

AD-A071 576

RUTGERS - THE STATE UNIV PISCATAWAY NJ COLL OF ENGIN--ETC F/G 9/2
AUTOMATED COURSE SCHEDULING SYSTEM FOR NAVAL TRAINING, (U)
JUN 79 B W LIN, G W HODAK

N61339-78-C-0029

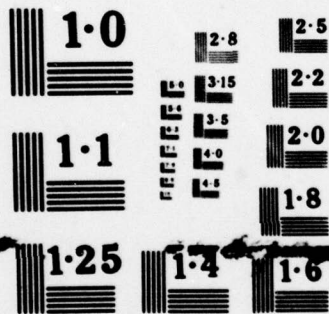
UNCLASSIFIED

TAEG-72

NL

1 OF 2
AD
A071576





NATIONAL BUREAU OF STANDARDS
MICROCOPY RESOLUTION TEST CHART

T A E G

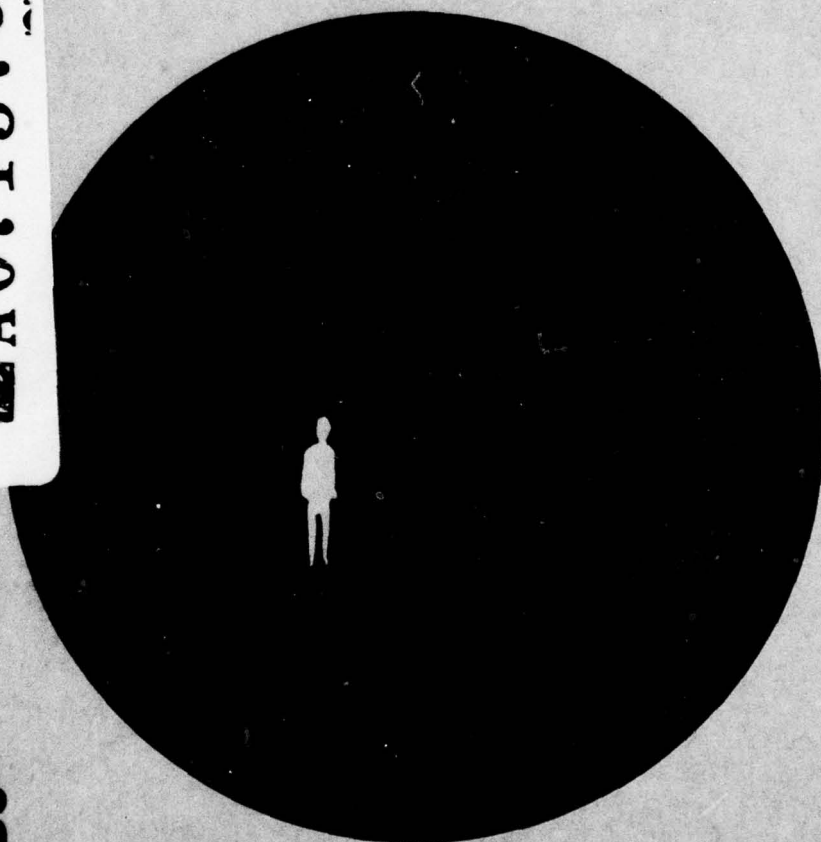
TRAINING
ANALYSIS
AND
EVALUATION
GROUP

TAEG REPORT
NO. 72

AUTOMATED COURSE SCHEDULING SYSTEM
FOR NAVAL TRAINING

AD A 071576

DDC FILE COPY.



LEVEL II
FOCUS
ON
THE
TRAINED
MAN

DDC
REFINED
JUL 24 1979
REGULATED
D

APPROVED FOR PUBLIC RELEASE;
DISTRIBUTION IS UNLIMITED

79 07 23 065

JUNE 1979

TRAINING ANALYSIS AND EVALUATION GROUP

ORLANDO FLORIDA 32813

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

TAEG Report No. 72

AUTOMATED COURSE SCHEDULING SYSTEM
FOR NAVAL TRAINING

Benjamin W. Lin
Rutgers University

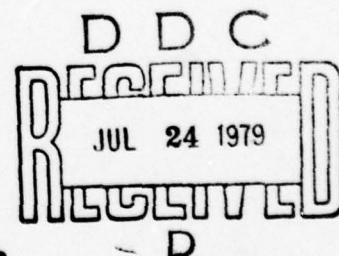
and

Gary W. Hodak
Training Analysis and Evaluation Group

June 1979

GOVERNMENT RIGHTS IN DATA STATEMENT

Reproduction of this publication in whole or in part is permitted for any purpose of the United States Government.



Alfred F. Smode

ALFRED F. SMODE, Ph.D., Director,
Training Analysis and Evaluation Group

Worth Scanland

WORTH SCANLAND, Ph.D.
Assistant Chief of Staff for
Research and Program Development
Chief of Naval Education and Training

DISTRIBUTION STATEMENT A

Approved for public release;
Distribution Unlimited

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER (18) TAEG Report No. 72	2. GOVT ACCESSION NO. ✓	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) (6) AUTOMATED COURSE SCHEDULING SYSTEM FOR NAVAL TRAINING		5. TYPE OF REPORT & PERIOD COVERED
7. AUTHOR(s) (10) Benjamin W. Lin Gary W. Hodak		6. PERFORMING ORG. REPORT NUMBER
9. PERFORMING ORGANIZATION NAME AND ADDRESS College of Engineering Rutgers University Piscataway, New Jersey 08854		8. CONTRACT OR GRANT NUMBER(s) (15) N61339-78-C-0029 New
11. CONTROLLING OFFICE NAME AND ADDRESS Training Analysis and Evaluation Group Orlando, Florida 32813		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) (12) 107p.		11. REPORT DATE June 1979
		12. NUMBER OF PAGES 98
		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for Public Release; Distribution is Unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Scheduling Navy Technical Training Computer-Based Models Training Resource Utilization		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) → Throughout the Naval Education and Training Command (NAVEDTRACOM), the generation and maintenance of a feasible schedule for Navy training courses is labor-intensive. Minimal guidance is provided, resulting in scheduling processes subject to the vagaries of individual style and competency. In view of these problems, an effort is being made to develop a scheduling methodology that will assist Navy training-complex personnel in achieving an optimal/quasi-optimal schedule in a quick and easy way. →		

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102- LF-014-6601

Unclassified 393493
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

20. ABSTRACT (continued)

The Automated Course Scheduling System (ACSS) reported herein was developed to relieve Naval training personnel from the burden of the labor-intensive scheduling process and provide them with the capability of studying all possible schedules to achieve better utilization of school resources. The ACSS is a high-level, interactive, and user-oriented system intended for use at the school level. Its design is based on limited and low-cost hardware requirements. The system can be implemented readily on a Wang 2200 system with a dual floppy diskette drive and requires a memory of 16-K bytes.

With the support of a number of data bases, the ACSS can be used to perform: (1) random retrieval of training-course data; (2) feasibility assessment of training demands under available resources, (3) documentation of schedule work sheets, and (4) editing and printing of schedules. Presently, the research effort is directed toward evaluating and field-testing the system at various activities.

S/N 0102- LF- 014- 6601

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

FOREWORD

This task is a subelement of the Design of Training Systems (DOTS) project. The objectives of the DOTS program are in consonance with the requirements of Advanced Development Objective ZPN07 (formerly ADO 43-03X), Education and Training Development. ZPN07 includes a number of projects concerned with demonstrating and evaluating the technical, operational, and financial feasibility of applying advanced technological applications to improving the training process.

The Bureau of Naval Personnel initiated the original ADO in 1966 to make Naval training more responsive to the changing times. As one project under this effort, DOTS was designed to improve the process of managing training resources through application of the techniques of system analysis and system simulation as accomplished through mathematical modeling. The end objective is a family of computerized mathematical models enabling training management to more rapidly predict the impact of changes in training resource availability of requirements.

The Training Analysis and Evaluation Group, Dr. A. F. Smode, Director and Mr. M. G. Middleton, complemented the contracted effort by providing direction and guidance and establishing organizational interfaces. Appreciation is also acknowledged to Dr. M. M. Zajkowski for his editorial assistance, Mr. C. Guitard for his assistance in programming, and Mr. W. H. Lindahl (now of the Navy Manpower Reserve Affairs and Logistics) for his assistance in the early stages of the study.

TABLE OF CONTENTS		
Section		Page
I	INTRODUCTION	1
	Background	1
	Overview of the ACSS	2
	Purpose	2
	Organization of the Report	2
II	DESCRIPTION OF THE AUTOMATED COURSE SCHEDULING SYSTEM	3
	System Overview	3
	The ACSS Data Base	5
	Structure of the Data Base Module	6
	Structure of the Feasibility Study Module	8
	Structure of the Work Sheet Print Module	8
	Structure of the Schedule Plot Module	12
III	SUMMARY	15
Appendix		
A	USER'S GUIDE	16
	Implementation Guide	17
	Initialization	17
	Operating Guide	21
	An Illustration	27
B	ACSS PROGRAM LISTING	34

LIST OF FIGURES

Figure	Title	Page
II-1	System Schematic of the ACSS	4
II-2	Schematic of the Data Base Module	7
II-3	Schematic of the Feasibility Study Module	9
II-4	Logic for Course Requirements	10
II-5	Logic for Instructor Requirements	11
II-6	Schematic of the Work Sheet Print Module	13
II-7	Schematic of the Schedule Plot Module	14
A-1	Contents of the Program File Diskette	18
A-2	Contents of the Data File Diskette	18
A-3	Re-initialization of the Course File	19
A-4	Specifications of the Course File	19
A-5	Re-initialization of the Plot File	20
A-6	Specifications of the Plot File	20
A-7	System Control Menu	22
A-8	Data Base Maintenance Menu	22
A-9	Course Data Input Formats	24
A-10	Feasibility Study Sample Output	24
A-11	Course Schedule Work Sheet Sample Output	25
A-12	Course Schedule Plot Sample Output	26
A-13	Course Schedule Work Sheet	32
A-14	Course Schedule Plot	33

SECTION 1

INTRODUCTION

BACKGROUND

The scheduling process for training courses in the Naval Education and Training Command (NAVEDTRACOM) is labor-intensive. It entails generating and maintaining schedules to meet planned input requirements, subject to the suitability/availability of instructors, equipment, and facilities. In the process, schedules must not only be established for the current year, updated, and revised as necessary, but also projected for the out-year planning requirements of the 5-year defense plan. As a result, the goal of the scheduling process is to arrive at a feasible plan, even though that plan may not be optimal in terms of the utilization of school resources. Trade-offs in schedules are made by exception when a crisis situation arises.

Besides its labor intensiveness, guidance is minimal, resulting in scheduling processes that are subject to the vagaries of individual style and competency. In view of these problems, a study was conducted and reported in "An Heuristic Approach for the Scheduling of Navy Training Courses."¹ Findings of that study suggested that potential significant savings might result from the automation and optimization of the scheduling process. Furthermore, the study recommended development of a scheduling methodology that would help Navy training-complex personnel achieve an optimum schedule in a quick and easy way.

As a result of these findings and recommendations, a three-phase study has been undertaken. The first phase is to document and automate the present scheduling process with the objective of arriving at a feasible schedule easily and quickly. In the second phase, an optimum/quasi-optimum scheduling system will be developed. An optimum schedule can be defined either to optimize the utilization of school resources in meeting training requirements, or to maximize the training inputs subject to the suitability/availability of school resources. In the third and last phase, the research team will act as an innovation/change agent to assist Naval technical training schools in implementation of the developed system. The Automated Course Scheduling System (ACSS) reported herein is the product of the first phase.

¹ Lindahl, W.H., and B.W. Lin, 1977. An Heuristic Approach for the Scheduling of Navy Training Courses. TAEG Report No. 52, Training Analysis and Evaluation Group, Orlando, Florida.

OVERVIEW OF THE ACSS

The ACSS was developed to relieve Naval training personnel from the burden of the labor-intensive manual scheduling process and provide them with the capability of studying all possible schedules to achieve better utilization of school resources. Specifically, the automated system can be used to perform these functions:

- a. Storage of training course data.
- b. Random retrieval of training course data.
- c. Feasibility assessment of schedules.
- d. Documentation of schedule work sheets.
- e. Print feasible schedules for documentation.

PURPOSE

The purposes of this report are to:

- a. Describe the structure of the ACSS and its elements, along with the logic in each element.
- b. Provide a user's guide to familiarize Naval training-complex personnel with the use of the ACSS.
- c. Document the programming details of the ACSS.

ORGANIZATION OF THE REPORT

In addition to this introduction, two other sections are included in this report. Section II presents a description of the ACSS system and explains how each module in the system works. Section III is a summary of the research effort in Phase I. In Appendix A, a guide is provided on how to implement and modify the ACSS system if the school operates differently from some of the assumptions made in the system. A step-by-step procedure is supplied to illustrate how to use the system. Appendix B contains a listing of the computer program for the ACSS.

SECTION II

DESCRIPTION OF THE AUTOMATED COURSE SCHEDULING SYSTEM (ACSS)

The purpose of this section is to describe the structure of the ACSS and its elements. Since the ACSS evolved from the initial study, "An Heuristic Approach for the Scheduling of Navy Training Courses," some of the system logic used in the ACSS is derived from that study. The current manual scheduling process, course requirements, and instructor requirements are well documented in that report.

The ACSS is a high-level, interactive, and user-oriented system intended for use at the school level. Its design is based on limited and low-cost hardware requirements. The system can be readily implemented on a Wang 2200 system with a dual floppy diskette drive and requires a memory of 16-K bytes. The ACSS is packaged in two diskettes--one diskette contains the schedule program files and the other contains the schedule data files. Currently the system can handle up to 600 courses, and can be tailored to the specific need of the school.

SYSTEM OVERVIEW

Figure II-1 provides the system schematic of the ACSS. The SCHEDULE is the system control interfacing the user to the four modules--the data base module(SCH.BASE), the feasibility study module(SCH.FEAD), the print work sheet module(SCH.WSHT), the print schedule module(SCH.PLT). The data base module creates and maintains the course file with a capacity of storing as many as 600 courses. The feasibility study module is used to help assess schedule feasibility. If a schedule is infeasible, the module can be executed iteratively to determine a feasible schedule. Both the print work sheet and print schedule modules are used for documentation purposes. While the schedule work sheets provide such essential information as the number of convenings, instructor, and trainer requirements, the course schedule plots lay out the beginning and ending dates of each class convening for a desired course.

After the course data base has been created, the user can access it directly from each of the modules. When the user is done with the module just being worked on, the program control will always return to the SCHEDULE module and display the system control menu on the screen for further execution. Since the entire program requires a memory of approximately 32-K bytes (well over the assumed memory capacity), overlaying techniques are used which load only the portion of the program to be executed. Consequently, the user will encounter seconds of pause when some portion of the program is being loaded, and program loading times may vary due to the unequal size of the modules.

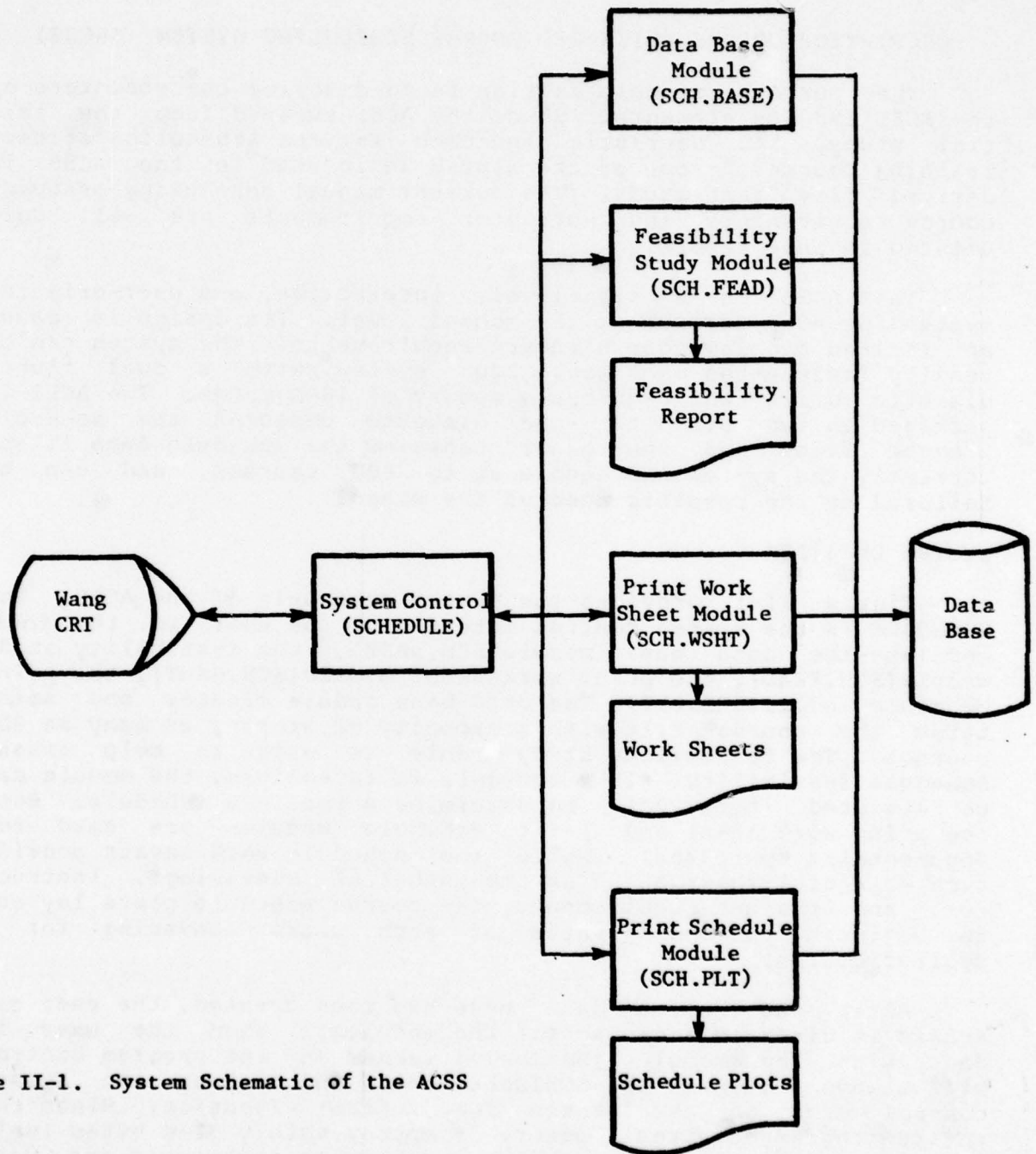


Figure II-1. System Schematic of the ACSS

THE ACSS DATA BASE

The ACSS Data Base is a collection of training course records, which provides data required for operation of the ACSS. Its creation and maintenance is a function of the training-complex personnel at the specific activity. With the Data Base Module to be described later, the user can add and delete courses, change data elements in a course, and examine the file contents in the data base. Courses in the course file can be accessed in a random manner.

The ACSS Data Base is implemented using Wang's Key File Access Method (KFAM) routines. KFAM is a system for storing and retrieving individual records in a file. For each file of records, KFAM creates and maintains an index of the individual records and their locations in the file. The index called the key is used to distinguish the record it identifies from all other records. The file of keys called the key file is itself kept as a cataloged file on a diskette. Associated with the key file is the user file containing the file of records. Thus, when the user instructs in a program, for instance, "FIND ME THE RECORD FOR COURSE NUMBER A123," KFAM routines incorporated into the program will search the key file index and put the address of record A123 into the user file's current address parameter in the Device Table. The user can then read off the data of the record for use. For more details on KFAM, refer to Wang's "ISS Manual."

Before key and user files can be used for storing keys and course data, the files first must be formatted or initialized. Setup utilities are used for this purpose. A guide to initializing such files for the ACSS Data Base is provided in the User's Guide in Appendix A.

As the Course Data Processing (CDP) code is a unique identifier in the Navy, the code is used as the key for course identification. The data elements contained in each course record are:

- a. Course Data Processing (CDP) Code--a code that identifies each course of instruction by location. Identical courses taught in different locations will have individual CDP codes. It becomes the enlisted Training History Code.
- b. Course Identification Number(CIN)--the 10-position alphanumeric CIN code used in the Catalog of Navy Training Courses. This identifies the command sponsoring the course, the DOD skill for which the course trains and the sequence number to facilitate locating the course within the course catalog.
- c. Course Short Title--abbreviated descriptive title of the course.

- d. Planned Input--the current total anticipated annual demand for the course.
- e. Class Size--the current limiting capacity for the course.
- f. Course Length--the number of calendar days that expire from the class convening date (including the class convening date) to the class graduation date (including the class graduation date).
- g. Student/Instructor Ratio--the ratio of students to each instructor. Since each phase of training associated with the course may require a different ratio, several ratios may be entered for each course. In this report the term is shortened as "Contact Ratio."
- h. Total Course Contact Hours--the total contact hours is the number on the Master Schedule for group-paced courses and the average total contact hours for self-paced courses. The total instructional contact hours taught at the various ratios must equal the total course contact hours.
- i. Instructors Available--the number of instructors available for the course.
- j. Other Instructors Available--the number of non-Navy instructors (e.g. Coast Guard) available for the course.
- k. Trainers Available--the number of simulated and/or operational training equipment available for the course.
- l. Support Available--the number of support personnel available for the course.
- m. Support Manpower Authorization (Support MPA)--the number of support personnel required for the course.

All the data elements except items i, j, k, l, and m are available in the Master Course Reference File (MCRF) EXTRACT TAPE of the Navy Integrated Training Resources and Administration System (NITRAS). Thus, there are two ways to create the course data file (user file). One way is to input course data course by course. The other way, easier if equipment for transferring data from tape to disk is available, is to extract the required data from a NITRAS MCRF EXTRACT TAPE. After the extraction is completed, any required data not available in the NITRAS TAPE can be added for each course.

STRUCTURE OF THE DATA BASE MODULE

A schematic of the ACSS Data Base module structure is shown in Figure II-2. Besides the key file and the course file discussed earlier, the module also contains the Course Delete File. The Course Delete File keeps track of the addresses of the deleted

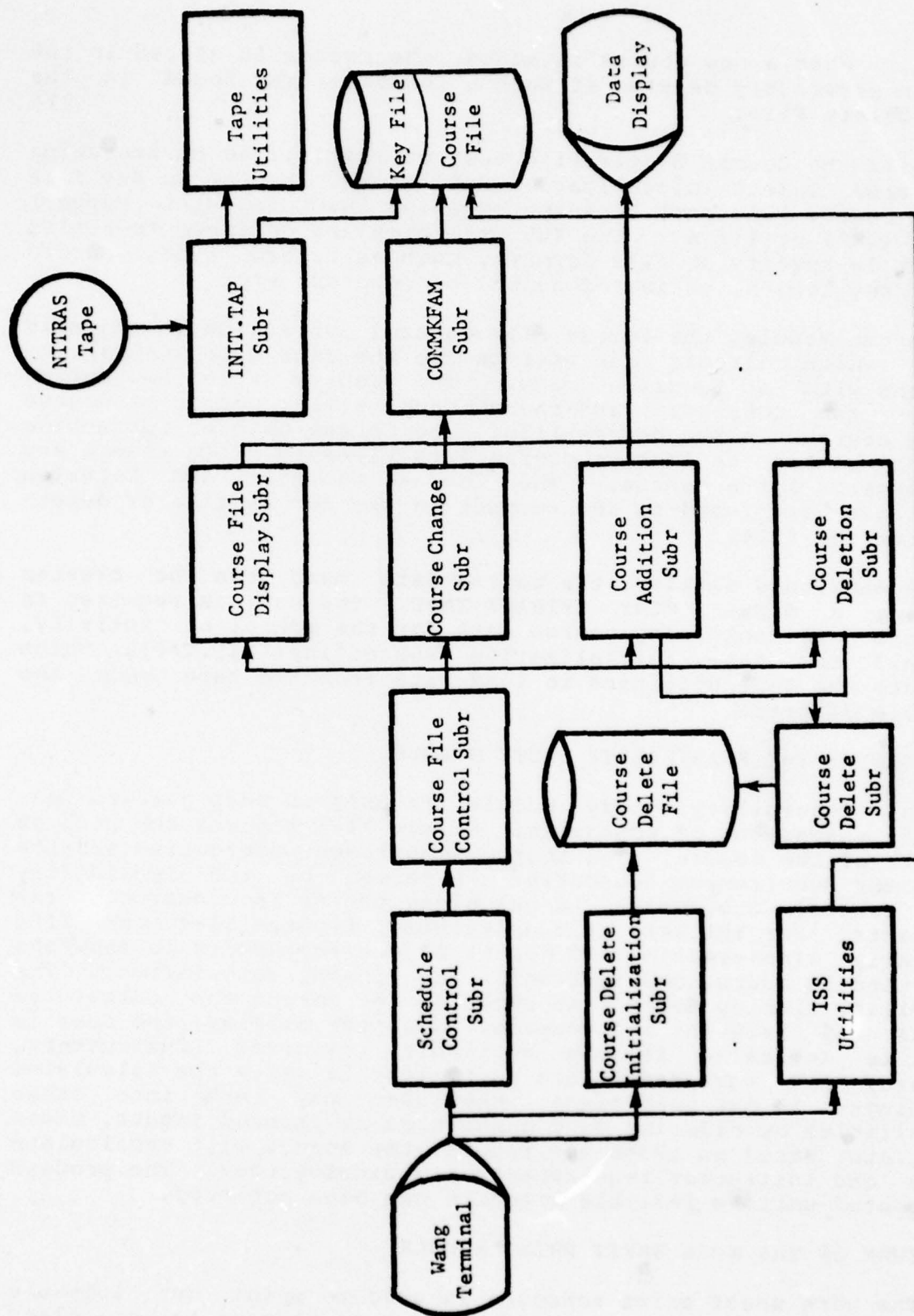


Figure II-2. Schematic of the Data Base Module

courses. When a new course is added, the course is stored in the location previously deleted if such a location is found in the Course Delete File.

While the Course Delete File must be initialized by executing the Course Delete Initialization Subroutine, the Course Key File and the Course File must be formatted using the Integrated Support System (ISS) Utilities. The ISS Utilities are designed to assist the user in specifying file formats, such as record type, record length, key length, estimated number of records, etc.

In the module, the Course File Control Subroutine is the main control which directs the user to use the four data manipulation functions with a function menu. The Course File Subroutine displays the CDP, CIN, and course short title, course by course for all courses in the course file. The Course Change Subroutine is used either to look up course data elements or to change any data elements for a course. The Course Addition and Deletion Subroutines are used to add courses to the course file or delete them from the file.

As mentioned earlier, the course data base can be created utilizing a NITRAS MCRF EXTRACT TAPE. The user is required to mount the tape containing course data for the school or activity, and call the tape initialization subroutine (INIT.TAP), which instructs the Tape Utilities to load data from the tape onto the data file diskette.

STRUCTURE OF THE FEASIBILITY STUDY MODULE

The feasibility study module is used to help perform feasibility assessment of schedules. Figure II-3 depicts the program structure of the module. The Course Requirement Subroutine and the Instructor Requirement Subroutine are called by the Feasibility Study Control Subroutine to calculate course and instructor requirements. For the sake of completeness, Figures II-4 and II-5 originally displayed in TAEG Report 52 are reproduced to show the logic used in course and instructor requirement calculations. The Feasibility Display Subroutine shows on the screen the calculated results and relevant information. With the display, the user is asked to determine if the available resources (instructors, trainers, and equipment) are sufficient to offer the calculated convenings. If not sufficient, the user may look into other possibilities by reducing such quantities as planned inputs, class sizes, etc. Based on these new inputs, the module will recalculate course and instructor requirements and display them. The process is repeated until a feasible schedule has been achieved.

STRUCTURE OF THE WORK SHEET PRINT MODULE

The work sheet print schedule is used to print out schedule work sheets containing such information as planned inputs, class size and length, convening frequency, concurrent classes, required and available instructors, support, and trainers. The format of

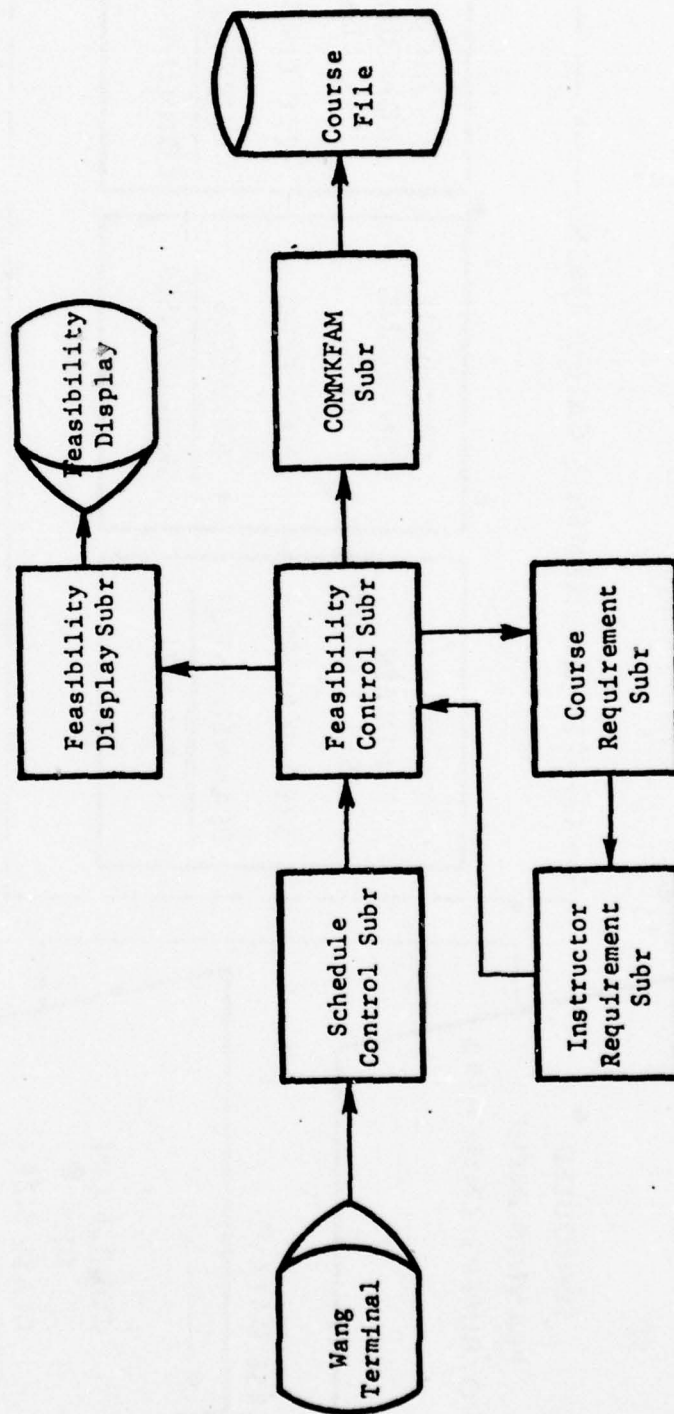


Figure II-3. Schematic of the Feasibility Study Module

LOGIC FOR COURSE REQUIREMENTS

INPUTS (FOR A PARTICULAR COURSE CIN)

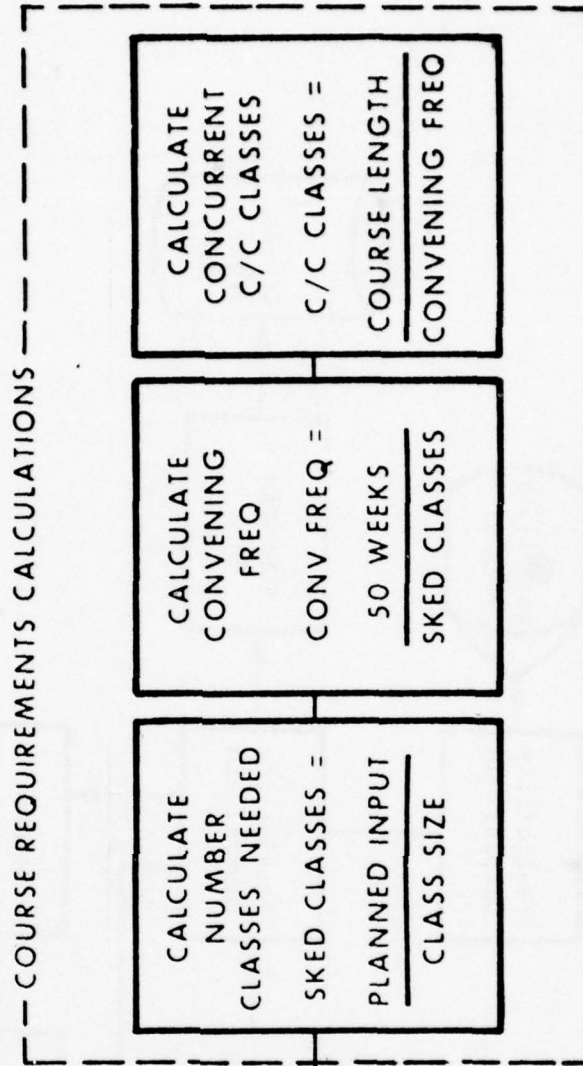
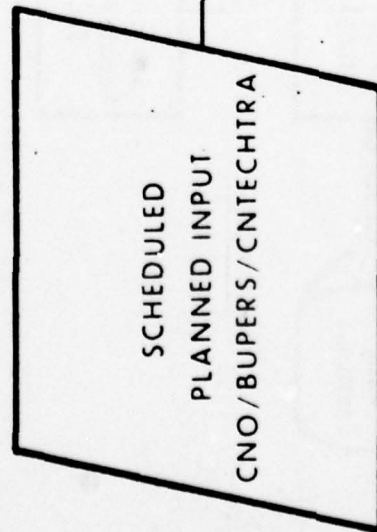


Figure II-4. Logic for Course Requirements

LOGIC FOR INSTRUCTOR REQUIREMENTS

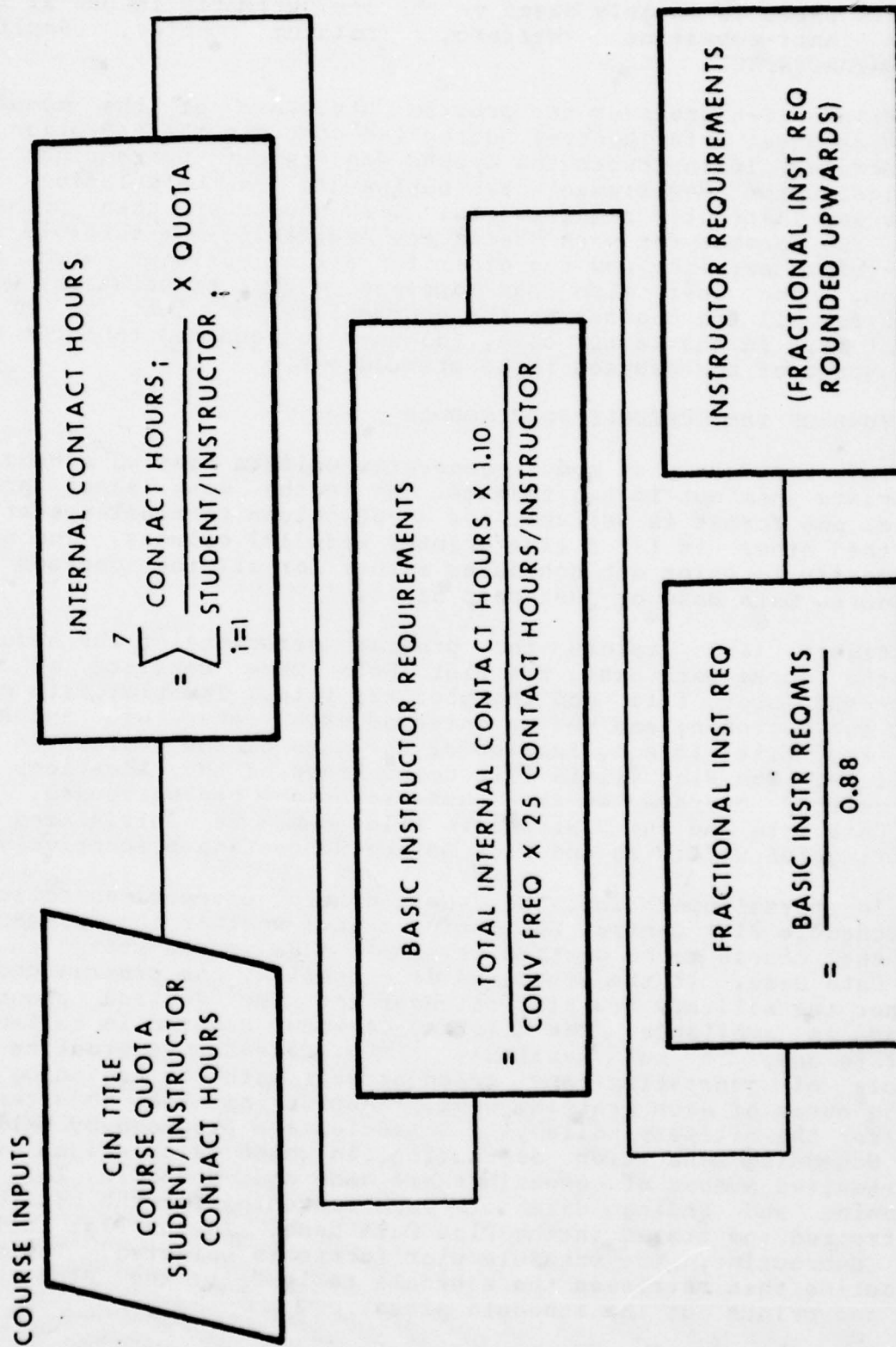


Figure II-5. Logic for Instructor Requirements

the work sheet is largely based on the one currently in use at the Fleet Anti-submarine Warfare Training Center, Pacific (FLEASWTRACENPAC).

Figure II-6 provides the program structure of the module. The Work Sheet Print Control Subroutine controls the operation of this module. It instructs the Course Requirement Subroutine and the Instructor Requirement Subroutine to do calculations for course and instructor requirements. Work sheets are then printed out. Two formats for work sheets are available--one suitable for an 80-column terminal and the other for a line printer with 132 columns. The user also has options either to print out work sheets for all the courses in the course file or just to print part of it. In the latter case, the user is required to input the CDP numbers of the courses to be printed out.

STRUCTURE OF THE SCHEDULE PLOT MODULE

The schedule plot module generates uniform loading schedules and prints them out in two formats. As in the work sheet print module, one format is suitable for an 80-column terminal operation and the other is for a line printer with 132 columns. The user can specify to print out schedules either for all the courses in the Course Data Base or just part of it.

Figure II-7 depicts the program structure of the module. Like the Course Data Base, the Plot Data Base consists of two files--the plot file and the plot key file. The plot file contains the beginning and ending dates of each convening, and the plot key file is a collection of the keys of the courses in the plot file. The Plot Delete File keeps track of the locations of the deleted courses so that those locations can be reused. The Plot Data Base and the Plot Delete File must be initialized by executing ISS Utilities and Plot Delete Subroutine respectively.

In normal operation, the module control proceeds as follows. The Schedule Plot Control Subroutine checks whether the schedules for the course to be plotted have been made up and stored in the Plot Data Base. If the schedules do not exist, the program checks whether the military training calendar for the desired planning period is available. The Military Calendar Program is called to generate one, if not available. The Calendar Subroutine is capable of generating any calendar year with the beginning and ending dates of each training week, considering the adjustments made for the military holidays. Schedules are produced by calling the Schedule Generation Subroutine, in which calculations as to the required number of convenings are made and a table for the beginning and ending date of each convening for the course is constructed and stored in the Plot Data Base. In the Plot Selection Subroutine, the schedule plot format is selected. The Plot subroutine then retrieves the schedule table from the Plot Data Base and prints out the schedule plots.

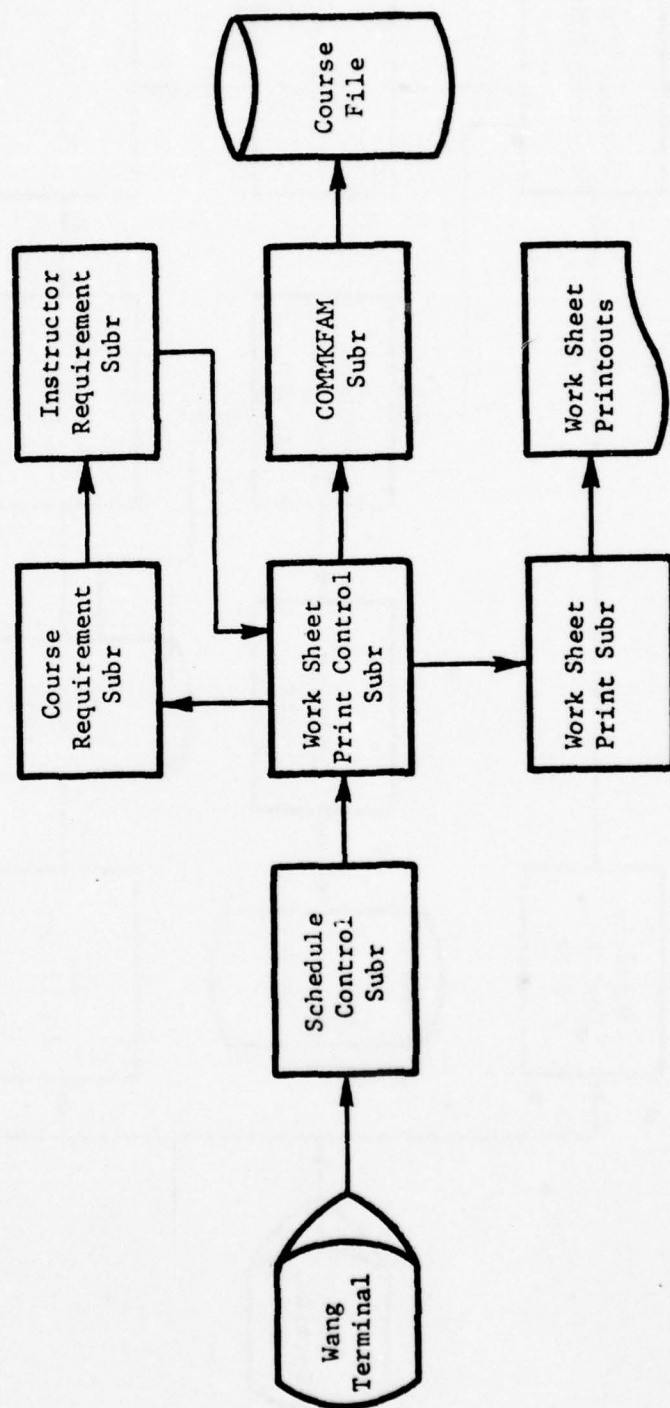


Figure II-6. Schematic of the Work Sheet Print Module

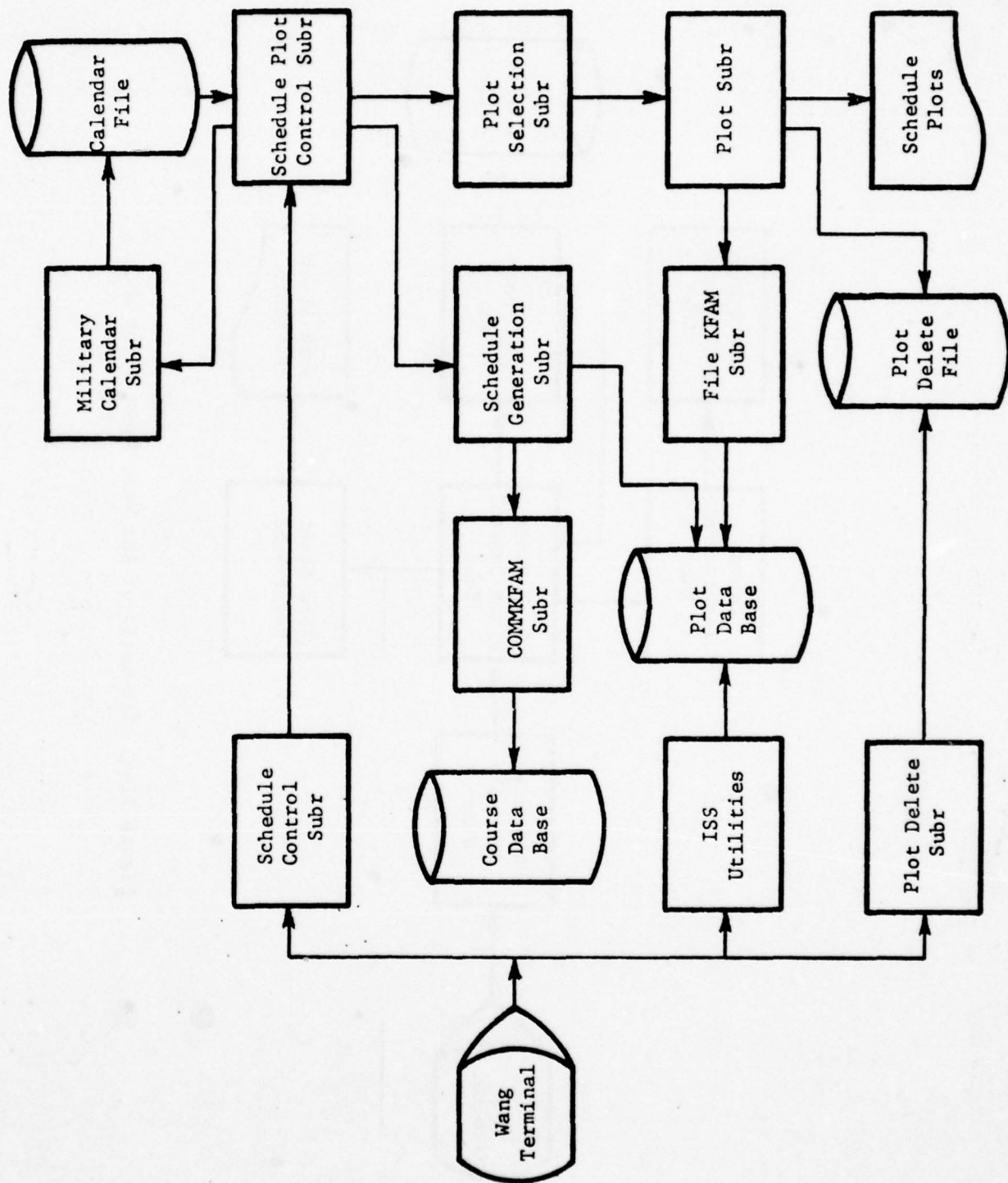


Figure II-7. Schematic of the Schedule Plot Module

SECTION III

SUMMARY

This report documents part one of Phase I of the effort to relieve NAVEDTRACOM personnel of the labor intensive course scheduling procedure. This initial effort was undertaken with the intent of using the Fleet ASW Training Center existing computer resources to develop feasible schedules quickly and easily.

The results of Phase I indicate that:

- a. Automation of the present scheduling procedures used at the FLEASWTRACEN is possible.
- b. The ACSS as developed will run on limited, low cost hardware, i.e., a Wang 2200 system with dual floppy diskette drive and 16-K bytes of memory.
- c. The system designed is highly interactive and user oriented.

Part two of Phase I is directed toward evaluating and field testing the ACSS at various activities with the goal of improving the system design and modifying formats as necessary.

During field testing of the ACSS, an attempt will be made to identify the attributes of an optimal course schedule. This information will then be used in the second phase of the study. This phase will begin with building a mathematical model for the course scheduling problem. It will be followed by the development of an algorithm for solving the model, if existing solution techniques are inappropriate. A computerized data base and an interactive, conversational program will also be developed. The final output of this phase will be the development of an optimal course scheduling methodology.

APPENDIX A
USER'S GUIDE

USER'S GUIDE

The purpose of this appendix is to provide the user with a guide on how to implement and modify the ACSS if the school operates differently from some of the assumptions made in the system. The user is assumed to have some familiarity with the operation of a Wang 2200 system with a dual floppy diskette drive. The user's guide is divided into two parts, the implementation guide and the operating guide. The former is provided to help the user in system verification, initialization, and modification; and the latter addresses itself to the step-by-step use of the system.

IMPLEMENTATION GUIDE

As mentioned earlier, the ACSS is packaged in two floppy diskettes. When the system is used the first time, it is recommended that the contents of the diskettes be verified as shown in Figures A-1 and A-2 by entering the LIST DC command.

INITIALIZATION

The system has been initialized for a capacity of 600 courses in the Data Base. However, sometimes files contain erroneous information that cannot be eliminated by any means. The user is recommended to re-initialize them.

CORSF101 and CORSK101 are the course file and the key file, whereas CORS.DEL is a data file keeping the CDP number of the course deleted in the course key file. Thus, if any of the three files is in error, all of them must be reinitialized. CORS.DEL is re-initialized by loading and executing DEL.CORS. To re-initialize CORSF101 and CORSK101, the user has to mount the ISS Initialization Utilities diskette, load the START Program, and answer the questions displayed on the screen exactly the same as in Figure A-3. In the process, the user is asked to verify the specifications of the course file. If incorrect, enter "N" at the question: "ARE FILE SPECIFICATIONS OK?" The system is ready to respecify the course file. At the end, a correct hard copy of the file specifications is printed out, as shown in Figure A-4.

PLOTF101 and PLOTK101 are the plot data file and the plot key file. PLOT.DEL contains the CDP number of the courses deleted in the schedule key file. Like the course file, these three files must also be re-initialized when any one of them contains errors. Load and run DEL.PLOT to re-initialize PLOT.DEL. Follow the same procedure as in the re-initialization of the course file and answer the questions exactly the same as in Figure A-5. The verification of the plot file specifications is similar to that of the course file, and correct file specifications are as shown in Figure A-6.

310 CATALOG 05/01/78
 INDEX SECTORS = 5
 END CAT. AREA = 1023
 CURRENT END = 319

ITEM	NAME	TYPE	START	END	USED	FREE
1	COMMKFAM	P	193	222	23	2
2	FILEKFAM	P	72	90	17	2
3	DEL.PLOT	P	192	197	4	2
4	INIT.TAP	P	223	260	36	2
5	PLOT.001	P	193	191	7	2
6	PLOT.PLT	P	91	114	22	2
7	PLOT.SEL	P	60	71	10	2
8	SCH.BASE	P	136	170	33	2
9	SCH.FEAS	P	294	318	23	2
10	SCH.PLOT	P	261	293	31	2
11	SCH.SCH	P	171	192	10	2
12	SCH.WSHT	P	20	59	38	2
13	SCHEDULE	P	5	19	13	2
14	TAPE.UTL	P	115	135	19	2

Figure A-1. Contents of the Program File Diskette

310 CATALOG 05/01/73
 INDEX SECTORS = 1
 END CAT. AREA = 1023
 CURRENT END = 992

ITEM	NAME	TYPE	START	END	USED	FREE
1	PLOT.DEL	D	935	992	6	2
2	PLOTF101	D	375	375	602	0
3	PLOTK101	D	339	374	36	0
4	CORSDEL	D	977	934	6	2
5	CORS101	D	37	338	302	0
6	CORS101	D	1	35	35	0

Figure A-2. Contents of the Data File Diskette

ENTER USER FILE NAME	CORSF101
DEVICE ADDRESS FOR USER FILE	B10
IS USER FILE CATALOGED	N
KEY FILE NUMBER	1
DEVICE ADDRESS FOR KEY FILE	B10
IS KEY FILE CATALOGED	N
RECORD TYPE	A
LOGICAL RECORD LENGTH	126
BLOCKING FACTOR	2
KEY LENGTH	4
STARTING POSITION OF KEY	3
ESTIMATED NUMBER OF RECORDS	600
ARE FILE SPECIFICATIONS OK	Y
HARD COPY PRINTOUT	Y

Figure A-3. Re-initialization of the Course file

DATA FILE NAME	SCHEF101	KEY FILE NAME	SCHEK101
DEVICE ADDRESS	310	DEVICE ADDRESS	301
RECORD TYPE	A	KEY FILE NUMBER	1
RECORD LENGTH	126	KEY LENGTH	4
BLOCKING FACTOR	2	KEY POSITION	3
NO. OF SECTORS	302	NO. OF SECTORS	36

Figure A-4. Specifications of the Course file.

ENTER USER FILE NAME	PLOTf101
DEVICE ADDRESS FOR USER FILE	B10
IS USER FILE CATALOGED	N
KEY FILE NUMBER	1
DEVICE ADDRESS FOR KEY FILE	B10
IS KEY FILE CATALOGED	N
RECORD TYPE	N
KEY LENGTH	4
STARTING POSITION OF KEY	1
ESTIMATED NUMBER OF RECORDS	600
ARE FILE SPECIFICATIONS OK	Y
HARD COPY PRINTOUT	Y

Figure A-5. Re-initialization of the Plot File.

DATA FILE NAME	PLOTf101	KEY FILE NAME	PLOTk101
DEVICE ADDRESS	310	DEVICE ADDRESS	310
RECORD TYPE	N	KEY FILL NUMBER	1
RECORD LENGTH	1	KEY LENGTH	4
(SECTORS)		KEY POSITION	0
NO. OF SECTORS	602	NO. OF SECTORS	36

Figure A-6. Specifications of the plot File.

OPERATING GUIDE

The information provided below is designed to familiarize the user with the operation of the ACSS. In the guide, a step-by-step operating procedure is outlined, followed by examples for illustrating the operating procedures.

OPERATING PROCEDURE

The procedure to be outlined below involves such steps as setting up, running, creating, and maintaining a data base, assessing feasibility, printing work sheets, and printing and editing course schedules. While the first two steps must be performed in that order, the other steps can be executed randomly; however they are numbered for convenience of discussion.

STEP 1: SET UP ACSS

Mount the Schedule Management Program diskette into disk drive 310 and the Schedule Management Data Base diskette into disk drive B10. (Note: These disk assignments can be modified by changing the SELECT statements in the SCHEDULE program.)

STEP 2: RUN ACSS

Load the SCHEDULE program from the program diskette into the CPU (Central Processing Unit) memory. After a few seconds of pause, the NAVAL TRAINING MANAGEMENT System Control Menu is displayed on the screen. The menu shown in Figure A-7 is to be referred to as the System Control Menu in the discussion. Upon entering the desired function code as called for on the menu, the desired module is loaded and ready for processing.

In the ACSS, three special function routines are provided to facilitate system interruptions. At any point in processing, depressing special function 0 (S.F.O.) key will terminate the entire processing. If further use of the ACSS is desired, clear the CPU memory and reload the SCHEDULE program. To terminate current operation and return to the System Control Menu, depress S.F.1. Similarly, to terminate current processing and return control to the same module, touch S.F.2.

STEP 3: MAINTAIN COURSE DATA BASE

To create and maintain the course data base, the function code "1" is entered on the System Control Menu. After the Data Base Maintenance Menu shown in Figure A-8 is displayed, the user can look up the data of a specific course, make changes to course data, add new courses, delete undesired courses, and examine the course file in the data base. To terminate data base maintenance, the function code "6" is keyed in. The control returns to step 2, and the System Control Menu is displayed. Figure A-9 shows the course data input formats.

NAVAL TRAINING SCHEDULE MANAGEMENT

THIS SYSTEM WILL PERFORM THE FOLLOWING

- 1 - MAINTAIN DATA BASE
- 2 - SCHEDULE FEASIBILITY STUDY
- 3 - PRINT SCHEDULE WORK SHEET
- 4 - PLOT SCHEDULE

ENTER CODE FOR DESIRED FUNCTION?

Figure A-7. System Control Menu.

DATA BASE MAINTENANCE MENU

CODE	FUNCTION
1	- DISPLAY COURSE DATA
2	- CHANGE COURSES
3	- ADD COURSES
4	- DELETE COURSES
5	- DISPLAY COURSE FILE
6	- END OF PROCESSING

ENTER FUNCTION CODE?

Figure A-8. Data Base Maintenance Menu.

STEP 4: PERFORM FEASIBILITY STUDY

To perform the feasibility study of a course, the function code "2" is entered on the System Control Menu. When the system is ready, the user is asked to input the CDP number of the course for which feasibility is to be assessed. Along with other pertinent data, the results of the instructor and trainer requirement calculations as exemplified in Figure A-10 are displayed. By comparing the available resources with the required ones, the user may evaluate the feasibility and make a number of trials to determine a feasible schedule, if the calculated schedule was infeasible. At the end of each feasibility evaluation, the user may choose to use other modules by depressing S.F.1.

STEP 5: PRINT SCHEDULE WORK SHEETS

To print schedule work sheets, the function code "3" is entered on the System Control Menu. When the system is ready, the user is asked to: (1) compile the courses to be printed out on the work sheets, (2) specify where to print--on the screen or on the line printer. A sample work sheet is shown in Figure A-11. After finishing with the module, the control returns to Step 2 and the System Control Menu again is displayed.

STEP 6: PRINT COURSE SCHEDULES

To print course schedules for documentation and dissemination the function code "4" is entered on the System Control Menu. The user is instructed to key in the CDP numbers of the courses for which schedules are to be generated. Figure A-12 is a sample schedule of a course. On the schedule, the course identification and other pertinent data are printed on the top of the schedule, followed by the beginning and ending dates of the 50 training weeks for the planning year. If one convening is scheduled in the week, it is so indicated under session 1 or 2 in that week. Two convenings may be scheduled in the same week. At the end of the processing, the control returns to step 2 and the System Control Menu is displayed.

Course File

Field Name	Data Type
CDP	A4*
CIN	A8
Course Short Title	A18
Planned Input	I5**
Class Size	I2
Course Length in Weeks	I2
Contact Ratios	I3
Contact Hours	I4
Manpower Allowances	I3
Other Instructors Available	I3
No. of Trainers Available	I2
Support Required	I2
Support MPA	I2

*A4 means a field of 4 alphanumeric characters.

**I5 means a field of 5 integer numeric digits.

Figure A-9. Course Data Input Formats.

FEASIBILITY STUDY FOR A0000018										
CDP		TITLE		WKS	DAYS	INPUT		AOB		
---		-----		---	---	-----		-----		
000A		DATA TESTING		11	75	150		30.82		
CLASS	SKED	CONV	C/C	INSTRUCTOR			SUPPORT		TRAINERS	
SIZE	CLASS	FREQ	CLASS	REQ	MPA	OTHER	REQ	MPA	REQ	AVL
----	----	----	----	----	----	----	----	----	----	----
10	15	3.33	3.30	1	01	003	02	02	2	01

WANT TO ALTER INPUT OR CLASS SIZE (Y OR N)?

Figure A-10. Feasibility Study Sample Output.

COURSE SCHEDULE WORK SHEET

CRS	LEN				CLASS	SKED	CONV	C/C		INSTRUCTOR		SUPPORT		TRANERS
WKS	DAYS	INPUT	A0B	SIZE	CLASS	FREQ	CLASS	REQ	MPA	OTHER	REQ	MPA	REQ	AVL
CDP 0310	CIN	A 4F0010			CRS TITLE	MSL INDOC/RESO								
1	5	100	1.36	10	10	5.00	0.20	1	2	003	02	02		1:02
CDP 0327	CIN	A 000111			CRS TITLE	SWO ADV COMMAN								
2	12	100	3.28	10	10	5.00	0.40	3	2	003	02	02		1:02
CDP 0328	CIN	A 000112			CRS TITLE	SWO ADV EXEC								
2	12	100	3.28	10	10	5.00	0.40	3	2	003	02	02		1:02
CDP 0335	CIN	A5510055			CRS TITLE	INVENTORY MGMT								
1	5	100	1.36	10	10	5.00	0.20	1	2	003	02	02		1:02
CDP 0336	CIN	A5510056			CRS TITLE	FINANCIAL MGMT								
1	5	100	1.36	10	10	5.00	0.20	1	2	003	02	02		1:02
CDP 0348	CIN	A 2A0013			CRS TITLE	HELO MCM FAM								
1	5	100	1.36	10	10	5.00	0.20	1	2	003	02	02		1:02
CDP 1155	CIN	A1500017			CRS TITLE	TERMINAL MAINT								
3	19	150	7.80	10	15	3.33	0.90	21	2	003	02	02		1:02
CDP 1211	CIN	K 2G0038			CRS TITLE	NTDS DATA UTIL								
3	19	100	5.20	10	10	5.00	0.60	4	2	003	02	02		1:02
CDP 1217	CIN	J6440911			CRS TITLE	SURF NUWPNS PR								
3	19	100	5.20	10	10	5.00	0.60	1	2	003	02	02		1:02
CDP 1370	CIN	A4310012			CRS TITLE	EOO BASIC SURF								
3	19	100	5.20	10	10	5.00	0.60	9	2	003	02	02		1:02

Figure A-11. Course Schedule Work Sheet Sample Output


```

CDP / CIN ..... > 0115 / A7802037
CRS TITLE ..... > DC-P250 PMP REP
CRS LEN (WKS/DAYS) . > 8 / 54
INPUTS / CLASS SIZ . > 100 / 10

```

WEEK	BEG	END	DAY SESSION 1 BEG - END	DAY SESSION 2 BEG - END
1	OCT 2	OCT 6	OCT 2 - NOV 24	
2	OCT 10	OCT 13		
3	OCT 16	OCT 20		
4	OCT 23	OCT 27		
5	OCT 30	NOV 3		
6	NOV 6	NOV 9	NOV 6 - JAN 5	
7	NOV 13	NOV 17		
8	NOV 20	NOV 24		
9	NOV 27	DEC 1		
10	DEC 4	DEC 8		
11	DEC 11	DEC 15	DEC 11 - FEB 9	
12	DEC 18	DEC 22		
13	JAN 2	JAN 5		
14	JAN 8	JAN 12		
15	JAN 15	JAN 19		
16	JAN 22	JAN 26	JAN 22 - MAR 16	
17	JAN 29	FEB 2		
18	FEB 5	FEB 9		
19	FEB 12	FEB 16		
20	FEB 20	FEB 23		
21	FEB 26	MAR 2	FEB 26 - APR 20	
22	MAR 5	MAR 9		
23	MAR 12	MAR 16		
24	MAR 19	MAR 23		
25	MAR 26	MAR 30		
26	APR 2	APR 6	APR 2 - MAY 25	
27	APR 9	APR 13		
28	APR 16	APR 20		
29	APR 23	APR 27		
30	APR 30	MAY 4		
31	MAY 7	MAY 11	MAY 7 - JUN 29	
32	MAY 14	MAY 18		
33	MAY 21	MAY 25		
34	MAY 29	JUN 1		
35	JUN 4	JUN 8		
36	JUN 11	JUN 15	JUN 11 - AUG 3	
37	JUN 18	JUN 22		
38	JUN 25	JUN 29		
39	JUL 2	JUL 6		
40	JUL 9	JUL 13		
41	JUL 16	JUL 20	JUL 16 - SEP 7	
42	JUL 23	JUL 27		
43	JUL 30	AUG 3		
44	AUG 6	AUG 10		
45	AUG 13	AUG 17		
46	AUG 20	AUG 24	AUG 20 - OCT 12	
47	AUG 27	AUG 31		
48	SEP 4	SEP 7		
49	SEP 10	SEP 14		
50	SEP 17	SEP 21		

Figure A-12. Course Schedule Plot Sample Output

In some parts of the system, fatal error messages are supplied and instructions are provided to restart execution. In the plot module, for instance, when execution is in error, the following may be displayed on the screen:

```
*****
* EXECUTE THE FOLLOWING COMMANDS TO RESTART:      *
*      :CLEAR V (RETURN (EXEC))                  *
*      :CLEAR P (RETURN (EXEC))                  *
*      :LOAD DCT #0, SCHEDULE (RETURN (EXEC))    *
*      :RUN (RETURN (EXEC))                      *
* ENTER COMMANDS NOW....                          *
*****
```

Entering commands as instructed in the above box will lead to loading of the SCHEDULE Program and returning to the system main control.

AN ILLUSTRATION

In the preceding discussion, the operation of the ACSS was outlined. The text that follows provides detailed examples on how the system is used. Note that the sequence of the presentation assumes that the Course Data Base has not been created. In actual use, the user can execute any logical steps and need not follow the same sequence as presented. For ease of reading, system inputs and responses are enclosed in boxes.

SET UP AND RUN ACSS

Mount the Schedule Management Program diskette into disk drive 310 and the Schedule Management Data Base diskette into disk drive B10. Enter the following.

```
*****
* :LOAD DCF "SCHEDULE" (HIT RETURN KEY)          *
* :RUN (HIT RETURN Key)                          *
*****
```

The system will respond with

```
*****
* TRAINING SCHEDULE PROGRAM                      *
* SYSTEM NOW LOADING COMMON KFAM ROUTINE        *
* PROGRAM BEING LOADED > COMMKFAM               *
*****
```

There will be a slight pause while the program is being loaded. When completed, the following will be displayed on the screen:

```
*****
*   NAVAL TRAINING SCHEDULE MANAGEMENT   *
*                                         *
*   THIS SYSTEM WILL PERFORM THE FOLLOWING *
*   1 - MAINTAIN DATA BASE               *
*   2 - SCHEDULE FEASIBILITY STUDY        *
*   3 - PRINT SCHEDULE WORK SHEET         *
*   4 - PLOT SCHEDULE                     *
*   ENTER CODE FOR DESIRED FUNCTION?      *
*****
```

Suppose the course data file is empty. To create a course file, we need to enter "1" and hit RETURN key.

MAINTAIN COURSE DATA BASE

After hitting the RETURN key, the following will be displayed:

```
*****
*   SYSTEM NOW LOADING DATA BASE MAINT. ROUTINE *
*   PROGRAM BEING LOADED > SCH.BASE             *
*****
```

After a few seconds of pause, the screen will display:

```
*****
*   DATA BASE MAINTENANCE MENU             *
*   CODE      FUNCTION                     *
*   1 - DISPLAY COURSE DATA               *
*   2 - CHANGE COURSES                     *
*   3 - ADD COURSES                        *
*   4 - DELETE COURSES                     *
*   5 - DISPLAY COURSE FILE               *
*   6 - END OF PROCESSING                  *
*   ENTER FUNCTION CODE?                   *
*****
```

ADD COURSES

Since we would like to create a course file, we enter "3" and hit the RETURN key.

```
*****
*   ENTER CDP TO BE ADDED? *
*****
```

After inputting a desired CDP number, the inputs for a course titled "BU/HEAVY CONST" may look like


```

*****
* ENTER CDP? 0111 *
* ENTER CIN? A7100018 *
* ENTER COURSE SHORT TITLE? BU/HEAVY CONST *
* ENTER PLANNED INPUT? 100 *
* ENTER CLASS SIZE? 10 *
* ENTER COURSE LENGTH IN WEEKS? 1 *
* ENTER NO. OF PHASES? 1 *
* ENTER LECTURE RATIO AND HOURS FOR FIRST PHASE (R,H)? 25,10 *
* ENTER MAN POWER ALLOWANCES? 2 *
* ENTER OTHER INSTRUCTORS AVAILABLE? 2 *
* ENTER NO OF TRAINERS AVAILABLE? 2 *
* ENTER NO OF SUPPORT REQUIRED AND MPA? 2,2 *
*****

```

(Note: Figure A-9 shows the course data input formats.)

The system will then ask if any more courses are to be added with the following display:

```

*****
* WANT TO ADD ANOTHER COURSE? *
*****

```

We may add courses by entering "Y" and the system will pose the same set of questions for data. Suppose we have entered a number of courses and suspect keying in some erroneous data in a course, for instance, "BU/HEAVY CONST." When we enter "N," the system will display the Data Base Maintenance Menu again.

LOOK UP COURSES

After we enter "1" and subsequently the CDP key "0111" to look up the course data for "BU/HEAVY CONST," the screen will display:

```

*****
* ENTER CDP TO BE LOOKED UP? 0111 *
* DATA FOR A7100018 TITLE > BU/HEAVY CONST *
* PLANNED INPUT 100 CLASS SIZE 10 LENGTH(WKS) 1 *
* MAN POWER ALLOWANCES 002 INSTRUCTORS 003 TRAINERS 02 *
* SUPPORT 02 MPA 02 *
*CONTACT RATIO/HOURS (025/0010),(000/0000),(000/0000),(000,0000)*
* (000/0000),(000/0000),(000/0000) *
*****

```

We may check out as many courses as we desire.

CHANGE COURSE DATA

Suppose the planned input for "BU/HEAVY CONST" should be 150 instead of 100. To make such changes, we have to enter function code "2." After entering the code and the CDP key, the system will respond:

```

*****
*   ENTER CDP TO BE LOOKED UP? 0111
*   DATA FOR A7100018          TITLE > BU/HEAVY CONST
*   PLANNED INPUT 100          CLASS SIZE 10          LENGTH(WKS) 1
*   MAN POWER ALLOWANCES 002    INSTRUCTORS 003        TRAINERS 02
*   SUPPORT 02                  MPA 02
*   CONTACT RATIO/HOURS (025/0010),(000/0000),(000/0000),(000/0000)
*   (000/0000),(000/0000),(000/0000)
*
* KEY* DATA ITEM      KEY* DATA ITEM      KEY* DATA ITEM
* 1 * CDP              2 * CIN              3 * CRS TITLE
* 4 * PLANNED INPUT    5 * CLASS SIZE      6 * CRS LEN (WKS)
* 7 * CNTCT RAT/HRS    8 * TRAINERS        9 * MAN POWER
* 10 * OTHER INSTRS    11 * SUPPORT        12 * NO CHANGES
* ENTER CHANGE KEY?
*****

```

Clearly, we should key in "4." The system will ask us to enter the new planned input. Other changes may be made to the course or another course.

EXAMINE COURSES IN COURSE FILE

Suppose we want to examine what the course file contains. This is done by entering the function code "5." The system will display 14 courses at a time and ask to continue. The screen may look like the following:

```

*****
*   ENTER FUNCTION CODE? 5
*   COURSES IN COURSE FILE (CDP/TITLE)
*   0111 / A7100018 / BU/HEAVY CONST
*   0115 / A7802037 / DC/P250 PMP REP
*   0125 / J7800400 / DIV DC PU
*   0152 / A 4E0045 / EOD RES ING
*   0180 / A 2F0020 / SWS WEP OFF POS
*   0201 / A 4N0017 / SALV INDOC
*   0218 / A 4N0013 / ENG DIV OFF
*   0273 / A 4N0018 / TREAT DIV CASUAL
*   0286 / A5510059 / TECH PUB MATL ID
*   0307 / A 2E0010 / TAR PCO/PXO
*   0310 / A 4F0010 / MGL INDOC/RESOFF
*   0327 / A 000111 / SWO ADV COMMAND
*   0328 / A 000112 / SWO ADV EXEC
*   0335 / A5510055 / INVENTORY MGMT
*   WANT TO CONTINUE DISPLAYING? Y
*   033C / A5510050 / FINANCIAL MGMT
*   0348 / A 2A0013 / HELO MCM FAM
*****

```

DELETE COURSES

If we want to delete courses, we can do it by entering the function code "4" and then entering their CDP keys one at a time.

TERMINATE DATA BASE MAINTENANCE

When we have completed Data Base creation and maintenance, we can proceed by entering the function code "6." After a few seconds of pause, the system will respond with the System Control Menu.

PERFORM FEASIBILITY STUDY

On entering the code "2," the system will be ready to help perform feasibility study. The following display is an example of a course entitled "DATA TESTING":

```
*****
*                                     FEASIBILITY STUDY FOR A0000018          *
*      CDP      TITLE      WKS  DAYS      INPUT      AOB      *
*      ---      - - - - -      - - -      - - -      - - -      - - -      *
*      000A      DATA TESTING      11      75      150      30.82      *
*
* CLASS  SKED  CONV  C/C      INSTRUCTOR      SUPPORT      TRAINERS      *
* SIZE  CLASS  FREQ  CLASS  REQ  MPA  OTHER  REQ  MPA  REQ  AVL      *
* ---  ---  ---  ---  ---  ---  ---  ---  ---  ---  ---      *
*   10    15  3.33  3.30    1  01    003    02    02    2    01      *
*
* WANT TO ALTER INPUT OR CLASS SIZE (Y OR N)?
*****
```

After examining the display, for instance, we have found that one trainer is not enough to offer concurrent classes of 3.30. Thus, a decrease in planned input or an increase in class size or a combination of them need to be made. The system allows us to make a number of trials to determine a feasible schedule.

PRINT SCHEDULE WORK SHEETS

By entering the code "3" on the System Control Menu, we can display schedule work sheets on the screen or print them on the line printer for files. A sample printout is shown in A-13.

PRINT SCHEDULE PLOTS

To print course plots, the code "4" on the System Control Menu is keyed in. The system will prompt us with what courses to be plotted. We can specify either all courses or part of them to be plotted. A sample schedule plot is shown in A-14.

COURSE SCHEDULE WORK SHEET

[illegible]

Figure A-13. Course Schedule Work Sheet

CDP / CIN > 0111 / A7100018
 CRS TITLE > BU/HEAVY CONST
 CRS LEN (WKS/DAYS) . > 1 / 5
 INPUTS / CLASS SIZ . > 60 / 1

WEEK	BEG	-	END	DAY SESSION 1	BEG	-	END	DAY SESSION 2	BEG	-	END
1	OCT 2	-	OCT 6	OCT 2	-	OCT 6		OCT 2	-	OCT 6	
2	OCT 10	-	OCT 13	OCT 10	-	OCT 13					
3	OCT 16	-	OCT 20	OCT 16	-	OCT 20					
4	OCT 23	-	OCT 27	OCT 23	-	OCT 27					
5	OCT 30	-	NOV 3	OCT 30	-	NOV 3					
6	NOV 6	-	NOV 9	NOV 6	-	NOV 9		NOV 6	-	NOV 9	
7	NOV 13	-	NOV 17	NOV 13	-	NOV 17					
8	NOV 20	-	NOV 24	NOV 20	-	NOV 24					
9	NOV 27	-	DEC 1	NOV 27	-	DEC 1					
10	DEC 4	-	DEC 8	DEC 4	-	DEC 8					
11	DEC 11	-	DEC 15	DEC 11	-	DEC 15		DEC 11	-	DEC 15	
12	DEC 18	-	DEC 22	DEC 18	-	DEC 22					
13	JAN 2	-	JAN 5	JAN 2	-	JAN 5					
14	JAN 8	-	JAN 12	JAN 8	-	JAN 12					
15	JAN 15	-	JAN 19	JAN 15	-	JAN 19					
16	JAN 22	-	JAN 26	JAN 22	-	JAN 26		JAN 22	-	JAN 26	
17	JAN 29	-	FEB 2	JAN 29	-	FEB 2					
18	FEB 5	-	FEB 9	FEB 5	-	FEB 9					
19	FEB 12	-	FEB 16	FEB 12	-	FEB 16					
20	FEB 20	-	FEB 23	FEB 20	-	FEB 23					
21	FEB 26	-	MAR 2	FEB 26	-	MAR 2		FEB 26	-	MAR 2	
22	MAR 5	-	MAR 9	MAR 5	-	MAR 9					
23	MAR 12	-	MAR 16	MAR 12	-	MAR 16					
24	MAR 19	-	MAR 23	MAR 19	-	MAR 23					
25	MAR 26	-	MAR 30	MAR 26	-	MAR 30					
26	APR 2	-	APR 6	APR 2	-	APR 6		APR 2	-	APR 6	
27	APR 9	-	APR 13	APR 9	-	APR 13					
28	APR 16	-	APR 20	APR 16	-	APR 20					
29	APR 23	-	APR 27	APR 23	-	APR 27					
30	APR 30	-	MAY 4	APR 30	-	MAY 4					
31	MAY 7	-	MAY 11	MAY 7	-	MAY 11		MAY 7	-	MAY 11	
32	MAY 14	-	MAY 18	MAY 14	-	MAY 18					
33	MAY 21	-	MAY 25	MAY 21	-	MAY 25					
34	MAY 29	-	JUN 1	MAY 29	-	JUN 1					
35	JUN 4	-	JUN 8	JUN 4	-	JUN 8					
36	JUN 11	-	JUN 15	JUN 11	-	JUN 15		JUN 11	-	JUN 15	
37	JUN 18	-	JUN 22	JUN 18	-	JUN 22					
38	JUN 25	-	JUN 29	JUN 25	-	JUN 29					
39	JUL 2	-	JUL 6	JUL 2	-	JUL 6					
40	JUL 9	-	JUL 13	JUL 9	-	JUL 13					
41	JUL 16	-	JUL 20	JUL 16	-	JUL 20		JUL 16	-	JUL 20	
42	JUL 23	-	JUL 27	JUL 23	-	JUL 27					
43	JUL 30	-	AUG 3	JUL 30	-	AUG 3					
44	AUG 6	-	AUG 10	AUG 6	-	AUG 10					
45	AUG 13	-	AUG 17	AUG 13	-	AUG 17					
46	AUG 20	-	AUG 24	AUG 20	-	AUG 24		AUG 20	-	AUG 24	
47	AUG 27	-	AUG 31	AUG 27	-	AUG 31					
48	SEP 4	-	SEP 7	SEP 4	-	SEP 7					
49	SEP 10	-	SEP 14	SEP 10	-	SEP 14					
50	SEP 17	-	SEP 21	SEP 17	-	SEP 21					

Figure A-14. Course Schedule Plot

APPENDIX B
ACSS PROGRAM LISTING

SCHEDULE System ---- (SCHEDULE)

```

100 Z## PROGRAM > SCHEDULE MANAGEMENT (SCHEDULE) VERSION 1.0 780530
110 REM
120 REM
130 REM
140 REM *****
150 REM *
160 REM * TRAINING SCHEDULE MANAGEMENT *
170 REM * WRITTEN BY *
180 REM * Phil Cooke *
190 REM * Ben Lin *
210 REM * Chuck Guitard *
220 REM *
230 REM *****
240 REM
250 REM
260 REM
270 REM
280 REM ** This Program will create and maintain a Training Schedule
290 REM ** Management Data Base.
300 REM **
310 REM ** This Program will allow compute and display a course
320 REM ** schedule work sheet and plot schedule chart.
330 REM **
340 REM ** Description of Variables >
350 REM **
360 REM ** C1$(2) 4 D RECORD ID- CDP
370 REM ** N1$(2) 8 D CIN
380 REM ** T$(2) 18 D COURSE TITLE
390 REM ** I1$(2) 5 D PLANNED INPUT
400 REM ** C2$(2) 2 D CLASS SIZE
410 REM ** W$(2) 2 D COURSE LENGTH (WKS)
420 REM ** R$(2,7) 3 D CONTACT RATIOS
430 REM ** H$(2,7) 4 D CONTACT HOURS
440 REM ** M1$(2) 3 D MAN POWER
450 REM ** I9$(2) 3 D INSTRUCTORS
460 REM ** E1$(2) 2 D TRAINERS
470 REM ** S1$(2) 2 D SUPPORT
480 REM ** S2$(2) 2 D MPA
490 REM ** W2 COURSE LENGTH (DAYS)
500 REM ** 0 AOB
510 REM **
520 REM *****
530 REM
540 REM DEFINE KFAM COMMON VARIABLES
550 REM
560 COM V7$8, T0$7, V9, V0$(3)2, T1(3)
570 COM V0$2, V1$8, V2$2, V3$2, V4$2, V6$1
580 COM Q2$2, Q3$2, V5$1, V8$1, T5$30, T7$30
590 COM T0, T9, T2$2, T4$3, T2, V8, T8, T1, T8$1, T2$(8)2, T(8)
600 COM T4, T5, V6, V7, V1
610 COM Q, Q$1, T9$2, T0$(4)60
620 COM V7$(3)8, Q4$2, Q9$2, Q0$(4)60
630 COM T6$1, T3$3, T1$(1)2, T3$(1)33, V9$2, T1$30
640 REM
650 REM DEFINE PROGRAM COMMON VARIABLES
660 REM
670 COM C1$(2)4, T$(2)18, I1$(2)5, C2$(2)2, R$(2,7)3, S8$(2,50)1, N1$(2)8
680 COM H$(2,7)4, M1$(2)3, I9$(2)3, E1$(2)2, S1$(2)2, S2$(2)2, D1$(250)3

```

```

690 COM I1(1),C2(1),W(1),R(1,7),H(1,7),M1(1),I9(1),E1(1),W$(2)2
700 COM S1(1),S2(1),C3(1),C4(1),W2(1),D(1),F1$8,F2$8,F3$8,F4$8
710 COM I2(1),I3(1),C5(1),C6(1),E3(1),M2(1),M1(1),S(1),E$64
720 REM
730 REM          INITIALIZE DISK DEVICE TABLE
740 REM
750          SELECT DISK 310                : REM SEL SYSTEM PROGRAM DISK
760          SELECT #1 B10                  : REM SEL DATA BASE KEY FILE
770          SELECT #2 B10                  : REM SEL DATA BASE USER FILE
780          SELECT #3 B10                  : REM SEL DATA BASE DELETE FILE
790          SELECT #4 B10                  : REM SEL PLOT KEY FILE
800          SELECT #5 B10                  : REM SEL PLOT USER FILE
810          SELECT #6 B10                  : REM SEL PLOT DELETE FILE
820 REM
830 REM          INITIALIZE DATA BASE FILENAMES
840 REM
850          F1$ = "SCHEF101"                : REM DATA BASE FILE NAME
860          F2$ = "PLOT101"                : REM SCHEDULE FILE NAME
862          F3$ = "SCHE.DEL"                : REM DAT BAS DEL FILE NAME
864          F4$ = "PLOT.DEL"                : REM DAT SCHED DEL FILE NAME
870 REM
880 REM          LOAD KFAM COMMON ROUTINES
890 REM
900          GOSUB / 102 ("KFAM COMMON","COMMKFAM","8000")
910 REM
920 REM          DISPLAY PROGRAM CONTROL MENU
930 REM
1000         SELECT PRINT 005(80)            : REM SEL SYSTEM OUTPUT DEVICE
1010         SELECT INPUT 001                : REM SEL SYSTEM INPUT DEVICE
1020 REM
1030 PRINT HEX(030A0A)
1040 PRINT "NAVAL TRAINING SCHEDULE MANAGEMENT"
1050 PRINT " THIS SYSTEM WILL PERFORM THE FOLLOWING"
1060 PRINT "      1. MAINTAIN DATA BASE"
1070 PRINT "      2. SCHEDULE FEASIBILITY STUDY"
1080 PRINT "      3. PRINT SCHEDULE WORK SHEET"
1090 PRINT "      4. PLOT SCHEDULE"
1100 PRINT "      5. EDIT SCHEDULE"
1110 PRINT HEX(0A0A)
1120 GOSUB / 101 (14,1,0,0)
1130 PRINT "ENTER CODE FOR DESIRED FUNCTION ----> ";
1140 INPUT AB
1150 IF AB<=0 THEN 1030: IF AB>=6 THEN 1030
1160 REM
1170 REM      !MAINTANCE !FEASIBILT !WORKSHEET !PLT SCHED !EDIT SCHED!
1180 REM      !-----!-----!-----!-----!-----!
1190 ON AB GOTO      1240      ,      1250      ,      1260      ,      1270      ,      1280
1200 REM      !-----!-----!-----!-----!-----!
1210 REM **
1220 REM **          LOAD OVERLAY....
1230 REM **
1240         GOSUB / 102 ("DATA BASE MAINTENANCE","SCH.BASE","1000")
1250         GOSUB / 102 ("FEASIBILITY STUDY","SCH.FEAS","1000")
1260         GOSUB / 102 ("SCHEDULE WORK SHEET","SCH.WSHT","1000")
1270         GOSUB / 102 ("PLOT SCHEDULE","SCH.PLOT","2000")
1280         GOSUB / 102 ("EDIT SCHEDULE","SCH.EDIT","1000")
1290 REM **
1300 REM **
1310         END
1320 REM **

```

```

1330 REM **
1340 REM ** Subroutine > ***** 1 0 1
1350 REM ** Function >
1360 REM ** Cursor Address and Screen erase
1370 REM ** Variables >
1380 REM ** Q1 > row position ! Q3 > lines to erase
1390 REM ** Q2 > col position ! Q4 > char(s) erase
1400 REM ** Q1$() > cursor vert ! Q2$() > cursor horz
1410 REM ** Q3$() > line ers ! Q4$() > micro cmd bufr
1420 REM ** Q5 > line indx !
1430 REM **
1440 REM *****
1450 DEFFN ' 101 (Q1,Q2,Q3,Q4) : REM ENTRY
1460 DIM Q1$(16),Q2$(64),Q3$(64),Q4$(10) : REM DEFIN BUFFRS
1470 INIT(0A) Q1$() : REM INIT ROW BUFR
1480 INIT(09) Q2$() : REM INIT COL BUFR
1490 INIT(20) Q3$() : REM INIT ERS BUFR
1500 Q1$(1) = HEX(01) : REM INIT TOP OF PAGE
1510 Q2$(1) = HEX(0D) : REM INIT BEG OF LINE
1520 IF Q3 = 0 THEN 1630 : REM NO LINES TO ERASE
1530 IF Q4 <> 0 THEN 1550 : REM CHARS TO ERASE
1540 Q4 = 65 - Q2 : REM ERS REST OF LINE
1550 FOR Q5 = 1 TO Q3 : REM ERASE Q3 LINES
1560 IF Q5 <> 1 THEN 1590 : REM NOT FIRST LINE
1570 $GID ROW/005 (A000,Q4$())Q1$(<1,Q1> : REM POSITION CURSOR
1580 GOTO 1600 : REM BRANCH TO POS COL
1590 $GID ROW/005 (A000,Q4$())Q1$(<2,1> : REM POSITION CURSOR LF
1600 $GID COL/005 (A000,Q4$())Q2$(<1,Q2> : REM POSITION CURSOR
1610 $GID ERS/005 (A000,Q4$())Q3$(<1,Q4> : REM ERASE TEXT
1620 NEXT Q5 : REM CONT ERASING
1630 $GID ROW/005 (A000,Q4$())Q1$(<1,Q1> : REM REPOSITION CURSOR
1640 $GID COL/005 (A000,Q4$())Q2$(<1,Q2> : REM REPOSITION CURSOR
1650 RETURN : REM EXIT
1660 REM **
1670 REM **
1680 REM ** Subroutine > ***** 1 0 2
1690 REM ** Function >
1700 REM ** To load an over lay Routine.
1710 REM ** Variables >
1720 REM ** Q6$() > description ! Q7$() program name
1730 REM ** Q8$() > begin line number !
1740 REM **
1750 REM *****
1760 DEFFN ' 102 (Q6$,Q7$,Q8$) : REM ENTRY
1770 PRINT HEX(030A0A) : REM CLEAR CRT
1780 PRINT USING 1910," " : REM PRINT TITLE
1790 PRINT HEX(0A0A0A) : REM FOR TC USE
1800 GOSUB ' 101 (0B,06,0,0) : REM POSITION CURSOR
1810 PRINT "System Now Loading > ";Q6$;" Routine" : REM PRINT DESCRIPTION
1820 PRINT HEX(0A0A0A) : REM FOR TC USE
1830 GOSUB ' 101 (10,06,0,0) : REM POSITION CURSOR
1840 PRINT "Program being Loaded > ";Q7$ : REM PRINT PROG NAME
1850 IF Q8$ <> "1000" THEN 1870 : REM NOT PROGRAM OVRLY
1860 LOAD DC T #0, Q7$ 100 , 7999
1870 IF Q8$ <> "2000" THEN 1890 : REM NOT PLOT OVRLY
1880 LOAD DC T #0, Q7$ 100
1890 LOAD DC T #0, Q7$ 8000
1900 RETURN : REM EXIT
1910 % TRAINING SCHEDULE PROGRAM %

```


SCHEDULE System --- (SCH.WSHT)

```

100 *** PROGRAM > SCHEDULE MANAGEMENT (SCH.WSHT) VERSION 1.0 780530
110 REM **
120 REM *****
130 REM **
140 REM ** THIS PROGRAM WILL CREATE THE SCHEDULE WORK **
150 REM ** SHEETS FOR ALL OR PART OF THE COURSE FILE **
160 REM ** DATA BASE. **
170 REM **
180 REM *****
190 REM **
200 REM **
1000 DIM D0$(1)4,D4$(1)18,D2$(1)5,D2(2),D3$(4)64 : REM DEFINE WRK BUFRS
1010 DIM P1$4 : REM DEFINE WRK FLAGS
1020 I8 = 0 : REM INIT
1030 I9 = 0 : REM INIT
1040 M8 = 0 : REM INIT
1050 M9 = 0 : REM INIT
1060 Z = 99 : REM INIT LINE INDEX
1062 A5 = 0 : REM FORMAT FLAG
1070 REM **
1080 REM **
1090 GOSUB / 200 : REM SEL OUTPUT FORMAT
1100 GOSUB / 201 : REM OPEN FILES
1110 GOSUB / 203 (1) : REM LOAD FRST REC
1120 GOSUB / 202 : REM SET PRINTR READY
1130 REM **
1140 GOSUB / 203 (2) : REM LOAD NXT REC
1150 IF Q$ = "E" THEN 1270 : REM END OF CRS FILE
1160 GOSUB / 110 : REM CONVERT TO NUMERIC
1170 GOSUB / 111 : REM CMPTUE NSRY INFO
1180 REM **
1190 REM ** ! 80 (REMOTE) ! 132 (LPT) !
1200 REM ** -----
1210 ON A5 GOTO 1240 , 1250 : REM BRANCH
1220 REM ** -----
1230 REM **
1240 GOSUB / 205 : GOTO 1140 : REM PRNT SCH FMT 1
1250 GOSUB / 206 : GOTO 1140 : REM PRNT SCH FMT 2
1260 REM **
1270 GOSUB / 204 : REM PRINT TOTALS
1280 GOSUB / 239 (1) : REM CLOSE KFAM FIL 1
1290 GOSUB / 102 ("MAIN CONTROL","SCH.SCH","1000") : REM LOAD SCHEDULE PROG
1300 END : REM END OF PROGRAM
1310 REM **
1320 REM **
1330 REM ** Subroutine ***** 0 3 2
1340 REM ** Abstract >
1350 REM ** This routine will display a fatal error message on
1360 REM ** console.
1370 REM ** Variables >
1380 REM ** E$ > error message
1390 REM **
1400 REM *****
1410 DEFFN / 32 (E$) : REM ENTRY
1420 SELECT PRINT 005(64) : REM SEL CONSOLE DEV
1430 PRINT HEX(07) : REM PRINT TONE
1440 GOSUB / 101 (16,1,0,0) : REM POSITION CURSOR
1450 PRINT " TOUCH 'RETURN' TO RESTART": : REM PRINT RESTRT MSG

```

```

1460 SELECT P3 : REM SEL PAUSE
1470 GOSUB ' 101 (14,1,1,0) : REM POSITION CURSOR
1480 PRINT
1490 GOSUB ' 101 (14,1,0,0) : REM POSITION CURSOR
1500 PRINT E$ : REM PRINT MSG
1510 SELECT P0 : REM DE-SEL PAUSE
1520 KEYIN F1$,1540,1540 : GOTO 1460 : REM RESPONSE WAIT
1530 REM **
1540 PRINT HEX(030A0A0A) : REM CLR CONSOLE
1550 PRINT " EXECUTE THE FOLLOWING COMMANDS TO RESTART:"
1560 PRINT " :CLEAR V (RETURN(EXEC))"
1570 PRINT " :CLEAR P (RETURN(EXEC))"
1580 PRINT " :LOAD DCT #0, ";HEX(22);"SCHEDULE";HEX(22);"(RETURN(EXEC))"
1590 PRINT " :RUN (RETURN(EXEC))"
1600 PRINT HEX(0A0A)
1610 STOP "ENTER COMMANDS NOW...."
1620 RETURN : REM EXIT
1630 REM **
1640 REM **
1650 REM ** Subroutine > ***** 1 0 1
1660 REM ** Function >
1670 REM ** Cursor Address and Screen erase
1680 REM ** Variables >
1690 REM ** Q1 > row position ! Q3 > lines to erase
1700 REM ** Q2 > col position ! Q4 > char(s) erase
1710 REM ** Q1$() > cursor vert ! Q2$() > cursor horz
1720 REM ** Q3$() > line ers ! Q4$() > micro cmdnd bufr
1730 REM ** Q5 > line indx !
1740 REM **
1750 REM *****
1760 DEFFN ' 101 (Q1,Q2,Q3,Q4) : REM ENTRY
1770 DIM Q1$(16),Q2$(64),Q3$(64),Q4$(10) : REM DEFIN BUFERS
1780 INIT(0A) Q1$() : REM INIT ROW BUFR
1790 INIT(09) Q2$() : REM INIT COL BUFR
1800 INIT(20) Q3$() : REM INIT ERS BUFR
1810 Q1$(1) = HEX(01) : REM INIT TOP OF PAGE
1820 Q2$(1) = HEX(0D) : REM INIT BEG OF LINE
1830 IF Q3 = 0 THEN 1940 : REM NO LINES TO ERASE
1840 IF Q4 <> 0 THEN 1860 : REM CHARS TO ERASE
1850 Q4 = 65 - Q2 : REM ERS REST OF LINE
1860 FOR Q5 = 1 TO Q3 : REM ERS Q3 LINES
1870 IF Q5 <> 1 THEN 1900 : REM NOT FIRST LINE
1880 $GID ROW/005 (A000,Q4$())Q1$(<1,Q1> : REM POSITION CURSOR
1890 GOTO 1910 : REM BRANCH TO POS COL
1900 $GID ROW/005 (A000,Q4$())Q1$(<2,1> : REM POSITION CURSOR LF
1910 $GID COL/005 (A000,Q4$())Q2$(<1,Q2> : REM POSITION CURSOR
1920 $GID ERS/005 (A000,Q4$())Q3$(<1,Q4> : REM ERASE TEXT
1930 NEXT Q5 : REM CONT ERASING
1940 $GID ROW/005 (A000,Q4$())Q1$(<1,Q1> : REM REPOSITION CURSOR
1950 $GID COL/005 (A000,Q4$())Q2$(<1,Q2> : REM REPOSITION CURSOR
1960 RETURN : REM EXIT
1970 REM **
1980 REM **
1990 REM ** Subroutine > ***** 1 0 2
2000 REM ** Function >
2010 REM ** To load an over lay Routine.
2020 REM ** Variables >
2030 REM ** Q6$() > description ! Q7$() program name
2040 REM ** Q8$() > begin line number !
2050 REM **

```

```

2060 REM *****
2070 DEFFN ' 102 (G6$,G7$,G8$) : REM ENTRY
2072 SELECT PRINT 005(64) : REM SEL OUTPUT DEV
2080 PRINT HEX(03) : REM CLEAR CRT
2090 PRINT USING 2180," " : REM PRINT TITLE
2100 GOSUB ' 101 (08,06,0,0) : REM POSITION CURSOR
2110 PRINT "System Now Loading > ";G6$;" Routine" : REM PRINT DESCRIPTION
2120 GOSUB ' 101 (10,06,0,0) : REM POSITION CURSOR
2130 PRINT "Program being Loaded > ";G7$ : REM PRINT PROG NAME
2140 IF G8$ <> "1000" THEN 2160 : REM NOT PROG OVRLY
2150 LOAD DC T #0, G7$ 100 , 7999
2160 LOAD DC T #0, G7$ 8000
2170 RETURN : REM EXIT
2180 % T R A I N I N G S C H E D U L E P R O G R A M #
2190 REM **
2200 REM **
2210 REM ** Subroutine ***** 1 0 4
2220 REM ** Abstract >
2230 REM ** This routine will round a number to the specified
2240 REM ** number of decimal digits.
2250 REM ** Variables >
2260 REM **
2270 REM **
2280 REM *****
2290 DEFFN ' 104 (D1,D2,D3,A9) : REM ENTRY
2300 A9 = 1 : REM INIT RETURN VALUE
2310 IF D1 <= 1.00 THEN 2410 : REM MINIMUM VALUE
2320 REM **
2330 D4 = D1 - INT(D1) : REM FTCH DECML VALUE
2340 D4 = INT(D4 * (10^(D3+1))) : REM MOVE DECML POINT
2350 D4 = ((D4/10) - INT(D4/10)) * 10 : REM FTCH CHK DIGIT
2360 REM **
2370 A9 = INT(D1*(10^(D3)))*10^(-D3) : REM FTCH BASE VALUE
2380 REM **
2390 IF D4 < D2 THEN 2410 : REM ROUND DOWN
2400 A9 = A9 + ((1.0)*(10^(-D3))) : REM ROUND UP
2410 RETURN : REM EXIT
2420 REM **
2430 REM **
2440 REM ** Subroutine ***** 1 1 0
2450 REM ** Abstract >
2460 REM ** This routine will convert input data record into
2470 REM ** numeric information for calculations.
2480 REM ** Variables >
2490 REM **
2500 REM **
2510 REM *****
2520 DEFFN ' 110 : REM ENTRY
2530 CONVERT I1$(Q) TO I1(1) : REM PLANNED INPUT
2540 CONVERT C2$(Q) TO C2(1) : REM CLAS SZ
2550 CONVERT W$(Q) TO W(1) : REM CRS LEN (WKS)
2560 CONVERT M1$(Q) TO M1(1) : REM MAN POWER
2570 CONVERT I9$(Q) TO I9(1) : REM INSTRUCTORS
2580 CONVERT E1$(Q) TO E1(1) : REM TRAINERS
2590 CONVERT S1$(Q) TO S1(1) : REM SUPPORT
2600 CONVERT S2$(Q) TO S2(1) : REM MPA
2610 REM **
2620 FOR K = 1 TO 7 : REM BEG CNTACT LOOP
2630 CONVERT H$(Q,K) TO H(1,K) : REM CNTACT HRS
2640 CONVERT R$(Q,K) TO R(1,K) : REM CNTACT RATIOS

```



```

2650 NEXT K                                     : REM CONT CNTACT LOOP
2660 REM **
2670 RETURN                                     : REM EXIT
2680 REM **
2690 REM **
2700 REM ** Subroutine ***** 1 1 1
2710 REM ** Abstract >
2720 REM ** This routine will perform the following caluclations -
2730 REM ** - Instructor/Comp ;
2740 REM ** - Course Length (days) ;
2750 REM ** - AOB (Average On Board) ;
2760 REM ** - Sum Type Hours (Inst/Comp)
2770 REM ** - Convening Frequency
2780 REM ** - Concurrent Scheduling
2790 REM ** Variables >
2800 REM **
2810 REM **
2820 REM *****
2830 DEFFN ' 111                                : REM ENTRY
2840 W2(1) = W(1)*7-2                            : REM CRS LEN (DAYS)
2850 O(1) = I1(1)*W2(1)/365                      : REM AOB
2860 S(1) = 0                                    : REM INIT TYPE HRS
2870 REM **
2880 FOR K = 1 TO 7                              : REM BEG RATIO LOOP
2890 IF H(1,K) = 0 THEN 2950                     : REM NO HRS
2900 IF R(1,K) = 0 THEN 2950                     : REM NO RATIO
2910 A2 = C2(1)/R(1,K)                          : REM CMPT QUOTA/RATIO
2920 GOSUB ' 104 (A2,5,1,A9)                    : REM ROUND QUOTA/RATIO
2930 GOSUB ' 104 (A9,3,0,A9)                    : REM ROUND QUOTA/RATIO
2940 S(1) = S(1) + (A9*H(1,K))                  : REM SUM TYPE HOURS
2950 NEXT K                                       : REM CONT RATIO LOOP
2960 REM **
2970 A2 = I1(1)/C2(1)                            : REM NUM OF CLASSES
2980 GOSUB ' 104 (A2,1,0,A9)                    : REM ROUND TO WHOLE
2990 C4(1) = A9                                  : REM STOR # OF CLAS
3000 C5(1) = 50/C4(1)                          : REM
3010 C6(1) = W(1)/C5(1)                        : REM
3020 I2(1) = (S(1)*1.10) / (C5(1)*25*.88) : REM # OF INSTRUCTRS
3030 GOSUB ' 104 (I2(1),1,0,A9)                : REM ROUND # OF INSTS
3040 I3(1) = A9                                : REM STOR # OF INSTS
3050 REM **
3060 E2 = C6(1)/2                                : REM
3070 E3(1) = INT(C6(1)/2)                      : REM
3080 IF E3(1)-E2 >= 0 THEN 3100                 : REM
3090 E3(1) = E3(1) + 1                          : REM
3100 RETURN                                     : REM EXIT
3110 REM **
3120 REM **
3130 REM ** Subroutine ***** 1 5 0
3140 REM ** Abstract >
3150 REM ** This routine will select an ALL or PART mode for the
3160 REM ** generation of the schedule.
3170 REM ** Variables >
3180 REM **
3190 REM **
3200 REM *****
3210 DEFFN ' 150 (A1)                            : REM ENTRY
3220 IF A1 = 1 THEN 3310                        : REM LOAD FIRST REG
3230 IF P1$ <> "ALL" THEN 3300                 : REM TST FOR ALL
3240 GOSUB ' 237 (1)                            : REM FIND NXT CRS

```

```

3250 IF          Q$ = " " THEN 3550 : REM BRNCH TO EXIT
3260 IF          Q$ = "E" THEN 3550 : REM BRNCH TO EXIT
3270 GOSUB ' 032 (" KFAM FIND NEXT ERROR... (150)") : REM LOOKUP ERROR
3280 GOTO        3550 : REM BRNCH TO EXIT
3290 REM **
3300 IF          P1$ = "PART" THEN 3450 : REM TST FOR PART
3310 PRINT      HEX(030A0A0A) : REM CLR CONSOLE
3320 GOSUB ' 101 (5,1,0,0) : REM POSITION CUSOR
3330 PRINTUSING 3560," " : REM PRINT MSG
3340 PRINTUSING 3570," ";
3350 INPUT      P1$ : REM INPUT RESPONCE
3360 REM **
3370 IF          P1$ <> "Y" THEN 3440 : REM NOT ALL
3380 IF          P1$ = "ALL" : REM SEL ALL FLAG
3390 GOSUB ' 235 (1) : REM FIND FIRST CRS
3400 IF          Q$ = " " THEN 3550 : REM BRNCH TO EXIT
3410 GOSUB ' 032 (" KFAM FIND FIRST ERROR... (150)") : REM LOOKUP ERROR
3420 GOTO        3550 : REM BRANCH TO EXIT
3430 REM **
3440 IF          P1$ = "PART" : REM SEL PART FLAG
3442 IF          A1 = 1 THEN 3550 : REM BRNCH TO EXIT
3450 GOSUB ' 151 (C1$) : REM ENTER CDP
3460 IF          C1$ <> "STOP" THEN 3500 : REM NOT END OF LIST
3470 IF          Q$ = "E" : REM SET EOF FLAG
3480 GOTO        3550 : REM BRANCH TO EXIT
3490 REM **
3500 GOSUB ' 232 (1,0,C1$) : REM FIND OLD CRS
3510 IF          Q$ = " " THEN 3550 : REM BRNCH TO EXIT
3520 PRINT      HEX(07);"RE-ENTER..." : REM INVALID CDP
3530 GOTO        3450 : REM RE-ENTER CDP
3540 REM **
3550 RETURN : REM EXIT
3560 X Do you wish to have ALL Worksheets printed #
3570 X Enter (Y - yes or N - no) -----> #
3580 REM **
3590 REM **
3600 REM ** Subroutine ***** 1 5 1
3610 REM ** Abstract >
3620 REM ** This routine will allow the CDP number to be
3630 REM ** input and verified for valid syntax.
3640 REM ** Variables >
3650 REM **
3660 REM **
3670 REM *****
3680 DEFFN ' 151 (C1$) : REM ENTRY
3682 SELECT      PRINT 005(64) : REM SEL OUTPUT DEV
3690 GOSUB ' 101 (14,1,1,0) : REM POSITION CUSOR
3700 PRINT      HEX(0A0A) : REM FOR TC USE
3710 GOSUB ' 101 (14,1,0,0) : REM POSITION CUSOR
3720 PRINT      "Enter CDP Number (STOP to Stop) -->"; : REM PRNT PROMPT
3730 INPUT      C1$ : REM INPUT RESPONCE
3740 REM **
3750 FOR          I = 1 TO 4 : REM BEG SYNTAX LOOP
3760 IF          STR(C1$,I,1) > "Z" THEN 3800 : REM INVALID CHAR
3770 IF          STR(C1$,I,1) >= "A" THEN 3840 : REM VALID CHAR
3780 IF          STR(C1$,I,1) > "9" THEN 3800 : REM INVALID CHAR
3790 IF          STR(C1$,I,1) >= "0" THEN 3840 : REM VALID CHAR
3800 PRINT HEX(07);" RE-ENTER...." : REM INVALID CDP
3810 I = 4 : REM SET END OF LOOP
3820 NEXT I : REM TERMINATE LOOP

```

```

3830                                GOTO 3710                                : REM RE-ENTER CDP
3840 NEXT I                                : REM CONT SYNTAX LOOP
3850 REM **
3860 GOSUB ' 202                                : REM SEL OUTPUT DEV
3862 RETURN                                : REM EXIT
3870 REM **
3880 REM **
3890 REM ** Subroutine ***** 2 0 0
3900 REM ** Abstract >
3910 REM ** This routine will select the width of the
3920 REM ** output device.
3930 REM ** Variables >
3940 REM ** A5 > ouput width indx
3950 REM **
3960 REM *****
3970 DEFFN ' 200                                : REM ENTRY
3980 PRINT HEX(030A0A0A)                                : REM CLR CONSOLE
3990 PRINT USING 4130," "                                : REM PRINT PROMPT
4000 PRINT USING 4140," "
4010 PRINT USING 4150," "
4020 PRINT USING 4160," "
4030 PRINT HEX(0A0A)
4040 GOSUB ' 101 (14,1,1,0)                                : REM POSITION CURSOR
4050 PRINT USING 4170," ";                                : REM PRINT PROMPT
4060 INPUT A5                                : REM RESPONSE WAIT
4070 IF A5 > 2 THEN 4090                                : REM INVALID RESPONSE
4080 IF A5 > 0 THEN 4120                                : REM VALID RESPONSE
4090 PRINT HEX(07);"RE-ENTER..."                                : REM PRINT ERR MSG
4100 GOTO 4040
4110 REM **
4120 RETURN                                : REM EXIT
4130 % # The following ouput widths are available for output
4140 % # of the Schedule Worksheet >
4150 % # 1 - 80 character (REMOTE)
4160 % # 2 - 132 character (LPT)
4170 % # Enter the code for desired output width (1-2) ----> "
4180 REM **
4190 REM **
4200 REM ** Subroutine ***** 2 0 1
4210 REM ** Abstract >
4220 REM ** This routine will open the nessary KFAM files.
4230 REM ** Variables >
4240 REM **
4250 REM **
4260 REM *****
4270 DEFFN ' 201                                : REM ENTRY
4280 PRINT HEX(030A0A0A)                                : REM CLR CONSOLE
4290 PRINT " MOUNT SCHEDULE DATA DISK...."                                : REM PRINT PROMPT
4300 PRINT HEX(0A0A0A)                                : REM FOR TC USE
4310 GOSUB ' 101 (8,1,0,0)                                : REM POSITION CURSOR
4320 PRINT " TOUCH 'RETURN' WHEN READY..";                                : REM PRINT PROMPT
4330 INPUT A$                                : REM RESPONSE WAIT
4340 REM **
4350 GOSUB ' 239 (1)                                : REM VERIFY CLOSED
4360 GOSUB ' 230 (1,1,2,1,F1$)                                : REM OPEN DATA BASE
4370 IF Q$ = " " THEN 4400                                : REM FILE OPENED
4380 GOSUB ' 032 (" KFAM OPEN ERROR.. FILE 1")                                : REM PRINT ERR MSG
4390 REM **
4400 GOSUB ' 235 (1)                                : REM FIND FIRST DAT FIL
4410 IF Q$ = " " THEN 4440                                : REM NO ERR FOUND

```



```

4420 GOSUB ' 032 (" KFAM FIND FIRST ERROR ..(201)") : REM DISPLAY ERROR MSG
4430 REM **
4440 RETURN : REM EXIT
4450 REM **
4460 REM **
4470 REM ** Subroutine ***** 2 0 2
4480 REM ** Abstract >
4490 REM ** This routine will initialize the output device.
4500 REM ** Variables >
4510 REM **
4520 REM **
4530 REM *****
4540 DEFFN ' 202 : REM ENTRY
4542 IF A5 <> 0 THEN 4660 : REM SET FORMAT ONLY
4550 PRINT HEX(030A0A0A) : REM CLR CONSOLE
4560 GOSUB ' 101 (5,1,0,0) : REM POSITION CURSOR
4570 PRINT USING 4730," " : REM PRINT READY MSG
4580 PRINT HEX(0A0A0A07) : REM FOR TC USE
4590 GOSUB ' 101 (8,1,0,0) : REM POSITION CURSOR
4600 PRINT USING 4740," " : REM PRINT PROMPT
4610 KEYIN F$,4610,4610 : REM CLR INPUT BUFR
4620 KEYIN F$,4660,4660 : GOTO 4620 : REM RESPONSE WAIT
4630 REM **
4640 REM ** ! 80 (REMOTE) ! 132 (LPT) !
4650 REM ** !-----!
4660 ON A5 GOTO 4690, 4700 : REM BRANCH
4670 REM ** !-----!
4680 REM **
4690 SELECT PRINT 215 (80) : GOTO 4720 : REM SEL OUT DEV 1
4700 SELECT PRINT 215 (132) : GOTO 4720 : REM SEL OUT DEV 2
4710 REM **
4720 RETURN : REM EXIT
4730 Z # INITIALIZE OUTPUT DEVICE ... LINEPRINTER
4740 Z # TOUCH ANY KEY TO CONTINE...
4750 REM **
4760 REM **
4770 REM ** Subroutine ***** 2 0 3
4780 REM ** Abstract >
4790 REM ** This routine will load a record from the data base.
4800 REM ** Variables >
4810 REM **
4820 REM **
4830 REM *****
4840 DEFFN ' 203 : REM ENTRY
4850 GOSUB ' 150 (A1) : REM FIND NXT CRS REC
4860 IF Q$ = "E" THEN 4980 : REM BRNCH TO EXIT
4870 KEYIN F$, 4880, 4880 : GOTO 4920 : REM INTRUPT
4880 IF F$ > HEX(1F) THEN 4920 : REM NOT CNTR CHAR
4890 Q$ = "E" : REM SET EOF FLAG
4900 GOTO 4980 : REM BRNCH TO EXIT
4910 REM **
4920 DATA LOAD DC #2, C1$(),N1$(), T$(),I1$()
: C2$(), W$(), R$(), H$()
: M1$(),I9$(),E1$(),S1$(),S2$()
4930 D0$(1) = C1$(Q) : REM FETCH CDP
4940 D4$(1) = T$(Q) : REM FETCH TITLE
4950 D2$(1) = W$(Q) : REM FETCH CRS LEN
4960 CONVERT D2$(1) TO D2(1) : REM CRS LEN (WKS)
4970 D2(2) = D2(1)*7-2 : REM CRS LEN (DAYS)
4980 RETURN : REM EXIT

```

```

4990 REM **
5000 REM **
5010 REM ** Subroutine ***** 2 0 4
5020 REM ** Abstract >
5030 REM ** This routine will print a total of the following
5040 REM ** values -
5050 REM ** Total Inputs;
5060 REM ** Total instructors required;
5070 REM ** Total instructors mpa;
5080 REM ** Total Zero.
5090 REM ** Variables >
5100 REM **
5110 REM **
5120 REM *****
5130 DEFFN ' 204 : REM ENTRY
5140 PRINT HEX(0C0E); : REM EXPAND PRINT
5150 PRINT USING 5260, " " : REM PRINT HEADING
5160 PRINT HEX(0A0A) : REM POSITION PAGE
5170 PRINT USING 5270, I8 : REM INPUTS
5180 PRINT
5190 PRINT USING 5280, I9 : REM INST REQ
5200 PRINT
5210 PRINT USING 5290, M8 : REM INST MPA
5220 PRINT
5230 PRINT USING 5300, M9 : REM NULL
5240 PRINT HEX(0C) : REM TOP OF FORM
5250 RETURN : REM EXIT
5260 % COURSE SCHEDULE WORK SHEET (SUMMARY) #
5270 % ***** TOTAL INPUTS FOR COURSES PRINTED ..... *****
5280 % ***** TOTAL INSTRUCTOR REQUIRED FOR COURSES PRINTED ..... *****
5290 % ***** TOTAL INSTRUCTORS MPA FOR COURSES PRINTED ..... *****
5300 % ***** TOTAL ))))))) ..... *****
5310 REM **
5320 REM **
5330 REM ** Subroutine ***** 2 0 5
5340 REM ** Abstract >
5350 REM ** This routine will output the schedule worksheet
5360 REM ** according to a 80 column format.
5370 REM ** Variables >
5380 REM **
5390 REM **
5400 REM *****
5410 DEFFN ' 205 : REM ENTRY
5420 IF Z < 55 THEN 5510
5430 PRINT HEX(0C0E); TAB(10); "COURSE SCHEDULE WORK SHEET"
5440 PRINT
5450 PRINT USING 5670, " " : REM PRINT HEADER
5460 PRINT USING 5580, " "
5470 PRINT USING 5590, " -----"
5480 PRINT USING 5600, " "
5490 PRINT USING 5610, " "
5500 Z=7
5510 PRINT USING 5640, C1$(Q); N1$(Q); T$(Q)
5520 PRINT USING 5650, " -----"
5530 PRINT USING 5660, W(1); W2(1); I1(1); O(1); C2(1); C4(1)
: C5(1); C6(1); I3(1); M1(1); I9$(Q);
S1$(Q); S2$(Q); E3(1); E1$(Q)
5540 PRINT USING 5670, " "
5550 I8=I8+I1(1); I9=I9+I3(1); M8=M8+M1(1); M9=M9+M2(1)

```

```

5560          Z = Z + 5                      : REM INCR LINE INDX
5570 RETURN                                     : REM EXIT
5580 Z#CRS LEN !      !      !      !      !      ! INSTRUCTOR !SUPPORT!TRAINER
S!
5590 Z#*****!      !      !CLASS!SKED !CONV !C/C !-----!-----!-----
-!
5600 Z#WKS!DAYS!INPUT! AOB !SIZE !CLASS!FREQ !CLASS!REQ!MPA!OTHER!REQ!MPA!REQ!AV
L!
5610 Z#====!====!====!====!====!====!====!====!====!====!====!====!====
=
5620 Z#                                     !      !      !      !
!
5640 Z CDP ***** CIN ***** ! CRS TITLE ***** !      !      !
!
5650 Z#*****!-----!-----!-----!-----!-----!-----!-----!-----
!
5660 Z ***! ***!*****!**.**.*****!*****!**.**.*****!*****! *** !***!***!***!***
#!
5670 Z#=====
=
5680 REM **
5690 REM **
5700 REM ** Subroutine ***** 2 0 6
5710 REM ** Abstract >
5720 REM ** This routine will print the schedule worksheets
5730 REM ** according to a 132 column format.
5740 REM ** Variables >
5750 REM **
5760 REM **
5770 REM *****
5780 DEFFN ' 206                               : REM ENTRY
5790 IF Z< 55 THEN 5880
5800 PRINT HEX(OCOE);TAB(10);"COURSE SCHEDULE WORK SHEET"
5810 PRINT
5820 PRINTUSING 5960, " "                      : REM PRINT HEADER
5822 PRINTUSING 5930, " "
5830 PRINTUSING 5940, " "
5840 PRINTUSING 5950, " "
5850 PRINTUSING 5960, " "
5860 PRINTUSING 5970, " "
5870 Z = 7
5880 PRINTUSING 5980, C1$(Q);N1$(Q);T$(Q);W(1);W2(1);
I1(1);O(1);C2(1);C4(1);C5(1);C6(1);
I3(1);M1(1);I9$(Q);S1$(Q);S2$(Q);E3(1);E1$(Q)
5890 PRINTUSING 5990, " "
5900 Z = Z+2
5910 I8=I8+I1(1); I9=I9+I3(1); M8=M8+M1(1); M9=M9+M2(1)
5920 RETURN                                     : REM EXIT
5930 Z#      !      !      ! LENGTH      !      ! CLASS ! SK
ED ! CONV ! C/C ! INSTRUCTOR !SUPPORT ! TRAINERS !      !
5940 Z#      !      !-----!-----!-----!-----!-----!-----!-----
!
5950 Z# CDP ! CIN ! TITLE ! WKS ! DAYS! INPUT ! AOB ! SIZE ! CL
ASS ! FREQ ! CLASS !REQ !MPA !OTHER !REQ !MPA! REQ !AVL !
5960 Z#====!-----!-----!-----!-----!-----!-----!-----!-----
=====
5970 Z#      !      !      !      !      !      !      !      !
!
5980 Z ***** ! ***** ! ***** ! *** ! *** ! ***** ! **.**.***** ! **
*** ! **.**.***** ! ***** ! *** ! *** ! ***** ! *** ! *** !
5990 Z#-----!-----!-----!-----!-----!-----!-----!-----
-----!-----!-----!-----!-----!-----!-----!-----

```


SCHEDULE System --- (PLOT.SEL)

```

100 Z## PROGRAM > TRAINING SCHEDULE (PLOT.SEL) VERSION 1.0 2200 T 780629
110 REM **
120 REM *****
130 REM **
140 REM ** THIS PROGRAM WILL ALLOW THE USER TO SELECT THE **
150 REM ** OUTPUT FORMAT OF THE SCHEDULE. ONCE SELECTED THE **
160 REM ** PROGRAM WILL LOAD THE PROPER PROGRAM AND THEN LOAD **
170 REM ** LOAD THE MAIN PLOTTING ROUTINE. **
180 REM ** **
190 REM *****
200 REM **
210 REM **
220 DIM DO$(1)4,D4$(1)18,D2$(1)5,D2(2),D3$(4)64 : REM DEFINE WRK BUFRS
230 DIM P1$4 : REM DEFINE WRK FLAGS
240 REM **
250 GOSUB / 102 ("RE-LOAD COMMKFAM","COMMKFAM","8000") : REM LOAD COMMKFAM
260 REM **
1000 Z## RE-ENTRY FROM COMMKFAM ROUTINE
1002 GOSUB / 200 : REM SEL DUPLUT FORMAT
1010 REM **
1020 REM ** ! FORMAT 1 ! FORMAT 2 ! FORMAT 3 ! FORMAT 4 !
1030 REM ** !-----!-----!-----!-----!
1040 ON A1 GOTO 1080, 1090, 1100, 1110
1050 REM ** !-----!-----!-----!-----!
1060 REM **
1070 REM **
1080 GOSUB / 102 ("PLOT FMT 1","PLOT.001","6000") : REM LOAD FORMAT 1
1090 GOSUB / 102 ("PLOT FMT 2","PLOT.002","6000") : REM LOAD FORMAT 2
1100 GOSUB / 102 ("PLOT FMT 3","PLOT.003","6000") : REM LOAD FORMAT 3
1110 GOSUB / 102 ("PLOT FMT 4","PLOT.004","6000") : REM LOAD FORMAT 4
1120 REM **
1130 REM **
1140 DEFFN / 255 : REM DEFINE RE-ENTRY
1150 GOSUB / 102 ("SCHEDULE PLOT","PLOT.PLT","1000") : REM LOAD PLOTTING PROG
1160 REM **
1170 END : REM END OF OVERLAY
1180 REM **
1190 REM **
1200 REM ** Subroutine > ***** 1 0 1
1210 REM ** Function >
1220 REM ** Cursor Address and Screen erase
1230 REM ** Variables >
1240 REM ** Q1 > row position ! Q3 > lines to erase
1250 REM ** Q2 > col position ! Q4 > char(s) erase
1260 REM ** Q1$() > cursor vert ! Q2$() > cursor horz
1270 REM ** Q3$() > line ers ! Q4$() > micro cmdnd bufr
1280 REM ** Q5 > line indx !
1290 REM **
1300 REM *****
1310 DEFFN / 101 (Q1,Q2,Q3,Q4) : REM ENTRY
1320 DIM Q1$(16)1,Q2$(64)1,Q3$(64)1,Q4$(10)1 : REM DEFIN BUFRS
1330 INIT(0A) Q1$() : REM INIT ROW BUFR
1340 INIT(09) Q2$() : REM INIT COL BUFR
1350 INIT(20) Q3$() : REM INIT ERS BUFR
1360 Q1$(1) = HEX(01) : REM INIT TOP OF PAGE
1370 Q2$(1) = HEX(0D) : REM INIT BEG OF LINE
1380 IF Q3 = 0 THEN 1490 : REM NO LINES TO ERASE
1390 IF Q4 <> 0 THEN 1410 : REM CHARS TO ERASE

```

```

1400          G4 = 65 - G2                      : REM ERS REST OF LINE
1410 FOR      G5 = 1 TO G3                      : REM ERASE G3 LINES
1420 IF      G5 <> 1                            THEN 1450 : REM NOT FIRST LINE
1430 $GIO ROW/005 (A000,G4$())G1$()<1,G1>      : REM POSITION CURSOR
1440 GOTO    1460                              : REM BRANCH TO POS COL
1450 $GIO ROW/005 (A000,G4$())G1$()<2,1>      : REM POSITION CURSOR LF
1460 $GIO COL/005 (A000,G4$())G2$()<1,G2>      : REM POSITION CURSOR
1470 $GIO ERS/005 (A000,G4$())G3$()<1,G4>      : REM ERASE TEXT
1480 NEXT    G5                              : REM CONT ERASING
1490 $GIO ROW/005 (A000,G4$())G1$()<1,G1>      : REM REPOSITION CURSOR
1500 $GIO COL/005 (A000,G4$())G2$()<1,G2>      : REM REPOSITION CURSOR
1510 RETURN                                     : REM EXIT
1520 REM **
1530 REM **
1540 REM ** Subroutine > ***** 1 0 2
1550 REM ** Function >
1560 REM ** To load an over lay Routine.
1570 REM ** Variables >
1580 REM ** G6$() > description ! G7$() program name
1590 REM ** G8$() > begin line number !
1600 REM **
1610 REM *****
1620 DEFFN ' 102 (G6$,G7$,G8$)                  : REM ENTRY
1630 PRINT    HEX(03)                          : REM CLEAR CRT
1640 PRINTUSING 1750," "                      : REM PRINT TITLE
1650 GOSUB ' 101 (08,06,0,0)                  : REM POSITION CURSOR
1660 PRINT    "System Now Loading > ";G6$;" Routine" : REM PRINT DESCRIPTION
1670 GOSUB ' 101 (10,06,0,0)                  : REM POSITION CURSOR
1680 PRINT    "Program being Loaded > ";G7$      : REM PRINT PROG NAME
1690 IF      G8$ <> "1000"                      THEN 1710 : REM NOT PROG OVRLY
1700 LOAD    DC T #0, G7$ 100 , 5999
1710 IF      G8$ <> "6000"                      THEN 1730 : REM NOT PLOT OVRLY
1720 LOAD    DC T #0, G7$ 6000 , 7999
1730 LOAD    DC T #0, G7$ 8000
1740 RETURN                                     : REM EXIT
1750 % TRAINING SCHEDULE PROGRAM #
1760 REM **
1770 REM **
1780 REM ** Subroutine ***** 2 0 0
1790 REM ** Abstract >
1800 REM ** This routine will allow the user to select the
1810 REM ** type of output format to print the schedule.
1820 REM ** Variables >
1830 REM **
1840 REM **
1850 REM *****
1860 DEFFN ' 200                                : REM ENTRY
1870 PRINT    HEX(030A0A0A)                    : REM CLR CONSOLE
1880 PRINTUSING 2050," "                      : REM PRINT MENU
1890 PRINTUSING 2060," "
1900 PRINT
1910 PRINTUSING 2070," "
1920 PRINTUSING 2080," "
1930 PRINTUSING 2090," "
1940 PRINTUSING 2100," "
1950 PRINT
1960 GOSUB ' 101 (14,1,1,0)                    : REM POSITION CURSOR
1970 PRINTUSING 2110," ";
1980 INPUT    A1                                : REM INPUT RESPONSE
1990 IF      A1 < 1                            THEN 2010 : REM INVALID RESPONSE

```

```

2000 IF          A1 < 5                      THEN 2040 : REM VALID RESPONSE
2010 PRINT USING 2120," "                    : REM ERR MSG
2020 GOTO        1960                        : REM RE-ENTER FMT CODE
2030 REM **
2040 RETURN                                          : REM EXIT
2050 % The Following Output Formats Are Available #
2060 % For Schedule Output. #
2070 % #      1 - Vertical Format   (80) <REMOTE>
2080 % #      2 - Vertical Format   (132) <LPT>
2090 % #      2 - Horiz. Format     (80) <REMOTE>
2100 % #      3 - Horiz. Format     (132) <LPT>
2110 %      Enter Format Code (1-4) #
2120 %      INVALID ENTRY --- RE-ENTER.. #

```


SCHEDULE System --- (PLOT.PLT)

```

100 Z## PROGRAM > TRAINING SCHEDULE (PLOT.PLT) VERSION 1.0 2200 T 780629
110 REM **
120 REM *****
130 REM **
140 REM ** THIS PROGRAM WILL PRINT THE SCHEDULE FOR ALL COURSE **
150 REM ** IN THE DATA BASE. **
160 REM **
170 REM *****
180 REM **
190 REM **
1000 DIM D0$(1)4,D4$(1)18,D2$(1)5,D2(2),D3$(4)64 : REM DEFINE WRK BUFRS
1010 DIM P1$4 : REM DEFINE WRK FLAGS
1020 REM **
1030 REM **
1040 GOSUB / 201 : REM OPEN FILES
1050 REM **
1060 GOSUB / 202 : REM LOAD RECORD
1070 IF Q$ = "E" THEN 1150 : REM END OF CRS FILE
1080 GOSUB / 110 : REM CONVERT TO NUMERIC
1090 GOSUB / 111 : REM COMPUTE NSRY INFO
1100 GOSUB / 112 : REM BUILD SCHEDULE
1110 GOSUB / 113 : REM SAVE SCHEDULE
1120 REM **
1130 GOSUB / 205 : GOTO 1060 : REM PRINT SCH FMT 1
1140 REM **
1150 GOSUB / 239 (1) : REM CLOSE KFAM FIL 1
1160 GOSUB / 239 (2) : REM CLOSE KFAM FIL 2
1165 COM CLEAR M$( ) : REM CLR COMMON
1170 GOSUB / 102 ("MAIN CONTROL","SCH.SCH","1000") : REM LOAD SCHEDULE PROG
1180 END : REM END OF PROGRAM
1190 REM **
1200 REM **
1210 REM ** Subroutine ***** 0 3 2
1220 REM ** Abstract >
1230 REM ** This routine will display a fatal error message on
1240 REM ** console.
1250 REM ** Variables >
1260 REM ** E$ > error message
1270 REM **
1280 REM *****
1290 DEFFN / 32 (E$) : REM ENTRY
1300 SELECT PRINT 005(64) : REM SEL CONSOLE DEV
1310 PRINT HEX(07) : REM PRINT TONE
1320 GOSUB / 101 (16,1,0,0) : REM POSITION CURSOR
1330 PRINT " TOUCH 'RETURN' TO RESTART": : REM PRINT RESTRT MSG
1340 SELECT P3 : REM SEL PAUSE
1350 GOSUB / 101 (14,1,1,0) : REM POSITION CURSOR
1360 PRINT
1370 GOSUB / 101 (14,1,0,0) : REM POSITION CURSOR
1380 PRINT E$ : REM PRINT MSG
1390 SELECT P0 : REM DE-SEL PAUSE
1400 KEYIN F1$,1420,1420 : GOTO 1340 : REM RESPONSE WAIT
1410 REM **
1420 PRINT HEX(030A0A0A) : REM CLR CONSOLE
1430 PRINT " EXECUTE THE FOLLOWING COMMANDS TO RESTART:"
1440 PRINT " :CLEAR V (RETURN(EXEC))"
1450 PRINT " :CLEAR P (RETURN(EXEC))"
1460 PRINT " :LOAD DCT #0, ";HEX(22);"SCHEDULE";HEX(22);"(RETURN(EXEC))"

```

```

1470 PRINT "      :RUN (RETURN(EXEC))"
1480 PRINT HEX(0A0A)
1490 STOP "ENTER COMMANDS NOW...."
1500 RETURN                                     : REM EXIT
1510 REM **
1520 REM **
1530 REM ** Subroutine > ***** 1 0 1
1540 REM ** Function >
1550 REM ** Cusor Address and Screen erase
1560 REM ** Variables >
1570 REM ** Q1 > row position ! Q3 > lines to erase
1580 REM ** Q2 > col position ! Q4 > char(s) erase
1590 REM ** Q1$() > cusor vert ! Q2$() > cusor horz
1600 REM ** Q3$() > line ers ! Q4$() > micro cmdnd bufr
1610 REM ** Q5 > line indx !
1620 REM **
1630 REM *****
1640 DEFFN ' 101 (Q1,Q2,Q3,Q4) : REM ENTRY
1650 DIM Q1$(16),Q2$(64),Q3$(64),Q4$(10) : REM DEFIN BUFERS
1660 INIT(0A) Q1$() : REM INIT ROW BUFR
1670 INIT(09) Q2$() : REM INIT COL BUFR
1680 INIT(20) Q3$() : REM INIT ERS BUFR
1690 Q1$(1) = HEX(01) : REM INIT TOP OF PAGE
1700 Q2$(1) = HEX(0D) : REM INIT BEG OF LINE
1710 IF Q3 = 0 THEN 1820 : REM NO LINES TO ERASE
1720 IF Q4 <> 0 THEN 1740 : REM CHARS TO ERASE
1730 Q4 = 65 - Q2 : REM ERS REST OF LINE
1740 FOR Q5 = 1 TO Q3 : REM ERASE Q3 LINES
1750 IF Q5 <> 1 THEN 1780 : REM NOT FIRST LINE
1760 $GIO ROW/005 (A000,Q4$())Q1$(<1,Q1) : REM POSITION CUSOR
1770 GOTO 1790 : REM BRANCH TO POS COL
1780 $GIO ROW/005 (A000,Q4$())Q1$(<2,1) : REM POSITION CUSOR LF
1790 $GIO COL/005 (A000,Q4$())Q2$(<1,Q2) : REM POSITION CUSOR
1800 $GIO ERS/005 (A000,Q4$())Q3$(<1,Q4) : REM ERASE TEXT
1810 NEXT Q5 : REM CONT ERASING
1820 $GIO ROW/005 (A000,Q4$())Q1$(<1,Q1) : REM REPOSITION CUSOR
1830 $GIO COL/005 (A000,Q4$())Q2$(<1,Q2) : REM REPOSITION CUSOR
1840 RETURN : REM EXIT
1850 REM **
1860 REM **
1870 REM ** Subroutine > ***** 1 0 2
1880 REM ** Function >
1890 REM ** To load an over lay Routine.
1900 REM ** Variables >
1910 REM ** Q6$() > description ! Q7$() program name
1920 REM ** Q8$() > begin line number !
1930 REM **
1940 REM *****
1950 DEFFN ' 102 (Q6$,Q7$,Q8$) : REM ENTRY
1960 PRINT HEX(03) : REM CLEAR CRT
1970 PRINT USING 2060," " : REM PRINT TITLE
1980 GOSUB ' 101 (08,06,0,0) : REM POSITION CUSOR
1990 PRINT "System Now Loading > ";Q6$;" Routine" : REM PRINT DESCRIPTION
2000 GOSUB ' 101 (10,06,0,0) : REM POSITION CUSOR
2010 PRINT "Program being Loaded > ";Q7$ : REM PRINT PROG NAME
2020 IF Q8$ <> "1000" THEN 2040 : REM NOT PROG OVRLY
2030 LOAD DC T #0, Q7$ 100 , 7999
2040 LOAD DC T #0, Q7$ 8000
2050 RETURN : REM EXIT
2060 X TRAINING SCHEDULE PROGRAM #

```

```

2070 REM **
2080 REM **
2090 REM ** Subroutine ***** 1 0 4
2100 REM ** Abstract >
2110 REM ** This routine will round a number to the specified
2120 REM ** number of decimal digits.
2130 REM ** Variables >
2140 REM **
2150 REM **
2160 REM *****
2170 DEFFN ' 104 (D1,D2,D3,A9) : REM ENTRY
2180 A9 = 1 : REM INIT RETURN VALUE
2190 IF D1 <= 1.00 THEN 2290 : REM MINIMUM VALUE
2200 REM **
2210 D4 = D1 - INT(D1) : REM FTCH DECML VALUE
2220 D4 = INT(D4 * (10↑(D3+1))) : REM MOVE DECML POINT
2230 D4 = ((D4/10) - INT(D4/10)) * 10 : REM FTCH CHK DIGIT
2240 REM **
2250 A9 = INT(D1*(10↑(D3)))*10↑(-D3) : REM FTCH BASE VALUE
2260 REM **
2270 IF D4 < D2 THEN 2290 : REM ROUND DOWN
2280 A9 = A9 + ((1.0)*(10↑(-D3))) : REM ROUND UP
2290 RETURN : REM EXIT
2300 REM **
2310 REM **
2320 REM ** Subroutine ***** 1 1 0
2330 REM ** Abstract >
2340 REM ** This routine will convert input data record into
2350 REM ** numeric information for calculations.
2360 REM ** Variables >
2370 REM **
2380 REM **
2390 REM *****
2400 DEFFN 110 : REM ENTRY
2410 CONVERT I1$(Q) TO I1(1) : REM PLANNED INPUT
2420 CONVERT C2$(Q) TO C2(1) : REM CLAS SIZ
2430 CONVERT W$(Q) TO W(1) : REM CKS LEN (WKS)
2440 CONVERT M1$(Q) TO M1(1) : REM MAN POWER
2450 CONVERT I9$(Q) TO I9(1) : REM INSTRUCTORS
2460 CONVERT E1$(Q) TO E1(1) : REM TRAINERS
2470 CONVERT S1$(Q) TO S1(1) : REM SUPPORT
2480 CONVERT S2$(Q) TO S2(1) : REM MPA
2490 REM **
2500 FOR K = 1 TO 7 : REM BEG CNTACT LOOP
2510 CONVERT H$(Q,K) TO H(1,K) : REM CNTACT HRS
2520 CONVERT R$(Q,K) TO R(1,K) : REM CNTACT RATIOS
2530 NEXT K : REM CONT CNTACT LOOP
2540 REM **
2550 RETURN : REM EXIT
2560 REM **
2570 REM **
2580 REM ** Subroutine ***** 1 1 1
2590 REM ** Abstract >
2600 REM ** This routine will perform the following calucations -
2610 REM ** - Instructor/Comp ;
2620 REM ** - Course Length (days) ;
2630 REM ** - AOB (Average On Board) ;
2640 REM ** - Sum Type Hours (Inst/Comp)
2650 REM ** - Convening Frequency
2660 REM ** - Concurrent Scheduling

```



```

2670 REM ** Variables >
2680 REM **
2690 REM **
2700 REM *****
2710 DEFFN ' 111
2720          W2(1) = W(1)*7-2           : REM ENTRY
2730          O(1) = I1(1)*W2(1)/365     : REM CRS LEN (DAYS)
2740          S(1) = 0                   : REM AOB
2750 REM **
2760 FOR      K = 1 TO 7                 : REM BEG RATIO LOOP
2770 IF      H(1,K) = 0                 THEN 2830 : REM NO HRS
2780 IF      R(1,K) = 0                 THEN 2830 : REM NO RATIO
2790 A2 = C2(1)/R(1,K)                  : REM CMPT QUOTA/RATIO
2800 GOSUB ' 104 (A2,5,1,A9)            : REM ROUND QUOTA/RATIO
2810 GOSUB ' 104 (A9,3,0,A9)            : REM ROUND QUOTA/RATIO
2820 S(1) = S(1) + (A9*H(1,K))          : REM SUM TYPE HOURS
2830 NEXT K                             : REM CONT RATIO LOOP
2840 REM **
2850 A2 = I1(1)/C2(1)                   : REM NUM OF CLASSES
2860 GOSUB ' 104 (A2,1,0,A9)            : REM ROUND TO WHOLE
2870 C4(1) = A9                         : REM STOR # OF CLAS
2880 C5(1) = 50/C4(1)                  : REM
2890 C6(1) = W(1)/C5(1)                 : REM
2900 I2(1) = (S(1)*1.10) / (C5(1)*25*.88) : REM # OF INSTRUCTRS
2910 GOSUB ' 104 (I2(1),1,0,A9)         : REM ROUND # OF INSTS
2920 I3(1) = A9                         : REM STOR # OF INSTS
2930 REM **
2940 E2 = C6(1)/2                       : REM
2950 E3(1) = INT(C6(1)/2)                : REM
2960 IF      E3(1)-E2 >= 0               THEN 2980 : REM
2970 E3(1) = E3(1) + 1                  : REM
2980 RETURN                             : REM EXIT
2990 REM **
3000 REM **
3010 REM ** Subroutine ***** 1 1 2
3020 REM ** Abstract >
3030 REM ** This routine will build a schedule for the current
3040 REM ** course and stor the information into schedule buffer
3050 REM ** for storage on disk and for later processing.
3060 REM ** Variables >
3070 REM ** S8$() > schedule buffer !
3080 REM **
3090 REM **
3100 REM *****
3110 DEFFN ' 112
3120 INIT(30) S8$()                    : REM ENTRY
3130 J5 = 1                             : REM CLR SCH BUFFER
3140 IF      C4(1)<50                     THEN 3250 : REM
3150 C4=C4(1)-50                          : REM
3160 C4(1)=C4(1)-C4                       : REM
3170 C9=INT(50/C4)                       : REM
3180 S=1                                  : REM
3190 REM **
3200 FOR      J1=J5 TO C4                 : REM
3210 S8$(2,S)="1"                        : REM
3220 S=S+C9                               : REM
3230 NEXT J1                             : REM
3240 REM **
3250 S2=INT(C5(1))                        : REM
3260 IF      C5(1)=S2                     THEN 3350 : REM

```

```

3270          C5(1)=S2+1                      : REM
3280          S3=C4(1)*C5(1)                  : REM
3290 IF      S3<=50                          THEN 3350 : REM
3300 IF      S2<> 0                          THEN 3340 : REM
3310          C5(1)=1                        : REM
3320 GOTO    3350                            : REM
3330 REM **
3340          C5(1)=S2                        : REM
3350          S=1                            : REM
3360 REM **
3370 FOR      J1=J5 TO C4(1)                  : REM
3380          S8$(1,S)="1"                   : REM
3390          S=S+C5(1)                       : REM
3400 NEXT     J1                             : REM
3410 REM **
3420 RETURN                                     : REM EXIT
3430 REM **
3440 REM **
3450 REM ** Subroutine ***** 1 1 3
3460 REM ** Abstract >
3470 REM **   This routine will save a schedule buffer on the
3480 REM **   disk.
3490 REM ** Variables >
3500 REM **
3510 REM **
3520 REM *****
3530 DEFFN ' 113                               : REM ENTRY
3540 DIM      D3$(4)64                        : REM DEFN I/O BUFFER
3550 MAT COPY C1$(<<(Q-1)*4+1,4> TO D3$(<<1,4> : REM LOAD CDP
3560 MAT COPY S8$(<<1,100> TO D3$(<<5,100>    : REM LOAD SCH BUFFER
3570 REM **
3580 GOSUB ' 232 (2,0,C1$)                     : REM FIND OLD REC
3590 IF      Q$ = "N"                          THEN 3640 : REM NO SUCH REC
3600 IF      Q$ = " "                          THEN 3660 : REM REC FOUND
3610 GOSUB ' 032 (" KFAM FINDOLD ERROR ... (113)") : REM FIND OLD ERROR
3620 GOTO    3690                             : REM BRNCH TO EXIT
3630 REM **
3640 GOSUB ' 114                               : REM FIND NEW FIL SPACE
3650 REM **
3660 LIMITS T #5, S4,S5,S6                    : REM FTCH SCTR ADRS
3670 DATA SAVE BA T#5, (S6,S5) D3$()        : REM SAVE BUFFER
3680 REM **
3690 RETURN                                     : REM EXIT
3700 REM **
3710 REM **
3720 REM ** Subroutine ***** 1 1 4
3730 REM ** Abstract >
3740 REM **   This routine will find new record in the SCHEDULE
3750 REM **   data base.
3760 REM ** Variables >
3770 REM **
3780 REM **
3790 REM *****
3800 DEFFN ' 114                               : REM ENTRY
3810 DATA LOAD DC OPEN T #6, "PLOT.DEL"      : REM OPEN DEL FILE
3820 DATA LOAD DC #6, D1$(),D1              : REM LOAD DEL BUFR
3830 REM **
3840 IF      D1 = 0                            THEN 3940 : REM NO PREVIOUS SPACE
3850          T8$ = "1"                        : REM SET KFAM DEL FLG
3860          T4$ = D1$(D1)                    : REM SET KFAM ADRS PNTR

```

```

3870          D1$(D1) = "000"          : REM CLR DEL BUF ELMNT
3880          D1 = D1 - 1                : REM DCRMNT DEL BUF IDX
3890 DBACKSPACE #6, BEG                 : REM RE-POSITION FILE
3900 DATA SAVE DC #6, D1$( ), D1      : REM RE-SAVE DEL BUFR
3910 GOSUB ' 234 (2,0,C1$,0)           : REM FINDNEW(HERE)
3920 GOTO      3950                     : REM BRANCH TO EXIT
3930 REM **
3940 GOSUB ' 233 (2,0,C1$,0)           : REM FIND NEW
3950 IF      Q$ = " " THEN 3980         : REM SPACE FOUND
3960 GOSUB ' 032 (" KFAM FINDNEW ERROR... (114)") : REM FINDNEW ERR
3970 REM **
3980 RETURN                             : REM EXIT
3990 REM **
4000 REM **
4010 REM ** Subroutine ***** 1 5 0
4020 REM ** Abstract >
4030 REM ** This routine will select an ALL or PART mode for the
4040 REM ** generation of the schedule.
4050 REM ** Variables >
4060 REM **
4070 REM **
4080 REM *****
4090 DEFFN ' 150                        : REM ENTRY
4100 IF      P1$ <> "ALL" THEN 4170     : REM TST FOR ALL
4110 GOSUB ' 237 (1)                   : REM FIND NXT CRS
4120 IF      Q$ = " " THEN 4410        : REM BRNCH TO EXIT
4130 IF      Q$ = "E" THEN 4410        : REM BRNCH TO EXIT
4140 GOSUB ' 032 (" KFAM FIND NEXT ERROR... (150)") : REM LOOKUP ERROR
4150 GOTO      4410                     : REM BRNCH TO EXIT
4160 REM **
4170 IF      P1$ = "PART" THEN 4320    : REM TST FOR PART
4180 PRINT    HEX(030A0A0A)            : REM CLR CONSOLE
4190 GOSUB ' 101 (5,1,0,0)            : REM POSITION CURSOR
4200 PRINTUSING 4420," "              : REM PRINT MSG
4210 PRINTUSING 4430," ";
4220 INPUT    P1$                      : REM INPUT RESPONSE
4230 REM **
4240 IF      P1$ <> "Y" THEN 4310       : REM NOT ALL
4250 P1$ = "ALL"                       : REM SEL ALL FLAG
4260 GOSUB ' 235 (1)                   : REM FIND FIRST CRS
4270 IF      Q$ = " " THEN 4410        : REM BRNCH TO EXIT
4280 GOSUB ' 032 (" KFAM FIND FIRST ERROR... (150)") : REM LOOKUP ERROR
4290 GOTO      4410                     : REM BRNCH TO EXIT
4300 REM **
4310 P1$ = "PART"                      : REM SEL PART FLAG
4320 GOSUB ' 151 (C1$)                 : REM ENTER CDP
4330 IF      C1$ <> "STOP" THEN 4360   : REM NOT END OF LIST
4340 Q$ = "E"                         : REM SET EOF FLAG
4350 GOTO      4410                     : REM BRNCH TO EXIT
4360 GOSUB ' 232 (1,0,C1$)            : REM FIND OLD CRS
4370 IF      Q$ = " " THEN 4410        : REM BRNCH TO EXIT
4380 PRINT    HEX(07);"RE-ENTER..." : REM INVALID CDP
4390 GOTO      4320                     : REM RE-ENTER CDP
4400 REM **
4410 RETURN                             : REM EXIT
4420 % Do you wish to have ALL schedules printed #
4430 % Enter (Y - yes or N - no) -----> #
4440 REM **
4450 REM **
4460 REM ** Subroutine ***** 1 5 1

```



```

4470 REM ** Abstract >
4480 REM ** This routine will allow the CDP number to be
4490 REM ** input and verified for valid syntax.
4500 REM ** Variables >
4510 REM **
4520 REM **
4530 REM *****
4540 DEFFN ' 151 (C1%) : REM ENTRY
4550 GOSUB ' 101 (14,1,1,0) : REM POSITION CUSOR
4560 PRINT HEX(0A0A) : REM FOR TC USE
4570 GOSUB ' 101 (14,1,0,0) : REM POSITION CUSOR
4580 PRINT " Enter CDP Number (STOP to Stop) -->": REM PRNT PROMPT
4590 INPUT C1% : REM INPUT RESPONCE
4600 REM **
4610 FOR I = 1 TO 4 : REM BEG SYNTAX LOOP
4620 IF STR(C1%,I,1) > "Z" THEN 4660 : REM INVALID CHAR
4630 IF STR(C1%,I,1) >= "A" THEN 4700 : REM VALID CHAR
4640 IF STR(C1%,I,1) > "9" THEN 4660 : REM INVALID CHAR
4650 IF STR(C1%,I,1) >= "0" THEN 4700 : REM VALID CHAR
4660 PRINT HEX(07):" RE-ENTER...." : REM INVALID CDP
4670 I = 4 : REM SET END OF LOOP
4680 NEXT I : REM TERMINATE LOOP
4690 GOTO 4570 : REM RE-ENTER CDP
4700 NEXT I : REM CONT SYNTAX LOOP
4710 REM **
4720 RETURN : REM EXIT
4730 REM **
4740 REM **
4750 REM ** Subroutine ***** 2 0 1
4760 REM ** Abstract >
4770 REM ** This routine will open the nessary KFAM files.
4780 REM ** Variables >
4790 REM **
4800 REM **
4810 REM *****
4820 DEFFN ' 201 : REM ENTRY
4830 PRINT HEX(03CA0A0A) : REM CLR CONSOLE
4840 GOSUB ' 101 (5,1,0,0) : REM POSITION CUSOR
4850 PRINT " MOUNT SCHEDULE DATA DISK...." : REM PRINT PROMPT
4860 GOSUB ' 101 (8,1,0,0) : REM POSITION CUSOR
4870 PRINT " TOUCH 'RETURN' WHEN READY..": REM PRINT PROMPT
4880 INPUT A% : REM RESPONCE WAIT
4890 REM **
4900 GOSUB ' 239 (1) : REM VERIFY CLOSED
4910 GOSUB ' 230 (1,1,2,1,F1%) : REM OPEN DATA BASE
4920 IF Q% = " " THEN 4950 : REM FILE OPENED
4930 GOSUB ' 032 (" KFAM OPEN ERROR.. FILE 1") : REM PRINT ERR MSG
4940 REM **
4950 GOSUB ' 239 (2) : REM VERIFY CLOSED
4960 GOSUB ' 230 (2,4,5,1,F2%) : REM OPEN PLOT FILE
4970 IF Q% = " " THEN 5040 : REM FILE OPENED
4980 GOSUB ' 032 (" KFAM OPEN ERROR.. FILE 2") : REM PRINT ERR MSG
4990 REM **
5040 RETURN : REM EXIT
5050 REM **
5060 REM **
5070 REM ** Subroutine ***** 2 0 2
5080 REM ** Abstract >
5090 REM ** This routine will load a record from the data base.
5100 REM ** Variables >

```

```

5110 REM **
5120 REM **
5130 REM *****
5140 DEFFN / 202 : REM ENTRY
5150 GOSUB / 150 : REM FIND NXT CRS REC
5160 IF Q$ = "E" THEN 5280 : REM BRNCH TO EXIT
5170 KEYIN F$, 5180, 5180 : GOTO 5220 : REM INTERRUPT
5180 IF F$ > HEX(1F) THEN 5220 : REM NOT CNTRL CHAR
5190 Q$ = "E" : REM SET EOF FLAG
5200 GOTO 5280 : REM BRNCH TO EXIT
5210 REM **
5220 DATA LOAD DC #2 , C1$(),N1$(), T$(),I1$()
: C2$(), W$(), R$(), H$()
: M1$(),I9$(),E1$(),S1$(),S2$()
5230 D0$(1) = C1$(Q) : REM FETCH CDP
5240 D4$(1) = T$(Q) : REM FETCH TITLE
5250 D2$(1) = W$(Q) : REM FETCH CRS LEN
5260 CONVERT D2$(1) TO D2(1) : REM CRS LEN (WKS)
5270 D2(2) = D2(1)*7-2 : REM CRS LEN (DAYS)
5272 D8 = Q : REM SAVE DATA POINTR
5280 RETURN : REM EXIT

```

SCHEDULE System --- (FILE.INI)

```

100 X## PROGRAM > SCHEDULE PROGRAMMING (FILE.INI) 3/7/79
110 REM **
120 REM **
130 REM ** ABSTRACT *****
140 REM **
150 REM **      This program will initialize all data base
160 REM **      files needed by the SCHEDULE programming
170 REM **      system.
180 REM **
190 REM ** NOTE.
200 REM **      parameters such as-
210 REM **          * disk address;
220 REM **          * max number of records in data base;
230 REM **          * max number of records in delete file;
240 REM **          * other KFAM (kir or kdr) parameters;
250 REM **      Should only be modified by a person fimilar
260 REM **      with the initializing of KFAM-3 data files.
270 REM **
280 REM *****
290 REM **
300 REM **
310 DIM      YO(6)                : REM FILE STATUS BUFR
320 DIM      P1$(250)3           : REM SYS DEL BUFFER
330 DIM      P1$8,P2$8           : REM DATA BASE FILNAMS
340 DIM      P3$8,P4$8           : REM PLOT BASE FILNAMS
350 P1 = 150                     : REM MAX REC DATA BASE
360 P2 = 250                     : REM MAX REC DEL FILE
370 P1$ = "SCHEF101"             : REM DATA FILE NAME
380 P2$ = "SCHEK101"             : REM KEY FILE NAME
390 P3$ = "PLOT F101"            : REM PLOT FILE NAME
400 P4$ = "PLOT K101"            : REM KEY FILE NAME
410 SELECT #1 B10                : REM SEL KFAM USER FILE
420 SELECT #2 B10                : REM SEL KFAM KEY FILE
430 SELECT #3 B10                : REM SYS DELETE FILE
431 PRINT HEX(03)
432 GOSUB '032("PGM READY (RETURN)", " ")
440 REM **
450 REM **      INITIALIZE DELETE FILES
460 REM **
470 P5 = INT((P2*4)/253) + 5
480 INIT(00) P1$()
490 P3 = 0
500 DATA SAVE DC OPEN T #3,P5,"SCHE.DEL"
510 DATA SAVE DC #3, P1$(),P3
520 DATA SAVE DC #3, END
530 DATA SAVE DC OPEN T #3,P5,"PLOT.DEL"
540 DATA SAVE DC #3, P1$(),P3
550 DATA SAVE DC #3, END
560 REM **
570 REM **      INITIALIZE DATA BASE
580 REM **
590 GOSUB '180(P1$,P2$,1,2,"A",1,126,2,4,3,P1,3,RO)
600 IF      RO = 0                THEN 670 : REM FILE INITIALIZED
610 GOSUB ' 032 ("FILE INIT ERR",P1$): REM PRINT ERR MSG
620 STOP "CHECK GOSUB ' 180"
630 REM **
640 REM **
650 REM **      INITIALIZE PLOT FILE

```



```

660 REM **
670 GOSUB '180(P3$,P4$,1,2,"N",1,256,1,4,0,P1,3,R0)
680 IF      R0 = 0      THEN 740 : REM FILE INITIALIZED
690 GOSUB ' 032 ("FILE INIT ERR",P3$): REM PRINT ERR MSG
700 STOP "CHECK GOSUB ' 180"
710 REM **
720 REM **                      FILES INITIALIZED
730 REM **
740      GOSUB ' 032 ("NORMAL PGM END"," ")
742      LIST DC T #2
750      END

```

```

1000 X## PROGRAM > CRT/LPT UTILITIES (PRINT.UTL) VERSION 1.0 2200 T/VP
1055 DIM E$24,E0$24
1060 DEFFN '032(E$,E0$)
: SELECT PRINT 005(80)
: PRINT HEX(0706)
1075 SELECT P2
: GOSUB '101(1,40,1,24)
: PRINT
: GOSUB '101(1,40,0,0)
: SELECT P3
: PRINT E$
: SELECT P0
: KEYIN F0$,1120,1120
: GOTO 1075

1120 GOSUB '101(1,40,1,24)
: PRINT E0$
: RETURN

1200 DEFFN '101(Q1,Q2,Q3,Q4)
: DIM Q1$(24),Q2$(80),Q3$(80),Q4$(10)
: INIT(0A)Q1$()
: INIT(09)Q2$()
: INIT(20)Q3$()
: Q1$(1)=HEX(01)
: Q2$(1)=HEX(0D)
: IF Q3=0 THEN 1290
: IF Q4<>0 THEN 1250
: Q4=81-Q2
1250 FOR Q5=1 TO Q3
: IF Q5<>1 THEN '270
: $GIOROW/005(4006A000,Q4$())Q1$(<1,Q1>
: GOTO 1275

1270 $GIOROW/005(4006A000,Q4$())Q1$(<2,1>
1275 $GIDCOL/005(A000,Q4$())Q2$(<1,Q2>
: $GIDERS/005(A000,Q4$())Q3$(<1,Q4>
: NEXT Q5
1290 $GIOROW/005(4006A000,Q4$())Q1$(<1,Q1>
: $GIDCOL/005(A000,Q4$())Q2$(<1,Q2>
: RETURN

1355 DEFFN '033
: DIM E2$10
: E2$=HEX(00202000000000)
1370 KEYIN E2$,1370,1370
: STR(E2$,1,1)=HEX(FF)
1380 $GIOTSTON/215(12214100,E2$)
: $IF ON /215,1450
: SELECT PRINT 005(80)
: KEYIN E2$,1400,1380
1400 IF STR(E2$,1,1)<HEX(F0) THEN 1410
: PRINT HEX(0706)
1410 SELECT P2
: GOSUB '101(1,40,1,40)
: PRINT
: GOSUB '101(1,40,0,0)
: PRINT "PRINTER NOT READY..."

```

```

: SELECT P0
: GOTO 1380

1450 GOSUB '101(1,40,1,40)
: SELECT PRINT 215(132)
: RETURN

1700 Z### PROGRAM > KFAM INITIALIZATION (KFAM.INI) VERSION 1.0 2200 T/VP
2170 DIM R1$8,R2$8,R4$1
2180 DEFFN '180(R1$,R2$,R7,R8,R4$,R,R1,R2,R3,R4,R6,R5,R0)
: DIM Q6$64
: DIM V1$8
: DIM K1$8,U1$8
: DIM X$64,X1$2,I$(32)8,Z$8,N$8,H$2,L$1,A(15),R(15)
: R0=0
: U1$=R1$
: D2=R8
: MAT A=ZER
: MAT R=ZER
: GOSUB '193(R8,U1$)
2280 A(D2)=Y0(6)
: C2=Y0(1)
: IF Y0(1)>=0 THEN 2360
: GOSUB '032("USER FILE SCRATCHED"," ")
: R0=1
: GOTO 3430

2360 D1=R7
: K1$=R2$
: GOSUB '193(R7,K1$)
: A(D1)=Y0(6)
: C1=Y0(1)
: IF Y0(1)>=0 THEN 2470
: GOSUB '032("KEY FILE SCRATCHED"," ")
: R0=2
: GOTO 3430

2470 V1$=R4$
: U,U1,U2=1
: K3=0
: IF V1$="A" THEN 2580
: IF V1$="C" THEN 2580
: IF V1$="N" THEN 2770
: IF V1$="M" THEN 2690
: GOSUB '032("INVALID RECORD TYPE"," ")
: R0=3
: GOTO 3430

2580 U1=R1
: U2=R2
: X=U1*U2
: IF X<2 THEN 2650
: IF V1$="C" THEN 2640
: X=X+3
2640 IF X<=256 THEN 2770
2650 GOSUB '032("BLOCKING OR LENGTH INVALID"," ")
: R0=4
: GOTO 3430

2690 U=R

```



```

      : IF U<2 THEN 2730
      : IF U<=255 THEN 2770
2730 GOSUB '032("MULTI SCTR OUT OF RANGE", " ")
      : R0=5
      : GOTO 3430

2770 U3=R3
      : IF U3<1 THEN 2800
      : IF U3<=30 THEN 2840
2800 GOSUB '032("KEY LEN OUT OF RANGE", " ")
      : R0=6
      : GOTO 3430

2840 T5=U3+3
      : V6=INT(240/T5)
      : U4=R4
      : IF V1$>"C" THEN 2920
      : IF V1$="C" THEN 2910
      : IF U4<3 THEN 2940
2910 IF U3+U4>U1+2 THEN 2940
2920 IF U3+U4>256*U THEN 2940
      : IF INT(U4/256)=INT((U3+U4)/256) THEN 2980
2940 GOSUB '032("OVERLAPING KEY", " ")
      : R0=7
      : GOTO 3430

2980 IF C2=0 THEN 3000
      : GOSUB '185(U1$,R8)
3000 U6=R6
      : U8=INT(U6/U2+.999)
      : IF U<2 THEN 3040
      : U8=U8*U
3040 U8=U8+2
      : IF C2=0 THEN 3100
      : IF U8>A+2 THEN 3260
      : U8=A+2
      : GOTO 3110

3100 R(D2)=U8
3110 IF C1=0 THEN 3200
      : GOSUB '185(K1$,R7)
      : K2=A+2
      : X=INT(V6*.75)-1
      : K1=INT(U6/X)+5
      : IF K2>=K1 THEN 3240
      : K3=1
      : GOTO 3240

3200 X=INT(V6*.6)-1
      : K2=INT(U6/X)+5
      : R(D1)=R(D1)+K2
3240 IF R(D1)>A(D1) THEN 3260
      : IF R(D2)<=A(D1) THEN 3320
3260 GOSUB '032("FIL SIZ EXCDS AVBL SPACE", " ")
      : R0=8
      : GOTO 3430

3320 ON R5+1 GOTO 3430, 3360, 3360, 3400, 3410
3360 GOSUB '032("INVALID KFAM FILE TYPE", " ")
      : R0=9

```

```

: GOTO 3430

3400 GOSUB '181(R7,R8)
: GOTO 3430

3410 GOSUB '182(R7,R8)
: GOTO 3430

3430 RETURN

3530 DEFFN '181(R7,R8)
: DIM Q2$2,Q3$2,V5$1,V8$1,V0$2,V2$2,V3$2,V6$1,T2$2,T4$3
: DIM T5$30,T7$30,T2$(8)2,T(8),T8$1,Q9$2,Q0$(4)60
: IF C1>0THEN 3570
: DATA SAVE DC OPEN T$#R7,K2,K1$
: DSKIP #R7,K2-25
: DATA SAVE DC $R7,END
: DATA SAVE DC CLOSE #R7
3570 IF C2>0THEN 3580
: DATA SAVE DC OPEN T$#R8,U8,U1$
: DSKIP #R8,U8-25
: DATA SAVE DC $R8,END
: DATA SAVE DC CLOSE #R8
3580 Q2$=HEX(0000)
: BIN(STR(Q2$,2))=U-1
: XOR (Q2$,FF)
: Y=U8-3
: X=INT(Y/256)
: BIN(Q3$)=X
: BIN(STR(Q3$,2))=Y-256*X
: BIN(V8$)=U2
: V5$=V8$
: BIN(STR(V1$,2))=U1
: X=INT(U4/256)
: BIN(STR(V1$,3))=X
: BIN(STR(V1$,4))=U4-256*X
: BIN(STR(V1$,5))=U3
3590 BIN(STR(V1$,6))=V6
: V2$=HEX(0001)
: Y=K2-3
: X=INT(Y/256)
: BIN(V3$)=X
: BIN(STR(V3$,2))=Y-256*X
: BIN(V6$)=U
: T2$=HEX(0001)
: T0=1
: V8=.5
: T8$="X"
: DATA LOAD DC OPEN T#R7,K1$
3600 DATA SAVE DC $R7,Q2$,Q3$,V5$,V8$,V0$,V1$,V2$,V3$,V6$,T2$,T0,T1,T2,V8,T4$,T
5$,T7$,T2$(),T(),T8$
: INIT(FF)Q0$()
: INIT(00)STR(Q0$(1),1,U3)
: Q9$=HEX(0001)
: DATA SAVE DC $R7,Q9$,Q0$()
: LIMITS T#R8,U1$,X,Y,Z
: Y=Y-1
3610 Q0$(1)=HEX(A0FD)
: STR(Q0$(1),3)=Q2$
: STR(Q0$(1),5)=V5$

```

```

: STR(Q0$(1),6)=V8$
: STR(Q0$(1),7)=V1$
: STR(Q0$(1),15)=V6$
: DATA SAVE BA T##R8,(Y,Y)Q0$()
: RETURN

3730 DEFFN '182(R7,R8)
: DIM Q2$2,Q3$2,V5$(4)1,V8$1,V2$2,V3$2,V6$1,T2$2
: DIM T8$(4)1,Q0$4,V4$(4)2,V2$(4)2,Q9$2,Q0$(4)60
: IF C1>0THEN 3770
: DATA SAVE DC OPEN T##R7,K2,K1$
: DSKIP #R7,K2-25
: DATA SAVE DC #R7,END
: DATA SAVE DC CLOSE #R7
3770 IF C2>0THEN 3780
: DATA SAVE DC OPEN T##R8,U8,U1$
: DSKIP #R8,U8-25
: DATA SAVE DC #R8,END
: DATA SAVE DC CLOSE #R8
3780 Q2$=HEX(0000)
: BIN(STR(Q2$,2))=U-1
: XOR (Q2$,FF)
: INIT(FF)V2$()
: Y=U8-3
: X=INT(Y/256)
: BIN(Q3$)=X
: BIN(STR(Q3$,2))=Y-256*X
: BIN(V8$)=U2
: FOR X=1TO 4
: V5$(X)=V8$
: NEXT X
: BIN(STR(V1$,2))=U1
: X=INT(U4/256)
: BIN(STR(V1$,3))=X
3790 BIN(STR(V1$,4))=U4-256*X
: BIN(STR(V1$,5))=U3
: BIN(STR(V1$,6))=V6
: V2$=HEX(0001)
: Y=K2-3
: X=INT(Y/256)
: BIN(V3$)=X
: BIN(STR(V3$,2))=Y-256*X
: BIN(V6$)=U
: T2$=HEX(0001)
: T0=1
: Q0$=" "
: INIT(FF)V4$()
: INIT(5A)T8$()
: DATA LOAD DC OPEN T##R7,K1$
3800 DATA SAVE DC #R7,Q2$,Q3$,V5$(),V8$,V1$,V2$,V3$,V6$,T2$,T0,T8$(),Q0$,V4$(),
V2$()
: INIT(FF)Q0$()
: INIT(00)STR(Q0$(1),1,U3)
: Q9$=HEX(0001)
: DATA SAVE DC #R7,Q9$,Q0$()
: LIMITS T##R8,U1$,X,Y,Z
: Y=Y-1
: X1$=HEX(A0FD)
3810 X$=HEX(A002A002A001A001A008A001A002)
: $PACK(F=X$)Q0$()FROMX1$,Q2$,V5$(),V8$,V1$,V6$,V2$()

```



```

: DATA SAVE BA T#R8, (Y, Y)G0$()
: RETURN

3920 DEFFN '185(N$, Z)
: LIMITS T#Z, N$, X, Y, H
: H=H-2
: A=Y-X-1
: IF A=H THEN 3950
: DATA LOAD DC OPEN T#Z, N$
: IF H<0 THEN 3940
: DSKIP #Z, END
: DATA SAVE DC $#Z, N$
3940 DSKIP #Z, A-H-1S
: DATA SAVE DC $#Z, END
: DATA SAVE DC CLOSE #Z
3950 RETURN

4000 %## PROGRAM > FILE STATUS (FILE.UTL) VERSION 1.0 2200 T/VP
4280 DEFFN '193(Y6, Y5$)
: DIM Y1$(16), Y2$8, Y3$1, Y4$2
4300 DEFFN X(X)=VAL(STR(Y1$(Y4), X+1, 1))
4310 DEFFN Y(Y)=256*VAL(STR(Y1$(Y4), Y, 1))+FNX(Y)
: DATA LOAD BA T#Y6, (0, Y9)Y1$()
4340 % ELSE CONTINUE # : REM NO DISK ERROR
4350 AND (STR(Y1$(1), 2, 1), 7F)
: Y7=VAL(STR(Y1$(1), 2, 1))
: Y4=1
: AND (STR(Y1$(Y4), 3, 1), 7F)
: AND (STR(Y1$(Y4), 5, 1), 7F)
: Y0(6)=FNY(5)-FNY(3)
: Y2$=Y5$
: XOR (STR(Y2$, 2), Y2$)
: Y3$=STR(Y2$, 8, 1)
: Y4$=HEX(0000)
: ADDC(Y4$, Y3$)
: ADDC(Y4$, Y3$)
4470 ADDC(Y4$, Y3$)
: ADD(STR(Y4$, 1, 1), STR(Y4$, 2, 1))
: Y8=VAL(STR(Y4$, 1, 1))
: Y8=Y8-INT(Y8/Y7)*Y7
: Y0=Y8
4530 DATA LOAD BA T#Y6, (Y8, Y9)Y1$()
: Y4=0
: FOR Y3=1 TO 16
: IF Y8+Y3<>1 THEN 4590
: Y3=2
4590 Y3$=STR(Y1$(Y3), 1, 1)
: IF Y3$=HEX(00) THEN 4650
: IF Y3$=HEX(10) THEN 4630
: IF Y3$<>HEX(11) THEN 4660
4630 IF STR(Y1$(Y3), 9, 8)<>Y5$ THEN 4660
: Y4=Y3
4650 Y3=16
4660 NEXT Y3
: IF Y3$=HEX(00) THEN 4770
: IF Y4<>0 THEN 4770
: Y3$=HEX(00)
: Y8=Y8-1
: IF Y8=Y0 THEN 4770
: IF Y8>=0 THEN 4530

```

```

: Y8=Y7-1
: GOTO 4530

4770 IF Y3<>HEX(00)THEN 4810
: Y0(1)=0
: Y0(2)=0
: Y0(3)=0
: Y0(4)=0
: Y0(5)=0
: GOTO 5030

4810 IF Y3<>HEX(10)THEN 4860
: Y0(1)=1
: IF STR(Y1$(Y4),2,1)=HEX(80)THEN 4910
: Y0(1)=Y0(1)+1
4860 IF Y3<>HEX(11)THEN 4910
: Y0(1)=-1
: IF STR(Y1$(Y4),2,1)=HEX(80)THEN 4910
: Y0(1)=Y0(1)-1
4910 AND (STR(Y1$(Y4),3,1),7F)
: AND (STR(Y1$(Y4),5,1),7F)
: Y0(2)=FNY(3)
: Y0(3)=FNY(5)
: Y0(5)=(Y0(3)-Y0(2))-1
: DATA LOAD BA T#Y6,(Y0(3),Y9)Y1$()
: Y3=Y4
: Y4=1
: Y0(4)=FNY(2)
: Y4=Y3
: DATA LOAD BA T#Y6,(Y8,Y9)Y1$()
5030 RETURN

```

SCHEDULE System --- (SCH.BASE)

```

0100 Z## PROGRAM > SCHEDULE MANAGEMENT (SCH.BASE) VERSION 1.0 780530
0110 REM
0120 REM *****
0130 REM **
0140 REM ** THIS PROGRAM WILL ALLOW THE COURSE FILE DATA **
0150 REM ** BASE TO BE MAINTAINED. **
0160 REM **
0170 REM ** THE FOLLOWING FUNCTIONS ARE AVAILABLE ... **
0180 REM ** - DISPLAY A COURSE RECORD; **
0190 REM ** - DISPLAY ENTIRE COURSE FILE; **
0200 REM ** - CHANGE ANY FIELD IN A COURSE RECORD; **
0210 REM ** - ADD A COURSE TO THE COURSE FILE; **
0220 REM ** - DELETE A COURSE FROM THE COURSE FILE; **
0230 REM **
0240 REM *****
0250 REM **
0260 REM **
1000 PRINT HEX(030A0A)
1010 PRINT " CODE FUNCTION"
1020 PRINT " 1 - DISPLAY COURSE DATA ."
1030 PRINT " 2 - CHANGE COURSES ."
1040 PRINT " 3 - ADD COURSES ."
1050 PRINT " 4 - DELETE COURSES ."
1060 PRINT " 5 - DISPLAY COURSE FILE ."
1070 PRINT " 6 - END OF MAINTENANCE ."
1080 INPUT "ENTER FUNCTION CODE",Z9
1090 IF Z9<=0 THEN 1080
: IF Z9>=7 THEN 1080
1100 REM
1110 REM !DISPLAY !CHANGE !ADD !DELETE !
1120 REM !-----!-----!-----!-----!
1130 ON Z9 GOTO 2540 , 1390 , 1860 , 2310 ,
1220 , 1190
1140 REM !-----!
1150 REM !DISPLAY !END OF EDT!
1160 REM !WHOLE FILE! !
1170 REM
1180 REM
1190 LOAD DC T/B10,"SCH.SCH" 100,7999

1200 RETURN

1210 REM Z+

```


----- PRINT COURSE FILE -----

```

1220 PRINT HEX(03); "COURSES IN COURSE FILE (CDP / CIN / TITLE)"
1230 Z = 0
1240 GOSUB ' 230 (1,1,2,1,F1$)
1250 GOSUB ' 235 (1)
1260 Z=0
1270 IF Q$=" " THEN 1320
      : PRINT "NO COURSES IN COURSE FILE"
      : GOTO 1360

1280 GOSUB ' 237 (1)
1290 IF Q$="E" THEN 1360
1300 Z=Z+1
      : IF Z<=13 THEN 1320
1305 PRINT HEX(03); "COURSES IN COURSE FILE (CDP / CIN / TITLE)"
1310 INPUT "WANT TO CONTINUE DISPLAYING",Y$
      : IF Y$= "N" THEN 1370
      : Z=0
1320 DATA LOAD DC #2,C1$(),N1$(),T$(),I1$(),C2$(),W$(),R$(),H$(),M1$(),I9$(),E1
      $(),S1$(),S2$()
1330 PRINT USING 1340,C1$(Q),N1$(Q),T$(Q)
1340 Z ##### / ##### / #####
1350 GOTO 1280

1360 INPUT "END OF FILE... TOUCH 'RETURN' TO CONTINUE",Y$
1370 GOSUB ' 239(1)
      : GOTO 1000

1380 REM Z↑

```

----- CHANGE COURSE DATA -----

```

1390 GOSUB ' 230 (1,1,2,1,F1$)
1400 INPUT "ENTER CDP TO BE CHANGED",C1$
      : PRINT HEX(03)
1410 GOSUB ' 232 (1,0,C1$)
1420 IF Q$ = " " THEN 1430
      : PRINT HEX(07);"INVALID CDP .... RE-ENTER"
      : PRINT HEX(0C0C)
      : GOTO 1400

1430 PRINT HEX(01)
      : GOSUB 2620
1440 GOSUB ' 101 (9,1,0,0)
1450 PRINT USING 1770," "
      : PRINT USING 1780," "
      : PRINT USING 1790," "
1460 PRINT USING 1800," "
      : PRINT USING 1810," "
      : PRINT USING 1820," "
1490 INPUT " ENTER CHANGE KEY",K
1500 IF K<0 THEN 1510
      : IF K>12 THEN 1510
      : GOTO 1520

1510 PRINT "INVALID CHANGE KEY ... RE-ENTER"
      : GOTO 1490

1520 GOSUB ' 101 (9,1,7,0)
      : PRINT
      : ON K GOTO 1530,1560,1570,1580,1590,1600,1610,1670,1680,1690,1700,1740
1530 GOSUB ' 231(1,0,C1$)
1540 INPUT "ENTER NEW CDP",C1$(Q)
      : GOSUB ' 234(1,0,C1$(Q),0)
      : DSKIP #2, 15
      : IF Q$=" " THEN 1710
1550 PRINT HEX(07);"INVALID CDP .... RE-ENTER"
      : PRINT HEX(0C0C)
      : GOTO 1540

1560 INPUT "ENTER CIN",N1$(Q)
      : GOTO 1710

1570 INPUT "ENTER COURSE SHORT TITLE",T$(Q)
      : GOTO 1710

1580 INPUT "ENTER PLANNED INPUT",I1$(Q)
      : GOTO 1710

1590 INPUT "ENTER CLASS SIZE",C2$(Q)
      : GOTO 1710

1600 INPUT "ENTER COURSE LENGTH IN WEEKS ",W$(Q)
      : GOTO 1710

1610 INPUT "ENTER NO. OF PHASES",P
      : IF P<=0 THEN 1610
1620 FOR K=1 TO P
1630 GOSUB ' 101 (12,1,0,0)

```

```

      : PRINT "ENTER CONTACT (RATIO,HOURS) FOR PHASE ";K;
      : INPUT R$(Q,K),H$(Q,K)
1650 NEXT K
1660 IF P = 7 THEN 1710
      : P=P+1
      : FOR K1=P TO 7
      : R$(Q,K1)="0"
      : H$(Q,K1)="0"
      : NEXT K1
      : GOTO 1710

1670 INPUT "ENTER MAN POWER ALLOWANCES",M1$(Q)
      : GOTO 1710

1680 INPUT "ENTER OTHER INSTRUCTORS AVAILABLE",I9$(Q)
      : GOTO 1710

1690 INPUT "ENTER NO OF TRAINERS AVAILABLE",E1$(Q)
      : GOTO 1710

1700 INPUT "ENTER NO OF SUPPORT REQUIRED AND MPA",S1$(Q),S2$(Q)
      : GOTO 1710

1710 C1$ = C1$(Q)
1720 DBACKSPACE #2 , 15
1730 DATA SAVE DC #2,C1$(),N1$(),T$(),I1$(),C2$(),W$(),R$(),H$(),M1$(),I9$(),E1
      $(),S1$(),S2$()
      : GOTO 1410

1740 INPUT "WANT TO CHANGE ANOTHER COURSE",Y$
1750 IF Y$="Y" THEN 1400
1760 GOSUB ' 239 (1)
      : GOTO 1000

1770 Z#!KEY* DATA ITEM      !KEY* DATA ITEM      !KEY* DATA ITEM      !
1780 Z#!=====            !=====            !=====            !
1790 Z#! 1 * CDP              ! 2 * CIN              ! 3 * CRS TITLE        !
1800 Z#! 4 * PLANNED INPUT    ! 5 * CLASS SIZE       ! 6 * CRS LEN (WKS)    !
1810 Z#! 7 * CNTCT RAT/HRS    ! 8 * TRAINERS         ! 9 * MAN POWER        !
1820 Z#!10 * OTHER INSTRS     !11 * SUPPORT          !12 * NO CHANGES     !
1850 REM Z↑

```


----- ADD A COURSE -----

```

1860 GOSUB ' 230 (1,1,2,1,F1$)
1870 F5=0
1880 INPUT "ENTER CDP TO BE ADDED",C1$
1890 FOR I=1 TO 4
1900 IF STR(C1$,I,1) <=HEX(2F) THEN 1930
1910 IF STR(C1$,I,1) >=HEX(5B) THEN 1930
1920 IF STR(C1$,I,1) <=HEX(39) THEN 1940
      : IF STR(C1$,I,1)>=HEX(41) THEN 1940
1930 PRINT HEX(07);"INVALID CDP...REENTER"
      : PRINT HEX(0C0C)
      : GOTO 1880

1940 NEXT I
      : GOTO 1960

1950 PRINT HEX(07);"INVALID CDP...REENTER"
      : PRINT HEX(0C0C)
      : GOTO 1880

1960 GOSUB ' 232 (1,0,C1$)
1970 IF Q$ = "N" THEN 1980
      : PRINT HEX(07);"DUPLICATE CDP .... RE-ENTER"
      : PRINT HEX(0C0C)
      : GOTO 1880

1980 DATA LOAD DC OPEN T #3,"SCHE.DEL"
1990 DATA LOAD DC #3,D1$(),D1
2000 IF D1<>0 THEN 2020
2010 GOSUB '233 (1,0,C1$,0)
      : GOTO 2060

2020 T8$="1"
      : T4$=D1$(D1)
      : D1$(D1)="000"
      : D1=D1-1
2030 DBACKSPACE #3 ,BEG
2040 DATA SAVE DC #3,D1$(),D1
2050 GOSUB '234 (1,0,C1$,0)
2060 IF Q=1 THEN 2090
2070 DATA LOAD DC #2,C1$(),N1$(),T$(),I1$(),C2$(),W$(),R$(),H$(),M1$(),I9$(),E1
      $(),S1$(),S2$()
2080 DBACKSPACE #2, 1 S
2090 GOSUB 2140
2100 DATA SAVE DC #2,C1$(),N1$(),T$(),I1$(),C2$(),W$(),R$(),H$(),M1$(),I9$(),E1
      $(),S1$(),S2$()
2110 INPUT "WANT TO ADD ANOTHER COURSE",Y$
2120 IF Y$="Y" THEN 1880
2130 GOSUB ' 239 (1)
      : GOTO 1000

2140 INPUT "ENTER CIN",N1$(Q)
2150 INPUT "ENTER COURSE SHORT TITLE",T$(Q)
2160 C1$(Q)=C1$
2170 INPUT "ENTER PLANNED INPUT",I1$(Q)
2180 INPUT "ENTER CLASS SIZE",C2$(Q)
2190 INPUT "ENTER COURSE LENGTH IN WEEKS ",W$(Q)
2200 INPUT "ENTER NO. OF PHASES",P

```

```

      : IF P<=0 THEN 2200
2210  FOR K=1 TO P
2220  IF K<=1 THEN 2230
      : INPUT "ENTER LECTURE RATIO AND HOURS FOR NEXT PHASE (R,H)",R$(Q,K),H$(Q,K)

      : GOTO 2240

2230  INPUT "ENTER LECTURE RATIO AND HOURS FOR FIRST PHASE (R,H)",R$(Q,K),H$(Q,K)
      )
2240  NEXT K
2250  P=P+1
      : FOR K1=P TO 7
      : R$(Q,K1)="0"
      : H$(Q,K1)="0"
      : NEXT K1
2260  INPUT "ENTER MAN POWER ALLOWANCES",M1$(Q)
2270  INPUT "ENTER OTHER INSTRUCTORS AVAILABLE",I9$(Q)
2280  INPUT "ENTER NO OF TRAINERS AVAILABLE",E1$(Q)
2290  INPUT "ENTER NO OF SUPPORT REQUIRED AND MPA",S1$(Q),S2$(Q)
      : RETURN

2300  REM %↑

```

----- DELETE A COURSE -----

```

2310 GOSUB ' 230 (1,1,2,1,F1$)
: GOSUB ' 230 (2,4,5,1,F2$)
2320 F5=0
2330 INPUT "ENTER CDP TO BE DELETED",C1$
2340 GOSUB ' 231 (1,0,C1$)
2350 IF G$<>" " THEN 2360
: IF T8$<>"1" THEN 2360
: GOTO 2370

2360 PRINT "COURSE NOT FOUND"
: INPUT "HIT RETURN KEY TO CONTINUE",Y$
: GOTO 2520

2370 DATA LOAD DC OPEN T #3,"SCHE.DEL"
2380 DATA LOAD DC #3,D1$(),D1
2390 D1=D1+1
: IF D1>=250 THEN 2420
: D1$(D1)=T4$
2400 DBACKSPACE #3 ,BEG
2410 DATA SAVE DC #3,D1$(),D1
2412 REM **
2420 GOSUB ' 231 (2,0,C1$)
: IF G$<>" " THEN 2500
2425 DATA LOAD DC OPEN T #6, "PLOT.DEL"
: DATA LOAD DC #6, D1$(),D1
2430 D1 = D1 + 1
: IF D1 > 250 THEN 2500
: D1$(D1) = T4$
: DBACKSPACE #6, BEG
: DATA SAVE DC #6, D1$(),D1
2440 REM **
2500 INPUT "DELETE ANOTHER COURSE",Y$
2510 IF Y$="Y" THEN 2330
2520 GOSUB ' 239 (1)
: GOSUB ' 239 (2)
: GOTO 1000

2530 REM %↑

```


----- DISPLAY COURSE -----

```

2540 GOSUB / 230 (1,1,2,1,F1$)
2550 INPUT "ENTER CDP TO BE LOOKED UP",C1$
2560 GOSUB / 232 (1,0,C1$)
2570 IF Q$ = " " THEN 2580
      : PRINT HEX(07); "INVALID CDP .... RE-ENTER"
      : PRINT HEX(0C0C)
      : GOTO 2550

2580 PRINT HEX(03)
      : GOSUB 2620
2590 INPUT "WANT TO LOOK UP ANOTHER COURSE",Y$
2600 IF Y$="Y" THEN 2550
2610 GOSUB / 239 (1)
      : GOTO 1000

2620 DATA LOAD DC #2,C1$( ),N1$( ),T$( ),I1$( ),C2$( ),W$( ),R$( ),H$( ),M1$( ),I9$( ),E1
      $( ),S1$( ),S2$( )
2630 PRINT USING 2720,T$(Q)
2640 PRINT
2650 PRINT USING 2730,C1$(Q),N1$(Q),T$(Q)
2660 PRINT USING 2740,I1$(Q),C2$(Q),W$(Q)
2670 PRINT USING 2750,M1$(Q),I9$(Q),E1$(Q)
2680 PRINT USING 2760,S1$(Q),S2$(Q)
2690 PRINT USING 2770,R$(Q,1),H$(Q,1),R$(Q,2),H$(Q,2),R$(Q,3),H$(Q,3),R$(Q,4),H$(
      Q,4)
2700 PRINT USING 2780,R$(Q,5),H$(Q,5),R$(Q,6),H$(Q,6),R$(Q,7),H$(Q,7)
2710 RETURN

2720 % Data Display for Course ---> #####
2730 % CDP/CIN ..... ####/##### TITLE ... #####
2740 % PLANNED INPUT .... ##### CLASS SIZE .. ### LENGTH(WKS) .. ###
2750 %MANPOWER ALLOWANCES ### INSTRUCTORS . ### TRAINERS ..... ###
2760 % SUPPORT ..... ### MPA ..... ###
2770 %CONTACT RATIO/HOURS (###/#####),(###/#####),(###/#####),(###/#####)
2780 % (###/#####),(###/#####),(###/#####)
2799 REM %↑

```

----- CURSOR CONTROL -----

```

2800 DEFFN '101(Q1,Q2,Q3,Q4)
: DIM Q1$(16),Q2$(64),Q3$(64),Q4$(10)
: INIT(0A)Q1$()
: INIT(09)Q2$()
: INIT(20)Q3$()
: Q1$(1)=HEX(01)
: Q2$(1)=HEX(0D)
: IF Q3=0 THEN 2840
: IF Q4<>0 THEN 2810
: Q4=65-Q2
2810 FOR Q5=1 TO Q3
: IF Q5<>1 THEN 2820
: $GIORDW/005(A000,Q4$())Q1$(<1,Q1>
: GOTO 2830

2820 $GIORDW/005(A000,Q4$())Q1$(<2,1>
2830 $GIORDW/005(A000,Q4$())Q2$(<1,Q2>
: $GIORDW/005(A000,Q4$())Q3$(<1,Q4>
: NEXT Q5
2840 $GIORDW/005(A000,Q4$())Q1$(<1,Q1>
: $GIORDW/005(A000,Q4$())Q2$(<1,Q2>
: RETURN

```

SCHEDULE System --- (SCH.SCH)

100 *** PROGRAM > SCHEDULE MANAGEMENT (SCH.SCH) VERSION 1.0 780530

110 REM

120 REM

130 REM

140 REM

150 REM

160 REM

170 REM

180 REM

190 REM

200 REM

210 REM

220 REM

230 REM

240 REM

250 REM

260 REM

270 REM **

This Program will create and maintain a Training Schedule
Mangement Data Base.

280 REM **

290 REM **

300 REM **

This Program will allow compute and display a course
schedule work sheet and plot schedule chart.

310 REM **

320 REM **

330 REM **

Description of Variables >

340 REM **

350 REM **

360 REM **

370 REM **

380 REM **

390 REM **

400 REM **

410 REM **

420 REM **

430 REM **

440 REM **

450 REM **

460 REM **

470 REM **

480 REM **

490 REM **

500 REM **

510 REM

520 REM

530 REM

540 REM

DISPLAY PROGRAM CONTROL MENU

1000

SELECT PRINT 005(80)

: REM SEL SYSTEM OUTPUT DEVICE

1010

SELECT INPUT 001

: REM SEL SYSTEM INPUT DEVICE

1020 REM

1030

PRINT HEX(030A0A)

1040

PRINT "NAVAL TRAINING SCHEDULE MANAGEMENT"

1050

PRINT " THIS SYSTEM WILL PERFORM THE FOLLOWING"

1060

PRINT " 1. MAINTAIN DATA BASE"

1070

PRINT " 2. SCHEDULE FEASIBILITY STUDY"

1080

PRINT " 3. PRINT SCHEDULE WORK SHEET"

1090

PRINT " 4. PLOT SCHEDULE"

1100

PRINT " 5. EDIT SCHEDULE"

1110

PRINT HEX(0A0A0A)

1120

PRINT " END (14,1,0,0)"


```

1130 PRINT "ENTER CODE FOR DESIRED FUNCTION ---> ";
1140 INPUT AB
1150 IF AB<=0 THEN 1030: IF AB>=6 THEN 1030
1160 REM
1170 REM !MAINTANCE !FEASIBILT!WORKSHEET !PLT SCHED !EDIT SCHED!
1180 REM !-----!-----!-----!-----!-----!
1190 ON AB GOTO 1240, 1250, 1260, 1270, 1280
1200 REM !-----!-----!-----!-----!-----!
1210 REM **
1220 REM ** LOAD OVERLAY....
1230 REM **
1240 GOSUB ' 102 ("DATA BASE MAINTENANCE","SCH.BASE","1000")
1250 GOSUB ' 102 ("FEASIBILITY STUDY","SCH.FEAS","1000")
1260 GOSUB ' 102 ("SCHEDULE WORK SHEET","SCH.WSHT","1000")
1270 GOSUB ' 102 ("PLOT SCHEDULE","SCH.PLOT","2000")
1280 GOSUB ' 102 ("EDIT SCHEDULE","SCH.EDIT","1000")
1290 REM **
1300 REM **
1310 END
1320 REM **
1330 REM **
1340 REM ** Subroutine > ***** 1 0 1
1350 REM ** Function >
1360 REM ** Cursor Address and Screen erase
1370 REM ** Variables >
1380 REM ** Q1 > row position ! Q3 > lines to erase
1390 REM ** Q2 > col position ! Q4 > char(s) erase
1400 REM ** Q1%() > cursor vert ! Q2%() > cursor horz
1410 REM ** Q3%() > line ers ! Q4%() > micro cmdn bufr
1420 REM ** Q5 > line indx !
1430 REM **
1440 REM *****
1450 DEFFN ' 101 (Q1,Q2,Q3,Q4) : REM ENTRY
1460 DIM Q1%(16),Q2%(64),Q3%(64),Q4%(10) : REM DEFIN BUFERS
1470 INIT(OA) Q1%() : REM INIT ROW BUFR
1480 INIT(O9) Q2%() : REM INIT COL BUFR
1490 INIT(20) Q3%() : REM INIT ERS BUFR
1500 Q1%(1) = HEX(01) : REM INIT TOP OF PAGE
1510 Q2%(1) = HEX(0D) : REM INIT BEG OF LINE
1520 IF Q3 = 0 THEN 1630 : REM NO LINES TO ERASE
1530 IF Q4 <> 0 THEN 1550 : REM CHARS TO ERASE
1540 Q4 = 65 - Q2 : REM ERS REST OF LINE
1550 FOR Q5 = 1 TO Q3 : REM ERASE Q3 LINES
1560 IF Q5 <> 1 THEN 1590 : REM NOT FIRST LINE
1570 $GID ROW/005 (A000,Q4%())Q1%(<1,Q1> : REM POSITION CURSOR
1580 GOTO 1600 : REM BRANCH TO POS COL
1590 $GID ROW/005 (A000,Q4%())Q1%(<2,1> : REM POSITION CURSOR LF
1600 $GID COL/005 (A000,Q4%())Q2%(<1,Q2> : REM POSITION CURSOR
1610 $GID ERS/005 (A000,Q4%())Q3%(<1,Q4> : REM ERASE TEXT
1620 NEXT Q5 : REM CONT ERASING
1630 $GID ROW/005 (A000,Q4%())Q1%(<1,Q1> : REM REPOSITION CURSOR
1640 $GID COL/005 (A000,Q4%())Q2%(<1,Q2> : REM REPOSITION CURSOR
1650 RETURN : REM EXIT
1660 REM **
1670 REM **
1680 REM ** Subroutine > ***** 1 0 2
1690 REM ** Function >
1700 REM ** To load an over lay Routine.
1710 REM ** Variables >
1720 REM ** Q6%() > description ! Q7%() program name

```

```

1730 REM **      QB$( ) > begin line number !
1740 REM **
1750 REM *****
1760 DEFFN ' 102  (Q6$,Q7$,QB$) : REM ENTRY
1770 PRINT      HEX(030A0A) : REM CLEAR CRT
1780 PRINTUSING 1910," " : REM PRINT TITLE
1790 PRINT      HEX(0A0A0A) : REM FOR TC USE
1800 GOSUB ' 101  (08,06,0,0) : REM POSITION CURSOR
1810 PRINT      "System Now Loading > ";Q6$;" Routine" : REM PRINT DESCRIPTION
1820 PRINT      HEX(0A0A0A) : REM FOR TC USE
1830 GOSUB ' 101  (10,06,0,0) : REM POSITION CURSOR
1840 PRINT      "Program being Loaded > ";Q7$ : REM PRINT PROG NAME
1850 IF      QB$ <> "1000" THEN 1870 : REM NOT PROGRAM OVRLY
1860 LOAD      DC T #0, Q7$ 100 , 7999
1870 IF      QB$ <> "2000" THEN 1890 : REM NOT PLOT OVRLY
1880 LOAD      DC T #0, Q7$ 100
1890 LOAD      DC T #0, Q7$ 8000
1900 RETURN : REM EXIT
1910 % T R A I N I N G   S C H E D U L E   P R O G R A M #

```

SCHEDULE System --- (PLOT.001)

```

6000 *** PROGRAM > TRAINING SCHEDULE (PLOT.001) VERSION 1.0 780707
6010 REM **
6020 REM **
6030      GOSUB ' 255                                : REM RETURN TO PLOT.PLT
6040 REM **
6050 REM **
6060 REM **
6070 REM ** Subroutine ***** 2 0 8
6080 REM ** Abstract >
6090 REM ** This routine will print the header on the output device.
6100 REM ** Variables >
6110 REM **
6120 REM **
6130 REM *****
6140 DEFFN ' 208                                : REM ENTRY
6150 PRINT      HEX(OC)                          : REM TOP OF FORM
6170      P1 = P1 + 1                            : REM INCR PAGE INDX
6180 PRINTUSING 6230,C1$(D8),N1$(D8),P1          : REM PRINT HEADER
6190 PRINTUSING 6232,T$(D8)
6200 PRINTUSING 6240,D2(1),D2(2)
6202 PRINTUSING 6242,I1(1),C2(1)
6210 PRINT
6220 RETURN                                : REM EXIT
6230 % CDP / CIN ..... >      **** / *****      ****
6232 % CRS TITLE ..... >      **** / *****
6240 % CRS LEN (WKS/DAYS) . >   **** / ****
6242 % INPUTS / CLASS SIZ . >  **** / ****
6250 REM **
6260 REM **
6270 REM ** Subroutine ***** 2 0 5
6280 REM ** Abstract >
6290 REM ** This routine will plot schedule according to format 1.
6300 REM ** Variables >
6310 REM **
6320 REM **
6330 REM *****
6340 DEFFN ' 205                                : REM ENTRY
6350 SELECT PRINT 215(132)
6360 GOSUB ' 208                                : REM PRNT HEADER
6370 PRINTUSING 6690," "                        : REM PRNT COL HEADINGS
6380 PRINTUSING 6700," "
6390 PRINTUSING 6710," "
6400 REM **
6410 FOR      I = 1 TO 50                      : REM BEG WEEK LOOP
6420 REM **
6430 CONVERT   STR(WO$(I,1),1,4) TO C1          : REM CONVERT YEAR (S)
6440 CONVERT   STR(WO$(I,1),5,2) TO C2          : REM CONVERT MONTH (S)
6450 CONVERT   STR(WO$(I,1),7,2) TO C3          : REM CONVERT DAY (S)
6460 CONVERT   STR(WO$(I,2),1,4) TO C4          : REM CONVERT YEAR (E)
6470 CONVERT   STR(WO$(I,2),5,2) TO C5          : REM CONVERT MONTH (E)
6480 CONVERT   STR(WO$(I,2),7,2) TO C6          : REM CONVERT DAY (E)
6490 REM **
6500 PRINTUSING 6720, I,M$(C2),C3,M$(C5),C6;    : REM BEG/END OF WEEK
6510 REM **
6520 FOR      J = 1 TO 2                      : REM BEG SESSN LOOP
6530 REM **
6540 CONVERT   STR(WO$(I+D2(1)-1,2),1,4) TO C7  : REM CONVERT YEAR (CE)
6550 CONVERT   STR(WO$(I+D2(1)-1,2),5,2) TO C8  : REM CONVERT MONTH (CE)

```



```

6560 CONVERT      STR(WD*(I+D2(1)-1,2),7,2) TO C9      : REM CONVERT DAY   (CE)
6570 REM **
6580 IF           SB*(J,I) = "0"                      THEN 6620 : REM NO SESSION SCHED
6590 PRINTUSING 6730,M*(C2),C3,M*(C8),C9;              : REM  BEG/END OF SESN
6600 GOTO        6640
6610 REM **
6620 PRINT      "  -----";                          : REM PRINT FILLER
6630 REM **
6640 NEXT        J                                      : REM CONT SESSN LOOP
6650 PRINT                                             : REM PRINT BUFFER
6660 NEXT        I                                      : REM CONT WEEK LOOP
6670 SELECT PRINT 005(80)
6680 RETURN                                           : REM EXIT
6690 Z#WEEK
6700 Z#          BEG  -  END      DAY SESSION 1      DAY SESSION 2
6710 Z#=====  BEG  -  END      BEG  -  END
6720 Z ##  ###  ## -  ###  ##  =====
6730 Z   ###  ## -  ###  ##

```

SCHEDULE System --- (SCH.PLOT)

```

100 X## PROGRAM > TRAINING SCHEDULE MANAGEMENT (SCH.PLOT) VERSION 1.0 780530
110 REM **
120 REM *****
130 REM **
140 REM ** THIS PROGRAM WILL GERERATE A HOLIADAY TABLE BASED ON **
150 REM ** THE MILITARY TRAINING CALENDAR. **
160 REM **
170 REM ** THIS PROGRAM OBSERVES THE FOLLOWING HOLIDAYS - **
180 REM ** JAN 1 - NEW YEARS DAY - FLOATING **
190 REM ** - WASH/LINCON BIRTH - 3RD MONDAY OF FEB **
200 REM ** - MEMORIAL DAY - LAST MONDAY OF MAY **
210 REM ** JUL 4 - 4TH OF JULY - FLOATING **
220 REM ** - LABOR DAY - 1ST MONDAY OF SEP **
230 REM ** - COLUMBUS DAY - 2ND MONDAY OF OCT **
240 REM ** NOV 11 - VETRANS DAY - FLOATING **
250 REM ** - TURKEY DAY - 4TH THURSDAY OF NOV **
260 REM ** DEC 25 - CHRISTMAS - FLOATING **
270 REM **
280 REM *****
290 REM **
300 REM **
1000 COM M$(12)3,H0$(27)8,W0$(120,2)9 : REM DEFINE COMM VARS
1010 DIM D$(7)3,N(7) : REM DEFINE PROG VARS
1020 DIM H4$(9)20,E$64,H8$1,H0(7) : REM DEFINE PROG VARS
1030 REM **
1040 GOSUB ' 050 : REM INITIALIZE TBLs
1050 REM **
1060 GOSUB ' 052 : REM ENTR STRT DATE
1070 REM **
1080 GOSUB ' 106 (Y2) : REM LOAD HOLIDAY TABLE
1090 GOSUB ' 107 : REM LIST HOLIDAYS
1100 GOSUB ' 108 : REM LOAD BEG/END TBL
1110 GOSUB ' 113 : REM LIST BEG/END TBL
1120 GOSUB ' 102 ("SCHEDULE PLOTING","PLOT.SEL","1000"); REM LOAD PLOT PROG
1130 END : REM END OF PROGRAM
1140 REM **
1150 REM **
1160 REM ** Subroutine ***** 0 3 2
1170 REM ** Abstract >
1180 REM ** This routine will display a fatal error message on
1190 REM ** console.
1200 REM ** Variables >
1210 REM ** E$ > error message
1220 REM **
1230 REM *****
1240 DEFFN ' 32 (E$) : REM ENTRY
1250 SELECT PRINT 005(64) : REM SEL CONSOLE DEV
1260 PRINT HEX(07) : REM PRINT TONE
1270 GOSUB ' 101 (16,1,0,0) : REM POSITION CUSOR
1280 PRINT " TOUCH 'RETURN' TO RESTART"; : REM PRINT RESTRT MSG
1290 SELECT P3 : REM SEL PAUSE
1300 GOSUB ' 101 (14,1,1,0) : REM POSITION CUSOR
1310 PRINT
1320 GOSUB ' 101 (14,1,0,0) : REM POSITION CUSOR
1330 PRINT E$ : REM PRINT MSG
1340 SELECT P0 : REM DE-SEL PAUSE
1350 KEYIN F1$,1370,1370 : GOTO 1290 : REM RESPONSE WAIT
1360 REM **

```

```

1370 PRINT HEX(030A0A0A) : REM CLR CONSOLE
1380 PRINT " EXECUTE THE FOLLOWING COMMANDS TO RESTART:"
1390 PRINT " :CLEAR V (RETURN(EXEC))"
1400 PRINT " :CLEAR P (RETURN(EXEC))"
1410 PRINT " :LOAD DCT #0, ":HEX(22);"SCHEDULE":HEX(22);"(RETURN(EXEC))"
1420 PRINT " :RUN (RETURN(EXEC))"
1430 PRINT HEX(0A0A)
1440 STOP "ENTER COMMANDS NOW...."
1450 RETURN : REM EXIT
1460 REM **
1470 REM **
1480 REM ** Subroutine ***** 0 5 0
1490 REM ** Abstract >
1500 REM ** This routine will initialize tables for the
1510 REM ** initialization of the schedule plotting routines
1520 REM ** Variables >
1530 REM ** M$( ) > table of month labels ! D$( ) > table of d-o-w labels
1540 REM ** H$( ) > table of holiday labels!
1550 REM **
1560 REM *****
1570 DEFFN / 050 : REM ENTRY
1580 MAT READ H0
1590 MAT READ D$
1600 MAT READ M$
1610 MAT READ H4$
1620 DATA 26,25,24,23,22,21,27
1630 REM **
1640 DATA "SUN","MON","TUE","WED","THU","FRI","SAT"
1650 REM **
1660 DATA "JAN","FEB","MAR","APR","MAY","JUN"
1670 DATA "JUL","AUG","SEP","OCT","NOV","DEC"
1680 REM **
1690 DATA "NEW YEAR","WASH/LICOLN BIRTH","MEMORIAL DAY"
1700 DATA "4TH OF JULY","LABOR DAY","COLUMBUS DAY"
1710 DATA "VETRANS DAY","TURKEY DAY","CHRISTMAS"
1720 REM **
1730 RETURN : REM EXIT
1740 REM **
1750 REM **
1760 REM ** Subroutine ***** 0 5 2
1770 REM ** Abstract >
1780 REM ** This routine will allow the user to enter the
1790 REM ** starting date of the schedule.
1800 REM ** Variables >
1810 REM ** M2 > starting month ! D2 > starting day
1820 REM ** Y2 > starting year !
1830 REM **
1840 REM *****
1850 DEFFN / 052 : REM ENTRY
1860 PRINT HEX(030A0A0A) : REM CLR CONSOLE
1870 GOSUB / 101 (2,35,3,6) : REM POSITION CURSOR
1880 GOSUB / 101 (2,1,0,0) : REM POSITION CURSOR
1890 PRINT "ENTER STARTING MONTH (1-12) " : REM ENTER STRT MTH
1900 INPUT M
1910 PRINT "ENTER STARTING DAY (1-31) " : REM ENTER STRT DAY
1920 INPUT D
1930 PRINT "ENTER STARTING YEAR (1581-????)" : REM ENTER STRT YEAR
1940 INPUT Y
1950 REM **
1960 IF M < 1 THEN 2020 : REM INVALID INPUT

```



```

1970 IF          M > 12          THEN 2020 : REM INVALID INPUT
1980 IF          D < 1          THEN 2020 : REM INVALID INPUT
1990 IF          D > 31          THEN 2020 : REM INVALID INPUT
2000 IF          Y > 1580       THEN 2070 : REM VALID DATE
2010 REM **
2020 PRINT        HEX(0A0A0A)    : REM FOR REMOTE USE
2030 GOSUB ' 101 (1,1,0,0)      : REM POSITION CURSOR
2040 PRINT        "INVALID DATE ..... RE-ENTER" : REM ERR MSG
2050 GOTO         1870          : REM RE-ENTER DATE
2060 REM **
2070             M2,M = INT(M)
2080             D2,D = INT(D)
2090             Y2,Y = INT(Y)
2100 REM **
2110 RETURN                          : REM EXIT
2120 REM **
2130 REM **
2140 REM ** Subroutine ***** 1 0 0
2150 REM ** Abstract >
2160 REM ** This routine will compute the day of week for the
2170 REM ** requested date. Will return invalid vaules for
2180 REM ** date out of range.
2190 REM ** Variables >
2200 REM ** M > lookup month      ! D > lookup day
2210 REM ** Y > lookup year       ! M1 > temp month bufr
2220 REM ** Y1 > temp year bufr  ! N > day of week indx
2230 REM **
2240 REM *****
2250 DEFFN ' 100 (M,D,Y,N,E$)      : REM ENTRY
2260             M1 = M            : REM STOR LOKUP MTH
2270             Y1 = Y            : REM STOR LOKUP YEAR
2280 IF          M > 2              THEN 2310: REM
2290             M = M + 12        : REM INCR MONTH
2300             Y = Y - 1        : REM DCRMNT YEAR
2310             N = D + INT((13*M+3)/5)+INT(5*Y/4)
                - INT(Y/100)+INT(Y/400) + 1 : REM CMPTUE D-D-W
2320             N = N - INT(N/7)*7 : REM
2330 REM **
2340             M = M1 + 1        : REM INCR MONTH
2350             Y = Y1            : REM RESET YEAR
2360 IF          M > 2              THEN 2390 : REM
2370             M = M + 12        : REM INCR MONTH
2380             Y = Y - 1        : REM DCRMNT YEAR
2390             M = INT((13*M+3)/5) + INT(5*Y/4) -
                INT(Y/100) + INT(Y/400) + 2 : REM CMPLUT NEW MTH
2400             M = M - INT(M/7)*7 : REM
2410 REM **
2420             E$ = " "         : REM CLR ERR FLAG
2430 IF          D < 29          THEN 2530 : REM VALID DATE
2440 IF          D > 31          THEN 2520 : REM INVALID DATE
2450 IF          M1 = 2          THEN 2500 : REM FEBUARY EXECPTN
2460 IF          M1 > 7          THEN 2480 : REM TST 31 DAYS
2470 IF          M1-INT(M1/2)*2 = 1 THEN 2530 : REM VALID DATE: REM
                ELSE:           GOTO 2490 : REM ELS TST 30 DAYS
2480 IF          M1-INT(M1/2)*2 = 0 THEN 2530 : REM VALID DATE: REM
                ELSE:           GOTO 2490 : REM ELS TST 30 DAYS
2490 IF          D > 30          THEN 2520 : REM INVALID DATE: REM
                ELSE:           GOTO 2530 : REM ELS VALID DATE
2500 IF          M-N > 4          THEN 2520 : REM INVALID DATE
2510 IF          M-N >= 1        THEN 2530 : REM VALID DATE

```

```

2520          E$ = "ERR"                                : REM SET ERR FLG
2530 RETURN                                             : REM EXIT
2540 REM **
2550 REM **
2560 REM ** Subroutine > ***** 1 0 1
2570 REM ** Function >
2580 REM **      Cusor Address and Screen erase
2590 REM ** Variables >
2600 REM **      Q1 > row position ! Q3 > lines to erase
2610 REM **      Q2 > col position ! Q4 > char(s) erase
2620 REM **      Q1$() > cusor vert ! Q2$() > cusor horz
2630 REM **      Q3$() > line ers ! Q4$() > micro cmd bufr
2640 REM **      Q5 > line indx !
2650 REM **
2660 REM *****
2670 DEFFN ' 101 (Q1,Q2,Q3,Q4) : REM ENTRY
2680 DIM Q1$(16),Q2$(64),Q3$(64),Q4$(10) : REM DEFIN BUFERS
2690 INIT(0A) Q1$() : REM INIT ROW BUFR
2700 INIT(09) Q2$() : REM INIT COL BUFR
2710 INIT(20) Q3$() : REM INIT ERS BUFR
2720 Q1$(1) = HEX(01) : REM INIT TOP OF PAGE
2730 Q2$(1) = HEX(00) : REM INIT BEG OF LINE
2740 IF Q3 = 0 THEN 2850 : REM NO LINES TO ERASE
2750 IF Q4 <> 0 THEN 2770 : REM CHARS TO ERASE
2760 Q4 = 65 - Q2 : REM ERS REST OF LINE
2770 FOR Q5 = 1 TO Q3 : REM ERS Q3 LINES
2780 IF Q5 <> 1 THEN 2810 : REM NOT FIRST LINE
2790 $GIO ROW/005 (A000,Q4$())Q1$(<1,Q1> : REM POSITION CUSOR
2800 GOTO 2820 : REM BRANCH TO POS COL
2810 $GIO ROW/005 (A000,Q4$())Q1$(<2,1> : REM POSITION CUSOR LF
2820 $GIO COL/005 (A000,Q4$())Q2$(<1,Q2> : REM POSITION CUSOR
2830 $GIO ERS/005 (A000,Q4$())Q3$(<1,Q4> : REM ERASE TEXT
2840 NEXT Q5 : REM CONT ERASING
2850 $GIO ROW/005 (A000,Q4$())Q1$(<1,Q1> : REM REPOSITION CUSOR
2860 $GIO COL/005 (A000,Q4$())Q2$(<1,Q2> : REM REPOSITION CUSOR
2870 RETURN : REM EXIT
2880 REM **
2890 REM **
2900 REM ** Subroutine > ***** 1 0 2
2910 REM ** Function >
2920 REM **      To load an over lay Routine.
2930 REM ** Variables >
2940 REM **      Q6$() > description ! Q7$() program name
2950 REM **      Q8$() > begin line number !
2960 REM **
2970 REM *****
2980 DEFFN ' 102 (Q6$,Q7$,Q8$) : REM ENTRY
2990 PRINT HEX(03) : REM CLEAR CRT
3000 PRINT USING 3090," " : REM PRINT TITLE
3010 GOSUB ' 101 (08,06,0,0) : REM POSITION CUSOR
3020 PRINT "System Now Loading > ";Q6$;" Routine" : REM PRINT DESCRIPTION
3030 GOSUB ' 101 (10,06,0,0) : REM POSITION CUSOR
3040 PRINT "Program being Loaded > ";Q7$ : REM PRINT PROG NAME
3050 IF Q8$ <> "1000" THEN 3070 : REM NOT PROGRAM OVRLY
3060 LOAD DC T #0, Q7$ 100 , 7999
3070 LOAD DC T #0, Q7$ 8000
3080 RETURN : REM EXIT
3090 % T R A I N I N G S C H E D U L E P R O G R A M #
3100 REM **
3110 REM **

```

```

3120 REM ** Subroutine ***** 1 0 3
3130 REM ** Abstract >
3140 REM ** This routine will load the holiday table for
3150 REM ** floating type holidays.
3160 REM ** Variables >
3170 REM ** C1 > month of holiday ! C2 > day of holiday
3180 REM ** C3 > year of holiday ! C4 > month of holiday <sat>
3190 REM ** C5 > day of holiday <sat> ! C6 > year of holiday <sat>
3200 REM ** C7 > month of holiday <sun> ! C8 > day of holiday <sun>
3210 REM ** C9 > year of holiday <sun> ! C0 > holiday index
3220 REM ** N > d-o-w index ! E$ > err return buffer
3230 REM **
3240 REM *****
3250 DEFFN ' 103 (C1,C2,C3,C4,C5,C6,C7,C8,C9,C0) : REM ENTRY
3260 GOSUB ' 100 (C1,C2,C3,N,E$) : REM LOOKUP HOLIDAY
3270 REM **
3280 IF E$ <> "ERR" THEN 3320 : REM NOT SYSTM ERR
3290 GOSUB ' 032 (" HOLIDAY LOOKUP ERROR <NO DATE>") : REM PRINT ERR
3300 GOTO 3480 : REM BRNCH TO EXIT
3310 REM **
3320 IF D$(N+1) <> "SAT" THEN 3380 : REM HOL NOT SAT
3330 CONVERT C4 TO STR(HO$(C0),5,2),(##) : REM SET MONTH
3340 CONVERT C5 TO STR(HO$(C0),7,2),(##) : REM SET DAY
3350 CONVERT C6 TO STR(HO$(C0),1,4),(####) : REM SET YEAR
3360 GOTO 3480 : REM BRNCH TO EXIT
3370 REM **
3380 IF D$(N+1) <> "SUN" THEN 3440 : REM HOL NOT SUN
3390 CONVERT C7 TO STR(HO$(C0),5,2),(##) : REM SET MONTH
3400 CONVERT C8 TO STR(HO$(C0),7,2),(##) : REM SET DAY
3410 CONVERT C9 TO STR(HO$(C0),1,4),(####) : REM SET YEAR
3420 GOTO 3480 : REM BRNCH TO EXIT
3430 REM **
3440 CONVERT C1 TO STR(HO$(C0),5,2),(##) : REM SET MONTH
3450 CONVERT C2 TO STR(HO$(C0),7,2),(##) : REM SET DAY
3460 CONVERT C3 TO STR(HO$(C0),1,4),(####) : REM SET YEAR
3470 REM **
3480 RETURN : REM EXIT
3490 REM **
3500 REM **
3510 REM ** Subroutine ***** 1 0 4
3520 REM ** Abstract >
3530 REM ** This routine will load holiday table for nth day
3540 REM ** type holidays.
3550 REM ** Variables >
3560 REM ** C1 > month of holiday ! C2 > nth day for holiday
3570 REM ** C3 > day of holiday <d-o-w> ! C4 > year of holiday
3580 REM ** C0 > holiday index ! E$ > err rtrun buffer
3590 REM ** N > d-o-w index ! C5 > day of holiday <index>
3600 REM **
3610 REM *****
3620 DEFFN ' 104 (C1,C2,C3,C4,C0) : REM ENTRY
3630 C5 = 0 : REM INIT DAY INDX
3640 MAT N = ZER : REM ZERO DOW COUNTS
3650 REM **
3660 C5 = C5 + 1 : REM INCR DAY INDX
3670 GOSUB ' 100 (C1,C5,C4,N,E$) : REM LOOKUP DATE
3680 REM **
3690 IF E$ <> "ERR" THEN 3730 : REM END OF MONTH
3700 GOSUB ' 032 (" HOLIDAY LOOKUP ERROR <EOM>") : REM DISPLAY ERR
3710 GOTO 3790 : REM BRNCH TO EXIT

```



```

3720 REM **
3730      N(N+1) = N(N+1) + 1      : REM INCR DOW INDX
3740 IF    N+1 <> C3      THEN 3660 : REM NOT LOKUP DAY
3750 IF    N(N+1) <> C2      THEN 3660 : REM NOT NTH DAY
3760      CONVERT C1 TO STR(HO$(CO),5,2),(##) : REM SET MONTH
3770      CONVERT C5 TO STR(HO$(CO),7,2),(##) : REM SET DAY
3780      CONVERT C4 TO STR(HO$(CO),1,4),(####) : REM STOR YEAR
3790 RETURN      : REM EXIT
3800 REM **
3810 REM **
3820 REM ** Subroutine ***** 1 0 5
3830 REM ** Abstract >
3840 REM **   This routine will load holiday table for last day
3850 REM **   type of holidays.
3860 REM ** Variables >
3870 REM **   C1 > month of holiday      ! C2 > day of holiday <d-o-w>
3880 REM **   C3 > year of holiday      ! C0 > holiday index
3890 REM **   C4 > day of holiday <indx> ! N > d-o-w index
3900 REM **   E$ > error return index    ! H() > holiday table
3910 REM **
3920 REM *****
3930 DEFFN ' 105 (C1,C2,C3,C0)      : REM ENTRY
3940      C4 = 32      : REM INIT DAY INDX
3950 REM **
3960      C4 = C4 - 1      : REM DCRMT DAY INDX
3970 GOSUB ' 100 (C1,C4,C3,N,E$)    : REM LOOKUP DATE
3980 REM **
3990 IF    E$ = "ERR"      THEN 3960 : REM NO SUCH DAY
4000 IF    N+1 <> C2      THEN 3960 : REM NOT LOOKUP DAY
4010      CONVERT C1 TO STR(HO$(CO),5,2),(##) : REM SET MONTH
4020      CONVERT C4 TO STR(HO$(CO),7,2),(##) : REM STOR DAY
4030      CONVERT C3 TO STR(HO$(CO),1,4),(####) : REM STOR YEAR
4040 RETURN      : REM EXIT
4050 REM **
4060 REM **
4070 REM ** Subroutine ***** 1 0 6
4080 REM ** Abstract >
4090 REM **   This routine will create the holiday lookup table.
4100 REM ** Variables >
4110 REM **
4120 REM **
4130 REM *****
4140 DEFFN ' 106 (B1)      : REM ENTRY
4150 REM **
4160 FOR    B2 = 0 TO 2      : REM BEG 3 YEAR LOOP
4170      B0 = B1 + B2      : REM CMPTUE CFY
4180      B3 = INT(B2*9)    : REM CMPTUE HOL INDX
4190 REM **
4200 GOSUB ' 103 (1,1,B0,12,31,B0-1,1,2,B0,1+B3) : REM NEW YEAR CFY
4210 GOSUB ' 104 (2,3,2,B0,2+B3) : REM WASH BIRTH CFY
4220 GOSUB ' 105 (5,2,B0,3+B3) : REM MEM DAY CFY
4230 GOSUB ' 103 (7,4,B0,7,3,B0,7,5,B0,4+B3) : REM 4TH JULY CFY
4240 GOSUB ' 104 (9,1,2,B0,5+B3) : REM LABOR DAY CFY
4250 GOSUB ' 104 (10,2,2,B0,6+B3) : REM CLMBUS DAY CFY
4260 GOSUB ' 103 (11,11,B0,11,10,B0,11,12,B0,7+B3) : REM VETRNS DAY CFY
4270 GOSUB ' 104 (11,4,5,B0,8+B3) : REM TURKEY DAY CFY
4280 GOSUB ' 103 (12,25,B0,12,27,B0,12,26,B0,9+B3) : REM CHRISTMAS CFY
4290 REM **
4300 NEXT    B2      : REM CONT 3 YEAR LOOP
4310 RETURN      : REM EXIT

```

```

4320 REM **
4330 REM **
4340 REM ** Subroutine ***** 1 0 7
4350 REM ** Abstract >
4360 REM ** This routine will list the table of holidays on
4370 REM ** the selected output device.
4380 REM ** Variables >
4390 REM ** H$() > table of holiday lables ! H0$() > table of holidays
4400 REM **
4410 REM *****
4420 DEFFN ' 107 : REM ENTRY
4430 PRINT HEX(030A0A0A) : REM CLR CRT SCREEN
4440 GOSUB ' 101 (8,1,0,0) : REM POSITION CURSOR
4450 PRINT "DO YOU WANT TO DISPLAY THE LIST OF HOLIDAYS (Y or N)";
4460 INPUT E$
4470 IF E$ <> "Y" THEN 4670 : REM DO NOT DISPLAY
4480 REM **
4490 SELECT PRINT 215(80) : REM SEL LPT DEVICE
4500 PRINT HEX(0C) : REM TOP OF FORM
4510 FOR J = 0 TO 2 : REM BEG 3 YEAR LOOP
4520 PRINT HEX(0A0A) : REM POSITION CURSOR
4530 PRINT USING 4680, Y2+J
4540 PRINT USING 4690
4550 FOR I = 1 TO 9
4560 K = I + INT(J*9) : REM CMPLTE HOL INDX
4570 CONVERT STR(H0$(K),5,2) TO C1
4580 CONVERT STR(H0$(K),7,2) TO C2
4590 CONVERT STR(H0$(K),1,4) TO C3
4600 PRINT USING 4700, M$(C1), C2, C3, H4$(I)
4610 NEXT I : REM CONT HOLIDAY LOOP
4620 PRINT USING 4690
4630 PRINT
4640 NEXT J : REM CONT 3 YEAR LOOP
4650 PRINT HEX(0C) : REM TOP OF FORM
4660 SELECT PRINT 005(80) : REM SEL CRT DEVICE
4670 RETURN : REM EXIT
4680 % ***** !
4690 %-----!
4700 %### - ## - #### *****!
4710 REM **
4720 REM **
4730 REM ** Subroutine ***** 1 0 8
4740 REM ** Abstract >
4750 REM ** This routine will load the begining/ending of week table.
4760 REM ** Variables >
4770 REM ** W$() > beg/end table
4780 REM **
4790 REM *****
4800 DEFFN ' 108 : REM ENTRY
4810 C1 = D2 - 1 : REM INIT DAY INDX
4820 C2 = M2 : REM INIT MTH INDX
4830 C3 = Y2 : REM INIT YEAR INDX
4840 C0 = 1 : REM INIT WEEK INDX
4850 REM **
4860 REM **
4870 REM ** FIND BEG OF WEEK
4880 C1 = C1 + 1 : REM INCR DAY INDX
4890 GOSUB ' 100 (C2,C1,C3,N,E$) : REM LOOKUP DATE
4900 REM **
4910 IF E$ <> "ERR" THEN 4990 : REM VALID DATE

```

```

4920          C2 = C2 + 1          : REM INCR MTH INDX
4930 IF      C2 < 13          THEN 4960 : REM VALID MONTH
4940          C3 = C3 + 1          : REM INCR YEAR INDX
4950          C2 = 1             : REM RESET MONTH INDX
4960          C1 = 0             : REM RESET DAY INDX
4970 GOTO    4880              : REM BRNCH TO NXT DAY
4980 REM **
4990 IF      N+1 <> 2          THEN 4880 : REM NOT A MONDAY
5000 GOSUB   ' 112 (C2,C1,C3,H8$) : REM X-MAS BREAK
5010 IF      H8$ <> "C"      THEN 5050 : REM CONT MONDAY CHK
5020          STR(WO$(C0-1,2),9,1) = H8$ : REM STOR HOL FLAG
5030 GOTO    4880              : REM CMPUTE NXT MON
5040 REM **
5050 GOSUB   ' 109 (C2,C1,C3,H8$) : REM TST FOR HOLIDAY
5060 IF      H8$ = " "      THEN 5090 : REM NOT A HOLIDAY
5070 GOSUB   ' 110 (C2,C1,C3) : REM INCR FOR MON HOL
5080 REM **
5090 CONVERT C2 TO STR(WO$(C0,1),5,2), (##) : REM STOR MONTH
5100 CONVERT C1 TO STR(WO$(C0,1),7,2), (##) : REM STOR DAY
5110 CONVERT C3 TO STR(WO$(C0,1),1,4), (####) : REM STOR YEAR
5120          STR(WO$(C0,1),9,1) = H8$ : REM STOR HOL FLAG
5130 REM **
5140 REM **
5150 REM **
5160          C1 = C1 + 1          : REM INCR DAY INDX
5170 GOSUB   ' 100 (C2,C1,C3,N,E$) : REM LOOKUP DATE
5180 REM **
5190 IF      E$ <> "ERR"      THEN 5270 : REM VALID DATE
5200          C2 = C2 + 1          : REM INCR MTH INDX
5210 IF      C2 < 13          THEN 5240 : REM VALID MONTH
5220          C3 = C3 + 1          : REM INCR YEAR INDX
5230          C2 = 1             : REM RESET MONTH INDX
5240          C1 = 0             : REM RESET DAY INDX
5250 GOTO    5160              : REM BRNCH TO NXT DAY
5260 REM **
5270 IF      N+1 <> 6          THEN 5160 : REM NOT A FRIDAY
5280 GOSUB   ' 109 (C2,C1,C3,H8$) : REM TST FOR HOLIDAY
5290 IF      H8$ = " "      THEN 5320 : REM NOT A HOLIDAY
5300 GOSUB   ' 111 (C2,C1,C3) : REM DCRMNT FOR FRI HOL
5310 REM **
5320 CONVERT C2 TO STR(WO$(C0,2),5,2), (##) : REM STOR MONTH
5330 CONVERT C1 TO STR(WO$(C0,2),7,2), (##) : REM STOR DAY
5340 CONVERT C3 TO STR(WO$(C0,2),1,4), (####) : REM STOR YEAR
5350          STR(WO$(C0,2),9,1) = H8$ : REM STOR HOL FLAG
5360 REM **
5370          C0 = C0 + 1          : REM INCR WEEK INDX
5380 IF      C0 <= 120        THEN 4880 : REM CONT COMPUTATIONS
5390 REM **
5400 RETURN
5410 REM **
5420 REM **
5430 REM ** Subroutine ***** 1 0 9
5440 REM ** Abstract >
5450 REM ** This routine will search the holiday table to test
5460 REM ** the current date for a holiday.
5470 REM ** Variables >
5480 REM **
5490 REM **
5500 REM *****
5510 DEFFN   ' 109 (C2,C1,C3,H8$) : REM ENTRY

```



```

5520 DIM          S$(1)8,S4$(1)2          : REM DEFINE SORT VARS
5530              H8$ = " "                : REM CLR HOL FLAG
5540 REM **
5550 CONVERT      C2 TO STR(S$(1),5,2), (##) : REM CNSTRUCT ID (MTH)
5560 CONVERT      C1 TO STR(S$(1),7,2), (##) : REM CNSTRUCT ID (DAY)
5570 CONVERT      C3 TO STR(S$(1),1,4), (####) : REM CNSTRUCT ID (YEAR)
5580 REM **
5590 MAT SEARCH   H0$( ), = S$(1) TO S4$( ) STEP 8 : REM SRCH HOLIDAY TBL
5600 IF           S4$(1) = HEX(0000)          THEN 5620 : REM NOT A HOLIDAY
5610              H8$ = "H"                  : REM SET HOL FLAG
5620 RETURN                                             : REM EXIT
5630 REM **
5640 REM **
5650 REM ** Subroutine ***** 1 1 0
5660 REM ** Abstract >
5670 REM ** This routine will increment the date to compensate
5680 REM ** for a monday holiday.
5690 REM ** Variables >
5700 REM ** C2 > current month                ! C1 > current day
5710 REM ** C3 > current year                  !
5720 REM **
5730 REM *****
5740 DEFFN / 110 (C2,C1,C3) : REM ENTRY
5750              C1 = C1 + 1 : REM INCR DAY INDX
5760 GOSUB / 100 (C2,C1,C3,N,E$) : REM LOOKUP DATE
5770 IF           E$ <> "ERR" THEN 5850 : REM VALID DATE
5780              C2 = C2 + 1 : REM INCR MTH INDX
5790 IF           C2 < 13 THEN 5820 : REM VALID MONTH
5800              C3 = C3 + 1 : REM INCR YEAR INDX
5810              C2 = 1 : REM RESET MTH INDX
5820              C1 = 1 : REM RESET DAY INDX
5830 GOTO         5760 : REM BRNCH TO LOOKUP
5840 REM **
5850 IF           N+1 = 3 THEN 5880 : REM NEW DAY IS TUE
5860 GOSUB / 032 (" MONDAY ADVANCE ERROR...") : REM DISPLAY ERR MSG
5870 REM **
5880 RETURN                                             : REM EXIT
5890 REM **
5900 REM **
5910 REM ** Subroutine ***** 1 1 1
5920 REM ** Abstract >
5930 REM ** This routine will decrement the date to compensate
5940 REM ** for a friday holiday.
5950 REM ** Variables >
5960 REM ** C2 > current month                ! C1 > current day
5970 REM ** C3 > current year
5980 REM **
5990 REM *****
6000 DEFFN / 111 (C2,C1,C3) : REM ENTRY
6010              C1 = C1 - 1 : REM DCRMENT DAY INDX
6020 GOSUB / 100 (C2,C1,C3,N,E$) : REM LOOKUP DATE
6030 IF           E$ <> "ERR" THEN 6140 : REM VALID DATE
6040              C2 = C2 - 1 : REM DCRMNT MTH INDX
6050 IF           C2 > 0 THEN 6080 : REM VALID MONTH INDX
6060              C3 = C3 - 1 : REM DCRMNT YEAR INDX
6070              C2 = 12 : REM RESET MTH INDX
6080              C1 = 31 : REM RESET DAY INDX
6090 GOSUB / 100 (C2,C1,C3,N,E$) : REM LOOKUP DATE
6100 IF           E$ <> "ERR" THEN 6140 : REM VALID DATE
6110              C1 = C1 - 1 : REM DCRMNT DAY INDX

```

AD-A071 576

RUTGERS - THE STATE UNIV PISCATAWAY NJ COLL OF ENGIN--ETC F/G 9/2
AUTOMATED COURSE SCHEDULING SYSTEM FOR NAVAL TRAINING, (U)
JUN 79 B W LIN, G W HODAK

UNCLASSIFIED

TAE6-72

N61339-78-C-0029

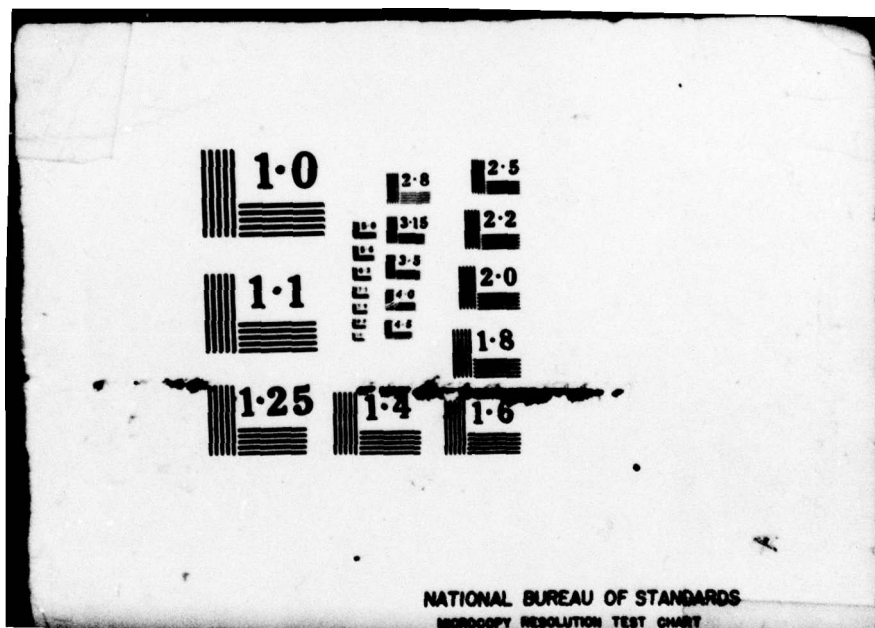
NL

2 OF 2
AD
A071576



END
DATE
FILMED

8-79
DDC




```

6120 GOTO          6090                      : REM BRNCH DATE LOOKUP
6130 REM **
6140 IF            N+1 = 5                    THEN 6170 : REM NEW DAY IS THUR
6150 GOSUB ' 032 ("FRIDAY HOLIDAY DECRMNT ERROR...") : REM PRINT ERR MSG
6160 REM **
6170 RETURN                      : REM EXIT
6180 REM **
6190 REM **
6200 REM ** Subroutine ***** 1 1 2
6210 REM ** Abstract >
6220 REM ** This routine will test if current week is the week
6230 REM ** of christmas.
6240 REM ** Variables >
6250 REM **
6260 REM **
6270 REM *****
6280 DEFFN ' 112 (C2,C1,C3,H8$)              : REM ENTRY
6290 H8$ = " "                               : REM CLR FLAG
6300 IF C2 <> 12                             THEN 6340 : REM NOT DECEMBER
6310 GOSUB ' 100 (12,25,C3,N,E$)           : REM CMPTUE DAY OF HOL
6320 IF C1 <> H0(N+1)                       THEN 6340 : REM NOT X-MAS WEEK
6330 H8$ = "C"                               : REM SET FLAG
6340 RETURN                                : REM EXIT
6350 REM **
6360 REM **
6370 REM ** Subroutine ***** 1 1 3
6380 REM ** Abstract >
6390 REM ** This routine will allow the user to list the table
6400 REM ** containing the begining and ending of schedule weeks.
6410 REM ** Variables >
6420 REM ** W0$() > beg/end table
6430 REM **
6440 REM *****
6450 DEFFN ' 113                             : REM ENTRY
6460 PRINT HEX(030A0A0A)                   : REM CLR CONSOLE
6470 GOSUB ' 101 (8,1,0,0)                 : REM POSITION CURSOR
6480 PRINT "DO YOU WANT TO DISPLAY THE BEGINING AND ENDING"
6490 PRINT "DAYS OF THE WEEK WHAT WERE GENERATED FOR THE "
6500 PRINT "SCHEDULE (Y or N)":
6510 INPUT E$                               : REM INPUT RESPONCE
6520 IF E$ <> "Y"                           THEN 6730 : REM DO NOT DISPLAY
6530 SELECT PRINT 215(80)                   : REM SEL LPT DEVICE
6540 PRINT HEX(0C)                          : REM TOP OF FORM
6550 PRINTUSING 6740
6560 PRINTUSING 6750
6570 REM **
6580 FOR I = 1 TO 120                       : REM BEGING WEEK LOOP
6590 CONVERT STR(W0$(I,1),1,4) TO C1         : REM FETCH STRT YEAR
6600 CONVERT STR(W0$(I,1),5,2) TO C2         : REM FETCH STRT MTH
6610 CONVERT STR(W0$(I,1),7,2) TO C3         : REM FETCH STRT DAY
6620 CONVERT STR(W0$(I,2),1,4) TO C4         : REM FETCH END YEAR
6630 CONVERT STR(W0$(I,2),5,2) TO C5         : REM FETCH END MTH
6640 CONVERT STR(W0$(I,2),7,2) TO C6         : REM FETCH END DAY
6650 PRINTUSING 6760, I,C3,M$(C2),C1,STR(W0$(I,1),9,1)
        , C6,M$(C5),C4,STR(W0$(I,2),9,1)
6660 IF I <> 60                             THEN 6700 : REM BRNCH TO NXT WEEK
6670 PRINT HEX(0C)                          : REM TOP OF FORM
6680 PRINTUSING 6740
6690 PRINTUSING 6750
6700 NEXT I                                : REM CONT WEEK LOOP

```

```

6710 SELECT          PRINT 005(80)          : REM SEL CRT DEVICE
6720 REM **
6730 RETURN          : REM EXIT
6740 % Week ! Start of Week ! End of Week !
6750 % -----!-----!-----!
6760 % *** ! ** - *** - **** # ! ** - *** - **** # !

```

SCHEDULE System --- (SCH.FEAS)

```

100 Z## PROGRAM >SCHEDULE MANAGEMENT (SCH.FEAS) VERSION 1.0 780530
110 REM
120 REM *****
130 REM **
140 REM ** THIS PROGRAM WILL ALLOW THE USER TO INTERACT **
150 REM ** WITH THE PROGRAM INABLING THE USER TO OPTIMISE **
160 REM ** THE CALCULATIONS FOR INSTRUCTOR COMP AND OTHER **
170 REM ** VALUE REQUIRED FOR CONSTRUCTION OF SCHEDULE. **
180 REM **
190 REM ** ***** IMPORTANT NOTE ***** **
200 REM **
210 REM ** ONCE THE DESIRED VALUES HAVE BEEN DETERMINED **
220 REM ** THE USER MUST CHANGE THE VALUES IN THE DATA **
230 REM ** BASE BY EXECUTING THE DATA MAINTENANCE OPTION **
240 REM ** OF THE SCHEDULE PROGRAMMING SYSTEM. **
250 REM **
260 REM ** THEREFORE, MAKE A NOTE OF VALUES CHANGED. **
270 REM **
280 REM *****
290 REM **
300 REM **
1000 DIM D0$(1)4,D2(2),D4$(1)18,D2$(1)5,D3$(4)64,P1$4: REM DEFINE WRK VARS
1005 F5 = 0 : REM INIT PRINT FLAG
1010 GOSUB ' 201 : REM INIT DISK FILES
1020 REM **
1030 GOSUB ' 203 : REM LOAD CRS RECORD
1040 IF G$ = "E" THEN 1100 : REM IF END OF LIST
1050 GOSUB ' 110 : REM CONV NUMERIC DATA
1060 GOSUB ' 111 : REM CMPT NESRY INFO
1070 GOSUB ' 205 : REM PRINT FEAS STUDY
1080 GOTO 1030 : REM DO NXT CRS
1090 REM **
1100 GOSUB ' 102 ("MAIN CONTROL","SCH.SCH","1000") : REM LOAD MAIN PROG
1110 RETURN : REM END OF ROUTINE
1120 REM **
1130 REM **
1140 REM ** Subroutine ***** 0 3 2
1150 REM ** Abstract >
1160 REM ** This routine will display a fatal error message on
1170 REM ** console.
1180 REM ** Variables >
1190 REM ** E$ > error message
1200 REM **
1210 REM *****
1220 DEFFN ' 32 (E$) : REM ENTRY
1230 SELECT PRINT 005(64) : REM SEL CONSOLE DEV
1240 PRINT HEX(07) : REM PRINT TONE
1250 GOSUB ' 101 (16,1,0,0) : REM POSITION CUSOR
1260 PRINT " TOUCH 'RETURN' TO RESTART"; : REM PRINT RESTRT MSG
1270 SELECT P3 : REM SEL PAUSE
1280 GOSUB ' 101 (14,1,1,0) : REM POSITION CUSOR
1290 PRINT
1300 GOSUB ' 101 (14,1,0,0) : REM POSITION CUSOR
1310 PRINT E$ : REM PRINT MSG
1320 SELECT P0 : REM DE-SEL PAUSE
1330 KEYIN F1$,1350,1350 : GOTO 1270 : REM RESPONSE WAIT
1340 REM **
1350 PRINT HEX(030A0A0A) : REM CLR CONSOLE

```



```

1360 PRINT " EXECUTE THE FOLLOWING COMMANDS TO RESTART:"
1370 PRINT " :CLEAR V (RETURN(EXEC))"
1380 PRINT " :CLEAR P (RETURN(EXEC))"
1390 PRINT " :LOAD DCT #0, ";HEX(22);"SCHEDULE";HEX(22);"(RETURN(EXEC))"
1400 PRINT " :RUN (RETURN(EXEC))"
1410 PRINT HEX(0A0A)
1420 STOP "ENTER COMMANDS NOW...."
1430 RETURN : REM EXIT
1440 REM **
1450 REM **
1460 REM ** Subroutine > ***** 1 0 1
1470 REM ** Function >
1480 REM ** Cursor Address and Screen erase
1490 REM ** Variables >
1500 REM ** Q1 > row position ! Q3 > lines to erase
1510 REM ** Q2 > col position ! Q4 > char(s) erase
1520 REM ** Q1%() > cursor vert ! Q2%() > cursor horz
1530 REM ** Q3%() > line ers ! Q4%() > micro cmdnd bufr
1540 REM ** Q5 > line indx !
1550 REM **
1560 REM *****
1570 DEFFN ' 101 (Q1,Q2,Q3,Q4) : REM ENTRY
1580 DIM Q1%(16),Q2%(64),Q3%(64),Q4%(10) : REM DEFIN BUFERS
1590 INIT(0A) Q1%() : REM INIT ROW BUFR
1600 INIT(09) Q2%() : REM INIT COL BUFR
1610 INIT(20) Q3%() : REM INIT ERS BUFR
1620 Q1%(1) = HEX(01) : REM INIT TOP OF PAGE
1630 Q2%(1) = HEX(0D) : REM INIT BEG OF LINE
1640 IF Q3 = 0 THEN 1750 : REM NO LINES TO ERASE
1650 IF Q4 <> 0 THEN 1670 : REM CHARS TO ERASE
1660 Q4 = 65 - Q2 : REM ERS REST OF LINE
1670 FOR Q5 = 1 TO Q3 : REM ERASE Q3 LINES
1680 IF Q5 <> 1 THEN 1710 : REM NOT FIRST LINE
1690 %GID ROW/005 (A000,Q4%())Q1%(<1,Q1> : REM POSITION CURSOR
1700 GOTO 1720 : REM BRANCH TO POS COL
1710 %GID ROW/005 (A000,Q4%())Q1%(<2,1> : REM POSITION CURSOR LF
1720 %GID COL/005 (A000,Q4%())Q2%(<1,Q2> : REM POSITION CURSOR
1730 %GID ERS/005 (A000,Q4%())Q3%(<1,Q4> : REM ERASE TEXT
1740 NEXT Q5 : REM CONT ERASING
1750 %GID ROW/005 (A000,Q4%())Q1%(<1,Q1> : REM REPOSITION CURSOR
1760 %GID COL/005 (A000,Q4%())Q2%(<1,Q2> : REM REPOSITION CURSOR
1770 RETURN : REM EXIT
1780 REM **
1790 REM **
1800 REM ** Subroutine > ***** 1 0 2
1810 REM ** Function >
1820 REM ** To load an over lay Routine.
1830 REM ** Variables >
1840 REM ** Q6%() > description ! Q7%() program name
1850 REM ** Q8%() > begin line number !
1860 REM **
1870 REM *****
1880 DEFFN ' 102 (Q6%,Q7%,Q8%) : REM ENTRY
1890 PRINT HEX(03) : REM CLEAR CRT
1900 PRINT USING 1990, " " : REM PRINT TITLE
1910 GOSUB ' 101 (Q8,Q6%,0,0) : REM POSITION CURSOR
1920 PRINT "System Now Loading > ";Q6%; " Routine" : REM PRINT DESCRIPTION
1930 GOSUB ' 101 (10,Q6%,0,0) : REM POSITION CURSOR
1940 PRINT "Program being Loaded > ";Q7% : REM PRINT PROG NAME
1950 IF Q8% <> "1000" THEN 1970 : REM NOT PROG OVRLY

```

```

1960 LOAD          DC T #0, Q7$ 100 , 7999
1970 LOAD          DC T #0, Q7$ 8000
1980 RETURN                                     : REM EXIT
1990 TRAINING SCHEDULE PROGRAM #
2000 REM **
2010 REM **
2020 REM ** Subroutine ***** 1 0 4
2030 REM ** Abstract >
2040 REM **   This routine will round a number to the specified
2050 REM **   number of decimal digits.
2060 REM ** Variables >
2070 REM **
2080 REM **
2090 REM *****
2100 DEFFN ' 104 (D1,D2,D3,A9)                  : REM ENTRY
2110          A9 = 1                             : REM INIT RETURN VALUE
2120 IF          D1 <= 1.00                      THEN 2220 : REM MINIMUM VALUE
2130 REM **
2140          D4 = D1 - INT(D1)                   : REM FTCH DECML VALUE
2150          D4 = INT(D4 * (10↑(D3+1)))          : REM MOVE DECML POINT
2160          D4 = ((D4/10) - INT(D4/10)) * 10    : REM FTCH CHK DIGIT
2170 REM **
2180          A9 = INT(D1*(10↑(D3)))*10↑(-D3)      : REM FTCH BASE VALUE
2190 REM **
2200 IF          D4 < D2                          THEN 2220 : REM ROUND DOWN
2210          A9 = A9 + ((1.0)*(10↑(-D3)))        : REM ROUND UP
2220 RETURN                                     : REM EXIT
2230 REM **
2240 REM **
2250 REM ** Subroutine ***** 1 1 0
2260 REM ** Abstract >
2270 REM **   This routine will convert input data record into
2280 REM **   numeric information for calculations.
2290 REM ** Variables >
2300 REM **
2310 REM **
2320 REM *****
2330 DEFFN ' 110
2340 CONVERT I1$(Q) TO I1(1)                   : REM ENTRY
2350 CONVERT C2$(Q) TO C2(1)                   : REM PLANNED INPUT
2360 CONVERT W$(Q) TO W(1)                     : REM CLAS SZ
2370 CONVERT M1$(Q) TO M1(1)                   : REM CRS LEN (WKS)
2380 CONVERT I9$(Q) TO I9(1)                   : REM MAN POWER
2390 CONVERT E1$(Q) TO E1(1)                   : REM INSTRUCTORS
2400 CONVERT S1$(Q) TO S1(1)                   : REM TRAINERS
2410 CONVERT S2$(Q) TO S2(1)                   : REM SUPPORT
2420 REM **
2430 FOR K = 1 TO 7                             : REM MPA
2440     CONVERT H$(Q,K) TO H(1,K)              : REM BEG CNTACT LOOP
2450     CONVERT R$(Q,K) TO R(1,K)              : REM CNTACT HRS
2460 NEXT K                                       : REM CNTACT RATIOS
2470 REM **                                     : REM CONT CNTACT LOOP
2480 RETURN                                     : REM EXIT
2490 REM **
2500 REM **
2510 REM ** Subroutine ***** 1 1 1
2520 REM ** Abstract >
2530 REM **   This routine will perform the following calucations -
2540 REM **   - Instructor/Comp ;
2550 REM **   - Course Length (days) ;

```



```

2560 REM **          - AOB (Average On Board) ;
2570 REM **          - Sum Type Hours (Inst/Comp)
2580 REM **          - Convening Frequency
2590 REM **          - Concurrent Scheduling
2600 REM ** Variables >
2610 REM **
2620 REM **
2630 REM *****
2640 DEFFN ' 111 : REM ENTRY
2650          W2(1) = W(1)*7-2 : REM CRS LEN (DAYS)
2660          O(1) = I1(1)*W2(1)/365 : REM AOB
2670          S(1) = 0 : REM INIT TYPE HRS
2680 REM **
2690 FOR      K = 1 TO 7 : REM BEG RATIO LOOP
2700          IF H(1,K) = 0 THEN 2760 : REM NO HRS
2710          IF R(1,K) = 0 THEN 2760 : REM NO RATIO
2720          A2 = C2(1)/R(1,K) : REM CMPLT QUOTA/RATIO
2730          GOSUB ' 104 (A2,5,1,A9) : REM ROUND QUOTA/RATIO
2740          GOSUB ' 104 (A9,3,0,A9) : REM ROUND QUOTA/RATIO
2750          S(1) = S(1) + (A9*H(1,K)) : REM SUM TYPE HOURS
2760 NEXT      K : REM CONT RATIO LOOP
2770 REM **
2780          A2 = I1(1)/C2(1) : REM NUM OF CLASSES
2790          GOSUB ' 104 (A2,1,0,A9) : REM ROUND TO WHOLE
2800          C4(1) = A9 : REM STOR # OF CLAS
2810          C5(1) = 50/C4(1) : REM
2820          C6(1) = W(1)/C5(1) : REM
2830          I2(1) = (S(1)*1.10) / (C5(1)*25*.88) : REM # OF INSTRUCTRS
2840          GOSUB ' 104 (I2(1),1,0,A9) : REM ROUND # OF INSTS
2850          I3(1) = A9 : REM STOR # OF INSTS
2860 REM **
2870          E2 = C6(1)/2 : REM
2880          E3(1) = INT(C6(1)/2) : REM
2890          IF E3(1)-E2 >= 0 THEN 2910 : REM
2900          E3(1) = E3(1) + 1 : REM
2910 RETURN : REM EXIT
2920 REM **
2930 REM **
2940 REM ** Subroutine ***** 1 5 0
2950 REM ** Abstract >
2960 REM ** This routine will select an ALL or PART mode for the
2970 REM ** generation of the schedule.
2980 REM ** Variables >
2990 REM **
3000 REM **
3010 REM *****
3020 DEFFN ' 150 : REM ENTRY
3030 GOSUB ' 151 (C1$) : REM ENTER CDP
3040 IF C1$ <> "STOP" THEN 3080 : REM NOT END OF LIST
3050          Q$ = "E" : REM SET EOF FLAG
3060 GOTO 3130 : REM BRANCH TO EXIT
3070 REM **
3080 GOSUB ' 232 (1,0,C1$) : REM FIND OLD CRS
3090 IF Q$ = " " THEN 3130 : REM BRNCH TO EXIT
3100 PRINT HEX(07);"RE-ENTER..." : REM INVALID CDP
3110 GOTO 3030 : REM RE-ENTER CDP
3120 REM **
3130 RETURN : REM EXIT
3140 REM **
3150 REM **

```



```

3160 REM ** Subroutine ***** 1 5 1
3170 REM ** Abstract >
3180 REM ** This routine will allow the CDP number to be
3190 REM ** input and verified for valid syntax.
3200 REM ** Variables >
3210 REM **
3220 REM **
3230 REM *****
3240 DEFFN ' 151 (C1$) : REM ENTRY
3250 GOSUB ' 101 (14,1,1,0) : REM POSITION CURSOR
3260 PRINT HEX(0A0A) : REM FOR TC USE
3270 GOSUB ' 101 (14,1,0,0) : REM POSITION CURSOR
3280 PRINT " Enter CDP Number (STOP to Stop) ---> " : REM PRNT PROMPT
3290 INPUT C1$ : REM INPUT RESPONSE
3300 REM **
3310 FOR I = 1 TO 4 : REM BEG SYNTAX LOOP
3320 IF STR(C1$,I,1) > "Z" THEN 3360 : REM INVALID CHAR
3330 IF STR(C1$,I,1) >= "A" THEN 3400 : REM VALID CHAR
3340 IF STR(C1$,I,1) > "9" THEN 3360 : REM INVALID CHAR
3350 IF STR(C1$,I,1) >= "0" THEN 3400 : REM VALID CHAR
3360 PRINT HEX(07); " RE-ENTER...." : REM INVALID CDP
3370 I = 4 : REM SET END OF LOOP
3380 NEXT I : REM TERMINATE LOOP
3390 GOTO 3270 : REM RE-ENTER CDP
3400 NEXT I : REM CONT SYNTAX LOOP
3410 REM **
3420 RETURN : REM EXIT
3430 REM **
3440 REM **
3450 REM ** Subroutine ***** 2 0 1
3460 REM ** Abstract >
3470 REM ** This routine will open the nessary KFAM files.
3480 REM ** Variables >
3490 REM **
3500 REM **
3510 REM *****
3520 DEFFN ' 201 : REM ENTRY
3530 PRINT HEX(030A0A0A) : REM CLR CONSOLE
3540 PRINT " MOUNT SCHEDULE DATA DISK...." : REM PRINT PROMPT
3550 PRINT HEX(0A0A0A) : REM FOR TC USE
3560 GOSUB ' 101 (8,1,0,0) : REM POSITION CURSOR
3570 PRINT " TOUCH 'RETURN' WHEN READY.."; : REM PRINT PROMPT
3580 INPUT A$ : REM RESPONSE WAIT
3590 REM **
3600 GOSUB ' 239 (1) : REM VERIFY CLOSED
3610 GOSUB ' 230 (1,1,2,1,F1$) : REM OPEN DATA BASE
3620 IF Q$ = " " THEN 3650 : REM FILE OPENED
3630 GOSUB ' 032 (" KFAM OPEN ERROR.. FILE 1") : REM PRINT ERR MSG
3640 REM **
3650 GOSUB ' 235 (1) : REM FIND FIRST DAT FIL
3660 IF Q$ = " " THEN 3690 : REM NO ERR FOUND
3670 GOSUB ' 032 (" KFAM FIND FIRST ERROR ..(201)") : REM DISPLAY ERROR MSG
3680 REM **
3690 RETURN : REM EXIT
3700 REM **
3710 REM **
3720 REM ** Subroutine ***** 2 0 3
3730 REM ** Abstract >
3740 REM ** This routine will load a record from the data base.
3750 REM ** Variables >

```

```

3760 REM **
3770 REM **
3780 REM *****
3790 DEFFN / 203 : REM ENTRY
3800 GOSUB / 150 : REM FIND NXT CRS REC
3810 IF Q$ = "E" THEN 3930 : REM BRNCH TO EXIT
3820 KEYIN F$, 3830, 3830 : GOTO 3870 : REM INTRUPT
3830 IF F$ > HEX(1F) THEN 3870 : REM NOT CNTR CHAR
3840 Q$ = "E" : REM SET EOF FLAG
3850 GOTO 3930 : REM BRNCH TO EXIT
3860 REM **
3870 DATA LOAD DC #2, C1$(), N1$(), T$(), I1$()
: C2$(), W$(), R$(), H$()
: M1$(), I9$(), E1$(), S1$(), S2$()
3880 D0$(1) = C1$(Q) : REM FETCH CDP
3890 D4$(1) = T$(Q) : REM FETCH TITLE
3900 D2$(1) = W$(Q) : REM FETCH CRS LEN
3910 CONVERT D2$(1) TO D2(1) : REM CRS LEN (WKS)
3920 D2(2) = D2(1)*7-2 : REM CRS LEN (DAYS)
3930 RETURN : REM EXIT
3940 REM **
3950 REM **
3960 REM ** Subroutine ***** 2 0 5
3970 REM ** Abstract >
3980 REM ** This routine will display the course feasibility.
3990 REM ** Variables >
4000 REM **
4010 REM **
4020 REM *****
4030 DEFFN / 205 : REM ENTRY
4032 F5 = 0
4040 PRINT HEX(030A0A0A) : REM CLR CONSOLE
4050 GOSUB / 101 (1,1,0,0) : REM POSITION CURSOR
4060 PRINT USING 4350, T$(Q)
4070 PRINT
4080 PRINT USING 4360, " "
4090 PRINT USING 4370, " "
4100 PRINT USING 4380, " "
4110 PRINT USING 4390, C1$(Q); T$(Q); W(1); W2(1); I1(1); O(1)
4120 PRINT
4130 PRINT USING 4400, " "
4140 PRINT USING 4410, " "
4150 PRINT USING 4420, " "
4160 PRINT USING 4430, C2(1), C4(1), C5(1); C6(1); I3(1);
M1$(Q); I9$(Q); S1$(Q); S2$(Q); E3(1); E1$(Q)
4170 SELECT PRINT 005(64)
4180 IF F5 <> 0 THEN 4340 : REM IF LPT PRINT
4182 GOSUB / 101 (13,1,1,0) : REM POSITION CURSOR
4190 INPUT "ALTER INPUT OR CLASS SIZE (Y or N)", Y$ : REM INPUT RESPONSE
4200 F5 = 0 : REM INIT CLR PRINT FLG
4210 IF Y$ <> "Y" THEN 4280 : REM IF NO CHANGES
4212 GOSUB / 101 (13,1,1,0) : REM POSITION CURSOR
4220 INPUT "ENTER PLANNED INPUT", I1$(Q) : REM CHANGE INPUT
4222 GOSUB / 101 (13,1,1,0) : REM POSITION CURSOR
4230 INPUT "ENTER CLASS SIZE", C2$(Q) : REM CHANGE SIZE
4240 GOSUB / 110 : REM CONV TO NUMERIC
4250 GOSUB / 111 : REM CMPT NEW VALUES
4260 GOTO 4050 : REM REPRINT DATA
4270 REM **
4280 GOSUB / 101 (13,1,1,0) : REM POSITION CURSOR

```

```

4282 INPUT      "WANT A PRINTOUT (Y OR N)",Y$      : REM INPUT RESPONSE
4290 IF         Y$ <> "Y"                          THEN 4340 : REM GET NEXT CRS
4300 SELECT PRINT 215(64)                          : REM SEL OUTPUT DEV
4310           F5 = 1                              : REM SET PRINT FLAG
4320 GOTO       4040                              : REM PRINT DATA
4330 REM **
4340 GOSUB ' 101 (13,1,1,0)                        : REM POSITION CURSOR
4342 RETURN                                           : REM EXIT
4350 Z          Feasibility Study For --> *****
4360 Z#         LENGTH
4370 Z#         CDP          TITLE          WKS  DAYS  INPUT  AOB
4380 Z#         -----
4390 Z          ****          *****          ***  ****  *****  ****.##
4400 Z#CLASS    SKED    CONV    C/C    INSTRUCTOR    SUPPORT    TRAINERS
4410 Z#SIZE    CLASS    FREQ    CLASS    REG    MPA    OTHER    REG    MPA    REG    AVL
4420 Z#-----
4430 Z          ****          ****          **.*.## **.*.##  ****  **          ****  **          ****  **

```


DISTRIBUTION LIST

Air Force

Headquarters, Air Training Command (XPTD)
Headquarters, Air Training Command (XPTIA, Mr. Goldman)
Air Force Human Resources Laboratory, Brooks Air Force Base
Air Force Human Resources Laboratory (Library), Lowry Air Force Base
Air Force Office of Scientific Research/AR (Dr. A. R. Fregly)
Headquarters Tactical Air Command (DOOS) Langley Air Force Base

Army

Commandant, TRADOC (Technical Library)
ARI (Dr. Ralph R. Canter, 316C; Dr. Edgar Johnson; Mr. James Baker;
Dr. H. F. O'Neil, Jr.; Dr. Beatrice Farr, PERI-OK)
ARI Field Unit - Fort Leavenworth

Coast Guard

Commandant, U.S. Coast Guard Headquarters (G-P-1/62; G-RT/81)

Marine Corps

CMC (OT)
CGMCDEC (Mr. Greenup)
Director, Marine Corps Institute
CO MARCORCOMMELECSCOL (ITS) (Mr. D. E. Mueller)

Navy

OASN (R&D) (Dr. S. Koslov) (MRA&L) (Dr. B. D. Rostker)
CNO (OP-102X, M. Malehorn; OP-987P10, Dr. R. Smith; OP-987, Mr. H. Stone)
NAVCOMPT (NCD-7)
ONR (458, 455)
CNM (MAT-08T2, Mr. A. L. Rubinstein)
CNET (01, 00A, N-5 (6 copies))
CNAVRES (02)
COMNAVSEASYS COM (03, 047C1, 047C12)
COMNAVAIRSYS COM (03, 340F)
CO NAVMEDRSCHDEVCOM (CAPT H. J. Connery)
CNET SUPPORT (00, 00A (2 copies))
CNTECHTRA (016, Dr. Kerr (5 copies))
CNATRA (Dr. F. Schufletowski)
COMTRALANT
COMTRALANT (Educational Advisor)

DISTRIBUTION LIST (continued)

Navy (continued)

COMTRAPAC (2 copies)
CO NAVPERSRANDCEN (Library (4 copies))
NAVPERSRANDCEN Liaison (01H)
Superintendent NAVPGSCOL (2124)
Superintendent Naval Academy Annapolis (Chairman, Behavioral Science Dept.)
CO NAVEDTRAPRODEVCE (AH3)
CO NAVEDTRASUPPCEN NORVA
CO NAVEDTRASUPPCENPAC (5 copies)
CO NAVAEROMEDRSCHLAB (Chief Aviation Psych. Div.)
CO FLTCOMDIRSYSTRACENPAC
CO NAMTRAGRU
CISO, NTTCC Corry Station
CO NAVTRAEEQUIPCEN (N-424 (2 copies), N-211, N-001, N-002)
Center for Naval Analyses (2 copies)
U.S. Naval Institute (CDR Bowler)
OIC NODAC (2)
CO TRITRAFAC (2 copies)
CO NAVSUBTRACENPAC (2 copies)
CO FLEASWTRACENPAC
CO FLETRACEN SDIEGO
CISO, SSC (Dr. Flesch)
Executive Director NAVINSTPRODEVDET
CO NAVTECHTRACEN Corry Station (Cryptologic Training Department)
NETSCPAC Field Office Mare Island
Supply Schools Training Officer (Code 730), Meridian

Other DOD

Military Assistant for Human Resources, OUSDR&E, Room 3D129, Pentagon
(CDR Paul Chatelier)
OASD (MRA&L) (LT COL Grosse)
Director, Human Resources Office, Defense Advanced Research Projects Agency
(Dr. Dexter Fletcher)
Institute for Defense Analyses (Dr. Jesse Orlansky)
COM National Cryptologic School (Code E-2)

Non-DOD

Essex Corporation (Dr. John Collins)

Information Exchanges

DDC (12 copies)
DLSIE (Mr. James Dowling)
Executive Editor, Psychological Abstracts, American Psychological Association
ERIC Processing and Reference Facility, Bethesda, MD (2 copies)