN
40 WRITE (NPRNT, 9004)
RETURN
50 J = MFELS - 2
WRITE (NPRNT, 9005) J
RETURN
60 WRITE (NPRNT, 9006)
RETURN
70 WRITE (NPRNT, 9007) MCLCT
RETURN
80 J = MFELS - 1
WRITE (NPRNT, 9008) J
RETURN

```

```

90 J = MFILS - NST - 1
   WRITE (NPRNT, 9009) J
   RETURN
100 WRITE (NPRNT, 9010)
   RETURN
110 WRITE (NPRNT, 9011)
   RETURN
9001 FORMAT (' ***** ERROR: Time between reports (TBR) must be g
1reater than zero. ')
9002 FORMAT (' ***** ERROR: Time between snapshots (TBS) must be
1 greater than zero. ')
9003 FORMAT (' ***** ERROR: BE(I) must be +/- 1, 2 or 3. ')
9004 FORMAT (' ***** ERROR: All elements of NROUTE must be 1, 2,
1 3 or 4. ')
9005 FORMAT (' ***** ERROR: A-courses must be numbered between 1
16 and', I5, ' (inclusive). ')
9006 FORMAT (' ***** ERROR: Lockstep courses must have positive'
1 , ' time between class convenings. ')
9007 FORMAT (' ***** ERROR: A maximum of', I5, ' student types are
1 allowed. ')
9008 FORMAT (' ***** ERROR: A maximum of', I5, ' courses are allow
1ed (including RTC's and BEE's). ')
9009 FORMAT (' ***** ERROR: Too many A courses in simulation spe
1cifications cards. ORL+GL+SD+MEM+... must be less than or equal t
1o', I5, '. ')
9010 FORMAT (' ***** ERROR: Maximum class size < minimum size. ')
9011 FORMAT (' ***** ERROR: Scaling factor (FACTOR) must be grea
1ter than zero. ')
   END

```

SUBROUTINE PRECHK

C Subroutine PRECHK performs a preliminary analysis of the data to look
C for any obvious bottlenecks.

```

   INTEGER WHR, BE
   CHARACTER PLACE*4, CNAM1*4, CNAM2*4, STYP1*4, STYP2*4, A*4
   REAL NAVY, MINS, MAXS
   COMMON/SCOM1/ATTRIB(100), DD(100), DDL(100), DTNOW, II, MFA, MSTOP, NCLNR,
1 NCRDR, NPRNT, NNRUN, NNSET, NTAPE, SS(100), SSL(100), TEXT, TNOW, XX(100)
   COMMON/NAVAL1/IECHO, ICASE, MM, LASTA, KKK, NST, NST1, MSTUD, NSTUD, NDNVY
   COMMON/NAVAL2/FLER, TBR, TBS, FACTOR, STUD(4), NAVY(12000)
   COMMON/NAVAL3/BE(49), NASCHD(49, 11), YD(49, 5), NROUTE(49, 11), IFAT
   COMMON/NAVAL4/FM(49, 12), DOWF(49, 5), BF(49, 3), PR(49)
   COMMON/NAVAL5/IMET, WHR(98), NCLSZ(98), YYY(98), YYF(98), LIMCOU
   COMMON/NAVAL7/FIRSTA(98), OPD(98), ADV(98), FAIL(98), SBACK(98)
   COMMON/NAVAL8/MINS(98), MAXS(98), WTIME(98)
   COMMON/NAVAL9/PLACE(13), CNAM1(98), CNAM2(98), STYP1(49), STYP2(49)
   X = FLER * 100.0
   WRITE (NPRNT, 9000) X

```

```

DO 100 NS = 1, MM
X = (YD(NS,1)+YD(NS,2)+YD(NS,3)+YD(NS,4)+YD(NS,5))/5
Y = X / FACTOR
X40 = 0.0
DO 30 L = 1, 3
C (FOR EACH BASE):
  X30 = X * BF(NS,L)
  WTIME(L) = WTIME(L) + X30
  X30 = X30 * (1 - FAIL(L)) * (1. + FLER)
  IF(BE(NS).GT.0.AND.STYP2(NS).NE.'AFUN') CALL BEECHK(NS,L,X30)
30 X40 = X40 + X30
C X40 = rate of student type NS after BEE (only after basic training
C   for A-before-BEE and AFUN types) from all bases.
  X = X40
  CALL ASCHK(NS,1,X)
  IF(STYP2(NS).EQ.'AFUN') THEN
    L = WHR(NASCHD(NS,1))
    CALL BEECHK(NS,L,X)
  END IF
  NASC11 = NASCHD(NS,11)
  IF(NASC11.LT.2) GO TO 51
  DO 50 NA = 2, NASC11
50 CALL ASCHK(NS,NA,X)
51 L = WHR(NASCHD(NS,NASC11))
  IF(BE(NS).LT.0) CALL BEECHK(NS,L,X)
  X = X / FACTOR
100 WRITE (NPRNT,9001) NS, STYP1(NS),STYP2(NS), Y, X

  WRITE (NPRNT, 9002)
  DO 200 NA = 1, LASTA
  IF(NA.GE.NST.OR.NA.LE.3) THEN
    X = MAXS(NA) * 260. / OPD(NA)
  ELSE
    X = MAXS(NA) * 260. / ADV(NA)
  END IF
  IF(X.LT.WTIME(NA)) A = '*****'
  X = X / FACTOR
  WTIME(NA) = WTIME(NA) / FACTOR
  Y = FAIL(NA) * 100.0
  IF(NA.GT.3) THEN
    Z = AMIN1(WTIME(NA), X) * (1.00 - FAIL(NA))
    WRITE (NPRNT, 9003) NA, CNAM1(NA),CNAM2(NA), X, WTIME(NA), Y, Z,A
  ELSE
    Z = WTIME(NA) * (1.00 - FAIL(NA))
    WRITE (NPRNT, 9004) NA, CNAM1(NA),CNAM2(NA), WTIME(NA), Y, Z
  END IF
  A = ' '
200 WTIME(NA) = 0.0
  WRITE (NPRNT, 9005)

```

```

      RETURN
9000 FORMAT (1H1,///,20X,'PRE-EXECUTION DATA ANALYSIS:',10X,'Fleet retu
1rnees =',F6.2,'% of basic training graduates.',///,10X,
2      'STEADY STATE STUDENT I/O RATES (if no bottlenecks)',/,10X,
3      'STUDENT TYPE      IN      OUT      (students per year)')
9001 FORMAT (1H , 3X, I5,2X,2A4,F12.1, F8.1)
9002 FORMAT (1H , ///,' COURSE MAXIMUM CAPACITIES AND FEED RATES (assum
ling no previous bottlenecks restricting feed):',/,
2      9X,'COURSE CAPACITY FEED RATE %FAIL GRAD RATE (studen
3ts per year)')
9003 FORMAT (1H , 1X,I4,2X,2A4,F8.1,F10.1,2F10.1,4X,A4)
9004 FORMAT (1H , 1X,I4,2X,2A4,' UNLIMITED',F8.1,2F10.1)
9005 FORMAT (1H ,//,' **** feed rate exceeds capacity.')
      END

```

```

      SUBROUTINE BEECHK(NS,L,X30)
C   BEE part of preliminary analysis for all BEE courses for student type
C   NS at base L.
      INTEGER BE
      COMMON/NAVAL3/BE(49),NASCHD(49,11),YD(49,5),NROUTE(49,11),IFAT
      COMMON/NAVAL7/FIRSTA(98),OPD(98),ADV(98),FAIL(98),SBACK(98)
      COMMON/NAVAL8/MINS(98),MAXS(98),WTIME(98)
      N = 3 * NROUTE(NS,L) + 1
      M = IABS(BE(NS)) + N - 1
      DO 10 N = N, M
      WTIME(N) = WTIME(N) + X30
10  X30 = X30 * (1 - FAIL(N))
      RETURN
      END

```

```

      SUBROUTINE ASCHK(NS,NA,X)
C   A-course preliminary analysis for NAtH course in sequence for type NS.
      COMMON/NAVAL3/BE(49),NASCHD(49,11),YD(49,5),NROUTE(49,11),IFAT
      COMMON/NAVAL7/FIRSTA(98),OPD(98),ADV(98),FAIL(98),SBACK(98)
      COMMON/NAVAL8/MINS(98),MAXS(98),WTIME(98)
      N = NASCHD(NS,NA)
      WTIME(N) = WTIME(N) + X
      X = X * (1 - FAIL(N))
      RETURN
      END

```

```

      SUBROUTINE EVENT(J)
      CHARACTER PLACE*4,CNAM1*4,CNAM2*4,STYP1*4,STYP2*4
      INTEGER WHR,COR,FLEA,CHS,PORH,GULP,BE,ORL,GL,SD,MEM,YYY,FLK,BACK
      INTEGER YYF
      REAL NAVY,MINS,MAXS
      COMMON/SCOM1/ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,NCLNR,
1  NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TEXT,TNOW,XX(100)
      COMMON/NAVAL0/ORL,GL,SD,MEM,COR,FLEA,NLON,CHS,MARI,PORH,GULP,NCRD2

```



```

COMMON/NAVAL1/IECHO, ICASE, MM, LASTA, KKK, NST, NST1, MSTUD, NSTUD, NDNNAVY
COMMON/NAVAL2/FLER, TBR, TBS, FACTOR, STUD(4), NAVY(12000)
COMMON/NAVAL3/BE(49), NASCHD(49, 11), YD(49, 5), NROUTE(49, 11), IFAT
COMMON/NAVAL4/FM(49, 12), DOWF(49, 5), BF(49, 3), PR(49)
COMMON/NAVAL5/IMET, WHR(98), NCLSIZE(98), YYY(98), YYF(98), LIMCOU
COMMON/NAVAL6/INTRAN, FLK(98), BACK(98), MAXSYS, MAXAUX, IFLER, IHST
COMMON/NAVAL7/FIRSTA(98), OPD(98), ADV(98), FAIL(98), SBACK(98)
COMMON/NAVAL8/MINS(98), MAXS(98), WTIME(98)
COMMON/NAVAL9/PLACE(13), CNAM1(98), CNAM2(98), STYP1(49), STYP2(49)

CNAV10
CNAV11
CNAV12
CNAV13
    IF(J.GT.LIMCOU) GO TO 200
    IF(J.GE.14) GO TO 100
    GO TO (1,2,3,4,5,6,7,8,9), J
1  CALL REPORT
   RETURN
2  CALL STATUS
   RETURN
3  CALL ERRORA(3)
   RETURN
4  CALL RCRT
   RETURN
5  CALL ERRORA(23)
   RETURN
6  K = IFIX(ATTRIB(1))
C  ATTRIB(1) holds course number.
   CALL CLASS(K)
   RETURN
7  CALL ERRORA(4)
   RETURN
8  CALL ROUTE
   RETURN
9  CALL TDST
   RETURN
100 CONTINUE
C  Events 14-25: A STUDENT HAS SECURED A CHAIR FOR BEE.
   I = J - 10
   YYY(I)=YYY(I)+1
   NCLSIZE(I) = NCLSIZE(I) + 1
   WTIME(I) = WTIME(I) + TNOW - NAVY(IFIX(ATTRIB(1))) + 4)
   BEETIM=ADV(I)
   CALL FAILUR(I,RN)
   IF (RN.LE.SBACK(I)) BEETIM = BEETIM * 1.1
   IF (RN.LE.FAIL(I)) BEETIM = BEETIM * 0.6 /1.1
   I = I + LIMCOU
   CALL SCHDL (I, BEETIM, ATTRIB)
   RETURN

```

```

C  ENDS OF CLASSES:
200 J = J - LIMCOU
C  Counting and dumping of failures should not be done here because
C  BEE's need to free their chairs first.
    NCLSIZE(J) = NCLSIZE(J) - 1
    IF (J.LT.NST) GO TO 212
    CALL CLASSE(J)
    RETURN
212 CONTINUE
    IR = J - 3
    IF(J.GT.3) CALL FREE(IR,1)
    CALL TRAVEL(J)
    RETURN
END

SUBROUTINE RCRT
C  Creates new recruits and puts them in line for basic trainings.
    CHARACTER PLACE*4,CNAM1*4,CNAM2*4,STYP1*4,STYP2*4
    INTEGER WHR,COR,FLEA,CHS,PORH,GULP,BE,ORL,GL,SD,MEM,YYY,FLK,BACK
    INTEGER YYF,CPUTIME
    REAL NAVY,MINS,MAXS
    COMMON/SCOM1/ATTRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,NCLNR,
1  NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TEXT,TNOW,XX(100)
    COMMON/NAVAL0/ORL,GL,SD,MEM,COR,FLEA,NLON,CHS,MARI,PORH,GULP,NCRD2
    COMMON/NAVAL1/IECHO,ICASE,MM,LASTA,KKK,NST,NST1,MSTUD,NSTUD,NDNAVY
    COMMON/NAVAL2/FLER,TBR,TBS,FACTOR,STUD(4),NAVY(12000)
    COMMON/NAVAL3/BE(49),NASCHD(49,11),YD(49,5),NROUTE(49,11),IFAT
    COMMON/NAVAL4/FM(49,12),DOWF(49,5),BF(49,3),PR(49)
    COMMON/NAVAL5/IMET,WHR(98),NCLSIZE(98),YYY(98),YYF(98),LIMCOU
    COMMON/NAVAL6/INTRAN,FLK(98),BACK(98),MAXSYS,MAXAUX,IFLER,IHST
    COMMON/NAVAL7/FIRSTA(98),OPD(98),ADV(98),FAIL(98),SBACK(98)
    COMMON/NAVAL8/MINS(98),MAXS(98),WTIME(98)
    COMMON/NAVAL9/PLACE(13),CNAM1(98),CNAM2(98),STYP1(49),STYP2(49)
CNAV10
CNAV11
CNAV12
CNAV13
    CALL SCHDL(4,1.,ATTRIB)
    NCLNR1 = NCLNR - 1
    NYR = INT(TNOW / 260.)
    NDAY = INT(TNOW - 260 * NYR)
    MONTH = INT(NDAY * 12 / 260) + 1
    NDOW = INT(NDAY - NDAY / 5 * 5) + 1
    NYR = NYR + 1
    DO 30 K = 1, 3
    N = NNQ(NCLNR1)
    IF(N.NE.0) CALL ERRORA(5)
C  At each base generate the students that will arrive for the day.
C  They are randomized by a HVF(2) storage in file NCLNR1.  ATTRIB(1)

```

```

C      holds NS during this temporary storage.
      DO 20 NS = 1, MM
      X = YD(NS,NYR)*FM(NS,MONTH)*DOWF(NS,NDOW)*BF(NS,K)
      N = IFIX(X + DRAND(1))
      IF (N.LE.0) GO TO 20
      DO 10 L=1,N
      ATRIB(1) = FLOAT(NS)
      ATRIB(2) = DRAND(5)
10  CALL FILEM(NCLNR1,ATRIB)
20  CONTINUE
C  After all students for that base for the day have been generated
C      and filed randomly in NCLNR1 they are pulled out, given their
C      credentials and put in the RTC queue.
      N = NNQ(NCLNR1)
      IF(N.EQ.0) GO TO 30
      DO 25 I = 1, N
      CALL RMOVE(1,NCLNR1,ATRIB)
      NS = IFIX(ATRIB(1))
      STUD(1) = TNOW
      STUD(2) = FLOAT(NS)
      STUD(3) = 0.
      STUD(4) = TNOW
      ATRIB(1) = FLOAT(MSTUD)
      ATRIB(2) = PR(NS)
      IF(MSTUD.NE.MSTUD/4*4) CALL ERRORA(6)
      CALL PUTAA(NSTUD,NAVY,MSTUD,STUD)
      IF(MSTUD.NE.MSTUD/4*4) CALL ERRORA(7)
      CALL FILEM(K,ATRIB)
25  KKK = KKK + 1
30  CONTINUE
C      X = CPUTIME(IMET) / 100.0
C      WRITE (3, 9000) TNOW, ATRIB(1), ATRIB(2), STUD, X
      IF(MAXAUX.LT.MSTUD) MAXAUX = MSTUD
      IF(MAXSYS.LT.MFA) MAXSYS = MFA
      RETURN
9000  FORMAT (1X, F10.0, ' LAST RECRUIT TODAY: ', 2F10.4, 10X, 5F10.2)
9001  FORMAT (10X, '**** SORRY! TOO MANY CONCURRENT STUDENTS. ****',
1      'TIME =', F8.2)
      END

```

SUBROUTINE TRAVEL(J)

```

C  Subroutine TRAVEL(J) receives basic training and BEE output from
C      EVENT, scheduled in CLASS(1-3) and events 14-25.
C  J = the file number of the course just finished.
C  First any failures are dropped.
C  Basic training graduates are counted and routed to BEE (Event 8,
C      ROUTE) or their first A courses (Event 9, TDST) after any
C      travel time (DELAY) as applicable.
C  BEE graduates are routed to their first A course (Event 9, TDST)

```

C for regular types, second A course (Event 5, TDST) for AFUN
 C types or STATS for A before BEE types.
 C Cloning of basic training graduates to form fleet returnees is done
 C here.

C

```

      CHARACTER PLACE*4,CNAM1*4,CNAM2*4,STYP1*4,STYP2*4
      INTEGER WHR,COR,FLEA,CHS,PORH,GULP,BE,ORL,GL,SD,MEM,YYY,FLK,BACK
      INTEGER YYF
      REAL NAVY,MINS,MAXS
      COMMON/SCOM1/ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,NCLNR,
1  NCRDR,NPRNT,NNRUN,NNSE,NTAPE,SS(100),SSL(100),TEXT,TNOW,XX(100)

```

```

      COMMON/NAVAL0/ORL,GL,SD,MEM,COR,FLEA,NLON,CHS,MARI,PORH,GULP,NCRD2
      COMMON/NAVAL1/IECHO,ICASE,MM,LASTA,KKK,NST,NST1,MSTUD,NSTUD,NDNAVY
      COMMON/NAVAL2/FLER,TBR,TBS,FACTOR,STUD(4),NAVY(12000)
      COMMON/NAVAL3/BE(49),NASCHD(49,11),YD(49,5),NROUTE(49,11),IFAT
      COMMON/NAVAL4/FM(49,12),DOWF(49,5),BF(49,3),PR(49)
      COMMON/NAVAL5/IMET,WHR(98),NCLSZ(98),YYY(98),YYF(98),LIMCOU
      COMMON/NAVAL6/INTRAN,FLK(98),BACK(98),MAXSYS,MAXAUX,IFLER,IHST
      COMMON/NAVAL7/FIRSTA(98),OPD(98),ADV(98),FAIL(98),SBACK(98)
      COMMON/NAVAL8/MINS(98),MAXS(98),WTIME(98)
      COMMON/NAVAL9/PLACE(13),CNAM1(98),CNAM2(98),STYP1(49),STYP2(49)

```

CNAV10

CNAV11

CNAV12

CNAV13

C ELIMINATE FALIURES.

```

      IF (ATRIB(1).GE.0.) GO TO 5
      FLK(J) = FLK(J) + 1
      RETURN

```

C BT AND BEE GRADS:

```

      5 YYF(J) = YYF(J) + 1
      NGET = IFIX(ATRIB(1))
      NGET3 = NGET + 3
      NS = IFIX(NAVY(NGET + 2))
      IF(NS.LT.0.OR.NS.GT.MM) GO TO 14
      IF(J.GT.3) GO TO 400

```

C END OF A BASIC TRAINING CLASS

C The following statement (K = J) is necessary for determining travel in
 C the event of a branch to 11. (DO NOT REMOVE.)

```

      K = J

```

C K = CURRENT BASE.

```

      IF(BE(NS).LT.0.OR.STYP2(NS).EQ.'AFUN') GO TO 11
      KBEE = NROUTE(NS,J)

```

C KBEE = BASE FOR BEE

```

      KBEE0 = KBEE * 3
      NAVY(NGET3) = FLOAT(KBEE0)
      DELAY = TRTIME(K, KBEE)
      CALL SCHDL(8, DELAY, ATRIB)

```



```

        IF(DRAND(3).LE.FLER) CALL FLERN(8)
        RETURN
400 CONTINUE
C  JUST FINISHED A BEE COURSE
    NAVY(NGET3) = FLOAT(J)
    K = INT((J-1) / 3)
C  K = CURRENT BASE.
    NBEE = J - K * 3
C  NBEE = NUMBER OF BEE COURSE JUST TAKEN.
    IF(IABS(BE(NS))-NBEE) 13, 10, 9
    9 INTRAN = INTRAN + 1
C  Incrementing INTRAN is necessary because it will be decremented in
C  ROUTE.
    CALL ROUTE
    RETURN
10 CONTINUE
C  FINISHED WITH ALL BEE. TRAVEL TO FIRST A?
    IF(STYP2(NS).NE.'AFUN') GO TO 105
    IF(NASCHD(NS,11).GE.2) GO TO 104
    CALL STATS(NGET)
    RETURN
104 L = WHR(NASCHD(NS,2))
    DELAY = TRTIME(K, L)
C  INTRAN incremented in TRTIME.
    NAVY(NGET3) = 2.
    CALL SCHDL(9, DELAY, ATRIB)
    RETURN
C  A SCHOOL NEXT:
105 IF (BE(NS).LT.0) GO TO 12
C  A SCHOOL NEXT:
    11 L = WHR(NASCHD(NS,1))
    DELAY = TRTIME(K, L)
C  INTRAN incremented in TRTIME.
    NAVY(NGET3) = 1.
    CALL SCHDL(9, DELAY, ATRIB)
    IF(BE(NS).GE.0.AND.STYP2(NS).NE.'AFUN') RETURN
    IF(DRAND(3).LE.FLER) CALL FLERN(9)
    RETURN
C  FINISHED BEE AFTER A.
    12 CALL STATS(NGET)
    RETURN
    13 CALL GETAA(NGET,NSTUD,NAVY,MSTUD,STUD)
    WRITE(NPRNT,9902) J, K, NBEE
    CALL ERRORA(8)
9902 FORMAT(80X,'J =',I5,'    K =',I5,'    NBEE =',I5)
    RETURN
    14 CALL GETAA(NGET,NSTUD,NAVY,MSTUD,STUD)
    WRITE(NPRNT,9901)J
9901 FORMAT(100X,'J =',I3)

```

```

CALL ERRORA(9)
RETURN
END

```

C

```

FUNCTION TRTIME(I, J)

```

C RETURNS TRAVEL TIME BETWEEN PLACES I AND J.

```

COMMON/SCOM1/ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,NCLNR,
1 NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TEXT,TNOW,XX(100)
COMMON/NAVAL6/INTRAN,FLK(98),BACK(98),MAXSYS,MAXAUX,IFLER,IHST
DIMENSION IME(121)

```

```

DATA IME/0, 4, 8, 3, 2, 8, 4, 1, 9, 8, 2,
2      4, 0, 7, 2, 3, 7, 3, 3, 7, 7, 3,
3      8, 7, 0, 6, 7, 0, 10, 8, 2, 1, 6,
4      3, 2, 6, 0, 1, 6, 4, 2, 7, 6, 1,
5      2, 3, 7, 1, 0, 7, 4, 2, 9, 7, 1,
6      8, 7, 0, 6, 7, 0, 10, 8, 2, 1, 6,
7      4, 3, 10, 4, 4, 10, 0, 3, 10, 10, 5,
8      1, 3, 8, 2, 2, 8, 3, 0, 9, 8, 2,
9      9, 7, 2, 7, 9, 2, 10, 9, 0, 1, 8,
*      8, 7, 1, 6, 7, 1, 10, 8, 1, 0, 7,
1      2, 3, 6, 1, 1, 6, 5, 2, 8, 7, 0/

```

```

INTRAN = INTRAN + 1

```

```

TRTIME = 1.

```

```

IF(DRAND(4).LE.0.15) TRTIME = IME((I-1)*11 + J)

```

```

IF(I.EQ.J) TRTIME = 0.0

```

```

IF(TRTIME.EQ.0.) RETURN

```

C Weekend removal:

```

WOD = 5.0 - TNOW + 5*INT(TNOW/5.)

```

```

IF(WOD.EQ.5.0)WOD = 0.0

```

10 IF(TRTIME.LE.WOD) RETURN

```

IF(TRTIME.LE.WOD+2.) GO TO 100

```

```

TRTIME = TRTIME - 2.

```

```

WOD = WOD + 5.

```

```

GO TO 10

```

100 TRTIME = WOD

```

RETURN

```

```

END

```

C

```

SUBROUTINE FLERN(J)

```

C Generates fleet returnees = FLER % of recruits at this point.

```

REAL NAVY,MINS,MAXS

```

```

COMMON/SCOM1/ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,NCLNR,
1 NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TEXT,TNOW,XX(100)

```

```

COMMON/NAVAL1/IECHO,ICASE,MM,LASTA,KKK,NST,NST1,MSTUD,NSTUD,NDNAVY

```

```

COMMON/NAVAL2/FLER,TBR,TBS,FACTOR,STUD(4),NAVY(12000)

```

```

COMMON/NAVAL6/INTRAN,FLK(98),BACK(98),MAXSYS,MAXAUX,IFLER,IHST

```

```

CALL COPAA(IFIX(ATRIB(1)),NSTUD,NAVY,STUD)

```

```

ATRIB(1) = MSTUD

```

```

STUD(1) = TNOW

```

```

    ATRIB(2) = ATRIB(2) + 100.
    CALL PUTAA(NSTUD,NAVY,MSTUD,STUD)
    IFLER = IFLER + 1
    INTRAN = INTRAN + 1
    IF(MAXAUX.LT.MSTUD) MAXAUX = MSTUD
    IF(MAXSYS.LT.MFA) MAXSYS = MFA
C   DO NOT CALL EVENT, because FLERN was called by TRAVEL which was
C       called by EVENT. Calling a subroutine while in a nest of
C       subroutines called by that subroutine will clobber the return
C       pointer in some systems. Events may be scheduled by calling
C       SCHDL.
    IF (J.EQ.9) GO TO 9
    CALL ROUTE
    RETURN
  9  NAVY(IFIX(ATRIB(1)) + 3) = 1.
    CALL TDST
    RETURN
  END

  SUBROUTINE ROUTE
C   FILES BASIC AND BEE GRADUATES FOR NEXT BEE COURSE.
C   Auxilliary attribute 3 (NAVY(NGET+3)) = file of previous BEE.
C
    CHARACTER PLACE*4,CNAM1*4,CNAM2*4,STYP1*4,STYP2*4
    INTEGER WHR,COR,FLEA,CHS,PORH,GULP,BE,ORL,GL,SD,MEM,YYY,FLK,BACK
    INTEGER YYF
    REAL NAVY,MINS,MAXS
    COMMON/SCOM1/ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,NCLNR,
  1  NCRDR,NPRNT,NNRUN,NNSSET,NTAPE,SS(100),SSL(100),TEXT,TNOW,XX(100)

    COMMON/NAVAL0/ORL,GL,SD,MEM,COR,FLEA,NLON,CHS,MARI,PORH,GULP,NCRD2
    COMMON/NAVAL1/IECHO,ICASE,MM,LASTA,KKK,NST,NST1,MSTUD,NSTUD,NDNAVY
    COMMON/NAVAL2/FLER,TBR,TBS,FACTOR,STUD(4),NAVY(12000)
    COMMON/NAVAL3/BE(49),NASCHD(49,11),YD(49,5),NROUTE(49,11),IFAT
    COMMON/NAVAL4/FM(49,12),DOWF(49,5),BF(49,3),PR(49)
    COMMON/NAVAL5/IMET,WHR(98),NCLSIZE(98),YYY(98),YYF(98),LIMCOU
    COMMON/NAVAL6/INTRAN,FLK(98),BACK(98),MAXSYS,MAXAUX,IFLER,IHST
    COMMON/NAVAL7/FIRSTA(98),OPD(98),ADV(98),FAIL(98),SBACK(98)
    COMMON/NAVAL8/MINS(98),MAXS(98),WTIME(98)
    COMMON/NAVAL9/PLACE(13),CNAM1(98),CNAM2(98),STYP1(49),STYP2(49)
CNAV10
CNAV11
CNAV12
CNAV13
    INTRAN = INTRAN - 1
    IF(ATRIB(1).LT.0.) CALL ERRORA(10)
    NGET = IFIX(ATRIB(1))
    K = IFIX(NAVY(NGET+3))
C   K = FILE OF COURSE JUST FINISHED.

```

```

      NS = IFIX(NAVY(NGET + 2))
      NAVY(NGET + 4) = TNOW
      K = K + 1
C   NAVY(NGET+3) will be updated in TRAVEL after graduation.
      IF(K.GE.NST) GO TO 99
      CALL ENTER(K,TRIB)
      RETURN
99  CALL GETAA(NGET,NSTUD,NAVY,MSTUD,STUD)
      WRITE(NPRNT,9901) K
      CALL ERRORA(11)
9901 FORMAT(100X,'K =',I10)
      RETURN
      END

      SUBROUTINE TDST
C   Files students for first A course if I = 9.
C   Files students for second A course if I = 5.
C   ALL STUDENTS TAKE AT LEAST ONE A COURSE.
C
      CHARACTER PLACE*4,CNAM1*4,CNAM2*4,STYP1*4,STYP2*4
      INTEGER WHR,COR,FLEA,CHS,PORH,GULP,BE,ORL,GL,SD,MEM,YYY,FLK,BACK
      INTEGER YYF
      REAL NAVY,MINS,MAXS
      COMMON/SCOM1/TRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,NCLNR,
1  NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TEXT,TNOW,XX(100)

      COMMON/NAVAL0/ORL,GL,SD,MEM,COR,FLEA,NLON,CHS,MARI,PORH,GULP,NCRD2
      COMMON/NAVAL1/IECHO,ICASE,MM,LASTA,KKK,NST,NST1,MSTUD,NSTUD,NDNAVY
      COMMON/NAVAL2/FLER,TBR,TBS,FACTOR,STUD(4),NAVY(12000)
      COMMON/NAVAL3/BE(49),NASCHD(49,11),YD(49,5),NROUTE(49,11),IFAT
      COMMON/NAVAL4/FM(49,12),DOWF(49,5),BF(49,3),PR(49)
      COMMON/NAVAL5/IMET,WHR(98),NCLSIZE(98),YYY(98),YYF(98),LIMCOU
      COMMON/NAVAL6/INTRAN,FLK(98),BACK(98),MAXSYS,MAXAUX,IFLER,IHST
      COMMON/NAVAL7/FIRSTA(98),OPD(98),ADV(98),FAIL(98),SBACK(98)
      COMMON/NAVAL8/MINS(98),MAXS(98),WTIME(98)
      COMMON/NAVAL9/PLACE(13),CNAM1(98),CNAM2(98),STYP1(49),STYP2(49)

CNAV10
CNAV11
CNAV12
CNAV13
      INTRAN = INTRAN - 1
      IF(TRIB(1).LT.0.) CALL ERRORA(12)
      NGET = IFIX(TRIB(1))
      NS = IFIX(NAVY(NGET + 2))
      NA = IFIX(NAVY(NGET+3))
      K = NASCHD(NS, NA)
      IF(K.GE.NST.AND.K.LE.LASTA) GO TO 90
      CALL GETAA(NGET,NSTUD,NAVY,MSTUD,STUD)
      WRITE(NPRNT,9901) K, NA

```



```

9901 FORMAT(100X,'K =',I5,' NA =',I3)
      CALL ERRORA(13)
      RETURN
90  IF(ICASE.EQ.0.OR.BE(NS).LT.0) GO TO 95
      L = WHR(K)
      IF(L.EQ.ICASE) GO TO 95
      CALL GETAA(NGET,NSTUD,NAVY,MSTUD,STUD)
      IF(MSTUD.NE.MSTUD/4*4) CALL ERRORA(14)
      RETURN
95  CONTINUE
      NAVY(NGET + 4) = TNOW
      CALL FILEM(K,ATRI)B)
      RETURN
      END

```

SUBROUTINE CLASS(K)

```

C  SUBROUTINE CLASS(K) FIRST DETERMINES THE TIME UNTIL NEXT CONVENTING
C  FOR COURSE K.
C  THE NEXT CONVENTING IS SCHEDULED.
C  ONE CLASS IS FORMED IF THERE ARE ENOUGH STUDENTS WAITING.
C  If it is a Basic Training class, there is no limit on the number
C  OF CLASSES WHICH CAN FORM AT A GIVEN TIME.
C  THE END OF THE CLASS IS SCHEDULED FOR EACH STUDENT (CLASS TIME =
C  REGULAR TIME + TIME BETWEEN FOR SETBACKS).
C  FAILURES: AUXILIARY ATTRIBUTES WILL BE DUMPED IN FAILUR AS CLASS
C  FORMS. FAILUR ALSO SETS ATRIB(1) = -K TO SERVE AS A FAILUR FLAG
C  AND COURSE IDENTIFIER WHEN STUDENT GETS OUT OF THE COURSE IN
C  CLASSE(K) OR TRAVEL(BASIC TRAINING).
C  Setbacks are counted in FAILUR as the class forms.
C  QUEUE WAITING TIMES ARE COLLECTED.
C

```

```

      CHARACTER PLACE*4,CNAM1*4,CNAM2*4,STYP1*4,STYP2*4
      INTEGER WHR,COR,FLEA,CHS,PORH,GULP,BE,ORL,GL,SD,MEM,YYY,FLK,BACK
      INTEGER YYF
      REAL NAVY,MINS,MAXS
      COMMON/SCOM1/ATRI(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,NCLNR,
1  NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TEXT,TNOW,XX(100)
      COMMON/NAVAL0/ORL,GL,SD,MEM,COR,FLEA,NLON,CHS,MARI,PORH,GULP,NCRD2
      COMMON/NAVAL1/IECHO,ICASE,MM,LASTA,KKK,NST,NST1,MSTUD,NSTUD,NDNAVY
      COMMON/NAVAL2/FLER,TBR,TBS,FACTOR,STUD(4),NAVY(12000)
      COMMON/NAVAL3/BE(49),NASCHD(49,11),YD(49,5),NROUTE(49,11),IFAT
      COMMON/NAVAL4/FM(49,12),DOWF(49,5),BF(49,3),PR(49)
      COMMON/NAVAL5/IMET,WHR(98),NCLSZ(98),YYY(98),YYF(98),LIMCOU
      COMMON/NAVAL6/INTRAN,FLK(98),BACK(98),MAXSYS,MAXAUX,IFLER,IHST
      COMMON/NAVAL7/FIRSTA(98),OPD(98),ADV(98),FAIL(98),SBACK(98)
      COMMON/NAVAL8/MINS(98),MAXS(98),WTIME(98)
      COMMON/NAVAL9/PLACE(13),CNAM1(98),CNAM2(98),STYP1(49),STYP2(49)

```

CNAV10

CNAV11

```

CNAV12
CNAV13
      OPDK = OPD(K)
C   Scheduling of the next class convening must be done before there are
C   any changes to ATRIB.
      CALL SCHDL(6, OPDK, ATRIB)
      IF(K.LE.3) OPDK = 5.
1    X = DRAND(6)
      MINK = INT(MINS(K) + X)
      IF (NNQ(K).LT.MINS(K)) RETURN
      MAXK = INT(MAXS(K) + X)
      IF(MINK.GT.MAXK) GO TO 99
      JMAX = MINO(NNQ(K), MAXK)
      IF (JMAX.LE.0) RETURN
      DO 10 I = 1, JMAX
      CALL RMOVE(1,K,ATLIB)
      WTIME(K) = WTIME(K) + TNOW - NAVY(IFIX(ATLIB(1)) + 4)
      CALL FAILUR(K, RN)
      DELAY=ADV(K)
      IF(RN.LE.SBACK(K).AND.ATLIB(1).GE.0) DELAY = DELAY + OPDK
      CALL SCHDL(K+LIMCOU,DELAY,ATLIB)
      YYY(K) = YYY(K) + 1
      NCLSZ(K) = NCLSZ(K) + 1
10   CONTINUE
      IF(K.LT.4) GO TO 1
      RETURN
99   WRITE (NPRNT, 9901) MINK, MAXK, X
      CALL ERRORA(22)
      STOP
9901  FORMAT (' ***** ERROR: MINK =',I5,'      MAXK =',I5,'      X =',F10.6)
      END

```

```

      SUBROUTINE CLASSE(K)
C   End of A Course K (Called by EVENT, scheduled in CLASS):
C   Eliminates failures.
C   Effects elimination and data collection on those finishing pipeline
C   (subroutine STATS).
C   Files graduates with more to go for next course in sequence
C   (NAVY(NGET+3) = ordinal number of A course in sequence).
C   Routes A-before-BEE's and AFUN's appropriately.
C
      CHARACTER PLACE*4,CNAM1*4,CNAM2*4,STYP1*4,STYP2*4
      INTEGER WHR,COR,FLEA,CHS,PORH,GULP,BE,ORL,GL,SD,MEM,YYY,FLK,BACK
      INTEGER YYF
      REAL NAVY,MINS,MAXS
      COMMON/SCOM1/ATLIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,NCLNR,
1    NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TEXT,TNOW,XX(100)

      COMMON/NAVALO/ORL,GL,SD,MEM,COR,FLEA,NLON,CHS,MARI,PORH,GULP,NCRD2

```

```

COMMON/NAVAL1/IECHO, ICASE, MM, LASTA, KKK, NST, NST1, MSTUD, NSTUD, NDNNAVY
COMMON/NAVAL2/FLER, TBR, TBS, FACTOR, STUD(4), NAVY(12000)
COMMON/NAVAL3/BE(49), NASCHD(49, 11), YD(49, 5), NROUTE(49, 11), IFAT
COMMON/NAVAL4/FM(49, 12), DOWF(49, 5), BF(49, 3), PR(49)
COMMON/NAVAL5/IMET, WHR(98), NCLSIZE(98), YYY(98), YYF(98), LIMCOU
COMMON/NAVAL6/INTRAN, FLK(98), BACK(98), MAXSYS, MAXAUX, IFLER, IHST
COMMON/NAVAL7/FIRSTA(98), OPD(98), ADV(98), FAIL(98), SBACK(98)
COMMON/NAVAL8/MINS(98), MAXS(98), WTIME(98)
COMMON/NAVAL9/PLACE(13), CNAM1(98), CNAM2(98), STYP1(49), STYP2(49)
CNAV10
CNAV11
CNAV12
CNAV13
      IF (ATRIB(1).GE.0.) GO TO 2
      FLK(K) = FLK(K) + 1
      RETURN
2  NGET = IFIX(ATRIB(1))
   NGET3 = NGET + 3
   NS = IFIX(NAVY(NGET + 2))
C  NA = ORDINAL NUMBER OF A COURSE JUST TAKEN.
   NA = IFIX(NAVY(NGET3))
   NASC11 = NASCHD(NS, 11)
   IF(NA.LE.NASC11) GO TO 3
   CALL GETAA(NGET, NSTUD, NAVY, MSTUD, STUD)
   WRITE(NPRNT, 9902)K, NA
   CALL ERRORA(15)
9902 FORMAT(80X, 'K =', I5, '    NA =', I5)
      RETURN
3  NASNSK = NASCHD(NS, NA)
   IF(NASNSK.EQ.K) GO TO 35
   CALL GETAA(NGET, NSTUD, NAVY, MSTUD, STUD)
   WRITE(NPRNT, 9903)K, NASNSK
   CALL ERRORA(16)
9903 FORMAT(80X, 'K =', I5, '    NASNSK =', I5)
      RETURN
35 YYF(NASNSK) = YYF(NASNSK) + 1
   IF(NA.EQ.1.AND.STYP2(NS).EQ.'AFUN') GO TO 20
   IF(NA.EQ.NASC11) GO TO 10
   NA = NA + 1
   NAVY(NGET3) = FLOAT(NA)
C  FILE FOR NEXT A COURSE
   NAVY(NGET + 4) = TNOW
   NASNXT = NASCHD(NS, NA)
   IF(NASNXT.LE.NST.OR.NASNXT.GT.NCLNR-2) GO TO 99
   CALL FILEM(NASNXT, ATRIB)
   RETURN
10 IF (BE(NS).LT.0) GO TO 20
   CALL STATS(NGET)
   RETURN

```

```

C  AFUN, BEE AFTER A:
  20 I = WHR(NASNSK)
    KBE = NROUTE(NS,I)
    NAVY(NGET3) = FLOAT(KBE * 3)
    DELAY = TRTIME(I, KBE)
C  INTRAN incremented in TRTIME.
    CALL SCHDL (8, DELAY, ATRIB)
    RETURN
  99 CALL GETAA(NGET,NSTUD,NAVY,MSTUD,STUD)
    WRITE(NPRNT,9901) NASNXT
    CALL ERRORA(17)
  9901 FORMAT(100X,'NASNXT =',I10)
    RETURN
    END

C
  SUBROUTINE STATS(NGET)
    REAL NAVY,MINS,MAXS
    COMMON/SCOM1/ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,NCLNR,
1  NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TEXT,TNOW,XX(100)
    COMMON/NAVAL1/IECHO,ICASE,MM,LASTA,KKK,NST,NST1,MSTUD,NSTUD,NDNAVY
    COMMON/NAVAL2/FLER,TBR,TBS,FACTOR,STUD(4),NAVY(12000)

CNAV10
CNAV11
CNAV12
CNAV13
    IF(ATRIB(1).LT.0.) CALL ERRORA(18)
    IF(MSTUD.NE.MSTUD/4*4) CALL ERRORA(19)
    CALL GETAA(NGET,NSTUD,NAVY,MSTUD,STUD)
    IF(MSTUD.NE.MSTUD/4*4) CALL ERRORA(20)
C  COLLECT DATA ON THOSE FINISHING PIPELINE.
    NS = 2 * IFIX(STUD(2))
    TSYS=TNOW-STUD(1)
    IF(ATRIB(2).LT.100.) NS = NS - 1
    CALL COLCT(TSYS,NS)
    RETURN
    END

C
  SUBROUTINE FAILUR(K,RN)
C  SUBROUTINE FAILUR TAGS FAILURES (ATRIB(1) = (-K)) AND PULLS THEIR
C  AUXILIARY ATTRIBUTES.  SETBACKS ARE CONNTED.  FAILURES WILL BE
C  COUNTED AT THE END OF THE COURSE.
C
  CHARACTER PLACE*4,CNAM1*4,CNAM2*4,STYP1*4,STYP2*4
  INTEGER WHR,COR,FLEA,CHS,PORH,GULP,BE,ORL,GL,SD,MEM,YYY,FLK,BACK
  INTEGER YYF
  REAL NAVY,MINS,MAXS
  COMMON/SCOM1/ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,NCLNR,
1  NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TEXT,TNOW,XX(100)

```



```

COMMON/NAVAL0/ORL, GL, SD, MEM, COR, FLEA, NLON, CHS, MARI, PORH, GULP, NCRD2
COMMON/NAVAL1/IECHO, ICASE, MM, LASTA, KKK, NST, NST1, MSTUD, NSTUD, NDNAVY
COMMON/NAVAL2/FLER, TBR, TBS, FACTOR, STUD(4), NAVY(12000)
COMMON/NAVAL3/BE(49), NASCHD(49,11), YD(49,5), NROUTE(49,11), IFAT
COMMON/NAVAL4/FM(49,12), DOWF(49,5), BF(49,3), PR(49)
COMMON/NAVAL5/IMET, WHR(98), NCLSIZE(98), YYY(98), YYF(98), LIMCOU
COMMON/NAVAL6/INTRAN, FLK(98), BACK(98), MAXSYS, MAXAUX, IFLER, IHST
COMMON/NAVAL7/FIRSTA(98), OPD(98), ADV(98), FAIL(98), SBACK(98)
COMMON/NAVAL8/MINS(98), MAXS(98), WTIME(98)
COMMON/NAVAL9/PLACE(13), CNAM1(98), CNAM2(98), STYP1(49), STYP2(49)

CNAV10
CNAV11
CNAV12
CNAV13
      NGET = IFIX(ATTRIB(1))
      IJKL = 2
C      IF(K.LE.3) IJKL = 7
C      Use PN stream 7 for basic trainings in order to get identical results
C      in basic training for any single student type using a given set
C      of driver data.
      RN=DRAND(IJKL)
      IF(RN.GT.SBACK(K)) RETURN
      IF(RN.LE.FAIL(K)) GO TO 05
      IF(K.LT.4.OR.K.GE.NST) BACK(K) = BACK(K) + 1
      RETURN
05 CONTINUE
      CALL GETAA(NGET, NSTUD, NAVY, MSTUD, STUD)
      IF(MSTUD.NE.MSTUD/4*4) CALL ERRORA(21)
      ATTRIB(1) = FLOAT(-1*K)
      RETURN
      END

C
      SUBROUTINE REPORT
      CHARACTER PLACE*4, CNAM1*4, CNAM2*4, STYP1*4, STYP2*4
      INTEGER WHR, COR, FLEA, CHS, PORH, GULP, BE, ORL, GL, SD, MEM, YYY, FLK, BACK
      INTEGER YYF
      REAL NAVY, MINS, MAXS
      COMMON/SCOM1/ATTRIB(100), DD(100), DDL(100), DTNOW, II, MFA, MSTOP, NCLNR,
1 NCRDR, NPRNT, NNRUN, NNSET, NTAPE, SS(100), SSL(100), TEXT, TNOW, XX(100)
      COMMON/NAVAL0/ORL, GL, SD, MEM, COR, FLEA, NLON, CHS, MARI, PORH, GULP, NCRD2
      COMMON/NAVAL1/IECHO, ICASE, MM, LASTA, KKK, NST, NST1, MSTUD, NSTUD, NDNAVY
      COMMON/NAVAL2/FLER, TBR, TBS, FACTOR, STUD(4), NAVY(12000)
      COMMON/NAVAL3/BE(49), NASCHD(49,11), YD(49,5), NROUTE(49,11), IFAT
      COMMON/NAVAL4/FM(49,12), DOWF(49,5), BF(49,3), PR(49)
      COMMON/NAVAL5/IMET, WHR(98), NCLSIZE(98), YYY(98), YYF(98), LIMCOU
      COMMON/NAVAL6/INTRAN, FLK(98), BACK(98), MAXSYS, MAXAUX, IFLER, IHST
      COMMON/NAVAL7/FIRSTA(98), OPD(98), ADV(98), FAIL(98), SBACK(98)
      COMMON/NAVAL8/MINS(98), MAXS(98), WTIME(98)
      COMMON/NAVAL9/PLACE(13), CNAM1(98), CNAM2(98), STYP1(49), STYP2(49)

```

CNAV10
CNAV11
CNAV12
CNAV13

COMMON/XCOM1/MFE(100),MLE(100),NQ(100)
COMMON/GCOM6/EENQ(100),IINN(100),KKRNK(100),MMAXQ(100),QQTIM(100),
1SSOBV(98,5),SSTPV(25,6),VVNQ(100)

C

IDSCAL(I) = INT(FLOAT(I) / FACTOR + 0.5)

C

10 CALL SCHDL(1, TBR, ATRIB)

ENTRY REP2

FLE100 = FLER * 100.

WRITE(NPRNT,01) TNOW, FACTOR, FLE100

C Queue data:

WRITE (NPRNT, 9090)

I1 = 0

DO 15 K = 1, LASTA

AWT2 = 0.0

NYK = YYY(K)

IF(NYK.NE.0) AWT2 = WTIME(K) / NYK

AVE = FFAVG(K) / FACTOR

STD=FFSTD(K) / FACTOR

MAXQK = IDSCAL(MMAXQ(K))

NQK = IDSCAL(NQ(K))

I1 = I1 + NQK

15 WRITE(NPRNT,90)K,CNAM1(K),CNAM2(K),PLACE(WHR(K)),AVE,

1 STD, MAXQK, NQK,AWT2

WRITE (NPRNT, 9000) I1

C Course data:

WRITE(NPRNT,01) TNOW, FACTOR, FLE100

WRITE (NPRNT, 9050)

J3 = 0

J5 = 0

J6 = 0

DO 20 K = 1, LASTA

A1=ADV(K)

A2 = MINS(K) / FACTOR

A3 = MAXS(K) / FACTOR

F1=FAIL(K) * 100.

S1 = (SBACK(K)-FAIL(K)) * 100.

I1 = IDSCAL(YYY(K))

I2 = IDSCAL(YYF(K))

I3 = IDSCAL(FLK(K))

I4 = IDSCAL(BACK(K))

I5 = IDSCAL(NCLSZE(K))

I6 = IDSCAL(NQ(K))

J3 = J3 + I3

J5 = J5 + I5

```

      J6 = J6 + I6
20  WRITE(NPRNT,50)K,CNAM1(K),CNAM2(K),PLACE(WHR(K)),A1,OPD(K),A2,
      1      A3,F1,S1,I1,I2,I3,I4,I5,I6
      WRITE (NPRNT, 9001) J3, J5, J6
C  Student type data:
      WRITE(NPRNT,01) TNOW, FACTOR, FLE100
      WRITE (NPRNT, 9060)
      I1 = 0
      I2 = 0
      DO 30 K=1,MM
      NASK11 = NASCHD(K, 11)
      IK = 2 * K
      IK1 = IK - 1
      IYY100 = IDSCAL(IFIX(CCNUM(IK1)))
      IYY200 = IDSCAL(IFIX(CCNUM(IK)))
      I1 = I1 + IYY100
      I2 = I2 + IYY200
      WRITE(NPRNT,60)K,STYP1(K),STYP2(K),IYY100,IYY200,CCAVG(IK1),
      1      CCAVG(IK),(CNAM1(NASCHD(K,I)), CNAM2(NASCHD(K,I))),
      2      I = 1, NASK11)
30  CONTINUE
      WRITE (NPRNT, 9002) I1, I2
      IKKK = IDSCAL(KKK)
      IK = IDSCAL(IFLER)
      IT = IDSCAL(INTRAN)
      MXSYS = MAXSYS/6
      WRITE(NPRNT,100) IKKK, IK, IT, MXSYS
      IF(IHST.EQ.1) CALL PRNTH(0)
      RETURN
01  FORMAT(1H1,///,10X,'TIME =',F10.4,10X,'SCALING FACTOR =',F5.2,10X,
      1      'FLEET RETURNEES =',F6.2,'% OF RECRUIT GRADUATES',/)
50  FORMAT(' ',I4,2X,2A4,1X,A4,6F7.1,I9,I8,I7,I6,I8,I6)
60  FORMAT(5X,I2,3X,A4,' - ',A4,I11,I8,2F9.2,2X,12(1X,2A4,' '))
90  FORMAT(5X,I3,4X,2A4,3X,A4,2(2X,F10.2),2(2X,I5),F12.2,F7.2)
100 FORMAT(' ',///,10X,'TOTAL NUMBER OF RECRUIT ARRIVALS TO SYSTEM =',
      1      I8,/, 10X,'TOTAL FLEET RETURNEES =',I8,/,
      2      10X,'TOTAL IN TRANSIT =',I5,/,
      3      10X,'MAXIMUM NUMBER OF CONCURRENT ENTRIES =', I8,' (NOT DESC
      4      4ALED)',/)
9090 FORMAT(71X,'AVERAGE',/,33X,'AVERAGE',7X,'STD.',9X,'CURRENT',
      1      4X,'WAITING',/,34X,'QUEUE',8X,'DEV.',4X,'MAXQ',7X,
      2      'TIME',/)
9050 FORMAT(22X,'-----SPECIFICATIONS-----',
      1      17(' '), 'RESULTS',16(' '),/,22X,'COURSE',
      2      24X,'% ',7X,'NO. NO. NO. NO. UNDER AWAIT',/
      3      ,22X,'LENGTH BET MIN MAX FAIL SETBACK START PAS
      4      4S FAIL SETBACK INSTR INSTR',/)
9060 FORMAT(33X,'NO.',/,30X,'FINISHING',8X,'AVERAGE',/,31X,'PIPELINE',
      1      9X,'TIME',9X,'A COURSES:',/,29X,'REG FLER REG FLE

```

```

2R',/)
9000 FORMAT(60X,'-----',/,1X,'TOTALS:',I57)
9001 FORMAT(82X,'-----',8X,'-----',/,1X,
1      'TOTALS:',I78,I14,I6)
9002 FORMAT(26X,'-----',/,1X,'TOTALS:',I24,I8)
END

```

```

SUBROUTINE STATUS
CHARACTER PLACE*4,CNAM1*4,CNAM2*4,STYP1*4,STYP2*4
INTEGER WHR,COR,FLEA,CHS,PORH,GULP,BE,ORL,GL,SD,MEM,YYY,FLK,BACK
INTEGER YYF
REAL NAVY,MINS,MAXS
COMMON/SCOM1/ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,NCLNR,
1 NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TEXT,TNOW,XX(100)
DIMENSION NSET(1)
COMMON QSET(1)
EQUIVALENCE (NSET(1), QSET(1))
COMMON/NAVAL0/ORL,GL,SD,MEM,COR,FLEA,NLON,CHS,MARI,PORH,GULP,NCRD2
COMMON/NAVAL1/IECHO,ICASE,MM,LASTA,KKK,NST,NST1,MSTUD,NSTUD,NDNAVY
COMMON/NAVAL2/FLER,TBR,TBS,FACTOR,STUD(4),NAVY(12000)
COMMON/NAVAL3/BE(49),NASCHD(49,11),YD(49,5),NROUTE(49,11),IFAT
COMMON/NAVAL4/FM(49,12),DOWF(49,5),BF(49,3),PR(49)
COMMON/NAVAL5/IMET,WHR(98),NCLSZ(98),YYY(98),YYF(98),LIMCOU
COMMON/NAVAL6/INTRAN,FLK(98),BACK(98),MAXSYS,MAXAUX,IFLER,IHST
COMMON/NAVAL7/FIRSTA(98),OPD(98),ADV(98),FAIL(98),SBACK(98)
COMMON/NAVAL8/MINS(98),MAXS(98),WTIME(98)
COMMON/NAVAL9/PLACE(13),CNAM1(98),CNAM2(98),STYP1(49),STYP2(49)

CNAV10
CNAV11
CNAV12
CNAV13
      CALL SCHDL(2,TBS,ATRIB)
      WRITE (NPRNT, 9000) TNOW, FACTOR
C  UNDER INSTRUCTION (Look for scheduled course-completion events):
C  The calendar is searched from start to end for each course.
      WRITE (NPRNT, 9010) TNOW
      DO 1000 K = 1, LASTA
      WRITE(NPRNT,9009) K, CNAM1(K), CNAM2(K), TNOW
      NC = 0
      CLSEND = FLOAT(K+LIMCOU)
      L = MMFE(NCLNR)
      GO TO 11
10  L = NSUCR(L)
11  IF(L.EQ.0) GO TO 1000
      IF(QSET(L+3).NE.CLSEND) GO TO 10
      NC = NC + 1
      TIME1 = QSET(L+4)
      NGET = IFIX(QSET(L+1))
      IF (NGET.LT.0) GO TO 40

```

```

      NS = IFIX(NAVY(NGET+2))
      TIME2 = NAVY(NGET+1)
      WRITE (NPRNT, 9011) NC, STYP1(NS), STYP2(NS), TIME1, TIME2
      GO TO 10
40  WRITE (NPRNT, 9012) NC, TIME1
      GO TO 10
1000 CONTINUE
C  QUEUE COMPOSITIONS:
      WRITE(NPRNT,9008)
      DO 2000 K = 1, LASTA
        WRITE(NPRNT,9007) K, CNAM1(K), CNAM2(K), TNOW
        NC = 1
        L = MMFE(K)
2050 IF(L.EQ.0) GO TO 2000
        NGET = IFIX(QSET(L+1))
        NS = IFIX(NAVY(NGET+2))
        TIME1 = NAVY(NGET+4)
        TIME2 = NAVY(NGET+1)
        WRITE(NPRNT,9006)NC,STYP1(NS),STYP2(NS),QSET(L+2),TIME1,TIME2
        NC = NC + 1
        L = NSUCR(L)
        GO TO 2050
2000 CONTINUE
C  IN TRANSIT:
      WRITE (NPRNT,9013) TNOW
      NC = 0
      L = MMFE(NCLNR)
      GO TO 51
50  L = NSUCR(L)
51  IF(L.EQ.0) RETURN
      QSETL3 = QSET(L+3)
      IF(QSETL3.NE.8.AND.QSETL3.NE.9) GO TO 50
      NC = NC + 1
      NS = IFIX(NAVY(IFIX(QSET(L+1))+2))
      TIME = QSET(L+4)
      IF(QSETL3.NE.8) GO TO 59
      NO = 12
      NP = IFIX(QSET(L+2))
      GO TO 60
59  NO = 13
      NP = WHR(NASCHD(NS,1))
60  WRITE(NPRNT,9014)NC,STYP1(NS),STYP2(NS),PLACE(NP),PLACE(NO),
1    TIME
      GO TO 50
9000 FORMAT(1H1,////,10X,'SYSTEM STATUS REPORT',10X,'TIME =',F10.4,10X,
1    'SCALING FACTOR =',F10.4,' (DATA IS NOT DESCALED)')
9004 FORMAT (1X, I4,5X, 2A4, F20.4)
9005 FORMAT(1X,I4, 5X, 'FAILURE ', F20.4)
9006 FORMAT(1X,I5, 4X, 2A4, F10.2, F40.2, F20.4)

```



```

9007 FORMAT(10X,I10,'. ',2A4,'    QUEUE COMPOSITION.    TIME =',F8.2,/,
1      5X,'RATE,    PRIORITY/ORDINAL NUMBER OF A COURSE    QUEUE ARR
2IVAL TIME    RTC ARRIVAL TIME')
9008 FORMAT(/,20X,'QUEUE COMPOSITIONS (AWAITING INSTRUCTION):')
9009 FORMAT(10X,I10,'. ',2A4,'    CLASS COMPOSITION.    TIME =',F8.2,/,
1      10X,'TYPE EXPECTED COMPLETION TIME    RTC ARRIVAL TIME')
9010 FORMAT(/,20X,'UNDER INSTRUCTION:',50X,'TIME =',F9.3)
9011 FORMAT(1X,I5,4X,2A4,F15.3,F21.3)
9012 FORMAT(1X,I5,4X,'FAILURE ',F15.3)
9013 FORMAT(/,20X,'TRANSIENTS:',30X,'TIME =',F10.2,/,10X,
1      'TYPE    DESTINATION FOR EXPECTED ARRIVAL TIME')
9014 FORMAT(1X,I5,2X,2A4,6X,A4,5X,A4,F20.4)
END

```

```

SUBROUTINE ERRORA(IEERROR)
CHARACTER PLACE*4,CNAM1*4,CNAM2*4,STYP1*4,STYP2*4,LABEL(30)*4
INTEGER WHR,COR,FLEA,CHS,PORH,GULP,BE,ORL,GL,SD,MEM,YYY,FLK,BACK
INTEGER YYF
REAL NAVY,MINS,MAXS
COMMON QSET(1)
DIMENSION NSET(1)
EQUIVALENCE (NSET(1),QSET(1))
COMMON/SCOM1/ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,NCLNR,
1 NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TEXT,TNOW,XX(100)
COMMON/NAVAL0/ORL,GL,SD,MEM,COR,FLEA,NLON,CHS,MARI,PORH,GULP,NCRD2
COMMON/NAVAL1/IECHO,ICASE,MM,LASTA,KKK,NST,NST1,MSTUD,NSTUD,NDNAVY
COMMON/NAVAL2/FLER,TBR,TBS,FACTOR,STUD(4),NAVY(12000)
COMMON/NAVAL3/BE(49),NASCHD(49,11),YD(49,5),NROUTE(49,11),IFAT
COMMON/NAVAL4/FM(49,12),DOWF(49,5),BF(49,3),PR(49)
COMMON/NAVAL5/IMET,WHR(98),NCLSIZE(98),YYY(98),YYF(98),LIMCOU
COMMON/NAVAL6/INTRAN,FLK(98),BACK(98),MAXSYS,MAXAUX,IFLER,IHST
COMMON/NAVAL7/FIRSTA(98),OPD(98),ADV(98),FAIL(98),SBACK(98)
COMMON/NAVAL8/MINS(98),MAXS(98),WTIME(98)
COMMON/NAVAL9/PLACE(13),CNAM1(98),CNAM2(98),STYP1(49),STYP2(49)
CNAV10
CNAV11
CNAV12
CNAV13
DATA LABEL/'EVN1','EVN2','EVN3','EVN7','RCRT','RCT2','RCT3','TRVL'
1  'TRV2','ROUT','ROU2','TDST','TDS2','TDS3','CLSE','CSE2','CSE3',
2  'STAT','STA2','STA3','FAIL','CLAS','EVN5',7*'----'/
WRITE (NPRNT, 9000) LABEL(IEERROR), TNOW, ATRIB(1), ATRIB(2), STUD
WRITE(NPRNT,9003)NST,ORL,GL,SD,MEM,COR,FLEA,NLON,CHS,MARI,PORH,
1  GULP,ICASE,FACTOR,TBR,TBS
WRITE(NPRNT,9002) MM, LASTA, LIMCOU
WRITE(NPRNT,9020) MSTUD, NSTUD, NDNAVY
MXAUX = MAXAUX / NSTUD
MXSYS = MAXSYS / 6
WRITE(NPRNT,9021) KKK, MXAUX, MXSYS

```

```

WRITE(NPRNT,9016) (I,(NAVY(I + J),J = 1, 20), I = 0, MAXAUX, 20)
IF(IECHO.NE.O)WRITE(NPRNT,9008)
DO 5 I = 1, MM
IF(IECHO.NE.O)WRITE(NPRNT,9005)I,STYP1(I),STYP2(I),BE(I),
1 (NROUTE(I,J),J=1,11),(NASCHD(I,J),J=1,11),PR(I)
5 CONTINUE
IF(IECHO.NE.O)WRITE (NPRNT,9014)
DO 9 I = 1, MM
IF(IECHO.NE.O)WRITE(NPRNT,9011)I,(YD(I,J),J=1,5),(BF(I,J),J=1, 3),
1 (FM(I, J), J = 1, 12), (DOWF(I, J), J = 1, 5)
9 CONTINUE
IF(IECHO.NE.O)WRITE(NPRNT,9001)(I,(NASCHD(I,J),J=1,11), I = 1, MM)
IF(IECHO.NE.O)WRITE (NPRNT, 9009)
DO 30 I = 1, LASTA
IF(IECHO.NE.O)WRITE(NPRNT,9007)I,CNAM1(I),CNAM2(I),FAIL(I),
1 SBACK(I), MINS(I), MAXS(I), ADV(I), FIRSTA(I), OPD(I)
30 CONTINUE
IF(IECHO.NE.O)WRITE (NPRNT,9015) WHR
CALL REP2
CALL SUMRY
C L = MMFE(NCLNR)
C J = 1
C WRITE (NPRNT,9017)
C 40 WRITE(NPRNT,9018)J,NSET(L),(QSET(L+I),I=1,4),NSET(L+5)
C J = J + 1
C L = NSUCR(L)
C IF(L.NE.O) GO TO 40
C WRITE(NPRNT,9019)((I,NSET(I+1),(QSET(I+J),J=2,5),NSET(I+6),
C 1 NSET(I+7),(QSET(I+J),J=8,11),NSET(I+12)),I=0,MAXSYS,12)
C WRITE(NPRNT,9023)((I,(NSET(I*10+J),J=1,10)),I=6000,6080)
C WRITE(NPRNT,9024)((I,(QSET(I*10+J),J=1,10)),I=6000,6080)
X = SQRT(-1.)
STOP
9000 FORMAT(///,20X,'***** A CALL WAS MADE TO ERRORA FROM ',A4,' AT TIM
1E =',F10.4,/,10X,'ATTRIBS = ',2F10.4,10X,'AUX ATRIBS = ',4F10.4)
9001 FORMAT (//,1X'DATA:',8X,'A COURSE SEQUENCES BY STUDENT TYPE:',/,
1 (1X, I5, 5X, 10I5, I8))
9002 FORMAT(///,10X, 'MM =', I3, ' LASTA =', I3, ' LIMCOU =', I5)
9003 FORMAT(///,1X,'NST =', I3,/, ' ORLANDO =', I3,/, ' GREAT LAKES =',
1 I3,/, ' SAN DIEGO =', I3,/,
2 ' MEMPHIS =', I3,/, ' CORRY STATION =', I3,/, ' FLEASWTRAC =', I3,
3/, ' NEW LONDON =', I3,/, ' CHARLESTON =', I3,/, ' MARE ISLAND =', I3,/,
4 ' PORT HUENEME =', I3,/, ' GULF PORT =', I3,/, ' ICASE =', I2,/,
5 ' SCALING FACTOR =', F5.2,/, ' TIME BETWEEN REPORTS =', F8.1,/,
6 ' TIME BETWEEN SNAPSHOTS =', F8.1)
9005 FORMAT (1X,I3,2X,2A4,I5,2X,11I3,2X,11I3,F8.1)
9007 FORMAT (1X, I3, 2X, 2A4, 2F6.2, 2F7.2, 3F8.1)
9008 FORMAT(//,20X,'STUDENT DATA:',//,9X,'TYPE BE NROUTE',
1 11X, 7X,10X,'NASCHD:',25X,'BEE PRI')

```

```

9009 FORMAT(///,20X,'COURSE DATA:',/,10X,'CNAME    FAIL SBACK',4X,
1      'MIN    MAX    ADV    FRST    BET')
9011 FORMAT (1X, I2, 1X, 5F5.0, 1X, 3F5.2, 1X, 12F5.2, 1X, 5F5.2)
9014 FORMAT (/,5X,'Negative "BE" indicates that A school comes before B
1EE.',/,5X,'TOTAL YEARLY DEMAND',7X,
2      'BASE FACTORS    MONTH FACTORS',48X,'DAY-OF-WEEK FACTORS')
9015 FORMAT (1X,'WHR:',/, (1X,10I3))
9016 FORMAT(///,1X,'NAVY:',/, (1X,I5,4X,20F6.0))
9017 FORMAT(//,1X,'EVENT CALENDAR:')
9018 FORMAT(10X,I5,I11,4F10.2,I6)
9019 FORMAT(/,1X,'QSET:',/, (1X,I5,4X,I6,3X,4F10.3,I6,I14,4F10.3,I6))
9020 FORMAT(10X,'MSTUD =',I5,'    NSTUD =',I2,'    NDNAVY =',I6)
9021 FORMAT(10X,'KKK =',I5,'    MXAUX =',I5,'    MXSYS =',I5)
9022 FORMAT(///,1X,'NSET:',/, (1X,I5,12I10))
9023 FORMAT(//,1X,'NSET (NETWORK):',/, (1X,I5,10I12))
9024 FORMAT(//,1X,'QSET (NETWORK):',/, (1X,I5,10F12.3))
      END

```


3	634	634	634	634	634	0.34	0.25	0.41	0.09	0.07	0.07	0.06	0.07	0.11	0.10	0.11	0.10	0.09	0.07	0.06	0.18	0.20	0.24	0.20	0.18
4	2608	2608	2608	2608	2608	0.22	0.48	0.30	0.09	0.07	0.07	0.06	0.07	0.11	0.10	0.11	0.10	0.09	0.07	0.06	0.18	0.20	0.24	0.20	0.18
5	640	640	640	640	640	0.31	0.52	0.17	0.09	0.07	0.07	0.06	0.07	0.11	0.10	0.11	0.10	0.09	0.07	0.06	0.18	0.20	0.24	0.20	0.18
6	601	601	601	601	601	0.21	0.46	0.14	0.09	0.07	0.07	0.06	0.07	0.11	0.10	0.11	0.10	0.09	0.07	0.06	0.18	0.20	0.24	0.20	0.18
7	122	122	122	122	122	0.24	0.66	0.10	0.09	0.07	0.07	0.06	0.07	0.11	0.10	0.11	0.10	0.09	0.07	0.06	0.18	0.20	0.24	0.20	0.18
8	549	549	549	549	549	0.81	0.10	0.09	0.09	0.07	0.07	0.06	0.07	0.11	0.10	0.11	0.10	0.09	0.07	0.06	0.18	0.20	0.24	0.20	0.18
9	731	731	731	731	731	0.19	0.56	0.26	0.09	0.07	0.07	0.06	0.07	0.11	0.10	0.11	0.10	0.09	0.07	0.06	0.18	0.20	0.24	0.20	0.18
10	255	255	255	255	255	0.22	0.31	0.47	0.09	0.07	0.07	0.06	0.07	0.11	0.10	0.11	0.10	0.09	0.07	0.06	0.18	0.20	0.24	0.20	0.18
11	664	664	664	664	664	0.21	0.49	0.30	0.09	0.07	0.07	0.06	0.07	0.11	0.10	0.11	0.10	0.09	0.07	0.06	0.18	0.20	0.24	0.20	0.18
12	499	499	499	499	499	0.17	0.55	0.28	0.09	0.07	0.07	0.06	0.07	0.11	0.10	0.11	0.10	0.09	0.07	0.06	0.18	0.20	0.24	0.20	0.18
13	395	395	395	395	395	0.20	0.55	0.24	0.09	0.07	0.07	0.06	0.07	0.11	0.10	0.11	0.10	0.09	0.07	0.06	0.18	0.20	0.24	0.20	0.18
14	226	226	226	226	226	0.09	0.60	0.31	0.09	0.07	0.07	0.06	0.07	0.11	0.10	0.11	0.10	0.09	0.07	0.06	0.18	0.20	0.24	0.20	0.18
15	118	118	118	118	118	0.24	0.43	0.33	0.09	0.07	0.07	0.06	0.07	0.11	0.10	0.11	0.10	0.09	0.07	0.06	0.18	0.20	0.24	0.20	0.18
16	436	436	436	436	436	0.13	0.73	0.14	0.09	0.07	0.07	0.06	0.07	0.11	0.10	0.11	0.10	0.09	0.07	0.06	0.18	0.20	0.24	0.20	0.18
17	1484	1484	1484	1484	1484	0.22	0.20	0.58	0.09	0.07	0.07	0.06	0.07	0.11	0.10	0.11	0.10	0.09	0.07	0.06	0.18	0.20	0.24	0.20	0.18
18	236	236	236	236	236	0.00	0.00	1.00	0.09	0.07	0.07	0.06	0.07	0.11	0.10	0.11	0.10	0.09	0.07	0.06	0.18	0.20	0.24	0.20	0.18
19	576	576	576	576	576	0.00	0.00	1.00	0.09	0.07	0.07	0.06	0.07	0.11	0.10	0.11	0.10	0.09	0.07	0.06	0.18	0.20	0.24	0.20	0.18
20	501	501	501	501	501	0.00	0.00	1.00	0.09	0.07	0.07	0.06	0.07	0.11	0.10	0.11	0.10	0.09	0.07	0.06	0.18	0.20	0.24	0.20	0.18
21	468	468	468	468	468	0.79	0.13	0.08	0.09	0.07	0.07	0.06	0.07	0.11	0.10	0.11	0.10	0.09	0.07	0.06	0.18	0.20	0.24	0.20	0.18
22	3292	3292	3292	3292	3292	0.33	0.33	0.33	0.09	0.07	0.07	0.06	0.07	0.11	0.10	0.11	0.10	0.09	0.07	0.06	0.18	0.20	0.24	0.20	0.18
23	714	714	714	714	714	0.33	0.33	0.33	0.09	0.07	0.07	0.06	0.07	0.11	0.10	0.11	0.10	0.09	0.07	0.06	0.18	0.20	0.24	0.20	0.18
24	316	316	316	316	316	0.33	0.33	0.33	0.09	0.07	0.07	0.06	0.07	0.11	0.10	0.11	0.10	0.09	0.07	0.06	0.18	0.20	0.24	0.20	0.18
25	1981	1981	1981	1981	1981	0.33	0.33	0.33	0.09	0.07	0.07	0.06	0.07	0.11	0.10	0.11	0.10	0.09	0.07	0.06	0.18	0.20	0.24	0.20	0.18
26	430	430	430	430	430	0.33	0.33	0.33	0.09	0.07	0.07	0.06	0.07	0.11	0.10	0.11	0.10	0.09	0.07	0.06	0.18	0.20	0.24	0.20	0.18
27	479	479	479	479	479	0.33	0.33	0.33	0.09	0.07	0.07	0.06	0.07	0.11	0.10	0.11	0.10	0.09	0.07	0.06	0.18	0.20	0.24	0.20	0.18

COURSE DATA

CHANE	FAIL BACK	MIN	MAX	LENGTH	FRST	BET
1	1	1	1	1	1	1
2	2	2	2	2	2	2
3	3	3	3	3	3	3
4	4	4	4	4	4	4
5	5	5	5	5	5	5
6	6	6	6	6	6	6
7	7	7	7	7	7	7
8	8	8	8	8	8	8
9	9	9	9	9	9	9
10	10	10	10	10	10	10
11	11	11	11	11	11	11
12	12	12	12	12	12	12
13	13	13	13	13	13	13
14	14	14	14	14	14	14
15	15	15	15	15	15	15
16	16	16	16	16	16	16
17	17	17	17	17	17	17
18	18	18	18	18	18	18
19	19	19	19	19	19	19
20	20	20	20	20	20	20
21	21	21	21	21	21	21
22	22	22	22	22	22	22
23	23	23	23	23	23	23
24	24	24	24	24	24	24
25	25	25	25	25	25	25
26	26	26	26	26	26	26
27	27	27	27	27	27	27
28	28	28	28	28	28	28
29	29	29	29	29	29	29
30	30	30	30	30	30	30

31	31	CDP 6115	10.00	10.00	4.00	8.00	60.0	0.0	5.0
32	32	CDP 6178	10.00	10.00	5.00	6.00	60.0	0.0	5.0
33	33	CDP 606C	1.00	1.00	5.00	10.00	16.0	0.0	5.0
34	34	CDP 8563	1.00	1.00	27.00	28.00	20.0	0.0	20.0
35	35	CDP 8565	1.00	1.00	5.00	10.00	30.0	0.0	10.0
36	36	CDP 606B	10.00	0.00	9.00	10.00	20.0	0.0	5.0
37	37	CDP 8564	1.00	1.00	17.00	20.00	27.0	0.0	20.0
38	38	(UNUSED)	100.00	0.00	0.00	0.00	99999.0	99999.0	99999.0
39	39	CDP 6073	5.00	21.00	5.00	10.00	40.0	0.0	5.0
40	40	CDP 605X	5.00	21.00	5.00	10.00	40.0	0.0	5.0
41	41	CDP 606D	15.00	6.00	6.00	7.00	35.0	0.0	5.0
42	42	(UNUSED)	100.00	0.00	0.00	0.00	99999.0	99999.0	99999.0
43	43	CDP 6218	5.00	7.00	31.00	32.00	10.0	0.0	5.0
44	44	CDP 6515	7.00	25.00	8.00	16.00	55.0	0.0	3.0
45	45	CDP 6220	5.00	7.00	14.00	18.00	10.0	0.0	5.0
46	46	CDP 6240	10.00	0.00	11.00	13.00	91.0	0.0	5.0
47	47	CDP 6245	10.00	25.00	4.00	5.00	100.0	0.0	5.0
48	48	CDP 6215	5.00	7.00	5.00	8.00	10.0	0.0	5.0
49	49	CDP 6530	5.00	20.00	11.00	12.00	46.0	1.0	10.0
50	50	CDP 6219	5.00	7.00	41.00	43.00	10.0	0.0	5.0
51	51	CDP 6239	10.00	0.00	33.00	36.00	91.0	0.0	5.0
52	52	CDP 6246	10.00	25.00	10.00	11.00	100.0	0.0	5.0
53	53	CDP 6221	5.00	7.00	8.00	9.00	10.0	0.0	5.0
54	54	CDP 6241	10.00	0.00	7.00	8.00	91.0	0.0	5.0
55	55	CDP 6246	10.00	25.00	1.00	2.00	100.0	0.0	5.0
56	56	CDP 6222	5.00	7.00	8.00	11.00	10.0	0.0	5.0
57	57	CDP 6242	10.00	0.00	5.00	9.00	69.0	0.0	5.0
58	58	CDP 6521	5.00	12.00	6.00	7.00	29.0	0.0	5.0
59	59	(UNUSED)	100.00	0.00	0.00	0.00	99999.0	99999.0	99999.0
60	60	CDP 605A	8.00	28.00	7.00	9.00	50.0	0.0	5.0
61	61	CDP 605B	3.00	9.00	6.00	8.00	15.0	0.0	5.0
62	62	CDP 605C	2.00	15.00	6.00	8.00	20.0	0.0	5.0
63	63	CDP 6163	15.00	25.00	5.00	7.00	75.0	0.0	5.0
64	64	CDP 604A	5.00	20.00	1.00	2.00	50.0	0.0	5.0
65	65	CDP 604B	2.00	15.00	1.00	2.00	20.0	0.0	5.0
66	66	CDP 604C	1.00	5.00	7.00	8.00	15.0	0.0	20.0
67	67	CDP 604D	1.00	6.00	1.00	2.00	20.0	0.0	5.0
68	68	CDP 802A	5.00	12.00	10.00	13.00	15.0	0.0	5.0
69	69	CDP 602B	10.00	28.00	9.00	13.00	50.0	0.0	5.0
70	70	CDP 602C	1.00	10.00	8.00	11.00	10.0	0.0	5.0
71	71	CDP 602D	2.00	10.00	8.00	11.00	20.0	0.0	5.0
72	72	CDP 603A	3.00	15.00	4.00	6.00	45.0	0.0	5.0
73	73	CDP 603B	1.00	15.00	4.00	5.00	20.0	0.0	5.0
74	74	(UNUSED)	100.00	0.00	0.00	0.00	99999.0	99999.0	99999.0
75	75	CDP 6015	10.00	4.00	5.00	10.00	30.0	0.0	5.0
76	76	CDP 6195	1.00	0.00	5.00	10.00	15.0	0.0	5.0
77	77	CDP 6196	1.00	0.00	5.00	10.00	10.0	0.0	5.0
78	78	CDP 6198	1.00	0.00	5.00	10.00	15.0	0.0	5.0
79	79	CDP 6118	2.00	2.00	5.00	10.00	35.0	0.0	5.0
80	80	CDP 641U	0.00	0.00	5.00	10.00	15.0	0.0	5.0
81	81	CDP 4571	1.00	0.00	5.00	10.00	20.0	0.0	5.0
82	82	CDP 6172	20.00	15.00	5.00	10.00	40.0	0.0	5.0
83	83	CDP 6402	10.00	10.00	5.00	10.00	20.0	0.0	5.0
84	84	CDP 6401	0.00	0.00	5.00	10.00	20.0	0.0	5.0
85	85	CDP 6460	10.00	10.00	5.00	10.00	20.0	0.0	5.0
86	86	CDP 600A	10.00	10.00	5.00	10.00	30.0	0.0	5.0
87	87	(UNUSED)	100.00	0.00	0.00	0.00	99999.0	99999.0	99999.0
88	88	CDP 5200	25.00	15.00	86.00	89.00	49.0	0.0	5.0
89	89	CDP 6137	10.00	0.00	11.00	12.00	60.0	0.0	15.0
90	90	CDP 6342	6.00	10.00	5.00	10.00	25.0	0.0	5.0
91	91	(UNUSED)	100.00	0.00	0.00	0.00	99999.0	99999.0	99999.0

92 92	CDP 6041	12.00	15.00	12.00	13.00	80.00	0.00	20.00
91 93	(UNUSED)	100.00	0.00	0.00	0.00	99999.00	99999.00	99999.00
94 94	CDP 6131	13.00	25.00	14.00	19.00	110.00	0.00	5.00
95 95	(UNUSED)	100.00	0.00	0.00	0.00	99999.00	99999.00	99999.00
96 96	CDP 6079	4.00	3.00	18.00	31.00	40.00	0.00	20.00
97 97	(UNUSED)	100.00	0.00	0.00	0.00	99999.00	99999.00	99999.00
98 98	CDP 6289	4.00	3.00	20.00	22.00	40.00	0.00	20.00

MHR:

1	2	3	1	1	2	2	2	3
3	3	4	4	1	1	2	2	2
2	2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	3	3
3	3	4	4	4	4	4	4	4
4	4	4	4	4	4	4	4	5
5	5	5	5	5	5	5	5	5
5	5	5	5	5	5	5	5	5
6	6	6	6	6	6	6	6	6
6	6	6	6	6	6	6	6	6
7	8	8	9	9	10	10	11	

PRE-CARULLIN DATA ANALYSIS: Fleet returnees = 10.00% of basic training graduates.

STEADY STATE STUDENT I/U RATES (if no bottlenecks)
STUDENT TYPE I/U UNIT
(students per year)

1	CE	144.0	244.2
2	CF4	424.0	252.7
3	DS	634.0	442.3
4	EW	2608.0	1946.6
5	ET AEF	680.0	332.6
6	ET NF	601.0	366.7
7	ET	122.0	74.6
8	EA	540.0	350.7
9	FIG	731.0	512.1
10	FIG SS	255.0	138.0
11	FM	604.0	439.6
12	GAG	492.0	400.1
13	G4M	395.0	316.8
14	GAT ASRU	250.0	200.5
15	GSE	118.0	91.8
16	GSM	436.0	308.4
17	IC	1484.0	1130.5
18	R4 SS	236.0	178.7
19	STG	576.0	411.5
20	ST8	501.0	271.9
21	T4	468.0	404.5
22	AE AFUN	3282.0	2591.2
23	AJ AFUN	714.0	489.5
24	ASE AFUN	316.0	254.1
25	AT AFUN	1981.0	1358.1
26	AL AFUN	450.0	294.8
27	TD AFUN	479.0	340.6

COURSE MAXIMUM CAPACITIES AND FEED RATES (assuming no previous bottlenecks reflecting feed):
COURSE CAPACITY FEED RATE %FAIL GRAD RATE (students per year)

1	PTC ORL UNLIMITED	5838.4	10.0	5254.5
2	MTC GL UNLIMITED	6865.3	10.0	6178.7
3	MTC SD UNLIMITED	7042.6	10.0	6338.4
4	BE1 ORL	7280.0	10.0	3035.3
5	BE2 ORL	7280.0	10.0	2122.5
6	BE3 ORL	5546.7	10.0	251.9
7	BE1 GL	7280.0	10.0	4031.8
8	BE2 GL	7280.0	10.0	2986.9
9	BE3 GL	5546.7	10.0	496.0
10	BE1 SD	7280.0	10.0	4042.5
11	BE2 SD	7280.0	10.0	3131.6
12	BE3 SD	5546.7	10.0	233.8
13	BE1 AFM	7280.0	10.0	6104.0
14	BE2 AFM	7280.0	10.0	0.0
15	BE3 AFM	5546.7	10.0	0.0
16	CUP 609J	572.0	10.0	0.0
17	(UNUSED)	0.0	100.0	0.0
18	CUP 6070	2444.0	5.0	1986.8
19	CUP 605Y	520.0	0.0	0.0
20	CUP 603V	520.0	15.0	392.4
21	CUP 604E	1404.0	15.0	368.7
22	CUP 603X	208.0	15.0	74.8
23	CUP 603M	104.0	15.0	0.0
24	CUP 603Z	104.0	15.0	0.0
25	CUP 604F	204.0	15.0	0.0
26	CUP 6377	936.0	9.0	533.4
27	CUP 6376	596.0	4.0	512.1

28	CUP 6021	756.6	512.5	14.0	457.9
29	CUP 6104	541.0	477.9	4.0	439.0
30	CUP 6400	676.0	448.0	10.0	400.1
31	CUP 6115	416.0	351.9	10.0	316.8
32	CUP 6178	312.0	226.8	10.0	200.5
33	CUP 606C	520.0	118.8	1.0	115.7
34	CUP 6563	464.0	430.3	1.0	360.4
35	CUP 6565	280.0	114.5	1.0	113.4
36	CUP 6068	520.0	349.6	10.0	314.7
37	CUP 8564	280.0	311.5	1.0	257.4
38	(UNUSED)	0.0	0.0	100.0	0.0
39	CUP 6973	520.0	1190.0	5.0	494.0
40	CUP 605X	520.0	0.0	5.0	0.0
41	CUP 6060	364.0	210.3	15.0	178.7
42	(UNUSED)	0.0	0.0	100.0	0.0
43	CUP 6218	1064.0	3258.8	5.0	1580.8
44	CUP 6515	1386.7	2780.2	7.0	1289.6
45	CUP 6220	936.0	706.8	5.0	671.4
46	CUP 6240	876.0	608.3	10.0	543.8
47	CUP 6245	280.0	541.9	10.0	234.0
48	CUP 6215	416.0	312.8	5.0	297.2
49	CUP 6530	312.0	267.5	5.0	254.1
50	CUP 6219	2236.0	1961.0	5.0	1862.9
51	CUP 6239	1672.0	1676.6	10.0	1509.0
52	CUP 6244	572.0	1509.0	10.0	514.8
53	CUP 6221	464.0	425.7	5.0	404.4
54	CUP 6241	416.0	362.9	10.0	327.5
55	CUP 6246	104.0	327.5	10.0	93.6
56	CUP 6222	572.0	474.2	5.0	450.5
57	CUP 6242	488.0	405.4	10.0	364.9
58	CUP 6521	364.0	364.9	5.0	345.8
59	(UNUSED)	0.0	0.0	100.0	0.0
60	CUP 605A	488.0	340.0	8.0	312.8
61	CUP 6058	416.0	312.4	3.0	303.4
62	CUP 605C	416.0	303.4	2.0	297.4
63	CUP 6101	364.0	297.4	15.0	252.7
64	CUP 604A	104.0	0.0	5.0	0.0
65	CUP 6048	104.0	0.0	2.0	0.0
66	CUP 604C	104.0	0.0	1.0	0.0
67	CUP 6040	104.0	0.0	1.0	0.0
68	CUP 602A	676.0	440.2	5.0	418.2
69	CUP 6028	676.0	418.4	10.0	376.4
70	CUP 602C	572.0	376.4	1.0	372.6
71	CUP 602D	572.0	372.6	2.0	365.2
72	CUP 603A	312.0	365.2	3.0	302.6
73	CUP 603B	280.0	354.2	1.0	257.4
74	(UNUSED)	0.0	0.0	100.0	0.0
75	CUP 6015	520.0	570.2	10.0	468.0
76	CUP 6195	520.0	513.2	1.0	508.1
77	CUP 6196	520.0	0.0	1.0	0.0
78	CUP 6198	520.0	0.0	1.0	0.0
79	CUP 6118	520.0	0.0	2.0	0.0
80	CUP 5410	520.0	0.0	0.0	0.0
81	CUP 4573	520.0	0.0	1.0	0.0
82	CUP 6172	520.0	377.6	20.0	302.1
83	CUP 6402	520.0	304.1	10.0	271.9
84	CUP 6401	520.0	0.0	0.0	0.0
85	CUP 6460	520.0	0.0	10.0	0.0
86	CUP 630A	520.0	0.0	10.0	0.0
87	(UNUSED)	0.0	0.0	100.0	0.0
88	CUP 5200	4628.0	204.5	25.0	153.4

89	CUP 6337	206.0	151.1	10.0	138.0
90	CUP 6342	526.0	401.8	6.0	377.6
91	(UNUSED)	0.0	0.0	100.0	0.0
92	CUP 6041	169.0	0.0	12.0	0.0
93	(UNUSED)	0.0	0.0	100.0	0.0
94	CUP 6131	986.0	508.4	13.0	442.3
95	(UNUSED)	0.0	0.0	100.0	0.0
96	CUP 6079	403.0	308.5	4.0	294.2
97	(UNUSED)	0.0	0.0	100.0	0.0
98	CUP 6289	286.0	0.0	4.0	0.0

**** feed rate exceeds capacity.

53	CDP 6221	MEM	10.0	5.0	8.0	9.0	5.0	7.0	720	660	20	80	40	80
54	CDP 6241	MEM	91.0	5.0	7.0	8.0	10.0	0.0	520	320	40	0	160	20
55	CDP 6246	MEM	100.0	5.0	1.0	2.0	10.0	25.0	100	80	0	40	20	220
56	CDP 6222	MEM	10.0	5.0	8.0	11.0	5.0	7.0	1020	960	20	100	40	120
57	CDP 6242	MEM	69.0	5.0	5.0	9.0	10.0	0.0	720	580	60	0	80	160
58	CDP 6521	MEM	29.0	5.0	6.0	7.0	5.0	12.0	400	400	0	80	0	180
59	(UNUSED)	MEM	9999.099999	0.0	0.0	0.0	100.0	0.0	0	0	0	0	0	0
60	CDP 605A	COR	50.0	5.0	7.0	9.0	8.0	28.0	580	440	40	220	100	0
61	CDP 605B	COR	15.0	5.0	6.0	8.0	3.0	9.0	400	360	40	0	0	40
62	CDP 605C	COR	20.0	5.0	6.0	8.0	2.0	15.0	340	320	0	40	20	20
63	CDP 6161	COR	75.0	5.0	5.0	7.0	15.0	25.0	320	140	80	60	100	0
64	CDP 604A	COR	50.0	5.0	1.0	2.0	5.0	20.0	0	0	0	0	0	0
65	CDP 604B	COR	20.0	5.0	1.0	2.0	2.0	15.0	0	0	0	0	0	0
66	CDP 604C	COR	15.0	20.0	7.0	8.0	1.0	5.0	0	0	0	0	0	0
67	CDP 604D	COR	20.0	5.0	1.0	2.0	1.0	5.0	0	0	0	0	0	0
68	CDP 602A	COR	15.0	5.0	10.0	13.0	5.0	12.0	760	720	20	100	20	0
69	CDP 602B	COR	50.0	5.0	9.0	11.0	10.0	28.0	720	580	40	180	100	0
70	CDP 602C	COR	10.0	5.0	8.0	11.0	1.0	10.0	560	540	0	20	20	20
71	CDP 602D	COR	20.0	5.0	8.0	11.0	2.0	10.0	500	440	0	40	60	40
72	CDP 603A	COR	25.0	5.0	4.0	6.0	3.0	15.0	300	280	0	20	20	140
73	CDP 603B	COR	20.0	5.0	4.0	6.0	3.0	15.0	220	180	0	20	40	40
74	(UNUSED)	COR	9999.099999	0.0	0.0	0.0	100.0	0.0	0	0	0	0	0	0
75	CDP 6015	FLEA	30.0	5.0	5.0	10.0	10.0	4.0	760	660	40	20	60	60
76	CDP 6195	FLEA	15.0	5.0	5.0	10.0	1.0	0.0	860	840	0	0	20	0
77	CDP 6196	FLEA	10.0	5.0	5.0	10.0	1.0	0.0	0	0	0	0	0	0
78	CDP 6198	FLEA	15.0	5.0	5.0	10.0	1.0	0.0	0	0	0	0	0	0
79	CDP 6118	FLEA	35.0	5.0	5.0	10.0	2.0	2.0	0	0	0	0	0	0
80	CDP 541U	FLEA	15.0	5.0	5.0	10.0	0.0	0.0	0	0	0	0	0	0
81	CDP 4571	FLEA	20.0	5.0	5.0	10.0	1.0	0.0	0	0	0	0	0	0
82	CDP 4172	FLEA	40.0	5.0	5.0	10.0	20.0	15.0	500	320	120	100	60	20
83	CDP 6402	FLEA	20.0	5.0	5.0	10.0	10.0	10.0	320	280	20	0	20	0
84	CDP 6401	FLEA	20.0	5.0	5.0	10.0	0.0	0.0	0	0	0	0	0	0
85	CDP 6460	FLEA	20.0	5.0	5.0	10.0	10.0	10.0	0	0	0	0	0	0
86	CDP 600A	FLEA	30.0	5.0	5.0	10.0	10.0	10.0	0	0	0	0	0	0
87	(UNUSED)	FLEA	9999.099999	0.0	0.0	0.0	100.0	0.0	0	0	0	0	0	0
88	CDP 5200	NIJON	29.0	5.0	8.0	89.0	25.0	15.0	280	260	20	60	0	60
89	CDP 6137	NIJON	60.0	15.0	11.0	12.0	10.0	10.0	180	140	20	0	20	80
90	CDP 6142	NIJON	29.0	5.0	5.0	10.0	8.0	10.0	580	520	0	60	80	80
91	(UNUSED)	NIJON	9999.099999	0.0	0.0	0.0	100.0	0.0	0	0	0	0	0	0
92	CDP 6041	CHAS	80.0	20.0	12.0	13.0	12.0	15.0	0	0	0	0	0	0
93	(UNUSED)	CHAS	9999.099999	0.0	0.0	0.0	100.0	0.0	0	0	0	0	0	0
94	CDP 6131	MAHI	110.0	5.0	14.0	19.0	13.0	25.0	800	480	100	300	220	20
95	(UNUSED)	MARI	9999.099999	0.0	0.0	0.0	100.0	0.0	0	0	0	0	0	0
96	CDP 6079	PORH	40.0	20.0	18.0	31.0	4.0	1.0	400	300	40	0	60	140
97	(UNUSED)	PORH	9999.099999	0.0	0.0	0.0	100.0	0.0	0	0	0	0	0	0
98	CDP 6289	CULP	40.0	20.0	20.0	22.0	4.0	3.0	0	0	0	0	0	0
TOTALS:														
											11420	9960	9600	

TIME = 520.0000			SCALING FACTOR = 0.05			FLEET RETURNEES = 10.00% OF RECRUIT GRADUATES		
	NO.	FINISHING PIPELINE	REG	FUEL	AVERAGE TIME	NEG	FLER	A COUNSES:
1	CE	-	240	20	118.71	86.00		CDP 6079,
2	CFM	-	140	0	318.64	0.00		CDP 605A, CDP 605B, CDP 605C, CDP 6161,
3	DS	-	480	0	208.58	0.00		CDP 6131,
4	EM	-	2680	320	147.97	105.44		CDP 6070,
5	ET A	-EF	320	0	282.13	0.00		CDP 603V,
6	ET N	-F	280	20	221.36	192.00		CDP 604E,
7	ET	-	40	0	267.00	0.00		CDP 603X,
8	E4	-	180	0	304.83	0.00		CDP 602B, CDP 602C, CDP 602D, CDP 603A, CDP 603B,
9	FIG	-	400	60	230.45	178.00		CDP 607A,
10	FIG	-SS	140	0	277.36	0.00		CDP 6376,
11	FTM	-	440	60	251.86	206.67		CDP 6337,
12	GMG	-	560	40	142.32	89.50		CDP 6108,
13	GMM	-	460	0	149.20	0.00		CDP 5200,
14	GNT	-ASRO	270	20	161.95	94.00		CDP 6077,
15	GSE	-	80	0	225.63	0.00		CDP 6115,
16	GSM	-	280	40	234.43	232.50		CDP 606C, CDP 8563, CDP 8565,
17	IC	-	640	60	228.78	191.67		CDP 8563, CDP 8564,
18	RA S	-S	260	20	120.27	87.00		CDP 6073,
19	STG	-	300	60	165.97	122.83		CDP 6060,
20	SIS	-	240	40	248.42	262.50		CDP 6015, CDP 6195,
21	TH	-	600	120	135.83	87.17		CDP 6172, CDP 6402,
22	AE	-AFUN	1720	120	235.50	199.33		CDP 6093,
23	AO	-AFUN	220	0	321.08	0.00		CDP 6218, CDP 6515,
24	ASE	-AFUN	240	20	175.50	150.00		CDP 6240, CDP 6245,
25	AT	-AFUN	460	40	312.15	294.00		CDP 6230, CDP 6530,
26	AX	-AFUN	80	0	371.45	0.00		CDP 6239, CDP 6244,
27	TD	-AFUN	400	0	254.95	0.00		CDP 6241, CDP 6246,
-----								CDP 6222, CDP 6242, CDP 6521,
TOTALS:			12190	1050				

TOTAL NUMBER OF RECRUIT ARRIVALS TO SYSTEM = 40100
TOTAL FLEET RETURNEES = 3160
TOTAL 14 TRANSIT = 40
MAXIMUM NUMBER OF CONCURRENT ENTRIES = 1058 (NOT DESCALED)

APPENDIX D

EXPANDING STUDENT TYPE AND COURSE NUMBER LIMITATIONS

Since students of any given type are split into two categories (regular recruits and fleet returnees), when statistics are specified two numbers are of interest: the maximum number of student types, and the maximum number of types of statistics, MCLCT (twice the number of student types).

The maximum number of histograms is stored in MHIST. There must be two STAT cards for every student type. Not all of these need specify histograms. The maximum number of cells output for all histograms is limited to 500 in the original version of SLAM. To increase this limit, the dimension of array JJCEL (COMMON /GCOM4/) must be increased. Remember, the number of cells printed in a single histogram is equal to the number specified, in the STAT SLAM input card, plus 2 (one cell for the range above and one for the range below that histogrammed).

The maximum number of courses is two (2) less than MFILS. MFILS is the maximum number of files, one of which must be the calendar file, and one must be the scratch file used by the randomizer in RCRT.

MCLCT, MHIST and MFILS are all initialized in SLAM subroutine DDFLT. Table 4 is a listing of labeled COMMON statements and the arrays affected by changing the above parameters.

TABLE 4
COMMON STATEMENTS AND ARRAYS AFFECTED
BY CHANGING LIMITS

Maximum Number of	Current	COMMON	Arrays affected
Student types	49	NAVAL3 NAVAL4 NAVAL9	BE,NASCHD,YD,NROUTE FM,DOWF,BF,PR STYP1,STYP2
Types of statistics (MCLCT)	98	GCOM6 GCOM4	SSOBV LLABC,NNCH
Histograms (MHIST)	98	GCOM4	HHLOW,HHWID,NNCEL,NNHC
Total number of cells in all histograms	500	GCOM4	JJCEL
Files (MFILS)	100	XCOM1 GCOM6 GCOM8	MFE,MLE,NQ EENQ,IINN,KKRNK,MMAQ, QQTIM,VVNQ LLQ
Courses (MFILS-2)	98	NAVAL5 NAVAL6 NAVAL7 NAVAL8 NAVAL9	WHR,NCLSZE,YYY,YYF FLK,BACK FIRSTA,OPD,ADV,FAIL,SBACK MINS,MAXS,WTIME CNAM1,CNAM2

DATA statements in BLOCK DATA:
WTIME,YYY,FLK,BACK,WHR,
NCLSZE,YYF

SOURCE: Pritsker and Associates, "Operational
Procedures for SLAM," 2/1/79, Pritsker and Associates, Inc.,
P. O. Box 2413, W. Lafayette, Indiana 47906.

If more than 1000 student types are to be modeled, a DO
statement in subroutine INTLC must be modified.

APPENDIX E

TIME AND SPACE SAVING IDEAS

1. Use a smaller scaling factor (FACTOR in simulation specification cards).

2. Dimension NSET/QSET and NAVY to smallest possible dimensions. This will also save time by minimizing the number of page faults required in virtual memory systems. Major reductions in the sizes of NSET/QSET and NAVY might be realized by combining attributes into codes (e.g., $ATTRIB(1) = \text{pointer} + \text{priority} / 1000.$; Aux. Atrib. $1 = \text{arrival time} + \text{type} * 10000 + \text{file/ordinal number} * 10^{**} 6 + \text{arrival time to queue} * 10^{**} 8.$). Although the code system is more susceptible to bugs, more difficult to maintain, and requires more CPU time for encoding and decoding, NSET/QSET and NAVY might be reducible to 1/4 their current sizes, thereby reducing the number of time-consuming page faults.

3. Set maximum numbers of courses and student types as small as possible. Other SLAM arrays can be shrunk according to reference 2. They are listed in Table 5.

4. Several COMMON statements can be shortened or even dropped from several subroutines. If one shortens a COMMON statement, though, one must be very careful about adding back to it.

TABLE 5
SLAM ARRAY SIZES

	Used by this program	Normally available with SLAM
Number of state equations	0	100
Number of state conditions	0	25
Number of Enter nodes	0	25
Number of Resource types	12	75
Number of Gates	0	25
Number of Histograms (twice the number of student types)	98	25
Number of cells per histogram	20	500
Number of COLCT variables (twice the number of student types)	98	25
Number of TIMST variables	0	25
Number of entity attributes	2	100
Number of files	100	100
Number of plots	0	10
Number of variables per plot	0	10
Number of random number streams	6	10
Maximum number that can be assigned to an activity	15	100
Number of Nodes	36	500
Number of activities with a STOPA specification	0	50

SOURCE: Pritsker and Associates, OPERATIONAL
PROCEDURES FOR SLAM.

5. Several arrays and variables may be reduced to one or two
bytes.

Good candidates for 2-byte variables and arrays include:

Variables
MSTUD, TBR, TBS, FACTOR,
INTRAN, MAXSYS, MAXAUX,
FIRSTA, OPD, ADV, FLER

Arrays
NAVY, STUD, YD, FM, DOWF,
BF, PR, NCLSZE
MINS, MAXS, FAIL, SBACK

Candidates for 1-byte variables include:

ORL, GL, SD, MEM, COR, ...	WHR, BE, NASCHD, NROUTE,
IECHO, ICASE, MM, LASTA, LIMCOU	
IFAT, NST, NST1, NSTUD	

APPENDIX F

ERROR MESSAGES

Two types of error messages are available: those issued by the model program and those issued by the SLAM processor. Of those issued by the model program, the ones headed with the word ERROR will prevent execution beyond reading and checking the data. The ones headed with the word WARNING will not prevent further execution, but will point out conditions the analyst may wish to be aware of.

Some of the SLAM error messages and possible causes are listed below:

1.

FATAL EXECUTION ERROR TYPE 1004 DETECTED BY SLAM AT TIME xxxxxxxx

ARGUMENT ICLT = xx IS OUT OF RANGE

This will come toward the end of the first Model report if enough STAT cards have not been provided.

2. **SLAM INPUT ERROR - DIMENSION OF NSET/QSET EXCEEDED**

This error will appear in the listing of SLAM statements (if ILIST = 'Y' on the GEN card -- otherwise it will appear alone) when the dimension on NSET/QSET has not been made large enough to accommodate all students forecast by the LIMITS card (MNTRY). If NSET/QSET is large enough to accommodate all the students but not all

the network coding, the error will appear in the network listing at the point where the processor runs out of room in the array for the coding of the SLAM input statements.

APPENDIX G

SLAM MODIFICATIONS

A copy of version 1.0, release 1.0 of SLAM (co. 1979) was used. The following modifications have been made:

1. The most troublesome modification was the random number generator, function DRAND. DRAND supplies 10 pseudorandom number (PN) streams. In its current form DRAND is suitable for 32-bit binary machines which allow overflow when multiplying. If the present form of DRAND is found to be inappropriate for the machine used, a suitable PN generator must be substituted. Reference 2 lists two PN generators.

2. Subroutine FFILE was modified from:

```
10 WRITE (NPRNT,230) TNOW
```

to:

```
10 CALL REP2  
   CALL SUMRY  
   WRITE (NPRNT,230) TNOW
```

in order to get Model and SLAM summary reports before aborting in the event of overflow conditions.

3. The main program and subroutines EVENT and INTLC were replaced.

4. The maximum number of histograms (MHIST) and COLCT variables (MCLCT) were raised to from 25 to 98.

5. Disabling the statment 'WRITE (NPRNT,2190)' in SLAM subroutine DDATN will elimitate the one-page heading at the top of all outputs.

GLOSSARY OF PROGRAM VARIABLES

ADV(K): duration of course K in working days (real).

ATRI(i): see ATTRIBUTES, REGULAR.

ATTRIBUTES, AUXILIARY:

STUD(1): arrival time (to system).

STUD(2): student type (NS).

STUD(3): file number of most recent BEE, ordinal number of "A" course in sequence.

STUD(4): queue entry time, used for determining waiting time in file.

ATTRIBUTES, REGULAR:

ATRI(1): pointer for auxiliary attributes (negative indicates course student is going to fail), also course number for class-convening event and student type briefly in RCRT.

ATRI(2): student priority for courses with prioritized queues. (HVF). Students with higher values in attribute 2 have higher priority.

BACK(K): number who have been setback in course K (integer).

BE(NS): 1,2 or 3. Number of BEE courses student type NS takes. Everyone starts with BEE1 and takes consecutive BEE courses up to and including the last one. Negative number indicates A before BEE (positive of AFUNs)(integer).

BF(NS,NB): Base factor (fraction of type NS which arrive at base NB for basic training).

CHS: Number of "A" courses taught at Charleston (integer).

CNAM1(K): first part of course name (4 characters).

CNAM2(K): second part of course name (4 characters).

COR: Number of "A" courses taught at Corry Station(integer).

DOWF(NS,ND): Day Of Week Factor -- of typical total weekly input of student type NS, this is the fraction which comes on the NDth weekday (average = 1/5).

FACTOR: scaling factor for the entire simulation = decimal fraction of specified students which are actually simulated.

FAIL(K): percentage of students who take course K who will fail (changed to a decimal fraction in subroutine INTLC).

FIRSTA(K): number of working days from the beginning until the first convening of course K.

FLEA: Number of "A" courses taught at Fleaswtrac (integer).

FLER: percentage of basic training graduates which will be cloned as fleet returnees.

FLK(K): number who have failed course K (integer).

FM(NS,MO): month factor for type NS student average = 1/12 (fraction of type NS which arrives to basic training in month MO).
 FSTR: time until first Model report.
 FSTS: time until first snapshot.
 GL: Number of "A" courses taught at Great Lakes (integer).
 GULP: Number of "A" courses taught at Gulf Port (integer).
 ICASE: 0 to simulate all A schools.
 1 to simulate only A schools in Orlando.
 2 to simulate only A schools in Great Lakes.
 3 to simulate only A schools in San Diego.
 IECHO: Flag for echo-print of data and preliminary data analysis Data is echoed and analysis is performed when IECHO \neq 0.
 IHIST: 1 will cause histograms to be printed in Model reports.
 K: course (file) number.
 KKK: counter for number of students who arrive to basic training.
 LASTA: File number of last "A" course.
 LIMCOU: Maximum of LASTA and 25 -- used to determine class ending event codes.
 MARI: Number of "A" courses taught at Mare Island (integer).
 MAXAUX: Largest value of MSTUD (MSTUD = pointer for auxiliary attributes): max. no. of aux. attributes stored (adjusted daily in RCRT and FLERN).
 MAXS(K): maximum number students allowed in a class for course K (real).
 MAXSYS: Largest value of MFA = max. no. of entries * 6 (adjusted daily in RCRT and FLERN).
 MEM: Number of "A" courses taught at Memphis (integer).
 MINS(K): minimum number of students required to form a class for course K (real).
 MM: number of student types.
 MSTUD: pointer to auxiliary attributes (in NAVY).
 NAVY(i) stores all auxiliary attributes for all nonfailing students (real).
 NCLSZ(K): number of students currently under instruction in course K.
 NLON: Number of "A" courses taught at New London (integer).
 NROUTE(NS,NB): Specifies where a student of type NS is to go for BEE from base NB.
 NS: student type.
 NST: first file for "A" courses.
 NST1: NST - 1.
 NSTUD: number of auxiliary attributes per entity.
 OPD(K): number of days between class convenings of course K.
 ORL: Number of "A" courses taught at Orlando (integer).
 PLACE(i): 4-character name of place i; PLACE(12) = 'BEE'; PLACE(13) = 'ASCH'.
 PORH: Number of "A" courses taught at Port Hueneme (integer).
 PR(NS): priority for BEE for student type NS (real).

PSEUDORANDOM NUMBER STREAMS:

- 1: generates random rounding element in RCRT for determining number of arrivals for the day.
- 2: used to determine pass/setback/fail in FAILUR.
- 3: used in generating fleet returnees (TRAVEL).
- 4: used in determining travel time (TRTIME).
- 5: used in randomizing new recruits (RCRT).
- 6: used in rounding MINS and MAXS when determining class sizes.
- 7: may be used (but currently is not) for pass/fail setback decision for basic training.

RANDOM NUMBER STREAMS -- see PSEUDORANDOM NUMBER STREAMS.

SBACK(K): percentage of students who take course K who will be setback (changed to SBACK(K) + FAIL(K) in INTLC).

SD: Number of "A" courses taught at San Diego (integer).

STUD(i): see ATTRIBUTES, AUXILIARY.

STYP1(NS): first part of rating (name) of student type NS (EM, ET, etc.)(4 characters).

STYP2(NS): second part of name of student type (: 'AFUN' for AFUN students)(4 characters).

TBE: Time before deliberate error report.

TBR: Time Between Model Reports.

TBS: Time Between Snapshots.

WHR(K): An array which tells where course K is taught (integer).

WTIME (K): total waiting time for all students who have waited for course K; scratch registers for feed rates in preliminary data analysis.

YD(NS, K): Total yearly demand for entire Navy for student type NS for year K (real).

YYF(K): number of graduates from course K (number finishing successfully)(integer).

YYY(K): number of students who have arrived to course K (integer).

NAVY ABBREVIATIONS

ABE: Aviation Boatswain's Mate - Equipment
ABF: Aviation Boatswain's Mate - Fuels
ABH: Aviation Boatswain's Mate - Handling
AC: Air Traffic Controller
AD: Aviation Machinist's Mate
AE: Aviation Electrician's Mate
AFFR: Aircraft Firefighting and Rescue
AFUN: Aviation Fundamentals
AG: Aerographer's Mate
AK: Aviation Storekeeper
AME: Aviation Structural Mechanic - Safety Equipment
AMH: Aviation Structural Mechanic - Hydraulics
AMS: Aviation Structural Mechanic - Structures
AO: Aviation Ordnanceman
AQ: Avionics Tech - Aviation Control Technician
ASE: Aviation Support Equipment Technician - Electrical
ASH/ASM
AT: Avionics Technician
AV: Non-Navy (Avionics Technician)
AW: Aviation Anti-Submarine Warfare Operator
AX: Avionics Technician Aviation Anti-Submarine Warfare
AZ: Aviation Maintenance Administration
BASHEL: Basic Helicopter
BEE: Basic Electricity and Electronics
BT: Boiler Technician PSI
BU: Builder
CDP: Course Data Processing number
CE: Construction Electrician
CIN: Catalog Identifying Number
CM: Construction Mechanic
CTA: Cryptologic Technician
CTI: Cryptologic Technician
CTM: Cryptologic Maintenance Technician
CTO: Cryptologic Technician O
CTR: Cryptologic Technician R
CTT: Cryptologic Technician T
DK: Disbursing Clerk
DF: Data Processing Technician
DS: Data Systems Technician
DT: Dental Technician
EA: Engineering Aid
EM: Electrician's Mate

EM: NF
EN: Engineman
EO: Equipment Operator
ET-AEF: Electronics Technician Adv Electronics
ET-NF: Electronics Technician Nuclear Field
ET: Electronics Technician Others
ET SS: Electronics Technician Submarine
EW: Electronic Warfare Technician
FTG: Fire Control Technician Guns
FTG SS: Fire Control Technician Underwater
FTM: Fire Control Technician Missiles
GMG: Gunner's Mate Guns
GMM: Gunner's Mate Missiles
GMT ASROC: Gunner's Mate Technician (ASROC)
GMT: Gunner's Mate Technician
GSE: Gas Turbine Systems Technician Elec
GSM: Gas Turbine Systems Technician Mech
HM: Hospitalman
HT: Hull Maintenance Technician
IC: Interior Communications Electrician
IM: Instrument Man
IS: Intelligence Specialist
JO: Information Specialist Journalist
LI: Lithographer
ML: Molder
MM: Machinist's Mate
MMN: Machinist's Mate (NF)
MN: Mineman
MR: Machinery Repairman
MS: Mess Management Specialist
MU: Music Basic
NUCP: Nuclear Power (must be from EM, ET, MM pipelines)
OM: Opticalman
OS: Operations Specialist
OT: Ocean Systems Technician
PC: Postal Clerk
PH: Photographer's Mate
PM: Pattern Maker
PN: Personnelman
PR: Aircrew Survival Equipmentman
QM: Quartermaster
RM: Radioman
RM(SS): Radioman Submarines
RP: Religious Program Specialist
SH: Ship's Serviceman
SK: Storekeeper
SM: Signalman
STG: Surface Sonar Technician
STS: Sonar Subsystem Technician - Submarine

SW: Steelworker
SWS Elec: Strategic Weapon Sys Electronics (from FTB, MT, EN)
TD: Avionics Technician Tradevmen
TM: Torpedoman's Mate Surface Oper
TM: Torpedoman's Mate Submarine Oper
TM: Torpedoman's Mate Technician
TMS: Training Management Systems
UT: Utilitiesman
YN: Yeoman

REFERENCES

- O'Reilly, Jean. Technical Reference Manual for the SLAM Simulation Program. West Lafayette, IN: Pritsker and Assoc., Inc., July, 1980.
- Pritsker and Associates, Inc., "Operational Procedures for SLAM" (internal publication). West Lafayette, Indiana, February 1, 1979.
- Pritsker, A. Alan B., and Pegden, Claude Dennis. Introduction to Simulation and SLAM. New York: Wiley and Sons, 1979.
- U.S. Department of Defense. Official Table of Distances, Continental United States, Alaska, Hawaii, Canada, Canal Zone, Central America, Mexico, Puerto Rico. Army Regulation No. 55-60 / Navy Publication P-2471 / Air Force Regulation No. 177-135. Washington D. C.: 1 December 1979.

INDEX

ALTER, SLAM subroutine	48
ASCHK, listing	116
ASCHK, subroutine	72
Attribute, auxiliary	38, 41, 43, 68
Attribute, regular	38
AWAIT node	65
BEE school	9
BEECHK, listing	116
BEECHK, subroutine	71
Calendar, event	41, 98
Class, defined	5
CLASS, listing	125
Class, lockstep	5, 42
CLASS, subroutine	47, 49, 66, 98
CLASSE, flowchart	67
CLASSE, listing	126
CLASSE, subroutine	47, 50, 66, 75, 77-78, 98
Cloning	8, 25, 27, 34, 39, 54, 58-60, 64, 82, 86, 158
COLCT, SLAM subroutine	68
COPAA, SLAM subroutine	46
Course, defined	5
Course, self-paced	6, 44
Data, course	7
Data, student	7
DDFLT, SLAM subroutine	149
DOWF, array	52
DRAND, SLAM function	58
Driver	6-8, 20, 23, 33, 48
ENTER node	65
ENTER, SLAM subroutine	64-65
ERRORA, listing	134
ERRORA, subroutine	25, 70
Event associated with an entity	42
EVENT node	99
EVENT node coding	99
Event not associated with an entity	41-42
Event, current	98
Event, defined	41
Event, future	41, 98

EVENT, listing	116
EVENT, network, coding	99
EVENT, subroutine	41, 47, 50, 59, 68, 99, 156
FACTOR	23
FAILUR, listing	128
FAILUR, subroutine	40, 47, 68, 75, 96, 160
FAT, listing	113
FAT, subroutine	49
FFILE, SLAM subroutine	69
FILEM, SLAM subroutine	51
Fleet returnees	6
FLER	8, 58
FLERN, listing	122
FLERN, subroutine	47, 58-59, 62, 64, 159
GETAA, SLAM subroutine	46, 68
Histogram	18-19, 23-24, 32, 36, 69, 86, 149, 152, 156
INTLC, listing	109
INTLC, subroutine	45-46, 52, 69, 150, 156
Main program, listing	108
NAVY, array	46
NROUTE, array	57
PRECHK, listing	114
PRECHK, subroutine	71, 73
Priority	10-11, 14, 16-17, 25, 27, 38, 42, 52-53, 58, 64, 82, 85-86, 98, 151
PRNTH, SLAM subroutine	19, 69
Program, main	32
PUTAA, SLAM subroutine	47, 51, 64
Randomizer	17, 40, 52-53
Rating, defined	4
RCRT, listing	118
RCRT, subroutine	17, 47, 51, 75
Report, error	14, 25
REPORT, listing	129
Report, model	14, 24, 34-36
Report, SLAM summary	15
Report, status	14, 24
Resources	43
ROUTE, listing	123

ROUTE, subroutine	47, 57, 64, 77, 98
Scaling	25, 48
SCHDL, SLAM subroutine	41
SETAA, SLAM subroutine	46
Setback, defined	8
SLAM processor	44
Snapshot	14, 24
STATS, listing	128
STATS, subroutine	60, 68, 77-78
STATUS, listing	132
Stopping the simulation	19
TDST, listing	124
TDST, subroutine	47, 50, 59-60, 64-65, 77-78, 98
Tiebreaking	50, 99
Time, CPU	22
Timing	11, 42, 65, 97
Travel time	9, 8, 10, 50, 53, 61, 97
Travel time, table	61
TRAVEL, listing	119
TRAVEL, subroutine	47, 50, 55, 57, 67, 75, 77, 98
TRTIME, function	61
TRTIME, listing	122
Type, A-before-BEE, defined	4
Type, AFUN, defined	4
Type, regular, defined	4
Waiting time	35
WHERE, listing	113
WHERE, subroutine	49
WHR, array	48