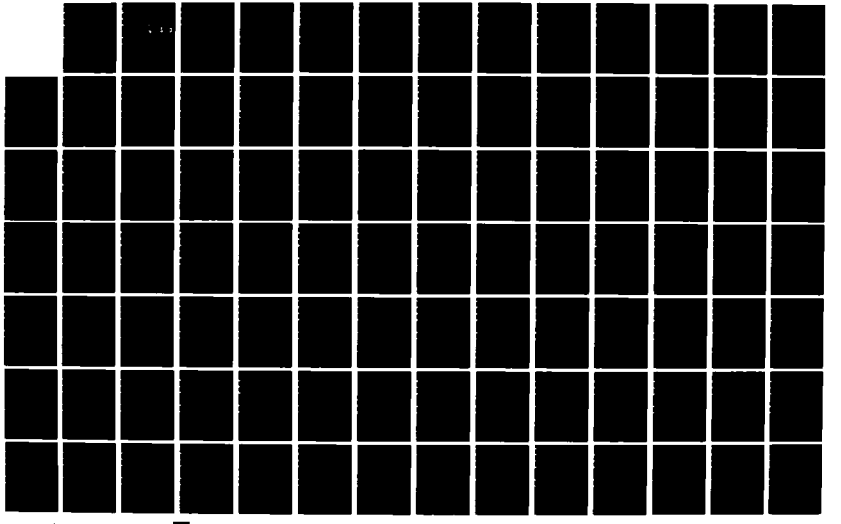
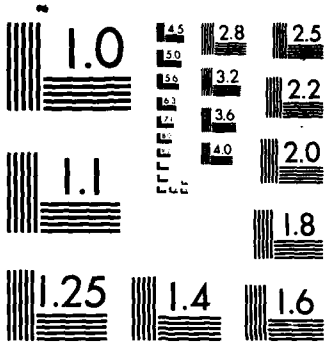
AD-A165 424 IMPLEMENTATION OF A PERSONNEL DATABASE SYSTEM IN HELLENIC ARMED FORCES FORMATIONS(U) NAVAL POSTGRADUATE SCHOOL MONTEREY CA P TSAGARIS ET AL. DEC 85 1/2

UNCLASSIFIED F/G 5/9 NL





(2)

NAVAL POSTGRADUATE SCHOOL

Monterey, California

THESIS

IMPLEMENTATION OF A PERSONNEL DATABASE SYSTEM IN HELLENIC ARMED FORCES FORMATIONS

by

Panagiotis Tsagaris
Constantinos Karaiskos

December 1985

Thesis Advisor: L. Rawlinson

Approved for public release, distribution unlimited

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release, distribution unlimited.
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S)	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION Naval Postgraduate School 6b. OFFICE SYMBOL (If applicable) 52	7a. NAME OF MONITORING ORGANIZATION Naval Postgraduate School
6c. ADDRESS (City, State, and ZIP Code) Monterey, California 93943-5100	7b. ADDRESS (City, State, and ZIP Code) Monterey, California 93943-5100
8a. NAME OF FUNDING/SPONSORING ORGANIZATION 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (City, State, and ZIP Code)	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO. / WORK UNIT ACCESSION NO.

11. TITLE (Include Security Classification)
IMPLEMENTATION OF A PERSONNEL DATABASE SYSTEM IN HELLENIC ARMED FORCES FORMATIONS

12. PERSONAL AUTHOR(S)
Tsagaris, Panagiotis and Karaiskos, Constantinos

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Master's Thesis	FROM _____ TO _____	1985 December	189

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Database

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

The Hellenic Armed Forces Formations currently manage all personnel data manually. The authors propose an automated system to perform this function using dBASE II with an IBM personal computer. Source programs and sample reports are included.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED
22a. NAME OF RESPONSIBLE INDIVIDUAL CDR Linda Rawlinson	22b. TELEPHONE (Include Area Code) 408-646-2735 22c. OFFICE SYMBOL 52Rv

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

SECURITY CLASSIFICATION OF THIS PAGE

Approved for public release, distribution unlimited

Implementation of a Personnel Database System in Hellenic Armed Forces Formations

by

Panagiotis Tsagaris
Lieutenant Colonel, Hellenic Army
B.A., Hellenic Army Academy, 1968

and

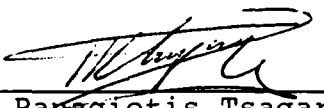
Constantinos Karaiskos
Lieutenant, Hellenic Navy
B.A., Hellenic Navy Academy, 1974

Submitted in partial fulfillment of the
requirements for the degree of

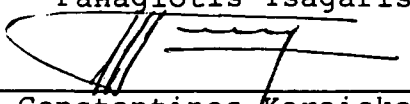
MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

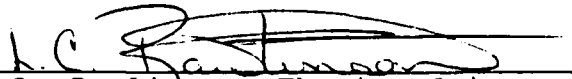
NAVAL POSTGRADUATE SCHOOL
December 1985

Authors: 

Panagiotis Tsagaris



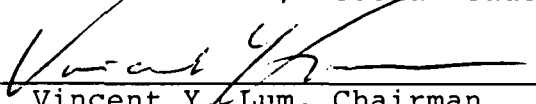
Constantinos Karaiskos

Approved by: 

L. Rawlinson, Thesis Advisor

Paul W Callahan

Paul W. Callahan, Second Reader



Vincent Y. Lum, Chairman,
Department of Computer Science

K.T. Marshall

Kneale T. Marshall
Dean of Information and Policy Sciences

ABSTRACT

The Hellenic Armed Forces Formations currently manage all personnel data manually. The authors propose an automated system to perform this function using dBASE II with an IBM personal computer. Source programs and sample reports are included.

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGEMENT

A number of people assisted us during the writing of this thesis.

We would like to express our deep gratitude to CDR L. C. Rawlinson of the Department of Computer Science and LCDR P. Callahan of the Department of Computer Science, too. In addition to giving encouragement, they read through the thesis, suggested improvements and corrections.

I. INTRODUCTION TO DATABASE CONCEPTS

A. INTRODUCTION

In recent years computer technology has evolved dramatically. As more and more organizations use computer, it is necessary to use systematic approaches for software solutions to their problems.

One approach which is widely used in the computer world is database systems.

Database systems today, play a central role in computer science for the facilities and data handling capabilities they provide.

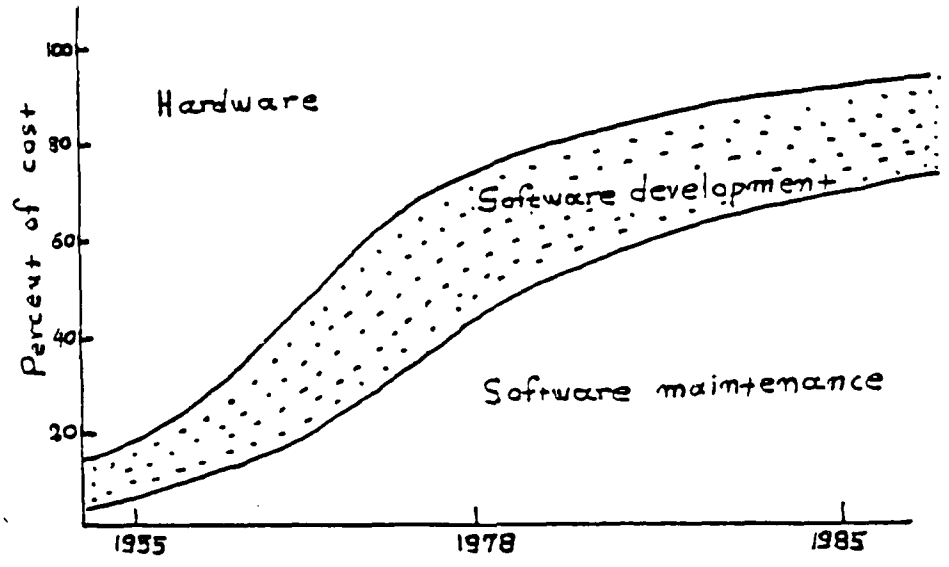


Figure 1. Changing Hardware/Software Cost Ratio

The fact that the hardware cost is decreasing rapidly and software cost continues to increase as shown by Figure 1, in a United States Air Force study [Ref. 1], leads us to consider systems that achieve the best utilization of software development productivity. These considerations motivated system designers to build advanced database systems in order to decrease software cost and obtain maximum benefit.

In our case the benefit will be the savings of manpower for other purposes.

B. DATABASE SYSTEMS VS MANUAL SYSTEMS IN HELLENIC ARMED FORCES FORMATIONS

Currently all of the information required by Formation Commanders are handled manually by the staff of the formation, resulting in time consuming operations and inaccuracies.

Because of the continuous changes concerning personnel and the associated data, it is extremely difficult for the staff personnel to keep track of these changes. Many systems are very inefficient and the Commander of the Formation does not have accurate and timely information in order to make fast decisions.

These problems could be solved by the implementation of a computerized personnel database system.

A database system has several important advantages over manual systems, which are described below.

First data can be shared. This reduces the time needed to develop new systems or to respond to various requests. In

effect, all the necessary information can be retrieved from existing data much faster and with a higher degree of accuracy.

The second advantage of a database system is the elimination or reduction of data duplication that can lead to a lack of data integrity in conflicting reports.

The third advantage is that the personnel involved in manual personnel management could be reduced considerably, freeing manpower for other tasks.

C. CONCLUSIONS

In order to increase the effectiveness of Hellenic Armed Forces, it is essential that personnel management be performed very efficiently. However, to manually manage all Armed Forces personnel is a very tedious, complex and time consuming job. Increased personnel volume has also increased the task of management making the operations more and more complicated. Furthermore, personnel managers and decision makers will need reliable information faster than is currently available. It is almost impossible to get all the information required by the personnel managers in the time frame allowed with a manual system.

On the other hand, an automated system could result in decreasing the number of personnel working in personnel management offices, freeing some for manning in other understaffed positions.

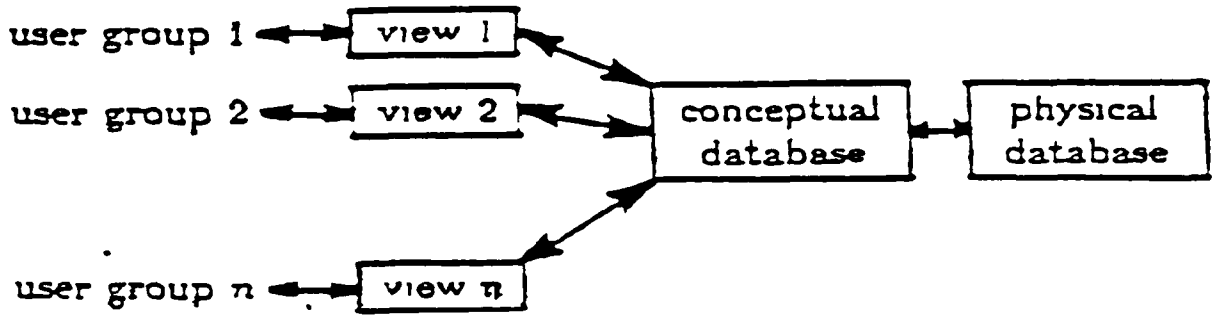


Figure 3. Levels of Abstraction in a Database System

3. Database Systems vs Traditional File Systems

A database management system (DBMS) is considerably different from a traditional file system.

A traditional file processing approach is shown in Figure 4.

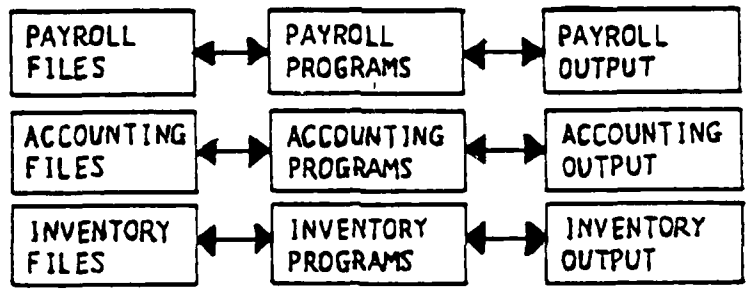


Figure 4. Traditional File Processing Approach

We observe that each file program system processes only its own file.

Figure 5 shows a database processing approach.

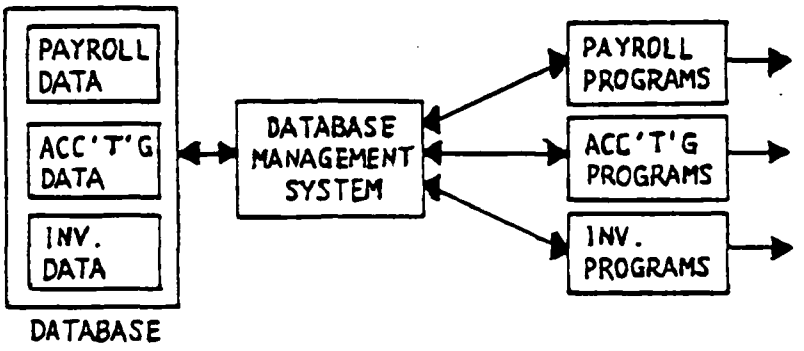


Figure 5. Database Processing Approach

Here the DBMS integrates the data and makes it much easier to get useful information from more than one file.

Data is not monitored and manipulated by the individual application programs, but instead by the DBMS.

Such a Database Management System is dBASE II, which will be used as the DBMS in our database system.

The disadvantages of the traditional file approach are:

(1) Uncontrolled redundancy.

(2) Inconsistent data.

(3) Inflexibility.

(4) Limited data sharing.

(5) Poor enforcement of standard.

(6) Low programmer productivity.

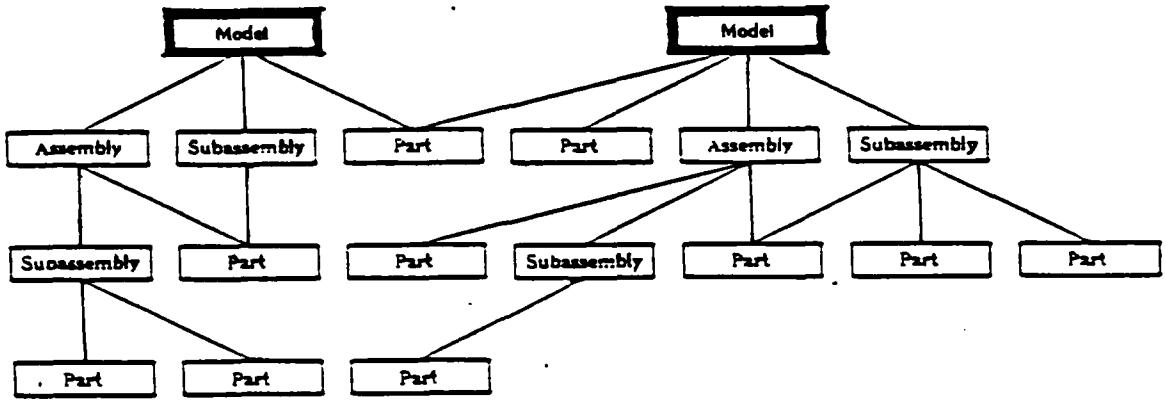


Figure 7. Network Data Model

A relational data model differs from HDM and NDM in architecture. Information is stored in two-dimensional tables which are called files.

These tables which are shown in Figure 8 have the following properties:

(1) Each column contains values about the same attribute.

(2) Each column has a distinct name.

(3) Each row is distinct.

(4) The sequence of the rows is immaterial.

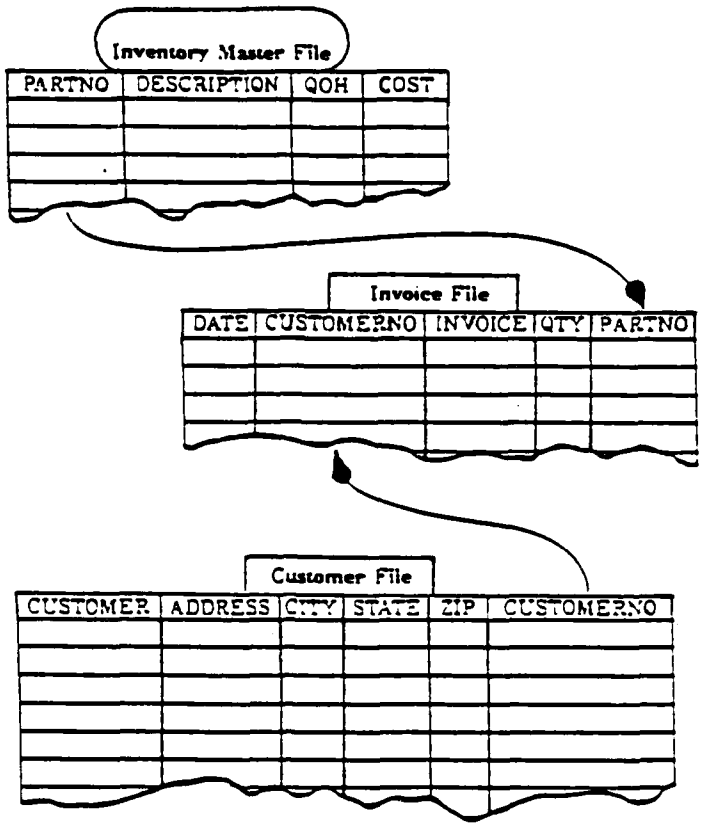


Figure 8. Relational Data Model

The above scheme deals with a parts inventory system. Information in the tables is accessed by the user based on any desired relationship.

E. dBASE II CONCEPTS

dBASE II is a relational database management system. However, since several files are generally used in a given application and the relationship between information in different files is not stored in the system, dBASE II is not a true database management system in the strictest sense of

the word. dBASE II is more like a file management system with relational features added. dBASE II does contain its own programming language, permitting a user to develop extremely powerful and complex programs that meet demanding applications like general personnel, accounting and inventory control.

1. Features of dBASE II [Ref. 4]

The most important features of dBASE II are:

a. Independence of programs and data. Changes in file structures do not affect programs.

b. Data can be easily updated.

c. Sorting and indexing capabilities.

d. Easy creation of reports by the report generator facility, or under program control.

e. Very high-level built-in language which supports structured programming.

2. Limitations of dBASE II [Ref. 3]

a. dBASE II allows only two files to be open at a time. This creates difficulties which can be overpassed by using special techniques but the system will slow down.

b. dBASE II allows only 32 fields per record, which is enough for most applications, and the maximum number of characters permitted per record is 1000.

c. Each field can be up to 254 characters long.

d. dBASE II allows 16 programs to be run at any given time, reduced by the number of data files in use. For example, if we have 2 files in use, then 14 programs are allowed.

e. dBASE II applications are slower than compiled programs.

(7) Excessive program maintenance.

The advantages of the database processing approach are:

(1) Minimal data redundancy.

(2) Consistency of data.

(3) Integration of data.

(4) Sharing of data.

(5) Enforcement of standard.

(6) Ease of application programs.

(7) Uniform security, privacy and integrity constraints.

(8) Data accessibility and responsiveness.

(9) Data independence.

(10) Reduced program maintenance.

4. Models of Database Management Systems

A MODEL is a representation of real-world objects, events and their association.

A DATA MODEL is an abstract representation of the data about entities, events, activities and their associations. The purpose of data models is to represent data in understandable terms. The main data models in use today are the hierarchical data model (HDM), network data model (NDM) and the relational data model (RDM) [Ref. 4]. Since the hierarchical model is a special case of the network model, actually there are two types. A complete discussion of these models is beyond the scope of this thesis but a brief overview is important as an introduction to dBASE II.

In a hierarchical or network system, information is stored in a structure that looks very much like a tree (Fig. 6).

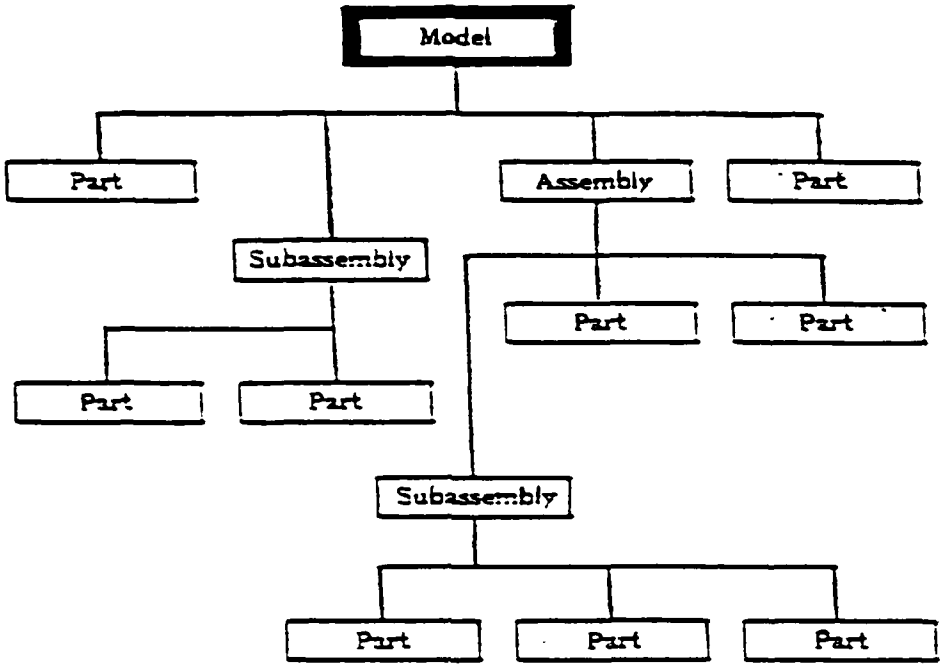


Figure 6. Hierarchical Data Model

An entity is composed of various assemblies and each assembly is composed of various subassemblies which in turn are composed of parts. Each part is a child of the parent assembly or subassembly that owns it. In a hierarchical model no child can have more than one parent.

In a Network model a child can have more than one parent as shown in Figure 7.

It represents data as a set of record types and pairwise relationships between record types.

Database systems have become important tools for retrieving timely and accurate information and is expected to provide its user with the required information within a specified time. Therefore, a standard database system should be developed for efficient personnel management in the Hellenic Armed Forces Formations.

D. GENERAL OVERVIEW OF A DATABASE SYSTEM

We begin with the definition and some of basic terminology of databases and then discuss the architecture and types of data models.

1. Definition and Basic Terminology

Figure 2 shows the relationship between the basic terms of a database, which are explained below.

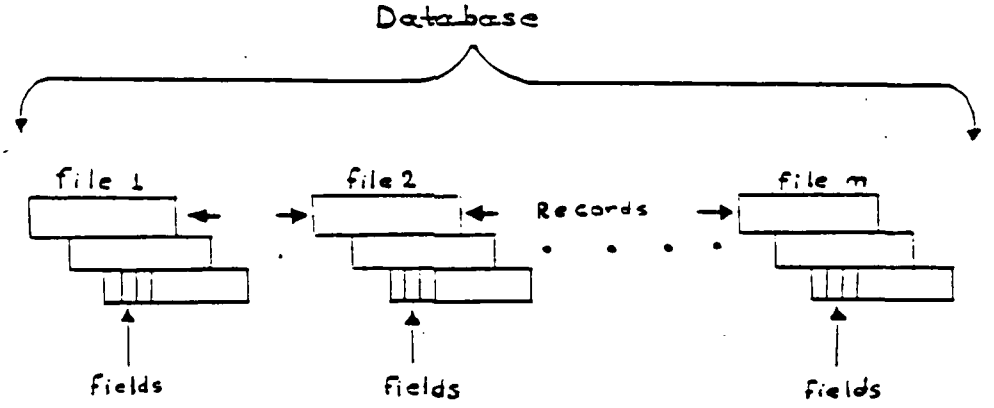


Figure 2. Relationship Between Basic Terms of a Database

a. Database

A shared collection of interrelated data designed to meet the varied information needs of an organization.

b. Database Management System (DBMS)

A software system that performs all user's requests (update, retrieval) for data.

c. File

Is a collection of records concerning entities of the same type.

d. Record

Is a collection of data concerning one entity of a file. Each record has an identical format.

e. Field

A field is part of the record and is the smallest unit of named data.

2. Architecture for a Database System

The architecture is divided into three general levels: internal, conceptual, and external [Ref. 2]. Figure 3 shows the standard viewpoints regarding the three levels.

In Figure 3 a single database, which may be one of many databases using the same DBMS, is viewed at three different levels. Only the physical database exists. The conceptual database is an abstract representation of the physical database and the views are either abstractions or portions of the conceptual database.

II. ANALYSIS PHASE

In this chapter the analysis phase for our personnel database system is discussed. This phase includes the definition of the system objectives, the definition of the output information needed to meet these objectives, the definition of output forms, and definition of input information needed to obtain the desired output.

A. SYSTEM OBJECTIVES

As we stated in the previous chapter personnel management in Hellenic Armed Forces Formations is handled manually. This results in time consuming operations, inefficiency, and inaccuracy, which in turn results in a need for additional personnel, leaving other serious positions unmanned. In addition, decision making may be late which could result in disorder as far as personnel management is concerned.

From the above discussion it is evident that a computerized system for handling personnel is needed.

With such a system it will be possible to have any information concerning personnel updated at any time, with less effort and maximum accuracy. This will result in better decision making, faster operations and a savings of personnel for other purposes.

More specifically we will consider the following objectives for the system [Ref. 7].

(1) Application development must be easier, cheaper, faster and more flexible.

(2) The data may have multiple uses.

(3) Clarity.

(4) Ease of use.

(5) Flexible usage.

(6) Ease of change.

(7) Low cost.

(8) Performance.

(9) Privacy.

(10) Availability.

(11) Reliability.

B. OUTPUT INFORMATION

Our database system can be applied to all the personnel of a formation but for the purpose of this thesis we will include only the officers.

To meet the above objectives the following output information is needed:

(1) List of officers in alphabetical order including serial number, rank, unit, and report date.

(2) List of officers ordered by rank including service entry date, unit and duty.

(3) List of units with their officers.

(4) List of officers with nonmilitary studies.

(5) List of officers who speak foreign languages.

(6) List of officers including marital status.

(7) List of officers with service time in current unit and total service time.

(8) List of officers with their addresses and phone numbers.

The above lists will be issued regularly every month, but they will also be available at any time.

C. INPUT INFORMATION

In order to keep track of all the officers belonging to a Formation we must consider the following:

Each officer has a serial number, rank, duty, and he belongs to a unit. We need to know his total service time, and the service time in current unit, as well as his marital status and where he lives, in order to respond to queries in the case of emergencies.

Each unit has several officers and is identified by a code number which is distinct for Army and Navy units.

For security purposes the personnel data contained within this thesis is purely artificial.

In addition, we need to know, the education of each officer, including non-military studies and foreign language knowledge. For the purpose of this thesis, we assume that each officer speaks at most one foreign language besides his mother tongue.

A detailed description and the input information included in the files, which will be created to support the above needs, are provided in the design phase.

III. DESIGN PHASE

In this phase the functions of the system are defined, the files of the database are designed and a collection of programs is defined, to support each function.

A. SYSTEM FUNCTIONS

The system is separated into three functional entities as follows:

1. Update Operations

This function allows the user to enter, delete and modify records in all the supporting files. These operations are performed daily.

2. Report Generators

This function is for retrieving all the necessary information from our database on a monthly basis, or upon request.

3. Miscellaneous

This function will include the following:

a. When a user enters the system to do a specific job a record is automatically created containing the name of who makes the change, the date of the change, and the kind of the job. This file can be printed and deleted as necessary to monitor changes.

b. When an officer is to be deleted from the MASTER file, before the deletion operation is performed, some data

concerning this officer will be transferred to a statistical file including the serial number, the name, the unit, the rank, the report date in the unit, and the date of deletion. This file will be available yearly or upon request.

c. A screen display or printer output for fast retrieval of the data concerning an officer.

The functional blocks of the system are shown in Figure 9.

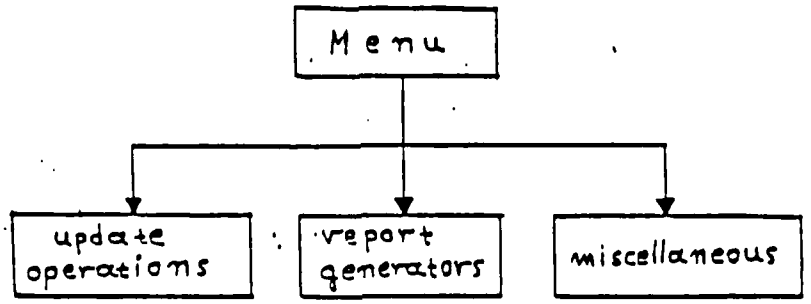


Figure 9. Functional Blocks of the System

B. FILE DESIGN

To support the above functions the following files with the corresponding structures were created. The names of the files and fields are the ones that are used in our programs.

1. Master File

It is the main file for our system containing the necessary information for each officer.

Structure for file: MASTER

FIELD	NAME	TYPE	WIDTH
01	SERNO	C	4
02	NAME	C	16
03	RANK	C	2
04	UNIT	C	4
05	SERENTRY	N	6
06	REPTDATE	N	6
07	DUTY	C	2
08	EDUCAT	C	2
09	DEGREE	C	1
10	FORLANG	C	2
11	MARSTAT	C	1
12	CHILDREN	C	1
13	ADDRESS	C	20
14	PHONE	C	7

Primary key: SERNO

2. Units File

Structure for file: UNITS

FIELD	NAME	TYPE	WIDTH
01	CODE	C	4
02	TITLE	C	12
03	LOCATION	C	10

Primary key: CODE

3. Ranks File

Structure for file: RANKS

FIELD	NAME	TYPE	WIDTH
01	CODE	C	2
02	ARMYNAME	C	4
03	NAVYNAME	C	4

Primary key: CODE

4. Duties File

Structure for file: DUTIES

FIELD	NAME	TYPE	WIDTH
01	CODE	C	2
02	NAME	C	20

Primary key: CODE

5. Forlangs File

Structure for file: FORLANGS

FIELD	NAME	TYPE	WIDTH
01	CODE	C	2
02	NAME	C	12

Primary key: CODE

6. Sciences File

Structure for file: SCIENCES

FIELD	NAME	TYPE	WIDTH
01	CODE	C	2
02	NAME	C	15

Primary key: CODE

7. Monitor File

It is a file that keeps track of who does what and when.

Structure for file: MONITOR

FIELD	NAME	TYPE	WIDTH
01	DATE	C	8
02	NAME	C	16
03	JOB	C	12
04	FLE:REPT	C	12
05	NUMREC	N	3

8. Statistic File

It is the file for statistical information for each deleted officer.

Structure for file: STATISTIC

FIELD	NAME	TYPE	WIDTH
01	SERNO	C	4
02	NAME	C	16
03	RANK	C	2
04	UNIT	C	4
05	REPTDATE	C	8
06	DELDATE	C	8

9. Tempor File

Structure for file: TEMPOR

FIELD	NAME	TYPE	WIDTH
01	PASSWORD	C	5
02	SELECT	C	2
03	COUNTER	C	3

C. EXPLANATION OF FIELDS

The explanation of the fields, where not obvious, is given below:

1. Master File

a. SERNO: The serial number of the officer.

b. SERENTRY: The date he entered the military academy.

c. REPTDATE: The date he was positioned in the current unit.

d. EDUCAT: His education besides the military studies.

e. FORLANG: Foreign languages he speaks.

f. MARSTAT: His marital status.

2. Monitor File

a. FLE:REPT: The file the user works with, or the kind of report he prints.

3. Statistic File

a. SERNO, REPTDATE: The same as above.

b. DELDATE: The date of deletion from the unit.

D. DATA FLOW DIAGRAM

The update operations are performed daily to update all the files of the system. These operations include insertion and deletion of records, as well as modifications in one or more fields. The output of the update operations are the same files updated from the transactions of the day. These files are then used as input by the report generator function, to obtain the various reports.

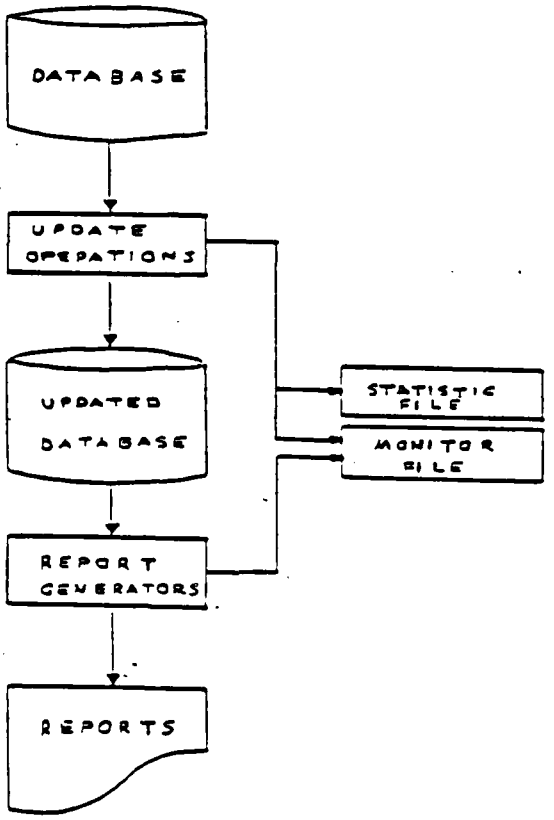


Figure 10. Data Flow Diagram for the Personnel Database System

During the update operations the STATISTIC file is created automatically.

The MONITOR file is created from the update operations, report generators, and miscellaneous operations.

The STATISTIC and MONITOR files are not seen by the user, so supporting the information hiding principle.

The data flow diagram of the system is shown in Figure 10.

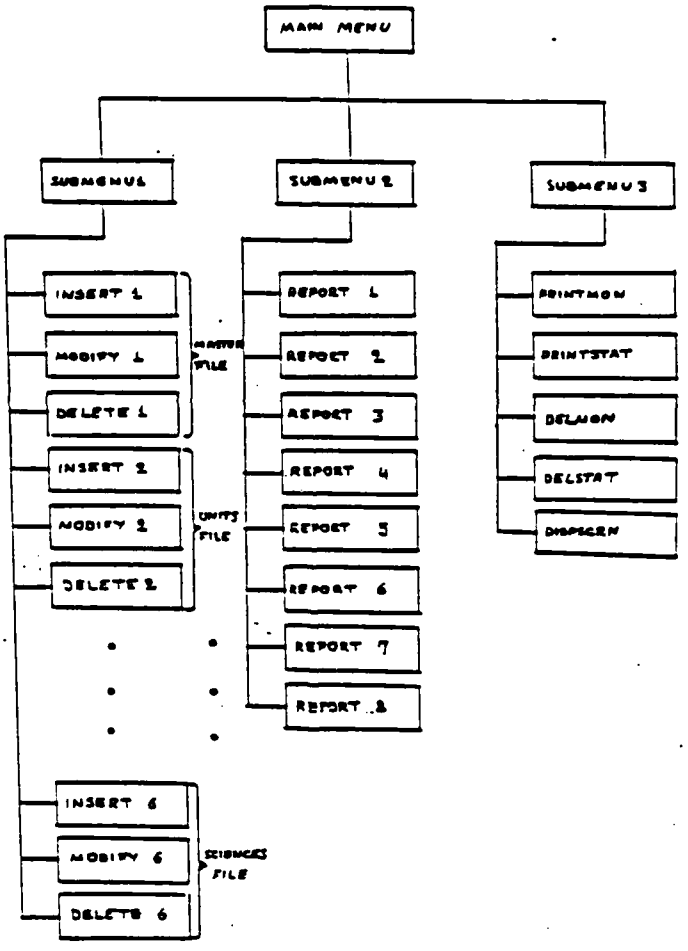


Figure 11. Program Map for the Personnel Database System

E. SUPPORTING PROGRAMS

The functional blocks we have already defined are selected by the user from the main menu of the system. The control is then transferred to the corresponding submenu (one for each function). From the submenus the appropriate job is selected from an option list. The main menu, submenus, and programs form a hierarchical relationship. The diagram of this relationship is the program map. The program map for the personnel database system is shown in Figure 11.

All the update operations, reports and printing, as well as deletion of MONITOR and STATISTIC files are selected from SUBMENU1, SUBMENU2, and SUBMENU3 respectively.

F. CODE SYSTEM FOR THE PERSONNEL DATABASE

1. Codes for Ranks

CODE	ARMY	NAVY
01	GENERAL (GEN)	ADMIRAL (ADM)
02	LT GENERAL (LTG)	VICE ADMIRAL (VADM)
03	MAJOR GENERAL (MG)	REAR ADMIRAL (RADM)
04	BRIG GENERAL (BG)	COMMODORE (COMD)
05	COLONEL (COL)	CAPTAIN (CAPT)
06	LT COLONEL (LTC)	COMMANDER (CDR)
07	MAJOR (MAJ)	LT COMMANDER (LCDR)
08	CAPTAIN (CPT)	LIEUTENANT (LT)
09	FIRST LIEUTENANT (1LT)	FIRST LIEUTENANT (1LT)
10	SECOND LIEUTENANT (2LT)	ENSIGN (ENS)

2. Codes for Units

CODE	TITLE
1000	1st Inf Div
1101	1st Inf Bn
1102	2nd Inf Bn
1103	3rd Inf Bn
1104	4th Inf Bn
1105	5th Inf Bn
1206	1st Arty Bn
1207	2nd Arty Bn
1308	Armour Bn
1409	Engineers Bn
1510	Signals Bn
1611	Sup/Trans Bn
2000	Navy Forces
2101	DD Squadron
2202	LST Squadron
2103	DD Miaoulis
2104	DD Kanaris
2105	DD Themis
2206	LST Argo
2207	LST Rhodes

The meaning of the four digits representing the codes of the units is as follows:

1ST digit

1: ARMY

2: NAVY

2ND digit

It distinguishes between the various kinds of units as follows:

ARMY

0: HEADQUARTERS

1: INFANTRY

2: ARTILLERY

3: ARMOUR

4: ENGINEERS

5: SIGNAL

6: SUPPLY-TRANSPORTATION

NAVY

0: COMMAND

1: DD SQUADRON

2: LST SQUADRON

The 3rd and 4th digit specify each unit.

3. Codes for Duties

CODE	NAME
01	COMMANDER
02	DTY COMMANDER
03	CHIEF OF STAFF
04	1st OFFICE MANAGER

05	2nd OFFICE MANAGER
06	3rd OFFICE MANAGER
07	4th OFFICE MANAGER
08	COMPANY COMMANDER
09	BATTERY COMMANDER
10	COMMANDING OFFICER
11	EXECUTIVE OFFICER
12	ASW OFFICER
13	NAVIGATION OFFICER
14	COMMUNICATION OFFICER
15	CIC OFFICER
16	ADJUTANT

4. Codes for Sciences

CODE	NAME
01	MATHEMATICS
02	PHYSICS
03	CHEMISTRY
04	AERONAUTICS
05	EL/ENGINEERING
06	MECH/ENGINEERING
07	ECONOMICS
08	OPER/RESEARCH
09	INFO/SYSTEMS
10	COMP SYSTEMS
11	COMP SCIENCE
12	MANAGEMENT

13	WEAPONS
14	OCEANOGRAPHY
15	SHIPBUILDING
16	METEOROLOGY
17	MEDICINE
18	LEGAL SCIENCES
19	ARCHITECTURE
20	CIV ENGINEERING

5. Codes for Education Degree

CODE	NAME
B	BACHELOR
M	MASTER
P	Ph.D

6. Codes for Foreign Languages

CODE	NAME
01	ENGLISH
02	GERMAN
03	ITALIAN
04	FRENCH
05	SPANISH
06	TURKISH
07	ARABIC
08	JAPANESE
09	CHINESE
10	PORTUGUESE
11	RUSSIAN

12	BULGARIAN
13	KOREAN
14	DANISH
15	SWEDISH
16	DUTCH

7. Codes for Marital Status

CODE	NAME
M	MARRIED
U	UNMARRIED
D	DIVORCED

IV. IMPLEMENTATION PHASE

A. MAIN MENU AND SUBMENUS OF THE SYSTEM

1. Mainmenu

This program controls the whole operation of the system, and is called from another program called BEGIN, which is the only program the user calls by name. The program BEGIN also sets drive B to be the default drive. After initializing basic dBASE II functions the program MAINMENU calls another program called PASSWORD which is for aborting unauthorized users, and then it proceeds by displaying the front page of the system. Then it displays on the screen the main menu of the system, and pauses waiting for the user to make his choice which is stored in the variable 'choice'. Then a CASE statement permits the program to branch to the corresponding SUBMENU, or exit either to the operating system, or to dBASE II. If the user selects a wrong choice, then the program prints an error message, rings the bell, and redisplays the screen. This is performed by a WHILE loop using the boolean value T.

2. Password

This program is called by the MAINMENU once at the beginning. Each user has a password to enter the system. If the password is the right one, the user is allowed to continue, otherwise the program gives one more change. If

the password is wrong again, the program automatically exits to dBASE II, displaying the appropriate message.

During the execution of this program, we have to store somewhere the password to be used by the MONITR program, for monitoring the changes. This is done by using a file called TEMPOR with three fields, password, select, and counter. This file contains only one record, which is modified properly during execution of the programs PASSWORD, and SUBMENUS. The MONITR program is called from the programs INSERT, DELETE, MODIFY, REPORTS, and MISCELLANEOUS.

3. Submenus

All three SUBMENUS operate with the same logic as the MAINMENU. The screen displays guide the user in what to do in each case. These programs also store in the second field of the file TEMPOR, the selection made by the user.

The listings of the programs are given in the next pages.

```
********** PROGRAM MAINMENU **********

* This program controls the whole operation of the system

* Initialize basic functions

SET TALK OFF
SELECT PRIMARY
SET FORMAT TO SCREEN
SET PRINT OFF
SET CONSOLE ON
DO password
DO frontpage
DO setdate
DO WHILE T
   ERASE
   @  1,52 SAY DATE()
   @  2,13 SAY "************************************************"
   @  3,13 SAY "************************************************"
   @  4,13 SAY "**                                            **"
   @  5,13 SAY "**              M A I N   M E N U             **"
   @  6,13 SAY "**           What do you want to do?          **"
   @  7,13 SAY "************************************************"
   @  8,13 SAY "**                                            **"
   @  9,13 SAY "**      0  =  EXIT TO OPERATING SYSTEM        **"
   @ 10,13 SAY "**              (When you are finished)       **"
   @ 11,13 SAY "**      1  =  UPDATE OPERATIONS               **"
   @ 12,13 SAY "**              (Insert,delete,modify records) **"
   @ 13,13 SAY "**      2  =  REPORT GENERATORS               **"
   @ 14,13 SAY "**              (Reports available)           **"
   @ 15,13 SAY "**      3  =  MISCELLANEOUS                   **"
   @ 16,13 SAY "**              (Files statistics)            **"
   @ 17,13 SAY "**      4  =  EXIT TO dBASE                   **"
   @ 18,13 SAY "**              (For programmers only)        **"
   @ 19,13 SAY "************************************************"
   @ 20,13 SAY "************************************************"
   @ 23,13 SAY "         Enter the corresponding number          "
   STORE " " TO choice
   WAIT TO choice

*  Accept choice and branch to corresponding submenu

   DO CASE
      CASE choice = "0"
         QUIT
      CASE choice = "1"
         DO submenu1
      CASE choice = "2"
         DO submenu2
      CASE choice = "3"
         DO submenu3
      CASE choice = "4"
         CANCEL
      OTHERWISE
```

```
*          Print an error message and give an other chance

           ERASE
           @ 10,17 SAY choice+" IS AN ILLEGAL CHOICE - TRY AGAIN"
           SET TALK OFF
           ? CHR(7)
           STORE 1 TO del
           DO WHILE del < 40
              STORE del+1 TO del
           ENDDO WHILE
   ENDCASE
ENDDO WHILE T
```

```
******** PROGRAM PASSWORD ********
*
* This program is for aborting unauthorized users
*
ERASE
@ 10,22 SAY "*****  ENTER PASSWORD  *****"
STORE "      " TO pasword
STORE 0 TO count
STORE T TO Okey
DO WHILE Okey
   SET CONSOLE OFF
   ACCEPT TO pasword
   SELECT PRIMARY
   USE TEMPOR
   DELETE ALL
   PACK
   APPEND BLANK
   REPLACE password WITH pasword
   USE
   IF pasword = "QWERT" .OR. pasword = "ASDFG"
      SET CONSOLE ON
      @ 15,25 SAY "OK, You may continue"
      SET TALK OFF
      STORE 1 to del
      DO WHILE del < 25
         STORE del+1 to del
      ENDDO
      RETURN
   ELSE
      SET CONSOLE ON
      ERASE
      ? CHR(7)
      @ 10,22 SAY "WRONG PASSWORD - TRY AGAIN"
      STORE count+1 TO count
      IF count = 2
         ERASE
         DO monitr
         @ 10,25 SAY "** UNAUTHORIZED USER **"
         SET DEFAULT TO A
         CANCEL
      ELSE
         STORE "      " TO pasword
      ENDIF
   ENDIF
ENDDO WHILE
```

```
******** PROGRAM FRONTPAGE ********

* This program prints the main heading of the system

ERASE
@  1,22 SAY "********************************"
@  2,22 SAY "****    ************************"
@  3,22 SAY "****    ************************"
@  4,22 SAY "            ********************"
@  5,22 SAY "****    ************************"
@  6,22 SAY "****    ************************"
@  7,22 SAY "********************************"
@  8,22 SAY "********************************"
@  9,22 SAY "********************************"
@ 10,22 SAY "********************************"
@ 12,15 SAY "************************************************"
@ 13,15 SAY "************************************************"
@ 14,15 SAY "**                                            **"
@ 15,15 SAY "**        A PERSONNEL DATABASE SYSTEM         **"
@ 16,15 SAY "**                  F O R                     **"
@ 17,15 SAY "**    HELLENIC ARMED FORCES FORMATIONS        **"
@ 18,15 SAY "**                                            **"
@ 19,15 SAY "************************************************"
@ 20,15 SAY "************************************************"
SET TALK OFF
STORE 1 TO delay
DO WHILE delay < 80
   STORE delay+1 TO delay
ENDDO WHILE
RETURN
```

```
******** PROGRAM SETDATE ********

* This program asks the user to set the current date

ERASE
SET TALK OFF
STORE "         " TO mdate
STORE T TO continue
DO WHILE continue
   ERASE
   @ 12,12 SAY "ENTER DATE (MM/DD/YY) " GET mdate;
                PICTURE "99/99/99"
   READ
   IF        VAL($(mdate,1,2)) < 1;
        .OR. VAL($(mdate,1,2)) > 12;
        .OR. VAL($(mdate,4,2)) < 1;
        .OR. VAL($(mdate,4,2)) > 31;
        .OR. VAL($(mdate,7,2)) <> 85
      STORE "         " TO mdate
      @ 22,15 SAY "INVALID DATE, PLEASE RETRY"
      STORE 1 TO del
      DO WHILE del < 35
         STORE del+1 TO del
      ENDDO WHILE
      LOOP
   ELSE
      STORE F TO continue
      SET DATE TO &mdate
   ENDIF
ENDDO WHILE
RETURN
```

```
******** PROGRAM MONITR ********

* This program adds a record in the file MONITOR each time a user
* enters the system, to keep track of who does what and when.

ERASE
STORE "              " TO mjob
STORE "              " TO mfle:rept

* Use the fields 'password'and 'select' of file TEMPOR and store
* the appropriate information to temporary memory variables

SELECT SECONDARY
USE tempor
IF password = "QWERT"
   STORE "Tsagaris  Panag" TO mname
ELSE
   IF password = "ASDFG"
      STORE "Karaiskos Const" TO mname
   ELSE
      STORE "Unauthorized    " TO mname
   ENDIF
ENDIF
DO CASE
   CASE select = "11"
      STORE "Insertion    " TO mjob
      STORE "Master       " TO mfle:rept
   CASE select = "12"
      STORE "Insertion    " TO mjob
      STORE "Units        " TO mfle:rept
   CASE select = "13"
      STORE "Insertion    " TO mjob
      STORE "Ranks        " TO mfle:rept
   CASE select = "14"
      STORE "Insertion    " TO mjob
      STORE "Duties       " TO mfle:rept
   CASE select = "15"
      STORE "Insertion    " TO mjob
      STORE "Forlangs     " TO mfle:rept
   CASE select = "16"
      STORE "Insertion    " TO mjob
      STORE "Sciences     " TO mfle:rept
   CASE select = "21"
      STORE "Deletion     " TO mjob
      STORE "Master       " TO mfle:rept
   CASE select = "22"
      STORE "Deletion     " TO mjob
      STORE "Units        " TO mfle:rept
   CASE select = "23"
      STORE "Deletion     " TO mjob
      STORE "Ranks        " TO mfle:rept
   CASE select = "24"
      STORE "Deletion     " TO mjob
      STORE "Duties       " TO mfle:rept
```

```
CASE select = "25"
   STORE "Deletion     " TO mjob
   STORE "Forlangs     " TO mfle:rept
CASE select = "26"
   STORE "Deletion     " TO mjob
   STORE "Sciences     " TO mfle:rept
CASE select = "31"
   STORE "Modification" TO mjob
   STORE "Master       " TO mfle:rept
CASE select = "32"
   STORE "Modification" TO mjob
   STORE "Units        " TO mfle:rept
CASE select = "33"
   STORE "Modification" TO mjob
   STORE "Ranks        " TO mfle:rept
CASE select = "34"
   STORE "Modification" TO mjob
   STORE "Duties       " TO mfle:rept
CASE select = "35"
   STORE "Modification" TO mjob
   STORE "Forlangs     " TO mfle:rept
CASE select = "36"
   STORE "Modification" TO mjob
   STORE "Sciences     " TO mfle:rept
CASE select = "1"
   IF sub3
      STORE "Printing     " TO mjob
      STORE "Monitor      " TO mfle:rept
   ELSE
      STORE "Printing     " TO mjob
      STORE "Report1      " TO mfle:rept
   ENDIF
CASE select = "2"
   IF sub3
      STORE "Printing     " TO mjob
      STORE "Statistic    " TO mfle:rept
   ELSE
      STORE "Printing     " TO mjob
      STORE "Report2      " TO mfle:rept
   ENDIF
CASE select = "3"
   IF sub3
      STORE "Deletion     " TO mjob
      STORE "Monitor      " TO mfle:rept
   ELSE
      STORE "Printing     " TO mjob
      STORE "Report3      " TO mfle:rept
   ENDIF
CASE select = "4"
   IF sub3
      STORE "Deletion     " TO mjob
      STORE "Statistic    " TO mfle:rept
   ELSE
      STORE "Printing     " TO mjob
```

```
            STORE "Report4      " TO mfle:rept
         ENDIF
      CASE select = "5"
         IF sub3
            STORE "Screen Disp " TO mjob
            STORE "Master       " TO mfle:rept
         ELSE
            STORE "Printing     " TO mjob
            STORE "Report5      " TO mfle:rept
         ENDIF
      CASE select = "6"
         STORE "Printing     " TO mjob
         STORE "Report6      " TO mfle:rept
      CASE select = "7"
         STORE "Printing     " TO mjob
         STORE "Report7      " TO mfle:rept
      CASE select = "8"
         STORE "Printing     " TO mjob
         STORE "Report8      " TO mfle:rept
ENDCASE
STORE counter TO mnumrec

* Append the record to the MONITOR file

SELECT SECONDARY
USE monitor
APPEND BLANK
REPLACE date WITH DATE()
REPLACE name WITH mname
REPLACE job WITH mjob
REPLACE fle:rept WITH mfle:rept
REPLACE numrec WITH mnumrec
USE
SELECT PRIMARY
RETURN
```

```
******** PROGRAM SUBMENU1 ********

* This program controls the Update Operations
*           of the system

ERASE
STORE "  " TO select
DO WHILE T
   ERASE
   @ 2,50 SAY DATE()
   @ 3,15 SAY  "*********************************************"
   @ 4,15 SAY  "*                                           *"
   @ 5,15 SAY  "*              S U B M E N U 1              *"
   @ 6,15 SAY  "*    U P D A T E   O P E R A T I O N S      *"
   @ 7,15 SAY  "*                                           *"
   @ 8,15 SAY  "*********************************************"
   @ 9,15 SAY  "*      OPTIONS                FILES         *"
   @ 10,15 SAY "*      -------                -----         *"
   @ 11,15 SAY "*                           1=MASTER        *"
   @ 12,15 SAY "*   0=Exit to MAIN MENU     2=UNITS         *"
   @ 13,15 SAY "*   1=Insertion             3=RANKS         *"
   @ 14,15 SAY "*   2=Deletion              4=DUTIES        *"
   @ 15,15 SAY "*   3=Modification          5=FORLANGS      *"
   @ 16,15 SAY "*                           6=SCIENCES      *"
   @ 17,15 SAY "*                                           *"
   @ 18,15 SAY "*********************************************"
   @ 20,15 SAY "  Make the appropriate selection by pressing"
   @ 21,15 SAY "the option number first and then the file   "
   @ 22,15 SAY "number. For example, if you want to insert  "
   @ 23,15 SAY "records in file UNITS, press 12.            "
   ACCEPT TO selection

*  Store selection to the field select of the file TEMPOR
*  in order to be used by the monitor program

   SELECT PRIMARY
   USE tempor
   REPLACE select WITH selection
   USE

*  Accept the selection and branch to the corresponding program

   DO CASE
      CASE selection = "0"
         RETURN
      CASE selection = "11"
         DO insert1
      CASE selection = "12"
         DO insert2
      CASE selection = "13"
         DO insert3
      CASE selection = "14"
         DO insert4
      CASE selection = "15"
```

```
            DO insert5
         CASE selection = "16"
            DO insert6
         CASE selection = "21"
            DO delete1
         CASE selection = "22"
            DO delete2
         CASE selection = "23"
            DO delete3
         CASE selection = "24"
            DO delete4
         CASE selection = "25"
            DO delete5
         CASE selection = "26"
            DO delete6
         CASE selection = "31"
            DO modify1
         CASE selection = "32"
            DO modify2
         CASE selection = "33"
            DO modify3
         CASE selection = "34"
            DO modify4
         CASE selection = "35"
            DO modify5
         CASE selection = "36"
            DO modify6
         OTHERWISE
            ? CHR(7)
            ERASE
            @ 10,17 SAY selection+" IS AN ILLEGAL SELECTION"
            @ 10,43 SAY ", PLEASE TRY AGAIN"
            SET TALK OFF
            STORE 1 TO delay
            DO WHILE delay < 40
               STORE delay+1 TO delay
            ENDDO WHILE
      ENDCASE
   ENDDO WHILE T
```

```
******** PROGRAM SUBMENU2 ********

* This program controls the Report Generators
*          function of the system

STORE T TO noexit
STORE F TO sub3
DO WHILE noexit
   SET PRINT OFF
   ERASE
   @  1,50 SAY DATE()
   @  2,13 SAY "************************************************"
   @  3,13 SAY "*                 S U B M E N U 2              *"
   @  4,13 SAY "*      R E P O R T S   A V A I L A B L E       *"
   @  5,13 SAY "************************************************"
   @  6,13 SAY "* 1=List of officers in alphabetical order      *"
   @  7,13 SAY "*                                              *"
   @  8,13 SAY "* 2=List of officers ordered by rank           *"
   @  9,13 SAY "*                                              *"
   @ 10,13 SAY "* 3=List of units with their officers          *"
   @ 11,13 SAY "*                                              *"
   @ 12,13 SAY "* 4=List of officers with non_military studies *"
   @ 13,13 SAY "*                                              *"
   @ 14,13 SAY "* 5=List of officers who know foreign languages*"
   @ 15,13 SAY "*                                              *"
   @ 16,13 SAY "* 6=List of officers including marital status  *"
   @ 17,13 SAY "*                                              *"
   @ 18,13 SAY "* 7=List of officers with service times        *"
   @ 19,13 SAY "*                                              *"
   @ 20,13 SAY "* 8=List of officers with addresses and phone# *"
   @ 21,13 SAY "************************************************"
   @ 22,13 SAY "      Enter the corresponding number OR         "
   @ 23,13 SAY "        enter 0 to exit to main menu            "
   STORE " " TO choice
   WAIT TO choice

*  Store choice to the "select" field of tempor to be used
*  by the monitor program

   SELECT SECONDARY
   USE tempor
   REPLACE select WITH choice
   USE

*  Accept the choice and branch to corresponding program

   DO CASE
      CASE choice = "0"
         RETURN
      CASE choice = "1"
         DO report1
      CASE choice = "2"
         DO report2
      CASE choice = "3"
```

```
            DO report3
         CASE choice = "4"
            DO report4
         CASE choice = "5"
            DO report5
         CASE choice = "6"
            DO report6
         CASE choice = "7"
            DO report7
         CASE choice = "8"
            DO report8
         OTHERWISE

*           Print an error message and give another chance

            ERASE
            ? CHR(7)
            @ 10,17 SAY choice+" IS AN ILLEGAL CHOICE - TRY AGAIN"
            STORE 1 TO del
            DO WHILE del < 40
               STORE del+1 TO del
            ENDDO WHILE
      ENDCASE
   ENDDO WHILE
```

```
******** PROGRAM SUBMENU3 ********
*
* This program controls the Miscellaneous function
*              of the system
*
ERASE
STORE T TO noexit
STORE T TO sub3
DO WHILE noexit
   ERASE
   SET PRINT OFF
   @  3,52 SAY DATE()
   @  4,16 SAY "*********************************************"
   @  5,16 SAY "*                                           *"
   @  6,16 SAY "*              S U B M E N U 3              *"
   @  7,16 SAY "*         MISCELLANEOUS  OPERATIONS         *"
   @  8,16 SAY "*                                           *"
   @  9,16 SAY "*********************************************"
   @ 10,16 SAY "*                                           *"
   @ 11,16 SAY "*        0  =  Exit to MAIN MENU            *"
   @ 12,16 SAY "*        1  =  Print MONITOR file           *"
   @ 13,16 SAY "*        2  =  Print STATISTIC file         *"
   @ 14,16 SAY "*        3  =  Delete MONITOR file          *"
   @ 15,16 SAY "*        4  =  Delete STATISTIC file        *"
   @ 16,16 SAY "*        5  =  Display an officer's data    *"
   @ 17,16 SAY "*              on the screen                *"
   @ 18,16 SAY "*********************************************"
   @ 20,16 SAY "       Enter the corresponding number        "
   STORE " " TO choice
   WAIT TO choice

*  Store choice to the "select" field of TEMPOR to be used
*  by the monitor program

   SELECT SECONDARY
   USE tempor
   REPLACE select WITH choice
   USE

*  Accept choice and branch to the corresponding program

   DO CASE
      CASE choice = "0"
         RETURN
      CASE choice = "1"
         DO printmon
      CASE choice = "2"
         DO printstat
      CASE choice = "3"
         DO delmon
      CASE choice = "4"
         DO delstat
      CASE choice = "5"
         DO dispscrn
```

```
      OTHERWISE

*        Print an error message and give another chance

         ERASE
         ? CHR(7)
         @ 11,17 SAY choice+" IS AN ILLEGAL CHOICE - TRY AGAIN"
         STORE 1 TO del
         DO WHILE del < 40
            STORE del+1 TO del
         ENDDO WHILE
   ENDCASE
ENDDO WHILE
```

B. PROGRAMS IMPLEMENTING THE UPDATE OPERATIONS

These programs permit the user to perform insertions, deletions, and modifications in the files of the database.

1. Insertion Programs

These programs are for adding records to the database files. The structure of all the programs follow the same logic, so we will describe only the program which adds records to the MASTER (PERSONNEL) file.

The whole structure of the program is as follows:

a. The MASTER file is opened.

b. The user is prompted to enter the new serial number or a blank in order to exit to SUBMENU1, when he is finished.

c. The program searches the MASTER file for the new serial number. If it already exists, prints an error message and gives another chance, otherwise it proceeds.

d. The proper memory variables, in which the new values are to be stored, are initialized to blanks. For simplicity the names of the memory variables are the same as the corresponding field names with the prefix 'm', which stands for memory. This holds throughout the whole system.

e. Then the program displays on the screen the new serial number, the names of the fields and the corresponding blank space. Then the user can enter the new values. Again the user has the opportunity to exit by pressing 'ENTER'.

f. Then a blank record is appended to the MASTER file and each field is replaced with the corresponding memory variable.

g. The whole operation is repeated within a WHILE loop, until the user exits.

h. The program MONITR which is for monitoring every job done in the database, is called only once before the program exits from either of the three possible exits, and after the number of changes is stored in the field 'counter' of the file TEMPOR. This is performed in all the programs that need to monitor the job done in the database.

The listings of the programs are given in the following pages.

```
******** PROGRAM INSERT1 ********

* This program adds records to MASTER(PERSONNEL) file

ERASE
SELECT PRIMARY
USE master INDEX master
STORE 0 TO count
STORE "***** ADDING RECORDS TO MASTER FILE *****" TO title1
STORE T TO continue
DO WHILE continue
   ERASE
   STORE "     " to mserno
   @ 5,60 SAY DATE()
   @ 7,5 SAY title1
   @ 10,5 SAY "ENTER A SERIAL NUMBER" GET mserno PICTURE "9999"
   @ 12,5 SAY "Press 'ENTER' to exit"
   READ
   IF $(mserno,1,1) = " "
      USE

      * Before the program returns, it stores the variable
      * count(# of changes) to the field 'counter' of TEMPOR
      * and records the changes. This is done in all three
      * exits of the program.

      IF count > 0
         SELECT SECONDARY
         USE tempor
         REPLACE counter WITH count
         USE
         DO monitr
      ENDIF
      RETURN
   ENDIF

   * Search the master file for the new serial number.
   * If it already exists print an error message, display
   * the record and let the user decide what to do.

   FIND &mserno
   IF # <> 0
      ? CHR(7)
      @ 14,5 SAY "The record already exists"
      @ 16,5 SAY "The current record is:"
      @ 18,5 SAY "serial number = "+serno
      @ 19,5 SAY "name          = "+name
      @ 21,5 SAY "TRY AGAIN?  (Y/N)"
      STORE " " TO answer
      WAIT TO answer
      IF answer = "y"
         LOOP
      ELSE
         USE
```

```
            IF count > 0
               SELECT SECONDARY
               USE tempor
               REPLACE counter WITH count
               USE
               DO monitr
            ENDIF
            RETURN
         ENDIF
      ENDIF

   * Initialize memory variables

      ERASE
      STORE "                    " TO mname
      STORE "   " TO mrank
      STORE "     " TO munit
      STORE "       " TO mserentry
      STORE "       " TO mreptdate
      STORE "   " TO mduty
      STORE "   " TO meducat
      STORE "  " TO mdegree
      STORE "    " TO mforlang
      STORE "  " TO mmarstat
      STORE "  " TO mchildren
      STORE "                         " TO maddress
      STORE "         " TO mphone
      @ 1,60 SAY DATE()
      @ 2,5 SAY title1
      @ 4,5 SAY "serial number = "+mserno

      *  Get the values for the new record

      @  5,5 SAY "name     " GET mname;
                             PICTURE "XXXXXXXXXXXXXXXX"
      @  6,5 SAY "rank     " GET mrank PICTURE "99"
      @  7,5 SAY "unit     " GET munit PICTURE "9999"
      @  8,5 SAY "serentry" GET mserentry PICTURE "999999"
      @  9,5 SAY "reptdate" GET mreptdate PICTURE "999999"
      @ 10,5 SAY "duty     " GET mduty PICTURE "99"
      @ 11,5 SAY "educat   " GET meducat PICTURE "99"
      @ 12,5 SAY "degree   " GET mdegree PICTURE "A"
      @ 13,5 SAY "forlang  " GET mforlang PICTURE "99"
      @ 14,5 SAY "marstat  " GET mmarstat PICTURE "A"
      @ 15,5 SAY "children" GET mchildren PICTURE "9"
      @ 16,5 SAY "address  " GET maddress;
                             PICTURE "XXXXXXXXXXXXXXXXXXXX"
      @ 17,5 SAY "phone    " GET mphone PICTURE "9999999"
      @ 20,5 SAY "Enter blanks to exit"
      READ
      IF $(mname,1,1) = " "
         USE
         IF count > 0
            SELECT SECONDARY
```

```
            USE tempor
            REPLACE counter WITH count
            USE
            DO monitr
         ENDIF
         RETURN
      ENDIF

* Append a new empty record in the master file and replace
* each field with the corresponding memory variable.

      APPEND BLANK
      REPLACE serno WITH mserno
      REPLACE name WITH mname
      REPLACE rank WITH mrank
      REPLACE unit WITH munit
      REPLACE serentry WITH mserentry
      REPLACE reptdate WITH mreptdate
      REPLACE duty WITH mduty
      REPLACE educat WITH meducat
      REPLACE degree WITH mdegree
      REPLACE forlang WITH mforlang
      REPLACE marstat WITH mmarstat
      REPLACE children WITH mchildren
      REPLACE address WITH maddress
      REPLACE phone WITH mphone
      STORE count+1 TO count
ENDDO WHILE continue
```

```
******** PROGRAM INSERT2 ********

* This program adds records to UNITS file

ERASE
SELECT PRIMARY
USE units INDEX units
STORE "***** ADDING RECORDS TO UNITS FILE *****" TO title2
STORE 0 TO count
STORE T TO continue
DO WHILE continue
   ERASE
   STORE "     " TO mcode
   @ 5,60 SAY DATE()
   @ 7,5 SAY title2
   @ 10,5 SAY "ENTER A UNIT CODE" GET mcode PICTURE "9999"
   @ 12,5 SAY "Press 'ENTER' to exit"
   READ
   IF $(mcode,1,1) = " "
      USE

      * Before the program returns, it updates the TEMPOR and
      * MONITOR files. This is done in all three exits of the
      * program

      IF count > 0
         SELECT SECONDARY
         USE TEMPOR
         REPLACE counter WITH count
         USE
         DO monitr
      ENDIF
      RETURN
   ENDIF

* Search the units file for the new unit code. If it already exists
* print an error message and give another chance otherwise proceed.

   FIND &mcode
   IF # <> 0
      ? CHR(7)
      @ 14,5 SAY "The record already exists"
      @ 16,5 SAY "The current record is:"
      @ 18,5 SAY "code     = "+code
      @ 19,5 SAY "title    = "+title
      @ 20,5 SAY "location = "+location
      @ 22,5 SAY "TRY AGAIN ?  (Y/N)"
      STORE " " TO answer
      WAIT TO answer
      IF answer = "y"
         LOOP
      ELSE
         USE
         IF count > 0
```

```
            SELECT SECONDARY
            USE tempor
            REPLACE counter WITH count
            USE
            DO monitr
         ENDIF
         RETURN
      ENDIF
   ENDIF

* Initialize memory variables

   ERASE
   STORE "                    " TO mtitle
   STORE "                    " TO mlocation
   @ 3,60 SAY DATE()
   @ 5,5 SAY title2
   @ 8,5 SAY "unit code = "+mcode
   @ 10,5 SAY "title    " GET mtitle PICTURE "XXXXXXXXXXXXXXX"
   @ 11,5 SAY "location" GET mlocation PICTURE "XXXXXXXXXXXX"
   @ 15,5 SAY "Enter blanks to exit"
   READ
   IF $(mtitle,1,1) = " "
      USE
      IF count > 0
         SELECT SECONDARY
         USE tempor
         REPLACE counter WITH count
         USE
         DO monitr
      ENDIF
      RETURN
   ENDIF

*  Append a blank record in the units file and replace
*  each field with the corresponding memory variables

   APPEND BLANK
   REPLACE code WITH mcode
   REPLACE title WITH mtitle
   REPLACE location WITH mlocation
   STORE count+1 TO count
ENDDO WHILE continue
```

```
******** PROGRAM INSERT3 ********

* This program adds records to RANKS file

ERASE
SELECT PRIMARY
USE ranks INDEX ranks
STORE "***** ADDING RECORDS TO RANKS FILE *****" TO title3
STORE 0 TO count
STORE T TO continue
DO WHILE continue
   ERASE
   STORE "  " TO mcode
   @ 5,60 SAY DATE()
   @ 7,5 SAY title3
   @ 10,5 SAY "ENTER A RANK CODE" GET mcode PICTURE "99"
   @ 12,5 SAY "Press 'ENTER' to exit"
   READ
   IF $(mcode,1,1) = " "
      USE
      IF count > 0
         SELECT SECONDARY
         USE tempor
         REPLACE counter WITH count
         USE
         DO monitr
      ENDIF
      RETURN
   ENDIF

*  Search the RANKS file for the new code. If it already
*  exists print an error message and give another chance
*  otherwise proceed

   FIND &mcode
   IF # <> 0
      ? CHR(7)
      @ 14,10 SAY " The record already exists"
      @ 16,5  SAY "The current record is:"
      @ 18,5  SAY "code      = "+code
      @ 19,5  SAY "armyname  = "+armyname
      @ 20,5  SAY "navyname  = "+navyname
      @ 22,5  SAY "TRY AGAIN ?  (Y/N)"
      STORE " " TO answer
      WAIT TO answer
      IF answer = "y"
         LOOP
      ELSE
         USE
         IF count > 0
            SELECT SECONDARY
            USE tempor
            REPLACE counter WITH count
            USE
```

```
            DO monitr
          ENDIF
          RETURN
       ENDIF
    ENDIF

*   Initialize memory variables

    ERASE
    STORE "     " TO marmyname
    STORE "     " TO mnavyname

*   Get the new values

    @ 4,60 SAY DATE()
    @ 5,5 SAY title3
    @ 8,5  SAY "Rank code = "+mcode
    @ 10,5 SAY "armyname" GET marmyname PICTURE "XXXX"
    @ 11,5 SAY "navyname" GET mnavyname PICTURE "XXXX"
    READ

*   Append the new record to the RANKS file

    APPEND BLANK
    REPLACE code WITH mcode
    REPLACE armyname WITH marmyname
    REPLACE navyname WITH mnavyname
    STORE count+1 TO count
ENDDO WHILE continue
```

```
********** PROGRAM INSERT4 **********

* This program adds records to DUTIES file

ERASE
SELECT PRIMARY
USE duties INDEX duties
STORE "***** ADDIND RECORDS TO DUTIES FILE *****" TO title4
STORE 0 TO count
STORE T TO continue
DO WHILE continue
   ERASE
   STORE "  " TO mcode
   @ 5,60 SAY DATE()
   @ 7,5  SAY title4
   @ 10,5 SAY "ENTER A DUTY CODE" GET mcode PICTURE "99"
   @ 12,5 SAY "Press 'ENTER' to exit"
   READ
   IF $(mcode,1,1) = " "
      USE
      IF count > 0
         SELECT SECONDARY
         USE tempor
         REPLACE counter WITH count
         USE
         DO monitr
      ENDIF
      RETURN
   ENDIF

* Search DUTIES file for the new code. If it exists print an
* error message and give another chance, otherwise proceed.

   FIND &mcode
   IF # <> 0
      ? CHR(7)
      @ 14,5  SAY "The record already exists"
      @ 16,5  SAY "The current record is:"
      @ 18,5  SAY "code = "+code
      @ 19,5  SAY "name = "+name
      @ 21,5  SAY "TRY AGAIN ?  (Y/N)"
      STORE " " TO answer
      WAIT TO answer
      IF answer = "y"
         LOOP
      ELSE
         USE
         IF count > 0
            SELECT SECONDARY
            USE tempor
            REPLACE counter WITH count
            USE
            DO monitr
         ENDIF
```

```
         RETURN
      ENDIF
   ENDIF
   STORE "                          " TO mname
   ERASE
   @ 5,60 SAY DATE()
   @ 7,5  SAY title4
   @ 10,5 SAY "duty code = "+mcode
   @ 12,5 SAY "name" GET mname PICTURE "XXXXXXXXXXXXXXXXXXX"
   READ

*  Append the record to DUTIES file

   APPEND BLANK
   REPLACE code WITH mcode
   REPLACE name WITH mname
   STORE count+1 TO count
ENDDO WHILE continue
```

```
******** PROGRAM INSERT5 ********

* This program adds records to FOREIGN LANGUAGES file

ERASE
SELECT PRIMARY
USE forlangs INDEX forlangs
STORE "***** ADDING RECORDS TO FORLANGS FILE *****" TO title5
STORE 0 TO count
STORE T TO continue
DO WHILE continue
   ERASE
   STORE "  " TO mcode
   @ 5,60 SAY DATE()
   @ 7,5 SAY title5
   @ 10,5 SAY "ENTER A LANGUAGE CODE" GET mcode PICTURE "99"
   @ 12,5 SAY "Press 'ENTER' to exit"
   READ
   IF $(mcode,1,1) = " "
      USE
      IF count > 0
         SELECT SECONDARY
         USE tempor
         REPLACE counter WITH count
         USE
         DO monitr
      ENDIF
      RETURN
   ENDIF

* Search the forlangs file for the new code. If it already
* exists, print an error message and give another change,
* otherwise proceed

   FIND &mcode
   IF # <> 0
      ? CHR(7)
      @ 14,5 SAY "The record already exists. Try again."
      @ 16,5 SAY "The current record is:"
      @ 18,5 SAY "code = "+code
      @ 19,5 SAY "name = "+name
      @ 21,5 SAY "TRY AGAIN?  (Y/N)"
      STORE " " TO answer
      WAIT TO answer
      IF answer = "y"
         LOOP
      ELSE
         USE
         IF count > 0
            SELECT SECONDARY
            USE tempor
            REPLACE counter WITH count
            USE
            DO monitr
```

```
         ENDIF
         RETURN
      ENDIF
   ENDIF

* Initialize memory variables.

   ERASE
   STORE "                " TO mname
   @ 4,60 SAY DATE()
   @ 5,5 SAY title5
   @ 8,5 SAY "language code = "+mcode

* Get the name for the new language.

  @ 10,5 SAY "name" GET mname PICTURE "XXXXXXXXXXXX"
  READ

* Append an empty record and replace the fields
* with the corresponding memory variables.

  APPEND BLANK
  REPLACE code WITH mcode
  REPLACE name WITH mname
  STORE count+1 TO count
ENDDO WHILE continue
```

```
******** PROGRAM INSERT6 ********

* This program adds records to SCIENCES file

ERASE
SELECT PRIMARY
USE sciences INDEX sciences
STORE "***** ADDIND RECORDS TO SCIENCES FILE *****" TO title6
STORE 0 TO count
STORE T TO continue
DO WHILE continue
   ERASE
   STORE "   " to mcode
   @ 5,60 SAY DATE()
   @ 7,5 SAY title6
   @ 10,5 SAY "ENTER A SCIENCE CODE" GET mcode PICTURE "99"
   @ 12,5 SAY "Press 'ENTER' to exit"
   READ
   IF $(mcode,1,1) = " "
      USE
      IF count > 0
         SELECT SECONDARY
         USE tempor
         REPLACE counter WITH count
         USE
         DO monitr
      ENDIF
      RETURN
   ENDIF

* Search the sciences file for the new code. If it already
* exists, print an error message and give another chance,
* otherwise proceed

   FIND &mcode
   IF # <> 0
      ? CHR(7)
      @ 14,5 SAY "The record already exists"
      @ 16,5 SAY "The current record is:"
      @ 18,5 SAY "code = "+code
      @ 19,5 SAY "name = "+name
      @ 21,5 SAY "TRY AGAIN?  (Y/N)"
      STORE " " TO answer
      WAIT TO answer
      IF answer = "y"
         LOOP
      ELSE
         USE
         IF count > 0
            SELECT SECONDARY
            USE tempor
            REPLACE counter WITH count
            USE
            DO monitr
```

```
         ENDIF
         RETURN
      ENDIF
   ENDIF

* Initialize memory variable

   ERASE
   STORE "                          " TO mname
   @ 4,60 SAY DATE()
   @ 5,5 SAY title6
   @ 8,5 SAY "science code = "+mcode

* Get the name for the new science

   @ 10,5 SAY "name" GET mname PICTURE "XXXXXXXXXXXXXXXXXXX"
   READ

* Append an empty record and replace the fields with
* the corresponding memory variable.

   APPEND BLANK
   REPLACE code WITH mcode
   REPLACE name WITH mname
   STORE count+1 TO count
ENDDO WHILE continue
```

2. Deletion Programs

These programs are for deleting records from the files of the database. Again we will describe only the program that performs deletions from the MASTER file since all the other programs have the same structure.

The structure of the program is as follows:

a. The MASTER file is opened.

b. The user is prompted to enter the serial number, which is the key value of the record to be deleted. He can also exit by pressing 'ENTER'.

c. When the user exits after he is finished, that is, after several deletions have been made, then the number of deletions which is kept in the variable 'count', is transferred to the field 'counter' of the file TEMPOR. Then the program MONITR is called to record the changes. If the user exits at the beginning, that is, without doing anything, then the program returns to SUBMENU1.

d. The MASTER file is searched for the serial number. If the record does not exist, the program prints an error message, and gives another chance. If the record exists, then it is displayed on the screen.

e. At this point the user is asked to confirm the deletion. If he is sure that the record must be deleted, he has just to enter 'Y', and the deletion operation is performed, otherwise he has to enter 'N' to abort the deletion. Then the program loops, repeating the whole operation.

The listings of the programs are given in the next pages.

```
******** PROGRAM DELETE1 ********

* This program deletes records from MASTER file

ERASE
SELECT PRIMARY
USE master INDEX master
STORE "***** DELETING RECORDS FROM MASTER FILE *****" TO title
STORE "     " TO mserno
STORE 0 TO count
STORE T TO continue
DO WHILE continue
   ERASE
   @ 5,60 SAY DATE()
   @ 7,5  SAY title
   @ 10,5 SAY "Enter serial number" GET mserno PICTURE "9999"
   @ 12,5 SAY "Press 'ENTER' to exit"
   READ
   IF $(mserno,1,1) = " "
      USE

      * If the user has already made deletions store the
      * number of deletions in the file TEMPOR, record
      * the changes and return to SUBMENU1, otherwise
      * just return

      IF count <> 0
         SELECT SECONDARY
         USE tempor
         REPLACE counter WITH count
         USE
         DO monitr
      ENDIF
      RETURN
   ENDIF

   * Search MASTER file for the serial number. If the
   * record does not exist print an error message and
   * give another chance, otherwise display the
   * record and let the user confirm the deletion.

   FIND &mserno
   IF # = 0
      ? CHR(7)
      @ 22,5 SAY "The record does not exist, try again"
      STORE 1 TO del
      DO WHILE del < 30
         STORE del+1 TO del
      ENDDO WHILE
      STORE "     " TO mserno
      LOOP
   ELSE
      ERASE
      @ 2,60 SAY DATE()
```

```
        @ 3,5  SAY title
        @ 5,5  SAY "Record to be deleted"
        @ 7,5  SAY "serno     = "+serno
        @ 8,5  SAY "name      = "+name
        @ 9,5  SAY "rank      = "+rank
        @ 10,5 SAY "unit      = "+unit
        @ 11,5 SAY "serentry  = "+serentry
        @ 12,5 SAY "reptdate  = "+reptdate
        @ 13,5 SAY "duty      = "+duty
        @ 14,5 SAY "educat    = "+educat
        @ 15,5 SAY "degree    = "+degree
        @ 16,5 SAY "forlang   = "+forlang
        @ 17,5 SAY "marstat   = "+marstat
        @ 18,5 SAY "children  = "+children
        @ 19,5 SAY "address   = "+address
        @ 20,5 SAY "phone     = "+phone
        ? CHR(7)
        @ 22,5 SAY "The record will be deleted. PROCEED ?  (Y/N)"
        STORE " " TO answer
        WAIT TO answer

        * If the user confirms the deletion some data
        * concerning the officer in question are
        * transfered to the STATISTIC file and then the
        * deletion is performed.

        IF answer = "y"
           SELECT SECONDARY
           USE statistic
           APPEND BLANK
           REPLACE serno WITH p.serno
           REPLACE name  WITH p.name
           REPLACE rank  WITH p.rank
           REPLACE unit  WITH p.unit
           REPLACE reptdate WITH ;
              $(p.reptdate,1,2)+"/"+$(p.reptdate,3,2)+"/";
              +$(p.reptdate,5,2)
           REPLACE deldate WITH DATE()
           USE
           SELECT PRIMARY
           DELETE
           PACK
           STORE count+1 TO count
        ENDIF
     ENDIF
     STORE "     " TO mserno
  ENDDO WHILE continue
```

```
******** PROGRAM DELETE2 ********

* This program deletes records from UNITS file

ERASE
SELECT PRIMARY
USE units INDEX units
STORE "***** DELETING RECORDS FROM UNITS FILE *****" TO tittle
STORE "     " TO mcode
STORE 0 TO count
STORE T TO continue
DO WHILE continue
   ERASE
   @ 5,60 SAY DATE()
   @ 7,5  SAY tittle
   @ 10,5 SAY "Enter unit code" GET mcode PICTURE "9999"
   @ 12,5 SAY "Press 'ENTER' to exit"
   READ
   IF $(mcode,1,1) = " "
      USE

*     If the user has already made deletions ,store the number
*     of deletions in the file TEMPOR, record the changes and
*     and return to SUBMENU1, otherwise just return.

      IF count <> 0
         SELECT SECONDARY
         USE tempor
         REPLACE counter WITH count
         USE
         DO monitr
      ENDIF
      RETURN
   ENDIF

*  Search UNITS file for the unit code. If the record does not
*  exist print an error message and give another chance,
*  otherwise display the record and let the user confirm
*  the deletion

   FIND &mcode
   IF # = 0
      ? CHR(7)
      @ 22,5 SAY " The record does not exist, try again"
      STORE 1 TO del
      DO WHILE del < 30
         STORE del+1 TO del
      ENDDO WHILE
      STORE "     " TO mcode
      LOOP
   ELSE
      ERASE
      @ 2,60 SAY DATE()
      @ 3,5  SAY tittle
```

```
        @ 5,5  SAY "Record to be deleted"
        @ 7,5  SAY "code     = "+code
        @ 8,5  SAY "title    = "+title
        @ 9,5  SAY "location = "+location
        ? CHR(7)
        @ 15,5 SAY "The record will be deleted. PROCEED ?  (Y/N)"
        STORE " " TO answer
        WAIT TO answer
        IF answer = "y"
           DELETE
           PACK
           STORE count+1 TO count
        ENDIF
      ENDIF
      STORE "     " TO mcode
ENDDO WHILE continue
```

```
******** PROGRAM DELETE3 ********

* This program deletes records from RANKS file

ERASE
CLEAR
SELECT PRIMARY
USE ranks INDEX ranks
STORE "***** DELETING RECORDS FROM RANKS FILE *****" TO title
STORE "   " TO mcode
STORE 0 TO count
STORE T TO continue
DO WHILE continue
   ERASE
   @ 5,60 SAY DATE()
   @ 7,5  SAY title
   @ 10,5 SAY "Enter rank code" GET mcode PICTURE "99"
   @ 12,5 SAY "Press 'ENTER' to exit"
   READ
   IF $(mcode,1,1) = " "
      USE

*     If the user has already made deletions, store the number
*     of deletions in the file TEMPOR, record the changes and
*     return to SUBMENU1, otherwise just return.

      IF count <> 0
         SELECT SECONDARY
         USE tempor
         REPLACE counter WITH count
         USE
         DO monitr
      ENDIF
      RETURN
   ENDIF

*  Search RANKS file for the code. If the record does not exist,
*  print an error message and give another chance, otherwise
*  display the record and let the user confirm the deletion.

   FIND &mcode
   IF # = 0
      ? CHR(7)
      @ 22,5 SAY "The record does not exist, try again"
      STORE 1 TO del
      DO WHILE del < 30
         STORE del+1 TO del
      ENDDO WHILE
      STORE "   " TO mcode
      LOOP
   ELSE
      ERASE
      @ 2,60 SAY DATE()
      @ 3,5  SAY title
```

```
         @ 5,5  SAY "Record to be deleted"
         @ 7,5  SAY "code      = "+code
         @ 8,5  SAY "armyname = "+armyname
         @ 9,5  SAY "navyname = "+navyname
         ? CHR(7)
         @ 15,10 SAY "The record will be deleted. PROCEED ?  (Y/N)"
         STORE " " TO answer
         WAIT TO answer
         IF answer = "y"
            DELETE
            PACK
            STORE count+1 TO count
         ENDIF
      ENDIF
      STORE "   " TO mcode
   ENDDO WHILE continue
```

```
******** PROGRAM DELETE4 ********

* This program deletes records from DUTIES file

ERASE
SELECT PRIMARY
USE duties INDEX duties
STORE "***** DELETING RECORDS FROM DUTIES FILE *****" TO title
STORE "   " TO mcode
STORE 0 TO count
STORE T TO continue
DO WHILE continue
   ERASE
   @ 5,60 SAY DATE()
   @ 7,5  SAY title
   @ 10,5 SAY "Enter duty code" GET mcode PICTURE "99"
   @ 12,5 SAY "Press 'ENTER' to exit"
   READ
   IF $(mcode,1,1) = " "
      USE

*     If the user has already made deletions store the number
*     of deletions in the file TEMPOR, record the changes and
*     return to SUBMENU1, otherwise just return.

      IF count <> 0
         SELECT SECONDARY
         USE tempor
         REPLACE counter WITH count
         USE
         DO monitr
      ENDIF
      RETURN
   ENDIF

*  Search DUTIES file for the duty code. If the record does not
*  exist, print an error message and give another chance, other-
*  wise display the record and let the user confirm the deletion

   FIND &mcode
   IF # = 0
      ? CHR(7)
      @ 22,5 SAY "The record does not exist, try again"
      STORE 1 TO del
      DO WHILE del < 30
         STORE del+1 TO del
      ENDDO WHILE
      STORE "   " TO mcode
      LOOP
   ELSE
      ERASE
      @ 2,60 SAY DATE()
      @ 3,5  SAY title
      @ 5,5  SAY "Record to be deleted"
```

```
        @ 7,5  SAY "code  = "+code
        @ 8,5  SAY "name  = "+name
        ? CHR(7)
        @ 15,5 SAY "The record will be deleted. PROCEED ?  (Y/N)"
        STORE " " TO answer
        WAIT TO answer
        IF answer = "y"
           DELETE
           PACK
           STORE count+1 TO count
        ENDIF
     ENDIF
     STORE "   " TO mcode
ENDDO WHILE continue
```

```
******** PROGRAM DELETE5 ********

* This program deletes records from FORLANGS file

ERASE
SELECT PRIMARY
USE forlangs INDEX forlangs
STORE "***** DELETING RECORDS FROM FORLANGS FILE *****" TO title
STORE "  " TO mcode
STORE 0 TO count
STORE T TO continue
DO WHILE continue
   ERASE
   @ 5,60 SAY DATE()
   @ 7,5  SAY title
   @ 10,5 SAY "Enter language code" GET mcode PICTURE "99"
   @ 12,5 SAY "Press 'ENTER' to exit"
   READ
   IF $(mcode,1,1) = " "
      USE

*     If the user has already made deletions, update the file
*     TEMPOR, record the changes and return to SUBMENU1,
*     otherwise just return.

      IF count <> 0
         SELECT SECONDARY
         USE tempor
         REPLACE counter WITH count
         USE
         DO monitr
      ENDIF
      RETURN
   ENDIF

*  Search FORLANGS file for the code. If the record does not
*  exist, print an error message and give another chance,
*  otherwise display the record and let the user confirm
*  the deletion

   FIND &mcode
   IF # = 0
      ? CHR(7)
      @ 22,5 SAY "The record does not exist, try again"
      STORE 1 TO del
      DO WHILE del < 30
         STORE del+1 TO del
      ENDDO WHILE
      STORE "  " TO mcode
      LOOP
   ELSE
      ERASE
      @ 2,60 SAY DATE()
      @ 3,5  SAY title
```

```
        @ 5,5  SAY "Record to be deleted"
        @ 7,5  SAY "code = "+code
        @ 8,5  SAY "name = "+name
        ? CHR(7)
        @ 15,5 SAY "The record will be deleted. PROCEED ?  (Y/N)"
        STORE " " TO answer
        WAIT TO answer
        IF answer = "y"
           DELETE
           PACK
           STORE count+1 TO count
        ENDIF
     ENDIF
     STORE "   " TO mcode
ENDDO WHILE continue
```

```
******** PROGRAM DELETE6 ********

* This program deletes records from SCIENCES file

ERASE
SELECT PRIMARY
USE sciences INDEX sciences
STORE "***** DELETING RECORDS FROM SCIENCES FILE *****" TO title
STORE "   " TO mcode
STORE 0 TO count
STORE T TO continue
DO WHILE continue
   ERASE
   @ 5,60 SAY DATE()
   @ 7,5  SAY title
   @ 10,5 SAY "Enter science code" GET mcode PICTURE "99"
   @ 12,5 SAY "Press 'ENTER' to exit"
   READ
   IF $(mcode,1,1) = " "
      USE

*     If the user has already made deletions, update the file
*     TEMPOR, record the changes and return to SUBMENU1,
*     otherwise just return.

      IF count <> 0
         SELECT SECONDARY
         USE tempor
         REPLACE counter WITH count
         USE
         DO monitr
      ENDIF
      RETURN
   ENDIF

*  Search SCIENCES file for the code. If the record does not
*  exist, print an error message and give another chance,
*  otherwise display the record and let the user confirm
*  the deletion

   FIND &mcode
   IF # = 0
      ? CHR(7)
      @ 22,5 SAY "The record does not exist, try again"
      STORE 1 TO del
      DO WHILE del < 30
         STORE del+1 TO del
      ENDDO WHILE
      STORE "   " TO mcode
      LOOP
   ELSE
      ERASE
      @ 2,60 SAY DATE()
      @ 3,5  SAY title
```

```
        @ 5,5  SAY "Record to be deleted"
        @ 7,5  SAY "code = "+code
        @ 8,5  SAY "name = "+name
        ? CHR(7)
        @ 15,5 SAY "The record will be deleted. PROCEED ?  (Y/N)"
        STORE " " TO answer
        WAIT TO answer
        IF answer = "y"
           DELETE
           PACK
           STORE count+1 TO count
        ENDIF
     ENDIF
     STORE "   " TO mcode
ENDDO WHILE continue
```

3. Modification Programs

These programs modify records in the files of the database. Since the structure of all the programs is the same, we will describe only the program MODIFY1 which makes changes in the MASTER file.

The structure of the program is as follows:

a. The MASTER file is opened.

b. An appropriate message is displayed on the screen, prompting the user to enter the serial number of the officer whose record is to be modified. At this point he may exit the program by pressing 'ENTER', either after he has made the modifications he wants, or at the beginning, without having done anything. In the latter case the program just returns to SUBMENU1. In the former case the program updates the MONITOR file and then returns. Before the program exits, it closes the MASTER file.

c. The MASTER file is searched for the serial number. If the record does not exist, the program displays an error message, and loops giving another chance.

d. If the record exists, then the field values are stored to temporary memory variables, and the record is displayed on the screen.

e. Now the user can modify any field he wants, by moving the cursor and overwriting the existing values.

f. When he is finished, the old field values are replaced by the modified memory variables, and the record is placed back in the file.

g. The whole operation continued under the control of a WHILE loop, until the user exits.

The listings of the programs are given in the following pages.

```
******** PROGRAM MODIFY1 ********

* This program modifies records in MASTER file

ERASE
SELECT PRIMARY
USE master INDEX master
STORE "***** MODIFYING RECORDS IN MASTER FILE *****" to TITLE
STORE 0 TO count
STORE T TO continue
DO WHILE continue
   ERASE
   STORE "     " TO mserno

   * Get the serial number

   @ 5,60 SAY DATE()
   @ 7,5  SAY title
   @ 10,5 SAY "Enter serial number" GET mserno PICTURE "9999"
   @ 12,5 SAY "Press 'ENTER' to exit"
   READ
   IF $(mserno,1,1) = " "
      USE

*     Update the TEMPOR file and record the changes

      IF count <> 0
         SELECT SECONDARY
         USE tempor
         REPLACE counter WITH count
         USE
         DO monitr
      ENDIF
      RETURN
   ENDIF

*  Look up MASTER file for the serial number

   FIND &mserno
   IF #=0
      ? CHR(7)
      @ 22,10 SAY "The record does not exist, try again"
      SET TALK OFF
      STORE 1 TO del
      DO WHILE del < 30
         STORE del+1 TO del
      ENDDO WHILE
      LOOP
   ENDIF

*  Store current field values to temporary memory variables

   STORE name TO mname
   STORE rank TO mrank
```

```
   STORE unit TO munit
   STORE serentry TO mserentry
   STORE reptdate TO mreptdate
   STORE duty TO mduty
   STORE educat TO meducat
   STORE degree TO mdegree
   STORE forlang TO mforlang
   STORE marstat TO mmarstat
   STORE children TO mchildren
   STORE address TO maddress
   STORE phone TO mphone

*  Display the record and get the new field values

   ERASE
   @ 1,60 SAY DATE()
   @ 2,5  SAY title
   @ 4,5  SAY "Serial number = "+mserno
   @ 6,5  SAY "Serno    " GET mserno
   @ 7,5  SAY "Name     " GET mname
   @ 8,5  SAY "Rank     " GET mrank
   @ 9,5  SAY "Unit     " GET munit
   @ 10,5 SAY "Serentry" GET mserentry
   @ 11,5 SAY "Reptdate" GET mreptdate
   @ 12,5 SAY "Duty     " GET mduty
   @ 13,5 SAY "Educat   " GET meducat
   @ 14,5 SAY "Degree   " GET mdegree
   @ 15,5 SAY "Forlang  " GET mforlang
   @ 16,5 SAY "Marstat  " GET mmarstat
   @ 17,5 SAY "Children" GET mchildren
   @ 18,5 SAY "Address  " GET maddress
   @ 19,5 SAY "Phone    " GET mphone
   READ

*  Replace old values with new values and put the record
*  back in the file

   REPLACE serno WITH mserno
   REPLACE name WITH mname
   REPLACE rank WITH mrank
   REPLACE unit WITH munit
   REPLACE serentry WITH mserentry
   REPLACE reptdate WITH mreptdate
   REPLACE duty WITH mduty
   REPLACE educat WITH meducat
   REPLACE degree WITH mdegree
   REPLACE forlang WITH mforlang
   REPLACE marstat WITH mmarstat
   REPLACE children WITH mchildren
   REPLACE address WITH maddress
   REPLACE phone WITH mphone
   STORE count+1 TO count
ENDDO WHILE continue
```

```
******** PROGRAM MODIFY2 ********

* This program modifies records in UNITS file

ERASE
SELECT PRIMARY
USE units INDEX units
STORE "***** MODIFYING RECORDS IN UNITS FILE *****" TO tittle
STORE 0 TO count
STORE T TO continue
DO WHILE continue
   ERASE
   STORE "     " TO mcode

   * Get the unit code

   @ 5,60  SAY DATE()
   @ 7,5   SAY tittle
   @ 10,5  SAY "Enter unit code" GET mcode PICTURE "9999"
   @ 12,5  SAY "Press 'ENTER' to exit"
   READ
   IF $(mcode,1,1) = " "
      USE

*     Update file TEMPOR and record the changes

      IF count <> 0
         SELECT SECONDARY
         USE tempor
         REPLACE counter WITH count
         USE
         DO monitr
      ENDIF
      RETURN
   ENDIF

*  Look up UNITS file for the code

   FIND &mcode
   IF #=0
      ? CHR(7)
      @ 22,10 SAY "The record does not exist, try again"
      SET TALK OFF
      STORE 1 TO del
      DO WHILE del < 30
         STORE del+1 TO del
      ENDDO WHILE
      LOOP
   ENDIF

*  Store current field values to temporary memory variables

   STORE title TO mtitle
   STORE location TO mlocation
```

```
*  Display the record and get the new field values

   ERASE
   @ 3,60  SAY DATE()
   @ 5,5   SAY tittle
   @ 8,5   SAY "Unit code = "+mcode
   @ 10,5  SAY "Code      " GET mcode
   @ 11,5  SAY "Title     " GET mtitle
   @ 12,5  SAY "Location  " GET mlocation
   READ

*  Replace old values with new values and put the record
*  back in the file

   REPLACE code WITH mcode
   REPLACE title WITH mtitle
   REPLACE location WITH mlocation
   STORE count+1 TO count
ENDDO WHILE continue
```

```
******** PROGRAM MODIFY3 ********

* This program modifies records in RANKS file

ERASE
SELECT PRIMARY
USE ranks INDEX ranks
STORE  "***** MODIFYING RECORDS IN RANKS FILE *****" to TITLE
STORE 0 TO count
STORE T TO continue
DO WHILE continue
   ERASE
   STORE "   " TO mcode

   * Get the rank code

   @ 5,60  SAY DATE()
   @ 7,5   SAY title
   @ 10,5  SAY "Enter rank code" GET mcode PICTURE "99"
   @ 12,5  SAY "Press 'ENTER' to exit"
   READ
   IF $(mcode,1,1) = " "
      USE

*     Monitor changes

      IF count <> 0
         SELECT SECONDARY
         USE tempor
         REPLACE counter WITH count
         USE
         DO monitr
      ENDIF
      RETURN
   ENDIF

*  Search RANKS file for the code

   FIND &mcode
   IF #=0
      ? CHR(7)
      @ 22,10 SAY "The record does not exist, try again"
      SET TALK OFF
      STORE 1 TO del
      DO WHILE del < 30
         STORE del+1 TO del
      ENDDO WHILE
      LOOP
   ENDIF

*  Store current field values to temporary memory variables

   STORE armyname TO marmyname
   STORE navyname TO mnavyname
```

```
*  Display the record and get the new values

   ERASE
   @ 3,60  SAY DATE()
   @ 5,5   SAY title
   @ 10,5  SAY "Rank code = "+mcode
   @ 12,5  SAY "Code     " GET mcode
   @ 13,5  SAY "Armyname" GET marmyname
   @ 14,5  SAY "Navyname" GET mnavyname
   READ

*  Replace old values with new values and put the record
*  back in the file

   REPLACE code WITH mcode
   REPLACE armyname WITH marmyname
   REPLACE navyname WITH mnavyname
   STORE count+1 TO count
ENDDO WHILE continue
```

```
 ******* PROGRAM MODIFY4 ********

* This program modifies records in DUTIES file

ERASE
SELECT PRIMARY
USE duties INDEX duties
STORE "***** MODIFYING RECORDS IN DUTIES FILE *****" TO title
STORE 0 TO count
STORE T TO continue
DO WHILE continue
   ERASE
   STORE "   " TO mcode

   * Get the duty code

   @ 5,60  SAY DATE()
   @ 7,5   SAY title
   @ 10,5  SAY "Enter duty code" GET mcode PICTURE "99"
   @ 12,5  SAY "Press 'ENTER' to exit"
   READ
   IF $(mcode,1,1) = " "
      USE

*     Monitor changes

      IF count <> 0
         SELECT SECONDARY
         USE tempor
         REPLACE counter WITH count
         USE
         DO monitr
      ENDIF
      RETURN
   ENDIF

*  Search DUTIES file for the code

   FIND &mcode
   IF #=0
      ? CHR(7)
      @ 22,10 SAY "The record does not exist, try again"
      SET TALK OFF
      STORE 1 TO del
      DO WHILE del < 30
         STORE del+1 TO del
      ENDDO WHILE
      LOOP
   ENDIF

*  Store current field values to temporary memory variables

   STORE name TO mname
```

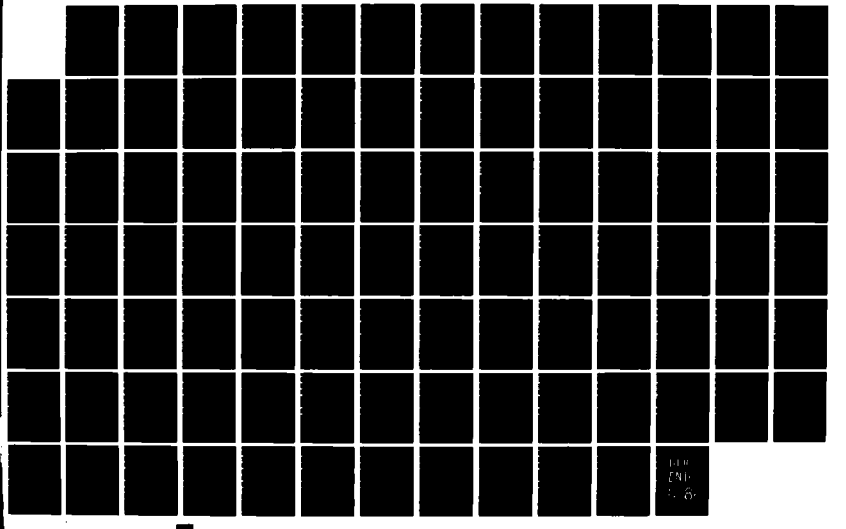
```
*  Display the record and get the new field values

   ERASE
   @ 3,60 SAY DATE()
   @ 4,5  SAY title
   @ 7,5  SAY "Duty code = "+mcode
   @ 9,5  SAY "code  " GET mcode
   @ 10,5 SAY "name  " GET mname
   READ

*  Replace old values with new values and put the record
*  back in the file

   REPLACE code WITH mcode
   REPLACE name WITH mname
   STORE count+1 TO count
ENDDO WHILE continue
```

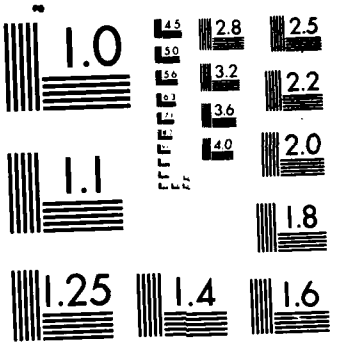
```
******** PROGRAM MODIFY5 ********

* This program modifies records in FORLANGS file

ERASE
SELECT PRIMARY
USE forlangs INDEX forlangs
STORE "***** MODIFYING RECORDS IN FORLANGS FILE *****" TO title
STORE 0 TO count
STORE T TO continue
DO WHILE continue
   ERASE
   STORE "   " TO mcode

   * Get the foreign language code

   @ 5,60  SAY DATE()
   @ 7,5   SAY title
   @ 10,5  SAY "Enter language code" GET mcode PICTURE "99"
   @ 12,5  SAY "Press 'ENTER' to exit"
   READ
   IF $(mcode,1,1) = " "
      USE

*     Monitor changes

      IF count <> 0
         SELECT SECONDARY
         USE tempor
         REPLACE counter WITH count
         USE
         DO monitr
      ENDIF
      RETURN
   ENDIF

*  Search FORLANGS file for the code

   FIND &mcode
   IF #=0
      ? CHR(7)
      @ 22,10 SAY "The record does not exist, try again"
      SET TALK OFF
      STORE 1 TO del
      DO WHILE del < 30
         STORE del+1 TO del
      ENDDO WHILE
      LOOP
   ENDIF

*  Store current field values to temporary memory variables

   STORE name TO mname
```

```
*  Display the record and get the new field values

   ERASE
   @ 3,60  SAY DATE()
   @ 4,5   SAY title
   @ 7,5   SAY "Code number = "+mcode
   @ 9,5   SAY "code   " GET mcode
   @ 10,5  SAY "name   " GET mname
   READ

*  Replace old values with new values and put the record
*  back in the file

   REPLACE code WITH mcode
   REPLACE name WITH mname
   STORE count+1 TO count
ENDDO WHILE continue
```

AD-A165 424 IMPLEMENTATION OF A PERSONNEL DATABASE SYSTEM IN HELLENIC ARMED FORCES FORMATIONS(U) NAVAL POSTGRADUATE SCHOOL MONTEREY CA P TSAGARIS ET AL. DEC 85 2/2

UNCLASSIFIED F/G 5/9 NL





```
******** PROGRAM MODIFY6 ********

* This program modifies records in SCIENCES file

ERASE
SELECT PRIMARY
USE sciences INDEX sciences
STORE "***** MODIFYING RECORDS IN SCIENCES FILE ***** " TO title
STORE 0 TO count
STORE T TO continue
DO WHILE continue
   ERASE
   STORE "   " TO mcode

   * Get the science code

   @ 5,60  SAY DATE()
   @ 7,5   SAY title
   @ 10,5  SAY "Enter code number" GET mcode PICTURE "99"
   @ 12,5  SAY "Press 'ENTER' to exit"
   READ
   IF $(mcode,1,1) = " "
      USE

*     Monitor changes

      IF count <> 0
         SELECT SECONDARY
         USE tempor
         REPLACE counter WITH count
         USE
         DO monitr
      ENDIF
      RETURN
   ENDIF

*  Search SCIENCES file for the code

   FIND &mcode
   IF #=0
      ? CHR(7)
      @ 22,10 SAY "The record does not exist, try again"
      SET TALK OFF
      STORE 1 TO del
      DO WHILE del < 30
         STORE del+1 TO del
      ENDDO WHILE
      LOOP
   ENDIF

*  Store current field values to temporary memory variables

   STORE name TO mname
```

```
*  Display the record and get the new field values

   ERASE
   @ 3,60 SAY DATE()
   @ 4,5  SAY title
   @ 7,5  SAY "Code number = "+mcode
   @ 9,5  SAY "code   " GET mcode
   @ 10,5 SAY "name   " GET mname
   READ

*  Replace old values with new values and put the record
*  back in the file

   REPLACE code WITH mcode
   REPLACE name WITH mname
   STORE count+1 TO count
ENDDO WHILE continue
```

C. REPORT GENERATORS

These programs permit the user to print all the necessary reports which must be available to the Commander, in order to help him in his decision making. The documentation, listing, and output of each program is as follows:

1. Program Report1

This program prints all the officers in alphabetical order including name, serial number, rank, the unit he belongs, and the report date.

The program uses the following structure:

a. The necessary memory variables are initialized and the heading of the report is printed.

b. The MASTER file is opened, and is sorted on the key 'name' to the temporary file TEMP, in ascending order.

c. The program uses the sorted file TEMP, and executes a WHILE loop to process each record of the file. Within the WHILE loop the program performs the following:

(1) Stores the appropriate field values to the corresponding memory variables.

(2) Searches RANKS and UNITS files and retrieves the appropriate rank name and the title of the unit.

(3) The main line of the output is printed. This line includes the data concerning the officer whose record is being processed. A line counter is kept to control the paging of the output.

d. The program continues with the next record until the end of file (EOF) is encountered.

e. The file TEMP is deleted, and the MONITOR file is updated.

The program listing and output is shown in the next pages.

```
******** PROGRAM REPORT1 ********

* This program prints all the officers in alphabetical order
* including nname, serial number, rank, the unit he belongs
* and the report date

ERASE

* Initialize memory variables

STORE "LIST OF OFFICERS IN ALPHABETICAL ORDER" TO tittle
STORE "======================================" TO underline
STORE 1 TO seqnum
STORE 0 TO linecount
STORE "                    " TO pname
STORE "     " TO pserno
STORE "       " TO preptdate
STORE "     " TO rankname
STORE "              " TO ptitle
STORE "   " TO mrank
STORE "     " TO munit
SET PRINT ON

* Print the heading

?           "                                                    "
?? "           "+DATE()
?           "                                                    "
?? "           --------"
?
?           "                "+tittle
?           "                "+underline
?
?           "-------------------------------------------"
?? "----------------------"
?           "¦SEQ#¦     N A M E      ¦ SERL#¦ RANK ¦   "
?? "U N I T    ¦ REPTDATE ¦"
?           "-------------------------------------------"
?? "----------------------"
STORE linecount+9 TO linecount
SELECT PRIMARY
USE master

* Sort MASTER file in alphabetical order of the officers'
* names into the temporary file TEMP, and use this file
* for the database processing

SORT ON name TO temp ASCENDING
USE temp
DO WHILE .NOT. EOF
   STORE name TO pname
   STORE serno TO pserno
   STORE reptdate TO preptdate
   STORE rank TO mrank
```

```
   STORE unit TO munit

*  Search RANKS file with key the field 'rank' of the
*  officer's records and get the appropriate rank name.

   SELECT SECONDARY
   USE ranks INDEX ranks
   FIND &mrank
   IF $(p.unit,1,1) = "1"
      STORE armyname TO rankname
   ELSE
      STORE navyname TO rankname
   ENDIF

*  Find the unit the officer belongs, and get the title
*  of the unit

   USE units INDEX units
   FIND &munit
   STORE title TO ptitle

*  If the line counter exceeds 53, continue in the next page.

   IF linecount > 53
      EJECT
      STORE 0 TO linecount
   ENDIF

*  Print the data concerning the officer

   ?            "¦    ¦                    ¦      ¦      ¦"
   ?? "              ¦          ¦"
   ?            "¦"+STR(seqnum,3)
   ?? "¦ "+pname
   ?? "¦ "+pserno
   ?? "¦ "+rankname
   ?? "¦ "+ptitle
   ?? "¦ "+$(preptdate,1,2)+"/"+$(preptdate,3,2)+"/";
         +$(preptdate,5,2)+" ¦"
   STORE seqnum+1 TO seqnum
   STORE linecount+2 TO linecount
   STORE "                    " TO pname
   STORE "     " TO pserno
   STORE "     " TO rankname
   STORE "              " TO ptitle
   STORE "         " TO preptdate
   STORE "   " TO mrank
   STORE "     " TO munit

*  Continue with the next record

   SELECT PRIMARY
   SKIP
ENDDO WHILE .NOT. EOF
```

```
* Update MONITOR file, delete file TEMP and return

DO monitr
DELETE FILE temp.DBF
RETURN
```

12/01/85

LIST OF OFFICERS IN ALPHABETICAL ORDER

SEQ#	NAME	SERL#	RANK	UNIT	REPTDATE
1	Adams Garry J	1030	CPT	2nd Inf Bn	09/20/84
2	Alkamo Jim P	1097	LTJG	LST Argo	07/20/83
3	Allen David E	1029	CPT	2nd Inf Bn	09/16/83
4	Boris Peter H	1084	LT	DD Miaoulis	07/16/84
5	Borrias Nick L	1083	LT	LST Rhodes	07/06/84
6	Brown Peter R	1028	CPT	1st Inf Bn	09/16/83
7	Bruce Mark P	1027	CPT	1st Inf Bn	08/15/84
8	Byron Larry M	1026	CPT	1st Inf Bn	08/21/84
9	Clark Tom L	1025	MAJ	Signals Bn	07/22/83
10	Cook Ryan K	1023	MAJ	Armour Bn	07/21/84
11	Crosby Jerry J	1022	MAJ	1st Arty Bn	07/20/83
12	Dalton John E	1021	MAJ	4th Inf Bn	07/20/83
13	Denton Mark S	1096	LTJG	LST Argo	07/19/84
14	Dorey Jim H	1019	MAJ	2nd Inf Bn	07/20/84
15	Durran Frank M	1020	MAJ	3rd Inf Bn	07/20/84
16	Emery Roy J	1018	MAJ	1st Inf Bn	07/18/83
17	Evans Isaac M	1065	1LT	2nd Arty Bn	07/16/83
18	Evans Tom M	1064	1LT	1st Arty Bn	07/15/83
19	Farmer Peter J	1063	1LT	5th Inf Bn	07/13/84
20	Faser Gas L	1062	1LT	4th Inf Bn	07/12/83
21	Felton John K	1061	1LT	3rd Inf Bn	07/10/84
22	Foley Mark L	1060	1LT	2nd Inf Bn	07/13/83

23	Ford	Roger	A	1057	CPT	Sup/Trans Bn	07/28/84
24	Foster	Kliff	J	1058	CPT	Sup/Trans Bn	07/29/83
25	Fox	Don	E	1099	LTJG	LST Rhodes	07/22/84
26	Frank	Paul	K	1059	1LT	1st Inf Bn	07/30/84
27	Franko	Jess	L	1098	LTJG	LST Rhodes	07/21/83
28	Freeman	Jim	E	1056	CPT	Sup/Trans Bn	07/27/83
29	Ganos	Jim	A	1085	LT	DD Miaoulis	07/10/84
30	Garret	Rex	D	1055	CPT	Signals Bn	07/26/83
31	Gilman	Perry	M	1054	CPT	Signals Bn	07/25/84
32	Good	John	L	1053	CPT	Signals Bn	07/24/84
33	Gorby	Glen	G	1051	CPT	Engineers Bn	07/22/84
34	Gordon	Jerry	W	1052	CPT	Engineers Bn	07/23/83
35	Gorman	Bruce	L	1095	LTJG	LST Argo	07/19/83
36	Hogan	Joe	K	1032	CPT	3rd Inf Bn	07/18/83
37	Horan	Kevin	L	1031	CPT	2nd Inf Bn	09/20/84
38	Ingals	Tomas	R	1035	CPT	4th Inf Bn	08/15/84
39	Jackson	Peter	M	1033	CPT	3rd Inf Bn	07/20/84
40	Jensen	Ron	P	1034	CPT	3rd Inf Bn	07/20/84
41	Jones	Peter	L	1100	LTJG	LST Rhodes	07/23/83
42	Kaan	Dave	J	1001	MG	1st Inf Div	07/20/84
43	Kamenos	Joe	M	1089	LT	DD kanaris	07/14/83
44	Kane	Bob	R	1002	BG	1st Inf Div	07/25/84
45	Karras	Mike	L	1005	LTC	1st Inf Div	07/25/84
46	Keen	Robt	N	1003	COL	1st Inf Div	08/26/83
47	King	Lewis	M	1007	LTC	1st Inf Bn	07/20/84
48	Kirk	Burt	N	1006	LTC	1st Inf Div	08/22/83
49	Kliff	Frank	E	1024	MAJ	Engineers Bn	07/21/84

50	Kontos	David	K	1088	LT	DD kanaris	07/13/83
51	Koom	Peter	H	1004	LTC	1st Inf Div	08/24/84
52	Kueny	John	S	1008	LTC	2nd Inf Bn	07/20/84
53	Larsen	Allen	A	1050	CPT	Engineers Bn	07/21/83
54	Larson	Roger	K	1049	CPT	Armour Bn	07/20/84
55	Layton	John	E	1048	CPT	Armour Bn	07/19/83
56	Lemos	Tim	N	1090	LT	DD kanaris	07/14/82
57	Lopez	Tom	L	1047	CPT	Armour Bn	07/13/84
58	Manos	John	K	1043	CPT	1st Arty Bn	07/07/84
59	Moore	Roger	L	1044	CPT	2nd Arty Bn	07/06/83
60	Morris	Roy	K	1046	CPT	2nd Arty Bn	07/13/83
61	Morton	Brian	C	1045	CPT	2nd Arty Bn	07/11/84
62	Newman	Ben	K	1066	1LT	Armour Bn	07/17/84
63	Newton	John	J	1067	1LT	Engineers Bn	07/18/83
64	Norton	Denis	E	1068	1LT	Signals Bn	07/10/84
65	Ocasio	Jim	E	1071	COMD	Navy Command	07/10/84
66	Odello	Bruno	A	1072	CAPT	DD Squadron	07/20/84
67	Olsen	Joe	N	1070	LTC	1st Inf Div	07/12/82
68	Onasis	George	E	1091	LTJG	DD kanaris	07/15/82
69	Oscar	Tom	K	1094	LTJG	DD Themis	07/18/83
70	Owens	Bill	L	1069	1LT	Sup/Trans Bn	07/11/83
71	Palmer	Bob	L	1011	LTC	5th Inf Bn	07/21/83
72	Pappas	Nick	C	1092	LTJG	DD Themis	07/16/82
73	Patton	Mike	H	1012	LTC	1st Arty Bn	07/22/83
74	Perry	Bill	H	1013	LTC	2nd Arty Bn	07/20/84
75	Peters	Mark	J	1009	LTC	3rd Inf Bn	07/18/83
76	Peters	Nick	K	1010	LTC	4th Inf Bn	07/19/84

77	Potter	Tom	E	1073	CAPT	LST Squadron	07/19/84
78	Quill	Kelvin	J	1075	CDR	DD kanaris	07/19/83
79	Quinn	Peter	C	1074	CDR	DD Miaoulis	07/12/83
80	Rigas	Ben	H	1093	LTJG	DD Themis	07/17/83
81	Rivera	Mario	L	1079	LCDR	DD Miaoulis	07/17/84
82	Roberts	Ben	J	1077	CDR	LST Argo	07/19/83
83	Rodes	James	K	1078	CDR	LST Rhodes	07/22/84
84	Rokos	Nick	A	1076	CDR	DD Themis	07/18/84
85	Ross	Allan	J	1080	LCDR	DD kanaris	07/25/84
86	Sanders	James	F	1014	LTC	Armour Bn	07/20/84
87	Scott	Paul	V	1015	LTC	Engineers Bn	08/15/83
88	Spencer	Tim	M	1081	LCDR	DD Themis	07/08/84
89	Stanley	Cris	K	1082	LCDR	LST Argo	07/09/83
90	Takas	Costas	L	1086	LT	DD Miaoulis	07/11/84
91	Torres	Alex	A	1017	LTC	Sup/Trans Bn	07/10/84
92	Turner	Carlos	B	1016	LTC	Signals Bn	08/15/83
93	Ullman	Rolf	G	1041	CPT	1st Arty Bn	07/03/83
94	Ulrey	Dan	A	1042	CPT	1st Arty Bn	07/06/84
95	Victor	David	R	1039	CPT	5th Inf Bn	07/20/83
96	Vongel	Mark	E	1040	CPT	5th Inf Bn	07/15/83
97	Warren	Geo	M	1036	CPT	4th Inf Bn	08/15/84
98	Waters	Gary	L	1038	CPT	5th Inf Bn	07/15/84
99	Watson	Ralph	D	1037	CPT	4th Inf Bn	07/20/83
100	Zikas	Tom	J	1087	LT	DD Miaoulis	07/12/84

2. Program Report2

This program prints all the officers ordered by rank, and within the rank by unit, including name, unit, and duty.

The structure of the program is as follows:

a. The necessary memory variables are initialized and the main heading of the list is printed.

b. We create a copy of the MASTER file into file TEMP, which includes only the fields necessary to obtain the output.

c. Since dBASE II cannot sort a file on more than one key at the same time, we do two consecutive sorts. The first one is done on the minor key 'unit', and the second one on the major key 'rank'. For the purpose of these sorts we use the temporary files TEMP1 and TEMP2 besides TEMP.

d. Using the file TEMP2 which is double sorted on the desired keys, the program performs a WHILE loop to process this file, with the following major functions:

(1) The field "rank" and the first character of the unit code are compared to the memory variables 'mrank' and 'funit' respectively. If either comparison yields a 'false' value this means that an officer of the different rank or different service has been encountered. The value of the field 'rank', and the first character of the unit code, are stored to the above memory variables, in order to be ready for the next comparison. Then the heading of the new rank is printed.

(2) The current field values are stored to the corresponding memory variables.

(3) The appropriate unit title, and duty name are retrieved from the files UNITS and DUTIES respectively.

(4) The main line of the output is printed, and the program proceeds to the next record.

e. The MONITOR file is updated, all the temporary files are deleted, and then the program returns to SUBMENU2.

The listing and output of the program are shown in the next pages.

```
******** PROGRAM REPORT2 ********

* This program prints all the officers ordered by rank
* and unit number including name, unit and duty.

ERASE

* Initialize memory variables

STORE 0 TO linecount
STORE "                    " TO pname
STORE " " TO funit
STORE "  " TO mrank
STORE "    " TO munit
STORE "              " TO punit
STORE "  " TO mduty
STORE "                        " TO pduty
STORE T TO maintitle
STORE "========================================" TO line1
STORE "======================" TO line2
STORE "****************************************" TO line3
STORE "**********************" TO line4

* Print the heading

SET PRINT ON
?            "                                                    "
?? "      "+DATE()
?            "                                                    "
?? "      --------"
?            "                LIST OF OFFICERS ORDERED BY RANK"
?            "                ================================"
?
? line3+line4
?            "*SEQ#¦     N A M E      ¦    U N I T   ¦       "
?? "D U T Y       *"
? line3+line4
STORE linecount+8 TO linecount

* Create a copy of the MASTER file into TEMP including only
* the appropriate fields and sort TEMP file on the minor key
* 'unit' first into TEMP1 and then use TEMP1 to sort the file
* on the major key 'rank' into TEMP2

SELECT PRIMARY
USE master
COPY TO temp FIELD serno,name,rank,unit,duty
USE temp
SORT ON unit TO temp1
USE temp1
SORT ON rank TO temp2
USE temp2
DO WHILE .NOT. EOF
```

```
*  If a new rank or an officer of a different service
*  is encountered print the rank heading

   IF rank <> mrank .OR. $(unit,1,1) <> funit
      STORE rank TO mrank
      STORE $(unit,1,1) TO funit
      STORE 1 TO seqno
      STORE T TO flag
         ?
         ?
         STORE linecount+2 TO linecount
      IF linecount > 53
         EJECT
         STORE 0 TO linecount
      ENDIF
      DO CASE
         CASE rank = "01"
            IF $(unit,1,1) = "1"
               ? line1+line2
               ? "                          G E N E R A L S     "
               ? line1+line2
            ELSE
               ? line1+line2
               ? "                          A D M I R A L S     "
               ? line1+line2
            ENDIF
         CASE rank = "02"
            IF $(unit,1,1) = "1"
               ? line1+line2
               ? "                       L T   G E N E R A L S "
               ? line1+line2
            ELSE
               ? line1+line2
               ? "                     V I C E   A D M I R A L S"
               ? line1+line2
            ENDIF
         CASE rank = "03"
            IF $(unit,1,1) = "1"
               ? line1+line2
               ? "                        M A J   G E N E R A L S"
               ? line1+line2
            ELSE
               ? line1+line2
               ? "                     R E A R   A D M I R A L S"
               ? line1+line2
            ENDIF
         CASE rank = "04"
            IF $(unit,1,1) =  "1"
               ? line1+line2
               ? "                     B R I G   G E N E R A L S"
               ? line1+line2
            ELSE
               ? line1+line2
               ? "                          C O M M O D O R E S"
```

```
      ? line1+line2
   ENDIF
CASE rank = "05"
   IF $(unit,1,1) = "1"
      ? line1+line2
      ? "                              C O L O N E L S"
      ? line1+line2
   ELSE
      ? line1+line2
      ? "                              C A P T A I N S"
      ? line1+line2
   ENDIF
CASE rank = "06"
   IF $(unit,1,1) = "1"
      ? line1+line2
      ? "                           L T  C O L O N E L S"
      ? line1+line2
   ELSE
      ? line1+line2
      ? "                           C O M M A N D E R S"
      ? line1+line2
   ENDIF
CASE rank = "07"
   IF $(unit,1,1) = "1"
      ? line1+line2
      ? "                              M A J O R S"
      ? line1+line2
   ELSE
      ? line1+line2
      ? "                        L T  C O M M A N D E R S"
      ? line1+line2
   ENDIF
CASE rank = "08"
   IF $(unit,1,1) = "1"
      ? line1+line2
      ? "                              C A P T A I N S"
      ? line1+line2
   ELSE
      ? line1+line2
      ? "                           L I E U T E N A N T S"
      ? line1+line2
   ENDIF
CASE rank = "09"
      ? line1+line2
      ? "                        1st  L I E U T E N A N T S"
      ? line1+line2
CASE rank = "10"
   IF $(unit,1,1) = "1"
      ? line1+line2
      ? "                        2nd  L I E U T E N A N T S"
      ? line1+line2
   ELSE
      ? line1+line2
      ? "                              E N S I G N S"
```

```
               ? line1+line2
            ENDIF
         OTHERWISE
            ERASE
            STORE F TO flag
            ? CHR(7)
            @ 10,10 SAY "ILLEGAL RANK CODE ENCOUNTERED IN RECORD"
            @ 12,10 SAY "       WITH SERIAL NUMBER "+serno
            STORE 1 TO del
            DO WHILE del < 50
               STORE del+1 To del
            ENDDO WHILE
      ENDCASE
      IF flag
         STORE linecount+3 TO linecount
      ENDIF
   ENDIF

*  Store field values to temporary  memory variables

   STORE name TO pname
   STORE unit TO munit
   STORE duty TO mduty
   SELECT SECONDARY
   USE units INDEX units

* Find and retrieve the unit title from UNITS file

   FIND &munit
   IF # <> 0
      STORE title TO punit
   ELSE
      ERASE
      ? CHR(7)
      @ 9,9 SAY "Record with key "+munit+" in UNITS file not found"
      STORE 1 TO del
      DO WHILE del < 50
         STORE del+1 TO del
      ENDDO WHILE
   ENDIF

*  Find and retrieve the duty name from DUTIES file

   USE duties INDEX duties
   FIND &mduty
   IF # <> 0
      STORE name TO pduty
   ELSE
     ERASE
     ? CHR(7)
     @ 9,9 SAY "Record with key "+mduty+" in DUTIES file not found"
     STORE 1 TO del
     DO WHILE del < 50
        STORE del+1 TO del
```

```
      ENDDO WHILE
   ENDIF

*  Print the main line

   IF linecount > 53
      EJECT
      STORE 0 TO linecount
   ENDIF
   ?              "¦     ¦                        ¦                  ¦"
   ?? "                          ¦"
   ?              "¦"+STR(seqno,3)
   ?? "¦ "+pname
   ?? "¦ "+punit
   ?? "¦ "+pduty+"¦"
   STORE linecount+2 TO linecount
   STORE seqno+1 TO seqno

   * Continue with the next record

   SELECT PRIMARY
   SKIP
ENDDO WHILE .NOT. EOF

* Update the MONITOR file and delete the temporary files

DO monitr
DELETE FILE temp.DBF
DELETE FILE temp1.DBF
DELETE FILE temp2.DBF
RETURN
```

12/01/85

LIST OF OFFICERS ORDERED BY RANK

*SEQ#	NAME	UNIT	DUTY
MAJ GENERALS			
1	Kaan Dave J	1st Inf Div	Commander
BRIG GENERALS			
1	Kane Bob R	1st Inf Div	Dty Commander
COMMODORES			
1	Ocasio Jim E	Navy Command	Commander
COLONELS			
1	Keen Robt N	1st Inf Div	Chief of Staff
CAPTAINS			
1	Odello Bruno A	DD Squadron	Commander
2	Potter Tom E	LST Squadron	Commander
LT COLONELS			
1	Koom Peter H	1st Inf Div	1st Office Manager
2	Karras Mike L	1st Inf Div	2nd Office Manager

3	Kirk	Burt	N	1st Inf Div	3rd Office Manager
4	Olsen	Joe	N	1st Inf Div	4th Office Manager
5	King	Lewis	M	1st Inf Bn	Commander
6	Kueny	John	S	2nd Inf Bn	Commander
7	Peters	Mark	J	3rd Inf Bn	Commander
8	Peters	Nick	K	4th Inf Bn	Commander
9	Palmer	Bob	L	5th Inf Bn	Commander
10	Patton	Mike	H	1st Arty Bn	Commander
11	Perry	Bill	H	2nd Arty Bn	Commander
12	Sanders	James	F	Armour Bn	Commander
13	Scott	Paul	V	Engineers Bn	Commander
14	Turner	Carlos	B	Signals Bn	Commander
15	Torres	Alex	A	Sup/Trans Bn	Commander

C O M M A N D E R S

1	Quinn	Peter	C	DD Miaoulis	Commanding Officer
2	Quill	Kelvin	J	DD kanaris	Commanding Officer
3	Rokos	Nick	A	DD Themis	Commanding Officer
4	Roberts	Ben	J	LST Argo	Commanding Officer
5	Rodes	James	K	LST Rhodes	Commanding Officer

M A J O R S

1	Emery	Roy	J	1st Inf Bn	Dty Commander
2	Dorey	Jim	H	2nd Inf Bn	Dty Commander
3	Durran	Frank	M	3rd Inf Bn	Dty Commander
4	Dalton	John	E	4th Inf Bn	Dty Commander

5	Crosby	Jerry	J	1st Arty Bn	Dty Commander
6	Cook	Ryan	K	Armour Bn	Dty Commander
7	Kliff	Frank	E	Engineers Bn	Dty Commander
8	Clark	Tom	L	Signals Bn	Dty Commander

L T C O M M A N D E R S

1	Rivera	Mario	L	DD Miaoulis	Executive Officer
2	Ross	Allan	J	DD kanaris	Executive Officer
3	Spencer	Tim	M	DD Themis	Executive Officer
4	Stanley	Cris	K	LST Argo	Executive Officer

C A P T A I N S

1	Byron	Larry	M	1st Inf Bn	Company Commander
2	Bruce	Mark	P	1st Inf Bn	Company Commander
3	Brown	Peter	R	1st Inf Bn	Company Commander
4	Allen	David	E	2nd Inf Bn	Company Commander
5	Adams	Garry	J	2nd Inf Bn	Company Commander
6	Horan	Kevin	L	2nd Inf Bn	Company Commander
7	Hogan	Joe	K	3rd Inf Bn	Company Commander
8	Jackson	Peter	M	3rd Inf Bn	Company Commander
9	Jensen	Ron	P	3rd Inf Bn	Company Commander
10	Ingals	Tomas	R	4th Inf Bn	Company Commander
11	Warren	Geo	M	4th Inf Bn	Company Commander
12	Watson	Ralph	D	4th Inf Bn	Company Commander
13	Waters	Gary	L	5th Inf Bn	Company Commander
14	Victor	David	R	5th Inf Bn	Company Commander

15	Vongel	Mark	E	5th Inf Bn	Company Commander
16	Ullman	Rolf	G	1st Arty Bn	Battery Commander
17	Ulrey	Dan	A	1st Arty Bn	Battery Commander
18	Manos	John	K	1st Arty Bn	Battery Commander
19	Moore	Roger	L	2nd Arty Bn	Battery Commander
20	Morton	Brian	C	2nd Arty Bn	Battery Commander
21	Morris	Roy	K	2nd Arty Bn	Battery Commander
22	Lopez	Tom	L	Armour Bn	Company Commander
23	Layton	John	E	Armour Bn	Company Commander
24	Larson	Roger	K	Armour Bn	Company Commander
25	Larsen	Allen	A	Engineers Bn	Company Commander
26	Gorby	Glen	G	Engineers Bn	Company Commander
27	Gordon	Jerry	W	Engineers Bn	Company Commander
28	Good	John	L	Signals Bn	Company Commander
29	Gilman	Perry	M	Signals Bn	Company Commander
30	Garret	Rex	D	Signals Bn	Company Commander
31	Freeman	Jim	E	Sup/Trans Bn	Company Commander
32	Ford	Roger	A	Sup/Trans Bn	Company Commander
33	Foster	Kliff	J	Sup/Trans Bn	Company Commander

L I E U T E N A N T S

1	Boris	Peter	H	DD Miaoulis	ASW Officer
2	Ganos	Jim	A	DD Miaoulis	Navigation Officer
3	Takas	Costas	L	DD Miaoulis	Commun/tion Officer
4	Zikas	Tom	J	DD Miaoulis	CIC Officer
5	Kontos	David	K	DD kanaris	ASW Officer

6	Kamenos	Joe	M	DD kanaris	Navigation Officer
7	Lemos	Tim	N	DD kanaris	Commun/tion Officer
8	Borrias	Nick	L	LST Rhodes	Executive Officer

1st L I E U T E N A N T S

1	Frank	Paul	K	1st Inf Bn	Adjutant
2	Foley	Mark	L	2nd Inf Bn	Adjutant
3	Felton	John	K	3rd Inf Bn	Adjutant
4	Faser	Gas	L	4th Inf Bn	Adjutant
5	Farmer	Peter	J	5th Inf Bn	Adjutant
6	Evans	Tom	M	1st Arty Bn	Adjutant
7	Evans	Isaac	M	2nd Arty Bn	Adjutant
8	Newman	Ben	K	Armour Bn	Adjutant
9	Newton	John	J	Engineers Bn	Adjutant
10	Norton	Denis	E	Signals Bn	Adjutant
11	Owens	Bill	L	Sup/Trans Bn	Adjutant

1st L I E U T E N A N T S

1	Onasis	George	E	DD kanaris	CIC Officer
2	Pappas	Nick	C	DD Themis	ASW Officer
3	Rigas	Ben	H	DD Themis	Commun/tion Officer
4	Oscar	Tom	K	DD Themis	CIC Officer
5	Gorman	Bruce	L	LST Argo	Navigation Officer
6	Denton	Mark	S	LST Argo	Commun/tion Officer
7	Alkamo	Jim	P	LST Argo	CIC Officer
8	Franko	Jess	L	LST Rhodes	Navigation Officer

9	Fox	Don	E	LST Rhodes	Commun/tion Officer
10	Jones	Peter	L	LST Rhodes	CIC Officer

3. Program Report3

This program prints all the units and the officers who belong to a particular unit, including name, rank, and duty.

The structure of the program is as follows:

a. The necessary memory variables are initialized and the headlines are printed.

b. A copy of the MASTER file is created which includes the fields, serial number, name, rank, unit, and duty. This file then is sorted by unit, and within the unit by rank.

c. The above finally sorted file, is processed sequentially within a WHILE loop, whose function is the following:

(1) The field 'unit' is compared to the field 'unit' of the previous record. If they are not equal, this means that a new unit has been encountered. Then the program prints the title and location of the unit.

(2) The rank and duty names are retrieved from the corresponding files, and the main line of the output is printed.

(3) The program proceeds with the next record.

d. Before the program returns, it updates the MONITOR file, and deletes the temporary files.

The listings and the output of the program are shown in the next pages.

```
******** PROGRAM REPORT3 ********

* This program prints all the units with their
* officers, including name, rank, and duty

ERASE

* Initialize memory variables

STORE 0 TO linecount
STORE "                    " TO pname
STORE "   " TO mrank
STORE "     " TO prank
STORE "   " TO mduty
STORE "                         " TO pduty
STORE "     " TO munit
STORE "               " TO punit
STORE "             " TO plocat
STORE 1 TO seqnum
STORE "**********************************" TO line1
STORE "************************" TO line2
STORE "========================================" TO line3

* Print the heading

SET PRINT ON
?           "                                                    "
?? DATE()
?           "                                                    "
?? "--------"
?           "              LIST OF UNITS WITH THEIR OFFICERS"
?           "              ================================="
?
? line1+line2
?           "*SEQ#|     N A M E        | RANK |"
?? "      D U T Y          *"
? line1+line2

STORE linecount+8 TO linecount

* Create a copy of the MASTER file into TEMP including
* only the necessary fields and sort TEMP file on
* ascending keys 'unit' and 'rank'

SELECT PRIMARY
USE master
COPY TO temp FIELD serno,name,rank,unit,duty
USE temp
SORT ON rank TO temp1
USE temp1
SORT ON unit TO temp2
USE temp2
DO WHILE .NOT. EOF
```

```
* If the current unit is different than the previous
* one print the unit heading

IF unit <> munit
   STORE unit TO munit
   STORE 1 TO seqnum
   IF linecount > 53
      EJECT
      STORE 0 TO linecount
   ENDIF
   ?
   ?
   STORE linecount+2 TO linecount
   IF linecount > 53
      EJECT
      STORE 0 TO linecount
   ENDIF
   SELECT SECONDARY
   USE units INDEX units
   FIND &munit
   ?              "       "+line3
   ?              "       UNIT: "+title+"  LOCATION: "+location
   ?              "       "+line3
   STORE linecount+3 TO linecount
ENDIF
SELECT PRIMARY

* Store current field values to memory variables

STORE name TO pname
STORE rank TO mrank
STORE duty TO mduty

* Retrieve rank and duty names

SELECT SECONDARY
USE ranks INDEX ranks
FIND &mrank
IF $(p.unit,1,1) = "1"
   STORE armyname TO prank
ELSE
   STORE navyname TO prank
ENDIF
USE duties INDEX duties
FIND &mduty
STORE name TO pduty

* Print the main line

IF linecount > 53
   EJECT
   STORE 0 TO linecount
ENDIF
?          "¦    ¦                    ¦      ¦"
```

```
      ?? "                              ¦"
      ?           "¦"+STR(seqnum,3)
      ?? "¦ "+pname
      ?? " ¦ "+prank
      ?? "¦ "+pduty+"   ¦"
      STORE linecount+2 TO linecount
      STORE seqnum+1 TO seqnum

      * Continue with the next record

      SELECT PRIMARY
      SKIP
ENDDO WHILE .NOT. EOF

* Update the MONITOR file and delete
* the temporary files

DO monitr
DELETE FILE temp.DBF
DELETE FILE temp1.DBF
DELETE FILE temp2.DBF
RETURN
```

12/01/85

LIST OF UNITS WITH THEIR OFFICERS

*SEQ#	N A M E			RANK	D U T Y
UNIT: 1st Inf Div LOCATION: Salinas					
1	Kaan	Dave	J	MG	Commander
2	Kane	Bob	R	BG	Dty Commander
3	Keen	Robt	N	COL	Chief of Staff
4	Koom	Peter	H	LTC	1st Office Manager
5	Karras	Mike	L	LTC	2nd Office Manager
6	Kirk	Burt	N	LTC	3rd Office Manager
7	Olsen	Joe	N	LTC	4th Office Manager
UNIT: 1st Inf Bn LOCATION: Monterey					
1	King	Lewis	M	LTC	Commander
2	Emery	Roy	J	MAJ	Dty Commander
3	Byron	Larry	M	CPT	Company Commander
4	Bruce	Mark	P	CPT	Company Commander
5	Brown	Peter	R	CPT	Company Commander
6	Frank	Paul	K	1LT	Adjutant
UNIT: 2nd Inf Bn LOCATION: Carmel					
1	Kueny	John	S	LTC	Commander
2	Dorey	Jim	H	MAJ	Dty Commander

3	Allen	David	E	CPT	Company Commander
4	Adams	Garry	J	CPT	Company Commander
5	Horan	Kevin	L	CPT	Company Commander
6	Foley	Mark	L	1LT	Adjutant

UNIT: 3rd Inf Bn LOCATION: Salinas

1	Peters	Mark	J	LTC	Commander
2	Durran	Frank	M	MAJ	Dty Commander
3	Hogan	Joe	K	CPT	Company Commander
4	Jackson	Peter	M	CPT	Company Commander
5	Jensen	Ron	P	CPT	Company Commander
6	Felton	John	K	1LT	Adjutant

UNIT: 4th Inf Bn LOCATION: Seaside

1	Peters	Nick	K	LTC	Commander
2	Dalton	John	E	MAJ	Dty Commander
3	Ingals	Tomas	R	CPT	Company Commander
4	Warren	Geo	M	CPT	Company Commander
5	Watson	Ralph	D	CPT	Company Commander
6	Faser	Gas	L	1LT	Adjutant

UNIT: 5th Inf Bn LOCATION: Pac Grove

1	Palmer	Bob	L	LTC	Commander
2	Waters	Gary	L	CPT	Company Commander
3	Victor	David	R	CPT	Company Commander
4	Vongel	Mark	E	CPT	Company Commander

5	Farmer	Peter	J	1LT	Adjutant

UNIT: 1st Arty Bn LOCATION: Monterey

1	Patton	Mike	H	LTC	Commander
2	Crosby	Jerry	J	MAJ	Dty Commander
3	Ullman	Rolf	G	CPT	Battery Commander
4	Ulrey	Dan	A	CPT	Battery Commander
5	Manos	John	K	CPT	Battery Commander
6	Evans	Tom	M	1LT	Adjutant

UNIT: 2nd Arty Bn LOCATION: Santa Cruz

1	Perry	Bill	H	LTC	Commander
2	Moore	Roger	L	CPT	Battery Commander
3	Morton	Brian	C	CPT	Battery Commander
4	Morris	Roy	K	CPT	Battery Commander
5	Evans	Isaac	M	1LT	Adjutant

UNIT: Armour Bn LOCATION: Salinas

1	Sanders	James	F	LTC	Commander
2	Cook	Ryan	K	MAJ	Dty Commander
3	Lopez	Tom	L	CPT	Company Commander
4	Layton	John	E	CPT	Company Commander
5	Larson	Roger	K	CPT	Company Commander
6	Newman	Ben	K	1LT	Adjutant

UNIT: Engineers Bn LOCATION: Monterey

1	Scott	Paul	V	LTC	Commander
2	Kliff	Frank	E	MAJ	Dty Commander
3	Larsen	Allen	A	CPT	Company Commander
4	Gorby	Glen	G	CPT	Company Commander
5	Gordon	Jerry	W	CPT	Company Commander
6	Newton	John	J	1LT	Adjutant

UNIT: Signals Bn LOCATION: Pac Grove

1	Turner	Carlos	B	LTC	Commander
2	Clark	Tom	L	MAJ	Dty Commander
3	Good	John	L	CPT	Company Commander
4	Gilman	Perry	M	CPT	Company Commander
5	Garret	Rex	D	CPT	Company Commander
6	Norton	Denis	E	1LT	Adjutant

UNIT: Sup/Trans Bn LOCATION: Carmel

1	Torres	Alex	A	LTC	Commander
2	Freeman	Jim	E	CPT	Company Commander
3	Ford	Roger	A	CPT	Company Commander
4	Foster	Kliff	J	CPT	Company Commander
5	Owens	Bill	L	1LT	Adjutant

UNIT: Navy Command LOCATION: Monterey

1	Ocasio	Jim	E	COMD	Commander

UNIT: DD Squadron LOCATION: Monterey

1	Odello Bruno A	CAPT	Commander

UNIT: DD Miaoulis LOCATION: Monterey

1	Quinn Peter C	CDR	Commanding Officer
2	Rivera Mario L	LCDR	Executive Officer
3	Boris Peter H	LT	ASW Officer
4	Ganos Jim A	LT	Navigation Officer
5	Takas Costas L	LT	Commun/tion Officer
6	Zikas Tom J	LT	CIC Officer

UNIT: DD kanaris LOCATION: Monterey

1	Quill Kelvin J	CDR	Commanding Officer
2	Ross Allan J	LCDR	Executive Officer
3	Kontos David K	LT	ASW Officer
4	Kamenos Joe M	LT	Navigation Officer
5	Lemos Tim N	LT	Commun/tion Officer
6	Onasis George E	LTJG	CIC Officer

UNIT: DD Themis LOCATION: Monterey

1	Rokos Nick A	CDR	Commanding Officer
2	Spencer Tim M	LCDR	Executive Officer
3	Pappas Nick C	LTJG	ASW Officer
4	Rigas Ben H	LTJG	Commun/tion Officer

5	Oscar	Tom	K	LTJG	CIC Officer

UNIT: LST Squadron LOCATION: Monterey

1	Potter	Tom	E	CAPT	Commander

UNIT: LST Argo LOCATION: Monterey

1	Roberts	Ben	J	CDR	Commanding Officer
2	Stanley	Cris	K	LCDR	Executive Officer
3	Gorman	Bruce	L	LTJG	Navigation Officer
4	Denton	Mark	S	LTJG	Commun/tion Officer
5	Alkamo	Jim	P	LTJG	CIC Officer

UNIT: LST Rhodes LOCATION: Monterey

1	Rodes	James	K	CDR	Commanding Officer
2	Borrias	Nick	L	LT	Executive Officer
3	Franko	Jess	L	LTJG	Navigation Officer
4	Fox	Don	E	LTJG	Commun/tion Officer
5	Jones	Peter	L	LTJG	CIC Officer

4. Program Report4

This program prints all the officers who have been awarded a degree in the sciences with an emphasis other than military sciences. This list includes name, rank, science, and degree, and is sorted by name.

The structure of the program is as follows:

a. The necessary memory variables are initialized and the heading of the report is printed.

b. The program uses two temporary files, TEMP1 and TEMP2. The MASTER file is used to create a copy into TEMP1, which includes only the necessary fields, serial number, name, rank, education, and degree. Only the officers whose the field 'educat' is not empty are copied. Then the file TEMP1 is sorted on key 'name' into the file TEMP2.

c. Then the program performs a WHILE loop to process each record of the file. More specifically within the WHILE loop it performs the following:

(1) Stores the current field values to the temporary memory variables.

(2) Retrieves from RANKS and SCIENCES files the appropriate rank name and science name respectively.

(3) A CASE statement is executed to store the appropriate degree to the corresponding variable.

(4) The main line of the output is printed.

d. The program proceeds with the next record until the end of file (EOF) is encountered.

e. Before the program returns it deletes the temporary files TEMP1 and TEMP2, and updates MONITOR file.

The listing and output of the programs are shown in the next pages.

```
******** PROGRAM REPORT4 ********

* This program prints all the officers in alphabetical order
* who have been awarded a degree in sciences other than
* military ones

ERASE

* Initialize memory variables

STORE "LIST OF OFFICERS WITH NON-MILITARY STUDIES" TO title
STORE "========================================" TO underline
STORE 1 TO seqnum
STORE 0 TO linecount
STORE "                    " TO pname
STORE "                   " TO mscience
STORE "   " TO meducat
STORE "          " TO pdegree
STORE "   " TO mrank

* Print the list heading

SET PRINT ON
?                "                                                    "
?? "     "+DATE()
?                "                                                    "
?? "     --------"
?
?                "           "+title
?                "           "+underline
?
?                "-------------------------------------------------"
?? "-------------"
?                "¦SEQ#¦      N A M E       ¦ RANK ¦     SCIENCE     "
?? " ¦  DEGREE  ¦"
?                "-------------------------------------------------"
?? "-------------"
STORE linecount+9 TO linecount

* Copy to temporary file TEMP1 those officers whose field
* 'educat'is not empty and only the necessary fields,
* and then sort this file by name into file TEMP2

SELECT PRIMARY
USE master
COPY TO temp1 FIELD serno,name,rank,educat,degree;
              FOR educat <> " "
USE temp1
SORT ON name TO temp2 ASCENDING
USE temp2
DO WHILE .NOT. EOF

   * Store current field values to memory variables
```

```
STORE name TO pname
STORE rank TO mrank
STORE educat TO meducat

* Retrieve the appropriate rank name and science name

SELECT SECONDARY
USE ranks INDEX ranks
FIND &mrank
IF $(p.serno,1,1) = "1"
   STORE armyname TO rankname
ELSE
   STORE navyname TO rankname
ENDIF
USE sciences INDEX sciences
FIND &meducat
STORE name TO mscience

* Store the appropriate degree depending on the value of the
* field "degree" of TEMP2 to the corresponding variable

DO CASE
   CASE p.degree = "B"
      STORE "Bachelor" TO pdegree
   CASE p.degree = "M"
      STORE "Master   " TO pdegree
   CASE p.degree = "P"
      STORE "Ph.D     " TO pdegree
   OTHERWISE
      ERASE
      ? CHR(7)
      @ 3,30 SAY "ILLEGAL DEGREE"
      @ 5,27 SAY p.degree+"    "+p.name
      STORE 0 TO del
      DO WHILE del < 50
         STORE del+1 TO del
      ENDDO WHILE
ENDCASE

* Print the main line of the output

IF linecount > 53
   EJECT
   STORE 0 TO linecount
ENDIF
?             "¦    ¦                    ¦       ¦"
?? "                ¦          ¦"
?             "¦"+STR(seqnum,3)
?? "¦ "+pname
?? " ¦ "+rankname
?? "¦ "+mscience
?? " ¦ "+pdegree+" ¦"
STORE seqnum+1 TO seqnum
STORE linecount+2 TO linecount
```

```
    STORE "                    " TO pname
    STORE "       " TO rankname
    STORE "                    " TO mscience
    STORE "   " TO meducat
    STORE "           " TO pdegree
    STORE "    " TO mrank
    SELECT PRIMARY
    SKIP
ENDDO WHILE .NOT. EOF

DO monitr

* Delete the temporary files TEMP1 and TEMP2

DELETE FILE temp1.DBF
DELETE FILE temp2.DBF
RETURN
```

12/01/85

LIST OF OFFICERS WITH NON-MILITARY STUDIES

SEQ#	N A M E			RANK	SCIENCE	DEGREE
1	Adams	Garry	J	CPT	Oper Research	Bachelor
2	Alkamo	Jim	P	1LT	Meteorology	Bachelor
3	Borrias	Nick	L	CPT	Weapons	Master
4	Bruce	Mark	P	CPT	Oper Research	Master
5	Clark	Tom	L	MAJ	Economics	Master
6	Crosby	Jerry	J	MAJ	Mec Engineering	Bachelor
7	Faser	Gas	L	1LT	Meteorology	Bachelor
8	Frank	Paul	K	1LT	Management	Master
9	Freeman	Jim	E	CPT	Management	Bachelor
10	Ganos	Jim	A	CPT	Weapons	Master
11	Gordon	Jerry	W	CPT	Comp Science	Ph.D
12	Jensen	Ron	P	CPT	Info Systems	Bachelor
13	Kaan	Dave	J	MG	Mathematics	Bachelor
14	Kamenos	Joe	M	CPT	Oceanography	Bachelor
15	King	Lewis	M	LTC	Chemistry	Bachelor
16	Koom	Peter	H	LTC	Physics	Bachelor
17	Lopez	Tom	L	CPT	Comp Science	Bachelor
18	Morton	Brian	C	CPT	Comp Science	Master
19	Newman	Ben	K	1LT	Meteorology	Bachelor
20	Norton	Denis	E	1LT	Medicine	Bachelor
21	Ocasio	Jim	E	BG	Weapons	Master
22	Odello	Bruno	A	COL	Weapons	Master

23	Olsen	Joe	N	LTC	Architecture	Bachelor
24	Onasis	George	E	1LT	Ship Building	Master
25	Palmer	Bob	L	LTC	Aeronautics	Master
26	Pappas	Nick	C	1LT	Ship Building	Master
27	Potter	Tom	E	COL	Weapons	Master
28	Quinn	Peter	C	LTC	Weapons	Master
29	Rigas	Ben	H	1LT	Ship Building	Ph.D
30	Roberts	Ben	J	LTC	Ship Building	Master
31	Ross	Allan	J	MAJ	Weapons	Ph.D
32	Spencer	Tim	M	MAJ	Weapons	Master
33	Turner	Carlos	B	LTC	El Engineering	Bachelor
34	Ulrey	Dan	A	CPT	Comp Systems	Master
35	Watson	Ralph	D	CPT	Comp Systems	Master
36	Zikas	Tom	J	CPT	Weapons	Master

5. Program Report5

This program prints all the officers who speak foreign languages. The list is sorted by language and name and, includes the language spoken as a heading, name, rank, and unit.

The structure of the program is as follows:

a. The necessary memory variables are initialized.

b. A list of all the languages is displayed on the screen and the user is prompted to select a particular, or all the languages to be printed.

c. Depending on the user's choice the program creates a copy fo the MASTER file to temporary file TEMP which includes all the officers who speak any or a particular language. If the user enters a wrong selection, the program goes back to step (b).

d. The main heading of the output is printed.

e. If the user decides to print all the languages, then, the program sorts the created copy on ascending keys 'forlang' and 'name' into file TEMP2, and a WHILE loop is executed to make the process which includes:

(1) The file TEMP2 is read sequentially. The field 'forlag' is compared to the variable 'lang' which contains the value of the field 'forlag' of the previous record. If it is different, this means that we have a new language, whose name is printed as a heading.

(2) The program retrieves the rank and unit names, from the corresponding files, and prints the main line of the output.

(3) The process is continued with the next record.

f. If the user decides to print a particular language, the corresponding language heading is printed, and the previously created copy is sorted by name, into TEMP1.

g. Then a WHILE loop is executed which performs the same things as above, except the comparison, because now we have only one language.

h. At the end all the temporary files are deleted, the monitor file is updated, and the program returns to SUBMENU2.

The listing and output of the program are shown in the next pages.

```
******** PROGRAM REPORT5 ********

* This program prints the officers who speak a particular
* or any foreign language

ERASE
SELECT PRIMARY

* Initialize memory variables

STORE 4 TO vert
STORE "_____" TO line
STORE "============" TO line1
STORE "   " TO lang
STORE "   " TO mlang
STORE "                    " TO pname
STORE "   " TO mrank
STORE "     " TO munit
STORE "     " TO prank
STORE "              " TO punit
STORE 0 TO linecount
STORE 1 TO seqnum

* Display on the screen the list of languages and ask the
* user to make his choice

STORE T TO wrong
DO WHILE wrong
   ERASE
   @ 1,30 SAY "LANGUAGES LIST"
   @ 2,30 SAY "=============="
   @ 3,30 SAY "00_____ALL"
   USE forlangs
   DO WHILE .NOT. EOF
      @ vert,30 SAY code+line+name
      STORE vert+1 TO vert
      SKIP
   ENDDO WHILE
   @ vert+2,27 SAY "Make your selection"
   @ vert+4,27 SAY "Enter two blanks to exit"
   STORE 4 TO vert
   STORE "   " TO choice
   ACCEPT TO choice
   IF choice = "   "
      USE
      RETURN
   ENDIF

   * Create the appropriate copy according to the user's
   * selection

   USE master
   IF choice = "00"
      COPY TO temp FIELD name,rank,unit,forlang;
```

```
                    FOR forlang <> "   "
      STORE F TO wrong
   ELSE
      IF VAL(choice) >= 01 .AND. VAL(choice) <= 16
         COPY TO temp FIELD name,rank,unit FOR forlang = choice
         STORE F TO wrong
      ELSE

         * If the user enters a wrong number, display an error
         * message and give him another chance

         ERASE
         @ 9,18 SAY choice+" is an illegal choice, try again"
         ? CHR(7)
         STORE 1 TO del
         DO WHILE del < 30
            STORE del+1 TO del
         ENDDO WHILE
      ENDIF
   ENDIF
ENDDO WHILE wrong

* Print the main heading

ERASE
SET PRINT ON
?           "                                                    "+DATE()
?           "                                                    --------"
?           "     LIST OF OFFICERS WHO SPEAK FOREIGN LANGUAGES"
?           "     ============================================"
?
?
?           "*********************************************************"
?           "*SEQ#¦      N A M E       ¦  RANK  ¦     U N I T     *"
?           "*********************************************************"
STORE linecount+9 TO linecount

* Do the appropriate process depending on the
* user's request

IF choice = "00"

   * Sort the created copy on ascending keys
   * 'forlang' and 'name'

   USE temp
   SORT ON name TO temp1
   USE temp1
   SORT ON forlang TO temp2
   USE temp2
   DO WHILE .NOT. EOF

      * If a new code for the field 'forlang' is encountered
      * print the new language heading
```

```
IF forlang <> lang
   STORE forlang TO lang
   STORE forlang TO mlang
   STORE 1 TO seqnum
   ?
   ?
   STORE linecount+2 TO linecount
   IF linecount > 53
      EJECT
      STORE 0 TO linecount
   ENDIF
   SELECT SECONDARY
   USE forlangs INDEX forlangs
   FIND &mlang
   ?              "                          "+line1
   ?              "                           "+name
   ?              "                          "+line1
   STORE linecount+3 TO linecount
ENDIF

* Store current field values to memory variables

SELECT PRIMARY
STORE name TO pname
STORE rank TO mrank
STORE unit TO munit

* Retrieve rank and unit names

SELECT SECONDARY
USE ranks INDEX ranks
FIND &mrank
IF $(munit,1,1) = "1"
   STORE armyname TO prank
ELSE
   STORE navyname TO prank
ENDIF
USE units INDEX units
FIND &munit
STORE title TO punit

* Print the main line

IF linecount > 53
   EJECT
   STORE 0 TO linecount
ENDIF
?           "¦    ¦                    ¦        ¦"
?? "                ¦"
?           "¦"+STR(seqnum,3)
?? "¦  "+pname
?? " ¦  "+prank
?? " ¦  "+punit+"  ¦"
```

```
      STORE seqnum+1 TO seqnum
      STORE linecount+2 TO linecount

      * Continue with the next record

      SELECT PRIMARY
      SKIP
   ENDDO WHILE
ELSE

   * Handle the case where the user has requested
   * a list of those officers who speak a
   * particular language

   SELECT SECONDARY
   USE forlangs INDEX forlangs
   FIND &choice
   ?
   ?
   ?           "                    "+line1
   ?           "                     "+name
   ?           "                    "+line1
   STORE linecount+5 TO linecount

   * Sort the created copy in ascending order
   * with key 'name'

   SELECT PRIMARY
   USE temp
   SORT ON name TO temp1
   USE temp1
   DO WHILE .NOT. EOF

      * Store current field values to memory variables

      STORE name TO pname
      STORE rank TO mrank
      STORE unit TO munit

      * Get the rank and unit names

      SELECT SECONDARY
      USE ranks INDEX ranks
      FIND &mrank
      IF $(p.unit,1,1) = "1"
         STORE armyname TO prank
      ELSE
         STORE navyname TO prank
      ENDIF
      USE units INDEX units
      FIND &munit
      STORE title TO punit

      * Print the main line
```

```
      IF linecount > 53
         EJECT
         STORE 0 TO linecount
      ENDIF
      ?           "¦     ¦                          ¦          ¦"
      ?? "                  ¦"
      ?           "¦"+STR(seqnum,3)
      ?? "¦   "+name
      ?? " ¦   "+prank
      ?? " ¦   "+punit+"   ¦"
      STORE seqnum+1 TO seqnum
      STORE linecount+2 TO linecount

      * Proceed with the next record

      SELECT PRIMARY
      SKIP
   ENDDO WHILE
ENDIF

* Delete the temporary files

IF choice = "00"
   DELETE FILE temp.DBF
   DELETE FILE temp1.DBF
   DELETE FILE temp2.DBF
ELSE
   DELETE FILE temp.DBF
   DELETE FILE temp1.DBF
ENDIF

* Update MONITOR file and return

DO monitr
RETURN
```

12/01/85

LIST OF OFFICERS WHO SPEAK FOREIGN LANGUAGES

English

*SEQ#	N A M E			RANK	U N I T *
1	Allen	David	E	CPT	2nd Inf Bn
2	Borrias	Nick	L	LT	LST Rhodes
3	Cook	Ryan	K	MAJ	Armour Bn
4	Dalton	John	E	MAJ	4th Inf Bn
5	Farmer	Peter	J	1LT	5th Inf Bn
6	Ford	Roger	A	CPT	Sup/Trans Bn
7	Fox	Don	E	LTJG	LST Rhodes
8	Ganos	Jim	A	LT	DD Miaoulis
9	Garret	Rex	D	CPT	Signals Bn
10	Gorby	Glen	G	CPT	Engineers Bn
11	Gorman	Bruce	L	LTJG	LST Argo
12	Jackson	Peter	M	CPT	3rd Inf Bn
13	Kaan	Dave	J	MG	1st Inf Div
14	Karras	Mike	L	LTC	1st Inf Div
15	Keen	Robt	N	COL	1st Inf Div
16	Larson	Roger	K	CPT	Armour Bn
17	Manos	John	K	CPT	1st Arty Bn
18	Newton	John	J	1LT	Engineers Bn
19	Onasis	George	E	LTJG	DD kanaris
20	Owens	Bill	L	1LT	Sup/Trans Bn

21	Palmer	Bob	L	LTC	5th Inf Bn
22	Perry	Bill	H	LTC	2nd Arty Bn
23	Peters	Mark	J	LTC	3rd Inf Bn
24	Quill	Kelvin	J	CDR	DD kanaris
25	Scott	Paul	V	LTC	Engineers Bn
26	Victor	David	R	CPT	5th Inf Bn

==========
German
==========

1	King	Lewis	M	LTC	1st Inf Bn

==========
Italian
==========

1	Dorey	Jim	H	MAJ	2nd Inf Bn

==========
French
==========

1	Bruce	Mark	P	CPT	1st Inf Bn
2	Torres	Alex	A	LTC	Sup/Trans Bn

==========
Spanish
==========

1	Evans	Isaac	M	1LT	2nd Arty Bn
2	Ingals	Tomas	R	CPT	4th Inf Bn

==========
Turkish
==========

1	Potter	Tom	E	CAPT	LST Squadron
2	Roberts	Ben	J	CDR	LST Argo

Arabian

1	Kamenos	Joe	M	LT	
2	Rivera	Mario	L	LCDR	

Japanese

1	Alkamo	Jim	P	LTJG	
2	Clark	Tom	L	MAJ	Signals Bn

Chinese

1	Horan	Kevin	L	CPT	2nd Inf Bn
2	Morton	Brian	C	CPT	2nd Arty Bn

Portuguese

1	Frank	Paul	K	1LT	1st Inf Bn
2	Watson	Ralph	D	CPT	4th Inf Bn

Russian

1	Ocasio	Jim	E	COMD	Navy Command
2	Spencer	Tim	M	LCDR	DD Themis

Bulgarian

1	Zikas	Tom	J	LT	DD Miaoulis

Korean

1	Rigas	Ben	H	LTJG	DD Themis

Danish

1	Ullman	Rolf	G	CPT	1st Arty Bn

Swedish

1	Felton	John	K	1LT	3rd Inf Bn
2	Lopez	Tom	L	CPT	Armour Bn

Dutch

1	Good	John	L	CPT	Signals Bn

6. Program Report6

This program lists all the officers in alphabetical order, including name, rank, marital status, and number of children.

The program has the following structure:

a. The appropriate memory variables are initialized and the main titles are printed.

b. We use two temporary files, TEMP1 and TEMP2. The first one is used for creating a copy from MASTER file, and which includes, serial number, name, rank, marital status, and number of children, i.e. only the necessary information for creating the report. Then this file is sorted by name into the file TEMP2.

c. Then the main process of the program is executed within a WHILE loop with the following functions:

(1) The current field values are stored to the memory variables.

(2) The appropriate rank name is retrieved from RANKS file and marital status is stored to the corresponding variable, depending on the value of the field 'marstat', of the file TEMP2.

(3) Then the program prints the main line and proceeds with the next record.

d. At the end of the WHILE loop the program deletes the files TEMP1 and TEMP2, and then returns to the calling program.

The listing and output of the program are shown in the next pages.

```
******** PROGRAM REPORT6 ********

* This program lists all the officers in alphabetical order
* including name, rank, marital status, and number of children

ERASE

* Initialize memory variables

STORE "LIST OF OFFICERS INCLUDING MARITAL STATUS" TO title
STORE "========================================" TO underline
STORE 1 TO seqnum
STORE 0 TO linecount
STORE "                    " TO pname
STORE "          " TO pmarstat
STORE " " TO pchildren
STORE "     " TO prank
STORE "   " TO mrank

* Print the list heading

SET PRINT ON
?              "                                                "
?? "             "+DATE()
?              "                                                "
?? "             --------"
?
?              "             "+title
?              "             "+underline
?
?              "----------------------------------------"
?? "--------------------"
?              "|SEQ#|      N A M E        | RANK | MARITAL"
?? "STATUS | CHILDREN |"
?              "----------------------------------------"
?? "--------------------"
STORE linecount+9 TO linecount

* Copy MASTER file to temporary file TEMP1 including only the
* necessary fields, and sort this file by name into TEMP2

SELECT PRIMARY
USE master
COPY TO temp1 FIELD serno,name,rank,marstat,children
USE temp1
SORT ON name TO temp2 ASCENDING
USE temp2
DO WHILE .NOT. EOF

*  Store current field values to memory variables

   STORE name TO pname
   STORE rank TO mrank
   STORE marstat TO pmarstat
```

```
   STORE children TO pchildren

*  Get the appropriate rank name

   SELECT SECONDARY
   USE ranks INDEX ranks
   FIND &mrank
   IF $(p.serno,1,1) = "1"
      STORE armyname TO prank
   ELSE
      STORE navyname TO prank
   ENDIF

*  Store the appropriate marital status name

   DO CASE
      CASE p.marstat = "M"
         STORE "Married   " TO pmarstat
      CASE p.marstat = "U"
         STORE "Unmarried" TO pmarstat
      CASE p.marstat = "D"
         STORE "Divorced " TO pmarstat
      OTHERWISE
         ERASE
         ? CHR(7)
         @ 10,30 SAY "ILLEGAL MARITAL STATUS"
         @ 12,30 SAY p.marstat+"***"+p.name
         STORE 1 TO del
         DO WHILE del < 50
            STORE del+1 TO del
         ENDDO
   ENDCASE

*  Print the main line of the list

   IF linecount > 53
      EJECT
      STORE 0 TO linecount
   ENDIF
   ?            "¦     ¦                          ¦       ¦"
   ?? "                 ¦           ¦"
   ?            "¦"+STR(seqnum,3)
   ?? "¦   "+pname
   ?? " ¦ "+prank
   ?? "¦    "+pmarstat
   ?? "    ¦     "+pchildren+"      ¦"
   STORE seqnum+1 TO seqnum
   STORE linecount+2 TO linecount
   STORE "                    " TO pname
   STORE "          " TO pmarstat
   STORE "     " TO prank
   STORE "  " TO pchildren
   STORE "   " TO mrank
   SELECT PRIMARY
```

```
   SKIP
ENDDO WHILE .NOT. EOF
DO monitr

* Delete the temporary files TEMP1 and TEMP2

DELETE FILE temp1.DBF
DELETE FILE temp2.DBF
RETURN
```

12/01/85

LIST OF OFFICERS INCLUDING MARITAL STATUS

SEQ#	N A M E			RANK	MARITAL STATUS	CHILDREN
1	Adams	Garry	J	CPT	Married	
2	Alkamo	Jim	P	1LT	Married	3
3	Allen	David	E	CPT	Divorced	1
4	Boris	Peter	H	CPT	Married	1
5	Borrias	Nick	L	CPT	Unmarried	
6	Brown	Peter	R	CPT	Married	1
7	Bruce	Mark	P	CPT	Divorced	
8	Byron	Larry	M	CPT	Unmarried	
9	Clark	Tom	L	MAJ	Unmarried	
10	Cook	Ryan	K	MAJ	Married	2
11	Crosby	Jerry	J	MAJ	Married	2
12	Dalton	John	E	MAJ	Married	
13	Denton	Mark	S	1LT	Married	2
14	Dorey	Jim	H	MAJ	Divorced	2
15	Durran	Frank	M	MAJ	Married	
16	Emery	Roy	J	MAJ	Divorced	
17	Evans	Isaac	M	1LT	Married	1
18	Evans	Tom	M	1LT	Unmarried	
19	Farmer	Peter	J	1LT	Unmarried	
20	Faser	Gas	L	1LT	Unmarried	
21	Felton	John	K	1LT	Unmarried	
22	Foley	Mark	L	1LT	Unmarried	

23	Ford	Roger	A	CPT	Married	2
24	Foster	Kliff	J	CPT	Married	3
25	Fox	Don	E	1LT	Unmarried	
26	Frank	Paul	K	1LT	Unmarried	
27	Franko	Jess	L	1LT	Divorced	1
28	Freeman	Jim	E	CPT	Married	1
29	Ganos	Jim	A	CPT	Married	2
30	Garret	Rex	D	CPT	Married	
31	Gilman	Perry	M	CPT	Divorced	
32	Good	John	L	CPT	Unmarried	
33	Gorby	Glen	G	CPT	Unmarried	
34	Gordon	Jerry	W	CPT	Married	2
35	Gorman	Bruce	L	1LT	Married	1
36	Hogan	Joe	K	CPT	Unmarried	
37	Horan	Kevin	L	CPT	Unmarried	
38	Ingals	Tomas	R	CPT	Divorced	
39	Jackson	Peter	M	CPT	Unmarried	
40	Jensen	Ron	P	CPT	Married	2
41	Jones	Peter	L	1LT	Married	2
42	Kaan	Dave	J	MG	Married	1
43	Kamenos	Joe	M	CPT	Divorced	2
44	Kane	Bob	R	BG	Married	2
45	Karras	Mike	L	LTC	Married	2
46	Keen	Robt	N	COL	Married	1
47	King	Lewis	M	LTC	Unmarried	
48	Kirk	Burt	N	LTC	Unmarried	
49	Kliff	Frank	E	MAJ	Unmarried	

50	Kontos	David	K	CPT	Married	3
51	Koom	Peter	H	LTC	Married	3
52	Kueny	John	S	LTC	Unmarried	
53	Larsen	Allen	A	CPT	Married	
54	Larson	Roger	K	CPT	Unmarried	
55	Layton	John	E	CPT	Divorced	1
56	Lemos	Tim	N	CPT	Unmarried	
57	Lopez	Tom	L	CPT	Married	1
58	Manos	John	K	CPT	Married	
59	Moore	Roger	L	CPT	Married	2
60	Morris	Roy	K	CPT	Married	1
61	Morton	Brian	C	CPT	Married	
62	Newman	Ben	K	1LT	Married	2
63	Newton	John	J	1LT	Married	3
64	Norton	Denis	E	1LT	Married	2
65	Ocasio	Jim	E	BG	Married	2
66	Odello	Bruno	A	COL	Divorced	1
67	Olsen	Joe	N	LTC	Married	3
68	Onasis	George	E	1LT	Unmarried	
69	Oscar	Tom	K	1LT	Married	2
70	Owens	Bill	L	1LT	Married	1
71	Palmer	Bob	L	LTC	Married	1
72	Pappas	Nick	C	1LT	Unmarried	
73	Patton	Mike	H	LTC	Married	3
74	Perry	Bill	H	LTC	Married	1
75	Peters	Mark	J	LTC	Divorced	1
76	Peters	Nick	K	LTC	Married	2

77	Potter	Tom	E	COL	Unmarried	
78	Quill	Kelvin	J	LTC	Unmarried	
79	Quinn	Peter	C	LTC	Married	1
80	Rigas	Ben	H	1LT	Married	1
81	Rivera	Mario	L	MAJ	Married	2
82	Roberts	Ben	J	LTC	Divorced	
83	Rodes	James	K	LTC	Married	2
84	Rokos	Nick	A	LTC	Married	2
85	Ross	Allan	J	MAJ	Married	2
86	Sanders	James	F	LTC	Unmarried	
87	Scott	Paul	V	LTC	Unmarried	
88	Spencer	Tim	M	MAJ	Unmarried	
89	Stanley	Cris	K	MAJ	Unmarried	
90	Takas	Costas	L	CPT	Married	2
91	Torres	Alex	A	LTC	Unmarried	
92	Turner	Carlos	B	LTC	Unmarried	
93	Ullman	Rolf	G	CPT	Married	1
94	Ulrey	Dan	A	CPT	Married	
95	Victor	David	R	CPT	Unmarried	
96	Vongel	Mark	E	CPT	Unmarried	
97	Warren	Geo	M	CPT	Married	
98	Waters	Gary	L	CPT	Married	3
99	Watson	Ralph	D	CPT	Divorced	2
100	Zikas	Tom	J	CPT	Married	2

7. Program Report7

This program calculates the service time in the current unit, as well as the total service time for each officer. The output of the program is directed to the printer only, while appropriate messages are displayed on the screen to inform the user of what is going on at the various steps of the program.

The structure of the program is as follows:

a. The necessary memory variables are initialized and the headlines are printed.

b. The MASTER file is copied to the file TEMP including only the fields 'name', 'serentry', and 'reptdate'. Then this file is sorted by name to the file TEMP1, which is used for the main process.

c. The program performs a WHILE loop to do the following:

(1) In order to do the calculations the proper way, we have to transform the format of the dates from MM/DD/YY to YY/MM/DD.

(2) We calculate the service time in current unit as follows: The day of the report date, is compared to the day of the day of the current date. If it is greater, we add 30 days to the day of the current date, and subtract 1 from the month of the current date, otherwise they remain unchanged. Then the month of report is compared to the month of the current date, either unchanged or reduced by 1.

as mentioned before. If it is greater we add 12 to the month of the current date and subtract 1 from the year of the current date, otherwise remain unchanged. This way we make sure that the subtraction is possible. We do the operation storing the number of years, months, and days to memory variables.

(3) With the same logic we calculate the total service time.

(4) The program prints the main line, reinitializes the proper memory variables to accept the new values, and continues with the next record.

d. At the end the MONITOR file is updated, the temporary files are deleted, and the program returns.

The listing and output of the program are shown in the next pages.

```
******** PROGRAM REPORT7 ********

* This program prints all the officers with their service
* time in the current unit and the total service time

ERASE
SET CONSOLE OFF
SET PRINT ON

* Initialize memory variables

STORE "                    " TO pname
STORE "        " TO mdate
STORE "        " TO mserentry
STORE "        " TO mreptdate
STORE "LIST OF OFFICERS INCLUDING SERVICE TIME" TO title1
STORE "IN CURRENT UNIT AND TOTAL SERVICE TIME" TO title2
STORE "========================================" TO line3
STORE "*    ¦                    ¦SERVICE TIME IN UNIT" TO title3
STORE "¦TOTAL SERVICE TIME *" TO title4
STORE "*SEQ#¦     N A M E      ¦-------------" TO midline1
STORE "-------¦------------------*" TO midline2
STORE "*    ¦                    ¦YEARS  MONTHS  DAYS" TO title5
STORE " ¦YEARS  MONTHS  DAYS*" TO title6
STORE "**************************************" TO line1
STORE "****************************" TO line2
STORE 0 TO linecount
STORE 1 TO seqnum
STORE 0 TO day
STORE 0 TO days
STORE 0 TO month
STORE 0 TO months
STORE 0 TO year
STORE 0 TO years
STORE 0 TO day1
STORE 0 TO month1
STORE 0 TO month2
STORE 0 TO year1

* Print main headings

@ 9,19 SAY "Now the headlines are being printed"
?              "                                  "
?? "                                "+DATE()
?              "                                  "
?? "                                --------"
?              "              "+title1
?              "              "+line3
?              "              "+title2
?              "              "+line3
?
?
? line1+line2
? title3+title4
```

```
? midline1+midline2
? title5+title6
? line1+line2
STORE linecount+13 TO linecount
ERASE

* Create a copy of the MASTER file and sort it by name

@ 9,17 SAY "Now wait a little, the MASTER file is being sorted"
SELECT PRIMARY
USE master
COPY TO temp FIELD name,serentry,reptdate
USE temp
SORT ON name TO temp1
USE temp1
ERASE
@ 9,21 SAY "Report7 is currently being printed"
DO WHILE .NOT. EOF

   * Store current field values in temporary variables
   * so that the report date and service entry are
   * transformed from the form MM/DD/YY to the form
   * YY/MM/DD

   STORE name TO pname
   STORE $(reptdate,5,2)+$(reptdate,1,2)+$(reptdate,3,2);
         TO mreptdate
   STORE $(serentry,5,2)+$(serentry,1,2)+$(serentry,3,2);
         TO mserentry
   STORE $(DATE(),7,2)+$(DATE(),1,2)+$(DATE(),4,2) TO mdate

   * Calculate the service time in current unit

   IF VAL($(mreptdate,5,2)) > VAL($(mdate,5,2))
      STORE VAL($(mdate,5,2)) + 30 TO day1
      STORE VAL($(mdate,3,2)) - 1 TO month1
   ELSE
      STORE VAL($(mdate,5,2)) TO day1
      STORE VAL($(mdate,3,2)) TO month1
   ENDIF
   IF VAL($(mreptdate,3,2)) > month1
      STORE month1 + 12 TO month2
      STORE VAL($(mdate,1,2)) - 1 TO year1
   ELSE
      STORE month1 TO month2
      STORE VAL($(mdate,1,2)) TO year1
   ENDIF
   STORE year1 - VAL($(mreptdate,1,2)) TO years
   STORE month2 - VAL($(mreptdate,3,2)) TO months
   STORE day1 - VAL($(mreptdate,5,2)) TO days

   * Calculate the total service time

   IF VAL($(mserentry,5,2)) > VAL($(mdate,5,2))
```

```
      STORE VAL($(mdate,5,2)) + 30 TO day1
      STORE VAL($(mdate,3,2)) - 1 TO month1
   ELSE
      STORE VAL($(mdate,5,2)) TO day1
      STORE VAL($(mdate,3,2)) TO month1
   ENDIF
   IF VAL($(mserentry,3,2)) > month1
      STORE month1 + 12 TO month2
      STORE VAL($(mdate,1,2)) - 1 TO year1
   ELSE
      STORE month1 TO month2
      STORE VAL($(mdate,1,2)) TO year1
   ENDIF
   STORE year1 - VAL($(mserentry,1,2)) TO year
   STORE month2 - VAL($(mserentry,3,2)) TO month
   STORE day1 - VAL($(mserentry,5,2)) TO day

   * Print the main line

   IF linecount > 53
      EJECT
      STORE 50 TO linecount
   ENDIF
   ?           "¦    ¦                    ¦          "
   ?? "           ¦                     ¦"
   ?           "¦"+STR(seqnum,3)
   ?? "¦  "+name
   ?? "¦   "+STR(years,2)+"      "+STR(months,2)+"     "
   ?? STR(days,2)+"    ¦ "
   ?? STR(year,2)+"      "+STR(month,2)+"     "
   ?? STR(day,2)+" ¦"
   STORE seqnum+1 TO seqnum
   STORE linecount+2 TO linecount

   * Reinitialize to 0 the memory variables that contain
   * the service times

   STORE 0 TO year
   STORE 0 TO month
   STORE 0 TO day
   STORE 0 TO years
   STORE 0 TO months
   STORE 0 TO days
   SKIP
ENDDO WHILE .NOT. EOF

* Delete the temporary files and update the MONITOR file

ERASE
@ 9,20 SAY "The temporary auxiliary files are being deleted"
DELETE FILE temp
DELETE FILE temp1
DO monitr
ERASE
```

```
SET CONSOLE ON
RETURN
```

12/01/85

LIST OF OFFICERS INCLUDING SERVICE TIME IN CURRENT UNIT AND TOTAL SERVICE TIME

SEQ#	NAME	SERV TIME IN UNIT			TOTAL SERVICE TIME		
		YY	MM	DD	YY	MM	DD
1	Adams Garry J	1	2	11	13	2	10
2	Alkamo Jim P	2	4	11	9	2	9
3	Allen David E	2	2	15	13	2	10
4	Boris Peter H	1	4	15	13	2	9
5	Borrias Nick L	1	4	25	16	2	9
6	Brown Peter R	2	2	15	13	2	10
7	Bruce Mark P	1	3	16	14	2	11
8	Byron Larry M	1	3	10	14	2	11
9	Clark Tom L	2	4	9	16	2	10
10	Cook Ryan K	1	4	10	15	2	9
11	Crosby Jerry J	2	4	11	16	2	10
12	Dalton John E	2	4	11	17	2	10
13	Denton Mark S	1	4	12	9	2	9
14	Dorey Jim H	1	4	11	18	2	11
15	Durran Frank M	1	4	11	18	2	11
16	Emery Roy J	2	4	13	18	2	11
17	Evans Isaac M	2	4	15	8	2	10
18	Evans Tom M	2	4	16	8	2	10
19	Farmer Peter J	1	4	18	8	2	10
20	Faser Gas L	2	4	19	9	2	11

21	Felton	John	K	1	4	21	9	2	11
22	Foley	Mark	L	2	4	18	10	2	12
23	Ford	Roger	A	1	4	3	11	2	12
24	Foster	Kliff	J	2	4	2	12	2	9
25	Fox	Don	E	1	4	9	9	2	9
26	Frank	Paul	K	1	4	1	10	2	12
27	Franko	Jess	L	2	4	10	9	2	9
28	Freeman	Jim	E	2	4	4	14	2	11
29	Ganos	Jim	A	1	4	21	13	2	9
30	Garret	Rex	D	2	4	5	12	2	9
31	Gilman	Perry	M	1	4	6	12	2	9
32	Good	John	L	1	4	7	14	2	11
33	Gorby	Glen	G	1	4	9	11	2	12
34	Gordon	Jerry	W	2	4	8	13	2	10
35	Gorman	Bruce	L	2	4	12	9	2	9
36	Hogan	Joe	K	2	4	13	12	2	9
37	Horan	Kevin	L	1	2	11	12	2	9
38	Ingals	Tomas	R	1	3	16	14	2	11
39	Jackson	Peter	M	1	4	11	12	2	9
40	Jensen	Ron	P	1	4	11	14	2	11
41	Jones	Peter	L	2	4	8	9	2	9
42	Kaan	Dave	J	1	4	11	30	2	10
43	Kamenos	Joe	M	2	4	17	13	2	9
44	Kane	Bob	R	1	4	6	28	2	12
45	Karras	Mike	L	1	4	6	25	2	11
46	Keen	Robt	N	2	3	5	26	2	9
47	King	Lewis	M	1	4	11	22	2	16

48	Kirk	Burt	N	2	3	9	26	2	9
49	Kliff	Frank	E	1	4	10	15	2	8
50	Kontos	David	K	2	4	18	13	2	9
51	Koom	Peter	H	1	3	7	26	2	9
52	Kueny	John	S	1	4	11	21	2	13
53	Larsen	Allen	A	2	4	10	11	2	12
54	Larson	Roger	K	1	4	11	14	2	11
55	Layton	John	E	2	4	12	12	2	9
56	Lemos	Tim	N	3	4	17	13	2	9
57	Lopez	Tom	L	1	4	18	11	2	12
58	Manos	John	K	1	4	24	12	2	9
59	Moore	Roger	L	2	4	25	11	2	12
60	Morris	Roy	K	2	4	18	11	2	12
61	Morton	Brian	C	1	4	20	11	2	12
62	Newman	Ben	K	1	4	14	7	2	9
63	Newton	John	J	2	4	13	7	2	9
64	Norton	Denis	E	1	4	21	7	2	9
65	Ocasio	Jim	E	1	4	21	28	2	9
66	Odello	Bruno	A	1	4	11	25	2	9
67	Olsen	Joe	N	3	4	19	9	2	11
68	Onasis	George	E	3	4	16	9	2	9
69	Oscar	Tom	K	2	4	13	9	2	9
70	Owens	Bill	L	2	4	20	8	2	10
71	Palmer	Bob	L	2	4	10	23	2	12
72	Pappas	Nick	C	3	4	15	9	2	9
73	Patton	Mike	H	2	4	9	23	2	12
74	Perry	Bill	H	1	4	11	23	2	12

75	Peters	Mark	J	2	4	13	21	2	13
76	Peters	Nick	K	1	4	12	22	2	16
77	Potter	Tom	E	1	4	12	25	2	9
78	Quill	Kelvin	J	2	4	12	20	2	9
79	Quinn	Peter	C	2	4	19	20	2	9
80	Rigas	Ben	H	2	4	14	9	2	9
81	Rivera	Mario	L	1	4	14	16	2	9
82	Roberts	Ben	J	2	4	12	20	2	9
83	Rodes	James	K	1	4	9	20	2	9
84	Rokos	Nick	A	1	4	13	20	2	9
85	Ross	Allan	J	1	4	6	16	2	9
86	Sanders	James	F	1	4	11	20	2	11
87	Scott	Paul	V	2	3	16	20	2	11
88	Spencer	Tim	M	1	4	23	16	2	9
89	Stanley	Cris	K	2	4	22	16	2	9
90	Takas	Costas	L	1	4	20	13	2	9
91	Torres	Alex	A	1	4	21	19	2	13
92	Turner	Carlos	B	2	3	16	19	2	13
93	Ullman	Rolf	G	2	4	28	13	2	10
94	Ulrey	Dan	A	1	4	25	14	2	11
95	Victor	David	R	2	4	11	12	2	9
96	Vongel	Mark	E	2	4	16	14	2	11
97	Warren	Geo	M	1	3	16	13	2	10
98	Waters	Gary	L	1	4	16	14	2	11
99	Watson	Ralph	D	2	4	11	13	2	10
100	Zikas	Tom	J	1	4	19	13	2	9

8. Program Report8

This program prints all the officers in alphabetical order including name, rank, address, and phone #.

The structure of the program is as follows:

a. The necessary memory variables are initialized and the main heading is printed.

b. The MASTER file is sorted into the temporary file TEMP in alphabetical order, with key the officer's name.

c. The program performs the main process within a WHILE loop which includes:

(1) Storing the appropriate field values to the corresponding memory variables.

(2) Retrieving the appropriate rank name from RANKS file.

(3) Printing the main line of the output, and proceeding with the next record.

d. The temporary file TEMP is deleted, the MONITOR file is updated, and then the program returns.

The listing and output of the program are shown in the next pages.

```
******** PROGRAM REPORT8 ********

* This program prints all the officers in alphabetical
* order including name, rank, address, and phone#

ERASE

* Initialize memory variables

STORE "LIST OFF OFFICERS WITH THEIR ADDRESS AND PHONE#" TO title
STORE "================================================" TO line
STORE 1 TO seqnum
STORE 0 TO linecount
STORE "                    " TO pname
STORE "    " TO rankname
STORE "   " TO mrank
STORE "                         " TO paddress
STORE "         " TO pphone
SET PRINT ON

* Print the heading

? "                                                  "
?? "              "           +DATE()
? "                                                  "
?? "              "        +"--------"
?
?            "          "+title
?            "          "+line
?
?            "--------------------------------"
?? "--------------------------------"
?            "¦SEQ#¦     N A M E      ¦ RANK ¦"
?? "    A D D R E S S     ¦ PHONE# ¦"
?            "--------------------------------"
?? "--------------------------------"
STORE linecount + 9 TO linecount
SELECT PRIMARY
USE master

* Sort MASTER file in alphabetical order of the officer's
* names into the temporary file TEMP, and use this file for
* the database processing

SORT ON name TO temp ASCENDING
USE temp
DO WHILE .NOT. EOF
   STORE name TO pname
   STORE rank TO mrank
   STORE address TO paddress
   STORE $(phone,1,3)+"-"+$(phone,4,4) TO pphone

   * Search RANKS file with key the field 'rank' of the
   * officer's record and get the appropriate rank name
```

```
   SELECT SECONDARY
   USE ranks INDEX ranks
   FIND &mrank
   IF $(p.unit,1,1) = "1"
      STORE armyname TO rankname
   ELSE
      STORE navyname TO rankname
   ENDIF

   * If the line counter exceeds 53, continue in the next page

   IF linecount > 53
      EJECT
      STORE 0 TO linecount
   ENDIF

   * Print the data concerning the officer

   ?            "¦    ¦                    ¦       ¦"
   ?? "                        ¦        ¦"
   ?            "¦"+STR(seqnum,3)
   ?? "¦ "+pname
   ?? "¦ "+rankname
   ?? "¦ "+paddress
   ?? " ¦"+pphone+"¦"
   STORE linecount+2 TO linecount
   STORE seqnum+1 TO seqnum
   STORE "                    " TO pname
   STORE "     " TO rankname
   STORE "   " TO mrank
   STORE "                        " TO paddress
   STORE "         " TO pphone

   * Continue with the next record

   SELECT PRIMARY
   SKIP
ENDDO WHILE .NOT. EOF

* Delete file TEMP, update MONITOR file and return

DELETE FILE temp.DBF
DO monitr
RETURN
```

```
******** PROGRAM PRINTMON ********

* This program prints statistical information
* of who did what and when

ERASE
DO monitr

* Initialize memory variables

STORE "MONITORING THE USERS' ACCESS TO THE SYSTEM" TO title1
STORE "=========================================" TO line1
STORE "  DATE          N A M E          JOB DONE" TO title2
STORE "     FILE/REP    # OF REC" TO title3
STORE "-----------------------------------------" TO line2
STORE "-------------------------" TO line3

* Print the headline

SET PRINT ON
? "                                                    "
?? "             "+DATE()
? "                                                    "
?? "             --------"
? "                   "+title1
? "                   "+line1
?
? "          "+title2
??  title3
? "          "+line2
?? line3
SELECT PRIMARY
USE monitor
DO WHILE .NOT. EOF

   * Print the main line

   ? "          "+date
   ?? "   "+name
   ?? "   "+job
   ?? "   "+fle:rept
   ?? "   "+STR(numrec,3)
   SKIP
ENDDO WHILE
RETURN
```

12/01/85

LIST OF OFFICERS WITH THEIR ADDRESS AND PHONE#

SEQ#	N A M E	RANK	A D D R E S S	PHONE#
1	Adams Garry J	CPT	49 Scott, San Jose	375-5720
2	Alkamo Jim P	LTJG	64 3rd, Otay	623-3460
3	Allen David E	CPT	36 Laine, Ford Ord	375-5619
4	Boris Peter H	LT	6 1st, Monterey	623-3133
5	Borrias Nick L	LT	10 Pine, Monterey	623-3232
6	Brown Peter R	CPT	17 River, Salinas	373-5518
7	Bruce Mark P	CPT	86 Monroe, Marina	375-5417
8	Byron Larry M	CPT	24 Pine, Monterey	373-5316
9	Clark Tom L	MAJ	66 Hills, Marina	373-5215
10	Cook Ryan K	MAJ	44 Carlos, Monterey	373-4913
11	Crosby Jerry J	MAJ	36 Forest, St Clara	373-4912
12	Dalton John E	MAJ	22 Mount, Carmel	373-4911
13	Denton Mark S	LTJG	63 2nd, Otay	623-3450
14	Dorey Jim H	MAJ	81 Lake, Salinas	373-4915
15	Durran Frank M	MAJ	77 Hills, San Jose	374-5816
16	Emery Roy J	MAJ	79 Forest, San Jose	373-6819
17	Evans Isaac M	1LT	18 David, Marina	373-6774
18	Evans Tom M	1LT	17 Della, Marina	373-6773
19	Farmer Peter J	1LT	16 Denise, Carmel	373-6772
20	Faser Gas L	1LT	15 Diane, Bonita	373-6771
21	Felton John K	1LT	14 Fox, Nestor	373-6770
22	Foley Mark L	1LT	59 Hofman, Fresno	568-6229

23	Ford	Roger	A	CPT	52 Morgan, Otay	565-6226
24	Foster	Kliff	J	CPT	55 Lilly, Carmel	565-6227
25	Fox	Don	E	LTJG	66 Belden, Carmel	623-3480
26	Frank	Paul	K	1LT	57 Pine, Otay	565-6228
27	Franko	Jess	L	LTJG	65 Desty, Moreno	623-3470
28	Freeman	Jim	E	CPT	51 Lase, Carmel	565-6225
29	Ganos	Jim	A	LT	10 Passo, San Diego	623-3134
30	Garret	Rex	D	CPT	49 Kity, Monterey	565-6224
31	Gilman	Perry	M	CPT	47 Laco, Carmel	565-6223
32	Good	John	L	CPT	45 David, Otay	565-6222
33	Gorby	Glen	G	CPT	41 9th, Fresno	565-6220
34	Gordon	Jerry	W	CPT	43 Pine, Otay	565-6221
35	Gorman	Bruce	L	LTJG	62 David, Otay	623-3440
36	Hogan	Joe	K	CPT	98 Camino, Marina	373-5922
37	Horan	Kevin	L	CPT	74 Ramona, Carmel	375-5821
38	Ingals	Tomas	R	CPT	11 Forest, St Clara	373-6225
39	Jackson	Peter	M	CPT	32 Grove, Monterey	373-6023
40	Jensen	Ron	P	CPT	55 Hilby, Salinas	373-6124
41	Jones	Peter	L	LTJG	67 Passo, Salinas	623-3481
42	Kaan	Dave	J	MG	71 Buna, Monterey	372-6402
43	Kamenos	Joe	M	LT	60 Desty, Monterey	623-3138
44	Kane	Bob	R	BG	53 Vista, Monterey	272-6403
45	Karras	Mike	L	LTC	19 Franklin, Marina	372-6406
46	Keen	Robt	N	COL	48 Cannery, Carmel	372-6404
47	King	Lewis	M	LTC	31 Trinity, Salinas	372-6408
48	Kirk	Burt	N	LTC	25 Ocean, Monterey	372-6407
49	Kliff	Frank	E	MAJ	52 Castro, Fresno	375-5114

50	Kontos	David	K	LT	10 Pine, Marina	623-3137
51	Koom	Peter	H	LTC	97 Nadina, Carmel	372-6405
52	Kueny	John	S	LTC	28 Mission, Monterey	372-6409
53	Larsen	Allen	A	CPT	39 6th, Fresno	565-6219
54	Larson	Roger	K	CPT	37 Marcy, Fresno	565-6218
55	Layton	John	E	CPT	35 Spencer, Carmel	565-6217
56	Lemos	Tim	N	LT	13 Dian, Otay	623-3139
57	Lopez	Tom	L	CPT	33 Pine, Bonita	565-6216
58	Manos	John	K	CPT	25 3rd, Carmel	565-6212
59	Moore	Roger	L	CPT	27 Market, Marina	565-6213
60	Morris	Roy	K	CPT	31 David, Otay	565-6215
61	Morton	Brian	C	CPT	27 Morena, Carmel	565-6214
62	Newman	Ben	K	1LT	19 Pine, Marina	373-6775
63	Newton	John	J	1LT	20 View, Monterey	373-6776
64	Norton	Denis	E	1LT	21 Side, Carmel	373-6777
65	Ocasio	Jim	E	COMD	25 Dexter, Carmel	373-6789
66	Odello	Bruno	A	CAPT	26 Hawk, Toro	373-6781
67	Olsen	Joe	N	LTC	23 Hofman, Fresno	373-6779
68	Onasis	George	E	LTJG	9 2nd, Carmel	623-3140
69	Oscar	Tom	K	LTJG	61 David, Otay	623-3430
70	Owens	Bill	L	1LT	22 Pine, Carmel	373-6778
71	Palmer	Bob	L	LTC	84 Tamara, Marina	372-6412
72	Pappas	Nick	C	LTJG	11 3rd, Otay	623-3410
73	Patton	Mike	H	LTC	69 Lowell, San Jose	372-6413
74	Perry	Bill	H	LTC	38 Casa, Santa Clara	373-6414
75	Peters	Mark	J	LTC	62 Story, Salinas	372-6410
76	Peters	Nick	K	LTC	75 Wanda, Carmel	372-6411

77	Potter	Tom	E	CAPT	27 Rosita, Carmel	373-6782
78	Quill	Kelvin	J	CDR	29 6th st, Alto	373-6784
79	Quinn	Peter	C	CDR	28 10th st, Alto	373-6783
80	Rigas	Ben	H	LTJG	33 Pine, Otay	623-3420
81	Rivera	Mario	L	LCDR	50 Pine, Alto	373-6788
82	Roberts	Ben	J	CDR	31 7th, Alto	373-6786
83	Rodes	James	K	CDR	32 David, Fresno	373-6787
84	Rokos	Nick	A	CDR	30 7th, Carmel	373-6785
85	Ross	Allan	J	LCDR	51 Jakobs, Carmel	373-6780
86	Sanders	James	F	LTC	57 Sinex, Marina	372-6415
87	Scott	Paul	V	LTC	42 Grand, Carmel	372-6416
88	Spencer	Tim	M	LCDR	99 David, Carmel	623-3130
89	Stanley	Cris	K	LCDR	15 David, Salinas	623-3131
90	Takas	Costas	L	LT	8 Elden, Salinas	623-3135
91	Torres	Alex	A	LTC	60 Sinex, Fresno	372-6418
92	Turner	Carlos	B	LTC	15 Grove, Monterey	372-6417
93	Ullman	Rolf	G	CPT	22 Maple, Fresno	565-6210
94	Ulrey	Dan	A	CPT	23 Marge, Carmel	565-6211
95	Victor	David	R	CPT	64 Pine, Carmel	373-6629
96	Vongel	Mark	E	CPT	80 Hills, Fresno	375-6731
97	Warren	Geo	M	CPT	37 Holman, Carmel	373-6326
98	Waters	Gary	L	CPT	26 Vina, Monterey	376-6528
99	Watson	Ralph	D	CPT	41 Dolores, Marina	375-6427
100	Zikas	Tom	J	LT	16 Gordiou, Carmel	623-3136

D. MISCELLANEOUS OPERATIONS

These are auxiliary operations, which have been included in our Database System, to provide the Commander of a Formation and the aiding personnel, some statistical information about any work done in the database, as well as, some information concerning officers deleted from the current Unit. This information is recorded automatically during the whole operation of the database, in the files MONITOR, and STATISTIC, respectively. These files can be printed or deleted upon request. In addition, we have included in this function, a screen display, or printer output, of all the data concerning an officer.

The programs implementing this function of the system, with a short description of what each performs, are given below:

1. Program Printmon

This program prints the MONITOR file, which keeps track of who does what and when.

2. Program Printstat

This program prints the STATISTIC file, which contains some information about the officers who have been deleted from their previous Unit.

3. Program Delmon

This program deletes the data from the MONITOR file.

4. Program Delstat

This program deletes the data from the STATISTIC file.

5. Program Dispscrn

This program displays on the screen, or prints, depending on the user's choice, all the information included in an officer's record.

The listings of the above programs, as well as, the output if any, are shown in the next pages.

```
******** PROGRAM PRINTMON ********

* This program prints statistical information
* of who did what and when

ERASE
DO monitr

* Initialize memory variables

STORE "MONITORING THE USERS' ACCESS TO THE SYSTEM" TO title1
STORE "========================================" TO line1
STORE "  DATE          N A M E          JOB DONE" TO title2
STORE "      FILE/REP    # OF REC" TO title3
STORE "----------------------------------------" TO line2
STORE "------------------------" TO line3

* Print the headline

SET PRINT ON
? "                                                    "
?? "             "+DATE()
? "                                                    "
?? "             --------"
? "                  "+title1
? "                  "+line1
?
? "          "+title2
??  title3
? "          "+line2
?? line3
SELECT PRIMARY
USE monitor
DO WHILE .NOT. EOF

   * Print the main line

   ? "          "+date
   ?? "   "+name
   ?? "   "+job
   ?? "   "+fle:rept
   ?? "   "+STR(numrec,3)
   SKIP
ENDDO WHILE
RETURN
```

```
******** PROGRAM PRINTSTAT ********

* This program prints the statistic file which contains
* some information about the officers who have been
* deleted from their previous unit

ERASE

* Initialize memory variables

STORE "STATISTIC INFORMATION OF DELETED OFFICERS" TO title1
STORE "=========================================" TO line1
STORE "SER#       N A M E       RANK  UNIT   REPTDATE" TO title2
STORE "   DELDATE" TO title3
STORE "-----------------------------------------------" TO line2
STORE "----------" TO line3


* Print the headline

SET PRINT ON
? "                                                    "
?? "         "+DATE()
? "                                                    "
?? "         --------"
? "                  "+title1
? "                  "+line1
?
? "          "+title2
?? title3
? "          "+line2
?? line3
SELECT PRIMARY
USE statistic
DO WHILE .NOT. EOF

   * Print the main line

   ? "          "+serno
   ?? "   "+name
   ?? "   "+rank
   ?? "   "+unit
   ?? "   "+reptdate
   ?? "   "+deldate
   SKIP
ENDDO WHILE
DO monitr
RETURN
```

```
***** PROGRAM DELMON *****

* This program deletes the data of the MONITOR file

ERASE
? CHR(7)
@ 7,30 SAY "*** WARNING ***"
@ 9,17 SAY "The data of the MONITOR file will be deleted"
@ 11,31 SAY "PROCEED?  Y/N"
STORE " " TO answer
WAIT TO answer
IF answer = "y"
   SELECT PRIMARY
   USE monitor
   GOTO TOP
   DELETE NEXT 1000
   PACK
   USE
   ERASE
   @ 9,13 SAY "If you have made a mistake...,sorry it is late"
   @ 11,19 SAY "MONITOR file has been deleted"
   STORE 1 TO delay
   DO WHILE delay < 80
      STORE delay+1 TO delay
   ENDDO WHILE
   DO monitr
ELSE
   ERASE
   ? CHR(7)
   @ 9,20 SAY "*** Be careful, time is money ***"
   STORE 1 TO del
   DO WHILE del < 30
      STORE del+1 TO del
   ENDDO WHILE
ENDIF
RETURN
```

```
******** PROGRAM DELSTAT ********

* This program deletes the statistic officers file

ERASE
? CHR(7)
@ 7,30 SAY "*** WARNING ***"
@ 9,21 SAY "The STATISTIC file will be deleted"
@ 11,31 SAY "PROCEED?  Y/N"
STORE " " TO answer
WAIT TO answer
IF answer = "y"
   SELECT PRIMARY
   USE statistic
   GOTO TOP
   DELETE NEXT 1000
   PACK
   USE
   ERASE
   @ 9,17 SAY "The STATISTIC file has now been deleted"
   STORE 1 TO del
   DO WHILE del < 40
      STORE del+1 TO del
   ENDDO WHILE
   DO monitr
ELSE
   ERASE
   ? CHR(7)
   @ 9,20 SAY "*** Be careful, time is money ***"
   STORE 1 TO delay
   DO WHILE delay < 30
      STORE delay+1 TO delay
   ENDDO WHILE
ENDIF
RETURN
```

```
******** PROGRAM DISPSCRN ********

* This program displays on the screen or printer all the
* information concerning an officer

ERASE

* Initialize memory variables

STORE "     " TO mserno
STORE "   " TO rankcode
STORE "     " TO unitcode
STORE "   " TO dutycode
STORE "   " TO langcode
STORE "   " TO sccode
STORE "     " TO rankname
STORE "               " TO unitname
STORE "                     " TO dutyname
STORE "              " TO scname
STORE "             " TO langname
STORE "          " TO degrname
STORE "           " TO marstname
STORE 0 TO count
SELECT PRIMARY
USE master INDEX master
STORE "y" TO answer
DO WHILE answer = "y"
   ERASE
   @ 2,10 SAY "DISPLAYING OFFICERS' RECORDS"
   @ 5,10 SAY "Enter serial number" GET mserno PICTURE "9999"
   @ 7,10 SAY "Press 'ENTER' to exit"
   READ
   IF $(mserno,1,1) = " "

   * Update TEMPOR and MONITOR files, and return

      IF count <> 0
         SELECT SECONDARY
         USE tempor
         REPLACE counter WITH count
         USE
         DO monitr
      ENDIF
      RETURN
   ENDIF
   ERASE
   @ 5,10 SAY "Where do you want the display to be directed?"
   @ 7,10 SAY "    1 = Screen"
   @ 8,10 SAY "    2 = printer"
   STORE " " TO answer
   WAIT TO answer
   IF answer = "2"
      SET FORMAT TO PRINT
   ENDIF
```

```
* Find the officer in MASTER file

FIND &mserno
IF # = 0
   ERASE
   ? CHR(7)
   @ 10,10 SAY "This record does not exist, try again"
   STORE 1 TO del
   DO WHILE del < 30
      STORE del+1 TO del
   ENDDO WHILE
   LOOP
ENDIF

* Store the codes of certain information concerning the
* officer, to memory variables in order to be used as keys
* for finding the real data from the appropriate files

STORE rank TO rankcode
STORE unit TO unitcode
STORE duty TO dutycode
STORE educat TO sccode
STORE forlang TO langcode

* Retrieve the information from the auxiliary files

SELECT SECONDARY
USE ranks INDEX ranks
FIND &rankcode
IF $(p.unit,1,1) = "1"
   STORE armyname TO rankname
ELSE
   STORE navyname TO rankname
ENDIF
USE units INDEX units
FIND &unitcode
STORE title TO unitname
USE duties INDEX duties
FIND &dutycode
STORE name TO dutyname
USE forlangs INDEX forlangs
FIND &langcode
STORE name TO langname
USE sciences INDEX sciences
FIND &sccode
STORE name TO scname

* Store the appropriate name for marital status and degree

SELECT PRIMARY
DO CASE
   CASE degree = "B"
      STORE "Bachelor" TO degrname
```

```
      CASE degree = "M"
         STORE "Master" TO degrname
      CASE degree = "P"
         STORE "Ph.D" TO degrname
   ENDCASE
   DO CASE
      CASE marstat = "M"
         STORE "Married" TO marstname
      CASE marstat = "U"
         STORE "Unmarried" TO marstname
      CASE marstat = "D"
         STORE "Divorced" TO marstname
   ENDCASE

   * Display the record in the screen

   ERASE
   @ 1,10  SAY "DISPLAYING OFFICERS' RECORDS"
   @ 2,10  SAY "============================="
   @ 5,10  SAY "SERIAL NUMBER : "+mserno
   @ 6,10  SAY "NAME          : "+name
   @ 7,10  SAY "RANK          : "+rankname
   @ 8,10  SAY "UNIT          : "+unitname
   @ 9,10  SAY "SERVICE ENTRY : "+serentry
   @ 10,10 SAY "REPORT DATE   : "+reptdate
   @ 11,10 SAY "DUTY          : "+dutyname
   @ 12,10 SAY "EDUCATION     : "+scname
   @ 13,10 SAY "DEGREE        : "+degrname
   @ 14,10 SAY "FOREIGN LANG  : "+langname
   @ 15,10 SAY "MARITAL STATUS: "+marstname
   @ 16,10 SAY "CHILDREN      : "+children
   @ 17,10 SAY "ADDRESS       : "+address
   @ 18,10 SAY "TELEPHONE #   : "+phone
   STORE count+1 TO count
   SET FORMAT TO SCREEN
   @ 21,10 SAY "DISPLAY ANOTHER RECORD? (Y/N)"
   WAIT TO answer
   STORE "     " TO mserno
ENDDO WHILE answer = "y"
SELECT SECONDARY
USE tempor
REPLACE counter WITH count
USE
DO monitr
RETURN
```

V. CONCLUSIONS AND RECOMMENDATIONS

In this thesis, we tried to develop a personnel database system model, suitable for implementation within the Hellenic Armed Forces Formations. This system could also be applied to any subdivision of the Hellenic Armed Forces hierarchy, with only slight modification.

Our main goal is to increase productivity, effectiveness, efficiency, accuracy, and speed, as far as personnel management is concerned, as well as to decrease the national expenditure, and release manpower for other purposes. Also, the Commander of a Formation will be able to make fast decisions concerning personnel, which is important as well.

We used dBASE II as a database management system, since it is based on the relational model, which increases independency, and reduces redundancy. In addition, dBASE II contains its own programming language, which is a structured, high-level language, which is generally very efficient for manipulating data in the database.

We have implemented the most usually needed reports, but a wide variety of other reports, or simple queries, could also be created. Also, we tried to keep the programs as user friendly as possible, in order to help, not hinder, the operators in their job.

The software life cycle has been taken into account during the program development process. Programs are easy

to modify to meet future improvement needs. In our application we have used the top-down design approach which serves the above goal.

In this implementation we have included only officers, and a certain amount of data for each of them. Future improvements could include the entire military personnel, and we could add more information to the record of each individual, such as place and date of birth, medical information, military education, and statistical information concerning the units he has been assigned up to the present time, to name just a few.

This thesis constitutes a good basis for the future computerization of personnel management, in the Hellenic Armed Forces Formations, as well as in subordinate units.

LIST OF REFERENCES

1. Fairley, Richard, Software Engineering Concepts, McGraw-Hill Book Company, 1985.

2. Ullman, Jeffrey D., Principles of Database Systems, Computer Science Press, Rockville, Maryland, 1985.

3. dBASE II, Assembly-Language Relational Database Management System, Ashton-Tate, Culver City, California, 1981 User Manual.

4. Townsend, Carl, Using dBASE II, Osborne/McGraw-Hill, Berkeley, California, 1984.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22304-6145	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, California 93943-5100	2
3.	Computer Technology Curricular Office Code 37 Naval Postgraduate School Monterey, California 93943-5100	1
4.	CDR L. Rawlinson, Code 52Rv Department of Computer Science Naval Postgraduate School Monterey, California 93943-5100	1
5.	LCDR Paul W. Callahan, Code 5208 Department of Computer Science Naval Postgraduate School Monterey, California 93943-5100	1
6.	Hellenic Army General Staff Stratopedo Papagou Holargos Athens Greece	2
7.	Hellenic Navy General Staff GEN/B4 Stratopedo Papagou Holargos Athens Greece	4
8.	LTC Tsagaris Panagiotis Hellenic Army General Staff Stratopedo Papagou Holargos Athens Greece	2
9.	LT Karaiskos Constantinos Hellenic Navy General Staff Stratopedo Papagou Holargos Athens Greece	2
10.	MJ Strouzas Ioannis SMC 2989 Naval Postgraduate School Monterey, California 93943-5100	1

11. MJ Panagakos Georgios 1
Hellenic Air Forces General Staff
Stratopedo Papagou
Holargos Athens Greece

12. MAJOR Kondylopoulos Haralabos 1
Hellenic Air Force General Staff
Stratopedo Papogou
Holargos Athens Greece

DTIC

END

4-86