THESIS

AN AUTOMATED INDIVIDUAL TRAINING RECORD MANAGEMENT SYSTEM ( prototypes )
UNITED STATES MARINE CORPS

by

Ronald E. Pruett
David P. Haeusler

March 1984

Thesis Advisor: Norman R. Lyons

Approved for public release; distribution unlimited
An Automated Individual Training Record Management System (Prototype), United States Marine Corps

Ronald E. Pruieit
David P. Haeusler

Naval Postgraduate School
Monterey, CA 93943

Naval Postgraduate School
Monterey, CA 93943

Approved for public release; distribution unlimited

Training, Prototype, Individual System Development, Individual Training Records, Training Management, and Base II
An Automated Individual Training Record Management System (Prototype),
United States Marine Corps

by

Ronald E. Pruett
Lieutenant Colonel, United States Marine Corps
B.S., United States Naval Academy, 1967
M.S., University of Southern California, 1982

and

David P. Haeusler
Captain, United States Marine Corps
B.A., University of Colorado, 1977

Submitted in partial fulfillment of the requirements for the degree of
MASTER OF SCIENCE IN INFORMATION SYSTEMS
from the
NAVAL POSTGRADUATE SCHOOL
March 1984

Authors: 

Ronald E. Pruett

David P. Haeusler

Approved by:

Norman R. Lyons
Thesis Advisor

Second Reader

Chairman, Department of Administrative Sciences

Dean of Information and Policy Sciences
This thesis examines the feasibility of implementing an existing manual system, the Individual Training Record Management System, on a microcomputer. To demonstrate conceptual feasibility, a prototype is designed and implemented utilizing a commercially available database management system (DBMS). The prototype is not intended to be a fully operational system. Rather it provides an opportunity to demonstrate functions that could be fully implemented in similarly designed systems utilizing the basic criteria of simplicity, utility and low cost.
# TABLE OF CONTENTS

## I. INTRODUCTION
- A. GENERAL DESCRIPTION
- B. BACKGROUND
- C. OBJECTIVES
- D. ENVIRONMENT
- E. SYSTEM CHARACTERISTICS
  1. User friendly
  2. Reliability
  3. Cost
  4. Expandability of data base
  5. Expandability of functions
  6. Speed of operation
- F. SUMMARY
  1. Scope of Development
  2. Concept

## II. SYSTEM DESCRIPTION AND REQUIREMENTS
- A. INPUT INFORMATION
- B. SYSTEM SPECIFICATIONS (FUNCTIONAL)
  1. Training Data
  2. Data Base Maintenance
  3. Report Formats
  4. Data Retrieval
  5. Standard Reports
  6. Decision Support Subsystem
  7. Nonstandard Reports
  8. System Controls
  9. Output Mode
- C. REQUIRED SPECIFICATIONS (NON-FUNCTIONAL)
V. CONCLUSION ........................................... 64
A. GENERAL ........................................... 64
E. SYSTEM SPECIFICATIONS (FUNCTIONAL) ........... 65
  1. Storage Limitations ............................... 65
  2. Retrieval Limitations .............................. 65
C. SYSTEM SPECIFICATIONS (NON-FUNCTIONAL) ...... 65
  1. Training ........................................... 65
  2. Error Checking ..................................... 66
  3. Table Mapping ..................................... 66
D. SUBSEQUENT SYSTEM DEVELOPMENT ................ 66
E. SUMMARY ........................................... 67

APPENDIX A: DATA ELEMENT DICTIONARY ............... 68
A. FIELD .............................................. 68
E. NAME .............................................. 68
C. TYPE .............................................. 68
E. WIDTH .............................................. 69
E. DESCRIPTION ....................................... 69
F. DATA BASE FILES ................................... 69
  1. Pers (Personnel) .................................. 69
  2. EST (Essential Subjects Testing) ................ 70
  3. QUAL (Qualifications) ............................. 70
  4. INFOMISC (Information Miscellaneous) ........... 71
  5. SECURITY (Access Information) ................... 71
  6. COMMENTS (Non-formatted Information) .......... 71

APPENDIX E: DECISION SUPPORT SYSTEM (DSS) ......... 73
  DESCRIPTION ....................................... 73
A. BACKGROUND ....................................... 73
E. PURPOSE ........................................... 73
C. FUNCTIONAL DESCRIPTION ......................... 73
E. USER'S VIEW ....................................... 75
  1. Summary Information ............................. 75
  2. Support Dept ..................................... 76
3. Interdependence ........................................ 76
4. Restrictions ............................................. 76

E. BUILDER'S VIEW ........................................ 76
1. Dialogue .................................................. 76
2. Modelling ................................................ 76

F. CONTENTS OF KNOWLEDGE BASE ....................... 77

APPENDIX C: MTR'S MASTER CONTROL MODULE LISTING .......... 78
A. MASTER ROUTINE ......................................... 78
E. PAUSE ROUTINE ......................................... 90

APPENDIX D: ACCESS MODULE LISTING .......................... 81
A. ACCESS ROUTINE .......................................... 81

APPENDIX E: COMMAND DETERMINER MODULE LISTING ........... 83
A. COMMAND DETERMINER ROUTINE ......................... 83
E. HELP2 ROUTINE .......................................... 84

APPENDIX F: NONSTANDARD COMMAND GENERATOR MODULE
LISTING ..................................................... 86
A. NONSTANDARD REPORT ROUTINE ......................... 86
E. NONSTANDARD HELP ROUTINE ............................ 99
C. SELECT ATTRIBUTE ROUTINE ............................. 100
E. SELECT DESCRIPTOR ROUTINE ............................ 104

APPENDIX G: STANDARD COMMAND GENERATOR MODULE
LISTINGS ................................................... 109
A. STANDARD REPORT ROUTINE ............................. 109
E. SELECT UNIT ROUTINE .................................... 111
C. UNIT ROSTER ROUTINE .................................. 113
D. TRAINING STATISTICS ROUTINE ......................... 115
E. TRAINING COUNT ROUTINE ............................... 126
F. QUALIFICATION COUNT ROUTINE ......................... 128
G. TRAINING FCSTER ROUTINE ............................. 129
H. TRAINING FCSTER FORMAT ROUTINE .................... 132
I. MARKSMANSHIP STATUS ROUTINE ......................... 133
J. CONVERT UNIT ROUTINE .......... 135
K. QUOTA STUDY ROUTINE .......... 136
L. QUOTA SCRATCH PAD ROUTINE .... 140
M. MARKSMANSHIP ROSTER ROUTINE .. 155
N. MOS STATUS ROUTINE ........... 157
O. MOS ROSTER ROUTINE ........... 159
F. PERSONNEL DATA ROUTINE ....... 161
G. PERSONNEL DATA FORMAT ROUTINE .. 164
R. EST STATUS ROUTINE ........... 165
S. EST COUNT ROUTINE ........... 170
T. EST ROSTER ROUTINE ........... 171
U. EST REPORT FORMAT ROUTINE .... 174
V. SWIM QUALIFICATION ROUTINE ... 175
W. SWIM REPORT ROUTINE .......... 178
X. HELP ROUTINE .................. 179

APPENDIX H: MAINTENANCE MODULE LISTING ........ 186
A. MAINTAIN CONTROL ROUTINE .......... 186
B. VIEW CONTROL ROUTINE ........... 188
C. ITR SCREEN FORMAT ROUTINE ....... 190
D. SCREEN DATA RETRIEVAL ROUTINE .... 192
E. COMMENT RETRIEVAL ROUTINE ....... 194
F. UPDATE CONTROL ROUTINE .......... 195
G. DATA UPDATE ROUTINE ........... 197
H. ITR CREATION CONTROL ROUTINE ... 206
I. ITR DATA INITIALIZATION ROUTINE .. 207
J. DATA ERROR CHECKING ROUTINE ...... 211
K. CALCULATION ROUTINE ........... 214
L. COMMENT UPDATE ROUTINE .......... 219
M. COMMENT CREATION ROUTINE ...... 220
N. DELETE ITR ROUTINE ........... 221
O. SYSTEM FUNCTION CONTROL ROUTINE .. 223
P. SYSTEM INSTALLATION ROUTINE ... 224
Q. SYSTEM RESET ROUTINE .......... 225
R. SYSTEM ACCESS CONTROL ROUTINE .................. 229

BIBLIOGRAPHY ........................................... 232

INITIAL DISTRIBUTION LIST ............................ 233
LIST OF FIGURES

3.1 Initial Data Flow Diagram ............... 29
3.2 Data Flow Diagram Expansion for DBMS ........ 29
3.3 Data Flow Diagram Expansion 2 ............ 30
3.4 Data Flow Expansion for Access Function ........ 31
3.5 Data Flow Expansion for Command Generators ........ 32
3.6 ITRM System Hierarchy ................. 34
4.1 Command Determiner Menu ................. 44
4.2 Attributes and Descriptors Display ........ 45
4.3 Standard Report Menu .................. 47
4.4 Personnel Roster Report ................. 48
4.5 Training Status Report ................. 49
4.6 Marksmanship Status Report ............. 51
4.7 Marksmanship Roster Report ............. 53
4.8 MOS Status Report .................... 53
4.9 MCS Roster Report .................... 54
4.10 EST Status Report .................... 55
4.11 Swimming Qualification Status Report .... 56
4.12 Maintenance Module Menu ............... 57
4.13 ITR Screen Format .................... 58
4.14 System Function Menu ................ 62
4.15 Access Function Menu ................ 63
B.1 Memory File ......................... 77
I. INTRODUCTION

A. GENERAL DESCRIPTION

This interactive prototype automates the existing manual system of maintaining individual training information on USMC personnel. The term "system" is used throughout this thesis to mean hardware and software. The software consists of customized code which overlays a commercially available data base management system (DBMS). The system is targeted for implementation at the infantry or artillery battalion or subordinate unit level. However, with minor modifications, primarily increasing the limitation on the number of subordinate units and increasing the size of the database, the system could be used by any U.S.M.C. organization, including the Marine Air Ground Task Force (MAGTF). No prior user experience in computer or automated systems is assumed. In addition to generating standard reports, the system provides limited decision support capabilities, responds to non-standard data base inquiries, and includes facilities for data base maintenance.

E. BACKGROUND

Individual training subjects in the U.S. Marine Corps encompass a variety of otherwise unrelated training topics that are conducted and evaluated periodically. The common characteristic of individual training vis-a-vis unit training is that training and testing relate to individual skills and individual levels of performance. Since the focus is on individual training and individual testing, data is maintained for each individual in a unit and consequently, the system is very data intensive.
Individual training includes subjects such as "essential subjects", physical fitness, rifle/pistol qualifications, career training, and military occupational specialty (MOS) training. Each subject or grouping of subjects typically includes its own set of training and testing requirements. For example, some individual training subjects require proficiency demonstration only once during an entire career while others must be evaluated annually. In total individual training information includes approximately 60 separate elements of information that must be maintained for each Marine. Individual Training Records (ITR's) are traditionally maintained manually at the company level. A database consisting of 200 records and 10,000 data elements in a company size unit is not uncommon. In addition to database maintenance, which includes frequent creation, deletion, and update of records, the database is often used to provide training management information concerning the current training status of the unit or as a basis for developing future training plans.

It is the contention of this thesis that the system can be operated and maintained by a technically unsophisticated user at a reasonable cost.

C. OBJECTIVES

The objective of this thesis is to design and implement a prototype system that is simple to use and that:
1. Maintains individual training records for a USMC battalion;
2. Provides standard reports upon request;
3. Provides nonstandard reports upon request;
4. Demonstrates a decision support application;
5. Provides a database maintenance capability.
I. ENVIRONMENT

The system is used and maintained by persons with no prior experience with computers and with little prior training on this system. The system's data base requires frequent maintenance. Approximately a 30% record turnover rate (i.e., 30% of old records will be deleted and 30% new records will be inserted into the data base) is expected each year. Additionally, records will be modified at the rate of approximately 50 data fields per work day for each company (based on an average strength of 150 Marines per company). The physical environment envisioned is a typical office setting with no unique system requirements anticipated.

2. SYSTEM CHARACTERISTICS

The following system characteristics are listed in order of importance.

1. User friendly

The system is simple to operate and maintain. It requires no extended training sessions and it provides prompts to the user.

2. Reliability

The reliability of the hardware and the DBMS is beyond the scope of this thesis. The code written to implement this design is the user-system interface. Accordingly, to ensure the integrity of the system, the DBMS remains transparent and inaccessible to the user. The overall reliability of the system is high. To this end, the system includes the following provisions:

a. Menu driven
t. No possibility of accidental erasure of data base.

c. No possibility of user to "escape" or default to DBMS.

2. Cost

The costs of a microcomputer and the proprietary software required to run the system are low compared to mainframe or mini-computer application.

4. Expandability of data base

The data base can be expanded subject to memory and DBMS constraints. The larger the data base the slower the system will respond to user requirements. This is, however, an acceptable tradeoff.

5. Expandability of functions

The system can be easily expanded to include added functions. This facility has been implemented by incorporating a modular structure in the design. A module can be added or modified with little difficulty.

6. Speed of operation

The speed of the system is relative to the size of the data base. However in all cases compared to the present manual system it is much faster.

1. SUMMARY

This thesis includes the determination of system requirements, system feasibility, system design and review, and implementation of a functional prototype.
2. Concept

The purpose of this thesis is to demonstrate feasibility of the concept: A fully automated individual training record management system for a battalion size unit can be implemented on a microcomputer. The system can be developed and operated at a reasonable cost. Furthermore, the system can be operated and maintained by persons having no prior computer experience after little training.
II. SYSTEM DESCRIPTION AND REQUIREMENTS

A. INPUT INFORMATION

The user's input consists of a sequence of menu prompted responses. Each user's response is either a specific request to the system for data base maintenance, a standard report, or data necessary to generate the desired nonstandard report.

B. SYSTEM SPECIFICATIONS (FUNCTIONAL)

1. Training Data

The system is capable of developing and maintaining training data for at least 800 Marines. The following information is maintained for each individual:

a. Name/SSN/personal data
b. Bank
c. Unit (company and platoon)
d. Annual training requirements status (essential subjects, leadership, SNCO/MCO training, etc.)
e. Rifle/pistol qualifications status (current year)
f. MOS qualifications
g. An unformatted data field to be utilized as desired by the user. It may be used to maintain the following kinds of information:

   (1) Formal schools attended.

   (2) Local schools attended.
(3) MCI courses completed.

(4) Participation in training exercises/training deployments.

2. **Data Base Maintenance**

   Maintenance of the data base is simple and provides positive user control at all times (i.e., user inputs data and views updated individual training record before data is stored into memory).

3. **Report Formats**

   Most standard reports are retrievable in a statistical mode or in a unit roster mode.

   a. **Statistical Mode**

      The statistical mode report provides the total or the percent of a unit that has satisfied a particular user-specified training criterion (What percent of a unit is swim qualified?).

   b. **Roster Mode**

      The roster mode report provides names of persons in a specified unit that have satisfied a training criterion.

4. **Data Retrieval**

   All standard reports are retrievable by unit designator (e.g., battalion, company/platoon). Training data is retrievable by individual, by unit, or by a user specified set of attributes (e.g., all sgts in A CO) utilizing the nonstandard reports option.
5. **Standard Reports**

The system provides the following standard reports:

a. **Unit Roster**

The Unit Roster Report will display in alphabetical order the NAME, RANK, and PRIMARY MOS for each member of the selected unit or subunit.

b. **Personal Data Report**

The Personal Data Report displays for each member of a unit or subunit all personal data including NAME, SSN, RANK, PRIMARY/SECONDARY MOS, UNIT/SUBUNIT NAME, JOIN DATE, EAS, BIRTHDATE, HEIGHT, WEIGHT and a one character boxclean comment block.

c. **Essential Subject Training Requirements Status Report**

The Essential Subject Training Requirements Status Report displays the percent and total number of unit or subunit members that have completed an Essential Subject requirement or the PFI (physical fitness test).

d. **Annual Training Requirements Status and Roster Reports**

The Training Status Report displays for each essential subject, the percent of the unit that has successfully completed each training element and the total number of unit members that have completed each training element. In addition, the Training Status Report includes completion statistics for PFT1, FPT2, SWIMQUAL, RIFLE QUAL, and PISTOL QUAL.

The Training Roster Report displays in alphabetical order each member of the selected unit or subunit and
his complete training status for the current year. If a training element has been successfully completed for the current year, a letter corresponding to that training element will be shown. An "*" symbol indicates that a training element has not been completed. For example, "FIRST AID AE*H**B**EF" indicates that elements ABDEF and H have been completed and five of the eleven elements have not been completed (i.e., C, G, I, J and K).

e. Rifle/Pistol Qualifications Status and Roster Reports

The Marksmanship Status Report will display the qualification results for the unit or subunit members that have fired the rifle or pistol for qualification during the current year. Also, the percent that have qualified in each category (e.g., EX, SS, MM, UN) and current year results for rifle and pistol will be shown. The pistol qualification results include only those members required to fire the pistol for qualification.

The Marksmanship Roster Report will display in alphabetical order the NAME, RANK, DATE OF QUALIFICATION, and QUALIFICATION RESULTS for each member of the selected unit or subunit. Current year results only will be displayed.

f. MOS Qualification Status and Roster Reports

The MOS Roster Report groups the selected unit or subunit by primary MOS and RANK and then displays in alphabetical order the NAME, RANK, PRIMARY MOS AND SECONDARY MOS of all members in the unit or subunit. This report will function regardless of the number or different types of MOS's contained in a unit.
g. Individual Training Record Report

The Individual Training Record Report prompts the user for the name of a member in the data base. After a name is entered, the complete individual training record for that individual is displayed. This includes all training data included in the data base on this particular individual (i.e., essential subject status, PFT status, current year rifle/pistol qualification results, swim qualification status) and personal data (i.e., name, rank, SSN, unit, primary MOS).

h. Swim Qualification Report

The Swim Qualification Report displays in alphabetical order for each member of the unit or subunit selected NAME, RANK, UNIT and current SWIM QUAL results.

6. Decision Support Subsystem

The purpose of this component is simply to demonstrate the decision support potential of this system using the available data base.

7. Nonstandard Reports

The system generates user defined non-standard reports. These reports are created from attributes that the user determines. A maximum of three attributes can be selected.

8. System Controls

Safety controls are provided to ensure that the data base cannot be accidentally written over or otherwise destroyed. Also, the system includes an executive access control facility, and control for "read only" access or "read/write" access. System access is permitted in one of the following three levels:
a. Data Base Administrator level is intended for the sole use of the Data Base Administrator and includes full system capabilities (install system, initialize data base, modify access directory, create/delete/modify data base records).

b. Create/modify data base level
c. Restricted to retrieve information in report format only from the data base.

9. **Output Mode**

   The prototype is capable of sending reports to a CRT screen. With minor modifications the user can be given an option of screen or printed copy output.

**C. REQUIRED SPECIFICATIONS (NON-FUNCTIONAL)**

   The following is a list of non-functional system specifications:

   1. The system is interactive.
   2. A high school graduate with no computer background can retrieve standard reports with no more than one hour of system training.
   3. A high school graduate with no computer background can create and maintain the data base with no more than six hours of training.
   4. A high school graduate with no computer background can retrieve a standard report after one hour of training.
   5. A high school graduate with no computer background can retrieve a multiple-criteria nonstandard report after an additional one hour of training.
   6. The system can be easily modified to change record formats, field formats, and standard report formats.
without a major redesign or reprogramming effort. A major redesign is considered any design effort that involves changing a module interface.

7. The system can retrieve standard reports in 15 minutes or less.
8. The system can retrieve non-standard reports in 30 minutes or less.
9. The system does "out of bounds" checking/parameter checking on all quantitative data and standard inputs.
10. The system automatically converts rifle, pistol, and PPI raw scores into corresponding qualifications (e.g. 245 rifle raw score => Expert).

D. DATA STRUCTURE DEFINITIONS (DATA DICTIONARY)

1. Introduction

The data dictionary should be maintained by the Data Base Administrator and documentation should be accessible to all users. This data dictionary facilitates maintenance, the integrity of the information in the data base and provides information on data relationships.

2. Data Base Management System

a. Commercial Software

The Data Base Management System (DBMS) kernel is a commercially available software system. The program dBase II, a relational DBMS, provides all of the necessary attributes (availability, record capacity, user friendly environment, etc.). Therefore, dBase II (Ashton-Tate version 2.4) is used, and data structures are implemented by the dBase II system.
1. Data Base Files

The information maintained on each individual is organized in a relational data base. The following records or tuples are defined for each individual (The number of characters per field is defined in the data element dictionary, see Appendix A):

(1) **Personal Information Record.** SSN, name, military occupational specialty (MOS), rank, weight, height, company/platoon, birthdate, joindate, expiration of active service (EAS), and gas mask size.

(2) **Essential Subject Training Requirements.** SSN, 9 essential subjects, and physical fitness information.

(3) **Rifle/Pistol/Swimming Qualifications.** SSN, rifle score current year, pistol score current year, swimming qualifications, and dates.

(4) **Information Miscellaneous.** SSN, 11 fields for training information.

(5) **Comment.** SSN; 250 characters that can be utilized, at the user's discretion, for formal/informal school data, deployments information, MCI courses completed or specific comments.

(6) **Security.** SSN, name, user identification code, password, and authorization level.
III. METHODOLOGY AND DESIGN

A. METHODOLOGY

1. Software

At the center of the software subsystem is the IBM Personal Computer Disk Operating System (DOS 2.0). System requirements for PC DCS operating systems includes:

a. 8088 based microprocessor

b. 128K bytes of RAM memory

c. Cursor addressable 24 line by 80 column CRT

The software layer immediately outside DOS 2.0 is dBase II (version 2.4) which is a commercially available relational data base application system developed and distributed by ASHTON-TATE. dBase II is a data base management tool that permits manipulation of user designed data base files using English-like commands which collectively define a query language.

The layer of code contiguous with and completely surrounding dBase II is what we designed and wrote to satisfy the thesis objective. The code is written in the dBase II query language. This layer of code, that will be referred to as the "cuter layer", is designed to be the only interface between the system user and the system. All system entries, data base transactions or manipulations, report inquiries, and system exits are through the cuter layer of code. This is an important and necessary design consideration for the following reasons:
a. Simplicity

In order to satisfy the system requirement for simplicity, it was necessary to impose a software layer between the user and the dBase II software. This outer layer presents to the user a series of menus. The user responds to each menu prompt by pressing the keyboard key that corresponds to action desired. The user's response in turn initiates a code sequence that causes the desired action to be taken. If additional user input is required to complete the action, the code causes another menu to be presented to the user. Each menu represents an exhaustive list of possible options that may be selected by the user. This sequence of menu presentation and user response is repeated until the desired action has been adequately defined. At this point the code executes using as variables the input provided by the user. Therefore, this outer layer includes all of the code that determines the functions available to the user and simplifies the operation of the system.

b. Layering

If the user had access to the DOS 2.0 or the dBase II layer he could easily modify the outer layer code. This action could cause a system failure. Therefore, this eventuality must be prevented. Another function of the outer layer is to prevent access to inner layers of software.

c. Data Integrity

Since the database files contain personal data, that needs to be protected, it is important to limit access to the data base. Also, uncontrolled access to data base files can result in lost data or catastrophic destruction of
the data base. To preclude this, the outer layer provides system security and ensures data integrity.

d. File Maintenance

Whenever data base files are accessed, housekeeping functions need to be performed such as opening or closing files, joining files, creating temporary files or deleting temporary files when no longer needed, and creating or releasing variables. The outer layer automatically performs these functions for the user.

E. SOFTWARE DESIGN

1. Structured Programming

Although the dBase II query language is not a structured programming language, structured programming techniques were employed during the software design. This was done to reduce system maintenance, to simplify system design and to make future changes to the system easier to implement. The following techniques were emphasized during the design phase of the development cycle.

a. Top down design and hierarchical structure

b. Modularization

c. Information hiding

d. In general, efforts were made to maximize module cohesion and to minimize coupling between modules.

2. Top-down Design

The overall design concept was the top-down design or step-wise refinement technique. This concept provides the capability to work from a simple idea and expand it to the final complex product. This process facilitates
and logical development of the design steps. The first step in the process is to determine how the data or an input is transformed to produce the desired output. The concentration is on 'What' is done not 'How' it is done. With this information data flow diagrams are produced. These diagrams should be considered arrows pointing to 'black boxes' which manipulate the data. The actual mechanisms of the manipulations are not important at this point in the design process. The diagrams show the information passing through the system and the transformation of data in the system. These diagrams are produced at an increasing level of detail until further expansion does not provide a significant change.

From the data flow diagrams, hierarchical charts are produced. Any information modifications that are related, are grouped into the same module. Once the hierarchy is established, algorithms are developed using structured programming techniques. These algorithms are then converted into code. The code is implemented using the top down programming technique, this technique implements the highest levels of the hierarchy first. As lower level modules are completed and tested, they are added to the existing modules. This technique simplifies a complicated program.

C. DATA FLOW DIAGRAMS

At first, it is necessary to decide what information goes into the system and what information exits from the system. The user inputs both data and commands from a terminal and receives, from the system, reports and information. The first iteration of the data flow diagram has two inputs and two outputs (see Figure 3.1).

The Individual Training Record Management (ITRM) System is responsible for maintaining a large amount of data. Handling of the information should be transparent to the
user. Therefore, a Data Base Management System (DBMS) was incorporated to maintain the data base. The DBMS requires specific commands as input and outputs information (see Figure 3.2).

The ITRM System is designed to have a restricted access. The user must enter the proper identification for the system to allow access. The access function produces a valid/invalid acknowledgement to the remaining portion of the ITRM system. Commands and data pass through the Command Determiner which selects the type of action the system is to execute. The three functions called by the Command Determiner are:

1. Standard Command Determiner
2. Nonstandard Command Determiner
3. Maintenance Handler

Figure 3.1 Initial Data Flow Diagram
These three functions interface with the DBMS by producing DBMS commands. The DBMS then communicates to the Report Generator the information necessary to produce the desired reports. Figure 3.3 shows the second expansion.

The access control of any system can be very complex. This system requires the user to input an identification code and a password. The system then matches these items to insure an authorized user. If access is valid, the user's authorization level is passed to the rest of the system. If access is invalid, the user gets one more opportunity to enter a correct access sequence. If the user fails again the system terminates the session. The information required to handle the access function is stored in the DBMS. The
Figure 3.3 Data Flow Diagram Expansion 2

The premise of the entire system is to prompt the user into entering the correct response, thereby reducing the need for any prior experience. The ITFM system prompts the user for the proper access codes and informs him when an error has been made. Figure 3.4 displays the Access data flow expansion.

Each of the Command Determiners prompts the user for responses from which DBMS commands are derived. The report
Figure 3.4  Data Flow Expansion for Access Function

generator then interprets the DBMS replies and prompts the user to format the reports (Figure 3.5).

The data flow diagrams, demonstrate the simple and logical flow of the data through the ITRM System. The next step is to logically organize the flow diagrams into a hierarchy of modules that accomplish the data transformations depicted in the data flow diagrams.
Figure 3.5 Data Flow Expansion for Command Generators

I. SYSTEM HIERARCHY

The IIBM System is relatively simple in design. It requires maintenance of the data base, production of
standard reports, and simple query capabilities for the user through nonstandard reports. The system must be initialized and sequential control established. The ITMS Master Control Module was created to handle the functions of system control and initialization. This module is the highest level of the hierarchy, level 1. All access functions are consolidated in one module, the Access Module. The Command Determiner Module controls which functions the user can invoke and checks for proper authorization level. The Access and Command Determiner Modules comprise the second level of the hierarchy.

The Standard Command Determiner, Nonstandard Command Determiner, and the Maintenance Handler are combined with portions of the Report Generator to form three modules, the Standard Report Generator, the Nonstandard Report Generator, and the Maintenance Module. These three modules interface with the DBMS and produce DBMS commands. In addition, they process the information returned from the DBMS and produce the appropriate reports. Consolidation of determiner and report generation functions simplifies the structure of the system and facilitates subsequent maintenance. The three modules, Standard Report Generator, Nonstandard Report Generator, and Maintenance, comprise the third level of the hierarchy. The lowest layer of the hierarchy contains the DBMS. The hierarchy displayed in Figure 3.6 demonstrates simplicity of design.

I. MODULE DESCRIPTIONS

1. ITMS Master Control Module
   a. Module Purpose

   The ITMS Master Control Module initializes the system and controls the logical sequencing of all modules.
Figure 3.6 ITRM System Hierarchy

1. Module Function

The ITRM Master Control Module initiates and coordinates all internal system processes (except operating system functions). The module responds to internal cues (from other modules) and to environmental cues (from the user). The module controls the access to all modules and shields the user from the resident operating system. The module will not allow the user to exit to the operating system.
c. Interfaces

The master control module is at level 1 which is the highest level in the module hierarchy. It interfaces with two modules at level 2, Access Module and Command Determiner Module.

c. Design Decisions

This module can be easily modified to handle any expansions by adding new module interfaces.

2. Access Module

a. Module Purpose

The Access Module prompts the user for an identification code and password. The module determines if the user has authorized access and assigns the corresponding level of access.

b. Module Function

The Access Module requests the user's identification code and password. It validates the ID to ensure the user is on the access list and to ensure that the password corresponds with the user ID. If the user ID is not on the access list an appropriate message is returned. If the password does not correspond with the user ID (after two iterations), an appropriate message is displayed and the session is terminated. Once the correspondence of ID and password has been determined, the module passes the access authorization level to the command determiner. The authorization level is used to determine which menu options are available to the specific user.
c. Interface

The Access Module receives control instructions from the ITRMS Master Control Module. The Access Module interfaces directly with the DBMS to manipulate the security file. After the access module completes its processing and if the user is authorized access, the level of access is passed back to the ITRMS Master Control Module.

d. Design Decision

The design allows modifications to be made to the access system without side effects.

3. **Command Determiner Module**

a. Module Purpose

The Command Determiner Module directs control to either the Standard Report Generator, Nonstandard Report Generator, or the Maintenance Module.

b. Module Function

The Command Determiner asks the user, through a standard set of menu prompts, what task he wishes to accomplish. Based on the user response, the Command Determiner Module transfers control to one of the three lower level modules. The Command Determiner Module also checks to see if the user has access to that module. If access is not authorized then an appropriate message is displayed.

c. Interfaces

The Command Determiner is called by the ITRMS Master Control Module and passes control to either the Standard Report Generator, Nonstandard Report Generator, or the Maintenance Module.
c. Interface

The Access Module receives control instructions from the ITRMS Master Control Module. The Access Module interfaces directly with the DBMS to manipulate the security file. After the access module completes its processing and if the user is authorized access, the level of access is passed back to the ITRMS Master Control Module.

d. Design Decision

The design allows modifications to be made to the access system without side effects.

3. Command Determiner Module

a. Module Purpose

The Command Determiner Module directs control to either the Standard Report Generator, Nonstandard Report Generator, or the Maintenance Module.

b. Module Function

The Command Determiner asks the user through a standard set of menu prompts, what task he wishes to accomplish. Based on the user response, the Command Determiner Module transfers control to one of the three lower level modules. The Command Determiner Module also checks to see if the user has access to that module. If access is not authorized then an appropriate message is displayed.

c. Interfaces

The Command Determiner is called by the ITRMS Master Control Module and passes control to either the Standard Report Generator, Nonstandard Report Generator, or the Maintenance Module.
c. Design Decisions

This module allows the separation of the Standard Report Generator, the Nonstandard Report Generator, and the Maintenance Module. This separation facilitates changes and reduces module coupling.

4. Nonstandard Report Generator Module

a. Module Purpose

The Nonstandard Report Generator allows the user to query the database in a controlled manner. The queries are generated from a series of prompts to the user.

b. Module Function

The module prompts the user, through a series of menus, to produce the desired query. When the module receives the appropriate prompted input from the user, the nonstandard command generator accesses the DBMS for the requested information. The information is then formatted and displayed.

c. Interfaces

The Command Determiner Module calls this module based on user selection. The Nonstandard Report Generator interacts directly with the user. This module constructs the queries and communicates them to the DBMS. It also processes the information received from the DBMS and formats the data into a report.

d. Design Decisions

If there is a future need to modify the types of queries that are available to the user, the changes will be confined to this module.
5. **Standard Report Generator Module**

a. Module Purpose

The Standard Report Generator produces the DBMS commands that will create the standard reports. These reports are produced on a regular basis, therefore, it is advantageous to standardize the reports.

b. Module Function

The Standard Report Generator displays option menus to the user and translates user's responses into the necessary sequence of DBMS instructions. After the user decides which standard reports will fulfill his needs, the module formats standard reports and displays the user requested information. The Standard Report Generator includes a DSS subset.

c. Interfaces

The Standard Report Generator is called by the Command Determiner. The Standard Report Generator communicates with the DBMS and processes information from the DBMS to produce its reports.

d. Design Decisions

This module was designed to take advantage of the fact that many applications are routine in nature. These are programmed in a single module and significantly simplify access to routine reports. If any of these routine reports need modification, the change can be isolated to this module.
6. Maintenance Module

a. Module Purpose

The Maintenance Module provides users the capability to create, delete, and update the data base records. Necessary system functions reside in this module. The Data Base Administrator maintains the access file from this module and initializes the system to his specific unit.

b. Module Function

To update a record, the Maintenance Module extracts the data in each data base file for a selected unit member. The data is combined into a single screen display representing all data contained in the system on the individual. To update a field, the data in the field can be typed over with current data. Upon completion, each data element is read and returned to its proper location in the appropriate data base file. Creation of new records is accomplished in the same manner. Records are deleted by a single deletion command. Therefore, this command usage is limited to avoid accidental and malicious destruction of data.

c. Interfaces

The Maintenance Module interfaces with the Command Determiner and the DBMS.

d. Design Decisions

All data modifications are implemented using the Maintenance Module. The physical and logical structure of all data base files is hidden from the user.
IV. IMPLEMENTATION

B. CONSTRAINTS

1. dBASE II

Following is a list of dBase II constraints that affected the design of this system and consequently imposed constraints on the implementation of the Individual Training Record Management System:

a. No more than 65535 records can be used in a single data base file.

b. No more than 32 fields can comprise any single record.

c. No more than 254 characters can be placed in any one field.

d. No more than 1000 characters can be placed into any single record.

e. No more than 64 variables can be located in working storage at any given time.

2. Hardware

Following is a list of development hardware constraints that affected the design of this system and consequently impose constraints on the implementation of the Individual Training Record Management System.

a. Main memory storage (RAM) is 512K.

b. Two double density double sided disk drives were used for a total of 720K bytes of disk storage.
c. The IBM PC 8088 microprocessor operates at 4.7 MHz.

To maximize the amount of data that the system maintains, a RAM disk drive is created. This RAM drive allows the program to be copied into memory and enables the full use of the two disk drives for data. Use of the RAM drive increases the speed of the program by allowing the various call/d routines to be read more quickly into the working storage. The memory access time for the IBM PC is much faster than disk access time. This increase in speed reduces the response time to the user. The objective of the prototype is to handle the data required to maintain a battalion sized unit's training records. The amount of storage required for 800 individuals is approximately two megabytes. This allows for the overhead that the DEMS uses and the room necessary to execute functions that require the creation of temporary files. The available storage in the prototype system limited the test data base to a company sized unit.

3. Environmental

The following environmental considerations constrain the system by placing added requirements on the ultimate design.

a. Inexperienced users
b. Frequent turnover of personnel
c. Limited training time available
d. Hardware maintenance must be readily available from outside commercial sources.
e. Software Maintenance is not intended to be performed at the user level because computer programming skills are not necessarily available.
f. No unusual requirements for physical environment such as smoke or dust free atmosphere, unusual electrical power characteristics or hookup, or vibration dampening, are required.

B. GENERAL

The outer layer consists of six separate modules divided into three levels. This section will briefly describe the routines that make up each module and the system output from each routine. The listings of the programs are in the appendices, and can be examined for further detail.

1. ITRM Master Control Module

a. Master Routine (Master.prg)

This module controls the sequencing of the program. It is an endless loop which requires the computer to be either turned off or re-booted to exit the ITRM system. The Master routine requires the user to switch diskettes. The user receives a message to physically exchange the program diskettes with the data diskettes. If the diskette exchange is not made or is made improperly the user is prompted again to make the proper exchange.

The Access Module is initiated by the Master Routine and if a valid indication is returned, the Master Routine will initiate the Command Determiner Module. Once the Command Determiner Module is exited, the Master Routine loops to the beginning to receive a user command to initiate the Access Module.

b. Pause Routine (Pause.prg)

This routine provides a time delay for the system. When this routine is called, it delays the functioning of the system for approximately thirty seconds.
This routine is used to delay error messages that are displayed for a short period of time.

2. **Access Module**

   a. Access Routine

   The Access Routine prompts the user for an identification code. The routine then instructs the DBMS to search for that code in the security data base file. If the code is not found a message is displayed and the user can try again. If the code is found the corresponding password from the data base is stored in memory. The user is then prompted for the password. When it is entered, if it does not match the password in memory an error message is displayed and the user can try again. If the password that the user enters matches the password in memory, a valid user state is passed to the Master Routine with a user authorization level.

3. **Command Determiner Module**

   a. Command Determiner Routine (Cmddet.prg)

   The Command Determiner Routine displays to the user a menu. The user can choose to initiate the Nonstandard Report Generator, Standard Report Generator, Data Entry and System Functions (Maintenance), Help, or Exit. The user enters the appropriate letter which is processed by a case statement that initiates the selected module. Figure 4.1 displays the Command Determiner Menu.

   b. Help2 Routine (Help2.prg)

   The Help2 Routine provides information to the user on the functions that can be initiated from the Command Determiner and the Maintenance Module. The information is presented to the user to help in the selection of the
appropriate module and is not designed to explain the functioning of the entire system. The system's built-in prompts walk the user through the necessary steps to accomplish the desired task.

4. Nonstandard Report Generator Module

a. Nonstandard Report Routine (Nostdrpt.prg)

This module gives the user an option to go directly to the report or to view a help module. If the user desires to go directly to the report, this routine queries the user for the first set of attributes and attribute descriptions. After the user's selections are obtained, the routine searches the appropriate data base files to locate all records that satisfy the user specified attribute and attribute description. Attributes are field names such as RANK, COMPANY, RFLQUAL, etc. Up to three attributes from any data base files (the data base file location is invisible to the user) may be specified by the user. Multiple attributes are automatically joined by the routine using a logical "and" operation. The routine
employs a series of case statements. For each attribute selected, the attribute name is stored. Slctdesc.prg is called which stores an attribute description such as F5 (for RANK) or A CO (for COMPANY) or EX (for RFLQUAL). After the user has finished selecting attributes and attribute descriptions, the routine identifies the correct database files and either creates a temporary file containing the needed information or, if only two database files are involved, primary and secondary files are identified. The report is then displayed. Figure 4.2 depicts the sequence of screen displays for attributes and descriptions of "COMPANY is A CO". AND. "RANK is E5". AND. "PLATOON is 1ST PLAT".

<table>
<thead>
<tr>
<th>Attributes</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMPANY</td>
<td>IS A CO</td>
</tr>
<tr>
<td>RANK</td>
<td>IS E5</td>
</tr>
<tr>
<td>PLATOON</td>
<td>IS 1ST PLAT</td>
</tr>
</tbody>
</table>

Press 'ENTER' to continue

Figure 4.2 Attributes and Descriptors Display

b. Nonstandard Help Routine (Nostdhlp.prg)

This routine responds to the user's request for "HELP" while in the Nonstandard Report Menu. It displays a brief explanation of the attribute and attribute description process.
c. Select Attribute Routine (Sltartt.prg)

This routine prompts the user to select attributes. After the first and second selection the user is asked if he wants to select another attribute (only three attributes are permitted, therefore after the third attribute and attribute description set have been selected the routine automatically executes the user request). If the response is yes, the selection procedure is repeated. This process is continued until the user answers no or three attributes have been selected. The routine locates the appropriate data base files and displays the NAME, SSN, and attribute information for each record that satisfies the user's specified criteria.

d. Select Descriptor Routine (Sltdesc.prg)

This routine prompts the user to select attribute descriptions. Each attribute description is stored into a memory variable for later use when the database files are searched. An attribute description is selected for each attribute.

5. Standard Report Generator Module

a. Standard Report Routine (Stdrt.prg)

This routine generates the primary menu for standard report selection and stores the user's response in a memory variable. Control is then passed to the appropriate subroutine to obtain the necessary data from the database. This routine will continue until the user responds with a "C" (QUIT). Figure 4.3 displays the Standard Report Menu.
STANDARD REPORT SELECTION
SELECT REPORT BY ENTERING THE APPROPRIATE LETTER

A. UNIT ROSTER
B. TRAINING STATUS REPORT
C. TRAINING ROSTER REPORT
D. INDIVIDUAL TRAINING RECORD
E. MARKSMANSHIP STATUS REPORT
F. MARKSMANSHIP ROSTER REPORT
G. MOS STATUS REPORT
H. MOS ROSTER REPORT
I. PERSONAL DATA REPORT
J. EST STATUS REPORT
K. EST ROSTER REPORT
L. SWIM QUALIFICATION REPORT
M. HELF
N. QUIT

SELECT OPTION ===>:

Figure 4.3 Standard Report Menu

b. Select Unit Routine (Selunit.pro)

This routine prompts the user to select the size and name of the unit (i.e., battalion, company, platcon) for the report generation. The user responds with a letter that corresponds to a user installed company or platcon designation.

c. Unit Roster Routine (Unitrost.pro)

This is a control routine which creates the personnel roster for the unit specified by the user's response to the Selunit Menu. Figure 4.4 displays a copy of the Personnel Roster Report.

d. Training Status Routine (Trngstat.pro)

For the selected unit, this routine calculates the total number of training elements completed for each training category. For example, the training category
UNIT ROSTER

<table>
<thead>
<tr>
<th>NAME</th>
<th>RANK</th>
<th>PRIMARY MOS</th>
</tr>
</thead>
<tbody>
<tr>
<td>BAKEF, STEVE N.</td>
<td>E4</td>
<td>0311</td>
</tr>
<tr>
<td>CAFY, R. V.</td>
<td>E5</td>
<td>0311</td>
</tr>
<tr>
<td>CASTIN, D. P.</td>
<td>E1</td>
<td>0311</td>
</tr>
<tr>
<td>DCNNELY, R. G.</td>
<td>E6</td>
<td>0331</td>
</tr>
</tbody>
</table>

PRESS 'ENTER' TO CONTINUE

Figure 4.4 Personnel Roster Report

"HISTORIC" consists of three training elements (A, B, C). A percent completed figure is calculated for each training element by dividing the total training elements completed (EX. ZZAHIS, ZZBBIS, ZZCHIS) by the total number of personnel in the unit (ZZTOTAL). Both the totals for each element and the percent completed figures are shown (see Figure 4.5).

e. Training Count Routine (Cntest.prg)

This routine counts the number of essential subject training elements that have been completed in the unit or subunit specified.

f. Qualification Count Routine (Cntqual.prg)

This routine saves all memory variables in a memory file (ZZTOTAL) and then counts the number of persons that have not completed SWIMQUAL, RPLQUAL, PSTQUAL and persons in the unit not required to qualify (NA).
Figure 4.5 Training Status Report

9. Training Roster Routine (Tngrost.prg)

This routine provides the Training Roster Report. It starts at the top of the Pers.dbf and looks at each record individually. If a record corresponds to the unit selected by the user, the routine Tngrpt.prg is called which collects and displays the training data that corresponds to the current Pers.dbf record (e.g., If the user wants training data on every member of 1ST PLAT, A CO, Tngrost.prg locates each record in the Pers.dbf which belongs to 1ST PLAT, A CO. Tngrpt.prg then displays the training data from the appropriate data base files.). Information on one individual is displayed in six rows. Four sets of data are displayed per screen. The user presses the "enter" key to scroll forward to each succeeding screen display.
h. Training Foster Format Routine (Trngrpt.prg)

This defines a coded report format which displays COC, HIS, HKS, COD, INT, TAC, NBC, PFT1CLSS, PFT2CLSS, UNI and ALT for each member of the unit or subunit selected.

i. Marksmanship Status Routine (Mksstat.prg)

This routine creates a temporary file which contains the current year rifle and pistol qualification results for each member in the selected unit or subunit. The results are summed by qualification category for the rifle and pistol. Percentages are calculated and displayed in a summary status report which is automatically displayed on the screen. From this module a simple decision support submodule may be called at the discretion of the user. If the user desires to analyze his remaining rifle range quotas and compare them with his total number of range requirements for the current year, he may elect to "Analyze Rifle Range Quotas" (See Quota Study Routine for a more complete description.). Figure 4.6 displays the Marksmanship Status Report.

j. Convert Unit Routine (Cvrtunit.prg)

This routine stores the name of the company and platoon selected (unit and subunit) by the user during routine 'S1ctunit'. This is necessary because when the user selects a unit and subunit, it is done by menu selection. Accordingly the only thing stored into memory is the corresponding letter from the menu provided by S1ctunit.prg (A, E, C, L, E). Some screen displays require the unit name (e.g., "A CC" vice "A") so a conversion is necessary.
MARKSMANSHIP STATUS REPORT

RIFLE EXPERT... 48  PISTOL EXPERT... 3
RIFLE SHARPSHOOTER... 36  PISTOL SHARPSHOOTER... 0
RIFLE MARKSMEN... 48  PISTOL MARKSMEN... 2
UNQUALIFIED... 16  UNQUALIFIED... 0

PERCENTAGES

RIFLE EXPERT... 32  PISTOL EXPERT... 60
RIFLE SHARPSHOOTER... 24  PISTOL SHARPSHOOTER... 0
RIFLE MARKSMEN... 32  PISTOL MARKSMEN... 40
UNQUALIFIED... 11  UNQUALIFIED... 0

PRESS 'ENTER' TO CONTINUE

Figure 4.6  Marksmanship Status Report

k. Quota Study Routine (Qtastudy.prg)

The Qtastudy.prg and the Quotapad.prg make up the decision support submodule. This module is used by Mksstat.prg and displays the range quotas that remain for the current year. The current month is extracted from the log on date. The range quotas for each unit are stored in a memory file during the annual initialization procedure. The appropriate set of range quotas for the unit involved is retrieved and stored into memory variables. Summary information is provided concerning the relationship between remaining range quota requirements and remaining range quota allocations. Upon conclusions of the module the user may return to the Standard Report Menu or to a scratch pad to revise quota assignments. Revised quota assignments from the scratch pad can either be permanently saved in the range quota memory file or ignored. See Appendix B for a more detailed description.
1. Quota Scratch Pad Routine (Quotepad.prg)

This module gives the user an opportunity to interactively modify the remaining monthly rifle range quotas for his unit. This can be done an unlimited number of times. Each time the screen display returns to the actual monthly allocation as base data. After each iteration, the new set of range quotas is totaled and compared to the number of rifle range quotas required by the unit for the remainder of the year. A comment is displayed which indicates the net result (scratch pad quotas less than requirements, scratch pad quotas greater than requirements, or scratch pad quotas equal to requirements). However, if the user elects to permanently replace the monthly quota allocation in memory, he may do so by selecting the "Replace Current Quotas" option.

2. Marksmanship Roster Routine (Mksrscst.prg)

This routine creates a temporary file which includes NAME, RPLQUAL, RFLDATE, PSTQUAL, PSTDATE for each member in the selected unit or subunit. A standard report format (B:Mksrscst) is used to display the Marksmanship Foster Report (see Figure 4.7).

3. MOS Status Routine (Mosstat.prg)

This routine produces the MOS Status Report. Each type of MOS in the specified unit or subunit is counted and totaled, regardless of the type or number of different MOS's in the particular unit or subunit. The results are then displayed by MOS and by rank (see Figure 4.8).

4. MOS Roster Routine (Mosrscst.prg)

This is a control routine which creates the Personnel Roster for the unit or subunit specified by the user's response to the Slectunit Menu (see Figure 4.9).
MARKSMANSHIP ROSTER REPORT

<table>
<thead>
<tr>
<th>NAME</th>
<th>RANK</th>
<th>QUAL</th>
<th>DATE</th>
<th>QUAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>EAKER, SIEVE N.</td>
<td>E4</td>
<td>100184</td>
<td></td>
<td>EX</td>
</tr>
<tr>
<td>CARY, R.V.</td>
<td>F5</td>
<td>050184</td>
<td></td>
<td>MM</td>
</tr>
<tr>
<td>CASTIN, D.P.</td>
<td>F1</td>
<td>050184</td>
<td></td>
<td>UN</td>
</tr>
<tr>
<td>LICNELY, R.W.</td>
<td>E6</td>
<td>DDMYY</td>
<td></td>
<td>**</td>
</tr>
</tbody>
</table>

PRESS 'ENTER' TO CONTINUE

Figure 4.7 Marksmanship Roster Report

MOS STATES REPORT A CO, 3RD PLAT

<table>
<thead>
<tr>
<th>MCS</th>
<th>RANK</th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>0311</td>
<td>E1</td>
<td>8</td>
</tr>
<tr>
<td>0311</td>
<td>E2</td>
<td>12</td>
</tr>
<tr>
<td>0311</td>
<td>E3</td>
<td>9</td>
</tr>
<tr>
<td>0311</td>
<td>E4</td>
<td>8</td>
</tr>
<tr>
<td>0311</td>
<td>E5</td>
<td>3</td>
</tr>
</tbody>
</table>

Figure 4.8 MOS Status Report

F. Personnel Data Routine (Persdata.prg)

This is a control routine which creates the Personnel Roster for the unit or subunit specified by the user's response to the Select Unit Menu.

G. Personnel Data Format Routine (Mosrpt.frm)

This defines a dBASE II report format which has already been determined and is stored in memory.
Figure 4.9  MOS Roster Report

1. EST Status Routine (Eststat.prg)

This routine creates a temporary file that includes all essential subject and PFT fields from the Est.dtf for members in a user specified unit or subunit. From this temporary file the routine counts the total number of unit members that have completed all elements in each training category. The EST Status Report is displayed and includes totals for each training category and the percent completed (see Figure 4.10).

2. EST Count Routine (Cntest.prg)

This is a short routine that counts the total training categories containing an "*" which indicates a training element has not been completed (i.e., It counts the number of training categories that currently include one or more training elements that have not been completed).

3. EST Roster Routine (Estrost.prg)

This is a control routine which creates the Personnel Roster Report for the unit or subunit specified by the user's response to the Select Unit Menu.
EST STATUS REPORT

CCDE CP CONDUCT... 42 PERCENT COMPLETED... 76
MARINE CORPS HISTORY... 84 PERCENT COMPLETED... 151
CLOSE ORDER DRILL... 39 PERCENT COMPLETED... 151
INTERIOR GUARD... 63 PERCENT COMPLETED... 113
FIRST AID... 8 PERCENT COMPLETED... 14

PRESS 'ENTER' TO CONTINUE

Figure 4.10 EST Status Report

u. EST Report Format Routine (Estrpt.prg)

This routine formats and displays the EST Roster Report. It is called by Estrcst.prg.

v. Swim Qualification Routine (swimqual.prg)

This routine displays the Swim Qual Report for the selected unit or subunit. The Swim Qual Report consists of each Marine's NAME, RANK, UNIT, and swimming qualification status. At the end of the report, summary data is displayed which includes the totals for each qualification category and the overall total for the unit or subunit. Figure 4.11 is a copy of the Swimming Qualification Status Report.

w. Swim Report Routine (swimrpt.prg)

This routine, called by Swimqual.prg counts and stores the number of personnel in the selected unit or subunit that are in each swimming qualification category. Also, this routine displays the NAME, RANK, COMPANY and swimming qualification for each member of the unit or subunit.
### Figure 4.11 Swimmer Qualification Status Report

<table>
<thead>
<tr>
<th>BAKER, STEVE N.</th>
<th>E4</th>
<th>A CO</th>
</tr>
</thead>
<tbody>
<tr>
<td>SWIM QUAL S3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CALEY, R. V.</td>
<td>E5</td>
<td>A CO</td>
</tr>
<tr>
<td>SWIM QUAL S2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PHILLIP, D. P.</td>
<td>E4</td>
<td>A CO</td>
</tr>
<tr>
<td>SWIM QUAL WS</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### SUMMARY DATA

<table>
<thead>
<tr>
<th>SWIM QUALIFICATION</th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>WS</td>
<td>3</td>
</tr>
<tr>
<td>S1</td>
<td>54</td>
</tr>
<tr>
<td>S2</td>
<td>49</td>
</tr>
<tr>
<td>UNQUALIFIED</td>
<td>36</td>
</tr>
</tbody>
</table>

**Press 'ENTER' to continue**

**x. Help Routine (Help.prg)**

This routine provides a functional description of each report listed in the Standard Report Selection menu.

**6. Maintenance Module**

**a. Maintain Control Routine (Maintain.prg)**

The Maintain Control Routine produces a menu, Figure 4.12, that allows the user to select the desired function. This module restricts access when the function has a critical capability. Only the Data Base Administrator (DBA) and the Assistant DBA have access to all functions.

**b. View Control Routine (Viewctr.prg)**

This module queries the user for the name of the individual whose training record is to be viewed. The
**MAINTENANCE MODULE MENU**

**SELECT ONE OF THE FOLLOWING OPTIONS**

- V...VIEW INDIVIDUAL DATA
- U...UPDATE DATA
- C...CREATE AN ITR
- D...DELETE AN ITR
- S...SYSTEM FUNCTIONS
- H...HELP
- *Q...QUIT*

**SELECT OPTION ===>:**

---

**Figure 4.12 Maintenance Module Menu**

The routine controls the search and creation of the formatted screen that displays the ITR. The routine asks the user a series of questions that can be answered with yes or no. This routine allows only the data to be displayed (see Figure 4.13).
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAME: LAST, FIRST MF.</td>
<td></td>
</tr>
<tr>
<td>BIRTHDATE: ....:DDMMYY:</td>
<td></td>
</tr>
<tr>
<td>CO:</td>
<td></td>
</tr>
<tr>
<td>MIL CODE OF CONDUCT: ......</td>
<td></td>
</tr>
<tr>
<td>HISTORY OF MARCO: ........</td>
<td></td>
</tr>
<tr>
<td>DRILL:</td>
<td></td>
</tr>
<tr>
<td>INTERIOR GUARD: ........</td>
<td></td>
</tr>
<tr>
<td>FIRST AID:</td>
<td></td>
</tr>
<tr>
<td>SWIM QUAL: ....: DDMMYY:</td>
<td></td>
</tr>
<tr>
<td>RIFLE QUAL: ....: DDMMYY:</td>
<td></td>
</tr>
<tr>
<td>FISTAL QUAL: ....: DDMMYY:</td>
<td></td>
</tr>
<tr>
<td>ELECTRONIC WARFARE: ......</td>
<td></td>
</tr>
<tr>
<td>OLD WEATHER:</td>
<td></td>
</tr>
<tr>
<td>LAW OF WAR:</td>
<td></td>
</tr>
<tr>
<td>TRAINING HAND:</td>
<td></td>
</tr>
<tr>
<td>LEAD RUSH:</td>
<td></td>
</tr>
<tr>
<td>CRIME &amp; MORAL FP. ........</td>
<td></td>
</tr>
<tr>
<td>CONTENTS:</td>
<td></td>
</tr>
</tbody>
</table>

**Figure 4.13** ITR Screen Format
c. ITR Screen Format Routine (Itrscrn.prg)

This routine produces the electronic ITR on which an individual's data is displayed.

d. Screen Data Retrieval Routine (Getdata.prg)

The Getdata.prg communicates with the DBMS and retrieves the data for a particular individual. It then sends the data to the appropriate position on the screen that corresponds to the ITR format. The DBMS locates individuals by their SSN. Four data base files are used to create an ITR; Fers.dbf, Est.dbf, Qual.dbf, and Infomisc.dbf.

e. Comment Retrieval Routine (Getcmmnt.prg)

If unstructured comments exist for an individual, the routine will instruct the DBMS to search for them. The comment information will then be displayed on the screen.

f. Update Control Routine (Updtitr.prg)

The Update Control Routine questions the user concerning which ITR is to be updated. The routine determines if the individual is in the data base and controls the update of the ITR. The routine allows the user to update as many ITR's as desired. The data is displayed on the screen in the ITR format and allows a modified form of full screen editing.

g. Data Update Routine (Update.prg)

This routine requires the user to type in the name of the field to be updated. The routine then highlights that field and permits data to be entered. When the data has been entered this routine initiates error checking.
and initiates calculations. The data is then stored in the database.

b. ITRM Creation Routine (Creatitz.prg)

The creation of a new ITR for an individual is controlled by this routine. It initiates the formatting of the screen and controls the routine which gathers the data for the individual. The routine allows multiple records to be created.

i. ITR Data Initialization Routine (Indata.prg)

This routine initializes the database and allows the user to enter data for an individual by field. Once the data has been entered the module initializes error checking and calculations that may pertain to the data fields. If unstructured comments are to be entered, the routine controls this process. The ITR Data Initialization Routine instructs the DBMS to store the data in the appropriate data base files.

j. Data Error Checking Routine (Error.prg)

This routine performs the error checking for the system during data entry. Since this is a prototype system, error checking is not extensive. Dates, value ranges, and acceptable characters are the limits of the error checking. Dates are only checked for limits in number of months, days in the month, and acceptable year window. Specific months are not matched to specific number of days. The system does catch the most common errors caused by gross typing mistakes but may not catch slight errors if the values are within specified limits.
k. Calculation Routine (Calculat.prg)

The Calculation Routine performs the calculations necessary to reproduce training tables concerning weight/height standards, physical fitness standards by age, rifle qualification levels, and pistol qualification levels. When the information that is needed for these calculations has been entered in the data base the system recalculates the related fields and enters the new qualifications in the data base.

l. Comment Update Routine (Wrtcommnt.prg)

This routine finds the individual's comment record in the data base and updates the information with new data. It then stores the data back into the data base.

m. Comment Creation Routine (Crtcommnt.prg)

The Comment Creation Routine creates an unformatted comment record for an individual in the data base. The comment record is filled with asterisks that will be typed over when data is inserted.

n. Delete ITR Routine (Deletitr.prg)

This routine is limited to the DBA and Assistant DBA only. The routine queries the user for the name of the individual to be deleted. It then finds the individuals ITRM information and displays the data in ITR format. A second verification, that this individual is to be deleted, is obtained at this time. The module then deletes the individual from all the data base files. This deletion is permanent and there is no recovery of deleted records.
c. System Function Control Routine (System.prg)

The control of system functions is accomplished by this routine. Like all previous routines, this routine is menu driven (see Figure 4.14). The user selects the menu option that performs the desired function, and calls the appropriate routine.

```
SYSTEM FUNCTION MENU
SELECT ONE OF THE FOLLOWING OPTIONS
I...INSTALL SYSTEM
S...SYSTEM RESET (YEARLY)
A...ACCESS LIST MAINTENANCE
Q...QUIT TO MAINTENANCE MENU
ENTER OPTION ===>:
```

Figure 4.14 System Function Menu

p. System Installation Routine (Install.prg)

Units in the Marine Corps often have different names. The system allows the DBA to name his units to correspond to the actual names of the battalion's units. The system also initializes the current year. This information is then stored on a file which is recalled whenever the system is functioning.

q. System Reset Routine (Sysreset.prg)

Training in the Marine Corps is based on a yearly cycle and requires that individuals requalify in certain areas once or more a year. This routine resets globally all the fields in the data base that must be
requalified each year. The routine allows the DBA to enter the yearly range quotas. The DBA can use this routine to update those quotas, but this is not recommended. The capability also exists in the Standard Report Generators DSS.

r. System Access Control Routine (Rsstac.css.prg)

The DBA must be able to change the access list by deletions or additions. This routine displays a menu, (Figure 4.15) that gives the DBA options to choose. This routine gives the DBA the ability to manage access to the system and maintain a list of users. It is not recommended that the DBA keep a printed copy of this list unless it is in a secured environment. The system will list each user, user identification code, user password, and authorization level.

```
SECURITY ACCESS LIST
CHOOSE OPTION TO BE EXECUTED
L...LIST ALL USERS
A...ADD TO ACCESS LIST
D...DELETE FROM ACCESS LIST
Q...QUIT TO SYSTEM FUNCTION MENU

ENTER OPTION ===>:
```

Figure 4.15 Access Function Menu
V. CONCLUSION

A. GENERAL

The thesis concept statement is written in general terms and therefore, is subject to qualitative evaluation only. This approach is intended for the following reason. During the conception phase of the project, we had only a vague intuitive "feel" for such things as reasonable system costs, acceptable response times, operator training requirements, and the ultimate "user friendliness" that could be achieved. Because functional requirements are not well defined at this point, an attempt to impose stringent quantitative standards is premature. Furthermore, quantitative goals shift the emphasis toward achieving system efficiency and away from defining system effectiveness. In other words, the purpose of this thesis is to prove a general concept through the development of a working prototype. The general concept includes simplicity of operation and maintenance. System efficiency is not a significant consideration. The reader will notice, however, that system specifications are defined using quantifiable objectives where feasible, the distinction being that the system requirements determined by the user may not be satisfied by the first generation prototype. As the user becomes more familiar with the capabilities of the system, requirements will become more clearly defined. With this approach the overall system concept may be valid while some system requirements have not yet been attained (i.e., in the first generation prototype).
E. SYSTEM SPECIFICATIONS (FUNCTIONAL)

All system specifications, except those referred to below, have been satisfied by the prototype.

1. Storage Limitations

The prototype cannot adequately handle the specified 800 records. The limiting constraint is available storage. All system software resides on four double-sided, double-density 5 1/4 inch floppy disks (360k bytes each). The first two disks are program disks which are stored directly into RAM memory. The second two disks contain all of the data base files. Data base files are read into working storage only when they are being used by the system. Each Marine in the data base requires approximately 550 bytes of data (or 4.4k bits). Due to the DBMS's required overhead the prototype can accommodate approximately 180 Marines. In order to achieve the specification requirement of 800 Marines, a storage device capacity of two megabytes or more is required (e.g., hard disk drive).

2. Retrieval Limitations

All standard reports can be retrieved by unit or subunit. However, the standard report does not include the facility to retrieve data by other user defined attributes. This facility is, however, included in the nonstandard report. The nonstandard report may include up to three user defined attributes and attribute descriptions.

C. SYSTEM SPECIFICATIONS (NON-FUNCTIONAL)

1. Training

Most of the non-functional specifications relate to the operator training effort required to implement the
system. It is our judgement that the prototype satisfies these requirements.

2. **Error Checking**

Operator input is range checked, data type checked, and checked for field length (characters in excess of the maximum field length are truncated). In a fully implemented system error checking would be expanded and diagnostic error messages sent to the user.

3. **Table Mapping**

Mapping of rifle, pistol and PFT scores into qualification categories is invisible to the user. In addition, a weight control table is embedded. When an individual's height and weight are entered the weight control table is referenced. If his weight is in excess of allowable standards for his height, he is automatically assigned to the "weight control program" by assigning a boolean variable of "T" to the weight control field in his training record.

**D. SUBSEQUENT SYSTEM DEVELOPMENT**

The next step in the development of the system is to place the system into the hands of the user to accomplish the following objectives:

a. Define user training requirements;

b. Achieve user familiarity;

c. Evaluate and quantify existing functional requirements;

d. Define additional quantifiable functional requirements;
e. Design and develop second generation prototype based on new set of user defined functional requirements.

2. SUMMARY

The objective of the Thesis was to prove the feasibility of a concept. It is reasonable to conclude that the concept has been proven through the design, the implementation, and the demonstration of a working prototype.
APPENDIX A
DATA ELEMENT DICTIONARY

All data elements that exist in the individual training data management system are contained in one of seven data base files. The only exception is the data element SSN. SSN is a key field and is therefore included in the record structure of multiple files. The following information is used to describe each data base file:

A. FIELD

Each data element belongs to a specific field within the record structure. Fields are numbered sequentially. For example, field 03 in the data base file called PARS is the data element named Rank.

B. NAME

Each data field has a unique name which always refers to that specific information that is contained in the data field (example: PRISMEMOS contains the primary MOS of each member in the data base).

C. TYPE

Each data field is defined to be one of three possible data types:
1. Character (C)
2. Numeric (N)
3. Logical (L)
D. Width

This characteristic describes the maximum length of the data field. Information in the data field may not exceed the number of spaces that have been set aside for that particular field (all characters and blanks are included when counting the number of spaces in a field).

E. Description

This column contains a narrative description of the field. In many instances, examples (Ex.) are used to describe the structure of a typical data element.

F. Data Base Files

1. Files (Personnel)

<table>
<thead>
<tr>
<th>Field</th>
<th>Name</th>
<th>Type</th>
<th>Width</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>SSN</td>
<td>C</td>
<td>09</td>
<td>Ex. 240707765; Last name followed by a comma, first name and middle initial. (Ex. Smith, Jack E.)</td>
</tr>
<tr>
<td>02</td>
<td>Name</td>
<td>C</td>
<td>30</td>
<td>Pay grade (Ex. E2, 01)</td>
</tr>
<tr>
<td>03</td>
<td>Rank</td>
<td>C</td>
<td>02</td>
<td>Primary Military Occupational Specialty (Ex. 0311)</td>
</tr>
<tr>
<td>04</td>
<td>PrimMOS</td>
<td>C</td>
<td>04</td>
<td>Secondary MOS</td>
</tr>
<tr>
<td>05</td>
<td>SecMOS</td>
<td>C</td>
<td>04</td>
<td>Ex. A CO, HHC CO, A BTRY</td>
</tr>
<tr>
<td>06</td>
<td>Company</td>
<td>C</td>
<td>06</td>
<td>Ex. 1ST PLAT, GUN PLAT, WPSN PLAT, HDQTRS</td>
</tr>
<tr>
<td>07</td>
<td>Faction</td>
<td>C</td>
<td>09</td>
<td>The date (DDMMYY) member joined the unit.</td>
</tr>
<tr>
<td>08</td>
<td>Joindate</td>
<td>C</td>
<td>06</td>
<td>Expiration of Active Service (DDMMYY)</td>
</tr>
<tr>
<td>09</td>
<td>SAS</td>
<td>C</td>
<td>06</td>
<td>Date of birth (DDMMYY)</td>
</tr>
<tr>
<td>10</td>
<td>Eftldate</td>
<td>C</td>
<td>06</td>
<td>A boolean variable (F/T) for which T indicates that a comment record exists.</td>
</tr>
<tr>
<td>11</td>
<td>Comment</td>
<td>L</td>
<td>01</td>
<td>Individual's height in inches</td>
</tr>
<tr>
<td>12</td>
<td>Height</td>
<td>C</td>
<td>02</td>
<td>Individual's weight in pounds</td>
</tr>
<tr>
<td>13</td>
<td>Weight</td>
<td>C</td>
<td>03</td>
<td>A boolean variable (F/T) for which T indicates that the individual is on weight control.</td>
</tr>
<tr>
<td>14</td>
<td>Bwmtct</td>
<td>L</td>
<td>01</td>
<td>Gas mask size (S, M, L)</td>
</tr>
<tr>
<td>15</td>
<td>Gasmask</td>
<td>C</td>
<td>01</td>
<td></td>
</tr>
</tbody>
</table>

69
### 2. EST (Essential Subjects Testing)

<table>
<thead>
<tr>
<th>Field</th>
<th>Name</th>
<th>Type Width</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>SSN</td>
<td>C 09</td>
<td>Code of Conduct contains an entry for each training element (A,B,C).</td>
</tr>
<tr>
<td>02</td>
<td>COC</td>
<td>C 03</td>
<td>History of the Marine Corps contains an entry for each training element (A,B,C).</td>
</tr>
<tr>
<td>03</td>
<td>HIS</td>
<td>C 03</td>
<td>Close Order Drill contains an entry for each training element (A,B,C).</td>
</tr>
<tr>
<td>04</td>
<td>COD</td>
<td>C 03</td>
<td>Interior Guard contains an entry for each training element (A,B,C).</td>
</tr>
<tr>
<td>05</td>
<td>INT</td>
<td>C 03</td>
<td>First Aid contains an entry for each training element (A,B,C).</td>
</tr>
<tr>
<td>06</td>
<td>AIL</td>
<td>C 11</td>
<td>Equipment &amp; Uniforms contains an entry for each training element (A,B,C).</td>
</tr>
<tr>
<td>07</td>
<td>UNI</td>
<td>C 03</td>
<td>Includes the numeric score for the first annual physical fitness test (RANGE: 00-300).</td>
</tr>
<tr>
<td>08</td>
<td>FPT1RAW</td>
<td>N 03</td>
<td>Date of first physical fitness test (DDMMYY).</td>
</tr>
<tr>
<td>09</td>
<td>FPT1DATE</td>
<td>C 06</td>
<td>Results of first physical fitness test (1,2,3,4).</td>
</tr>
<tr>
<td>10</td>
<td>FPT1CLSS</td>
<td>C 01</td>
<td>Includes the numeric score for the second annual physical fitness test (RANGE: 00-300).</td>
</tr>
<tr>
<td>11</td>
<td>FPT2RAW</td>
<td>N 03</td>
<td>Date of the second physical fitness test.</td>
</tr>
<tr>
<td>12</td>
<td>FPT2DATE</td>
<td>C 06</td>
<td>Results of second physical fitness test (1,2,3,4).</td>
</tr>
<tr>
<td>13</td>
<td>FPT2CLSS</td>
<td>C 01</td>
<td>NBC contains an entry for each training element (A,B,C,D,E,F).</td>
</tr>
<tr>
<td>14</td>
<td>NBC</td>
<td>C 06</td>
<td>Marksmanrship contains an entry for each training element (A,B,C,D,E,F).</td>
</tr>
<tr>
<td>15</td>
<td>MKS</td>
<td>C 06</td>
<td>Individual Tactical Measures contains an entry for each training element (A,B,C,D,E,F).</td>
</tr>
<tr>
<td>16</td>
<td>TAC</td>
<td>C 05</td>
<td>Description</td>
</tr>
</tbody>
</table>

### 3. Qualifications

<table>
<thead>
<tr>
<th>Field</th>
<th>Name</th>
<th>Type Width</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>SSN</td>
<td>C 09</td>
<td>EX. 240707765</td>
</tr>
<tr>
<td>02</td>
<td>SWIMQUAL</td>
<td>C 02</td>
<td>Swimming Qualification (WQ, 61.52-63)</td>
</tr>
<tr>
<td>03</td>
<td>SWIMDATE</td>
<td>C 06</td>
<td>SWIMQUAL date (DDMMYY).</td>
</tr>
<tr>
<td>04</td>
<td>FPLSCORE</td>
<td>C 03</td>
<td>Numeric rifle score (RANGE: 000-250).</td>
</tr>
<tr>
<td>05</td>
<td>FFLQUAL</td>
<td>C 02</td>
<td>Rifle Qualification (EX,SS, MM,UN).</td>
</tr>
<tr>
<td>06</td>
<td>RFILDATE</td>
<td>C 06</td>
<td>Date of RFILQUAL (DDMMYY).</td>
</tr>
<tr>
<td>07</td>
<td>FPSCORE</td>
<td>C 03</td>
<td>Numeric pistol score (Range: 000-300).</td>
</tr>
<tr>
<td>08</td>
<td>PSTQUAL</td>
<td>C 02</td>
<td>Pistol Qualification (EX,SS, MM,UN).</td>
</tr>
<tr>
<td>09</td>
<td>PSTDATE</td>
<td>C 06</td>
<td>Date of PSTQUAL (DDMMYY).</td>
</tr>
</tbody>
</table>
4. **IFCMISC (Information Miscellaneous)**

<table>
<thead>
<tr>
<th>Field</th>
<th>Name</th>
<th>Type</th>
<th>Width</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>SSN</td>
<td>C</td>
<td>09</td>
<td>EX. 240707765</td>
</tr>
<tr>
<td>02</td>
<td>PLECWAR</td>
<td>C</td>
<td>06</td>
<td>Electronic Warfare contains an entry for each training element (A, B, C, D, E, F).</td>
</tr>
<tr>
<td>03</td>
<td>CLEWTHR</td>
<td>C</td>
<td>06</td>
<td>Cold Weather Training contains an entry for each training element (A, B, C, D, E, F).</td>
</tr>
<tr>
<td>04</td>
<td>LAWWAR</td>
<td>C</td>
<td>06</td>
<td>Law of War contains an entry for each training element (A, B, C, D, E, F).</td>
</tr>
<tr>
<td>05</td>
<td>MOS</td>
<td>C</td>
<td>04</td>
<td>MOS Training contains an entry for each training element (A, B, C, D, E, F).</td>
</tr>
<tr>
<td>06</td>
<td>LDRSHP</td>
<td>C</td>
<td>06</td>
<td>Leadership Training contains an entry for each training element (A, B, C, D, E, F).</td>
</tr>
<tr>
<td>07</td>
<td>DRUG</td>
<td>C</td>
<td>06</td>
<td>Drug Abuse contains an entry for each training element (A, B, C, D, E, F).</td>
</tr>
<tr>
<td>08</td>
<td>ALCOHOL</td>
<td>C</td>
<td>06</td>
<td>Alcohol Abuse contains an entry for each training element (A, B, C, D, E, F).</td>
</tr>
<tr>
<td>09</td>
<td>HUMREL</td>
<td>C</td>
<td>06</td>
<td>Human Relations contains an entry for each training element (A, B, C, D, E, F).</td>
</tr>
<tr>
<td>10</td>
<td>FERSAPPR</td>
<td>C</td>
<td>06</td>
<td>Personal Affairs contains an entry for each training element (A, B, C, D, E, F).</td>
</tr>
<tr>
<td>11</td>
<td>UCMJ</td>
<td>C</td>
<td>06</td>
<td>UCMJ contains an entry for each training element (A, B, C, D, E, F).</td>
</tr>
<tr>
<td>12</td>
<td>CHRMORED</td>
<td>C</td>
<td>06</td>
<td>Character and Moral Education contains an entry for each training element (A, B, C, D, E, F).</td>
</tr>
</tbody>
</table>

5. **SECURITY (Access Information)**

<table>
<thead>
<tr>
<th>Field</th>
<th>Name</th>
<th>Type</th>
<th>Width</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>USERID</td>
<td>C</td>
<td>05</td>
<td>Numeric code that uniquely identifies each user.</td>
</tr>
<tr>
<td>02</td>
<td>USEBPASS</td>
<td>C</td>
<td>08</td>
<td>Alphanumeric code that uniquely identifies each user.</td>
</tr>
<tr>
<td>03</td>
<td>AUTOELEV</td>
<td>N</td>
<td>01</td>
<td>Numeric code that defines authorization level for each user.</td>
</tr>
<tr>
<td>04</td>
<td>NAME</td>
<td>C</td>
<td>15</td>
<td>Alphanumeric that identifies the user.</td>
</tr>
</tbody>
</table>

6. **COMMENTS (Non-formatted Information)**

<table>
<thead>
<tr>
<th>Field</th>
<th>Name</th>
<th>Type</th>
<th>Width</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>SSN</td>
<td>C</td>
<td>09</td>
<td>EX. 240707765</td>
</tr>
<tr>
<td>02</td>
<td>INFCTXT1</td>
<td>C</td>
<td>25</td>
<td>Unstructured comments.</td>
</tr>
<tr>
<td>03</td>
<td>INFCTXT2</td>
<td>C</td>
<td>25</td>
<td>Unstructured comments.</td>
</tr>
<tr>
<td>04</td>
<td>INFCTXT3</td>
<td>C</td>
<td>25</td>
<td>Unstructured comments.</td>
</tr>
<tr>
<td>05</td>
<td>INFCTXT4</td>
<td>C</td>
<td>25</td>
<td>Unstructured comments.</td>
</tr>
<tr>
<td>06</td>
<td>INFCTXT5</td>
<td>C</td>
<td>25</td>
<td>Unstructured comments.</td>
</tr>
<tr>
<td>07</td>
<td>INFCTXT6</td>
<td>C</td>
<td>25</td>
<td>Unstructured comments.</td>
</tr>
</tbody>
</table>

71
<table>
<thead>
<tr>
<th>No.</th>
<th>INFCTXT7</th>
<th>C</th>
<th>25</th>
<th>Unstructured comments.</th>
</tr>
</thead>
<tbody>
<tr>
<td>08</td>
<td>INFCTXT8</td>
<td>C</td>
<td>25</td>
<td>Unstructured comments.</td>
</tr>
<tr>
<td>09</td>
<td>INFCTXT9</td>
<td>C</td>
<td>25</td>
<td>Unstructured comments.</td>
</tr>
<tr>
<td>10</td>
<td>INFCTXTO</td>
<td>C</td>
<td>25</td>
<td>Unstructured comments.</td>
</tr>
</tbody>
</table>
APPENDIX B
DECISION SUPPORT SYSTEM (DSS) DESCRIPTION

A. BACKGROUND

The management of training records includes many routine tasks that are repetitious and highly structured. Many of these tasks incorporate low level, well-defined or semi-structured, independent decision making processes. Heretofore, these tasks have been manually implemented at the cost of many man-hours, and sometimes decisions have been made without making the best use of available information. Too often the result has been wasted effort, frustration, and incorrect decisions.

E. PURPOSE

The purpose of this submodule is to demonstrate the application of a "first generation" prototype DSS that will achieve the following:

a. Demonstrate typical capabilities of DSS in the training management environment.

b. Demonstrate the small knowledge base necessary to implement a properly designed DSS.

c. Serve as a "functioning prototype" to gain early user support and experience in order to further refine system requirements.

C. FUNCTIONAL DESCRIPTION

The DSS element of the software system product is implemented in the "Marksmanship Status Report" routine which is
called by the "Standard Report Generator". When the user selects the "Marksmanship Status Report" from the "Standard Report Menu" the DSS submodule is implemented. The DSS submodule performs the following functions at the company level:

a. Calculates and displays a current "Marksmanship Status Report" which includes:

(1) A summary of rifle and pistol current year requalification data (number of experts, sharpshooters, marksmen, and unqualified).

(2) The total number of persons that have fired for qualification during the current year.

(3) The percent of experts, sharpshooters, marksmen and unqualified shooters based on the total number of people that have fired for qualification during the current year.

b. A company rifle range quota analysis may then be performed at the user's discretion. This routine includes the following functions:

(1) Calculates and displays a range quota analysis.

(2) The following summary information is displayed:

(a) Total number of Marines fired for qualification during current year.

(b) Total number of Marines fired for qualification during current year, but failed to qualify.

(c) Total number of Marines not required to fire for qualification during the current year.

(d) Total number of Marines to qualify before the end of the current year.

(3) A chart is displayed which shows the months remaining in the current year, the number of range
quotas allocated for each month, and the total number of rifle range quotas that remain for the current year.

(4) As appropriate, one of the following elements of information is displayed:

(a) Additional quota requirements for the year

(b) Excess quotas allocated for the year

(c) Quota requirements (number of person left to qualify) equal quota allocations.

c. User's may manipulate allocations on a scratch pad to visualize different combinations of monthly quota allocations or to determine their effect on end of year results (the objective is to manage quotas so that quota requirements equal quotas allocated). If desired, a set of "scratch pad" range quotas can permanently replace the unit's range quota allocations for the remainder of the current year. If this option is elected all subsequent calculations will be based on the "new set of range quota allocations".

D. USER'S VIEW

1. SUMMARY INFORMATION

Decision making is supported by providing summary information for the company. This snapshot of a unit's current marksmanship status demonstrates the degree of success of the unit's marksmanship training program. By comparing this summary to unit marksmanship objectives the system can be used as a control mechanism. The following types of decisions can be supported:

(1) Evaluate marksmanship training program for weaknesses as indicated by unsatisfactory qualification results to date.
(2) Evaluate unit qualification objectives for feasibility.

2. Support Depth

Decision making is supported at many levels from the platoon or company training NCO to the battalion operations officer.

3. Interdependence

Interdependent (sequential and pooled) decisions are supported. For example, if one company permanently changes rifle range quota allocations this decision is captured by the system and reflected in any subsequent analysis.

4. Restrictions

Due to the overriding system requirement for limited user training all processes are menu driven. This restricts the degree of user control and flexibility. This is a significant compromise which reduces its effectiveness as a ISS.

I. BUILDER'S VIEW

1. Dialogue

The dialogue subsystem (implemented in dBase II) creates the user's view as discussed above. It performs all necessary database functions, extracts and processes summary information.

2. Modelling

The model base subsystem includes the mathematical and process models necessary to use the "scratch pad" utilities. For example, a model file is constructed in memory for each unit which has allocated quotas. Quotas can be
selectively extracted, compared, permanently changed by the user (provided the user has the appropriate system access authorization). Models are not integrated, but each model is fully integrated with the DBMS through the query language. Integration of models is a desirable feature (e.g., battalion should have the capability to total the quotas of all companies on a monthly basis or on a remaining current year basis) and could be implemented. This is the type of added requirement that should be defined by the user during operation of this "first generation" prototype.

F. CONTENTS OF KNOWLEDGE BASE

Data Base files (DBF) contain qualification status of each member for the current year and members not required to fire for qualification during the current year.

Memory files (MEM) containing rifle range quotas for each company are stored in separate memory files (see Figure B.1 for example).

| CORNGQTA.MEM |
|---------------|---------------|
| VARIABLE NAME | VALUE         |
| OJAN          | 25            |
| OPEE          | 25            |
| OMAP          | 50            |
| OMAP           | 50           |
| OJUL          | 50            |
| OJUL          | 50            |
| OUGE          | 50            |
| OUGE          | 50            |
| OSEP          | 50            |
| OSEP          | 50            |
| ODEC          | 00            |
| ODEC          | 00            |

Figure B.1 Memory File
**APPENDIX C**

**ITRS MASTER CONTROL MODULE LISTING**

### A. MASTER ROUTINE

- **Routine Name:** Master.prg
- **Module Name:** ITRS Master Control Module
- **Version:** 1.0
- **Author:** D.P. Haeusler
- **Date:** 28 Oct 83
- **Variables Used:** Thru, Valpass
- **Variables Modified:** None
- **Variables Created:** None
- **Variables Released:** None
- **Temp Files Created:** None
- **Using Subroutines:** None

**Description:** This is the routine that controls the flow of the program. This routine is an endless loop which requires the user to turn the computer off in order to enter another system. This routine does not allow the users to directly enter the DBMS, the general intent is to make the DBMS completely transparent to users. Some house keeping is done to insure that the proper disks are in place. This occurs by reading stored files on the disks and then checking the values. The system will not continue until the disks are in the proper drive.

```plaintext
SET CCNSCLE ON
SET TALK OFF
SET ESCAPE OFF

STORE T TO AABENDLES
STORE P TO SWITCH
DO WHILE .NOT. SWITCH

ERASE
@ 9,25 SAY " REMOVE DISKETTES FROM THE DRIVES"
@ 10,25 SAY " PLACE 'A' DATA DISK IN DRIVE 'A':"
@ 11,25 SAY " PLACE 'B' DATA DISK IN DRIVE 'B':"
@ 23,25 SAY "***** PRESS ANY KEY TO CONTINUE *****"
SET CCNSCLE OFF
WAIT
SET CCNSCLE ON
RESTORE FROM A:MEMDISK ADDITIVE
ENDDO

DO WHILE AABENDLES

ERASE
@ 10,27 SAY " UNITED STATES MARINE CORPS"
@ 12,34 SAY " (PROTOTYPE)"

ENDO

DO WHILE .NOT. AABENDLES

ERASE

DO..."
@ 11.24 SAY "ENTER TODAY'S DATE"
ACCEPT " " DATE (DDMMYY) ====> " TO SYSDAY
IF VAL($)SYSDAY1,2]) < 32 AND. VAL($)SYSDAY3,2]) < 13 AND. VAL($)SYSDAY5,2]) = CNTY.
STORE SYSDAY TO ADATE
STORE T TO SYSDATE
ELSE
ERASE
@ 8.24 SAY "IMPROPER DATE--REENTER"
@ 8.24 SAY "**PRESS ANY KEY TO CONTINUE**"
SET CONSOLE OFF
WAIT
SET CONSOLE ON
ENDIF
ENDDO
RELEASE ALL EXCEPT AA??????
IF AAvaltas
DO Cmddet
ENDIF
STORE T TO AAENDLESS
ENDDC
E. PAUSE ROUTINE

* Routine Name: Pause.prg
* Module Name: ITBS Master Control Module
* Version: 1.0.0.0.1.C
* Author: D.E. Hausler
* Date: 30 Nov 83
* Variables Used: pause
* Variables Modified: pause
* Variables Created: pause
* Variables Released: pause
* Files (opened/closed): none
* Temporary Files Created: none
* Using Subroutines: Access.prg, Maintain.prg, Master.prg
* Description: This routine creates a delay while a message
* is being displayed.

STORE 1 TO pause
DO WHILE pause < 75
  STORE pause + 1 TO pause
ENDDO

RELEASE pause
RETURN
APPENDIX D
ACCESS MODULE LISTING

A. ACCESS ROUTINE

* Routine Name: Access.prg
* Module Name: Access Module
* Version: 2.0
* Author: L.P. Haeusler
* Date: 28 Oct 83
* Variables Used: aavalpass, password, idnum, validid,
counter, thru, pause, passrec, aalevel
* Variables Modified: thru, aavalpass, validid, idnum,
password, counter, pause, passrec,
aalevel
* Variables Created: aavalpass, password, idnum, validid,
counter, pause, thru, passrec,
aalevel
* Variables Released: counter, pause, passrec, validid,
thru
* Files (opened/closed): a:security (opened/closed)
* Temporary Files Created: none
* Using Subroutines: Master
* Description: This routine conducts the login procedures
to gain access into the system. The routine will query
the user for a user ID and a Password, after three
incorrect responses the session will be terminated.
* BEGIN Access

STORE f TO thru
STORE f TO Aavalpass
STORE " " TO password
STORE " " TO idnum
STORE 0 TO counter
DO WHILE .NOT. validid .AND. .NOT. thru
 ERASE
 @ 12,25 SAY " ENTER USER NUMBER OR " QUIT TO TERMINATE SESSION "

ACCEPT " ENTER USER NUMBER ===> " TO idnum
SICRE !(idnum) TO idnum
IF idnum = 'QUIT'
STORE t TO thru
ELSE
 ERASE
 USE A:SECURITY INDEX A:SECINDEX
 FIX &IDNUM
 IF # = 0
 USE
 ELSE
 STORE TRIM (USERPASS) TO PASSREC
 SICRE AUTHLEV TO AALEVEL
 STORE t TO VALIDID
 ENDIF
ENDIF
81
IF .NOT. valid .AND. .NOT. thru .AND. COUNTER <> 2
  STORE counter+1 TO counter
  ERASE
  @ 13, 25 SAY "INVALID USER NUMBER"
  @ 17, 25 SAY "CHECK YOUR NUMBER"
  DO pause
ELSE
  IF .NOT. valid .AND. .NOT. thru .AND. COUNTER = 2
  STORE COUNTER+1 TO COUNTER
  ENDFIP
ENDIF
IF COUNTER = 3
  STORE t TO thru
  ERASE
  @ 12, 20 SAY "***************************************************************************"
  @ 13, 20 SAY "************ TERMINATE ACCESS ************"
  @ 14, 20 SAY "***************************************************************************"
ENDIF
END DO
STORE 0 TO counter
DO WHILE .NOT. Avalpas .AND. .NOT. thru
  ERASE
  @ 7, 25 SAY "ENTER YOUR PASSWORD OR "
  @ 7, 25 SAY "QUIT" TO TERMINATE THE SESSION"
  @ 11, 18 SAY "THE PASSWORD WILL NOT BE DISPLAYED "
  @ 14, 25 SAY "ENTER PASSWORD ===> "
  SET CCONSOLE OFF.
  ACCEPT " ENTER PASSWORD ===> " TO password
  SET CCONSOLE ON
  IF !(password) = 'QUIT'
    STORE t TO thru
  ELSEIF !(password) = passrec
    STORE t TO Avalpas
  ELSEIF COUNTER <> 2
    STORE counter+1 TO counter
    ERASE
    @ 12, 25 SAY "INVALID PASSWORD"
    @ 14, 25 SAY "CHECK YOUR PASSWORD"
    DO pause
  ELSE
    STORE COUNTER+1 TO COUNTER
  ENDFIF
ENDIF
ENDIF
END IF
END WHILE
SET CCONSOLE ON.
RELEASE counter, thru, passrec, valid, thru.
RETURN
APPENDIX E
COMMAND DETERMINER MODULE LISTING

A. COMMAND DETERMINER ROUTINE

* Routine Name: Cmddet.prg
* Module Name: Command Determiner Module
* Version: 3.0
* Author: D.P. Haeusler
* Date: 30 Oct 83
* Variables Used: aafinish, cmdopt, helpopt
* Variables Modified: aafinish, cmdopt, helpopt
* Variables Created: aafinish, cmdopt, helpopt
* Variables Released: cmdopt, aafinish, helpopt
* Files (opened/closed): none
* Temporary Files Created: none
* Using Subroutines: Master
* Description: This routine creates a menu which is used to determine which options of the program will be executed by the user. These options allow the user to control the use of the DBMS by the program.

STORE f TO Aafinish
DO WHILE .NOT. Aafinish
  ERASE
  @ 4,17 SAY " USMC INDIVIDUAL TRAINING 
  @ 5,17 SAY " RECORD SYSTEM 
  @ 6,17 SAY " MAIN MENU 
  @ 7,17 SAY " SELECT ONE OF THE FOLLOWING OPTIONS BY 
  @ 8,17 SAY " ENTERING THE APPROPRIATE LETTER 
  @ 9,17 SAY " D...DATA ENTRY AND SYSTEM FUNCTIONS 
  @10,17 SAY " S...STANDARD REPORTS 
  @11,17 SAY " N...NONSTANDARD REPORTS 
  @12,17 SAY " R...HELP 
  @13,17 SAY " Q...QUIT 
  ACCEPT: " SELECT OPTION =##>>" TO cmdopt

DC CASE
  CASE !(cmdopt) = 'D'
    DO maint
      STORE f TO Aafinish
  CASE !(cmdopt) = 'S'
    DO STDRPT
      STORE f TO Aafinish
  CASE !(cmdopt) = 'N'
    DO nststdet
      STORE f TO Aafinish
  CASE !(cmdopt) = 'H'
    STORE t TO HELPOPT
      DO B:HELP2
        STORE f TO Aafinish
  CASE !(cmdopt) = 'Q'
    STORE t TO Aafinish
ENDCASE
ENDDC
RELEASE cmdopt, HELPOPT, aafinish
RETURN
E. HELP2 ROUTINE

* Routine Name: Help2.prg
* Module Name: Maintenance Module
* Version: 3.0.1
* Author: D.P. Haeusler
* Date: 6.0.1.0
* Variables Used: helropt
* Variables Modified: none
* Variables Created: none
* Variables Released: none
* Files (opened/closed): none
* Temporary Files Created: none
* Using Subroutines: Maintain.prg, Cadet.prg
* Description: This routine gives a brief description of
* the commands that can be invoked by the user.

ERASE
LO CASE
CASE HELROPT = 1
TEXT

D...DATA ENTRY AND SYSTEM FUNCTIONS: This module
allows the user to enter data on individual Marines into
the data base and to modify data for system functions.
Installation occurs in this module and any manipulation
of the data such as corrections and updates of of the
information.

S...STANDARD REPORTS: This module produces reports
that are used on a day in and day out basis. The reports
are displayed on the screen and if a paper copy is required
while the data is still on the screen, press the shift key and the
enter key at the same time, this will print everything on
the screen.

N...NONSTANDARD REPORTS: This module allows the users
to construct their own queries of the data base. It
prompts you for the information to make a query. Paper
Copies of the reports can obtained by the same method as in
Standard Reports.
ENDTEXT
@ 23.15 SAY "**** PRESS ANY KEY TO CONTINUE ****"
SET CONSOLE OFF
WAIT
SET CONSOLE ON
CASE HELROPT = 2
TEXT

V...VIEW INDIVIDUAL DATA: This module prompts you for
the name of an individual whose ITR you wish to see. This
module does not allow you to change any fields all you can
do is scan the data.

C...UPDATE DATA: This module allows you to change or
enter data in any field of any ITR. The ITR must already
exist and is identified by the individual's name. You only
need to enter enough of the name to be able to distinguish
it from another Marine's name.

C...CREATE AN ITR: This module allows the creation of
new ITR's. It will prompt you to enter all the appropriate
data. Enter the data in one field at a time. If no data
is to be entered in that field, strike the enter key leaving
the field empty.

D...DELETE AN ITR: This module allows the Data Base
Administrator to delete records in the data base. Once a
record has been deleted it is gone forever and can not be recovered. Access to this module is restricted.

S...SYSTEM FUNCTIONS: This module allows the housekeeping of the system to be accomplished by the Data Base Administrator. Access to this module is restricted.

ENDTEXT
@ 23.15 SAY " **** P SS ANY KEY TO CONTINUE ****"
SET CONSOLE OFF
WAIT
SET CONSOLE ON
ENDCASE
RETURN
APPENDIX F

NONSTANDARD COMMAND GENERATOR MODULE LISTING

A. NONSTANDARD REPORT ROUTINE

* ROUTINE NAME: NOSTEFPRT.PRG
* MODULE NAME: NONSTANDARD COMMAND GENERATOR
* VERSION: 4.0
* AUTHOR: R. Z. PRUIETT
* DATE: 05PFE94
* VARIABLES USED: MAIR1, DESCI1, MATRI2, DESCI2, MAIRI3,
* DESCI3, SICT1, SICT2, SICT3
* VARIABLES MODIFIED: MREPEAT, COUNT, PROCEED
* VARIABLES CREATED: MREPEAT, COUNT, PROCEED
* VARIABLES RELEASED: ???
* FILES CLEARED/CLOSED: A:PEFS, EST, B:QUAL
* TEMP FILES CREATED: TEMP, SICTDATA, TEMP2, TEMP3
* USING SUBROUTINES: COMMAND LETERMINER MODULE
* DESCRIPTION: THIS MODULE SEARCHES THE APPROPRIATE
* DATABASE FILES IN CEDER TC LOCATE AND DISPLAY ALL RECORDS
* THAT SATISFY A SET OF FROM ONE TO THREE USER SPECIFIED
* ATTRIBUTES.
* STORE T TO MREPEAT
* DO WHILE MREPEAT = T
* STORE F TO MREPEAT
* ERASE
* @ 10,25 SAY "NONSTANDARD COMMAND GENERATOR"
* @ 20,0 SAY "PRESS 'B' FOR HELP"
* @ 21,0 SAY "CB"
* ACCEPT "PRESS 'ENTER' TO CONTINUE" TO PROCEED
* IF IF (PROCEED) = "H"
* DO A:NCSTHLP
* STOR$: T TO MREPEAT
* ENDF
* ENDDC
* ERASE
* @ 1,25 SAY "NONSTANDARD COMMAND GENERATOR"
* @ 2,2 SAY "\n When prompted enter an attribute and its "
* "DESCRIPTION FOR EACH"
* @ 4,5 SAY "CONDITION STATEMENT DESIRED:"
* @ 6,1C SAY "LOCATE ALL MARINES THAT SATISFY THESE:
* CONDITIONS:"
* @ 9,1C SAY "<ATTRIBUTE> <DESCRIPTION> IS "
* @ 10,0 SAY "--------
* ACCPET "PRESS 'ENTER' TO CONTINUE" TO PROCEED
* STORE T TO MREPEAT
* STORE O TO CUNT
* DO WHILE MREPEAT = T AND CUNT <= 3
* STORE CUNT + 1 TO CUNT
* STORE F TO MREPEAT
* DO SICTATRI
* ERASE
* DO CASE
* CASE CUNT = 1
* @ 1,0 SAY "<ATTRIBUTE> <DESCRIPTION>"
* @ 2,0 SAY "
* @ 4,0 SAY "---
* 86
CASE COUNT = 2
CASE COUNT = 3
CASE COUNT = 3
CASE COUNT = 3

IF COUNT < 3
ENDIF
IF PROCEED = "N" .CR. COUNT >= 3
ENDIF
ENDDO

IF COUNT = 1
DO CASE
CASE (SLCT1 >= "A" .AND. SLCT1 <= "I")
USE A:PERSON
DISPLAY ALL PCF &ATRI1 = "&DESC1" OFF FIELDS;
NAME, SSN, &ATRI1
IF ECF
@ 23,50 SAY "SEARCH COMPLETE"
ENDIF
CASE (SLCT1 = "J" .OR. SLCT1 = "K")
USE EST
CCF TO A:TEMP FIELD SSN, &ATRI1 FOR &ATRI1 =;
"&DESC1"
SELECT PRIMARY
USE A:TEMP
SELECT SECONDARY
USE A:PERSON
JCIN TO SLCTDATA FOR F:SSN=S:SSN FIELDS;
NAME, SSN, &ATRI1
USE SLCTDATA
DISPLAY ALL
IF ECF
@ 23,50 SAY "SEARCH COMPLETE"
ENDIF
CASE (SLCT1 >= "I" .AND. SLCT1 <= "N")
USE EQUAL
CCF TO A:TEMP FIELD SSN, &ATRI1 FOR &ATRI1 =;
"&DESC1"
SELECT PRIMARY
USE A:TEMP
SELECT SECONDARY USE A:PERS JCIN TO SELECT DATA FOR P.SSN=S.SSN FIELDS;
NAME SSN &ATRI1
USE LCDDATA DISPLAY ALL

IF ECAS 23,50 SAY "SEARCH COMPLETE"
ENDIF EN

CASE ACCEI PRESS 'ENTER' TO PROCEED INDIF IF CCASI = 2 DO CASE
CASE (SLCT1 \(\geq\) 'A' .AND. SLCT1 \(\leq\) 'I') AND (SLCT2 \(\geq\) 'A' .AND. SLCT2 \(\leq\) 'I') USE A:PERS CCPI TO A:EMP FIELD NAME, SSN, &ATRI1, &ATRI2 FOR;
SATRI1 = "DESC1" USE A:TEMP DISPLAY FOR SATRI2 = "DESC2" OFF FIELDS;
NAME SSN, &ATRI1, &ATRI2
If ECF 23,50 SAY "SEARCH COMPLETE" EID:IF
CASE (SLCT1 \(\geq\) 'A' .AND. SLCT1 \(\leq\) 'I') AND (SLCT2 \(\geq\) 'I' .AND. SLCT2 \(\leq\) 'N') USE B:QUAL COPY TO A:TEMP FIELD SSN, &ATRI2 FOR &ATRI2 = "DESC3" SELECT PRIM FIELDS;
USE A:TEMP SELECT SECONDARY USE A:PERS JCIN TO SELECT DATA FOR P.SSN=S.SSN FIELDS;
NAME SSN, &ATRI1, &ATRI2
If ECF 23,50 SAY "SEARCH COMPLETE" EID:IF
CASE (SLCT1 \(\geq\) 'A' .AND. SLCT1 \(\leq\) 'I') AND (SLCT2 \(\geq\) 'A' .AND. SLCT2 \(\leq\) 'I') USE A:PERS CCPI TO A:EMP FIELD NAME, SSN, &ATRI1, &ATRI2 FOR;
SATRI1 = "DESC1" USE A:TEMP DISPLAY FOR SATRI2 = "DESC2" OFF FIELDS;
NAME SSN, &ATRI1, &ATRI2
If ECF 23,50 SAY "SEARCH COMPLETE" EID:IF
CASE (SLCT1 \(\geq\) 'A' .AND. SLCT1 \(\leq\) 'I') AND (SLCT2 \(\geq\) 'I' .AND. SLCT2 \(\leq\) 'N') USE B:QUAL COPY TO A:TEMP FIELD SSN, &ATRI2 FOR &ATRI2 = "DESC3" SELECT PRIM FIELDS;
USE A:TEMP SELECT SECONDARY USE B:EMP COPY TO A:TEMP FIELD SSN, &ATRI2 FOR &ATRI2 = "DESC2" SELECT PRIM FIELDS;
USE A:TEMP SELECT SECONDARY USE A:PERS JCIN TO SELECT DATA FOR P.SSN=S.SSN FIELDS;
NAME SSN, &ATRI1, &ATRI2
If ECF 23,50 SAY "SEARCH COMPLETE" EID:IF
CASE (SLCT1 \(\geq\) 'A' .AND. SLCT1 \(\leq\) 'I') AND (SLCT2 \(\geq\) 'I' .AND. SLCT2 \(\leq\) 'N') USE B:QUAL COPY TO A:TEMP FIELD SSN, &ATRI2 FOR &ATRI2 = "DESC3" SELECT PRIM FIELDS;
USE A:TEMP SELECT SECONDARY USE B:EMP COPY TO A:TEMP FIELD SSN, &ATRI2 FOR &ATRI2 = "DESC2" SELECT PRIM FIELDS;
USE A:TEMP SELECT SECONDARY USE A:PERS JCIN TO SELECT DATA FOR P.SSN=S.SSN FIELDS;
NAME SSN, &ATRI1, &ATRI2
If ECF 23,50 SAY "SEARCH COMPLETE" EID:IF
JOIN TO SLCTDATA FOR F.SSN=S.SSN Fields;
NAME, S.SN, &ATRI1, &ATRIZ
USE SLCTDATA
DISPLAY FOR &ATRI1 = "&DESC1" OFF Fields;
NAME, S.SN, &ATRI1, &ATRIZ
IF EOF
@ 23, 50 SAY "SEARCH COMPLETE"
ENDIF
CASE (SLCT1 = "J" OR SLCT1 = "K") AND;
(SLCT2 = "J" OR SLCT2 = "K")
USE B: TEMP
COPY TO A:TEMP FIELD S.SN, &ATRI1, &ATRI2 FOR &ATRI2;
= "&DESC2" AND &ATRI1 = "&DESC1"
SELECT PRIMARY
USE A: TEMP
SELECT SECONDARY
USE A: PERS
JOIN TO SLCTDATA FOR F.SSN=S.SSN Fields;
NAME, S.SN, &ATRI1, &ATRI2
USE SLCTDATA
DISPLAY ALL
IF EOF
@ 23, 50 SAY "SEARCH COMPLETE"
ENDIF
CASE (SLCT1 = "J" OR SLCT1 = "K") AND;
(SLCT2 >= "1") AND. SLCT2 <= "N")
USE B: QUAL
COPY TO A: TEMP FIELD S.SN, &ATRI1 FOR &ATRI1 =;
"&DESC1"
SELECT PRIMARY
USE A: TEMP
SELECT SECONDARY
USE A: TEMP2
JOIN TO A: TEMP3 FOR F.SSN=S.SSN Fields;
SSN, &ATRI1, &ATRI2
SELECT PRIMARY
USE A: TEMP3
SELECT SECONDARY
USE A: PERS
JOIN TO SLCTDATA FOR F.SSN=S.SSN Fields;
NAME, S.SN, &ATRI1, &ATRI2
USE SLCTDATA
DISPLAY ALL
IF EOF
@ 23, 50 SAY "SEARCH COMPLETE"
ENDIF
CASE (SLCT1 = "J" OR SLCT1 = "K") AND;
(SLCT2 >= "1") AND. SLCT2 <= "N")
USE A: PERS
COPY TO A: TEMP FIELD NAME, S.SN, &ATRI2 FOR &ATRI2 =;
"&DESC2"
SELECT PRIMARY
USE A: TEMP
SELECT SECONDARY
USE B: QUAL
JOIN TO SLCTDATA FOR F.SSN=S.SSN Fields;
NAME, S.SN, &ATRI1, &ATRI2
USE SLCTDATA
DISPLAY FOR &ATRI1 = "&DESC1" OFF Fields;
NAME, S.SN, &ATRI1, &ATRI2
IF EOF
@ 23, 50 SAY "SEARCH COMPLETE"
ENDIF
CASE (SLCT1 = "J" OR SLCT1 = "K") AND;
(SLCT2 = "J" OR SLCT2 = "K")
USE B: EST
%7,p

89
COPY TO A:TEMP FIELD SSN,&ATRI2 FOR &ATRI2 = "DESC2"
USE B:QUAL
COPY TO A:TEMP2 FIELD SSN,&ATRI1 FOR &ATRI1 = "DESC1"
SELECT PRIMARY
USE A:TEMP
SELECT SECONDARY
USE A:TEMP2
JOIN TO A:TEMP3 FOR P.SSN=S.SSN FIELDS: SSN,&ATRI1,&ATRI2
SELECT PRIMARY
USE A:TEMP3
SELECT SECONDARY
USE A:PERSON
JOIN TO SLCTDATA FOR P.SSN=S.SSN FIELDS: NAME,SSN,&ATRI1,&ATRI2
USE SLCTDATA
DISPLAY ALL IF EOF
@ 23,50 SAY "SEARCH COMPLETE"
ENDIF
CASE (SLCT1 = "I" .AND. SLCT1 <= "N") .AND.:
(SLCT2 = "L" .AND. SLCT2 <= "N")
USE B:QUAL
COPY TO A:TEMP FIELD SSN,&ATRI1,&ATRI2 FOR &ATRI1 = "DESC1" .AND. &ATRI2 = "DESC2"
SELECT PRIMARY
USE A:TEMP
SELECT SECONDARY
USE A:PERSON
JOIN TO SLCTDATA FOR P.SSN=S.SSN FIELDS: NAME,SSN,&ATRI1,&ATRI2
USE SLCTDATA
DISPLAY ALL IF EOF
@ 23,50 SAY "SEARCH COMPLETE"
ENDIF
ENDCASE
ACCEPT "PRESS 'ENTER' TO CONTINUE" TO PROCEED
ENDIF
IF CCTN = 3
DO CASE
CASE (SLCT1 = "A" .AND. SLCT1 <= "I")
DO CASE
CASE (SLCT2 = "A" .AND. SLCT2 <= "I")
DO CASE
CASE (SLCT3 = "A" .AND. SLCT3 <= "I")
USE A:PERSON
COPY TO A:TEMP FIELD NAME,SSN,&ATRI1,;
&ATRI2,&ATRI3 FOR &ATRI1="DESC1"
USE A:TEMP
COPY TO A:TEMP2 FIELD NAME,SSN,&ATRI1,;
&ATRI2,&ATRI3 FOR &ATRI2="DESC2"
USE A:TEMP
DISPLAY FOR &ATRI3="DESC3" OFF FIELDS: NAME,SSN,&ATRI1,&ATRI2,&ATRI3
IF EOF
@ 23,50 SAY "SEARCH COMPLETE"
ENDIF
CASE (SLCT3 = "J" .OR. SLCT3 = "K")
USE E:EST
COPY TO A:TEMP FIELD SSN,&ATRI3 FOR &ATRI3 = "DESC3"
SELECT PRIMARY
USE A:TEMP
SELECT SECONDARY
USE A:PERSON
90
JOIN TO SLCTDATA FOR P.SSN=S.SSN FIELDS: NAME, SSN, &ATRI1, &ATRI2, &ATRI3
USE SLCTDATA
DISPLAY FOR &ATRI1 = "$DESC1" . AND. &ATRI2 = "$DESC2" OFF FIELDS NAME, SSN, &ATRI1,;
&ATRI2, &ATRI3
IF ECF
   @ 2,50 SAY "SEARCH COMPLETE"
ENDIF
CASE (SLCT3 >= "1" .AND. SLCT3 <= "N")
USE #:QUAL
COPY TO A:TEMP FIELD SSN,&ATRI3 FOR &ATRI3;
= "$DESC2"
SELECT PRIMARY
USE #:TEMP
SELECT SECONDARY
USE #:PERS
JOIN TO SLCTDATA FOR P.SSN=S.SSN FIELDS:
NAME, SSN, &ATRI1, &ATRI2, &ATRI3
USE SLCTDATA
DISPLAY FOR &ATRI1 = "$DESC1" . AND. &ATRI2 = "$DESC2" OFF FIELDS NAME, SSN, &ATRI1,;
&ATRI2, &ATRI3
IF ECF
   @ 2,50 SAY "SEARCH COMPLETE"
ENDIF
ENDCASE
CASE (SLCT2 >= "L" .AND. SLCT2 <= "N")
DO CASE
   CASE (SLCT3 >= "A" .AND. SLCT3 <= "I")
   USE #:QUAL
   COPY TO A:TEMP FIELD SSN,&ATRI2 FOR &ATRI2;
   = "$DESC2"
   SELECT PRIMARY
   USE #:TEMP
   SELECT SECONDARY
   USE #:PERS
   JOIN TO SLCTDATA FOR P.SSN=S.SSN FIELDS:
   NAME, SSN, &ATRI1, &ATRI2, &ATRI3
   USE SLCTDATA
   DISPLAY FOR &ATRI1 = "$DESC1" . AND. &ATRI2 = "$DESC3" OFF FIELDS NAME, SSN, &ATRI1,;
   &ATRI2, &ATRI3
   IF ECF
      @ 2,50 SAY "SEARCH COMPLETE"
   ENDIF
   END
   CASE (SLCT3 >= "I" .AND. SLCT3 <= "N")
   USE #:PERS
   COPY TO A:TEMP FIELD NAME,SSN,&ATRI1 FOR;
   &ATRI1 = "$DESC1"
   SELECT PRIMARY
   USE #:TEMP
   SELECT SECONDARY
   USE #:QUAL
   JOIN TO SLCTDATA FOR P.SSN=S.SSN FIELDS:
   NAME, SSN, &ATRI1, &ATRI2, &ATRI3
   USE SLCTDATA
   DISPLAY FOR &ATRI2 = "$DESC1" . AND. &ATRI3 = "$DESC3" OFF FIELDS NAME, SSN, &ATRI1,;
   &ATRI2, &ATRI3
   IF ECF
      @ 2,50 SAY "SEARCH COMPLETE"
   ENDIF
   END
   CASE (SLCT3 = "J"
   .OR. SLCT3 = ",")
   USE #:PERS
   COPY TO A:TEMP FIELD NAME,SSN,&ATRI1 FOR;
   &ATRI1 = "$DESC1"
   SELECT PRIMARY
   USE #:TEMP

91
SELECT SECONDARY
USE E:QUAL
JOIN TO A:TEMP FOR P.SSN=S.SSN FIELDS:
NAME, SSN, &ATRI1, &ATRI2
SELECT PRIMARY
USE A:TEMP
SELECT SECONDARY
USE E:EST
JOIN TO SLCTDATA FOR F.SSN=S.SSN FIELDS:
NAME, SSN, &ATRI1, &ATRI2, &ATRI3
USE SLCTDATA
DISPLAY FOR &ATRI2 = "&DESC2" .AND. &ATRI3;
&ATRI2, &ATRI3
IF ECF
@ 23,50 SAY "SEARCH COMPLETE"
ENDIF
ENDCASE
CASE (SLCT2 = "J" .OR. SLCT2 = "K")
DO CASE
CASE (SLCT3 >= "A" .AND. SLCT3 <= "I")
USE E:EST
COPY TO A:TEMP FIELD SSN,&ATRI2 FOR &ATRI2;
= "&DESC2"
SELECT PRIMARY
USE A:TEMP
SELECT SECONDARY
USE A:PERS
JOIN TO SLCTDATA FOR F.SSN=S.SSN FIELDS:
NAME, SSN, &ATRI1, &ATRI2, &ATRI3
USE SLCTDATA
DISPLAY FOR &ATRI1 = "&DESC1" .AND. &ATRI3;
&ATRI2, &ATRI3
IF ECF
@ 23,50 SAY "SEARCH COMPLETE"
ENDIF
CASE (SLCT3 >= "I" .AND. SLCT3 <= "N")
USE A:PERS
COPY TO A:TEMP FIELD NAME,SSN,&ATRI1 FOR;
&ATRI1 = "&DESC1"
USE E:EST
COPY TO A:TEMP2 FIELD SSN,&ATRI2 FOR;
&ATRI2 = "&DESC2"
SELECT PRIMARY
USE A:TEMP
SELECT SECONDARY
USE A:TEMP
JOIN TO A:TEMP3 FOR P.SSN=S.SSN FIELDS:
NAME, SSN, &ATRI1, &ATRI2, &ATRI3
SELECT PRIMARY
USE A:TEMP3
SELECT SECONDARY
USE E:QUAL
JOIN TO SLCTDATA FOR P.SSN=S.SSN FIELDS:
NAME, SSN, &ATRI1, &ATRI2, &ATRI3
USE SLCTDATA
DISPLAY FOR &ATRI3 = "&DESC3" OFF FIELDS:
NAME, SSN, &ATRI1,;
&ATRI2, &ATRI3
IF ECF
@ 23,50 SAY "SEARCH COMPLETE"
ENDIF
CASE (SLCT3 = "J" .OR. SLCT3 = "K")
USE A:PERS
COPY TO A:TEMP FIELD NAME,SSN,&ATRI1 FOR;
&ATRI1 = "&DESC1"
SELECT PRIMARY
USE A:TEMP
SELECT SECONDARY
USE E:EST
JOIN TO SLCTDATA FOR P.SSN=S.SSN FIELDS:
NAME, SSN, &ATRI1, &ATRI2, &ATRI3
USE SLCTDATA
DISPLAY FOR &ATRI2 = "&DESC2" .AND. &ATRI3 = "&DESC3" OFF FIELDS NAME, SSN, &ATRI1;
&ATRI4, &ATRI3
IF ECM
@ 23,50 SAY "SEARCH COMPLETE"
ENDIF
ENDCASE
**BEGIN QUAL / * / *
CASE (SLCT1 >= "I" .AND. SLCT1 <= "N")
**BEGIN QUAL / PERS / *
DO CASE
CASE (SLCT2 >= "A" .AND. SLCT2 <= "I")
DO CASE
CASE (SLCT3 >= "A" .AND. SLCT3 <= "I")
USE A:PERS
COPY TO A:TEMP FIELD NAME, SSN, &ATRI2:
&ATRI3 = "&DESC2" .AND.
&ATRI3 = "&DESC3"
SELECT PRIMARY
USE A:TEMP
SELECT SECONDARY
USE E:QUAL
JOIN TO SLCTDATA FOR P.SSN=S.SSN FIELDS:
NAME, SSN, &ATRI1, &ATRI2, &ATRI3
USE SLCTDATA
DISPLAY FOR &ATRI1 = "&DESC1" OFF FIELDS;
NAME, SSN, &ATRI1, &ATRI2, &ATRI3
IF ECM
@ 23,50 SAY "SEARCH COMPLETE"
ENDIF
CASE (SLCT3 = "J" .OR. SLCT3 = "K")
USE E:EST
COPY TO A:TEMP FIELD SSN, &ATRI3 FOR &ATRI3;
= "&DESC3"
USE E:QUAL
COPY TO A:TEMP FIELD SSN, &ATRI1 FOR;
&ATRI1 = "&DESC1"
SELECT PRIMARY
USE A:TEMP
SELECT SECONDARY
USE A:TEMP
JOIN TO SLCTDATA FOR P.SSN=S.SSN FIELDS:
NAME, SSN, &ATRI1, &ATRI3
SELECT PRIMARY
USE A:TEMP
SELECT SECONDARY
USE A:TEMP
JOIN TO SLCTDATA FOR P.SSN=S.SSN FIELDS:
NAME, SSN, &ATRI1, &ATRI2, &ATRI3
USE SLCTDATA
DISPLAY FOR &ATRI1 = "&DESC1"
OFF FIELDS NAME, SSN, &ATRI1, &ATRI2, &ATRI3
IF ECM
@ 23,50 SAY "SEARCH COMPLETE"
ENDIF
CASE (SLCT3 >= "L" .AND. SLCT3 <= "N")
USE E:QUAL
COPY TO A:TEMP FIELD SSN, &ATRI1, &ATRI3;
FOR &ATRI3 = "&DESC2" .AND. &ATRI1 = "&DESC3"
SELECT PRIMARY
USE A:TEMP
SELECT SECONDARY
USE A:TEMP

93
JOIN TO SLCTDATA FOR P.SSN=S.SSN FIELDS;
NAME, SNN, &ATRI1, &ATRI2, &ATRI3
USE SLCTDATA
DISPLAY FOR &ATRI2 = "EDESC2":
OFF FIELDS NAME, SNN, &ATRI1, &ATRI2, &ATRI3
IF ECF
  @ 2350 SAY "SEARCH COMPLETE"
ENDIF
ENDCASE
CASE (SLCT2 = "L") .AND. SLCT2 = "N")
DO CASE
  CASE (SLCT3 = "A") .AND. SLCT3 = "I")
    USE E:QUAL
    COPY TO A:TEMP FIELD SNN, &ATRI1, &ATRI2;
    FOR &ATRI1 = "EDESC1" .AND. &ATRI2 = "EDESC2":
    SELECT PRIMARY
    USE A:TEMP
    SELECT SECONDARY
    USE A:TEMP
    JOIN TO SLCTDATA FOR P.SSN=S.SSN FIELDS;
NAME, SNN, &ATRI1, &ATRI2, &ATRI3
USE SLCTDATA
DISPLAY FOR &ATRI3 = "EDESC3":
OFF FIELDS NAME, SNN, &ATRI1, &ATRI2, &ATRI3
IF ECF
  @ 2350 SAY "SEARCH COMPLETE"
ENDIF
CASE (SLCT3 = "A") .AND. SLCT3 <= "I")
ENDCASE
CASE (SLCT3 = "J") .OR. SLCT3 = "K")
ENDCASE
CASE (SLECT2 = "J" .OR. SLECT2 = "K")
CASE (SLECT3 >= "A" .AND. SLECT3 <= "I")
USE E:EST
COPY TO A:TEMP FIELD SSN,&ATRI2 FOR &ATRI2 = "&DESC2"
USE E:QUAL
COPY TO A:TEMP2 FIELD SSN,&ATRI1 FOR:
&ATRI1 = "&DESC1"
SELECT PRIMARY
USE A:TEMP
SELECT SECONDARY
USE A:TEMP2
JOIN TO A:TEMP3 FOR P.SSN=S.SSN FIELDS:
SSN,&ATRI1,&ATRI2
SELECT PRIMARY
USE A:TEMP3
SELECT SECONDARY
USE A:TEMP4
DISPLAY FOR S.&ATRI1="&DESC1" .AND. &ATR1 = "&DESC2"
SELECT PRIMARY
USE A:TEMP4
SELECT SECONDARY
USE A:TEMP1
DISPLAY FOR P.&ATRI1="&DESC1" .AND. &ATR1 = "&DESC2"
SELECT PRIMARY
USE A:TEMP1
SELECT SECONDARY
USE A:TEMP2
JOIN TO A:TEMP3 FOR P.SSN=S.SSN FIELDS:
SSN, &ATRI1, &ATRI2
SELECT PRIMARY
USE A:TEMP2
SELECT SECONDARY
USE A:TEMP
JOIN TO A:TEMP3 FOR P.SSN=S.SSN FIELDS:
SSN, &ATRI1, &ATRI2, &ATRI3
SELECT PRIMARY
USE A:TEMP
SELECT SECONDARY
USE A:TEMP
JOIN TO A:TEMP3 FOR P.SSN=S.SSN FIELDS:
SSN, &ATRI1, &ATRI2, &ATRI3
SELECT PRIMARY
USE A:TEMP
SELECT SECONDARY
USE A:TEMP
JOIN TO A:TEMP3 FOR P.SSN=S.SSN FIELDS:
SSN, &ATRI1, &ATRI2, &ATRI3
SELECT PRIMARY
USE A:TEMP
SELECT SECONDARY
USE A:TEMP
JOIN TO A:TEMP3 FOR P.SSN=S.SSN FIELDS:
SSN, &ATRI1, &ATRI2, &ATRI3
SELECT PRIMARY
USE A:TEMP
SELECT SECONDARY
USE A:TEMP
JOIN TO A:TEMP3 FOR P.SSN=S.SSN FIELDS:
SSN, &ATRI1, &ATRI2, &ATRI3
SELECT PRIMARY
USE A:TEMP
SELECT SECONDARY
USE A:TEMP
JOIN TO A:TEMP3 FOR P.SSN=S.SSN FIELDS:
SSN, &ATRI1, &ATRI2, &ATRI3
SELECT PRIMARY
USE A:TEMP
SELECT SECONDARY
USE A:TEMP
JOIN TO A:TEMP3 FOR P.SSN=S.SSN FIELDS:
SSN, &ATRI1, &ATRI2, &ATRI3
SELECT PRIMARY
USE A:TEMP
SELECT SECONDARY
USE A:TEMP
JOIN TO A:TEMP3 FOR P.SSN=S.SSN FIELDS:
SSN, &ATRI1, &ATRI2, &ATRI3
SELECT PRIMARY
USE A:TEMP
SELECT SECONDARY
USE A:TEMP
JOIN TO A:TEMP3 FOR P.SSN=S.SSN FIELDS:
SSN, &ATRI1, &ATRI2, &ATRI3
SELECT PRIMARY
USE A:TEMP
SELECT SECONDARY
USE A:TEMP
JOIN TO A:TEMP3 FOR P.SSN=S.SSN FIELDS:
SSN, &ATRI1, &ATRI2, &ATRI3
SELECT PRIMARY
USE A:TEMP
SELECT SECONDARY
USE A:TEMP
JOIN TO A:TEMP3 FOR P.SSN=S.SSN FIELDS:
SSN, &ATRI1, &ATRI2, &ATRI3
SELECT PRIMARY
USE A:TEMP
SELECT SECONDARY
USE A:TEMP
JOIN TO A:TEMP3 FOR P.SSN=S.SSN FIELDS:
SSN, &ATRI1, &ATRI2, &ATRI3
SELECT PRIMARY
USE A:TEMP
SELECT SECONDARY
USE A:TEMP
JOIN TO A:TEMP3 FOR P.SSN=S.SSN FIELDS:
SSN, &ATRI1, &ATRI2, &ATRI3
SELECT PRIMARY
USE A:TEMP
SELECT SECONDARY
USE A:TEMP
JOIN TO A:TEMP3 FOR P.SSN=S.SSN FIELDS:
SSN, &ATRI1, &ATRI2, &ATRI3
SELECT PRIMARY
USE A:TEMP
SELECT SECONDARY
USE A:TEMP
JOIN TO A:TEMP3 FOR P.SSN=S.SSN FIELDS:
SSN, &ATRI1, &ATRI2, &ATRI3
SELECT PRIMARY
USE A:TEMP
SELECT SECONDARY
USE A:TEMP
JOIN TO A:TEMP3 FOR P.SSN=S.SSN FIELDS:
SSN, &ATRI1, &ATRI2, &ATRI3
SELECT PRIMARY
USE A:TEMP
SELECT SECONDARY
USE A:TEMP
JOIN TO A:TEMP3 FOR P.SSN=S.SSN FIELDS:
SSN, &ATRI1, &ATRI2, &ATRI3
SELECT PRIMARY
USE A:TEMP
SELECT SECONDARY
USE A:TEMP
JOIN TO A:TEMP3 FOR P.SSN=S.SSN FIELDS:
SSN, &ATRI1, &ATRI2, &ATRI3
SELECT PRIMARY
USE A:TEMP
SELECT SECONDARY
USE A:TEMP
JOIN TO A:TEMP3 FOR P.SSN=S.SSN FIELDS:
SSN, &ATRI1, &ATRI2, &ATRI3
SELECT PRIMARY
USE A:TEMP
SELECT SECONDARY
USE A:TEMP
JOIN TO A:TEMP3 FOR P.SSN=S.SSN FIELDS:
SSN, &ATRI1, &ATRI2, &ATRI3
SELECT PRIMARY
USE A:TEMP
SELECT SECONDARY
USE A:TEMP
JOIN TO A:TEMP3 FOR P.SSN=S.SSN FIELDS:
SSN, &ATRI1, &ATRI2, &ATRI3
SELECT PRIMARY
USE A:TEMP
SELECT SECONDARY
USE A:TEMP
JOIN TO A:TEMP3 FOR P.SSN=S.SSN FIELDS:
** END CASE **

** BEGIN EST / **

** BEGIN EST /

CASE (SLCT1 = "J" . OR. SLCT1 = "K")

** BEGIN EST /

CASE (SLCT2 >= "A" . AND. SLCT2 <= "I")

DO CASE

CASE (SLCT3 >= "A" . AND. SLCT3 <= "I")

USE A: PERS

COPY TO A: TEMP FIELD NAME, SSN, &ATRI2;

&ATRI3 FOR &ATRI2 = "DESC2" . AND. &ATRI3 = "DESC3"

USE E: EST

COPY TO A: TEMP2 FIELD SSN, &ATRI1 FOR;

&ATRI1 = "DESC1"

SELECT PRIMARY

USE A: TEMP

SELECT SECONDARY

USE A: PERS

DISPLAY FOR P.SSN=S.SSN OFF FIELDS NAME,;

SSN, &ATRI1, &ATRI2, &ATRI3

IF ECF

@ 23,50 SAY "SEARCH COMPLETE"

END CASE

CASE (SLCT3 = "J") . OR. SLCT3 = "K")

USE E: EST

COPY TO A: TEMP FIELD SSN, &ATRI1, &ATRI3;

FOR &ATRI1 = "DESC1" . AND. &ATRI3 = "DESC3"

SELECT PRIMARY

USE A: TEMP

SELECT SECONDARY

USE A: PERS

DISPLAY FOR P.SSN=S.SSN .AND. S.&ATR12 =;

"DESC2" OFF FIELDS NAME, SSN, &ATRI1,;

&ATRI2, &ATRI3

IF ECF

@ 23,50 SAY "SEARCH COMPLETE"

END CASE

CASE (SLCT3 = "L" . AND. SLCT3 <= "N")

USE E: QUAL

COPY TO A: TEMP FIELD SSN,&ATRI3 FOR &ATRI3;

= "DESC3"

USE E: EST

COPY TO A: TEMP2 FIELD SSN, &ATRI1 FOR;

&ATRI1 = "DESC1"

SELECT PRIMARY

USE A: TEMP

SELECT SECONDARY

USE A: TEMP

JOIN TO SLCTDATA FOR P.SSN=S.SSN.

FIELDS;

SSN, &ATRI1, &ATRI3

SELECT PRIMARY

USE SLCTDATA

SELECT SECONDARY

USE A: PERS

DISPLAY FOR P.SSN=S.SSN .AND. &ATRI2 =;

"DESC2" OFF FIELDS NAME, SSN, &ATRI1,;

&ATRI2, &ATRI3

IF ECF

@ 23,50 SAY "SEARCH COMPLETE"

END CASE

CASE (SLCT2 = "L" . AND. SLCT2 <= "N")

DO CASE

CASE (SLCT3 = "A" . AND. SLCT3 <= "I")

USE E: QUAL

COPY TO A: TEMP FIELD SSN, &ATRI2 FOR &ATRI2;
= "EDESC2"
USE E:EST
COPY TO A:TEMP2 FIELD SSN,&ATRI1 FOR;
&ATRI1="EDESC1"
SELECT PRIMARY
USE A:TEMP
SELECT SECONDARY
USE A:TEMP2
JOIN TO SLCTDATA FOR P.SSN=S.SSN FIELDS:
SSN, &ATRI1, &ATRI2
SELECT PRIMARY
USE SLCTDATA
SELECT SECONDARY
USE A:TEMP
DISPLAY FOR &ATRI3 = "EDESC3" .AND.;
P.SSN=S.SSN OFF FIELDS NAME, SSN, &ATRI1,;
&ATRI2, &ATRI3
IF ECF
  @ 22,50 SAY "SEARCH COMPLETE"
ENDIF
CASE (SLCT3 >= "I" .AND. SLCT3 <= "N")
USE E:QUAL
COPY TO A:TEMP FIELD SSN,&ATRI2,&ATRI3;
FOR &ATRI2="EDESC2", &ATRI3="EDESC3"
USE E:EST
COPY TO A:TEMP2 FIELD SSN,&ATRI1 FOR;
&ATRI1="EDESC1" .AND. &ATRI3="EDESC3"
USE A:TEMP
SELECT PRIMARY
USE A:TEMP
SELECT SECONDARY
USE A:TEMP2
JOIN TO SLCTDATA FOR P.SSN=S.SSN FIELDS:
SSN, &ATRI1, &ATRI2, &ATRI3
SELECT PRIMARY
USE SLCTDATA
SELECT SECONDARY
USE A:TEMP
DISPLAY FOR P.SSN=S.SSN OFF FIELDS NAME,;
SSN, &ATRI1, &ATRI2, &ATRI3
IF ECF
  @ 22,50 SAY "SEARCH COMPLETE"
ENDIF
CASE (SLCT3 = "J" .OR. SLCT3 = "K")
USE E:EST
COPY TO A:TEMP FIELD SSN,&ATRI1,&ATRI3;
FOR &ATRI1="EDESC1", &ATRI3="EDESC3"
USE E:QUAL
COPY TO A:TEMP2 FIELD SSN,&ATRI2 FOR;
&ATRI2="EDESC2"
SELECT PRIMARY
USE A:TEMP
SELECT SECONDARY
USE A:TEMP2
JOIN TO A:TEMP FOR P.SSN=S.SSN FIELDS;
SSN, &ATRI1, &ATRI2, &ATRI3
SELECT PRIMARY
USE A:TEMP3
SELECT SECONDARY
USE A:TEMP
DISPLAY FOR P.SSN=S.SSN OFF FIELDS NAME,;
SSN, &ATRI1, &ATRI2, &ATRI3
IF ECF
  @ 22,50 SAY "SEARCH COMPLETE"
ENDIF
ENDCASE
CASE (SLCT2 = "J" .CR. SLCT2 = "K")
  TO CASE
CASE (SLCT3 >= "A" .AND. SLCT3 <= "I")
  USE E:EST
COPY TO A:TEMP FIELD SSN, &ATRI1, &ATRI2;
FOR &ATRI1 = "&DESC1" .AND. &ATRI2 = "&DESC2"
SELECT PRIMARY
USE A:TEMP
SELECT SECONDARY
USE A:TEMP
DISPLAY FOR P.SSN = S.SSN .AND. &ATRI3 = "&DESC3"
OFF FIELDS NAME, SSN, &ATRI1,
&ATRI2, &ATRI3
IF ECF @ 23,50 SAY "SEARCH COMPLETE"
ENDF
CASE (SLCT3 = "J") .OR. SLCT3 = "K"
USE A:EST
COPY TO A:TEMP FIELD SSN, &ATRI1, &ATRI2;
FOR &ATRI1 = "&DESC1" .AND. &ATRI2 = "&DESC2"
USE E:QUAL
COPY TO A:TEMP2 FIELD SSN, &ATRI3 FOR;
&ATRI3 = "&DESC3"
SELECT PRIMARY
USE A:TEMP
SELECT SECONDARY
USE A:TEMP2
JOIN TO SLCTDATA FOR P.SSN = S.SSN
FIELDS:
SSN, &ATRI1, &ATRI2, &ATRI3
SELECT PRIMARY
USE SLCTDATA
SELECT SECONDARY
USE A:TEMP
DISPLAY FOR P.SSN = S.SSN
OFF FIELDS NAME, SSN, &ATRI1, &ATRI2, &ATRI3
IF ECF @ 23,50 SAY "SEARCH COMPLETE"
ENDF
CASE (SLCT3 = "J") .OR. SLCT3 = "K"
USE A:EST
COPY TO A:TEMP FIELD SSN, &ATRI1, &ATRI2;
&ATRI3 FOR &ATRI1 = "&DESC1" .AND. &ATRI2 = "&DESC2"
&ATRI3 = "&DESC3"
SELECT PRIMARY
USE A:TEMP
SELECT SECONDARY
USE A:TEMP
DISPLAY FOR P.SSN = S.SSN
OFF FIELDS NAME, SSN, &ATRI1, &ATRI2, &ATRI3
IF ECF @ 23,50 SAY "SEARCH COMPLETE"
ENDF
CASE (SLCT3 = "J") .OR. SLCT3 = "K"
USE A:EST
COPY TO A:TEMP FIELD SSN, &ATRI1, &ATRI2;
&ATRI3 FOR &ATRI1 = "&DESC1" .AND. &ATRI2 = "&DESC2"
&ATRI3 = "&DESC3"
SELECT PRIMARY
USE A:TEMP
SELECT SECONDARY
USE A:TEMP
DISPLAY FOR P.SSN = S.SSN
OFF FIELDS NAME, SSN, &ATRI1, &ATRI2, &ATRI3
IF ECF @ 23,50 SAY "SEARCH COMPLETE"
ENDF
ENDCASE
ACÆCT "PRESS 'ENTER' TO CONTINUE" TO PROCEED
ENDF
RELEASE ALL LIKE A:TEMP?
RELEASE REPEAT.COUNT
RELEASE ALL LIKE A:TEMP
RELEASE ALL LIKE SLCT????
RELEASE ALL LIKE DESC????
SET CCNSCLE OFF
DELETE FILE A:TEMP
DELETE FILE A:TEMP2
DELETE FILE A:TEMP3
SET CCNSCLE ON
RETURN
E. NONSTANDARD HELP ROUTINE

* ROUTINE NAME: NOSTHELP.PRG
* MODULE NAME: NONSTANDARD REPORT GENERATOR
* VERSION: 4,1,0
* AUTHOR: R.E. PRUETT
* DATE: 10FEB84
* VARIABLES USED: NONE
* VARIABLES MODIFIED: NONE
* VARIABLES CREATED: PROCEED
* VARIABLES RELEASED: NONE
* FILES OPENED/CLOSED: NONE
* TIME FILES CREATED: NONE
* USING SUBROUTINES: NOSTDRPT.PRG (4,0)
* DESCRIPTION: THIS ROUTINE SUPPLIES RESPONSE TO THE USER'S REQUEST FOR "HELP" WHILE IN THE NONSTANDARD REPORT MENU.

FRASE
@0,25 SAY "NONSTANDARD COMMAND GENERATOR"
@2,0 SAY "LOCATE ALL MARINES THAT SATISFY THE FOLLOWING; CONDITION:"
@4,0 SAY "<ATTRIBUTE> <DESCRIPTOR>
E5"
@5,0 SAY "RANK IS
E5"
@6,0 SAY "UP TO THREE CONDITION STATEMENTS MAY BE;
JOINED"
@7,0 SAY "LOCATE ALL MARINES THAT SATISFY THE FOLLOWING; CONDITIONS"
@14,0 SAY "<ATTRIBUTE> <DESCRIPTOR>
E5"
@15,0 SAY "RANK IS
E5"
@16,0 SAY "COMPANY AND IS A CO"
@17,0 SAY "PRIMARY MOS IS 0331"
@19,0 SAY "JOINING IF THESE THREE CONDITION STATEMENTS GIVE; YOU THE NAME"
@22,0 SAY "AND SSN OF ALL E5'S IN A CO. THAT HAVE A 0331; PRIMARY MOS"
@22,0 SAY "PRESS 'ENTER' TO CONTINUE" TO PROCEED
RETURN
C. SELECT ATTRIBUTE FOUTINE

* ROUTINE NAME: SLCTATEI.PRG
* MODUKE NAME: NONSTANDARD REPORT GENERATOR
* VERSION: 4.2.0
* AUTHOR: R.E. PRUIETT
* DATE: 12FEB84
* VARIABLES USED: COUNT, SLCT1, SLCT2, SLCT3
* VARIABLES MODIFIED: NONE
* VARIABLES CREATED: REPEAT, MATRIBUT, MATRI1, ATRI1,
  MATRI2, MATRI2, MATRI3, ATRI3
* VARIABLES RELEASED: NONE
* FILES OPENED/CLOSED: NONE
* USER FILES CREATED: NONE
* USING SUBROUTINES: NCSTDRT.FEG [4.0]
* DESCRIPTION: THIS ROUTINE INCURPTS THE USER TO SELECT
  Attributes. After the first and second selection the
  user is asked if he wants to select another attribute.
  If the response is YES, the selection procedure is
  repeated. This process is continued until the user
  answers NO or three attributes have been selected. The
  routine will automatically sort and locate the
  appropriate database files and display the name, SSN and
  attribute information for each record that satisfies the
  user specified criteria.

STORE T TO REPEAT
DO WHILE REPEAT = T
STORE F TO REPEAT
ERASE
IF COUNT = 2 OR COUNT = 3
DO CASE
CASE SLCT1 = "A"
   @ 1,1 SAY "*
CASE SLCT1 = "B"
   @ 2,1 SAY "*
CASE SLCT1 = "C"
   @ 3,1 SAY "*
CASE SLCT1 = "D"
   @ 4,1 SAY "*
CASE SLCT1 = "E"
   @ 5,1 SAY "*
CASE SLCT1 = "F"
   @ 6,1 SAY "*
CASE SLCT1 = "G"
   @ 7,1 SAY "*
CASE SLCT1 = "H"
   @ 8,1 SAY "*
CASE SLCT1 = "I"
   @ 9,1 SAY "*
CASE SLCT1 = "J"
   @ 10,1 SAY "*
CASE SLCT1 = "K"
   @ 11,1 SAY "*
CASE SLCT1 = "L"
   @ 12,1 SAY "*
CASE SLCT1 = "M"
   @ 13,1 SAY "*
CASE SLCT1 = "N"
   @ 14,1 SAY "*
ENDCASE
ENDCASE
ENDCASE
IF COUNT = 3
DO CASE
CASE SLCT2 = "A"
   @ 1,1 SAY "*
CASE SLCT2 = "B"
   @ 2,1 SAY "*
CASE SLCT2 = "C"
   @ 3,1 SAY "*
ENDCASE
100
CASE SICT2 = "C"
   3,1 SAY "*"
CASE SICT2 = "D"
   4,1 SAY "*"
CASE SICT2 = "E"
   5,1 SAY "*"
CASE SICT2 = "F"
   6,1 SAY "*"
CASE SICT2 = "G"
   7,1 SAY "*"
CASE SICT2 = "H"
   8,1 SAY "*"
CASE SICT2 = "I"
   9,1 SAY "*"
CASE SICT2 = "J"
  10,1 SAY "*"
CASE SICT2 = "K"
  11,1 SAY "*"
CASE SICT2 = "L"
  12,1 SAY "*"
CASE SICT2 = "M"
  13,1 SAY "*"
CASE SICT2 = "N"
  14,1 SAY "*"
ENDCASE
ENDIF

&0,29 SAY "SELECT ATTRIBUTE"
& 1,2 SAY "A...RANK"
& 2,2 SAY "B...PRIMARY MOS"
& 3,2 SAY "C...SECONDARY MOS"
& 4,2 SAY "D...COMPANY"
& 5,2 SAY "E...PLATOON"
& 6,2 SAY "F...JOIN DATE"
& 7,2 SAY "G...EAS"
& 8,2 SAY "H...BIRTH DATE"
& 9,2 SAY "I...WEIGHT CONTROL"
&10,2 SAY "J...PFT CSE"
&11,2 SAY "K...PFT TRO"
&12,2 SAY "L...SWIM QUALIFICATION"
&13,2 SAY "M...RIFLE QUALIFICATION"
&14,2 SAY "N...PISTOL QUALIFICATION"
&22,0 ACCEPT "SELECT ATTRIBUTE ===>" TO MATRI

IF COUNT = 1
STORE MATRIBUT TO SLCT1
DO CASE
   CASE !(SLCT1) = "A"
      STORE 'RANK' TO MATRI
      STORE 'RANK' TO ATRI
   CASE !(SLCT1) = "B"
      STORE 'PRIMARY MOS' TO MATRI
      STORE 'PRIMENOS' TO ATRI
   CASE !(SLCT1) = "C"
      STORE 'SECONDARY MOS' TO MATRI
      STORE 'SECMOS' TO ATRI
   CASE !(SLCT1) = "D"
      STORE 'COMPANY' TO MATRI
      STORE 'COMPANY' TO ATRI
   CASE !(SLCT1) = "E"
      STORE 'PLATOON' TO MATRI
      STORE 'PLATOON' TO ATRI
   CASE !(SLCT1) = "F"
      STORE 'JOIN DATE' TO MATRI
      STORE 'JOIN DATE' TO ATRI
   CASE !(SLCT1) = "G"
      STORE 'EAS' TO MATRI
      STORE 'EAS' TO ATRI
   CASE !(SLCT1) = "H"
      STORE 'BIRTH DATE' TO MATRI
      STORE 'BIRTH DATE' TO ATRI
STORE 'BIRTH DATE' TO ATRI1
CASE 1(SLCT1) = "Y"
  STORE 'HEIGHT CONTROL' TO MATRI1
  STORE 'WT CONTROL' TO ATRI1
CASE 1(SLCT1) = "J"
  STORE 'FT ONE' TO MATRI1
  STORE 'FT1 CLSS' TO ATRI1
CASE 1(SLCT1) = "K"
  STORE 'FT TWO' TO MATRI1
  STORE 'FT2 CLSS' TO ATRI1
CASE 1(SLCT1) = "I"
  STORE 'SWIM QUALIFICATION' TO MATRI1
  STORE 'SWIM QUAL' TO ATRI1
CASE 1(SLCT1) = "H"
  STORE 'TRIPLE QUALIFICATION' TO MATRI1
  STORE 'RPLQUAL' TO ATRI1
CASE 1(SLCT1) = "N"
  STORE 'PISTOL QUALIFICATION' TO MATRI1
  STORE 'PSTQUAL' TO ATRI1
OTHERWISE
  STORE I TO REPEAT
ENDCASE
ENDIF

IF CCENT = 2
  STORE MATRIBUT TO SLCT2
  DO CASE
    CASE 1(SLCT2) = "A"
      STORE 'RANK' TO MATRI2
      STORE 'RANK' TO ATRI2
    CASE 1(SLCT2) = "E"
      STORE 'PRIMARY MCS' TO MATRI2
      STORE 'PRIM MCS' TO ATRI2
    CASE 1(SLCT2) = "C"
      STORE 'SECONDARY MCS' TO MATRI2
      STORE 'SEC MCS' TO ATRI2
    CASE 1(SLCT2) = "D"
      STORE 'COMPANY' TO MATRI2
      STORE 'COMPANY' TO ATRI2
    CASE 1(SLCT2) = "F"
      STORE 'PLATCON' TO MATRI2
      STORE 'PLATCON' TO ATRI2
    CASE 1(SLCT2) = "F"
      STORE 'JOIN DATE' TO MATRI2
      STORE 'JOIN DATE' TO ATRI2
    CASE 1(SLCT2) = "G"
      STORE 'EAS' TO MATRI2
      STORE 'EAS' TO ATRI2
    CASE 1(SLCT2) = "E"
      STORE 'BIRTH DATE' TO MATRI2
      STORE 'BIRTH DATE' TO ATRI2
    CASE 1(SLCT2) = "H"
      STORE 'HEIGHT CONTROL' TO MATRI2
      STORE 'WT CONTROL' TO ATRI2
    CASE 1(SLCT2) = "J"
      STORE 'FT ONE' TO MATRI2
      STORE 'FT1 CLSS' TO ATRI2
    CASE 1(SLCT2) = "K"
      STORE 'FT TWO' TO MATRI2
      STORE 'FT2 CLSS' TO ATRI2
    CASE 1(SLCT2) = "L"
      STORE 'SWIM QUALIFICATION' TO MATRI2
      STORE 'SWIM QUAL' TO ATRI2
    CASE 1(SLCT2) = "M"
      STORE 'TRIPLE QUALIFICATION' TO MATRI2
      STORE 'RPLQUAL' TO ATRI2
    CASE 1(SLCT2) = "N"
      STORE 'PISTOL QUALIFICATION' TO MATRI2
      STORE 'PSTQUAL' TO ATRI2
  ENDCASE
ENDIF
ELSE\n   STORE T TO REPEAT\nENDCASE\nENDIF\nIF COUNT = 3\nSTORE MATRIBUT TO SLCT3\nDO CASE\n   CASE !(SLCT3) = "A"\n      STORE "RANK" TO MATRI3\n      STORE "RANK" TO ATRI3\n   CASE !(SLCT3) = "B"\n      STORE "PRIMARY MOS" TO MATRI3\n      STORE "PRIMEMOS" TO ATRI3\n   CASE !(SLCT3) = "C"\n      STORE "SECONDARY MOS" TO MATRI3\n      STORE "SECHMS" TO ATRI3\n   CASE !(SLCT3) = "D"\n      STORE "COMPANY" TO MATRI3\n      STORE "COMPANY" TO ATRI3\n   CASE !(SLCT3) = "E"\n      STORE "PLATCON" TO MATRI3\n      STORE "PLATCON" TO ATRI3\n   CASE !(SLCT3) = "F"\n      STORE "JOINATE" TO MATRI3\n      STORE "JOINATE" TO ATRI3\n   CASE !(SLCT3) = "G"\n      STORE "EAS" TO MATRI3\n      STORE "EAS" TO ATRI3\n   CASE !(SLCT3) = "H"\n      STORE "BIRTH DATE" TO MATRI3\n      STORE "BIRTHATE" TO ATRI3\n   CASE !(SLCT3) = "I"\n      STORE "WEIGHT CONTROL" TO MATRI3\n      STORE "WTCONT" TO ATRI3\n   CASE !(SLCT3) = "J"\n      STORE "PTT ONE" TO MATRI3\n      STORE "PTT1CLSS" TO ATRI3\n   CASE !(SLCT3) = "K"\n      STORE "PTT TWO" TO MATRI3\n      STORE "PTT2CLSS" TO ATRI3\n   CASE !(SLCT3) = "L"\n      STORE "SWIM QUALIFICATION" TO MATRI3\n      STORE "SWIMQUAL" TO ATRI3\n   CASE !(SLCT3) = "M"\n      STORE "RIFLE QUALIFICATION" TO MATRI3\n      STORE "RFLQUAL" TO ATRI3\n   CASE !(SLCT3) = "N"\n      STORE "PISTOL QUALIFICATION" TO MATRI3\n      STORE "PSTQUAL" TO ATRI3\n   OTHERWISE\n      STORE T TO REPEAT\nENDCASE\nENDIF\nDO SLCTDESC\nENDDOC
RETURN
**D. SELECT DESCRIPTOR ROUTINE**

* ROUTINE NAME: SLCTDESC.PRG
* MODULE NAME: NONSTANDARD REPORT GENERATOR
* VERSION: 4.2.1.0
* AUTHOR: R.E. PRUIETT
* DATE: 14Feb84
* VARIABLES USED: COUNT
* VARIABLES MODIFIED: DESC1, DESC2, DESC3
* VARIABLES CREATED: MRANK, MSOCOS, MCOMPANY, MPLATCON,
  MJOINTE, MEAS, MTCNTL, MPROLE, MPFTONE, MSWQUAL,
  MRIFQAL, MPRISQUAL, MRANK2, MPRIMCOS2, MCOMPANY2,
  MPLAT2, MJOINTE2, MEAS2, MTCNTL2, MPROLE2,
  MPFTONE2, MIFQAL2, MSW QUAL2, MPRANK3, MPIMCOS3,
  MCOMPANY3, MPLAT3, MJOINTE3, MEAS3, MTCNTL3,
  MPROLE3, MPFTONE3, MSW QUAL3, MRIFQAL3,
* VARIABLES RELEASED: NONE
* FILES OPENED/CLOSED: NONE
* TFE+ FILES CREATED: NONE
* USING SUBROUTINES: SLCTTATRI.PRG (4.2.0)
* DESCRIPTION: THIS ROUTINE PECMPTS THE USER AFTER EACH REQUEST FOR A NEW ATTRIBUTE TO SELECT THE DESCRIPTION FOR THE CORRESPONDING ATTRIBUTE. FOR EXAMPLE, IF THE USER HAS SELECTED THE ATTRIBUTE COMPANY, THIS ROUTINE WILL ASK WHICH COMPANY AND PROVIDES A RANGE/EXAMPLE OF APPROPRIATE DESCRIPTIONS (E.G., A CO, B CO, ETC).
* ERASE IF COUNT = 1
DO CASE
CASE 1(SLCT1) = "A"
  STORE "0" TO MRANK
  @8.10 SAY "TYPE IN DESIRED RANK DESCRIPTION";
  @8.48 GET MRANK PICTURE 'XX'
  STORE MRANK TO DESC1
CASE 1(SLCT1) = "B"
  STORE "0" TO MPRIMCS
  @8.10 SAY "TYPE IN DESIRED PRIMARY MOS DESCRIPTION";
  @8.58 GET MPRIMCS PICTURE 'XX
  STORE MPRIMCS TO DESC1
CASE 1(SLCT1) = "C"
  STORE "0" TO MSECOS
  @8.10 SAY "TYPE IN DESIRED SECONDARY MOS DESCRIPTION";
  @8.60 GET MSECOS PICTURE 'XX
  STORE MSECOS TO DESC1
CASE 1(SLCT1) = "D"
  STORE "0" TO MCOMPANY
  @8.10 SAY "TYPE IN DESIRED COMPANY DESCRIPTION (A":
  @8.58 GET MCOMPANY PICTURE 'XXXXX'
  STORE MCOMPANY TO DESC1
CASE 1(SLCT1) = "E"
  STORE "0" TO MPLATCON
  @8.10 SAY "TYPE IN DESIRED PLatoon DESCRIPTION (1ST":
  @8.60 GET MPLATCON PICTURE 'XXXXXXXX'
  STORE MPLATCON TO DESC1
CASE 1(SLCT1) = "F"
  STORE "0" TO MJOINTE
  @8.10 SAY "TYPE IN DESIRED Jfoin Date DESCRIPTION";
"(220183,0183,83,ETC.)"  
@ 664 GET MEAS PICTURE 'XXXXXXXX'  
READ  
STORE MEAS TO DESC1  
CASE !(SICT1) = "G"  
STORE ' ' TO MEAS  
@ 8,0 SAY "TYPE IN DESIRED EAS DESCRIPTION";  
"(220183,0183,83,ETC.)"  
@ 8,55 GET MEAS PICTURE 'XXXXXXXX'  
READ  
STORE MEAS TO DESC1  
CASE !(SICT1) = "H"  
STORE ' ' TO MBIRDATE  
@ 8,0 SAY "TYPE IN DESIRED BIRTH DATE DESCRIPTION";  
"(220183,0183,83,ETC.)"  
@ 8,55 GET MBIRDATE PICTURE 'XXXXXXXX'  
READ  
STORE MBIRDATE TO DESC1  
CASE !(SICT1) = "I"  
STORE ' ' TO WTCTRL  
@ 8,0 SAY "TYPE IN DESIRED WEIGHT CONTROL DESCRIPTION";  
"(YES CR NO)"  
@ 8,55 GET WTCTRL PICTURE 'XXXXXXXX'  
READ  
STORE WTCTRL TO DESC1  
CASE !(SICT1) = "J"  
STORE ' ' TO MPFONE  
@ 8,0 SAY "TYPE IN DESIRED PFT ONE DESCRIPTION";  
"(5,1,2,3,F.*)"  
@ 8,55 GET MPFONE PICTURE 'X'  
READ  
STORE MPFONE TO DESC1  
CASE !(SICT1) = "K"  
STORE ' ' TO MFPTWO  
@ 8,0 SAY "TYPE IN DESIRED PFT TWO DESCRIPTION";  
"(5,1,2,3,F.*)"  
@ 8,55 GET MFPTWO PICTURE 'X'  
READ  
STORE MFPTWO TO DESC1  
CASE !(SICT1) = "L"  
STORE ' ' TO MSWQUAL  
@ 8,0 SAY "TYPE IN DESIRED SWIMMING QUALIFICATION DESCRIPTION";  
"(YES CR NO,*)"  
@ 9,17 GET MSWQUAL PICTURE 'XX'  
READ  
STORE MSWQUAL TO DESC1  
CASE !(SICT1) = "M"  
STORE ' ' TO MRFQUAL  
@ 8,0 SAY "TYPE IN DESIRED RIFLE QUALIFICATION DESCRIPTION";  
"(YES CR NO,*)"  
@ 9,18 GET MRFQUAL PICTURE 'XX'  
READ  
STORE MRFQUAL TO DESC1  
CASE !(SICT1) = "N"  
STORE ' ' TO MFSQUAL  
@ 8,0 SAY "TYPE IN DESIRED PISTOL QUALIFICATION DESCRIPTION";  
"(YES CR NO,*)"  
@ 9,18 GET MFSQUAL PICTURE 'XX'  
READ  
STORE MFSQUAL TO DESC1  
ENDCASE  
ENDIF  
IF COUNT = 2  
ENDCASE  
CASE !(SICT2) = "A"  
STORE ' ' TO MFRANK2  
105
a 8.0 SAY "TYPE IN DESIRED RANK DESCRIPTION:
(ET, E5, E2, ETC.)"
a 8.58 GET RANK2 PICTURE 'XX'
REAL
STORE MRANK2 TO DESC2
CASE 1(SLCT2) = "F"
STORE TC PRIMCS2
a 8.0 SAY "TYPE IN DESIRED PRIMARY MOS DESCRIPTION:
(03, 11, E31, ETC.)"
@ 8.58 GET PRIMCS2 PICTURE 'XXXX'
REAL
STORE PRIMCS2 TO DESC2
CASE 1(SLCT2) = "C"
STORE TC SEC MCS2
a 8.0 SAY "TYPE IN DESIRED SECONDARY MOS DESCRIPTION:
(03, 11, 031, ETC.)"
@ 8.58 GET SEC MCS2 PICTURE 'XXXX'
REAL
STORE SEC MCS2 TO DESC2
CASE 1(SLCT2) = "I"
STORE TC COMPNY2
a 8.0 SAY "TYPE IN DESIRED COMPANY DESCRIPTION (A;
AUTHENS CO, HQS CC)"
@ 8.58 GET COMPNY2 PICTURE 'XXXXXX'
REAL
STORE COMPNY2 TO DESC2
CASE 1(SLCT2) = "1"
STORE TC MELAT2
a 8.0 SAY "TYPE IN DESIRED PLATOON DESCRIPTION (1ST;
PLATOON, ETC.)"
@ 8.58 GET MELAT2 PICTURE 'XXXXXXXX'
REAL
STORE MELAT2 TO DESC2
CASE 1(SLCT2) = "E"
STORE TC JNDATE2
@ 8.0 SAY "TYPE IN DESIRED JOIN DATE DESCRIPTION:
(22, 0183, 0183, ETC.)"
@ 8.58 GET JNDATE2 PICTURE 'XXXXXX'
REAL
STORE JNDATE2 TO DESC2
CASE 1(SLCT2) = "E"
STORE TC MEAS2
@ 8.0 SAY "TYPE IN DESIRED EAS DESCRIPTION:
(22, 0183, 0183, ETC.)"
@ 8.58 GET MEAS2 PICTURE 'XXXXXX'
REAL
STORE MEAS2 TO DESC2
CASE 1(SLCT2) = "F"
STORE TC MBIRDATE2
@ 8.0 SAY "TYPE IN DESIRED BIRTH DATE DESCRIPTION:
(22, 0183, 0183, ETC.)"
@ 8.58 GET MBIRDATE2 PICTURE 'XXXXXX'
REAL
STORE MBIRDATE2 TO DESC2
CASE 1(SLCT2) = "I"
STORE "TO EWT CL2"
@ 8.0 SAY "TYPE IN DESIRED WEIGHT CONTROL DESCRIPTION:
(36, cr NO)"
@ 8.58 GET EWT CL2 PICTURE 'XXX'
REAL
STORE EWT CL2 TO DESC2
CASE 1(SLCT2) = "O"
STORE "TO MPFTN2"
@ 8.0 SAY "TYPE IN DESIRED PFT ONE DESCRIPTION:
(36, cr NO)"
@ 8.58 GET MPFTN2 PICTURE 'X'
REAL
STORE MPFTN2 TO DESC2
CASE 1(SLCT2) = "X"
106
STORE ! TC MPFTTWO2
@ 8.0 SAY "TYPE IN DESIRED PFT TWO DESCRIPTION;"
@ 8.50 GET MPFTTWO2 PICTURE 'X'
READ
STORE MPFTTWO2 TC DESC2
CASE 1 (SICT2) = "1"
STORE ! TO MSFWMT2L2
@ 8.0 SAY "TYPE IN DESIRED SWIMMING QUALIFICATION;"
@ 9.0 SAY "(S1, S2, S3, UN, **)"
@ 9.17 GET MSFWMT2L2 PICTURE 'XX'
READ
STORE MSFWMT2L2 TC DESC2
CASE 1 (SICT2) = "E"
STORE ! TO MRFQAL2
@ 8.0 SAY "TYPE IN DESIRED RIFLE QUALIFICATION;"
@ 9.0 SAY "(EX, SS, MM, UN, **)"
@ 9.18 GET MRFQAL2 PICTURE 'XX'
READ
STORE MRFQAL2 TC DESC2
CASE 1 (SICT2) = "P"
STORE ! TO MPTSQL2
@ 8.0 SAY "TYPE IN DESIRED PISTOL QUALIFICATION;"
@ 9.0 SAY "(EX, SS, MM, UN, **)"
@ 9.18 GET MPTSQL2 PICTURE 'XX'
READ
STORE MPTSQL2 TC DESC2
CASE 1 (SICT3) = "A"
STORE ! TO MFPNK3
@ 8.0 SAY "TYPE IN DESIRED RANK DESCRIPTION;"
@ 8.10 GET MFPNK3 PICTURE 'XX'
READ
STORE MFPNK3 TO LESC3
CASE 1 (SICT3) = "E"
STORE ! TO MPRIIMCS3
@ 8.0 SAY "TYPE IN DESIRED PRIMARY MOS DESCRIPTION;"
@ 8.10 GET MPRIIMCS3 PICTURE 'XXX'
READ
STORE MPRIIMCS3 TC DESC3
CASE 1 (SICT3) = "C"
STORE ! TO MSECMS3
@ 8.0 SAY "TYPE IN DESIRED SECONDARY MOS DESCRIPTION;"
@ 8.10 GET MSECMS3 PICTURE 'XXX'
READ
STORE MSECMS3 TC DESC3
CASE 1 (SICT3) = "L"
STORE ! TO MCOMPNY3
@ 8.0 SAY "TYPE IN DESIRED COMPANY DESCRIPTION (A;"
@ 8.59 GET MCOMPNY3 PICTURE 'XXXXX'
READ
STORE MCOMPNY3 TC DESC3
CASE 1 (SICT3) = "P"
STORE ! TO MLPALT3
@ 8.0 SAY "TYPE IN DESIRED PLATOON DESCRIPTION (1ST;
@ 8.60 GET MLPALT3 PICTURE 'XXXXXXXXXXX'
READ
STORE MLPALT3 TO LESC3

107
CASE (SLCT3) = "F"
  STORE 'TC MJDTE3
  @ 8,0 SAY "TYPE IN DESIRED JOIN DATE DESCRIPTION;"
  (220163,0183,83,ETC.)"
  & 8,54 GET MJDTE3 PICTURE 'XXXXXX'
  REAL
  STORE MJDTE3 TC DESC3
CASE (SLCT3) = "G"
  STORE 'TO MEAS3
  @ 8,0 SAY "TYPE IN DESIRED EAS DESCRIPTION;"
  (220183,0183,83,ETC.)"
  & 8,55 GET MEAS3 PICTURE 'XXXXXX'
  REAL
  STORE MEAS3 TC DESC3
CASE (SLCT3) = "H"
  STORE 'TC MBIRDTE3
  @ 8,0 SAY "TYPE IN DESIRED BIRTH DATE DESCRIPTION;"
  (220183,0183,83,ETC.)"
  & 8,50 GET MBIRDTE3 PICTURE 'XXXXXX'
  REAL
  STORE MBIRDTE3 TC DESC3
CASE (SLCT3) = "I"
  STORE 'TO MWTCKN13
  @ 8,0 SAY "TYPE IN DESIRED WEIGHT CONTROL DESCRIPTION;"
  (XXX,CR,NO)"
  & 8,55 GET MWTCKN13 PICTURE 'XXX'
  REAL
  STORE MWTCKN13 TC DESC3
CASE (SLCT3) = "J"
  STORE 'TO MPFTCNE3
  @ 8,0 SAY "TYPE IN DESIRED PFT ONE DESCRIPTION;"
  (S,1,2,3,F, *)"
  & 8,50 GET MPFTCNE3 PICTURE 'X'
  REAL
  STORE MPFTCNE3 TC DESC3
CASE (SLCT3) = "K"
  STORE 'TC MPFTWO3
  @ 8,0 SAY "TYPE IN DESIRED PFT TWO DESCRIPTION;"
  (S,1,2,3,F, *)"
  & 8,50 GET MPFTWO3 PICTURE 'X'
  REAL
  STORE MPFTWO3 TC DESC3
CASE (SLCT3) = "L"
  STORE 'TO MSWMQL3
  @ 8,0 SAY "TYPE IN DESIRED SWIMMING QUALIFICATION;"
  DESCRIPTION"
  @ 9,0 SAY "(S1,S2,S3,UN,**)"
  & 9,17 GET MSWMQL3 PICTURE 'XX'
  REAL
  STORE MSWMQL3 TC DESC3
CASE (SLCT3) = "M"
  STORE 'TO MRIFQL3
  @ 8,0 SAY "TYPE IN DESIRED RIFLE QUALIFICATION;"
  DESCRIPTION"
  @ 9,0 SAY "(EX,SS,MM,UN,**)"
  & 9,16 GET MRIFQL3 PICTURE 'XX'
  REAL
  STORE MRIFQL3 TO DESC3
CASE (SLCT3) = "N"
  STORE 'TO MPISQAL3
  @ 8,0 SAY "TYPE IN DESIRED PISTOL QUALIFICATION;"
  DESCRIPTION"
  @ 9,0 SAY "(EX,SS,MM,UN,**)"
  & 9,18 GET MPISQAL3 PICTURE 'XX'
  READ
  STORE MPISQAL3 TC DESC3
ENDCASE
ENDIF
RETURN

108
APPENDIX G
STANDARD COMMAND GENERATOR MODULE LISTINGS

A. STANDARD REPORT EXECUTIVE

* ROUTINE NAME: STDREPT.PRG
* MODULE NAME: STANDARD REPORT GENERATOR
* VERSION: 5.0
* AUTHOR: R.E. PBIETTI
* DATE: 6NOV83
* VARIABLES USED: NONE
* VARIABLES MODIFIED: NONE
* VARIABLES CREATED: AASTDOF, AADONE, PROCEED
* VARIABLES RELEASED: AASTDOF, AADONE, ALL EXCEPT A????
* FILES CLEARED/CLOSED: NONE
* TEMPE FILES CREATED: NONE
* USING SUBRoutines: COMMAND DETERMINER
* DESCRIPTION: GENERATES THE PRIMARY MENU FOR STANDARD
* REPORT SELECTION AND STORES THE USER'S RESPONSE IN
* AASTDOF. CONTROL IS THEN PASSED TO THE APPROPRIATE
* SUBROUTINE TO OBTAIN THE NECESSARY DATA FROM THE
* DATABASE. THIS ROUTINE WILL CONTINUE UNTIL THE USER
* RESECEDES WITH A "Q".

STORE P TO AADONE
DO WHILE .NOT. AADONE
ERASE
   1.  SAY " PRIMARY MENU FOR STANDARD REPORT SELECTION"
   2.  SAY " STANDARD REPORT SELECTION"
   3.  SAY " SELECT A REPORT BY ENTERING THE APPROPRIATE;"
   4.  SAY " "
   5.  SAY "A...UNIT ROSTER"
   6.  SAY "E...TRAINING STATUS REPORT"
   7.  SAY "C...TRAINING ROSTER REPORT"
   8.  SAY "D...INDIVIDUAL TRAINING RECORD"
   9.  SAY "F...MARKSMANSHIP STATUS REPORT"
  10. SAY "G...MARKSMANSHIP ROSTER REPORT"
  11. SAY "H...MOS STATUS REPORT"
  12. SAY "I...MOS ROSTER REPORT"
  13. SAY "J...PERSONAL DATA REPORT"
  14. SAY "K...EST STATUS REPORT"
  15. SAY "L...EST ROSTER REPORT"
  16. SAY "M...SWIM QUALIFICATION REPORT"
  17. SAY "N...HELP"
  18. SAY "O...QUIT"
  19. SAY "P...HELP"
  20. SAY "Q...QUIT"
  21. SAY "R...HELP"
ACCEP "SELECT OPTION ======> " TO AASTDOF

DO CASE
   CASE !(AASTDOF) = 'A'
      DC SLCTUNIT
      DC UNITROST
      ACCEP "PRESS 'ENTER' KEY TO CONTINUE" TO PROCEED
   CASE !(AASTDOF) = 'E'
      DC SLCTUNIT
      DO TRNGSTAT
   CASE !(AASTDOF) = 'C'
      DC SLCTUNIT
      DO TRNGRST
ENDCASE

109
ACCEH "PRESS 'ENTER' KEY TO CONTINUE" TO PROCEED

CASE ! (AASTDCPT) = 'E'
   ERASE
   DC VIEWCTR
CASE I (AASTDCPT) = 'E'
   DO SELCTUNIT
   DC BKSTSTAT
CASE I (AASTDCPT) = 'F'
   DC SELCTUNIT
   DO MSBOST
CASE I (AASTDCPT) = 'G'
   DO SELCTUNIT
   DO MOSSTAT
   @ 22,0
   ACCEPT "PRESS 'ENTER' KEY TO CONTINUE" TO PROCEED
CASE I (AASTDCPT) = 'H'
   DO SELCTUNIT
   DO MSBOST
   @ 22,0
   ACCEPT "PRESS 'ENTER' KEY TO CONTINUE" TO PROCEED
CASE I (AASTDCPT) = 'I'
   DO SELCTUNIT
   DC PERSDATA
   @ 22,0
   ACCEPT "PRESS 'ENTER' KEY TO CONTINUE" TO PROCEED
CASE I (AASTDCPT) = 'J'
   DO SELCTUNIT
   DC ESISTAT
CASE I (AASTDCPT) = 'K'
   DO SELCTUNIT
   DC ESIBOST
   @ 22,0
   ACCEPT "PRESS 'ENTER' KEY TO CONTINUE" TO PROCEED
CASE I (AASTDCPT) = 'L'
   DO SELCTUNIT
   DC SWIMQUAL
CASE I (AASTDCPT) = 'M'
   DO A:SELF
ENDCASE
IF ! (AASTDCPT) = 'O'
   STORE T TO AADONE
ENDIF
ENDDC
RELEASE AADONE, AASTDCPT
RELEASE ALL EXCEPT AA????
USE
RETURN
B. SELECT UNIT ROUTINE

* ROUTINE NAME: SLCTUNIT.PRG
* MODULE NAME: STANDARD REPORT GENERATOR
* VERSION: 5.1.0
* AUTHOR: P.E. RUIEITT
* DATE: NOV 83
* VARIABLES USED: ONECO, TWOCO, THREECO, FOURCO, ONEELT,
  TWENTY, THREEPLT, FOURPLT
* VARIABLES MODIFIED: NONE
* VARIABLES CREATED: MUNIT, MPLAT
* VARIABLES RELEASED: NONE
* FILES OPENED/CLOSED: NONE
* TEMP FILES CREATED: NONE
* USING PROGRAMS: STERT.PRG
* DESCRIPTION: THIS ROUTINE PROMPTS THE USER TO SELECT
  THE SIZE AND NAME OF THE UNIT (I.E., BATTALION,COMPANY,
  PLATOON) FOR THE REPORT GENERATION. THE USER RESPONDS
  WITH A LETTER THAT CORRESPONDS TO A USER INSTALLED CC OR
  PLT DESIGNATION. THIS ROUTINE WILL EXECUTE ONCE EACH
* TIME CALLED.

STORE F TO SLCTDONE
STORE F FROM A:UNITMEN ADDITIVE

DO WHILE .NOT. SLCTDONE
IF ! E A SAY "SELECT UNIT
IF ! E B SAY "A...
IF ! E C SAY "B...
IF ! E D SAY "C...
IF ! E E SAY "D...
IF ! E F SAY "E...
IF ! E G SAY "F...
IF ! E H SAY "H...
IF ! E I SAY "I...
IF ! E J SAY "J...
IF ! E K SAY "K...
IF ! E L SAY "L...
IF ! E M SAY "M...
IF ! E N SAY "N...
IF ! E O SAY "O...
IF ! E P SAY "P...
IF ! E Q SAY "Q...
IF ! E R SAY "R...
IF ! E S SAY "S...
IF ! E T SAY "T...
IF ! E U SAY "U...
IF ! E V SAY "V...
IF ! E W SAY "W...
IF ! E X SAY "X...
IF ! E Y SAY "Y...
IF ! E Z SAY "Z...
IF ! E A SAY "A...
IF ! E B SAY "B...
IF ! E C SAY "C...
IF ! E D SAY "D...
IF ! E E SAY "E...
IF ! E F SAY "F...
IF ! E G SAY "G...
IF ! E H SAY "H...
IF ! E I SAY "I...
IF ! E J SAY "J...
IF ! E K SAY "K...
IF ! E L SAY "L...
IF ! E M SAY "M...
IF ! E N SAY "N...
IF ! E O SAY "O...
IF ! E P SAY "P...
IF ! E Q SAY "Q...
IF ! E R SAY "R...
IF ! E S SAY "S...
IF ! E T SAY "T...
IF ! E U SAY "U...
IF ! E V SAY "V...
IF ! E W SAY "W...
IF ! E X SAY "X...
IF ! E Y SAY "Y...
IF ! E Z SAY "Z...
IF ! E A SAY "A...
IF ! E B SAY "B...
IF ! E C SAY "C...
IF ! E D SAY "D...
IF ! E E SAY "E...
IF ! E F SAY "F...
IF ! E G SAY "G...
IF ! E H SAY "H...
IF ! E I SAY "I...
IF ! E J SAY "J...
IF ! E K SAY "K...
IF ! E L SAY "L...
IF ! E M SAY "M...
IF ! E N SAY "N...
IF ! E O SAY "O...
IF ! E P SAY "P...
IF ! E Q SAY "Q...
IF ! E R SAY "R...
IF ! E S SAY "S...
IF ! E T SAY "T...
IF ! E U SAY "U...
IF ! E V SAY "V...
IF ! E W SAY "W...
IF ! E X SAY "X...
IF ! E Y SAY "Y...
IF ! E Z SAY "Z...
ENDF

STORE F TO SLCTDONE
DO WHILE .NOT. SLCTDONE
AND ! (MUNIT) < 'E'
IF ! (MUNIT) = 'A' .OR. ! (MUNIT) = 'B' .OR. ! (MUNIT) = 'C' .OR.
  ! (MUNIT) = 'D' .OR. ! (MUNIT) = 'E'
STORE F TO SLCTDONE
ENDF

ENDT

STORE F TO SLCTDONE
DO WHILE .NOT. SLCTDONE
AND ! (MUNIT) < 'E'
IF ! (MUNIT) = 'A' .OR. ! (MUNIT) = 'B' .OR. ! (MUNIT) = 'C' .OR.
  ! (MUNIT) = 'D' .OR. ! (MUNIT) = 'E'
STORE F TO SLCTDONE

111
ENDIP
ENDDC
IF ! (MUNIT) = 'E'
    STORE 'Z' TO MPLAT
ENDIP
RELEASE SLCTDONE
USE
RETURN
C. UNIT ROSTER ROUTINE

* ROUTINE NAME: UNITFCST.PRG
* MODULE NAME: STANDARD REPORT GENERATOR
* VERSION: 5.2.0
* AUTHOR: R. E. FRIEIT
* DATE: 6NOV83
* VARIABLES USED: MUNIT, MPLAT, CNECO, TWO, THREECO, FOURCO, CNEP1T, THREEP1T, FOURP1T
* VARIABLES MODIFIED: NONE
* VARIABLES CREATED: NONE
* VARIABLES RELEASED: MUNIT, MPLAT
* FILES OPENED/CLOSED: A: FERS INDEX A: ALPHPERS
* FILES CREATED: NONE
* USING SUBROUTINES: SDFRPT.PRG

* DESCRIPTION: THIS IS A CONTROL ROUTINE WHICH CREATES THE PERSONNEL ROSTER FOR THE UNIT SPECIFIED BY THE USER'S RESPONSE TO THE SLCUNIT MENU (5.1.0).

USE A: FERS INDEX A: ALPHPERS
DO CASE
CASE I(MUNIT) = 'F'
  REPORT FORM B: FCSSTRPT
CASE I(MUNIT) = 'A' .AND. I(MUNIT) = 'A'
  REPORT FORM B: FCSSTRPT FOR COMPANY = ONECO
  FIALTCN = ONEP1T
CASE I(MUNIT) = 'A' .AND. I(MELAT) = 'B'
  REPORT FORM B: FCSSTRPT FOR COMPANY = ONECO .AND.
  FIALTCN = TWOPLT
CASE I(MUNIT) = 'A' .AND. I(MELAT) = 'C'
  REPORT FORM B: FCSSTRPT FOR COMPANY = ONECO .AND.
  FIALTCN = THREEP1T
CASE I(MUNIT) = 'A' .AND. I(MELAT) = 'D'
  REPORT FORM B: FCSSTRPT FOR COMPANY = ONECO .AND.
  FIALTCN = FOURP1T
CASE I(MUNIT) = 'E' .AND. I(MELAT) = 'E'
  REPORT FORM B: FCSSTRPT FOR COMPANY = TWOCO
CASE I(MUNIT) = 'E' .AND. I(MELAT) = 'A'
  REPORT FORM B: FCSSTRPT FOR COMPANY = TWOCO .AND.
  FIALTCN = ONEP1T
CASE I(MUNIT) = 'E' .AND. I(MELAT) = 'C'
  REPORT FORM B: FCSSTRPT FOR COMPANY = TWOCO .AND.
  FIALTCN = THREEP1T
CASE I(MUNIT) = 'E' .AND. I(MELAT) = 'D'
  REPORT FORM B: FCSSTRPT FOR COMPANY = TWOCO .AND.
  FIALTCN = FOURP1T
CASE I(MUNIT) = 'E' .AND. I(MELAT) = 'C'
  REPORT FORM B: FCSSTRPT FOR COMPANY = THREECO
CASE I(MUNIT) = 'E' .AND. I(MELAT) = 'A'
  REPORT FORM B: FCSSTRPT FOR COMPANY = THREECO .AND.
  FIALTCN = ONEP1T
CASE I(MUNIT) = 'E' .AND. I(MELAT) = 'B'
  REPORT FORM B: FCSSTRPT FOR COMPANY = THREECO .AND.
  FIALTCN = TWOPLT
CASE I(MUNIT) = 'E' .AND. I(MELAT) = 'C'
  REPORT FORM B: FCSSTRPT FOR COMPANY = THREECO .AND.
  FIALTCN = THREEP1T
CASE I(MUNIT) = 'E' .AND. I(MELAT) = 'D'
  REPORT FORM B: FCSSTRPT FOR COMPANY = THREECO .AND.
  FIALTCN = FOURP1T
CASE I(MUNIT) = 'E' .AND. I(MELAT) = 'A'
  REPORT FORM B: FCSSTRPT FOR COMPANY = FOURCO
  FIALTCN = FOURP1T
CASE !(MUNIT) = 'E' .AND. !(MELAT) = 'A'
  EFFORT FORM B: ECSTRFT FOR COMPANY = FOURCO .AND.;
  FLATCEN = ONEPIT
CASE !(MUNIT) = 'E' .AND. !(MELAT) = 'B'
  EFFORT FORM B: ECSTRFT FOR COMPANY = FOURCO .AND.;
  FLATCEN = TWOPIT
CASE !(MUNIT) = 'E' .AND. !(MELAT) = 'C'
  EFFORT FORM B: ECSTRFT FOR COMPANY = FOURCO .AND.;
  FLATCEN = THREEPIT
CASE !(MUNIT) = 'E' .AND. !(MELAT) = 'D'
  EFFORT FORM B: ECSTRFT FOR COMPANY = FOURCO .AND.;
  FLATCEN = FOURPIT
ENDCASE
USE
RELEASE MUNIT, MELAT
RETURN
**D. TRAINING STATISTICS ROUTINE**

* **ROUTINE NAME:** TRNGSTAT.PRG  
* **MODULE NAME:** STANDARD REPORT GENERATOR  
* **VERSION:** 5.3.0  
* **AUTHOR:** R. E. PROUTIT  
* **DATE:** 22DEC83

**VARIABLES USED:**
- UNIT  
- MPLAT, ZZACOC, ZZTOTAL, ZZBCOC,  
- ZZCCOC, ZZAHIS, ZZCHIS, ZZACOD, ZZBCOD, ZZCCOC,  
- ZZINT, ZZBTINT, ZZLINT, ZZBINT, ZZCINT, ZZCATD, ZZCATD, ZZLAID, ZZLAI,  
- ZZAI, ZZFAID, ZZGAID, ZZHAID, ZZJAID, ZZKAID, ZZKAI,  
- PROCEFL, ZZAIN, ZZUNI, ZZHIS, ZZAIN, ZZCUNI, ZZCUNI, ZZANBU, ZZBNUC, ZZBNBC, ZZCNBU,  
- ZZDNBC, ZZSBNBC, ZZFBBC, ZZAMKS, ZZBAMKS, ZZCAMS, ZZAMRS,  
- ZZEMRS, ZZEPRRS, ZZATTAC, ZZBTAC, ZZCTAC, ZZDTAC, ZZETAC,  
- ZZPFT1, ZZPFT2, ZZSWM, ZZRFL, ZZPST, ZZNAPST, ZZNARFL  

**VARIABLES MODIFIED:** None  
**VARIABLES RELEASED:** ZZ??????  
**FILES (OPENED/CLOSED):** EST, PERS, QUAL (All FILES CLOSED)  
**TEMP FILES CREATED:** TEMP  
**USING SUBROUTINES:** STDRPRT.PRG  
**DESCRIPTION:** For the selected unit, this routine calculates the total number of training elements completed for each training category. For example, the training category "history" consists of three training elements: (ex. ZZAHIS, ZZCHIS, ZZCSIS) by the total number of personnel in the unit (ZZTOTAL). Both the absolute totals for each element and the percentage completed figures are shown.

ERASE  
DO CASE  
CASE 1(MUNIT) = 'E'  
  USE B:EST  
  DO CNTRNG  
CASE 1(MUNIT) = 'A'.AND. 1(MPLAT) = 'F'  
  SELECT PRIMARY  
  USE A:PERS  
  SELECT SECONDARY  
  USE B:EST  
  SELECT PRIMARY  
  JOIN TO TEMP FCF (F.COMPANY = ONECO .AND. P.SSN= S.SSN)  
  FIELDS COC, HIS, CODE, INT.AID, UNI, PFT1.CLS, PFT2.CLS, NBC, MRS, TAC  
  USE TEMP  
  DO CNTRNG  
  DELETE FILE TEMP  
CASE 1(MUNIT) = 'F'.AND. 1(MPLAT) = 'A'  
  SELECT PRIMARY  
  USE A:PERS  
  SELECT SECONDARY  
  USE B:EST  
  SELECT PRIMARY  
  JOIN TO TEMP FCF (F.COMPANY = ONECO .AND. P.PLATCON =;  
  FIELDS COC, HIS, CODE, INT.AID, UNI, PFT1.CLS, PFT2.CLS, NBC, MRS, TAC  
  USE TEMP  
  DO CNTRNG  
  DELETE FILE TEMP  
CASE 1(MUNIT) = 'A'.AND. 1(MPLAT) = 'B'  
  SELECT PRIMARY  
  USE A:PERS  
  SELECT SECONDARY  
  USE B:EST  

115
SELECT PRIMARY JOIN TO TEMP FCf (P.COMPANY = ONECO . AND. P.PLATCON =; T=CFIL. AND. P.SSN=S.SSN) FIELDS: COC, HIS, COD, INT, AID, Uni, PFT1CLASS,; PFT2CLASS, NBC, MFS, TAC USE TEMP DO CONTINUE DELETE FILE TEMP CASE I (MUNIT) = 'A' . AND. (MPLAT) = 'C'
SELECT PRIMARY USE A: PERS SELECT SECCNDAFY USE B: EST USE PRIMARY JOIN TO TEMP FCf (P.COMPANY = ONECO . AND. P.PLATCON =; T=CFIL. AND. P.SSN=S.SSN) FIELDS: COC, HIS, COD, INT, AID, Uni, PFT1CLASS,; PFT2CLASS, NBC, MFS, TAC USE TEMP DO CONTINUE DELETE FILE TEMP CASE I (MUNIT) = 'A' . AND. (MPLAT) = 'D'
SELECT PRIMARY USE A: PERS SELECT SECCNDAFY USE B: EST USE PRIMARY JOIN TO TEMP FCf (P.COMPANY = ONECO . AND. P.PLATCON =; T=CFIL. AND. P.SSN=S.SSN) FIELDS COC, HIS, COD, INT, AID, Uni, PFT1CLASS,; PFT2CLASS, NBC, MFS, TAC USE TEMP DO CONTINUE DELETE FILE TEMP CASE I (MUNIT) = 'E' . AND. (MPLAT) = 'E'
SELECT PRIMARY USE A: PERS SELECT SECCNDAFY USE B: EST USE PRIMARY JOIN TO TEMP FCf (P.COMPANY = ONECO . AND. P.PLATCON =; T=CFIL. AND. P.SSN=S.SSN) FIELDS COC, HIS, COD, INT, AID, Uni, PFT1CLASS,; PFT2CLASS, NBC, MFS, TAC USE TEMP DO CONTINUE DELETE FILE TEMP CASE I (MUNIT) = 'E' . AND. (MPLAT) = 'E'
SELECT PRIMARY USE A: PERS SELECT SECCNDAFY USE B: EST USE PRIMARY JOIN TO TEMP FCf (P.COMPANY = ONECO . AND. P.PLATCON =; T=CFIL. AND. P.SSN=S.SSN) FIELDS COC, HIS, COD, INT, AID, Uni, PFT1CLASS,; PFT2CLASS, NBC, MFS, TAC USE TEMP DO CONTINUE DELETE FILE TEMP CASE I (MUNIT) = 'E' . AND. (MPLAT) = 'E'
SELECT PRIMARY USE A: PERS SELECT SECCNDAFY USE B: EST USE PRIMARY JOIN TO TEMP FCf (P.COMPANY = ONECO . AND. P.PLATCON =; T=CFIL. AND. P.SSN=S.SSN) FIELDS COC, HIS, COD, INT, AID, Uni, PFT1CLASS,; PFT2CLASS, NBC, MFS, TAC USE TEMP DO CONTINUE DELETE FILE TEMP CASE I (MUNIT) = 'E' . AND. (MPLAT) = 'E'
SELECT PRIMARY USE A: PERS SELECT SECCNDAFY USE B: EST USE PRIMARY JOIN TO TEMP FCf (P.COMPANY = ONECO . AND. P.PLATCON =; T=CFIL. AND. P.SSN=S.SSN) FIELDS COC, HIS, COD, INT, AID, Uni, PFT1CLASS,; PFT2CLASS, NBC, MFS, TAC USE TEMP DO CONTINUE DELETE FILE TEMP
Case 1. (UNIT) = 'E' . AND . !(MPLAT) = 'C'
   SELECT PRIMARY
   USE A:PERS
   SELECT SECCNDARY
   USE B:EST
   SELECT PRIMARY
   JOIN TO TEMP FCR (P.COMPANY = THRSEC0 . AND . P.PLATCON =
   THREETLT . AND . F.SSN = S.SSN) FIELDS COC, HIS, CED,;
   INT, AID, UNI, PFT1CLASS, PFT2CLASS, NBC, MKS, TAC
   USE TEMP
   DO CNTTRNG
   DELETE FILE TEMP
   Case 1. (UNIT) = 'E' . AND . !(MPLAT) = 'D'
   SELECT PRIMARY
   USE A:PERS
   SELECT SECCNDARY
   USE B:EST
   SELECT PRIMARY
   JOIN TO TEMP FCR (P.COMPANY = THRSEC0 . AND . P.PLATCON =
   = CHSETLT . AND . F.SSN = S.SSN) FIELDS COC, HIS, CED,;
   INT, AID, UNI, PFT1CLASS, PFT2CLASS, NBC, MKS, TAC
   USE TEMP
   DO CNTTRNG
   DELETE FILE TEMP
   Case 1. (UNIT) = 'C' . AND . !(MPLAT) = 'E'
   SELECT PRIMARY
   USE A:PERS
   SELECT SECCNDARY
   USE B:EST
   SELECT PRIMARY
   JOIN TO TEMP FCR (P.COMPANY = THRSEC0 . AND . P.PLATCON =
   = CNSETLT . AND . F.SSN = S.SSN) FIELDS COC, HIS, CED,;
   INT, AID, UNI, PFT1CLASS, PFT2CLASS, NBC, MKS, TAC
   USE TEMP
   DO CNTTRNG
   DELETE FILE TEMP
   Case 1. (UNIT) = 'C' . AND . !(MPLAT) = 'A'
   SELECT PRIMARY
   USE A:PERS
   SELECT SECCNDARY
   USE B:EST
   SELECT PRIMARY
   JOIN TO TEMP FCR (P.COMPANY = THRSEC0 . AND . P.PLATCON =
   = CNSETLT . AND . F.SSN = S.SSN) FIELDS COC, HIS, CED,;
   INT, AID, UNI, PFT1CLASS, PFT2CLASS, NBC, MKS, TAC
   USE TEMP
   DO CNTTRNG
   DELETE FILE TEMP
   Case 1. (UNIT) = 'C' . AND . !(MPLAT) = 'B'
   SELECT PRIMARY
   USE A:PERS
   SELECT SECCNDARY
   USE B:EST
   SELECT PRIMARY
   JOIN TO TEMP FCR (P.COMPANY = THRSEC0 . AND . P.PLATCON =
   = CNSETLT . AND . F.SSN = S.SSN) FIELDS COC, HIS, CED,;
   INT, AID, UNI, PFT1CLASS, PFT2CLASS, NBC, MKS, TAC
   USE TEMP
   DO CNTTRNG
   DELETE FILE TEMP
   Case 1. (UNIT) = 'C' . AND . !(MPLAT) = 'C'
   SELECT PRIMARY
   USE A:PERS
   SELECT SECCNDARY
   USE B:EST
   SELECT PRIMARY
   JOIN TO TEMP FCR (P.COMPANY = THRSEC0 . AND . P.PLATCON =
   = THREEPLT . AND . F.SSN = S.SSN) FIELDS COC, HIS, CED,;
CASE I (UNIT) = 'C'. AND. !(MPLAT) = 'D'
USE PRIMARY
SELECT SECONDARY
USE B:EST
SELECT PRIMARY
JOIN TO TEMP FCR (P.COMPANY = THREECO .AND. P.FLATCON;
= FOURPLT .AND. F.SSN=S.SSN) FIELDS C,CC,HIS,COD,;
INT,UNIT,PFT1CLSS,PFT2CLSS,NBC,MKS,TAC
USE TEMP
DELETE FILE TEMP
CASE I (UNIT) = 'L'. AND. !(MPLAT) = 'E'
SELECT PRIMARY
USE A:PERS
SELECT SECONDARY
USE B:EST
SELECT PRIMARY
JOIN TO TEMP FCR (P.COMPANY = FOURCO .AND. P.FLATCON;
= THREEPLT .AND. F.SSN=S.SSN) FIELDS C,CC,HIS,COD,;
INT,UNIT,PFT1CLSS,PFT2CLSS,NBC,MKS,TAC
USE TEMP
DELETE FILE TEMP
CASE I (UNIT) = 'L'. AND. !(MPLAT) = 'A'
SELECT PRIMARY
USE A:PERS
SELECT SECONDARY
USE B:EST
SELECT PRIMARY
JOIN TO TEMP FCR (P.COMPANY = FOURCO .AND. P.FLATCON;
= THREEPLT .AND. F.SSN=S.SSN) FIELDS C,CC,HIS,COD,;
INT,UNIT,PFT1CLSS,PFT2CLSS,NBC,MKS,TAC
USE TEMP
DELETE FILE TEMP
CASE I (UNIT) = 'L'. AND. !(MPLAT) = 'B'
SELECT PRIMARY
USE A:PERS
SELECT SECONDARY
USE B:EST
SELECT PRIMARY
JOIN TO TEMP FCR (P.COMPANY = FOURCO .AND. P.FLATCON;
= THREEPLT .AND. F.SSN=S.SSN) FIELDS C,CC,HIS,COD,;
INT,UNIT,PFT1CLSS,PFT2CLSS,NBC,MKS,TAC
USE TEMP
DELETE FILE TEMP
CASE I (UNIT) = 'L'. AND. !(MPLAT) = 'C'
SELECT PRIMARY
USE A:PERS
SELECT SECONDARY
USE B:EST
SELECT PRIMARY
JOIN TO TEMP FCR (P.COMPANY = FOURCO .AND. P.FLATCON;
= THREEPLT .AND. F.SSN=S.SSN) FIELDS C,CC,HIS,COD,;
INT,UNIT,PFT1CLSS,PFT2CLSS,NBC,MKS,TAC
USE TEMP
DELETE FILE TEMP
CASE I (UNIT) = 'L'. AND. !(MPLAT) = 'D'
SELECT PRIMARY
USE A:PERS
SELECT SECONDARY
USE BEST
SELECT PRIMARY
JOIN ON TEMP PCB (P.CCMIANY =FOUCO AND P.FLACTCN;
= FOURFL AND P.SSN=S.SSN) FIELDS:
CC, CSC, ID, INT, AID, UNI, PFTYCLSS;
PFTYCLSS, NUC, MFS, TAC
USE TEMP
DO CNTTRNG
DELETE PILE TEMP
ENDCASE
ERASE
0,25 SAY "TRAINING STATUS REPORT"
2,25 SAY "CODE OF CONDUCT..."
3,30 SAY "PERCENT COMPLETED A..."
2,52 SAY ZZACOC USING '999'
3,25 SAY (ZZBCOC/ZZTOTAL) *100 USING '999'
3,30 SAY "PERCENT COMPLETED B..."
3,52 SAY ZZBCOC USING '999'
4,30 SAY "PERCENT COMPLETED C..."
4,52 SAY ZZCCOC USING '999'
5,0 SAY "MARINE COFFS HISTORY..."
5,25 SAY (ZZAHIS/ZZTOTAL) *100 USING '999'
5,30 SAY "PERCENT COMPLETED A..."
5,52 SAY ZZAHIS USING '999'
6,30 SAY "PERCENT COMPLETED B..."
6,52 SAY ZZBHIS USING '999'
7,30 SAY "PERCENT COMPLETED C..."
7,52 SAY ZZCHIS USING '999'
8,0 SAY "CLOSE ORDER DRILL..."
8,25 SAY (ZZACOD/ZZTOTAL) *100 USING '999'
9,30 SAY "PERCENT COMPLETED A..."
9,52 SAY ZZACCD USING '999'
10,25 SAY (ZZBCCD/ZZTOTAL) *100 USING '999'
11,30 SAY "PERCENT COMPLETED B..."
11,52 SAY ZZBCCD USING '999'
12,30 SAY "PERCENT COMPLETED C..."
12,52 SAY ZZCCCD USING '999'
13,0 SAY "INTERIOR GUARD..."
13,25 SAY (ZZAINT/ZZTOTAL) *100 USING '999'
14,30 SAY "PERCENT COMPLETED A..."
14,52 SAY ZZAINT USING '999'
15,25 SAY (ZZBINT/ZZTOTAL) *100 USING '999'
16,30 SAY "PERCENT COMPLETED B..."
16,52 SAY ZZBINT USING '999'
17,30 SAY "PERCENT COMPLETED C..."
17,52 SAY ZZCINT USING '999'
18,0 SAY "FIRST AID..."
18,25 SAY (ZZAIAID/ZZTOTAL) *100 USING '999'
19,30 SAY "PERCENT COMPLETED A..."
19,52 SAY ZZAIAID USING '999'
20,25 SAY (ZZBIAID/ZZTOTAL) *100 USING '999'
21,30 SAY "PERCENT COMPLETED B..."
21,52 SAY ZZBIAID USING '999'
22,30 SAY "PERCENT COMPLETED C..."
22,52 SAY ZZCIAID USING '999'
23,0 SAY "CLOS E ORDER DRILL..."
23,25 SAY (ZZACOD/ZZTOTAL) *100 USING '999'
24,30 SAY "PERCENT COMPLETED A..."
24,52 SAY ZZACCD USING '999'
25,25 SAY (ZZBCCD/ZZTOTAL) *100 USING '999'
26,30 SAY "PERCENT COMPLETED B..."
26,52 SAY ZZBCCD USING '999'
27,30 SAY "PERCENT COMPLETED C..."
27,52 SAY ZZCCCD USING '999'
28,0 SAY "INTEBIOR GUARD..."
28,25 SAY (ZZAINT/ZZTOTAL) *100 USING '999'
29,30 SAY "PERCENT COMPLETED A..."
29,52 SAY ZZAINT USING '999'
30,25 SAY (ZZBINT/ZZTOTAL) *100 USING '999'
31,30 SAY "PERCENT COMPLETED B..."
31,52 SAY ZZBINT USING '999'
32,30 SAY "PERCENT COMPLETED C..."
32,52 SAY ZZCINT USING '999'
33,0 SAY "FIRST AID..."
33,25 SAY (ZZAIAID/ZZTOTAL) *100 USING '999'
34,30 SAY "PERCENT COMPLETED A..."
34,52 SAY ZZAIAID USING '999'
35,25 SAY (ZZBIAID/ZZTOTAL) *100 USING '999'
36,30 SAY "PERCENT COMPLETED B..."
36,52 SAY ZZBIAID USING '999'
37,30 SAY "PERCENT COMPLETED C..."
37,52 SAY ZZCIAID USING '999'
38,0 SAY "CLOS E ORDER DRILL..."
38,25 SAY (ZZACOD/ZZTOTAL) *100 USING '999'
39,30 SAY "PERCENT COMPLETED A..."
40,52 SAY ZZACCD USING '999'
41,25 SAY (ZZBCCD/ZZTOTAL) *100 USING '999'
42,30 SAY "PERCENT COMPLETED B..."
42,52 SAY ZZBCCD USING '999'
43,30 SAY "PERCENT COMPLETED C..."
43,52 SAY ZZCCCD USING '999'
44,0 SAY "INTEBIOR GUARD..."
44,25 SAY (ZZAINT/ZZTOTAL) *100 USING '999'
45,30 SAY "PERCENT COMPLETED A..."
45,52 SAY ZZAINT USING '999'
46,25 SAY (ZZBINT/ZZTOTAL) *100 USING '999'
47,30 SAY "PERCENT COMPLETED B..."
47,52 SAY ZZBINT USING '999'
48,30 SAY "PERCENT COMPLETED C..."
48,52 SAY ZZCINT USING '999'
49,0 SAY "FIRST AID..."
49,25 SAY (ZZAIAID/ZZTOTAL) *100 USING '999'
50,30 SAY "PERCENT COMPLETED A..."
50,52 SAY ZZAIAID USING '999'
51,25 SAY (ZZBIAID/ZZTOTAL) *100 USING '999'
52,30 SAY "PERCENT COMPLETED B..."
52,52 SAY ZZBIAID USING '999'
20.54 SAY "PERCENT COMPLETED F...
20.76 SAY ZZFAID USING '999'
21.20 SAY (ZZLAI D/ZZTOTAL)*100 USING '999'
21.46 SAY "PERCENT COMPLETED G...
21.50 SAY ZZHAID USING '999'
21.76 SAY ZZFAID USING '999'
22.20 SAY (ZZLAI D/ZZTOTAL)*100 USING '999'
22.46 SAY "PERCENT COMPLETED I...
22.50 SAY ZZHAID USING '999'
22.76 SAY ZZFAID USING '999'
23.20 SAY (ZZLAI D/ZZTOTAL)*100 USING '999'
23.46 SAY ZZKAI D USING '999'
ACCESS "PRESS 'ENTER' TO CONTINUE" TO PROCEED

0.0 SAY "TRAINING STATUS REPORT"
0.5 SAY "EQUIPMENT/UNIFORMS...
0.5 SAY (ZZUNI/ZZTOTAL)*100 USING '999'
0.5 SAY "PERCENT COMPLETED A...
0.5 SAY ZZUNI USING '999'
2.25 SAY ZZBUNI USING '999'
3.25 SAY ZZCUNI USING '999'
4.25 SAY ZZDUNI USING '999'
5.25 SAY ZZFUNI USING '999'
6.25 SAY (ZZANEC/ZZTOTAL)*100 USING '999'
6.25 SAY "PERCENT COMPLETED A...
6.25 SAY ZZANEC USING '999'
7.25 SAY ZZBNEC USING '999'
8.25 SAY ZZCNEC USING '999'
9.25 SAY ZZDNEC USING '999'
10.25 SAY ZZFNEC USING '999'
11.25 SAY ZZAMKS USING '999'
12.25 SAY ZZBMKS USING '999'
13.25 SAY ZZCMKS USING '999'
14.25 SAY ZZDMKS USING '999'
15.25 SAY ZZEMKS USING '999'
16.25 SAY ZZFMKS USING '999'
ACCEPT "FREES 'ENTER' TO CONTINUE" TO PROCEED

12.25 SAY "TRAINING STATUS REPORT"
2.0 SAY "INDIVID TAC MEASURES..."
2.30 SAY "PERCENT COMPLETED A..."
2.42 SAY ZZATAC USING '999'
2.50 SAY ZZATAC USING '999'
2.59 SAY ZZATAC USING '999'
2.60 SAY ZZATAC USING '999'
2.75 SAY ZZATAC USING '999'
2.80 SAY ZZATAC USING '999'
3.25 SAY ZZATAC USING '999'
3.32 SAY ZZATAC USING '999'
3.52 SAY ZZATAC USING '999'
3.62 SAY ZZATAC USING '999'
3.72 SAY ZZATAC USING '999'
3.82 SAY ZZATAC USING '999'
4.25 SAY ZZATAC USING '999'
4.32 SAY ZZATAC USING '999'
4.52 SAY ZZATAC USING '999'
5.12 SAY ZZATAC USING '999'
5.22 SAY ZZATAC USING '999'
5.42 SAY ZZATAC USING '999'
5.52 SAY ZZATAC USING '999'
5.62 SAY ZZATAC USING '999'
8.02 SAY "FPT1..."
USE A:PERS
SELECT SECCNDAF7
USE E:QUAL
SELECT PRIMARY
JOIN TO TEMP PCR (P.COMPANY = ONECO .AND. P.PLATCCN =;
THREEF, AND. E.SSN=S.SSN) FIELDS;
SSN,SWIMQUAL,RFQUAL,PSTQUAL
USE TEMP
DO CNTQUAL
DELETE FILE TEMP
CASE I (MUNIT) = 'A' .AND. !(MPLAT) = 'D'
SELECT PRIMARY
USE A:PERS
SELECT SECCNDAF7
USE E:QUAL
SELECT PRIMARY
JOIN TO TEMP PCR (P.COMPANY = ONECO .AND. P.PLATCCN =;
THREEF, AND. E.SSN=S.SSN) FIELDS;
SSN,SWIMQUAL,RFQUAL,PSTQUAL
USE TEMP
DO CNTQUAL
DELETE FILE TEMP
CASE I (MUNIT) = 'E' .AND. !(MPLAT) = 'E'
SELECT PRIMARY
USE A:PERS
SELECT SECCNDAF7
USE E:QUAL
SELECT PRIMARY
JOIN TO TEMP PCR (P.COMPANY = TWOCO .AND. P.PLATCCN =;
CNPLAT .AND. E.SSN=S.SSN) FIELDS;
SSN,SWIMQUAL,RFQUAL,PSTQUAL
USE TEMP
DO CNTQUAL
DELETE FILE TEMP
CASE I (MUNIT) = 'F' .AND. !(MPLAT) = 'A'
SELECT PRIMARY
USE A:PERS
SELECT SECCNDAF7
USE E:QUAL
SELECT PRIMARY
JOIN TO TEMP PCR (P.COMPANY = TWOCO .AND. P.PLATCCN =;
CNPLAT .AND. E.SSN=S.SSN) FIELDS;
SSN,SWIMQUAL,RFQUAL,PSTQUAL
USE TEMP
DO CNTQUAL
DELETE FILE TEMP
CASE I (MUNIT) = 'F' .AND. !(MPLAT) = 'E'
SELECT PRIMARY
USE A:PERS
SELECT SECCNDAF7
USE E:QUAL
SELECT PRIMARY
JOIN TO TEMP PCR (P.COMPANY = TWOCO .AND. P.PLATCCN =;
CNPLAT .AND. E.SSN=S.SSN) FIELDS;
SSN,SWIMQUAL,RFQUAL,PSTQUAL
USE TEMP
DO CNTQUAL
DELETE FILE TEMP
CASE I (MUNIT) = 'F' .AND. !(MPLAT) = 'C'
SELECT PRIMARY
USE A:PERS
SELECT SECCNDAF7
USE E:QUAL
SELECT PRIMARY
JOIN TO TEMP PCR (P.COMPANY = TWOCO .AND. P.PLATCCN =;
CNPLAT .AND. E.SSN=S.SSN) FIELDS;
SSN,SWIMQUAL,RFQUAL,PSTQUAL
USE TEMP
DO CNTQUAL
DELETE FILE TEMP

122
CASE !(MUNIT) = 'E' .AND. !(MPLAT) = 'D'
    SELECT PRIMARY
    USE APERS
    SELECT SECONDARY
    USE E:QUAL
    SELECT PRIMARY
    JOIN TO TEMP FCF (P.COMPANY = TWOCO .AND. P.PLATCON =;
    USE P:QUAL .AND. F.SSN=S.SSN) FIELDS;
    USE TEMP
    DO CNTQUAL
    DELETE FILE TEMP

CASE !(MUNIT) = 'C' .AND. !(MPLAT) = 'E'
    SELECT PRIMARY
    USE APERS
    SELECT SECONDARY
    USE E:QUAL
    SELECT PRIMARY
    JOIN TO TEMP FCF (P.COMPANY = THREECO .AND. P.PLATCON;
    = CNEELT .AND. F.SSN=S.SSN) FIELDS;
    USE TEMP
    DO CNTQUAL
    DELETE FILE TEMP

CASE !(MUNIT) = 'C' .AND. !(MPLAT) = 'A'
    SELECT PRIMARY
    USE APERS
    SELECT SECONDARY
    USE E:QUAL
    SELECT PRIMARY
    JOIN TO TEMP FCF (P.COMPANY = THREECO .AND. P.PLATCON;
    = THREEPLT .AND. F.SSN=S.SSN) FIELDS;
    USE TEMP
    DO CNTQUAL
    DELETE FILE TEMP

CASE !(MUNIT) = 'C' .AND. !(MPLAT) = 'B'
    SELECT PRIMARY
    USE APERS
    SELECT SECONDARY
    USE E:QUAL
    SELECT PRIMARY
    JOIN TO TEMP FCF (P.COMPANY = THREECO .AND. P.PLATCON;
    = THREEPLT .AND. F.SSN=S.SSN) FIELDS;
    USE TEMP
    DO CNTQUAL
    DELETE FILE TEMP

CASE !(MUNIT) = 'C' .AND. !(MPLAT) = 'C'
    SELECT PRIMARY
    USE APERS
    SELECT SECONDARY
    USE E:QUAL
    SELECT PRIMARY
    JOIN TO TEMP FCF (P.COMPANY = THREECO .AND. P.PLATCON;
    = THREEPLT .AND. F.SSN=S.SSN) FIELDS;
    USE TEMP
    DO CNTQUAL
    DELETE FILE TEMP

CASE !(MUNIT) = 'C' .AND. !(MPLAT) = 'D'
    SELECT PRIMARY
    USE APERS
    SELECT SECONDARY
    USE E:QUAL
    SELECT PRIMARY
    JOIN TO TEMP FCF (P.COMPANY = THREECO .AND. P.PLATCON;
    = THREEPLT .AND. F.SSN=S.SSN) FIELDS;
    USE TEMP

DO CNTQUAL
DELETE FILE TEME
CASE !(MUNIT) = 'C'.AND. !(MPLAT) = 'E'
SELECT PRIMARY
USE A:PERS
SELECT SECONDARY
USE E:QUAL
SELECT PRIMARY
JOIN TO TEMP FCF (P.COMPANY =FOURCO .AND.
P.SSN=S.SSN) FIELDS SSN,SIMQUAL,RFLQUAL,PSTQUAL
USE TEMP
DO CNTQUAL
DELETE FILE TEME
CASE !(MUNIT) = 'D'.AND. !(MPLAT) = 'A'
SELECT PRIMARY
USE A:PERS
SELECT SECONDARY
USE E:QUAL
SELECT PRIMARY
JOIN TO TEMP FCF (P.COMPANY =FOURCO .AND. P.PLATCCN;
= CMPANY .AND. P.SSN=S.SSN) FIELDS;
SSN,SIMQUAL,RFLQUAL,PSTQUAL
USE TEMP
DO CNTQUAL
DELETE FILE TEME
CASE !(MUNIT) = 'D'.AND. !(MPLAT) = 'B'
SELECT PRIMARY
USE A:PERS
SELECT SECONDARY
USE E:QUAL
SELECT PRIMARY
JOIN TO TEMP FCF (P.COMPANY =FOURCO .AND. P.PLATCCN;
= THEPRT .AND. P.SSN=S.SSN) FIELDS;
SSN,SIMQUAL,RFLQUAL,PSTQUAL
USE TEMP
DO CNTQUAL
DELETE FILE TEME
CASE !(MUNIT) = 'D'.AND. !(MPLAT) = 'C'
SELECT PRIMARY
USE A:PERS
SELECT SECONDARY
USE E:QUAL
SELECT PRIMARY
JOIN TO TEMP FCF (P.COMPANY =FOURCO .AND. P.PLATCCN;
= THEPRT .AND. P.SSN=S.SSN) FIELDS;
SSN,SIMQUAL,RFLQUAL,PSTQUAL
USE TEMP
DO CNTQUAL
DELETE FILE TEME
CASE !(MUNIT) = 'D'.AND. !(MPLAT) = 'D'
SELECT PRIMARY
USE A:PERS
SELECT SECONDARY
USE E:QUAL
SELECT PRIMARY
JOIN TO TEMP FCF (P.COMPANY =FOURCO .AND. P.PLATCCN;
= THEPRT .AND. P.SSN=S.SSN) FIELDS;
SSN,SIMQUAL,RFLQUAL,PSTQUAL
USE TEMP
DO CNTQUAL
DELETE FILE TEME
ENDCASE
SAY "SWIMQUAL...
SAY ((ZZTOTAL-ZZSWIM)/ZZTOTAL)*100 USING '999'
SAY "PERCENT SWIM QUALIFIED..."
SAY ((ZZTOTAL-ZZRFL-ZZNARFL)/ZZTOTAL)*100 USING;
@ 14,31 SAY "PERCENT FIRE QUALIFIED..."
@ 14,60 SAY (ZTZOTAL-ZZRFZ-ZZMARFL) USING '999'
@ 16,0 SAY "PISTOL QUAL..."
@ 16,24 SAY ((ZTZOTAL-ZZEST-ZZNAFST)/ZTZOTAL)*100 USING '999'
@ 16,31 SAY "PERCENT PISTOL QUALIFIED..."
@ 16,60 SAY (ZTZOTAL-ZZPFST-ZZNAFST) USING '999'
@ 20,0 ACCEPT "PRESS 'ENTER' TO CONTINUE" TO PROCEED
RELEASE ALL LIKE ZZ??????
USE
RETURN
I. TRAINING COUNT ROUTINE

* ROUTINE NAME: CNTTE1.fRG
* MODULE NAME: STANDARD REPORT GENERATOR
* VERSION: 5.3.1.0
* AUTHOR: R.E. PRUETT
* DATE: 6NOV83
* VARIABLES USED: ONICO, TWCCC, THREECO, FOURCO, ONEFEL, *TWCEUT, THREEPLT, *F6RPL
* VARIABLES MODIFIED: NONE
* VARIABLES RELEASED: NONE
* FILES OPENED/CLOSED: NONE
* TEMP FILES CREATED: ZZTEMP (SAVES ALL VARIABLES UPRCN
* ENTIFING THIS ROUTINE)
* FILES: TRENGSTAT.fRG
* DESCRIPTION: THIS ROUTINE COUNTS THE NUMBER OF ESSENTIAL SUBJEIT TRAINING ELEMENTS THAT HAVE BEEN COMPLETED.

SAVE TO ZZTEMP
RELEASE ALL
COUNT TO ZZTOTAL
COUNT FOR *A* (COC) TC ZZACOC
COUNT FOR *E* (COC) TC ZZBCOC
COUNT FOR *C* (COC) TC ZZCCOC
COUNT FOR *A* (HIS) TO ZZAHIS
COUNT FOR *E* (HIS) TO ZZBHTS
COUNT FOR *C* (HIS) TO ZZCHIS
COUNT FOR *A* (COD) TO ZZACOD
COUNT FOR *E* (COD) TO ZZBCCD
COUNT FOR *C* (COD) TO ZZCCOD
COUNT FOR *A* (INT) TO ZZAINH
COUNT FOR *E* (INT) TO ZZBINT
COUNT FOR *C* (INT) TO ZZCINT
COUNT FOR *A* (AID) TO ZZAINH
COUNT FOR *E* (AID) TO ZZBINT
COUNT FOR *C* (AID) TO ZZCINT
COUNT FOR *D* (AID) TO ZZAINT
COUNT FOR *E* (AID) TO ZZBINT
COUNT FOR *A* (KID) TO ZZBINT
COUNT FOR *E* (KID) TO ZZCINT
COUNT FOR *A* (UNI) TO ZZAHIS
COUNT FOR *E* (UNI) TO ZZBHTS
COUNT FOR *A* (NLC) TO ZZCCOD
COUNT FOR *E* (NLC) TO ZZBCCD
COUNT FOR *C* (NLC) TO ZZCCOD
COUNT FOR *A* (NLC) TO ZZCINT
COUNT FOR *E* (NLC) TO ZZCINT
COUNT FOR *A* (NLC) TO ZZCINT
COUNT FOR *A* (MKS) TO ZZACOC
COUNT FOR *E* (MKS) TO ZZBCOC
COUNT FOR *C* (MKS) TO ZZCCOC
COUNT FOR *D* (MKS) TO ZZACOC
COUNT FOR *E* (MKS) TO ZZBCOC
COUNT FOR *A* (MKS) TO ZZACOC
COUNT FOR *E* (MKS) TO ZZBCOC
COUNT FOR *A* (TAC) TO ZZBHTS
COUNT FOR *E* (TAC) TO ZZBHTS

126
COUNT PCR "$\{TAC\}$ TC ZZCTAC
COUNT FOR "$\{TAC\}$ TC ZZDTAC
COUNT FOR "$\{TAC\}$ TC ZZETAC
COUNT FOR "$\{PFT1CLASS\}$ TO ZZFPT1
COUNT FOR "$\{PFT2CLASS\}$ TO ZZFPT2
RETURN
F. QUALIFICATION COUNT ROUTINE

* ROUTINE NAME: CNTQUAL.PRG
* MODULE NAME: STANDARD REPORT GENERATOR
* VERSION: 5.3.2.0
* AUTHOR: R.E. PRUIETT
* DATE: 12DEC83
* VARIABIES USED: NONE
* VARIABIES MODIFIED: NONE
* VARIABIES CREATED: ZZSWIM, ZZRFL, ZZNARFL, ZZPST, ZZNAPST
* VARIABIES RELEASED: NONE
* FILES OPENED/CLOSED: NONE
* TEXT FILES CREATED: NONE
* USING SUBROUTINES: TRNGSTAT.PRG
* DESCRIPTION: THIS ROUTINE FIRST SAVES ALL MEMORY
  * VARIABLES IN A MEMORY FILE (ZZTOTAL) AND THEN COUNTS THE
  * NUMBER OF PERSONS THAT HAVE NOT COMPLETED SWIMQUAL,
  * RFLQUAL, ESTQUAL AND THOSE PERSONS NOT REQUIRED TO
  * QUALIFY WITH THE RIFLE OR PISTOL (NA).
  * COUNT TO ZZTOTAL
  * COUNT FOR '***$ (SWIMQUAL) TO ZZSWIM
  * COUNT FOR '***$ (RFLQUAL) TO ZZRFL
  * COUNT FOR '***$ (ESTQUAL) TO ZZEST
  * RETURN
G. TRAINING ROSTER ROUTINE

* ROUTINE NAME: TRNGRPT.PRG
* MODULE NAME: STANDARD REPORT GENERATOR
* VERSION: 5.4.0
* AUTHOR: B. E. FRUITIT
* DATE: 20SEP83
* VARIABLES USED: MUNIT, MPLAT, CNECO, TWOCO, THREECO,
  FOURPLT, CNELT, TWCEL, THREEPL, FOURPLT
* VARIABLES MODIFIED: NONE
* VARIABLES CREATED: COUNT, PROCEED
* VARIABLES RELEASED: NONE
* FILES (OPENED/CLOSED): A: PERS INDEX A: ALPERS (ALL FILES CLOSED)
* TEMP FILES CREATED: NONE
* USING SUBROUTINES: STDREPT.PRG (5.0)
* DESCRIPTION: THIS ROUTINE PROVIDES THE "TRAINING ROSTER"
  ROUTINE THAT STARTS AT THE TOP OF THE PERS.DBF AND ICCKS
  AT EACH SEPERATE INDIVIDUALLY. IF A RECORD CORRESPONDS TO
  THE RECORD SELECTED BY THE USER, THE ROUTINE TRNGRPT.PRG IS
  CALLED. EACH RECORDS AND DISPLAYS THE TRAINING DATA THAT
  COBES TO THE CURRENT PERS.DBF RECORD (E.G. IF THE
  USER WANTS TRAINING DATA ON EVERY MEMBER OF 1ST PLAT, A
  CALL TO TRNGRPT.PRG LOCATES EACH RECORD IN THE PERS.DBF
  WHICH BELONGS TO 1ST PLAT, A CO. TRNGRPT.PRG THEN
  DISPLAYS THE TRAINING DATA FROM THE APPROPRIATE DATABASE
  FILES). INFORMATION ON THE INDIVIDUAL IS DISPLAYED IN
  SIX ROWS AND FOUR SETS OFDATA ARE DISPLAYED PER SCREEN.
  THE USER PRESSES THE ENTER KEY TO SCROLL FORWARD TO EACH
  SUCCESSIVE SET OF INDIVIDUAL DATA.

ERASE
2 128 SAY "TRAINING ROSTER REPORT"
2 128 ACCEPT "PRESS 'ENTER' KEY TO CONTINUE" TO PROCEED
ERASE
SELECT PRIMARY
USE A: PERS INDEX A: ALPERS
GOTC TOP
STORE 0 TO COUNT
DO WHILE .NOT. ZCP
DO CASE
  CASE I(MUNIT) = 'E'
  DO TRNGRPT
  STORE COUNT + 1 TO COUNT
  CASE !MUNIT) = 'A' .AND. !MPLAT = 'E'
  IF COMPANY = ONECO
  DO TRNGRPT
  STORE COUNT + 1 TO COUNT
  ENDIF
  CASE !MUNIT) = 'A' .AND. !MPLAT = 'A'
  IF COMPANY = ONECO .AND. PLATOON = ONEPLT
  DO TRNGRPT
  STORE COUNT + 1 TO COUNT
  ENDIF
  CASE !MUNIT) = 'A' .AND. !MPLAT = 'B'
  IF COMPANY = ONECO .AND. PLATOON = TWOPLT
  DO TRNGRPT
  STORE COUNT + 1 TO COUNT
  ENDIF
  CASE !MUNIT) = 'A' .AND. !MPLAT = 'C'
  IF COMPANY = ONECO .AND. PLATOON = THREEPL
  DO TRNGRPT
  STORE COUNT + 1 TO COUNT
  ENDIF
  CASE !MUNIT) = 'A' .AND. !MPLAT = 'D'
  IF COMPANY = ONECO .AND. PLATOON = FOURPLT
  DO TRNGRPT
129
SICRE CCUNT + 1 TO CCUNT
ENDIF
CASE !(MUNIT) = 'E' .AND. !(MPLAT) = 'E'
  IF COMPANY = TWCCO
  DO TRNGBPT
  STORE CCUNT + 1 TO CCUNT
  ENDFI
CASE !(MUNIT) = 'E' .AND. !(MPLAT) = 'A'
  IF COMPANY = TWCCO .AND. PLATOON = ONEPLT
  DO TRNGBPT
  STORE CCUNT + 1 TO CCUNT
  ENDFI
CASE !(MUNIT) = 'E' .AND. !(MPLAT) = 'B'
  IF COMPANY = TWCCO .AND. PLATOON = TWOPLT
  DO TRNGBPT
  STORE CCUNT + 1 TO CCUNT
  ENDFI
CASE !(MUNIT) = 'E' .AND. !(MPLAT) = 'C'
  IF COMPANY = TWCCO .AND. PLATOON = THREEPLT
  DO TRNGBPT
  STORE CCUNT + 1 TO CCUNT
  ENDFI
CASE !(MUNIT) = 'E' .AND. !(MPLAT) = 'D'
  IF COMPANY = TWCCO .AND. PLATOON = FOURPLT
  DO TRNGBPT
  STORE CCUNT + 1 TO CCUNT
  ENDFI
CASE !(MUNIT) = 'E' .AND. !(MPLAT) = 'C'
  IF COMPANY = TWCCO .AND. PLATOON = THREEPLT
  DO TRNGBPT
  STORE CCUNT + 1 TO CCUNT
  ENDFI
CASE !(MUNIT) = 'E' .AND. !(MPLAT) = 'A'
  IF COMPANY = TWCCO .AND. PLATOON = ONEPLT
  DO TRNGBPT
  STORE CCUNT + 1 TO CCUNT
  ENDFI
CASE !(MUNIT) = 'E' .AND. !(MPLAT) = 'B'
  IF COMPANY = TWCCO .AND. PLATOON = TWOPLT
  DO TRNGBPT
  STORE CCUNT + 1 TO CCUNT
  ENDFI
CASE !(MUNIT) = 'E' .AND. !(MPLAT) = 'C'
  IF COMPANY = TWCCO .AND. PLATOON = THREEPLT
  DO TRNGBPT
  STORE CCUNT + 1 TO CCUNT
  ENDFI
CASE !(MUNIT) = 'E' .AND. !(MPLAT) = 'D'
  IF COMPANY = TWCCO .AND. PLATOON = FOURPLT
  DO TRNGBPT
  STORE CCUNT + 1 TO CCUNT
  ENDFI
CASE !(MUNIT) = 'E' .AND. !(MPLAT) = '!'
  IF COMPANY = PCURCO
  DO TRNGBPT
  STORE CCUNT + 1 TO CCUNT
  ENDFI
CASE !(MUNIT) = 'E' .AND. !(MPLAT) = 'A'
  IF COMPANY = PCURCO .AND. PLATOON = ONEPLT
  DO TRNGBPT
  STORE CCUNT + 1 TO CCUNT
  ENDFI
CASE !(MUNIT) = 'E' .AND. !(MPLAT) = 'B'
  IF COMPANY = PCURCO .AND. PLATOON = TWOPLT
  DO TRNGBPT
  STORE CCUNT + 1 TO CCUNT
  ENDFI
CASE !(MUNIT) = 'E' .AND. !(MPLAT) = 'C'
  IF COMPANY = PCURCO .AND. PLATOON = THREEPLT

DO TRNGEPT
STORE CCUNT + 1 TO CCUNT
ENDDO
CASE "!(HUNIT) = 'C' .AND. !(HFLAT) = 'D'
IF COMPANY = FC1R1CO .AND. PLATOON = FOURPLT
DC TRNGEPT
STORE CCUNT + 1 TO CCUNT
ENDDO
ENDCASE
IF CCUNT = 4
@ 22, 0
ACCEPT "PRESS 'ENTER' TO CONTINUE" TO PROCEED
STORE 0 TO CCUNT
ENDDO
SELECT PRIMARY
SKIP
ENDDO
RELEASE CCUNT, PROC2EL, MSSID
USE
RETURN
B. TRAINING ROSTER FORMAT ROUTINE

* ROUTINE NAME:
* MODULE NAME: STANDARD REPORT GENERATOR
* VERSION: 5.4.1.0
* AUTHOR: R. E. PRUEIT
* DATE: 15 DEC 83
* VARIABLES USED: COUNT, PLATNAME, UNITNAME, MPLAT, MUNIT
* VARIABLES MODIFIED: NONE
* VARIABLES CREATED: FCW, MSSN
* VARIABLES RELEASED: NONE
* FILES OPENED/CLOSED: E:QUAL
* FILES CREATED:
* USING SUBROUTINES: TRNGROST.PRG
* DESCRIPTION: THIS ROUTINE PROVIDES THE FORMAT AND
* DISPLAYS THE "TRAINING ROSTER REPORT".

STORE (COUNT * 6) TO ROW
@ ROW, 0 SAY NAME
@ ROW, 30 SAY RANK
@ ROW, 33 SAY COMPANY
STORE SSN TO MSSN
SELECT SECONDARY
USE B:QUAL INDEX B:QUALSSN
FIND & MSSN
@ ROW, 39 SAY "SWIM QUAL"
@ ROW, 49 SAY SWIMQUAL
@ ROW, 52 SAY "RIFLE QUAL"
@ ROW, 63 SAY RFLQUAL
@ ROW, 78 SAY PSTQUAL
SELECT SECONDARY
USE E:EST INDEX E:ESTSSN
FIND & MSSN
@ ROW + 1, 0 SAY "CODE CP CONDUCT"
@ ROW + 1, 16 SAY COC
@ ROW + 1, 24 SAY "MARKS HIS"
@ ROW + 1, 32 SAY HIS
@ ROW + 1, 42 SAY "MARKSMANSHIP"
@ ROW + 1, 56 SAY MKS
@ ROW + 1, 67 SAY "DRILL"
@ ROW + 1, 74 SAY COD
@ ROW + 2, 0 SAY "INTERIOR GUARD"
@ ROW + 2, 16 SAY INT
@ ROW + 2, 24 SAY "MARKSMANSHIP"
@ ROW + 2, 32 SAY MKS
@ ROW + 2, 43 SAY "TAC MEASURES"
@ ROW + 2, 56 SAY TAC
@ ROW + 2, 67 SAY "NBC"
@ ROW + 2, 74 SAY NBC
@ ROW + 3, 0 SAY "PFT1"
@ ROW + 3, 6 SAY PFT1CLSS
@ ROW + 3, 10 SAY "PFT2"
@ ROW + 3, 16 SAY PFT2CLSS
@ ROW + 3, 24 SAY "EQUIP/UNIFORM"
@ ROW + 3, 36 SAY UNIT
@ ROW + 3, 43 SAY "FIRST ID"
@ ROW + 3, 54 SAY AID
FETUEN
I. SKILLSHIP STATUS ROUTINE

* ROUTINE NAME: EKSSES, FRG
* MODULE NAME: STANDARD REPORT GENERATOR
* VERSION: 5.0
* AUTHOR: R.E. FRUEN
* DATE: 28NOV83
* VARIABLES USED: UNITNAME, PLATNAME
* VARIABLES MODIFIED: NO GLOBAL VARIABLES MODIFIED
* VARIABLES CREATED: RFLEX, RFLSS, RFLMM, RFLUN, PSTEX, PSTSS, PSTUN, RFLESHOT, PSTEX, PSTSS, PSTUN, PSTQAL;
* VARIABLES RELEASED: EST??????, PCT??????, RFLEX, RFLSS, RFLESHOT, RFLESHOT, RFLESHOT
* FILES OPENED/CLOSED: ALL CLOSED
* TEMPE FILES CREATED: MRKSTEMP, MRKSTEMP
* DESCRIPTION: THIS ROUTINE CREATES A TEMPORARY FILE WHICH CONTAINS THE CURRENT YEAR RIFLE AND PISTOL QUALIFICATION RESULTS FOR EACH MEMBER IN THE SELECTED COMPANY AND/OR PLATOON. THE RESULTS ARE TOTALE BY QUALIFICATION CATEGORY FOR THE RIFLE AND PISTOL. PERCENTAGES ARE CALCULATED AND DISPLAIRED IN A SUMMARY STATUS REPORT WHICH IS AUTOMATICALLY DISPLAYED ON THE SCREEN.

ERASE LOAD UNIT
DO CASE
CASE !(MUNIT) = 'E'
USE P:QUAL
CASE !(MUNIT) <> 'E'. AND. !(MPLAT) = 'E'
USE A:PERSON
COPY TO MRKSTEMP ALL FIELD SSN FOR COMPANY=UNITNAME USE MRKSTEMP
SELECT SECONDARY USE P:QUAL
SELECT PRIMARY
JOIN TO MRKSTEMP FOR E.SSN = S.SSN FIELDS RFLQUAL;
ESTQUAL
USE MRKSTEMP
CASE !(MUNIT) <> 'E'. AND !(MPLAT) <> 'E'
USE A:PERSON
COPY TO MRKSTEMP ALL FIELD SSN FOR COMPANY=UNITNAME; AND, PLATOON = PLATNAME USE MRKSTEMP
SELECT SECONDARY USE P:QUAL
SELECT PRIMARY
JOIN TO MRKSTEMP FOR E.SSN = S.SSN FIELDS RFLQUAL;
ESTQUAL
USE MRKSTEMP

ENDCASE
COUNT FOR RFLQUAL = 'EX' TO RFLEX
COUNT FOR RFLQUAL = 'ES' TO RFLSS
COUNT FOR RFLQUAL = 'EM' TO RFLMM
COUNT FOR RFLQUAL = 'UN' TO RFLUN
COUNT FOR PSTQAL = 'EX' TO PSTEX
COUNT FOR PSTQAL = 'SS' TO PSTSS
COUNT FOR PSTQAL = 'EM' TO PSTMM
COUNT FOR PSTQAL = 'UN' TO PSTUN
ERASE
STORE (RFLEX*RFLESHOT+RFLMM*RFLESHOT) TO RFLESHOT
IF RFLESHOT > 0
STORRFLEX/RFLESHOT*100 TO PSTEX
STORE (RFLSS*RFLESHOT*100) TO PSTSS
STORE (RFLMM*RFLESHOT*100) TO PSTMM
STORE (RFLUN*RFLESHOT*100) TO PSTUN
ENDIF
STORE (PSTEX*PSTSS*PSTMM*PSTUN) TO PSTSHOT

133
IF PSTSHCT > 0
    STCFE (PSEX/PSTSHCT*100) TO PCTEXPST
    STCFE (PSSS/PSTSHCT*100) TO PCTSSPST
    STCFE (PSTM/PSTSHCT*100) TO PCTMMPST
    STCFE (PSTUN/PSTSHCT*100) TO PCTUNPST
ENDIF

1,10 SAY "MARKSMANSHIP STATUS REPORT"
3,1 SAY "RIFLE EXPERTS"..........
3,3 SAY "PISTOL EXPERTS"..........
5,1 SAY "RIFLE SHARPSHOOTERS"....
7,1 SAY "RIFLE MARKSMEN"........
9,1 SAY "UNQUALIFIED"............

IF PSTSHCT > 0
    @18,1 SAY "RIFLE EXPERTS"..........
    @18,2 SAY PCTEXP
    @19,1 SAY "RIFLE SHARPSHOOTERS"....
    @19,2 SAY PCTSSPST
    @20,1 SAY "RIFLE MARKSMEN"........
    @20,2 SAY PCTMMPST
    @21,1 SAY "UNQUALIFIED"............
    @21,2 SAY PCTUNPST
ELSE
    @18,1 SAY "NO UNIT MEMBERS HAVE FIRED THE "
    @20,1 SAY "RIFLE FOR QUALIFICATION THIS YEAR."
ENDIF

IF PSTSHCT > 0
    @18,3 SAY "PISTOL EXPERTS"..........
    @18,4 SAY PCTEX
    @19,3 SAY "PISTOL SHARPSHOOTERS"....
    @19,4 SAY PCTSSPST
    @20,3 SAY "PISTOL MARKSMEN"........
    @20,4 SAY PCTMMPST
    @21,3 SAY "UNQUALIFIED"............
    @21,4 SAY PCTUNPST
ELSE
    @18,3 SAY "NO UNIT MEMBERS HAVE FIRED THE"
    @20,3 SAY "PISTOL FOR QUALIFICATION THIS YEAR"
ENDIF

2,10 ACCEPT "PRESS 'ENTER' KEY TO CONTINUE" TO PROCEED
RELEASE ALL LIKE PST??????
RELEASE ALL LIKE PCT??????
IF "DO YOU WANT TO DO A COMPANY/BATTALION RANGE QUOTA; ANALYSIS? (Y/N)" = TO ANALYZE
IF ! (ANALYZE) = 'Y'
DC :CTASTUDY
ENDIF
RELEASE RLSS, RFLMM, RFLUN, RFLS, PROCEED, ANALYZE
USE
RETURN
J. CONVERT UNIT ROUTINE

* ROUTINE NAME: CVRTUNIT.PRG
* MODULE NAME: STANDARD REPORT GENERATOR
* VERSION: 5.6.1.0
* AUTHOR: R. E. FREIETT
* DATE: 28NOV83
* VARIABLES USED: MUNIT, MIAT, ONECO, TWOCO, THREECC,
  FOURCO, CMPLT, TWCPLT, THREEPLT, FOURPLT
* VARIABLES MODIFIED: NONE
* VARIABLES CREATED: UNITNAME, PLATNAME
* VARIABLES RELEASED: NONE
* FILES (OPENED/CLOSED): NONE
* TEMPL FILES CREATED: NONE
* USING SUBROUTINES: MKSSTAT.PRG (5.6.0) AND MOSSTAT.PRG
  (5.6.0)
* DESCRIPTION: THIS ROUTINE STORES THE NAME OF THE COMPANY
  AND PLATCEN SELECTED BY THE USER DURING ROUTINE
  SELECTUNIT. IT IS NECESSARY BECAUSE WHEN THE USER
  SELECTS A UNIT AND SUBUNIT IT IS DONE BY MENU SELECTION.
  ACCORDINGLY THE ONLY THING STORED INTO MEMORY IS THE
  CORRESPONDING LETTER FROM THE MENU PROVIDED BY THE
  SELECTUNIT.PRG (A, B, C, D, E). SOME SCREEN DISPLAYS
  REQUIRE THE UNIT NAME (E.G., "A CO" VICE "A") SO A
  CONVERSIOIN IS NECESSARY.

DO CASE
  CASE 1(MUNIT) = 'A'
    STORE ONECO TO UNITNAME
  CASE 1(MUNIT) = 'B'
    STORE TWOCO TO UNITNAME
  CASE 1(MUNIT) = 'C'
    STORE THREECO TO UNITNAME
  CASE 1(MUNIT) = 'D'
    STORE FOURCO TO UNITNAME
ENDCASE
IF I(MUNIT) <> 'E'
  LO CASE
  CASE 1(MELAT) = 'A'
    STORE ONEPLT TO PLATNAME
  CASE 1(MELAT) = 'B'
    STORE TWOPLT TO PLATNAME
  CASE 1(MELAT) = 'C'
    STORE THREEPLT TO PLATNAME
  CASE 1(MELAT) = 'D'
    STORE FOURPLT TO PLATNAME
ENDCASE
INDIF
RETURN
K. QUOTA STUDY ROUTINE

* ROUTINE NAME: QTA_STUDY.PRG
* VERSION: 5.6.2.0
* AUTHOR: J.E. PRUEITT
* DATE: 15 DEC 83
* VARIABLES USED: UNITNAME, PLATNAME, RPLSHOT, RFLEX,
  RFLSS, RFLLM, RFLLN, MUNIT, QMAR,
* ... QDEC
* VARIABLES MODIFIED: NO GLOBAL VARIABLES MODIFIED
* VARIABLES CREATED: RFLLA, MUNIT, TOTALREC, MLEFT, MSUM,
  NEEDLE, EXTRA, WHATIF
* VARIABLES RELEASED: MUNIT, TOTALREC, RFLLA, QMARI
* FILES (OPENED/CLOSED): NONE
* TEMPELPS CREATED: NONE
* USING SUBROUTINES: KKSSTAT.PRG
* DESCRIPTION: THIS MODULE IS USED BY KKSSTAT.PRG AND
  DISPLAYS THE RANGE QUOTAS THAT REMAIN FOR THE CURRENT
  YEAR. THE CURRENT MONTH IS EXTRACTED FROM THE 'LOG ON
  DATE' AND THE RANGE QUOTAS FOR EACH UNIT IS STORED IN A
  MEMORY FILE DURING THE ANNUAL INITIALIZATION PROCEDURE.
  SUMMARY INFORMATION IS PROVIDED CONCERNING THE RELATION-
  SHIPS BETWEEN REMAINING RANGE QUOTA REQUIREMENTS AND
  PREVIOUS RANGE QUOTA ALLOCATIONS. UPON CONCLUSION OF
  THE MODULE THE USER MAY RETURN TO THE STANDARD REPORT
  MENU OR TO A SCRATCH PAD TO REVISE QUOTA ASSIGNMENTS.
  THESE QUOTA ASSIGNMENTS FROM THE SCRATCH PAD CAN EITHER
  BE SAVED OR IGNORED.

CASE
  CASE !(MUNIT) <> 'E' ANL !(MPLAT) = 'E'
  @ 1,15 SAY "RANGE QUOTA ANALYSIS FOR" 
  @ 1,40 SAY UNITNAME
  CASE !(MUNIT) <> 'E' ANL !(MPLAT) <> 'E'
  @ 1,15 SAY "RANGE QUOTA ANALYSIS FOR" 
  @ 1,40 SAY UNITNAME
  @ 1,46 SAY "" 
  @ 1,48 SAY PLATNAME
  CASE !(MUNIT) = 'E'
  @ 1,15 SAY "RANGE QUOTA ANALYSIS FOR BATTALION" 
ENDCASE
COUNT FOR RELQUAL = 'NA' TO RFLLA
STORE $(AADATE,3,2) TC MONT
GOTO 047TCM
STORE 0 TC TOTALREC
2 4 1 SAY "NUMBER OF MARINES FIRED FOR QUALIFICATION:" 
2 5 1 SAY "CURRENT:" 
2 5,56 SAY RPLSHOT
2 6 1 SAY "NUMBER OF MARINES FIRED FOR QUALIFICATION:" 
2 7 1 SAY "CURRENT YEAR BUT DID NOT:" 
2 7,56 SAY RFLLN
2 8 1 SAY "NUMBER OF MARINES NOT REQUIRED TO QUALIFY:" 
2 9 1 SAY "CURRENT:" 
2 9,56 SAY RFLLA
2 10 1 SAY "MARINES LEFT TO QUALIFY DURING CURRENT:" 
2 10,56 SAY MLEFT 
LO CASE
CASE \( \text{MCNT} = '1' \)
  RESTORE FROM A:ARNGQTA ADDITIVE
CASE \( \text{MCNT} = '2' \)
  RESTORE FROM A:ARNGQTA ADDITIVE
CASE \( \text{MCNT} = '3' \)
  RESTORE FROM A:ARNGQTA ADDITIVE
CASE \( \text{MCNT} = '4' \)
  RESTORE FROM A:ARNGQTA ADDITIVE
CASE \( \text{MCNT} = '5' \)
  RESTORE FROM A:ARNGQTA ADDITIVE
CASE \( \text{MCNT} = '6' \)
  RESTORE FROM A:ARNGQTA ADDITIVE
ENDCASE

ENDCASE

CASE \( \text{MCNT} = '01' \)
  @ 12,4 SAY "FEB MAR APR MAY JUN JUL AUG;"
  SEE OCT NCV LEC"
  @ 14,4 SAY STR(CFEB,3)
  @ 14,10 SAY STR(CMAR,3)
  @ 14,16 SAY STR(CMAY,3)
  @ 14,22 SAY STR(CJUN,3)
  @ 14,28 SAY STR(CJUL,3)
  @ 14,34 SAY STR(CAUG,3)
  @ 14,40 SAY STR(CSEP,3)
  @ 14,46 SAY STR(COCT,3)
  @ 14,52 SAY STR(CDEC,3)
  STCRE (QFEB+QMAR+QAPR+QJAN+QJUN+QJUL+QAUG+QSEP+QOCT+QNCV+QLEC) TO MSUM
  @ 16,4 SAY "A TOTAL OF "
  @ 16,10 SAY STR(MSUM,3)
  @ 16,16 SAY "QUOTAS REMAIN FOR THE CURRENT YEAR."
CASE \( \text{MCNT} = '02' \)
  @ 12,4 SAY "MAR APR MAY JUN JUL AUG SEP;"
  OCT NCV LEC"
  @ 14,4 SAY STR(CMAR,3)
  @ 14,10 SAY STR(CMAY,3)
  @ 14,16 SAY STR(CJUN,3)
  @ 14,22 SAY STR(CJUL,3)
  @ 14,28 SAY STR(CAUG,3)
  @ 14,34 SAY STR(CSEP,3)
  @ 14,40 SAY STR(COCT,3)
  @ 14,46 SAY STR(CDEC,3)
  STCRE (QMAR+QAPR+QJAN+QJUN+QJUL+QAUG+QSEP+QOCT+QNCV+QLEC) TO MSUM
  @ 16,4 SAY "A TOTAL OF "
  @ 16,10 SAY STR(MSUM,3)
  @ 16,16 SAY "QUOTAS REMAIN FOR THE CURRENT YEAR."
CASE \( \text{MCNT} = '03' \)
  @ 12,4 SAY "APR MAY JUN JUL AUG SEP OCT;"
  NCV LEC"
  @ 14,4 SAY STR(CAPR,3)
  @ 14,10 SAY STR(CMAY,3)
  @ 14,16 SAY STR(CJUN,3)
  @ 14,22 SAY STR(CJUL,3)
  @ 14,28 SAY STR(CAUG,3)
  @ 14,34 SAY STR(CSEP,3)
  @ 14,40 SAY STR(COCT,3)
  @ 14,46 SAY STR(CDEC,3)
  STCRE (QAPR+QMAY+QJUN+QJUL+QAUG+QSEP+QOCT+QNCV+QDEC) TO MSUM
  @ 16,4 SAY "A TOTAL OF "
  @ 16,10 SAY STR(MSUM,3)
  @ 16,16 SAY "QUOTAS REMAIN FOR THE CURRENT YEAR."
CASE \( \text{MCNT} = '04' \)
  @ 12,4 SAY "MAY JUN JUL AUG SEP OCT NOV; LEC"

137
CASE MONTH = '04'
14, 4 SAY "JUL AUG SEP OCT NOV DEC"
14, 10 SAY STR(QJUN,3)
14, 16 SAY STR(QJUL,3)
14, 22 SAY STR(QAUG,3)
14, 28 SAY STR(QSEP,3)
14, 34 SAY STR(QOCT,3)
14, 40 SAY STR(QNOV,3)
14, 46 SAY STR(QDEC,3)
SIR (QJUN+QJUL+QAUG+QSEP+QOCT+QNOV+QDEC) TO MSUM
16, 4 SAY "A TOTAL OF"
16, 15 SAY STR(MSUM,3)
CASE MONTH = '05 '
14, 4 SAY "JUN JUL AUG SEP OCT NOV DEC"
14, 10 SAY STR(QJUN,3)
14, 16 SAY STR(QJUL,3)
14, 22 SAY STR(QAUG,3)
14, 28 SAY STR(QSEP,3)
14, 34 SAY STR(QOCT,3)
14, 40 SAY STR(QNOV,3)
14, 46 SAY STR(QDEC,3)
SIR (QJUN+QJUL+QAUG+QSEP+QOCT+QNOV+QDEC) TO MSUM
16, 4 SAY "A TOTAL OF"
16, 15 SAY STR(MSUM,3)
CASE MONTH = '06 '
14, 4 SAY "JUN JUL AUG SEP OCT NOV DEC"
14, 10 SAY STR(QJUN,3)
14, 16 SAY STR(QJUL,3)
14, 22 SAY STR(QAUG,3)
14, 28 SAY STR(QSEP,3)
14, 34 SAY STR(QOCT,3)
14, 40 SAY STR(QNOV,3)
14, 46 SAY STR(QDEC,3)
SIR (QJUN+QJUL+QAUG+QSEP+QOCT+QNOV+QDEC) TO MSUM
16, 4 SAY "A TOTAL OF"
16, 15 SAY STR(MSUM,3)
CASE MONTH = '07 '
14, 4 SAY "SEP OCT NOV DEC"
14, 10 SAY STR(QSEP,3)
14, 16 SAY STR(QOCT,3)
14, 22 SAY STR(QNOV,3)
14, 28 SAY STR(QDEC,3)
SIR (QSEP+QOCT+QNOV+QDEC) TO MSUM
16, 4 SAY "A TOTAL OF"
16, 15 SAY STR(MSUM,3)
CASE MONTH = '08 '
14, 4 SAY "SEP OCT NOV DEC"
14, 10 SAY STR(QSEP,3)
14, 16 SAY STR(QOCT,3)
14, 22 SAY STR(QNOV,3)
14, 28 SAY STR(QDEC,3)
SIR (QSEP+QOCT+QNOV+QDEC) TO MSUM
16, 4 SAY "A TOTAL OF"
16, 15 SAY STR(MSUM,3)
CASE MONTH = '09 '
14, 4 SAY "OCT NOV DEC"
14, 10 SAY STR(QSEP,3)
14, 16 SAY STR(QOCT,3)
14, 22 SAY STR(QNOV,3)
14, 28 SAY STR(QDEC,3)
SIR (QSEP+QOCT+QNOV+QDEC) TO MSUM
16, 4 SAY "A TOTAL OF"
16, 15 SAY STR(MSUM,3)
CASE MONTH = '10'
14, 4 SAY "OCT NOV DEC"
14.4 SAY STR(QNCV,3)
14.10 SAY STR(QDEC,3)
STORE (QNCV+QDEC) TO MSU
16.4 SAY "A TOTAL OF"
16.15 SAY STR(MSUM,3)
16.19 SAY "QUOTAS REMAIN FOR THE CURRENT YEAR."
CASE MONTH = '11'
12.4 SAY "DEC"
14.4 SAY STR(QDEC,3)
STORE (QDEC) TO MSUM
16.4 SAY "A TOTAL OF"
16.15 SAY STR(MSUM,3)
16.19 SAY "QUOTAS REMAIN FOR THE CURRENT YEAR."
CASE MONTH = '12'
14.10 SAY "THIS IS THE LAST MONTH IN CALENDAR YEAR"
ENDCASE
TO CASE
CASE (MSUM = MLEFT)
  18.4 SAY "REMAINING QUOTAS EQUAL NUMBER OF MARINES;"
  18.64 SAY STR(MSUM,3)
CASE (MLEFT > MSUM)
  18.4 SAY "YOU NEED"
  18.13 SAY STR(NEEDED,3)
  18.19 SAY "MORE QUOTAS FOR THE YEAR."
CASE (MSUM > MLEFT)
  18.4 SAY "YOU HAVE"
  18.4 STORE (MSUM - MLEFT) TO EXTRA
  18.13 SAY STR(EXTRA,3)
  18.16 SAY "EXTRA QUOTAS."
ENDCASE
19.1 SAY "SELECT ONE"
20.1 SAY "A...CHANGE QUOTAS ON SCRATCH PAD"
21.1 SAY "B...RETURN TO STANDARD REPORT MENU"
22.0 ACCEPT "SELECT OPTION ==> " TO WHATIF
IF I(WHATIF) = 'A'
  DO B:QUOTAPAD
ENDIF
RELEASE MONTH, WHATIF, EXTRA, NEEDED, MSUM, MLEFT, TOTALREC, RFLNA
RELEASE ALL LIKE Q??? RETURN

139
L. QUOTA SCRATCH PAD ROUTINE

* ROUTINE NAME: QUOTAED.PRG
* MODULE NAME: STANDARD REPORT GENERATOR
* VERSION: 5.6.2.1.0
* AUTHOR: R.E. PRUITT
* DATE: SPEC3
* VARIABLES USED: MLEFT, MUNIT
* VARIABLES MODIFIED:
* VARIABLES CREATED: FICKMnth, TEMPSUM, WHICHWAY, CONTINUE,
  OPEE,...OEE, EXTRA
* VARIABLES RELEASED: C???, MLEFT, TEMPSUM, WHICHWAY,
  PICKMnth, CONTINUE, EXTRA
* FILES (OPENED/CLOSED): NONE
* TIME FILES CREATED: NONE
* USING SUBROUTINES: CTASTDY.ERG
* DESCRIPTION: THIS MODULE GIVES THE USER AN OPPORTUNITY TO
  INTER-INTERACTIVELY MODIFY THE REMAINING MONTHLY RANGE
  QUOTAS. THIS CAN BE DONE AN UNLIMITED NUMBER OF TIMES
  EACH TIME RETURNING TO THE ACTUAL MONTHLY ALLOCATION AS
  BASE DATA. HOWEVER, IF THE USER ELECTS TO PERMANENTLY
  REPLACE THE MONTHLY QUOTA ALLOCATION IN MEMORY, HE MAY DO
  SO BY SELECTING THE 'REPLACE CURRENT QUOTAS' OPTION.

STORE T TO FICKMnth
SET CCLCN OFF
DO CASE
  CASE MONT = '01'
    DO WHILE FICKMnth = T
      DO CASE
        CASE ! (MUNIT) = 'A'
          RESTORE FROM A:ARNGQTAS ADDITIVE
        CASE ! (MUNIT) = 'B'
          RESTORE FROM A:BRNGQTAS ADDITIVE
        CASE ! (MUNIT) = 'C'
          RESTORE FROM A:CRNGQTAS ADDITIVE
        CASE ! (MUNIT) = 'D'
          RESTORE FROM A:DRNGQTAS ADDITIVE
        CASE ! (MUNIT) = 'E'
          RESTORE FROM A:BRNGQTAS ADDITIVE
      ENDCASE
    ENDDO CASE
  ENDO CASE
  2 SAY "FILE RANGE QUOTA SCRATCH PAD"
  4 SAY "SPACE BELOW THIS LINE IS FOR:
       CALCULATIONS AND WILL BE SAVED"
  5 SAY "ONLY IF YOU SPECIFY THE 'REPLACE CURRENT QUOTAS' OPTION"
  6 SAY "***************
       *******************
  8 SAY "FEB MAR APR MAY JUN JUL; Aug Sep Cct Nov Dec"
  10.4 GET QFEB PICTURE '999'
  10.10 GET QMAR PICTURE '999'
  10.16 GET QAPR PICTURE '999'
  10.22 GET QMAY PICTURE '999'
  10.28 GET QJUN PICTURE '999'
  10.34 GET QJUL PICTURE '999'
  10.40 GET QAUG PICTURE '999'
  10.46 GET QCCT PICTURE '999'
  10.52 GET QNOV PICTURE '999'
  10.58 GET QCDEC PICTURE '999'
  10.64 GET QJCC PICTURE '999'
  REAL
  12.4 SAY "A TOTAL OF "
STORE (QFEB+QMAR+QAPR+QSMAY+QJUN+QJUL );
AUG+QSEP+QOCI+QNOV+QDEC) TO TEMPSUM
@ 12,15 SAY STR(TEMPSUM,3)
@ 12,19 SAY "QUOTAS REMAIN FOR THE CURRENT YEAR."

DC CASE
   CASE (TEMPSUM = MLEFT)
      @ 14,4 SAY "QUOTAS EQUAL NUMBER OF MARINES;
      LEFT TO QUALIFY..."
   CASE (MLEFT > TEMPSUM)
      @ 14,4 SAY "YOU NEED "
      STORE (MLEFT - TEMPSUM) TO NEEDED
      @ 14,13 SAY STR(NEEDED,3)
      @ 14,19 SAY "MORE QUOTAS,"
   CASE (TEMPSUM > MLEFT)
      @ 14,4 SAY "YOU HAVE"
      STORE (TEMPSUM - MLEFT) TO EXTRA
      @ 14,13 SAY STR(EXTRA,3)
      @ 14,18 SAY "EXCESS QUOTAS."
   END CASE
   END CASE
   @ 15,24 SAY "SELECT ONE OPTION"
@ 16,18 SAY "A...START AGAIN WITH ORIGINAL QUOTAS"
@ 17,18 SAY "B...RETURN TO STANDARD REPORT;"
@ 18,18 SAY "C...PERMANENTLY REPLACE ORIGINAL QUOTAS"
ACCEPT "SELECT ONE ===> " TO WHICHWAY

DC CASE
   CASE (WHICHWAY) = 'B'
   STORE F TC PICKMNTH
   CASE (WHICHWAY) = 'C'
      ACCEPT "DO YOU WANT TO PERMANENTLY REPLACE THE;
      ORIGINAL SET OF QUOTAS (Y/N)?"; TO CONTINUE
      IF CONTINUE
        DC CASE
          CASE (MUNIT) = 'A'
          SAVE TC A:ARNGQTAS ALL LIKE Q???
          CASE (MUNIT) = 'B'
          SAVE TC A:BRNGQTAS ALL LIKE Q???
          CASE (MUNIT) = 'C'
          SAVE TC A:CRNGQTAS ALL LIKE Q???
          CASE (MUNIT) = 'D'
          SAVE TC A:DRNGQTAS ALL LIKE Q???
          CASE (MUNIT) = 'E'
          SAVE TC A:ENRNGQT AS ALL LIKE Q???
          ENDCASE
      END CASE
   ELSE
      STORE F TO PICKMNTH
   END IF
   ENDCASE

ENDDO

CASE MONTH = '02'
DO WHILE PICKMNTH = T
   DO CASE
      CASE (MUNIT) = 'A'
      RESTORE PFCM A:ARNGQTAS ADDITIVE
      CASE (MUNIT) = 'B'
      RESTORE PFCM A:BRNGQTAS ADDITIVE
      CASE (MUNIT) = 'C'
      RESTORE PFCM A:CRNGQTAS ADDITIVE
      CASE (MUNIT) = 'D'
      RESTORE PFCM A:DRNGQTAS ADDITIVE
      CASE (MUNIT) = 'E'
      RESTORE PFCM A:ENRNGQTAS ADDITIVE
   ENDCASE
   ENDDO
ERASE
@ 2.15 SAY "PIECE RANGE QUOTA SCRATCH PAD"
@ 4.5 SAY "SPACE BELOW THIS LINE IS FOR:
CALCULATIONS AND WILL BE SAVED"
@ 5.5 SAY "ONLY IF YOU SPECIFY THE 'REPLACE:
CURRENT QUOTAS' OPTION".
@ 6.0 SAY "**********************************************************************
* * * * * * * * * * * * *
* **********************************************************************
@ 8.5 SAY "MAR APR MAY JUN JUL;
AUG SEPT OCT NOV DEC"
@ 10.10 GET MARS PICTURE '999'
@ 10.16 GET CAPR PICTURE '999'
@ 10.22 GET MAY PICTURE '999'
@ 10.28 GET CJUN PICTURE '999'
@ 10.34 GET CJUL PICTURE '999'
@ 10.40 GET CAUG PICTURE '999'
@ 10.46 GET SEP PICTURE '999'
@ 10.52 GET CCT PICTURE '999'
@ 10.58 GET CEC PICTURE '999'
READ
@ 12.4 SAY "A TOTAL OF"
STORE [MARS+CAPR+MAY+CJUN+CJUL+
CAUG+SEP+CCT+CEC] TO TEMPSUM
@ 12.15 SAY STR(TEMPSUM, 3)
@ 12.19 SAY "QUOTAS REMAIN FOR THE CURRENT YEAR."
DO CASE
CASE (TEMPSUM = MLEFT)
@ 14.4 SAY "QUOTAS EQUAL NUMBER OF MARINES;
LEFT TO QUALIFY..."
@ 14.53 SAY STR(TEMPSUM, 3)
CASE (MLEFT > TEMPSUM)
@ 14.4 SAY "YOU NEED"
STORE (MLEFT - TEMPSUM) TO NEEDED
@ 14.13 SAY STR(NEEDED, 3)
@ 14.19 SAY "MORE QUOTAS."
CASE (TEMPSUM > MLEFT)
@ 14.4 SAY "YOU HAVE"
STORE (TEMPSUM - MLEFT) TO EXTRA
@ 14.13 SAY STR(EXTRA, 3)
@ 14.18 SAY "EXCESS QUOTAS..."
ENDCASE
@ 15.24 SAY "SELECT ONE OPTION"
@ 16.18 SAY "A...START AGAIN WITH ORIGINAL QUOTAS"
@ 17.18 SAY "E...RETURN TO STANDARD REPORT;
SELECTION MENU"
@ 18.18 SAY "C...PERMANENTLY REPLACE ORIGINAL;
QUOTAS"
ACCEPT "SELECT ONE ===> " TO WHICHWAY
DO CASE
CASE ! (WHICHWAY) = 'B'
STORE P TO PICKMTH
CASE ! (WHICHWAY) = 'C'
ACCEPT "IF YOU WANT TO PERMANENTLY REPLACE THE;
ORIGINAL SET OF QUOTAS (Y/N)?" TO CONTINUE
IF CONTINUE
DO CASE
CASE ! (MUNIT) = 'A'
SAVE TO A:ARNGQTAS ALL LIKE Q???
CASE ! (MUNIT) = 'B'
SAVE TO A:BRNGQTAS ALL LIKE Q???
CASE ! (MUNIT) = 'C'
SAVE TO A:CRNGQTAS ALL LIKE Q???
CASE ! (MUNIT) = 'D'
SAVE TO A:DRNGQTAS ALL LIKE Q???
CASE I(MUNIT) = 'E'
  SAVE TO A:BNRNGQTA ALL LIKE Q???
ENDCASE
ELSE
  STORE F TO PICKMNTH
ENDIF
ENDCASE
ENDDO

CASE MCNTH = '03'
  DO WHILE PICKMNTH = T
    DO CASE
      CASE I(MUNIT) = 'A'
        RESTORE PFCM A:ARNGQTA ADDITIVE
      CASE I(MUNIT) = 'B'
        RESTORE PFCM A:BRNGQTA ADDITIVE
      CASE I(MUNIT) = 'C'
        RESTORE PFCM A:CRNGQTA ADDITIVE
      CASE I(MUNIT) = 'D'
        RESTORE PFCM A:DRNGQTA ADDITIVE
      CASE I(MUNIT) = 'E'
        RESTORE PFCM A:BNRNGQTA ADDITIVE
    ENDCASE
    ERASE
    2 215 SAY "RIFLE RANGE QUOTA SCRATCH PAD"
    2 245 SAY "SPACE BELOW THIS LINE IS FOR:
    2 275 SAY "CALCULATIONS AND WILL BE SAVED"
    2 305 SAY "OK IF YOU SPECIFY THE 'REPLACE:
    2 335 SAY "CURRENT QUOTAS' OPTION"
    2 365 SAY "**************************************************************************";
    2 395 SAY "APR  MAY  JUN  JUL;
    2 425 SAY "AUG  SEP  OCT  Nov  DEC"
    2 455 GET APR PICTURE '999'
    2 465 GET MAY PICTURE '999'
    2 475 GET JUN PICTURE '999'
    2 485 GET JUI PICTURE '999'
    2 495 GET AUG PICTURE '999'
    2 505 GET SEP PICTURE '999'
    2 515 GET OCT PICTURE '999'
    2 525 GET NOV PICTURE '999'
    2 535 GET DEC PICTURE '999'
    REAL
    2 545 SAY "A TOTAL OF 
    2 555 STORE (APR+MAY+JUN+JUL+;
    2 565 GET OCT+NOV+DEC) TO TEMPSUM
    2 575 SAY STR(TEMPSUM,3)
    2 585 SAY "QUOTAS REMAIN FOR THE CURRENT YEAR."
    DCASE
    CASE TEMPSUM = MLEFT
      2 595 SAY "QUOTAS EQUAL NUMBER OF MARINES;
      2 605 LEFT TO QUALIFY..."
      2 615 SAY STR(TEMPSUM,3)
    CASE (MLEFT > TEMPSUM)
      2 625 SAY "YOU NEED 
      2 635 STORE (MLEFT - TEMPSUM) TO NEEDED
      2 645 SAY STR(NEEDED,3)
      2 655 SAY "MORE QUOTAS."
    CASE (TEMPSUM > MLEFT)
      2 665 SAY "YOU HAVE 
      2 675 STORE (TEMPSUM - MLEFT) TO EXTRA
      2 685 SAY STR(EXTRA,3)
      2 695 SAY "EXCESS QUOTAS."
ENDCASE
@ 15,24 SAY "SELECT ONE OPTION"
@ 16,18 SAY "A...START AGAIN WITH ORIGINAL QUOTAS"
@ 17,18 SAY "E...RETURN TO STANDARD REPORT;"
@ SELECTION MENU
@ 18,18 SAY "C...PERMANENTLY REPLACE ORIGINAL QUOTAS;"
ACCEPT "SELECT ONE ===> " TO WHICHWAY
DO CASE
CASE I (WHICHWAY) = 'A'
STORE F TO PICKMNTH
CASE I (WHICHWAY) = 'C'
ACCEPT "WE YOU WANT TO PERMANENTLY REPLACE THE;
ORIGINAL SET OF QUOTAS (Y/N)?" TO CONTINUE IF CONTINUE
DO CASE
CASE I (MUNIT) = 'A'
SAVE TO A:ARNGQTAS ALL LIKE Q???
CASE I (MUNIT) = 'B'
SAVE TO A:BRNGQTAS ALL LIKE Q???
CASE I (MUNIT) = 'C'
SAVE TO A:CRNGQTAS ALL LIKE Q???
CASE I (MUNIT) = 'D'
SAVE TO A:DRNGQTAS ALL LIKE Q???
CASE I (MUNIT) = 'E'
SAVE TO A:BNRNGQTAS ALL LIKE Q???
ENDCASE
STORE F TO PICKMNTH
ELSE
STORE F TO PICKMNTH
ENDIF
ENDCASE
ENDDC
CASE MCNTF = '04'
DO WHILE PICKMNTH = T
DO CASE
CASE I (MUNIT) = 'A'
RESTORE FROM A:ARNGQTAS ADDITIVE
CASE I (MUNIT) = 'B'
RESTORE FROM A:BRNGQTAS ADDITIVE
CASE I (MUNIT) = 'C'
RESTORE FROM A:CRNGQTAS ADDITIVE
CASE I (MUNIT) = 'D'
RESTORE FROM A:DRNGQTAS ADDITIVE
CASE I (MUNIT) = 'E'
RESTORE FROM A:BNRNGQTAS ADDITIVE
ENDCASE
ERASE
@ 2,15 SAY "FILE RANGE QUOTA SCRATCH PAD"
@ 4,5 SAY "SPACE BELOW THIS LINE IS FOR;
CALCULATIONS AND WILL BE SAVED"
@ 6,5 SAY "ONLY IF YOU SPECIFY THE 'REPLACE;
CURRENT QUOTAS' OPTION"
@ 8,5 SAY "*******************************;
*******************************"
@ 8,5 SAY "MAY JUN JUL; SEP OCT NOV DEC"
@ 10,22 GET CAUG PICTURE '999'
@ 10,28 GET CJUN PICTURE '999'
@ 10,34 GET CJUL PICTURE '999'
@ 10,40 GET CSEP PICTURE '999'
@ 10,46 GET CCT PICTURE '999'
@ 10,52 GET CCCT PICTURE '999'
@ 10,58 GET CDEC PICTURE '999'
@ 10,64 GET CDEC PICTURE '999'
READ
A TOTAL OF
STOR ((MAY+JUN+JUL+
CAUG+SEP+OCT+NOV+CDEC) TO TEMPSUM
A SAY "QUOTAS REMAIN FOR THE CURRENT YEAR."

DC CASE
CASE (TEMPSUM = MLEFT) 
A SAY "QUOTAS EQUAL NUMBER OF MARINES;"
LEFT TO QUALIFY..."
A SAY STR(TEMPSUM,3)
CASE (MLEFT > TEMPSUM)
A SAY "YOU NEED"
STORE (MLEFT - TEMPSUM) TO NEEDED
A SAY STR(NEEDED,3)
CASE (TEMPSUM > MLEFT)
A SAY "YOU HAVE"
STORE (TEMPSUM - MLEFT) TO EXTRA
A SAY STR(EXTRA,3)
A SAY "EXCESS QUOTAS."
ENDCASE

SELECT ONE OPTION
A START AGAIN WITH ORIGINAL QUOTAS"
A "RETURN TO STANDARD REPORT;"
SELECT MENU"
A PERMANENTLY REPLACE ORIGINAL;
QUOTAS"
ACCEPT "SELECT ONE OF " TO WHICHWAY

DO CASE
CASE !(WHICHWAY) = 'B'
STORE F TO PICKMNTH
CASE !(WHICHWAY) = 'C'
ACCEPT "DO YOU WANT TO PERMANENTLY REPLACE THE;
ORIGINAL SET OF QUOTAS, (Y/N)?" TO CONTINUE
IF CONTINUE
DO CASE
CASE !(MUNIT) = 'A'
RESTORE FROM A:ARNGQTAS ALL LIKE Q???
CASE !(MUNIT) = 'B'
RESTORE FROM A:BRNGQTAS ALL LIKE Q???
CASE !(MUNIT) = 'C'
RESTORE FROM A:CRNGQTAS ALL LIKE Q???
CASE !(MUNIT) = 'D'
RESTORE FROM A:DRNGQTAS ALL LIKE Q???
CASE !(MUNIT) = 'E'
RESTORE FROM A:BNRNGQTAS ALL LIKE Q???
ENDCASE
STORE F TO PICKMNTH
ELSE
STORE F TO PICKMNTH
ENDIF
ENDCASE

CASE MCNT = '05'
DO WHILE PICKMNTH = T
DO CASE
CASE !(MUNIT) = 'A'
RESTORE FROM A:ARNGQTAS ADDITIVE
CASE !(MUNIT) = 'B'
RESTORE FROM A:BRNGQTAS ADDITIVE
CASE !(MUNIT) = 'C'
RESTORE FROM A:CRNGQTAS ADDITIVE
CASE !(MUNIT) = 'D'
RESTORE FROM A:DRNGQTAS ADDITIVE

CASE I (MUNIT) = 'E'
RESTORE FROM A:BRNGQTA ADDITIVE
ENDCASE

@ 2,15 SAY "FILE RANGE QUOTA SCRATCH PAD"
@ 4,5 SAY "SPACE BELOW THIS LINE IS FOR;
CALCULATIONS AND WILL BE SAVED"
@ 5,5 SAY "ONLY IF YOU SPECIFY THE "REPLACE;" 
CURRENT QUOTAS" OPTION"
@ 6,0 SAY "*******************************************************************************
@ 8,5 SAY "JUN JUL;
AUG SEP OCT NOV DEC"
@ 10,28 GET JUN PICTURE '999'
@ 10,34 GET JUL PICTURE '999'
@ 10,40 GET AUG PICTURE '999'
@ 10,46 GET SEP PICTURE '999'
@ 10,52 GET OCT PICTURE '999'
@ 10,58 GET NOV PICTURE '999'
@ 10,64 GET DEC PICTURE '999'
FINAL

@ 12,4 SAY "A TOTAL OF"
STORE (JUN+JUL+*
AUG+SEP+OCT+NOV+DEC) TO TEMPSUM
@ 12,15 SAY STR(TEMPSUM,)
@ 12,19 SAY "QUOTAS REMAIN FOR THE CURRENT YEAR."

DC CASE
CASE (TEMPSUM = MLEFT)
@ 14,0 SAY "QUOTAS EQUAL NUMBER OF MARINES;
LEFT TO QUALIFY..."
@ 14,53 SAY STR(TEMPSUM, 3)
CASE (MLEFT > TEMPSUM)
@ 14,4 SAY "YOU NEED"
STORE (MLEFT - TEMPSUM) TO NEEDED
@ 14,13 SAY STR(NEEDED, 3)
@ 14,19 SAY "MORE QUOTAS."
CASE (TEMPSUM > MLEFT)
@ 14,4 SAY "YOU HAVE"
STORE (TEMPSUM - MLEFT) TO EXTRA
@ 14,13 SAY STR(EXTRA, 3)
@ 14,18 SAY "EXCESS QUOTAS."
ENDCASE

@ 15,24 SAY "SELECT ONE OPTION"
@ 16,18 SAY "A...START AGAIN WITH ORIGINAL QUOTAS"
@ 17,18 SAY "B...RETURN TO STANDARD REPORT;
SELECTION MENU"
@ 18,18 SAY "C...PERMANENTLY REPLACE ORIGINAL QUOTAS"
ACCEPT "SELECT ONE ==> " TO WHICHWAY

DC CASE
CASE !(WHICEWAY) = 'B'
STORE F TO PICKMNTH
CASE !(WHICEWAY) = 'C'
ACCEPT "DO YOU WANT TO PERMANENTLY REPLACE THE;
ORIGINAL SET OF QUOTAS (Y/N)?" TO CONTINUE
IF CONTINUE
DO CASE
CASE !(MUNIT) = 'A'
SAVE TO A:BRNGQTA ALL LIKE Q???
CASE !(MUNIT) = 'B'
SAVE TO A:BRNGQTA ALL LIKE Q???
CASE !(MUNIT) = 'C'
SAVE TO A:BRNGQTA ALL LIKE Q???
CASE !(MUNIT) = 'D'
SAVE TO A:BRNGQTA ALL LIKE Q???
CASE !(MUNIT) = 'E'

146
SAVE TC A:BRNGQTA ALL LIKE Q???
ENDCASE
STORE F TO PICKMNTH
ELSE
STORE F TO PICKMNTH
ENDIF
ENDCASE

CASE MCNTH = '06'
DO CASE
CASE ! (MUNIT) = 'A'
RESTORE PFCM A:ABNGCTA ADDITIVE
CASE ! (MUNIT) = 'B'
RESTORE PFCM A:ABNGCTA ADDITIVE
CASE ! (MUNIT) = 'C'
RESTORE PFCM A:CRNGCTA ADDITIVE
CASE ! (MUNIT) = 'D'
RESTORE PFCM A:DRNGCTA ADDITIVE
CASE ! (MUNIT) = 'E'
RESTORE PFCM A:BRNGQTA ADDITIVE
ENDCASE

ERASE
@ 2, 5 SAY "RIFLE RANGE QUOTA SCRATCH PAD"
@ 4 SAY "SPACE BELOW THIS LINE IS FOR:
CALCULATIONS AND WILL BE SAVED"
@ 5, 5 SAY "ONLY IF YOU SPECIFY THE 'REPLACE;
CURRENT QUOTAS' OPTION"
@ 6, 0 SAY "**********************************************************************
*****************************************************************************
" @ 8, 5 SAY " JUL;
@ 10, 34 GET CJUL PICTURE '999'
@ 10, 40 GET CAUG PICTURE '999'
@ 10, 46 GET CCSE PICTURE '999'
@ 10, 52 GET CCCT PICTURE '999'
@ 10, 58 GET CJCW PICTURE '999'
@ 10, 64 GET CECC PICTURE '999'
BEAD
@ 12, 4 SAY "A TOTAL OF 
STORE (CJUL+CAUG+CSE+CCCT+CJW+CECC) TO TEMPSUM
@ 12, 15 SAY STR (TEMPSUM, 3)
@ 12, 19 SAY "QUOTAS REMAIN FOR THE CURRENT YEAR."
LO CASE
CASE (TEMPSUM = MLEFT)
@ 14, 4 SAY "QUOTAS EQUAL NUMBER OF MARINES;
LEFT TO QUALIFY..."
@ 14, 5 SAY STR (TEMPSUM, 3)
CASE (MLEFT > TEMPSUM)
@ 14, 4 SAY "YOU NEED 
STORE (MLEFT - TEMPSUM) TO NEEDED
@ 14, 13 SAY STR (NEEDED, 3)
@ 14, 19 SAY "MORE QUOTAS."
CASE (TEMPSUM > MLEFT)
@ 14, 4 SAY "YOU HAVE 
STORE (TEMPSUM - MLEFT) TO EXTRA
@ 14, 13 SAY STR (EXTRA, 3)
@ 14, 18 SAY "EXCESS QUOTAS."
ENDCASE
@ 15, 24 SAY "SELECT ONE OPTION"
@ 16, 18 SAY "A...START AGAIN WITH ORIGINAL QUOTAS"
@ 17, 18 SAY "E...RETURN TO STANDARD REPORT;
SELECTION MENU"
18
SAY "C...PERMANENTLY REPLACE ORIGINAL QUOTAS"
ACCEPT "SELECT ONE =/>" TO WHICH WAY

DC CASE
CASE I(WHICHWAY) = 'B'
STORE F TO PICMNTH
CASE I(WHICHWAY) = 'C'
ACCEPT "DO YOU WANT TO PERMANENTLY REPLACE THE:
ORIGINAL SET OF QUOTAS (Y/N)?"; TO CONTINUE
IF CONTINUE
DO CASE
CASE I(MUNIT) = 'A'
SAVE TO A:ARNGQTAS ALL LIKE Q???
CASE I(MUNIT) = 'B'
SAVE TO A:BRNGQTAS ALL LIKE Q???
CASE I(MUNIT) = 'C'
SAVE TO A:CRNGQTAS ALL LIKE Q???
CASE I(MUNIT) = 'D'
SAVE TO A:DRNGQTAS ALL LIKE Q???
CASE I(MUNIT) = 'E'
SAVE TO A:BNRNGQTAS ALL LIKE Q???
ENDCASE
STORE F TO PICMNTH
ELSE
STORE F TO PICMNTH
ENDIF
ENDCASE
ENDDO
CASE MCNTE = '07'
DO WHILE PICMNTH = T
TO CASE
CASE I(MUNIT) = 'A'
RESTORE FROM A:ARNGQTAS ADDITIVE
CASE I(MUNIT) = 'B'
RESTORE FROM A:BRNGQTAS ADDITIVE
CASE I(MUNIT) = 'C'
RESTORE FROM A:CRNGQTAS ADDITIVE
CASE I(MUNIT) = 'D'
RESTORE FROM A:DRNGQTAS ADDITIVE
CASE I(MUNIT) = 'E'
RESTORE FROM A:BNRNGQTAS ADDITIVE
ENDCASE
ERASE
@ 2,15 SAY "FILE RANGE QUOTA SCRATCH PAD"
@ 4,5 SAY "SPACE BELOW THIS LINE IS FOR:
CALCULATIONS AND WILL BE SAVED"
@ 5,5 SAY "ONLY IF YOU SPECIFY THE 'REPLACE:
CURRENT QUOTAS' OPTION"
@ 6,0 SAY "************************************************************************
*************************************************************************
@ 8,5 SAY "AUG SEP CCT NOV DEC"
@ 10,40 GET CAUG PICTURE '999'
@ 10,46 GET CSEP PICTURE '999'
@ 10,52 GET Cpecting PICTURE '999'
@ 10,58 GET CCCT PICTURE '999'
@ 10,64 GET CNOV PICTURE '999'
@ 12,14 SAY "A TOTAL OF"
STORE (CAUG+CSEP+CCT+CNOV+QDEC) TO TEMPSUM
@ 12,15 SAY "THE QUOTAS REMAIN FOR THE CURRENT YEAR."
DC CASE
CASE I(TEMPSUM = MLLEPT)
@ 14,4 SAY "QUOTAS EQUAL NUMBER OF MARINES;"
LEFT TO QUALIFY...
CASE (LEFT > TEMPSUM)
  @ 14.53 SAY "YOU NEED" 
  STORE (LEFT - TEMPSUM) TO NEEDED
  @ 14.13 SAY STR(NEEDED, 3)
  @ 14.19 SAY "MORE QUOTAS."
CASE (TEMPSUM > LEFT)
  @ 14.4 SAY "YOU HAVE" 
  STORE (TEMPSUM - LEFT) TO EXTRA
  @ 14.13 SAY STR(EXTRA, 3)
  @ 14.18 SAY "EXCESS QUOTAS."
ENDCASE

@ 15.24 SAY "SELECT ONE OPTION"
@ 16.18 SAY "A...START AGAIN WITH ORIGINAL QUOTAS"
@ 17.18 SAY "B...RETURN TO STANDARD REPORT:
SELECTION MENU"
@ 18.18 SAY "C...PERMANENTLY REPLACE ORIGINAL QUOTAS"
ACCEPT "SELECT ONE ====> " TO WHICH WAY
DO CASE
CASE I (WHICHEWAY) = 'A'
  STORE F TO PICKMTH
CASE I (WHICHEWAY) = 'C'
  ACCEPT "EC YOU WANT TO PERMANENTLY REPLACE THE:
ORIGINAL SET OF QUOTAS (Y/N)?"; TO CONTINUE
IF CONTINUE
  DO CASE
  CASE I (MUNIT) = 'A'
    SAVE TO A:ARNGQTAS ALL LIKE Q???
  CASE I (MUNIT) = 'B'
    SAVE TO A:BRNGQTAS ALL LIKE Q???
  CASE I (MUNIT) = 'C'
    SAVE TO A:CRNGQTAS ALL LIKE Q???
  CASE I (MUNIT) = 'D'
    SAVE TO A:DRNGQTAS ALL LIKE Q???
  CASE I (MUNIT) = 'E'
    SAVE TO A:ENFNGQTAS ALL LIKE Q???
ENDCASE
  ELSE
    STORE F TO PICKMTH
  ENDIF
ENDIF
END CASE

ENDDC
CASE MNT = '08'
DO WHILE PICKMTH = T
  TO CASE
  CASE I (MUNIT) = 'A'
    RESTORE FROM A:ARNGQTAS ADDITIVE
  CASE I (MUNIT) = 'B'
    RESTORE FROM A:BRNGQTAS ADDITIVE
  CASE I (MUNIT) = 'C'
    RESTORE FROM A:CRNGQTAS ADDITIVE
  CASE I (MUNIT) = 'D'
    RESTORE FROM A:DRNGQTAS ADDITIVE
  CASE I (MUNIT) = 'E'
    RESTORE FROM A:ENFNGQTAS ADDITIVE
ENDCASE
ERASE
@ 4.15 SAY "SINGLE RANGE QUOTA SCRATCH PAD"
@ 4.55 SAY "SPACE BELOW THIS LINE IS FOR:
CALCULATIONS AND WILL BE SAVED"
@ 5.5 SAY "ONLY IF YOU SPECIFY THE 'REPLACE;
CURRENT QUOTAS' OPTION"
@ 6.0 SAY "***************************************************************************"
**********
12.4 SAY "A TOTAL OF "
STORE (QSEP+CCCT+QNCV+QDEC) TO TEMPSUM
12.15 SAY SIR (TEMPSUM, 3)
12.19 SAY "QUOTAS REMAIN FOR THE CURRENT YEAR."

DO CASE
  CASE (TEMPSUM = MLEFT)
    14.4 SAY "QUOTAS EQUAL NUMBER OF MARINES;"
    14.5 SAY "LEAVE TO QUALIFY..."
  CASE (MLEFT > TEMPSUM)
    14.4 SAY "YOU NEED"
    STORE (MLEFT - TEMPSUM) TO NEEDED
    14.13 SAY STR (NEEDED, 3)
    14.19 SAY "MORE QUOTAS."
  CASE (TEMPSUM > MLEFT)
    14.4 SAY "YOU HAVE"
    STORE (TEMPSUM - MLEFT) TO EXTRA
    14.13 SAY STR (EXTRA, 3)
    14.18 SAY "EXCESS QUOTAS."
ENDCASE

15.24 SAY "SELECT ONE OPTION"
16.18 SAY "A...START AGAIN WITH ORIGINAL QUOTAS"
17.18 SAY "E...RETURN TO STANDARD REPORT;"
SFICATION MENU"
18.18 SAY "C...PERMANENTLY REPLACE ORIGINAL QUOTAS"
ACCEPT "SELECT ONE ===> " TO WHICHWAY

DO CASE
  CASE !( WHICHWAY ) = 'B'
    STORE F TO PICKMTH
  CASE !( WHICHWAY ) = 'C'
    ACCEPT "DO YOU WANT TO PERMANENTLY REPLACE THE;"
    ORIGINAL SET OF QUOTAS (Y/N) ?" TO CONTINUE
    IF CONTINUE
      DO CASE
        CASE !( MUNIT ) = 'A'
          SAVE TO A:ARNGQTAS ALL LIKE Q???
        CASE !( MUNIT ) = 'B'
          SAVE TO A:BRNGQTAS ALL LIKE Q???
        CASE !( MUNIT ) = 'C'
          SAVE TO A:CRNGQTAS ALL LIKE Q???
        CASE !( MUNIT ) = 'D'
          SAVE TO A:DRNGQTAS ALL LIKE Q???
        CASE !( MUNIT ) = 'E'
          SAVE TO A:BNRNGQTAS ALL LIKE Q???
      ENDCASE
      STORE F TO PICKMTH
    ELSE
      STORE F TO PICKMTH
    ENDIF
  ENDCASE
ENDDO

CASE MONTF = '09'
  DO WHILE PICKMTH = 'T'
    TO CASE
      CASE !( MUNIT ) = 'A'

150
RESTORE FROM A:APNGQTAS ADDITIVE
CASE (MUNIT) = 'B'
  RESTORE FROM A:APNGQTAS ADDITIVE
CASE (MUNIT) = 'C'
  RESTORE FROM A:CRNGQTAS ADDITIVE
CASE (MUNIT) = 'D'
  RESTORE FROM A:DRNGQTAS ADDITIVE
CASE (MUNIT) = 'E'
  RESTORE FROM A:BNPNGQTA ADDITIVE
ENDCASE

CASE
  2.15 SAY "FIRE RANG QT A SCRATCH PAD"
  4.5 SAY "SPACE BELOW HI LINE IS FOR:
      CALCULATIONS AND WILL BE SAVED"
  5.5 SAY "ONLY IF YOU SPECIFY THE REPLACE:
      CURRENT QUOTAS" OPTION"
  6.0 SAY "*********************************
      AS ABOVE**************
  8.5 SAY ";
  11.5 GET (CCT PICTURE '999')
  13.5 GET (CNV PICTURE '999')
  14.6 GET (DEC PICTURE '999')
REAL
  12.4 SAY "A TOTAL OF"
  14.15 SAY TEMP SUM TO TEMP SUM
  12.19 SAY "QUOTAS REMAIN FOR THE CURRENT YEAR."
LO CASE
  CASE (TEMP SUM = MLEFT)
    @ 14.4 SAY "QUOTAS EQUAL NUMBER OF MARINES;
      LEFT TO QUALIFY..."
    @ 14.5 SAY TEMP SUM TO (TEMP SUM, 3)
  CASE (MLEFT > TEMP SUM)
    @ 14.4 SAY "YOU NEED "
    STORE (MLEFT - TEMP SUM) TO NEEDED
    @ 14.13 SAY STR(NEEDED, 3)
    @ 14.19 SAY "MORE QUOTAS."
  CASE (TEMP SUM > MLEFT)
    @ 14.4 SAY "YOU HAVE"
    STORE (TEMP SUM - MLEFT) TO EXTRA
    @ 14.13 SAY STR(EXTRA, 3)
    @ 14.18 SAY "EXCESS QUOTAS."
ENDCASE
  15.24 SAY "SELECT ONE OPTION"
  16.18 SAY "A.. START AGAIN WITH ORIGINAL QUOTAS"
  17.18 SAY "E.. RETURN TO STANDARD REPORT;
      SELECTION MENU"
  18.18 SAY "C.. PERMANENTLY REPLACE ORIGINAL QUOTAS"
ACCEPT "SELECT ONE ==> " TO WHICH WAY
DO CASE
  CASE (WHICHWAY) = 'B'
    STORE F TO PICKMETH
  CASE (WHICHWAY) = 'A'
    IF CO CASE
      CASE (MUNIT) = 'A'
        SAVE TO A:APNGQTAS ALL LIKE Q???
      CASE (MUNIT) = 'B'
        SAVE TO A:APNGQTAS ALL LIKE Q???
      CASE (MUNIT) = 'C'
        SAVE TO A:CRNGQTAS ALL LIKE Q???
      CASE (MUNIT) = 'D'
        SAVE TO A:DRNGQTAS ALL LIKE Q???
      CASE (MUNIT) = 'E'
        SAVE TO A:BNPNGQTA ALL LIKE Q???
      ENDCASE
CASE (MUNIT) = 'E'  
SAVE TO A:BNRNGQTA ALL LIKE Q???
ENDCASE
STORE F TO PICKMTH
ELSESTORE F TO PICKMTH
ENDIF
ENDCASE
ENDDO

CASE MTHNM = '10'
DO WHILE PICKMTH = T
DO CASE
CASE (MUNIT) = 'A'
RESTORE PECM A:ABNGCTAS ADDITIVE
CASE (MUNIT) = 'B'
RESTORE PECM A:BRNGCTAS ADDITIVE
CASE (MUNIT) = 'C'
RESTORE PECM A:CBRNGCTAS ADDITIVE
CASE (MUNIT) = 'D'
RESTORE PECM A:DBNGCTAS ADDITIVE
CASE (MUNIT) = 'E'
RESTORE PECM A:BNRNGQTA ADDITIVE
ENDCASE
IF CASE
  ERASE a 2,15
  SAY "BIFLE RANGE QUOTA SCRATCH PAD"
  ERASE a 4,5
  SAY "SPACE BELOW THIS LINE IS FOR CULTIVATIONS AND WILL BE SAVED"
  ERASE a 5,6
  SAY "ONLY IF YOU SPECIFY THE 'REPLACE CURRENT QUOTAS' OPTION"
  ERASE a 6,7
  SAY "*****************************************************************************"
  ERASE a 8,8
  SAY "************

NOV DEC

10,58 GET CNNV PICTURE '999';
10,64 GET CDEC PICTURE '999';
READ

a 1,4
SAY "A TOTAL OF ":
STOR (QNOV+CDEC) TC TEMPSUM
a 12,15
SAY STR(TEMPSUM,3)
SAY "QUOTAS REMAIN FOR THE CURRENT YEAR."
DO CASE
CASE (TEMPSUM = MLEFT)
  a 4,4
  SAY "QUOTAS EQUAL NUMBER OF MARINES; LEFT TO QUALIFY..."
  a 14,53
  SAY STR(TEMPSUM,3)
CASE (MLEFT > TEMPSUM)
  a 4,4
  SAY "YOU NEED ":
  a 14,13
  SAY STR(MLEFT - TEMPSUM,3)
  a 14,19
  SAY "MORE QUOTAS."
CASE (TEMPSUM > MLEFT)
  a 4,4
  SAY "YOU HAVE ":
  a 14,13
  SAY STR(TEMPSUM - MLEFT,3)
  a 14,18
  SAY "EXCESS QUOTAS."
ENDCASE
a 15,24
SAY "SELECT ONE OPTION"
16,18
SAY "A...START AGAIN WITH ORIGINAL QUOTAS"
17,18
SAY "E...RETURN TO STANDARD REPORT"
SELECTION MENU
18,18
SAY "C...PERMANENTLY REPLACE ORIGINAL QUOTAS"
ACCEPT "SELECT ONE ==> " TO WHICHWAY
LC CASE
CASE 1 (WHICHWAY) = 'B'
STORE I TC PICKMNTH
CASE 1 (WHICHWAY) = 'C'
ACCEPT "DO YOU WANT TO PERMANENTLY REPLACE THE
ORIGINAL SET OF QUOTAS (Y/N)?" TO CONTINUE
IF CONTINUE
DO CASE
CASE 1 (MUNIT) = 'A'
SAVE TO A:ARNGQTAS ALL LIKE Q???
CASE 1 (MUNIT) = 'B'
SAVE TO A:BRNGQTAS ALL LIKE Q???
CASE 1 (MUNIT) = 'C'
SAVE TO A:CRNGQTAS ALL LIKE Q???
CASE 1 (MUNIT) = 'D'
SAVE TO A:DRNGQTAS ALL LIKE Q???
CASE 1 (MUNIT) = 'E'
SAVE TO A:BRNGQTQTA ALL LIKE Q???
ENDCASE
ELSE
STORE F TO PICKMNTH
ENDIF
ENDCASE
ENDDC
CASE MCNTF = '11'
DO WHILE PICKMNTH = T
TO CASE
CASE 1 (MUNIT) = 'A'
RESTORE PFCM A:ARNGQTAS ADDITIVE
CASE 1 (MUNIT) = 'B'
RESTORE PFCM A:BRNGQTAS ADDITIVE
CASE 1 (MUNIT) = 'C'
RESTORE PFCM A:CRNGQTAS ADDITIVE
CASE 1 (MUNIT) = 'D'
RESTORE PFCM A:DRNGQTAS ADDITIVE
CASE 1 (MUNIT) = 'E'
RESTORE PFCM A:BRNGQTQTA ADDITIVE
ENDCASE
ERASE
@ 2,15 SAY "FIRE RANGE QUOTA SCRATCH PAD"
@ 4,5 SAY "SPACE BELOW THIS LINE IS FOR:
CALCULATIONS AND WILL BE SAVED"
@ 5,5 SAY "ONLY IF YOU SPECIFY THE 'REPLACE;
CURRENT QUOTAS' OPTION"
@ 6,0 SAY "******************************************************************;
************"
@ 8,5 SAY :
@ 10,64 GET CIEC PICTURE '999'
FEAL
@ 12,4 SAY "A TOTAL OF"
STORE (QEB+ECHAR+QAPF+QWAY+QJUN+QJUL+;
CAUC+QSEP+QQCI+QHOV+CIEC) TO TEMPSUM
@ 12,15 SAY STR(TEMPSUM,3)
@ 14,19 SAY "QUOTAS REMAIN FOR THE CURRENT YEAR."
DO CASE
CASE (TEMPSUM = MLEFT)
@ 14,4 SAY "QUOTAS EQUAL NUMBER OF MARINES;
LEFT TO QUALIFY...
"@ 14,53 SAY STR(TEMPSUM,3)
CASE (MLEFT > TEMPSUM)
@ 14,4 SAY "YOU NEED"
STORE (MLEFT - TEMPSUM) TO NEEDED
@ 14,13 SAY STR(NEEDED,3)
@ 14, 19 SAY "MORE QUOTAS."
CASE (TEMSUE > MLEFT)
  @ 14, 4 SAY "YOU HAVE"
  STORE (TEMSUM - MLEFT) TO EXTRA
  @ 14, 13 SAY STEM(EXTRA, 3)
  @ 14, 18 SAY "EXCESS QUOTAS."
ENDCASE

@ 15, 24 SAY "SELECT ONE OPTION"
@ 16, 18 SAY "A...START AGAIN WITH ORIGINAL QUOTAS"
@ 17, 18 SAY "E...RETURN TO STANDARD REPORT;
SELECTION MENU"
@ 18, 18 SAY "C...PERMANENTLY REPLACE ORIGINAL QUOTAS"
ACCEPT "SELECT ONE ==> " TO WHICH WAY

DC CASE
  CASE I((WHICEWAY) = 'B')
  STORE F TO PICKMNTH
  CASE I((WHICEWAY) = 'C')
  
  ACCEPT "DO YOU WANT TO PERMANENTLY REPLACE THE ORIGI
 ナルセットOF QUOTAS (Y/N)?" TO CONTINUE
  IF CONTINUE
  DO CASE
    CASE I(MUNIT) = 'A'
    SAVE TO A:ARNGQTAS ALL LIKE Q??
    CASE I(MUNIT) = 'B'
    SAVE TO A:BREGQTAS ALL LIKE Q??
    CASE I(MUNIT) = 'C'
    SAVE TO A:CRNGQTAS ALL LIKE Q??
    CASE I(MUNIT) = 'D'
    SAVE TO A:DRNGQTAS ALL LIKE Q??
    CASE I(MUNIT) = 'E'
    SAVE TO A:BRNGQTAS ALL LIKE Q??
  ENDCASE
  STORE F TO PICKMNTH
  ELSE
  STORE F TO PICKMNTH
  ENDIF
ENDIF

INICASE

ENDDC

CASE MONTH = '12'
ERASE
  @ 4, 5 SAY "RIPLE RANGE QUOTA SCRATCH PAD"
  @ 5, 5 SAY "SPACE BELOW THIS LINE IS FOR CALCULATIONS;
  AND WILL BE SAVIE"
  @ 6, 5 SAY "ONLY IF YOU SPECIFY THE REPLACE CURRENT;
  QUOTAS' OPTION"
  @ 7, 5 SAY "******************************
  *********************
  END CASE"

ENDIF CASE

ENDCASE

SET CLUC ON
RELEASE ALL LIKE Q??
RELEASE MLEFT, TEMPSUE, WHICH WAY, PICKMNTH, CONTINUE
RETURN
II. MARKSMANSHIP ROSTER ROUTINE

* ROUTINE NAME: MKESRCT.PRG
* MODULE NAME: STANDARD REPORT GENERATOR
* VERSION: 5.7.0
* AUTHOR: F. E. TRUETT
* DATE: 28 DEC 83
* VARIABLES USED: MUNIT, MPLAT
* VARIABLES MODIFIED: NONE
* VARIABLES CREATED: NONE
* VARIABLES RELEASED: MUNIT, MPLAT
* FILES OPENED/CLOSED: PERS, QUAL (ALL FILES CLOSED)
* TIME FILES MODIFIED: MKESTDAT (CREATED AND DELETED)
* USING SUBROUTINES: SDRPT.PRG
* DESCRIPTION: THIS ROUTINE CREATES A TEMPORARY FILE WHICH
* INCLUDES NAME, RFICUAL, RFLDATE, RSTQUAL, RSTDATE FOR
* EACH MEMBER IN THE SELECTED UNIT. A STANDARD REPORT
* FORMAT (B:MRKSRPT) IS USED TO DISPLAY THE "MARKSMANSHIP
* ROSTER REPORT".

USE PERS
SELECT SECONDARY
USE QIAL
JOIN TO MKESTDATA FOR E:SSN = S:SSN FIELD;
NAME, RFICUAL, RFLDATE, RSTQUAL, RSTDATE
USE MKESTDATA
ERASE
DO CASE
CASE (MUNIT) = 'A'
  REPORT FORM B: MKEKSRPT
CASE (MUNIT) = 'B'
  REPORT FORM B: MKEKSRPT FOR COMPANY = ONECO
CASE (MUNIT) = 'C'
  REPORT FORM B: MKEKSRPT FOR COMPANY = TWOCO
CASE (MUNIT) = 'D'
  REPORT FORM B: MKEKSRPT FOR COMPANY = THREECO
CASE (MUNIT) = 'E'
  REPORT FORM B: MKEKSRPT FOR COMPANY = FOURECO
CASE 1 (MUNIT) = 'E' .AND. !(MELAT) = 'E'
ENGINE FORM B:MRKSRPT FOR COMPANY = THREECO .AND.;
FLATCCN = THREEFLT
CASE 1 (MUNIT) = 'E' .AND. !(MELAT) = 'A'
ENGINE FORM B:MRKSRPT FOR COMPANY = FOURCO.
CASE 1 (MUNIT) = 'E' .AND. !(MELAT) = 'B'
ENGINE FORM B:MRKSRPT FOR COMPANY = FOURCO .AND.;
FLATCCN = CNEFLT
CASE 1 (MUNIT) = 'E' .AND. !(MELAT) = 'C'
ENGINE FORM B:MRKSRPT FOR COMPANY = FOURCO .AND.;
FLATCCN = TWOFLT
CASE 1 (MUNIT) = 'E' .AND. !(MELAT) = 'D'
ENGINE FORM B:MRKSRPT FOR COMPANY = FOURCO .AND.;
FLATCCN = THREEFLT
CASE 1 (MUNIT) = 'E' .AND. !(MELAT) = 'E'
ENGINE FORM B:MRKSRPT FOR COMPANY = FOURCO .AND.;
FLATCCN = FOURFLT
ENDCASE
USE
RELEASE MUNIT, MELAT
DELETE FILE MAKSDATA
RETURN
N. MCS STATUS ROUTINE

* ROUTINE NAME: MOSSTAT.PRG
* MODULE NAME: STANDARD REPORT GENERATOR
* VERSION: 5.8.0
* AUTHOR: R.E. PRUITI
* DATE: NOV86
* VARIABLES USED: ONECO, TWOQO, THREECO, FOURCO, ONEPLT,
  * TWOPLT, THREEPLT, FCURPLT, MUNIT, MPLAT, UNITNAME,
  * PLATNAME
* VARIABLES MODIFIED: NONE
* VARIABLES CREATED: COUNT, MOS, RK, PROCEED
* VARIABLES RELEASED: PROCEED, RQW, RK, COUNT, UNITNAME,
  * PLATNAME
* FILES CLOSED/CLOSED: A:PEBS INDEX A:MOSENDX (ALL FILES
  * CLOSED)
* FILES CREATED: NONE
* USING ROUTINES: STLSPT.PRG
* DESCRIPTION: THIS ROUTINE PRODUCES THE "MOS STATUS
  * REPORT". EACH TYPE OF MOS IN THE SPECIFIED UNIT IS
  * COUNTED AND TOTALED, REGARDLESS OF THE TYPE OR NUMBER OF
  * DIFFERENT MOS'S IN THE PARTICULAR UNIT. THE RESULTS ARE
  * THEN DISPLAYED BY RCS AND BY RANK.

ERASE
LO CVUNIT
USE A:PEBS INDEX A:MOSENDX
4 0.20 SAY "MOS STATUS REPORT"
IF I (MUNIT)='E'
  @ 0.38 SAY "BATTALION"
ENDIF
IF I (MUNIT) <> 'E'
  @ 0.38 SAY UNITNAME
ENDIF
IF I (MPLAT)<> 'P'. AND. I (MUNIT) <> 'E'
  @ 0.46 SAY PLATNAME
ENDIF
3 2.10 SAY "MOS"
3 2.30 SAY "RANK"
3 2.50 SAY "TOTAL"
STORE 4 TO EW
LO WHILE NOT. EOF
STORE PRIMEMOS TO E0
STORE RANK TO RK
STORE 0 TO COUNT
LC CASE
  CASE I (MUNIT)='E'
    DO WHILE PRIMEMOS=MOS .AND. RANK=RK .AND. .NOT.;
    ECP
    STORE COUNT+1 TO CCCUNT
    SKIP
  ENDCASE
  CASE I (MPLAT)='E'
    DC WHILE PRIMEMOS=MOS .AND. RANK=RK .AND. .NOT.;
    ECP
    IP COMPANY=UNITNAME
    STORE COUNT+1 TO COUNT
    ENDIF
    SKIP
  ENDCASE
  OTHERWISE
    DC WHILE PRIMEMOS=MOS .AND. RANK=RK .AND. .NOT.;
    ECP
    IP COMPANY=UNITNAME .AND. PLATCON=PLATNAME
    STORE COUNT+1 TO COUNT
    ENDIF
    SKIP
  ENDCASE
ENDO
157
ENCASE
 IF CCINT > 0
   @ ROW, 30 SAY MCS
   @ ROW, 30 SAY RK
   @ FCW, 50 SAY CCINT
   STORE ROW+2 TO FCW

ENDIF
 IF ROW > 20
   @ 23, 0
   ACCEPT "PRESS 'ENTER' TO CONTINUE" TO PROCEED
   STORE 4 TO ROW
   @ 0, 20 SAY "MOS STATUS REPORT"
   IF ! (MUNIT) = 'E'
     @ 0, 38 SAY "FATTALION"
   ENDIF
   IF !(MUNIT) <> 'F'
     @ 0, 38 SAY UNITNAME
   ENDIF
   IF !(MUNIT) <> 'E', AND !(MUNIT) <> 'E'
     @ 0, 46 SAY PLATNAME
   ENDIF
   @ 2, 10 SAY "MOS"
   @ 2, 30 SAY "RANK"
   @ 2, 55 SAY "TOTAL"
 ENDIF
 ENDDC
 RELEASE PROCEED, ROW, ECS, RK, COUNT, UNITNAME, PLATNAME
 USE
 RETURN
C. MCS FOSTER ROUTINE

* ROUTINE NAME: MOSRCST.PRG
* MODULE NAME: STANDARD REPORT GENERATOR
* VERSION: 5.9.0
* AUTHOR: R. E. PRUITT
* DATE: 6 NOV 83
* VARIABLES USED: MUNIT, MPLAT, ONECO, TWO CO, THREE CO,
  FOUR CO, CNEPIT, TWECFT, THREEPLT, FOURPLT
* VARIABLES MODIFIED: NONE
* VARIABLES CREATED: NONE
* VARIABLES RELEASED: MUNIT, MPLAT
* FILES (OPENED/CLOSED): A: PERS INDEX A: MOSNDX
* TEMP FILES CREATED: NONE
* USING SUBROUTINES: STDRPT.PRG
* DESCRIPTION: THIS IS A CONTROL ROUTINE WHICH CREATES THE
  PERSONNEL ROSTER FOR THE UNIT SPECIFIED BY THE USER'S
  RESPONSE TO THE SLCTUNIT MENU (5.1.0).
* ERASE
* USE A: PERS INDEX A: MOSNDX
* DO CASE
  CASE ! (MUNIT) = 'E'
    RE FOR T FORM B: MCSR PT
  CASE ! (MUNIT) = 'A' AND. ! (MPLAT) = 'E'
    RE FOR T FORM B: MCSR PT FOR COMPANY = ONE CO
  CASE ! (MUNIT) = 'A'. AND. ! (MPLAT) = 'A'
    RE FOR T FORM B: MCSR PT FOR COMPANY = ONECO .AND.;
    ELATCCN = ONEP IT
  CASE ! (MUNIT) = 'B'. AND. ! (MPLAT) = 'E'
    RE FOR T FORM B: MCSR PT FOR COMPANY = TWO CO .AND.;
    ELATCCN = TWOP IT
  CASE ! (MUNIT) = 'B'. AND. ! (MPLAT) = 'A'
    RE FOR T FORM B: MCSR PT FOR COMPANY = TWO CO .AND.;
    ELATCCN = THREEP IT
  CASE ! (MUNIT) = 'C'. AND. ! (MPLAT) = 'E'
    RE FOR T FORM B: MCSR PT FOR COMPANY = TWO CO .AND.;
    ELATCCN = THREEP IT
  CASE ! (MUNIT) = 'C'. AND. ! (MPLAT) = 'A'
    RE FOR T FORM B: MCSR PT FOR COMPANY = THREE CO .AND.;
    ELATCCN = CNEP IT
  CASE ! (MUNIT) = 'C'. AND. ! (MPLAT) = 'C'
    RE FOR T FORM B: MCSR PT FOR COMPANY = THREE CO .AND.;
    ELATCCN = THREEP IT
  CASE ! (MUNIT) = 'D'. AND. ! (MPLAT) = 'C'
    RE FOR T FORM B: MCSR PT FOR COMPANY = THREE CO .AND.;
    ELATCCN = THREEP IT
  CASE ! (MUNIT) = 'D'. AND. ! (MPLAT) = 'A'
    RE FOR T FORM B: MCSR PT FOR COMPANY = FOUR CO
  CASE ! (MUNIT) = 'E'. AND. ! (MPLAT) = 'A'
BEFORE FORM B: FCSRPT FOR COMPANY = FOURCO . AND.;
FLATCCN = ONEPIT
CASE MUNIT = 'L' . AND. ! (MFLAT) = 'B'
BEFORE FORM B: FCSRPT FOR COMPANY = FOURCO . AND.;
FLATCCN = TWOPIT
CASE MUNIT = 'L' . AND. ! (MFLAT) = 'C'
BEFORE FORM B: FCSRPT FOR COMPANY = FOURCO . AND.;
FLATCCN = THREEILT
CASE MUNIT = 'L' . AND. ! (MFLAT) = 'D'
BEFORE FORM B: FCSRPT FOR COMPANY = FOURCO . AND.;
FLATCCN = FOURILT
ENDCASE
USE
RELEASE MUNIT, MFLAT
RETURN
I. PERSONNEL DATA ROUTINE

* ROUTINE NAME: PERSDATA.PRG
* MODULE NAME: STANDARD REPORT GENERATOR
* VERSION: 5.10.0
* AUTHOR: R. E. FRUITER
* DATE: 2 INCY83
* VARIABLES USED: MUNIT, MPLAT, ONECO, TWOPL, THREEPLT, FOURPLT
* VARIABLES MODIFIED: COUNT
* VARIABLES CREATED: COUNT, PROCEED
* VARIABLES RELEASED: COUNT, PROCEED
* FILES (OPENED/CLOSED): A: PERS INDEX A: ALPHPERS
* TEXT FILES CREATED: NONE
* USING SUBROUTINES: STDRPT.PRG
* DESCRIPTION: THIS IS A CONTROL ROUTINE WHICH CREATES THE
  PERSONNEL ROSTER FOR THE UNIT SPECIFIED BY THE USER'S
  RESPONSE TO THE SLCTUNIT MENU (5.1.0).

ERASE
USE A: PERS INDEX A: ALPHPERS
GOTO TOP
STORE 0 TO COUNT
DO WHILE .NOT. EOF
TO CASE
CASE 'E': TO PERSEPT
STORE COUNT + 1 TO COUNT
CASE 'A'.AND. !(MPLAT) = 'E'
  IF COMPANY = ONECO
     TO PERSEPT
     STORE COUNT + 1 TO COUNT
   ENDIF
CASE 'A'.AND. !(MPLAT) = 'A'
  IF COMPANY = ONECO .AND. PLATOON = ONEPLT
     TO PERSEPT
     STORE COUNT + 1 TO COUNT
   ENDIF
CASE 'A'.AND. !(MPLAT) = 'B'
  IF COMPANY = ONECO .AND. PLATOON = TWOPLT
     TO PERSEPT
     STORE COUNT + 1 TO COUNT
   ENDIF
CASE 'A'.AND. !(MPLAT) = 'C'
  IF COMPANY = ONECO .AND. PLATOON = THREEPLT
     TO PERSEPT
     STORE COUNT + 1 TO COUNT
   ENDIF
CASE 'A'.AND. !(MPLAT) = 'D'
  IF COMPANY = ONECO .AND. PLATOON = FOURPLT
     TO PERSEPT
     STORE COUNT + 1 TO COUNT
   ENDIF
CASE 'E'.AND. !(MPLAT) = 'E'
  IF COMPANY = TWOPL
     TO PERSEPT
     STORE COUNT + 1 TO COUNT
   ENDIF
CASE 'E'.AND. !(MPLAT) = 'A'
  IF COMPANY = TWOPL .AND. PLATOON = ONEPLT
     TO PERSEPT
     STORE COUNT + 1 TO COUNT
   ENDIF
CASE 'E'.AND. !(MPLAT) = 'B'
  IF COMPANY = TWOPL .AND. PLATOON = TWOPLT
     TO PERSEPT
     STORE COUNT + 1 TO COUNT
   ENDIF
161
CASE ! (MUNIT) = 'E' .AND. ! (MPLAT) = 'C'
  IF COMPANY = TCCC0 .AND. PLATOON = THREEPLT
  DO PERSRPT
  STORE COUNT + 1 TO CCUNT
  ENDP
CASE ! (MUNIT) = 'E' .AND. ! (MPLAT) = 'D'
  IF COMPANY = TCCC0 .AND. PLATOON = FOURPLT
  DO PERSRPT
  STORE COUNT + 1 TO CCUNT
  ENDP
CASE ! (MUNIT) = 'C' .AND. ! (MPLAT) = 'C'
  IF COMPANY = TCCC0 .AND. PLATOON = THREEPLT
  DO PERSRPT
  STORE COUNT + 1 TO CCUNT
  ENDP
CASE ! (MUNIT) = 'C' .AND. ! (MPLAT) = 'C'
  IF COMPANY = TCCC0 .AND. PLATOON = THREEPLT
  DO PERSRPT
  STORE COUNT + 1 TO CCUNT
  ENDP
CASE ! (MUNIT) = 'C' .AND. ! (MPLAT) = 'D'
  IF COMPANY = TCCC0 .AND. PLATOON = FOURPLT
  DO PERSRPT
  STORE COUNT + 1 TO CCUNT
  ENDP
CASE ! (MUNIT) = 'E' .AND. ! (MPLAT) = 'E'
  IF COMPANY = FCCUCO
  DO PERSRPT
  STORE COUNT + 1 TO CCUNT
  ENDP
CASE ! (MUNIT) = 'E' .AND. ! (MPLAT) = 'A'
  IF COMPANY = FCCUCO .AND. PLATOON = ONEPLT
  DO PERSRPT
  STORE COUNT + 1 TO CCUNT
  ENDP
CASE ! (MUNIT) = 'E' .AND. ! (MPLAT) = 'A'
  IF COMPANY = FCCUCO .AND. PLATOON = ONEPLT
  DO PERSRPT
  STORE COUNT + 1 TO CCUNT
  ENDP
CASE ! (MUNIT) = 'E' .AND. ! (MPLAT) = 'B'
  IF COMPANY = FCCUCO .AND. PLATOON = TWOPLT
  DO PERSRPT
  STORE COUNT + 1 TO CCUNT
  ENDP
CASE ! (MUNIT) = 'E' .AND. ! (MPLAT) = 'B'
  IF COMPANY = FCCUCO .AND. PLATOON = TWOPLT
  DO PERSRPT
  STORE COUNT + 1 TO CCUNT
  ENDP
CASE ! (MUNIT) = 'E' .AND. ! (MPLAT) = 'C'
  IF COMPANY = FCCUCO .AND. PLATOON = THREEPLT
  DO PERSRPT
  STORE COUNT + 1 TO CCUNT
  ENDP
CASE ! (MUNIT) = 'E' .AND. ! (MPLAT) = 'C'
  IF COMPANY = FCCUCO .AND. PLATOON = THREEPLT
  DO PERSRPT
  STORE COUNT + 1 TO CCUNT
  ENDP
CASE ! (MUNIT) = 'E' .AND. ! (MPLAT) = 'D'
  IF COMPANY = FCCUCO .AND. PLATOON = FOURPLT
  DO PERSRPT
  STORE COUNT + 1 TO CCUNT
  ENDP
ENDCASE
IF CCUNT = 4
  ID 22:0
  ACCEPT "PRESS 'ENTER' TO CONTINUE" TO PROCEED
  STORE 0 TO COUNT
ENDS
RELEASE COUNT, PROCEED
GSE
RETURN
Q. PERSONNEL DATA FORMAT ROUTINE

* ROUTINE NAME: PERSREPT.PRG
* MODULE NAME: STANDARD REPORT GENERATOR
* VERSION: 5.10.1.0
* DATE: 21 NOV 83
* VARIABLES USED: COUNT
* VARIABLES MODIFIED: NONE
* VARIABLES CREATED: NONE
* VARIABLES RELEASED: NONE
* FILES OPENED/CLOSED: NONE
* USING SUBROUTINES: FERSDATA.ERG
* DESCRIPTION: THIS IS A REPORT FORMAT COMMAND FILE WHICH
  * DISPLAYS THE FORMAT FOR THE PERSONAL DATA REPORT AND GETS
  * THE FIELD VARIABLES FROM THE PERS.DBF WHICH IS OPENED AND
  * CLOSED IN FERSDATA.ERG.

COUNT 6.0 SAY "NAME"
3.5 SAY NAME
4.2 SAY "SSN"
4.7 SAY SSN
5.0 SAY "RANK"
5.5 SAY RANK
5.2 SAY "PRIMARY MOS"
5.3 SAY PRIMEMOS
5.2 SAY "SECONDARY MOS"
7.0 SAY SECMOS
11.0 SAY "COMPANY"
8.8 SAY COMPANY
5.0 SAY "PLATOON"
3.0 SAY PLATOON
4.2 SAY "JOIN DATE"
5.3 SAY JOINDATE
6.2 SAY "EAS"
6.7 SAY EAS
11.0 SAY "BIRTHDATE"
7.7 SAY BIRTHDATE
9.2 SAY "HEIGHT"
9.2 SAY HEIGHT
12.2 SAY "WEIGHT"
5.0 SAY WEIGHT
6.2 SAY "COMMENTS"
RETURN
**EST STATUS ROUTINE**

* ROUTINE NAME: ESTSTAT.FRG
* MODIFIED NAME: STANDARD REPORT GENERATOR
* VERSION: 5.1.1.0
* AUTHOR: R. F. PRUITT
* DATE: 22DEC83
* VARIABLES USED: MUNIT, MPLAT, ONECO, TWOCO, THREECO,
  POURCO, CMPLT, TMWIT, THERELC, FOURPLT, ZZTOTAL, ZZCOC,
  ZZHIS, ZZCOD, ZZINT, ZZAI, ZZUNI, ZZNBC, ZZHKS, ZZTAC,
  ZZPFT1, ZZPFT2
* VARIABLES MODIFIED: NONE
* VARIABLES CREATED: PROCEED
* VARIABLES RELEASED: ZZ???????
* FILES (OPENED/CLOSED): E:EST, A:PERS (ALL FILES CLOSED)
* TEMP FILES CREATED: TEMP (CREATED AND DELETED)
* USING SUBROUTINES: STDRPT.FRG
* DESCRIPTION: THIS ROUTINE CREATES A TEMPORARY FILE THAT
  INCLUDES THE ALL ESSENTIAL SUBJECT AND PET FIELDS FROM
  THE EST. REP FOR MEMBERS IN A USER SPECIFIED UNIT. FRM
  THIS FILE IT COUNTS THE TOTAL NUMBER OF UNIT MEMBERS THAT
  HAVE COMPLETED ALL ELEMENTS IN EACH TRAINING CATEGORY.
  THE "EST STATUS REPORT" IS DISPLAYED WHICH INCLUDES
  TOTALS FOR EACH TRAINING CATEGORY AND THE PERCENTAGE
  COMPLETED FOR EACH TRAINING CATEGORY.

```fortran
ERASE
DO CASE
  CASE ! (MUNIT) = 'E'
    USE E:EST
    DC CNTEST
  CASE ! (MUNIT) = 'A'.AND. ! (MPLAT) = 'E'
    SELECT PRIMARY
    USE A:PERS
    SELECT SECCNDARY
    USE E:EST
    SELECT PRIMARY
    JOIN TO TEMP FCF (P.COMPA?Y=ONECO .AND. P.SSN=SSN):
      FIELDS COC,HIS,COD,INT,AID,UNI,PFT1CLSS,PFT2CLSS,
      NEC,NKS,TAC
      USE TEMP
      DC CNTET
      DELETE FILE TEMP
  CASE ! (MUNIT) = 'E'.AND. ! (MPLAT) = 'A'
    SELECT PRIMARY
    USE A:PERS
    SELECT SECCNDARY
    USE E:EST
    SELECT PRIMARY
    JOIN TO TEMP FCF (P.COMPA?Y=ONECO .AND. P.SPLATCCN =:
      CMPLT .AND. P.SSN=SSN) FIELDS COC,HIS,COD,INT,AID,
      UNI,PFT1CLSS,PFT2CLSS,NBC,NKS,TAC
      USE TEMP
      DC CNTET
      DELETE FILE TEMP
  CASE ! (MUNIT) = 'A'.AND. ! (MPLAT) = '5'
    SELECT PRIMARY
    USE A:PERS
    SELECT SECCNDARY
    USE E:EST
    SELECT PRIMARY
    JOIN TO TEMP FCF (P.COMPA?Y =ONECO .AND. P.PLATCCN=:
      TWOCO .AND. P.SSN=SSN) FIELDS COC,HIS,COD,INT,
      ALL,UNI,PFT1CLSS,PFT2CLSS,NBC,NKS,TAC
      USE TEMP
      DC CNTET
      DELETE FILE TEMP
  CASE ! (MUNIT) = 'A'.AND. ! (MPLAT) = 'C'
```
SELECT PRIMARY
USE A: PERS
SELECT SECONDARY
USE B: EST
SELECT PRIMARY
JOIN TO TEMP FCF (P.CCMFANY = ONECO , AND, P.PLA TCO : = THREEPLT . AND, P.SSN=S.SSN) FIELDS OCC, HIS, CCD, ; INT, AID, UNI, PFT1 CLSS, PFT2 CLSS, NBC, MKS, TAC
USE TEMP
DO C Ni EST
DELETE FILE TEMP
CASE ! (MUNIT) = ' 3 ' . AND. ! (MFLAT) = ' D'
SELECT PRIMARY
USE A: PERS
SELECT SECONDARY
USE B: EST
SELECT PRIMARY
JOIN TO TEMP FCF (P.CCMFANY = ONECO , AND, P.PPLATCO : = FOURPLT AND, P.SSN=S.SSN) FIELDS OCC, HIS, CCD, ; INT, AID, UNI, PFT1 CLSS, PFT2 CLSS, NBC, MKS, TAC
USE TEMP
DO C Ni EST
DELETE FILE TEMP
CASE ! (MUNIT) = ' 4 ' . AND. ! (MFLAT) = ' E'
SELECT PRIMARY
USE A: PERS
SELECT SECONDARY
USE B: EST
SELECT PRIMARY
JOIN TO TEMP FCF (P.CCMFANY = TWO CO , AND, P.PLA TCO : = FIVEPLT AND, P.SSN=S.SSN) FIELDS OCC, HIS, CCD, ; INT, AID, UNI, PFT1 CLSS, PFT2 CLSS, NBC, MKS, TAC
USE TEMP
DO C Ni EST
DELETE FILE TEMP
CASE ! (MUNIT) = ' 5 ' . AND. ! (MFLAT) = ' A'
SELECT PRIMARY
USE A: PERS
SELECT SECONDARY
USE B: EST
SELECT PRIMARY
JOIN TO TEMP FCF (P.CCMFANY = TWO CO , AND, P.PLA TCO : = SIXPLT AND, P.SSN=S.SSN) FIELDS OCC, HIS, CCD, ; INT, AID, UNI, PFT1 CLSS, PFT2 CLSS, NBC, MKS, TAC
USE TEMP
DO C Ni EST
DELETE FILE TEMP
CASE ! (MUNIT) = ' 6 ' . AND. ! (MFLAT) = ' B'
SELECT PRIMARY
USE A: PERS
SELECT SECONDARY
USE B: EST
SELECT PRIMARY
JOIN TO TEMP FCF (P.CCMFANY = TWO CO , AND, P.PLA TCO : = SEVENPLT AND, P.SSN=S.SSN) FIELDS OCC, HIS, CCD, ; INT, AID, UNI, PFT1 CLSS, PFT2 CLSS, NBC, MKS, TAC
USE TEMP
DO C Ni EST
DELETE FILE TEMP
CASE ! (MUNIT) = ' 7 ' . AND. ! (MFLAT) = ' C'
SELECT PRIMARY
USE A: PERS
SELECT SECONDARY
USE B: EST
SELECT PRIMARY
JOIN TO TEMP FCF (P.CCMFANY = TWO CO , AND, P.PLA TCO : = EIGHTPLT AND, P.SSN=S.SSN) FIELDS OCC, HIS, CCD, ; INT, AID, UNI, PFT1 CLSS, PFT2 CLSS, NBC, MKS, TAC
USE TEMP
DO CNTST
DELETE FILE TEMP
CASE 1 (MUNIT) = 'E' .AND. ! (MFLAT) = 'D'
  SELECT PRIMARY
  USE A: PERS
  SELECT SECONDARY
  USE B: EST
  SELECT PRIMARY
  JOIN TO TEMP PCE (P. COMPA NY = TWO CO . AND. P. PLATCON =:
  = TWO PL T . AND. F. SSN = S. SSN) FIELDS COC, HIS, COD, INT;
  AIC, UNI, PPT1CLS, PPT2CLS, NBC, MKS, TAC
  USE TEMP
  DO CNTST
  DELETE FILE TEMP
  CASE 1 (MUNIT) = 'C' .AND. ! (MFLAT) = 'E'
  SELECT PRIMARY
  USE A: PERS
  SELECT SECONDARY
  USE B: EST
  SELECT PRIMARY
  JOIN TO TEMP PCE (P. COMPA NY = THREE CO . AND. P. SSN =:
  = THREE PL T . AND. F. SSN = S. SSN) FIELDS COC, HIS, COD, INT;
  AIC, UNI, PPT1CLS, PPT2CLS, NBC, MKS, TAC
  USE TEMP
  DO CNTST
  DELETE FILE TEMP
  CASE 1 (MUNIT) = 'C' .AND. ! (MFLAT) = 'A'
  SELECT PRIMARY
  USE A: PERS
  SELECT SECONDARY
  USE B: EST
  SELECT PRIMARY
  JOIN TO TEMP PCE (P. COMPA NY = THREE CO . AND. P. PLATCON =:
  = THREE PL T . AND. F. SSN = S. SSN) FIELDS COC, HIS, COD, INT;
  AIC, UNI, PPT1CLS, PPT2CLS, NBC, MKS, TAC
  USE TEMP
  DO CNTST
  DELETE FILE TEMP
  CASE 1 (MUNIT) = 'C' .AND. ! (MFLAT) = 'B'
  SELECT PRIMARY
  USE A: PERS
  SELECT SECONDARY
  USE B: EST
  SELECT PRIMARY
  JOIN TO TEMP PCE (P. COMPA NY = THREE CO . AND. P. PLATCON =:
  = THREE PL T . AND. P. SSN = S. SSN) FIELDS COC, HIS, COD, INT;
  AIC, UNI, PPT1CLS, PPT2CLS, NBC, MKS, TAC
  USE TEMP
  DO CNTST
  DELETE FILE TEMP
  CASE 1 (MUNIT) = 'C' .AND. ! (MFLAT) = 'C'
  SELECT PRIMARY
  USE A: PERS
  SELECT SECONDARY
  USE B: EST
  SELECT PRIMARY
  JOIN TO TEMP PCE (P. COMPA NY = THREE CO . AND. P. PLATCON =:
  = THREE PL T . AND. P. SSN = S. SSN) FIELDS COC, HIS, COD, INT;
  AIC, UNI, PPT1CLS, PPT2CLS, NBC, MKS, TAC
  USE TEMP
  DO CNTST
  DELETE FILE TEMP
  CASE 1 (MUNIT) = 'C' .AND. ! (MFLAT) = 'D'
  SELECT PRIMARY
  USE A: PERS
  SELECT SECONDARY
  USE B: EST
  SELECT PRIMARY
  JOIN TO TEMP PCE (P. COMPA NY = THREE CO . AND. P. PLATCON =
= FOURPLT .AND. P.SSN=S.SSN) FIELDS COC, HIS, CCD, 11;
HIS, CCD, INT, AID, UNI, PFT1CLASS, PFT2CLASS, NEC, MKS, 1AC
USE TEMP
DO CNTEST
DELETE FILE TEMP
CASE 1 (MUNIT) = 'E' .AND. !(MPLAT) = 'E'
SELECT PRIMARY
USE A: PERS
SELECT SECCNDAF7
USE E: 18T
SELECT PRIMARY
JOIN TO TEMP PCK (P.COMPANY = FOURCO .AND. ;
F.SSN=S.SSN) FIELDS COC, HIS, CCD, INT, AID, UNI, PFT1CLASS, ;
PFT2CLASS, NEC, MKS, TAC
USE TEMP
DO CNTEST
DELETE FILE TEMP
CASE 1 (MUNIT) = 'E' .AND. !(MPLAT) = 'A'
SELECT PRIMARY
USE A: PERS
SELECT SECCNDAFY
USE E: 18T
SELECT PRIMARY
JOIN TO TEMP PCK (P.COMPANY = FOURCO .AND. P.PLATCCN;
FOURPLT .AND. F.SSN=S.SSN) FIELDS COC, HIS, CCD, 1;
AID, UNI, PFT1CLASS, PFT2CLASS, NEC, MKS, TAC
USE TEMP
DO CNTEST
DELETE FILE TEMP
CASE 1 (MUNIT) = 'E' .AND. !(MPLAT) = 'B'
SELECT PRIMARY
USE A: PERS
SELECT SECCNDAF7
USE E: 18T
SELECT PRIMARY
JOIN TO TEMP PCK (P.COMPANY = FOURCO .AND. P.PLATCCN;
FOURPLT .AND. F.SSN=S.SSN) FIELDS COC, HIS, CCD, 1;
AID, UNI, PFT1CLASS, PFT2CLASS, NEC, MKS, TAC
USE TEMP
DO CNTEST
DELETE FILE TEMP
CASE 1 (MUNIT) = 'E' .AND. !(MPLAT) = 'C'
SELECT PRIMARY
USE A: PERS
SELECT SECCNDAFY
USE E: 18T
SELECT PRIMARY
JOIN TO TEMP PCK (P.COMPANY = FOURCO .AND. P.PLATCCN;
FOURPLT .AND. F.SSN=S.SSN) FIELDS COC, HIS, CCD, 1;
AID, UNI, PFT1CLASS, PFT2CLASS, NEC, MKS, TAC
USE TEMP
DO CNTEST
DELETE FILE TEMP
CASE 1 (MUNIT) = 'E' .AND. !(MPLAT) = 'D'
SELECT PRIMARY
USE A: PERS
SELECT SECCNDAFY
USE E: 18T
SELECT PRIMARY
JOIN TO TEMP PCK (P.COMPANY = FOURCO .AND. P.PLATCCN;
FOURPLT .AND. F.SSN=S.SSN) FIELDS COC, HIS, CCD, 1;
AID, UNI, PFT1CLASS, PFT2CLASS, NEC, MKS, TAC
USE TEMP
DO CNTEST
DELETE FILE TEMP
ENDCASE
&0,0" SAY "EST STATIS REPORT"
&2,0" SAY "CODE CP Conduct..."
a,5 SAY "ZZTOTAL-ZZCOE/ZZTOTAL*100 USING '999'  
a,5 SAY "PERCENT COMPLETED ..."  
a,5 SAY "CLOSE ORDER DRILL..."  
a,5 SAY "ZZTOTAL-ZZZCOC USING '999'  
a,5 SAY "PERCENT COMPLETED ..."  
a,5 SAY "INTERIOR GUARD..."  
a,5 SAY "PERCENT COMPLETED ..."  
a,5 SAY "FIRST AID..."  
a,5 SAY "ZZTOTAL-ZZTAC/ZZTOTAL*100 USING '999'  
a,5 SAY "PERCENT COMPLETED ..."  
a,5 SAY "EQUIPMENT, UNIFORMS..."  
a,5 SAY "PERCENT COMPLETED ..."  
a,5 SAY "PARTS..."  
a,5 SAY "PERCENT COMPLETED ..."  
a,5 SAY "INDIVID. TAC MEASURES..."  
a,5 SAY "PERCENT COMPLETED ..."  
a,5 SAY "ZZTOTAL-ZZTAC USING '999'  
a,5 SAY "IPFT1..."  
a,5 SAY "PERCENT COMPLETED IPFT ONE..."  
a,5 SAY "IPFT2..."  
a,5 SAY "PERCENT COMPLETED IPFT TWO..."  
ACCEPT "PRESS ENTER TO CONTINUE" TO PROCEED  
RELEASE ALL LIKE ZZ???????
S. EST COUNT ROUTINE

* ROUTINE NAME: CNTES1.PRG
* MODULE NAME: STANDAED REPORT GENERATOR
* VERSION: 5.11.1.0
* AUTHOR: B.E. PEUIETT
* DATE: 12DEC83
* VARIABLES USED: ZZCTAL, ZZCCC, ZZHIS, ZZCOD, ZZINT,
  ZZAID, ZZUNI, ZZNBC, ZZMKS, ZZTAC, ZZPFT1, ZZPFT2
* VARIABLES MODIFIED:
* VARIABLES CREATED: NONE
* VARIABLES RELEASED: NONE
* FILES OPENED/CLOSED: NONE
* TEMP FILES CREATED: NONE
* USING SUBROUTINES: ESTSTAT.PRG
* DESCRIPTION: THIS IS A SIMPLE ROUTINE THAT COUNTS THE
  TOTAL TRAINING CATEGORIES THAT INCLUDE A "*" (I.E. COUNIS
  THE NUMBER OF TRAINING CATEGORIES THAT CURRENTLY INCLUDE
  ONE OR MORE TRAINING ELEMENTS THAT HAVE NOT BEEN
  COMPLETED.

COUNT TO ZZTOTAL
COUNT FOR **S (COC) TC ZZCO.C
COUNT FOR **S (HIS) TC ZZHIS
COUNT FOR **S (COD) TC ZZCOD
COUNT FOR **S (INT) TC ZZINT
COUNT FOR **S (AID) TC ZZAID
COUNT FOR **S (UNI) TC ZZUNI
COUNT FOR **S (NEC) TC ZZNBC
COUNT FOR **S (MKS) TC ZZMKS
COUNT FOR **S (TAC) TC ZZTAC
COUNT FOR **S (PFT1CSS) TO ZZPFT1
COUNT FOR **S (PFT2CSS) TO ZZPFT2
RETURN
I. ESI ESTER ROUTINE

* ROUTINE NAME: ESTRCST.PRG
* MODULE NAME: STANDARD REPORT GENERATOR
* VERSION: 5.12.0
* AUTHOR: R. E. PROUFI
* DATE: 20DEC83
* VARIABLES USED: MUNIT, MPLAT, ONECO, TWOCO, THREEPLT, FOURPLT
* VARIABLES MODIFIED: NONE
* VARIABLES CREATED: COUNT, PROCEED
* VARIABLES RELEASED: COUNT, PROCEED
* FILES (OPENED/CLOSED): A:ernes INDEX A:ALPHPERS, E:EST
* INDEX E:ESTISSN (ALL FILES ARE CLOSED).
* TEMP FILES CREATED: NONE
* USING SUBROUTINES: STDRPT.PRG
* DESCRIPTIVE: THIS IS A CONTROL ROUTINE WHICH CREATES THE
  * PERSONNEL POSTER FOR THE UNIT SPECIFIED BY THE USER'S
  * RESPONSE TO THE SLCLUNIT FEND (5.1.0).

EREASE
SELECT PRIMARY
USE A:ernes INDEX A:ALPHPERS
SELECT SECONDARY
USE E:EST INDEX B:ESTISSN
SELECT PRIMARY
GOTC TOP
STORE 0 TO COUNT
DO WHILE .NOT. EOF
  DO CASE
  CASE !(MUNIT) = 'E'
    DO ESTRPT
    STORE COUNT + 1 TO COUNT
    ENDIF
  CASE !(MUNIT) = 'A'.AND. !(MPLAT) = 'E'
    IF COMPANY = ONECO
      DO ESTRPT
      STORE COUNT + 1 TO COUNT
      ENDLIF
    ENDIF
  CASE !(MUNIT) = 'A'.AND. !(MPLAT) = 'A'
    IF COMPANY = ONECO .AND. PLATOON = ONEPLT
      DO ESTRPT
      STORE COUNT + 1 TO COUNT
      ENDLIF
    ENDIF
  CASE !(MUNIT) = 'A'.AND. !(MPLAT) = 'B'
    IF COMPANY = ONECO .AND. PLATOON = TWOPLT
      DO ESTRPT
      STORE COUNT + 1 TO COUNT
      ENDLIF
    ENDIF
  CASE !(MUNIT) = 'A'.AND. !(MPLAT) = 'C'
    IF COMPANY = ONECO .AND. PLATOON = THREEPLT
      DO ESTRPT
      STORE COUNT + 1 TO COUNT
      ENDLIF
    ENDIF
  CASE !(MUNIT) = 'A'.AND. !(MPLAT) = 'D'
    IF COMPANY = ONECO .AND. PLATOON = FOURPLT
      DO ESTRPT
      STORE COUNT + 1 TO COUNT
      ENDLIF
    ENDIF
  CASE !(MUNIT) = 'E'.AND. !(MPLAT) = 'E'
    IF COMPANY = TWOCCO
      DO ESTRPT
      STORE COUNT + 1 TO COUNT
      ENDLIF
    ENDIF
  CASE !(MUNIT) = 'E'.AND. !(MPLAT) = 'A'
    IF COMPANY = TWOCCO .AND. PLATOON = ONEPLT
      DO ESTRPT
      STORE COUNT + 1 TO COUNT
      ENDLIF
  ENDOF

171
CASE I (MUNIT) = 'E' .AND. !(MPLAT) = 'B'
  IF COMPANY = TCUSCO .AND. PLATOON = TWOPLT
   DO ESTRPT
   STORE COUNT + 1 TO COUNT
  ENDF
CASE I (MUNIT) = 'E' .AND. !(MPLAT) = 'C'
  IF COMPANY = TCUSCO .AND. PLATOON = THREEPLT
   DO ESTRPT
   STORE COUNT + 1 TO COUNT
  ENDF
CASE I (MUNIT) = 'E' .AND. !(MPLAT) = 'D'
  IF COMPANY = TCUSCO .AND. PLATOON = FOURPLT
   DO ESTRPT
   STORE COUNT + 1 TO COUNT
  ENDF
CASE I (MUNIT) = 'E' .AND. !(MPLAT) = 'E'
  IF COMPANY = TCUSCO
   DO ESTRPT
   STORE COUNT + 1 TO COUNT
  ENDF
CASE I (MUNIT) = 'C' .AND. !(MPLAT) = 'A'
  IF COMPANY = THREECO .AND. PLATOON = ONEPLT
   DO ESTRPT
   STORE COUNT + 1 TO COUNT
  ENDF
CASE I (MUNIT) = 'C' .AND. !(MPLAT) = 'B'
  IF COMPANY = THREECO .AND. PLATOON = TWOPLT
   DO ESTRPT
   STORE COUNT + 1 TO COUNT
  ENDF
CASE I (MUNIT) = 'C' .AND. !(MPLAT) = 'C'
  IF COMPANY = THREECO .AND. PLATOON = THREEPLT
   DO ESTRPT
   STORE COUNT + 1 TO COUNT
  ENDF
CASE I (MUNIT) = 'C' .AND. !(MPLAT) = 'D'
  IF COMPANY = THREECO .AND. PLATOON = FOURPLT
   DO ESTRPT
   STORE COUNT + 1 TO COUNT
  ENDF
CASE I (MUNIT) = 'E' .AND. !(MPLAT) = 'E'
  IF COMPANY = PCUSCO
   DO ESTRPT
   STORE COUNT + 1 TO COUNT
  ENDF
CASE I (MUNIT) = 'E' .AND. !(MPLAT) = 'A'
  IF COMPANY = PCUSCO .AND. PLATOON = ONEPLT
   DO ESTRPT
   STORE COUNT + 1 TO COUNT
  ENDF
CASE I (MUNIT) = 'E' .AND. !(MPLAT) = 'B'
  IF COMPANY = PCUSCO .AND. PLATOON = TWOPLT
   DO ESTRPT
   STORE COUNT + 1 TO COUNT
  ENDF
CASE I (MUNIT) = 'E' .AND. !(MPLAT) = 'C'
  IF COMPANY = PCUSCO .AND. PLATOON = THREEPLT
   DO ESTRPT
   STORE COUNT + 1 TO COUNT
  ENDF
CASE I (MUNIT) = 'E' .AND. !(MPLAT) = 'D'
  IF COMPANY = PCUSCO .AND. PLATOON = FOURPLT
   DO ESTRPT
   STORE COUNT + 1 TO COUNT
  ENDF
ENDCASE
IF COUNT = 4
& 22 C
ACCEPT "PRESS 'ENTER' TO CONTINUE" TO PROCEED
STORE 0 TO COUNT

ENTRY

SELECT PRIMARY

SKIP

ENDDC

RELEASE COUNT, PROCEED

ELSE

RETURN
U. IPSTE FORR 87 SOURINE

NAME: EPROG

MOEUL:

NAME: STANDAPR

REPCR GENERATOR

VRSICN: 5.12. 1.0

AUTEOR: R.E. PPUIEll

DAIE: NCV83

VAFIAEIES USED: NON-VARIABLES MODIFIED: ONE

VABIAEIES CREATED: BOW, MSSN

VARIABlES RELEASED: EOW

FIIS CPENED/CLOSEd: NONE

TEMP FILES CHEATED: NONE

USING SUBPOUTINES: EPROGST.PSG

DESCRIFTICN: THIS CUTINE USES FORMATS AND DISPLAYS THE EST RCSTER REPORT.

STORE (CCUNI * 6) TO FCW

ROWI SAY NAME a ROW

A SAY RANK a RCW, 33

SAY COMPANY

STORE SSN 'IC MSSN

SELECT SECCNDARY FIND CMSSN

a ROW + 1,0 SAY "CODE ON"

+ 1,16 SAY "CODE ON"

+ 1,22 SAY "MABCOR HIS"

+ 1,35 SAY "MAPESMANSEIP"

+ 1,43 SAY "DRILL"

+ 1,74 SAY "INTErIOR GUARD"

+ 2,16 SAY "MAFBSMANSHIP"

+ 2,22 SAY "MKS"

+ 2,35 SAY "MKS"

+ 2,43 SAY "TAC"

+ 2,56 SAY "TAC"

+ 2,67 SAY "TAC"

+ 2,74 SAY "TAC"

- 10 SAY "PF1"

- 2,10 SAY "PF1"

- 2,16 SAY PFT2CLSS

- 2,2 SAY "1E UIP/UNIFORM"

- 3,16 SAY "PFT2 UNIFORM" 

- 3,2 SAY "PFT2 UNIFORM"

- 3,3 SAY AID
V. SWIM QUALIFICATION ROUTINE

* ROUTINE NAME: SWIMQUAL.PRG
* MODULE NAME: STANDARD REPORT GENERATOR
* VERSION: 5.13.0
* AUTHOR: R. E. PRUITT
* DATE: 20DEC83
* VARIABLES USED: MUNIT, MPLAT, ONECO, TWOCO, THREECO,
*                 FOURCO, NEPLT, TWECPLT, THREEPLT, FOURPLT
* VARIABLES MODIFIED: NONE
* VARIABLES CREATED: PROCEED, COUNT, S1TOTAL, S2TOTAL,
* S3TOTAL, UQTOTAL,
* VARIABLES RELEASED: COUNT, PROCEED, MSSN, S1TOTAL,
* S2TOTAL, S3TOTAL, UQTOTAL,
* FILES (OPENED/CLOSED): AI:FEIS INDEX A:ALPHERS
* TIME FILES CREATED: NONE
* USING SUBROUTINES: STDRPT.PRG
* DESCRIPTION: THIS ROUTINE DISPLAYS THE SWIM QUAL REPORT
* FOR THE SELECTED UNIT. THE SWIM QUAL REPORT CONSIST OF
* EACH MARINES NAME, RANK, UNIT NAME, AND SWIMMING
* QUALIFICATION STANAD. AT THE END OF THE REPORT, SUMMARY
* DATA IS DISPLAYED WHICH INCLUDES THE TOTALS FOR EACH
* QUALIFICATION CATEGORY AND THE OVERALL TOTAL FOR THE
* UNIT.

ERASE
@ 12.27 SAY "SWIM QUALIFICATION REPORT"
@ 22 DEC
ACCEPT "PRESS 'ENTER' KEY TO CONTINUE" TO PROCEED
ERASE
SELECT PRIMARY
USE AI:FEIS INDEX A:AI:FEIS
CUTO TCF
STORE 0 TO S1TOTAL
STORE 0 TO S2TOTAL
STORE 0 TO S3TOTAL
STORE 0 TO UQTOTAL
STORE 0 TO COUNT
LO WHILE .NOT. EOF
DO CASE
CASE !(MUNIT) = 'I'
DC SWIMRPT
STORE COUNT + 1 TO COUNT
CASE !(MUNIT) = 'I' .AND. !(MPLAT) = 'I'
IF COMPANY = ONECO
DO SWIMRPT
STORE COUNT + 1 TO COUNT
ENDIF
CASE !(MUNIT) = 'I' .AND. !(MPLAT) = 'A'
IF COMPANY = ONECO .AND. PLATOON = ONEPLT
DO SWIMRPT
STORE COUNT + 1 TO COUNT
ENDIF
CASE !(MUNIT) = 'I' .AND. !(MPLAT) = 'B'
IF COMPANY = ONECO .AND. PLATOON = TWOPLT
DO SWIMRPT
STORE COUNT + 1 TO COUNT
ENDIF
CASE !(MUNIT) = 'I' .AND. !(MPLAT) = 'C'
IF COMPANY = ONECO .AND. PLATOON = THREEPLT
DO SWIMRPT
STORE COUNT + 1 TO COUNT
ENDIF
CASE !(MUNIT) = 'I' .AND. !(MPLAT) = 'D'
IF COMPANY = ONECO .AND. PLATOON = FOURPLT
DO SWIMRPT
STORE COUNT + 1 TO COUNT
ENDIF
END
CASE I(MUNIT) = 'E' .AND. !(MPLAT) = 'E'
IF COMPANY = TWCCO
DO SWIMRPT
STORE CCUNT + 1 TO CCUNT
ENDIF
CASE I(MUNIT) = 'E' .AND. !(MPLAT) = 'A'
IF COMPANY = TWCCO .AND. PLATOON = ONEPLT
DO SWIMRPT
STORE CCUNT + 1 TO CCUNT
ENDIF
CASE I(MUNIT) = 'E' .AND. !(MPLAT) = 'B'
IF COMPANY = TWCCO .AND. PLATOON = TWOPLT
DO SWIMRPT
STORE CCUNT + 1 TO CCUNT
ENDIF
CASE I(MUNIT) = 'E' .AND. !(MPLAT) = 'C'
IF COMPANY = TWCCO .AND. PLATOON = THREEPLT
DO SWIMRPT
STORE CCUNT + 1 TO CCUNT
ENDIF
CASE I(MUNIT) = 'E' .AND. !(MPLAT) = 'D'
IF COMPANY = TWCCO .AND. PLATOON = FOURPLT
DO SWIMRPT
STORE CCUNT + 1 TO CCUNT
ENDIF
CASE I(MUNIT) = 'E' .AND. !(MPLAT) = 'F'
IF COMPANY = TWCCO .AND. PLATOON = FIVEPLT
DO SWIMRPT
STORE CCUNT + 1 TO CCUNT
ENDIF
CASE I(MUNIT) = 'E' .AND. !(MPLAT) = 'A'
IF COMPANY = TERICECO .AND. PLATOON = ONEPLT
DO SWIMRPT
STORE CCUNT + 1 TO CCUNT
ENDIF
CASE I(MUNIT) = 'E' .AND. !(MPLAT) = 'B'
IF COMPANY = TERICECO .AND. PLATOON = TWOPLT
DO SWIMRPT
STORE CCUNT + 1 TO CCUNT
ENDIF
CASE I(MUNIT) = 'E' .AND. !(MPLAT) = 'C'
IF COMPANY = TERICECO .AND. PLATOON = THREEPLT
DO SWIMRPT
STORE CCUNT + 1 TO CCUNT
ENDIF
CASE I(MUNIT) = 'E' .AND. !(MPLAT) = 'D'
IF COMPANY = TERICECO .AND. PLATOON = FOURPLT
DO SWIMRPT
STORE CCUNT + 1 TO CCUNT
ENDIF
CASE I(MUNIT) = 'E' .AND. !(MPLAT) = 'E'
IF COMPANY = TERICECO .AND. PLATOON = FIVEPLT
DO SWIMRPT
STORE CCUNT + 1 TO CCUNT
ENDIF
CASE I(MUNIT) = 'E' .AND. !(MPLAT) = 'F'
IF COMPANY = TERICECO .AND. PLATOON = SIXPLT
DO SWIMRPT
STORE CCUNT + 1 TO CCUNT
ENDIF
CASE I(MUNIT) = 'E' .AND. !(MPLAT) = 'A'
IF COMPANY = FCRBCO .AND. PLATOON = ONEPLT
DO SWIMRPT
STORE CCUNT + 1 TO CCUNT
ENDIF
CASE I(MUNIT) = 'E' .AND. !(MPLAT) = 'B'
IF COMPANY = FCRBCO .AND. PLATOON = TWOPLT
DO SWIMRPT
STORE CCUNT + 1 TO CCUNT
ENDIF
CASE I(MUNIT) = 'E' .AND. !(MPLAT) = 'C'
IF COMPANY = FCRBCO .AND. PLATOON = THREEPLT
DO SWIMRPT
STORE CCUNT + 1 TO CCUNT
ENDIF
CASE I(MUNIT) = 'E' .AND. !(MPLAT) = 'D'
IF COMPANY = FCRBCO .AND. PLATOON = FOURPLT
DO SWIMRPT
STORE CCUNT + 1 TO CCUNT
ENDIF
CASE I(MUNIT) = 'E' .AND. !(MPLAT) = 'E'
IF COMPANY = FCRBCO .AND. PLATOON = FIVEPLT
DO SWIMRPT
STORE CCUNT + 1 TO CCUNT
ENDIF
CASE I(MUNIT) = 'E' .AND. !(MPLAT) = 'F'
IF COMPANY = FCRBCO .AND. PLATOON = SIXPLT
DO SWIMRPT
STORE CCUNT + 1 TO CCUNT
ENDIF
CASE I(MUNIT) = 'E' .AND. !(MPLAT) = 'A'
IF COMPANY = FCRBCO .AND. PLATOON = ONEPLT
DO SWIMRPT
STORE CCUNT + 1 TO CCUNT
ENDIF
CASE I(MUNIT) = 'E' .AND. !(MPLAT) = 'B'
IF COMPANY = FCRBCO .AND. PLATOON = TWOPLT
DO SWIMRPT
STORE CCUNT + 1 TO CCUNT
ENDIF
CASE I(MUNIT) = 'E' .AND. !(MPLAT) = 'C'
IF COMPANY = FCRBCO .AND. PLATOON = THREEPLT
DO SWIMRPT
STORE CCUNT + 1 TO CCUNT
ENDIF
CASE I(MUNIT) = 'E' .AND. !(MPLAT) = 'D'
IF COMPANY = FCRBCO .AND. PLATOON = FOURPLT
DO SWIMRPT
STORE CCUNT + 1 TO CCUNT
ENDIF
CASE I(MUNIT) = 'E' .AND. !(MPLAT) = 'E'
IF COMPANY = FCRBCO .AND. PLATOON = FIVEPLT
DO SWIMRPT
STORE CCUNT + 1 TO CCUNT
ENDIF
CASE I(MUNIT) = 'E' .AND. !(MPLAT) = 'F'
IF COMPANY = FCRBCO .AND. PLATOON = SIXPLT
DO SWIMRPT
STORE CCUNT + 1 TO CCUNT
ENDIF
CASE I(MUNIT) = 'E' .AND. !(MPLAT) = 'A'
IF COMPANY = FCRBCO .AND. PLATOON = ONEPLT
DO SWIMRPT
STORE CCUNT + 1 TO CCUNT
ENDIF
CASE I(MUNIT) = 'E' .AND. !(MPLAT) = 'B'
IF COMPANY = FCRBCO .AND. PLATOON = TWOPLT
DO SWIMRPT
STORE CCUNT + 1 TO CCUNT
ENDIF
CASE I(MUNIT) = 'E' .AND. !(MPLAT) = 'C'
IF COMPANY = FCRBCO .AND. PLATOON = THREEPLT
DO SWIMRPT
STORE CCUNT + 1 TO CCUNT
ENDIF
CASE I(MUNIT) = 'E' .AND. !(MPLAT) = 'D'
IF COMPANY = FCRBCO .AND. PLATOON = FOURPLT
DO SWIMRPT
STORE CCUNT + 1 TO CCUNT
ENDIF
CASE I(MUNIT) = 'E' .AND. !(MPLAT) = 'E'
IF COMPANY = FCRBCO .AND. PLATOON = FIVEPLT
DO SWIMRPT
STORE CCUNT + 1 TO CCUNT
ENDIF
CASE I(MUNIT) = 'E' .AND. !(MPLAT) = 'F'
IF COMPANY = FCRBCO .AND. PLATOON = SIXPLT
DO SWIMRPT
STORE CCUNT + 1 TO CCUNT
ENDIF
CASE I(MUNIT) = 'E' .AND. !(MPLAT) = 'A'
IF COMPANY = FCRBCO .AND. PLATOON = ONEPLT
DO SWIMRPT
STORE CCUNT + 1 TO CCUNT
ENDIF
CASE I(MUNIT) = 'E' .AND. !(MPLAT) = 'B'
IF COMPANY = FCRBCO .AND. PLATOON = TWOPLT
DO SWIMRPT
STORE CCUNT + 1 TO CCUNT
ENDIF
CASE I(MUNIT) = 'E' .AND. !(MPLAT) = 'C'
IF COMPANY = FCRBCO .AND. PLATOON = THREEPLT
DO SWIMRPT
STORE CCUNT + 1 TO CCUNT
ENDIF
CASE I(MUNIT) = 'E' .AND. !(MPLAT) = 'D'
IF COMPANY = FCRBCO .AND. PLATOON = FOURPLT
DO SWIMRPT
STORE CCUNT + 1 TO CCUNT
ENDIF
CASE I(MUNIT) = 'E' .AND. !(MPLAT) = 'E'
IF COMPANY = FCRBCO .AND. PLATOON = FIVEPLT
DO SWIMRPT
STORE CCUNT + 1 TO CCUNT
ENDIF
CASE I(MUNIT) = 'E' .AND. !(MPLAT) = 'F'
IF COMPANY = FCRBCO .AND. PLATOON = SIXPLT
DO SWIMRPT
STORE CCUNT + 1 TO CCUNT
ENDIF

176
CASE 1 (MUNIT) = 'E' AND I (MPLAT) = 'D'
IF COMPANY = PCURCO AND PLATOON = FOURPLT
GO SWIMRPT
STORE CCUNT + 1 TO CCUNT
ENDIF
ENCLSE
IF CCUNT = 7
4 22 C
ACCEPT "PRESS 'ENTER' TO CONTINUE" TO PROCEED
STORE 0 TO CCOUNT
ENDIF
SELECT PRIMARY
SKIP
ENDDO
4 22 C
ACCEPT "PRESS 'ENTER' TO CONTINUE" TO PROCEED
ENDIF
4 2,20 SAY "SUMMARY DATA"
4 4,25 SAY "SWIM QUALIFICATION"
4 6,25 SAY "TOTAL"
4 6,25 SAY S1 TOTAL USING '999'
4 7,25 SAY S2 TOTAL USING '999'
4 8,25 SAY S3 TOTAL USING '999'
4 9,25 SAY "UNQUALIFIED"
4 10,25 SAY "TOTAL USING '999'
4 12,25 SAY "TOTAL USING '9999"'
4 22 C
ACCEPT "PRESS 'ENTER' TO CONTINUE" TO PROCEED
RELEASE CCOUNT, PROCEED, MSSN, S1 TOTAL, S3 TOTAL,;
UQ TOTAL, FLOW
USE
RETURN
**Routine Name:** SWIMSPR.FRG
**Module Name:** STANDARD REPORT GENERATOR
**Version:** E.13.1.0
**Author:** R.E. PRUEETT
**Date:** 12DEC83
**Variables Used:** None
**Variables Modified:** None
**Variables Created:** FGW, MSSN, S1TOTAL, S2TOTAL, S3TOTAL, UQCTAL
**Variables Released:** None
**Files Opened/Closed:** QUAL INDEX QUALSSN
**Files Created:** BCN
**Using Subroutines:** SWIMQUAL.PRG
**Description:** This routine counts and stores the number of personnel in the selected unit that are in each swimming qualification category.

```plaintext
STORE (COUNT * 3) TO ROW
@ ROW.0 SAY NAME
@ ROW.30 SAY RANK
@ ROW.33 SAY COMPANY
STORE MSSN TO BCN
SELECT SECONDARY
USE B:QUAL INDEX B:QUALSSN
FIND MSSN
@ ROW+1,10 SAY "SWIMQUAL"
@ FLOW+1,10 SAY SWIMQUAL
DO CASE
  CASE SWIMQUAL='S1':
    STORE S1TOTAL + 1 TO S1TOTAL
  CASE SWIMQUAL='S2':
    STORE S2TOTAL + 1 TO S2TOTAL
  CASE SWIMQUAL='S3':
    STORE S3TOTAL + 1 TO S3TOTAL
  CASE SWIMQUAL='UQ':
    STORE UQCTAL + 1 TO UQCTAL
ENDCASE
END
RETURN
```
I. HELP ROUTINE

* ROUTINE NAME: HELP.FRG
* MODULE NAME: STANDARD REPORT GENERATOR
* VERSION: 5.14.0
* AUTHOR: F.E. PRUIETT
* DATE: 18JAN84
* VARIABLES USED: NONE
* VARIABLES MODIFIED: NONE
* VARIABLES CREATED: HELPOPT, PROCEED
* VARIABLES RELEASED: NONE
* FILES CREATED/CLOSED: NONE
* USING ROUTINES: STDRPT
* DESCRIPTION: THIS ROUTINE PROVIDES A FUNCTIONAL
* DESCRIPTION OF EACH REPORT LISTED IN THE STANDARD REPORT
* SELECTION MENU.

ERASE
@2,35 SAY "HELP MENU"
@3,35 SAY ""--.--."'
@5,20 SAY "A...HELP UNIT ROSTER"
@9,20 SAY "B...HELP TRAINING STATUS REPORT"
@8,20 SAY "C...HELP TRAINING ROSTER REPORT"
@9,20 SAY "D...HELP I ndividual Training Record"
@10,20 SAY "E...HELP MARKSMANSHIP STATUS REPORT"
@11,20 SAY "F...HELP MARKSMANSHIP ROSTER REPORT"
@12,20 SAY "G...HELP MOS STATUS REPORT"
@13,20 SAY "H...HELP MCS ROSTER REPORT"
@14,20 SAY "I...HELP PERSONAL DATA REPORT"
@15,20 SAY "J...HELP EST STATUS REPORT"
@16,20 SAY "K...HELP EST ROSTER REPORT"
@17,20 SAY "L...HELP SWIM QUALIFICATION REPORT"
ACCEPt "SELECT OPTION ======> " TO HELPOPT
ERASE
DO CASE
CASE ! (HELPOPT) = 'A'
TEXT
1. At the cursor, next to "select option ======>", type the
letter (A-F) that corresponds to the unit desired. Enter
the letter by pressing the enter key.

2. If a company was selected (A-D), you will next be asked
to select a subunit. Enter the appropriate letter. If you
want the report to include the entire unit selected, enter
"E" (i.e., ALL OF UNIT).

3. If option "F" (i.e., BATTALION) was selected in the
select unit option, you will go directly to the report and
will not be asked to select subunit.

4. The UNIT-ROSTER REPORT will display in alphabetical order
the NAME, FAXR, and PRIMARY MOS for each member of the
selected unit or subunit. The screen will scroll
automatically unless stopped at the terminal.

5. At the end of the report press the enter key to return
to the STANDARD REPORT MENU.
ENDITEM
CASE !(HELPOPT) = 'B'
TEXT
CITION B...TRAINING STATUS REPORT
1. At the cursor, next to "select option ======>", type and
enter the letter (A-F) that corresponds to the unit
2. If a company was selected (A-D), you will next be asked to select a subunit. Enter the appropriate letter. If you want the report to include the entire unit selected, enter "E" (i.e., ALL OF UNIT).

3. If option "E" (i.e., BATTALION) was selected in the select unit option, you will go directly to the report and will not be asked to select a subunit.

4. A "WORKING" signal on the screen will be shown to indicate that the computer is searching for the information you requested.

5. The TRAINING STATUS REPORT provides for all essential subjects, the PERCENT of the unit that has successfully completed each training element and the TOTAL number of unit members that have completed each training element. In addition, the report includes completion status for PFT1, PFT2, SWIMQUAL, RIFLE QUAL, and PISTOL QUAL.

6. Press the enter key to return to the STANDARD REPORT MENU.

END TEXT

CASE 1(FELFCPT) = 'C'

TEXT

OPTION C...TRAINING ROSTER REPORT

1. At the cursor, next to "select option ===>", type the letter (A-E) that corresponds to the unit desired. Enter the letter by pressing the enter key.

2. If a company was selected (A-D), you will next be asked to select a subunit. Enter the appropriate letter. If you want the report to include the entire unit selected, enter "E" (i.e., ALL OF UNIT).

3. If option "E" (i.e., BATTALION) was selected in the select unit option, you will go directly to the report and will not be asked to select a subunit.

4. The TRAINING ROSTER REPORT heading will appear in the center of the screen. Press the enter key to begin the report.

END TEXT

& Z 0

ACCEPT "PRESS 'ENTER' KEY TO CONTINUE" TO PROCEED

FRASI

TEXT

5. The TRAINING ROSTER REPORT displays in alphabetic order each member of the selected unit or subunit and his complete training status for the current year. If a training element has been successfully completed for the current year, a letter corresponding to that training element will be shown. A "*" symbol indicates that a training element has not been completed. For example, "FIRST AID AE*H**B**E" indicates that elements ABDE and H have been completed and five of the eleven elements have NOT been completed (i.e., C, G, I, J, and K). Four records will be displayed on the screen and you will be asked to press the enter key to scroll the report.

6. At the end of the report press the enter key to return to the STANDARD REPORT MENU.

END TEXT

CASE 1(FELFCPT) = 'D'

TEXT
1. At the cursor, next to "select option ===>", type the letter (A-E) that corresponds to the unit desired. Enter the letter by pressing the enter key.

2. If a company was selected (A-D), you will next be asked to select a subunit. Enter the appropriate letter. If you want the report to include the entire unit selected, enter "E" (i.e., ALL OF UNIT).

3. If option "E" (i.e., BATTALION) was selected in the select unit option, you will go directly to the report and will not be asked to select a subunit.

4. At the end of the report press the enter key to return to the STANDARD REPORT MENU.

ENDTEXT

CASE !(ELEFCPT) = 'E'

TEXT

OPTION D...MARKSMANSHIP STATUS REPORT

1. At the cursor, next to "select option ===>", type the letter (A-E) that corresponds to the unit desired. Enter the letter by pressing the enter key.

2. If a company was selected (A-D), you will next be asked to select a subunit. Enter the appropriate letter. If you want the report to include the entire unit selected, enter "E" (i.e., ALL OF UNIT).

3. If option "E" (i.e., BATTALION) was selected in the select unit option, you will go directly to the report and will not be asked to select a subunit.

4. The MARKSMANSHIP STATUS REPORT will display the qualification results for the unit or subunit members that have fired for qualification during the current year. Also, the percent that has qualified in each category (e.g., PRESS, EM, UNC) and current year results for rifle and pistol will be shown. The pistol qualification results include only those members required to fire the pistol for qualification. Press the enter key to continue with the report.

ENDTEXT

ACCEPT "PRESS 'ENTER' KEY TO CONTINUE" TO PROCEED

BASI

TEXT

5. You will be asked if you want to do a company or battalion range quota analysis. If you answer no ("N"), you will automatically return to the STANDARD REPORT MENU. The range quota analysis allows you to observe, manipulate or permanently change the rifle range quotas that are assigned to the unit for the rest of the calendar year. Range quota analysis may only be done for the battalion or for a selected company (not platoons).

6. At the top of the screen general rifle range quota information is displayed followed by the number of quotas remaining in the current calendar year. Finally, an indication whether remaining quotas are less than, greater than, or equal to the number of Marines left to qualify is shown.

7. From this point you may change the quota allocation on a scratch pad or you may return to the STANDARD REPORT MENU. The original set of quotas have not been modified.
8. The scratch pad again shows the remaining quota allocations. However, you may make any temporary quota changes you desire on the scratch pad by simply overtyping the current quota allocation(s). The enter key will automatically move the cursor to the next quota, and a buzzer will sound when you reach the end of each field.

9. After you've reached the last month's quota, you will be asked to select one of the following three options:
   A...Start over on the scratch pad with the original set of quotas.
   B...Return to the STANDARD REPORT MENU
   C...PERMANENTLY REPLACE ORIGINAL QUOTAS with the set of quotas that are currently displayed on the screen. This is the only option that will actually modify the original set of quotas for the unit.

10. Option A may be repeated indefinitely. Option B or option C will return you directly to the STANDARD REPORT MENU.

ENDTEXT

CASE !(EFLPCPT) = 'F'
TEXT
OPTION F...MARKSMANSHIP ROSTER REPORT
1. At the cursor, next to "select option ===>", type the letter (A-F) that corresponds to the unit desired. Enter the letter by pressing the enter key.

2. If a company was selected (A-D), you will next be asked to select a subunit. Enter the appropriate letter. If you want the report to include the entire unit selected, enter "E" (i.e., ALL OF UNIT).

3. If option "E" (i.e., BATTALION) was selected in the select unit option, you will go directly to the report and will not be asked to select a subunit.

4. The MARKSMANSHIP ROSTER REPORT will display in alphabetical order the NAME, RANK, DATE OF QUALIFICATION, and QUALIFICATION RESULTS for each member of the unit or subunit. Current year results only will be displayed.

5. At the end of the report press the enter key to return to the STANDARD REPORT MENU.

ENDIFIT

CASE !(EFLPCPT) = 'G'
TEXT
OPTION G...MCS STATUS REPORT
1. At the cursor, next to "select option ===>", type the letter (A-F) that corresponds to the unit desired. Enter the letter by pressing the enter key.

2. If a company was selected (A-D), you will next be asked to select a subunit. Enter the appropriate letter. If you want the report to include the entire unit selected, enter "E" (i.e., ALL OF UNIT).

3. If option "E" (i.e., BATTALION) was selected in the select unit option, you will go directly to the report and will not be asked to select a subunit.
4. The FCS STATUS REPORT displays the number of unit or subunit members grouped by MOS and RANK. The screen will scroll automatically unless stopped from your terminal.

5. At the end of the report press the enter key to return to the STANDARD REPORT MENU.
ENDDIT

CASE I (EELPCPT) = 'H'
TEXT

OPTION H...MOS ROSTER REPORT

1. At the cursor, next to "select option ====>", type the letter (A-E) that corresponds to the unit desired. Enter the letter by pressing the enter key.

2. If a company was selected (A-D), you will next be asked to select a subunit. Enter the appropriate letter. If you want the report to include the entire unit selected, enter "E" (i.e., ALL OF UNIT).

3. If option "E" (i.e., BATTALION) was selected in the select unit option, you will go directly to the report and will not be asked to select a subunit.

4. The FCS ROSTER REPORT groups by primary MOS and displays in alphabetical order the NAME, RANK, PRIMARY MOS AND SECONDARY MOS of all members in the unit or subunit.

5. At the end of the report press the enter key to return to the STANDARD REPORT MENU.
ENDDIT

CASE I (EELPCPT) = 'I'
TEXT

OPTION I...PERSONAL DATA REPORT

1. At the cursor, next to "select option ====>", type the letter (A-E) that corresponds to the unit desired. Enter the letter by pressing the enter key.

2. If a company was selected (A-D), you will next be asked to select a subunit. Enter the appropriate letter. If you want the report to include the entire unit selected, enter "E" (i.e., ALL OF UNIT).

3. If option "E" (i.e., BATTALION) was selected in the select unit option, you will go directly to the report and will not be asked to select a subunit.

4. The PERSONAL DATA REPORT displays for each member of the unit or subunit all personal data including NAME, SSN, RANK, PRIMARY/SECONDARY MOS, UNIT/SUBUNIT NAME, JOIN DATE, EAS, EIGHTDATE, WEIGHT and a one character comment block. If the comment block is "T" (true), a comment is contained on the individual concerned in the comment file. The screen will scroll upon pressing the enter key.

5. At the end of the report press the enter key to return to the STANDARD REPORT MENU.
ENDDIT

CASE I (EELPCPT) = 'J'
TEXT

OPTION J...EST STATUS REPORT

1. At the cursor, next to "select option ====>", type the letter (A-E) that corresponds to the unit desired. Enter the letter by pressing the enter key.

2. If a company was selected (A-D), you will next be asked
to select a subunit. Enter the appropriate letter. If you want the report to include the entire unit selected, enter "E" (i.e., ALL OF UNIT).

3. If option "E" (i.e., BATTALION) was selected in the select unit option, you will go directly to the report and will not be asked to select a subunit.

4. The EST STATUS REPORT includes the percent and total number of unit or subunit members that have completed each ESSENTIAL SUBJECT and the PFT.

5. At the end of the report press the enter key to return to the STANDARD REPORT MENU.

ENDTEXT

CASE !((EELPCPT) = 'K') TEXT

OPTION K...EST ROSTER REPORT

1. At the cursor, next to "select option ===>", type the letter (A-E) that corresponds to the unit desired. Enter the letter by pressing the enter key.

2. If a company was selected (A-D), you will next be asked to select a subunit. Enter the appropriate letter. If you want the report to include the entire unit selected, enter "E" (i.e., ALL OF UNIT).

3. If option "E" (i.e., BATTALION) was selected in the select unit option, you will go directly to the report and will not be asked to select a subunit.

4. The EST ROSTER REPORT displays for each member of the unit or subunit all of his essential subject training and PFT results for the current year. A "*" character indicates an element that has not been completed. The screen will scroll upon pressing the enter key.

5. At the end of the report press the enter key to return to the STANDARD REPORT MENU.

ENDTEXT

CASE !((EELPCPT) = 'L') TEXT

OPTION L...SWIM QUALIFICATION REPORT

1. At the cursor, next to "select option ===>", type the letter (A-E) that corresponds to the unit desired. Enter the letter by pressing the enter key.

2. If a company was selected (A-D), you will next be asked to select a subunit. Enter the appropriate letter. If you want the report to include the entire unit selected, enter "E" (i.e., ALL OF UNIT).

3. If option "E" (i.e., BATTALION) was selected in the select unit option, you will go directly to the report and will not be asked to select a subunit.

4. The SWIM QUALIFICATION REPORT displays in alphabetic order for each member of the unit or subunit NAME, RANK, UNIT and CURRENT SWIM QUAL RESULTS.

5. At the end of the report a SUMMARY DATA TABLE is displayed which shows the number of unit or subunit members that have qualified in each category (S1, S2, S3, UNQ).

6. Press the enter key to return to the STANDARD REPORT MENU.

ENDTEXT
ENDCASE
RETURN

ACCEPT "PRESS 'ENTER' TO CONTINUE" TO PROCEED

ENCASE
RETURN

23C
APPENDIX H
MAINTENANCE MODULE LISTING

A. MAINTAIN CONTROL ROUTINE

* Routine Name: Maintain.Prg
* Module Name: Maintenance Module
* Version: 6.0
* Author: D.P. Hauser
* Date: 30 Nov 83
* Variables Used: mdone, mntnopt
* Variables Modified: mdone, mntnopt
* Variables Released: mdone, mntnopt
* Files (opened/closed): none
* Temporary Files Created: none
* Using Subroutines: Cmdnet
* Description: This routine produces a module which gives
  the user the option of which maintenance function to
  activate. Each function has an authorization level
  associated with and only appropriate users may activate
  certain modules.

STORE f TO AAdone
DO WHILE .NOT. AAdone
  ERASE
  @ 4,29 SAY "MAINTENANCE MODULE MENU"
  @ 6,24 SAY "SELECT ONE OF THE FOLLOWING OPTIONS"
  @ 8,31 SAY "V...VIEW INDIVIDUAL DATA"
  @ 9,31 SAY "U...UPDATE DATA"
  @ 10,31 SAY "C...CREATE AN ITR"
  @ 11,31 SAY "D...DELETE AN ITR"
  @ 12,31 SAY "S...SYSTEM FUNCTIONS"
  @ 13,31 SAY "H...HELP"
  @ 14,29 SAY "* Q...QUIT *"
  @ 15,0 SAY " "
  ACCEPT " SELECT OPTION ====>" TO mntnopt
  DC CASE
    CASE !(mntnopt) = 'V'
      DO viewitr
    CASE !(mntnopt) = 'U'
      IF AAlevel < 3
        DO uptitr
      ELSE
        ERASE
        @ 8,31 SAY "ACCESS UNAUTHORIZED"
        DO pause
      ENDIF
    CASE !(mntnopt) = 'C'
      IF AAlevel < 3
        DO creatitr
      ELSE
        ERASE
        @ 8,31 SAY "ACCESS UNAUTHORIZED"
        DO pause
      ENDIF
    CASE !(mntnopt) = 'D'
      IF AAlevel = 1
        DO deletitr

ELSE
  ERASE @ 8,31 SAY "ACCESS UNAUTHORIZED"
  DO pause
ENDIF
CASE !(mntnopt) = 'S'
  IF AAAlev e = 1
    DO syste r
  ELSE
    ERASE @ 8,31 SAY "ACCESS UNAUTHORIZED"
    DO pause
ENDIF
CASE !(mntnopt) = 'H'
  STORE 2 TO HELP2
  DO B:HELP2
CASE !(mntnopt) = 'Q'
  STORE t TO AAdone
  ERASE
ENDCASE
ENDDO
RELEASE mntnopt, HELP2
RELEASE AAdone
RETURN
E. VIEW CONTROL ROUTINE

* Routine Name: Viewitr.prg
* Module Name: Maintenance Module
* Version: 6.1.0
* Author: E.F. Haehl
* Date: 1 Dec 83
* Variables Used: vname, vfinish, vname, vcomp, vagain, vssn, vend, view
* Variables Modified: same as above
* Variables Created: same as above
* Variables Released: all like ???
* Files (opened/closed): a:pers (opened/closed)
* Temporary Files Created: none
* Using Subroutines: maintain.prg, stdrpt.prg
* Description: This routine controls the creation of the
* ITR for viewing only. It does not allow any changes to
* data.

STORE F TC VNOFIND
STORE F TC VDONE
DO WHILE .NOT. VDONE
  STORE F TO VFINISH
  DC WHILE .NOT. VFINISH .AND. .NOT. VDONE
  ERASE
  @ 6,31 SAY "TO LOCATE AN ITR"
  @ 6,31 SAY "ENTER NAME (LAST, FIRST MI.)"
  ACCEPT "NAME ="=>" TO VNAME
  STORE ! (VNAME) TO VNAME
  USE A:PERSONS INDEX A:ALPHABETS
  FIND VNAME
  IF # = 0
    ERASE
    @ 6,31 SAY "INDIVIDUAL IS NOT IN THE DATA BASE"
    STORE P TO VCOMP
    DO WHILE .NOT. VCCME
      ACCEPT "DO YOU WANT TO TRY AGAIN (Y/N)" ;
      TO VAGAIN
      STORE ! (VAGAIN) TO VAGAIN
    DO CASE
      CASE VAGAIN = 'N'
        STORE T TO VNAME
        STORE T TO VCOMP
        STORE T TO VFINISH
      CASE VAGAIN = 'Y'
        STORE T TO VCCME
        STORE T TO VNOFIND
    ENDCASE
  END DO
ELSE
  STORE SSN TO AASSN
  STORE COMMENT TO STCCMMNT
  USE
  STORE P TO VEND
  DO WHILE .NOT. VEND
    STORE T TO VEND
    DO ITSCEN
    DO GETDATA
    @ 24,24 SAY "** PRESS ANY KEY TO CONTINUE **"
    SET CONSOLE OFF
    WAIT
    SET CONSOLE ON
    STORE F TO VIEW
IF STCOMET
  STORE F TO VCOMP
  DC WHILE .NOT. VCCMP
  ACCEPT " DO YOU WANT TO VIEW COMMENTS:
  (Y/N)?" TC VAGAIN
  DO CASE
    CASE !(VAGAIN) = 'N'
      STORE T TO VCOMP
      STORE F TO VIEW
      STORE T TO VFINISH
      STORE T TO VDONE
    CASE !(VAGAIN) = 'Y'
      STORE T TO VCOMP
      STORE T TO VIEW
      ENCASE
  ENDDO
  IF VIEW
    DO GETCMNT
  ENDF
  ENDF
  @ 24.0
  STORE T TC VFINISH
  STORE P TC VCOMP
  DO WHILE .NOT. VCOMP .AND. VIEW
    ACCEPT " DO YOU WANT TO VIEW IT AGAIN:
    (Y/N)?" TC VAGAIN
    DC CASE
      CASE !(VAGAIN) = 'Y'
        STORE T TC VCOMP
        STORE F TO VEND
      CASE !(VAGAIN) = 'N'
        STORE T TO VCOMP
        STORE T TC VEND
      ENCASE
    ENDDO
    ENDO
    ENDO
    STORE E TO VCOMP
    DC WHILE .NOT. VCCMP .AND. .NOT. VNOFIND
    @ 11.24 SAY " TC VIEW ANOTHER IT"H" ENTER (Y/N) ===>";
    TO VAGAIN
    DC CASE
      CASE !(VAGAIN) = 'Y'
        STORE T TC VCOMP
      CASE !(VAGAIN) = 'N'
        STORE T TC VDONE
      ENCASE
    ENDDO
    ENDDC
    RELEASE ALL EXCEPT AA??????
    RETURN
C. ITR SCREEN FORMAT ROUTINE

* Routine Name: Itrscrn.prq
* Module Name: Maintenance Module
* Version: 6.1.0.1
* Author: C.P. Haeusler
* Date: 1 DEC 83
* Variables Used: NONE
* Variables Modified: NONE
* Variables Created: NONE
* Variables Released: NONE
* Files (opened/closed): NONE
* Temporary Files Created: NONE
* Using Subroutines: Viewitr, Creatitr, Updtitr, Deletitr

Description: This routine formats the screen in the representation of an ITR.

ERASE

0,25 SAY "INIVIDUAL TRAINING RECORD"
1,46 SAY "NAME:"  "/:
1,57 SAY "GRADE:"  "/:
1,66 SAY "MOS:"  "/:
2,8 SAY "BIRTHDATE.....:"  "/:
2,40 SAY "SSN.....:"  "/:
3,8 SAY "CO:"  "/:
3,46 SAY "PLAT:"  "/:
3,48 SAY "JOIN:"  "/:
3,46 SAY "EAS:"  "/:
3,58 SAY "GAS MASK:"  "/:
4,8 SAY "KIL CODE OF CONDUCT:"  "/:
4,40 SAY "PFT1.......:"  "/:
4,55 SAY "DATE:"  "/:
5,61 SAY "HISTORY OF MARCO:"  "/:
5,40 SAY "PFT2.......:"  "/:
5,55 SAY "DATE:"  "/:
6,8 SAY "DRILL.................:"  "/:
6,40 SAY "NBC.......:"  "/:
7,8 SAY "INTERIOR GUARD.......:"  "/:
7,40 SAY "MARKSMANSHIP.......:"  "/:
8,8 SAY "FIRST AID.......:"  "/:
8,40 SAY "INDIVIDUAL TACTICAL:"  "/:
9,8 SAY "EQUIP & UNIFORM.......:"  "/:
9,40 SAY "MEASURES.................:"  "/:
11,8 SAY "SKIN QUAL.................:"  "/:
11,24 SAY "DATE:"  "/:
11,40 SAY "WEIGHT CONTROL.......:"  "/:
12,8 SAY "RIFLE QUAL.......:"  "/:
12,25 SAY "WEIGHT.......:"  "/:
12,31 SAY "DATE:"  "/:
12,49 SAY "HEIGHT.......:"  "/:
13,6 SAY "PISTOL QUAL.......:"  "/:
13,25 SAY "":  "/:
13,31 SAY "DATE:"  "/:
13,49 SAY "HEIGHT.......:"  "/:
15,6 SAY "ELECTRONIC WARFARE.......:"  "/:
15,40 SAY "DRUG ABUSE.................:"  "/:
15,6 SAY "CLD WEATHER.................:"  "/:
16,40 SAY "ALCOHOL ABUSE.................:"  "/:
17,25 SAY "LAW & WAR.......:"  "/:
17,40 SAY "HUMAN RELATIONS.................:"  "/:
18,8 SAY "TRAINING MOS.................:"  "/:
18,40 SAY "PERSONAL AFFAIRS.................:"  "/:
19,8 SAY "LEADERSHIP.................:"  "/:
19,40 SAY "UCMJ.................:"
@ 20, & SAY "CHAR & MCPAL ED.....:"
@ 21, & SAY "CMMENTS:"
RETURN
**E. SCREEN DATA RETRIEVAL ROUTINE**

* Routine Name: Getdata.prg
* Module Name: Maintenance Module
* Version: 6.1.0.2
* Author: D. P. Haesler
* Date: 1 Dec 83
* Variables Used: vssn
* Variables Modified: NONE
* Variables Created: NONE
* Variables Released: NONE
* Files (opened/closed): a: est(opened/closed),
  b: est(opened/closed),
  b: qual(opened/closed),
  b: infcmisc(opened/closed)
* Temporary Files Created: NONE
* Using Subroutines: Viewitr, Updtitr, Deltitr
* Description: This routine retrieves the information
  stored in the database for the individual specified by
  vssn. It then arranges it in the screen format created
  by Itscreen.prg.

STORE AASSN TO VSSN
USE A: ESS INDEX A: PEI SSSN
FIND &VSS
  1, 12 SAY NAME
  1, 12 SAY BANK
  1, 61 SAY FRIMNOS
  1, 61 SAY SECMNOS
  2, 28 SAY ERTHDATE
  2, 49 SAY SSN
  3, 11 SAY COMPANY
  3, 23 SAY ILATOON
  3, 38 SAY JOINDATE
  3, 51 SAY EAS
  3, 65 SAY GASMASK
  11, 59 SAY WTCOMT
  12, 69 SAY WEIGHT
  13, 59 SAY HEIGHT
  21, 17 SAY COMMENT
USE B: ESS INDEX B: ES1SSN
FIND &VSS
  4, 28 SAY COC
  4, 28 SAY F111RAW
  4, 57 SAY F1111CSS
  4, 66 SAY F111DATE
  5, 28 SAY HIS
  5, 51 SAY F112RAW
  5, 57 SAY F112CSS
  5, 66 SAY F112DATE
  6, 28 SAY COD
  6, 28 SAY NPC
  7, 28 SAY INT
  7, 52 SAY MKS
  8, 62 SAY AID
  8, 62 SAY UNI
  9, 62 SAY TAC
USE B: QUAL INDEX B: QUALSSN
FIND &VS
  11, 20 SAY SWIMQUAL
  11, 9 CASE SWIMDATE
  12, 5 CASE RLSCORP
  12, 7 CASE RLQUAL

192
12 3E SAY RFLDATE
a 13:21 SAY PSTSCORE
d 13,27 SAY PSTQUAL
a 13:36 SAY PSTDATE
ESP B:INFCMISC INDEX E:IX4FOSSN
FIND EVSS a 15,29 SAY ELEC~kF
a 15,59 SAY DP UG
a 16,29 SAY CLDWTH1R
a 16,59 SAY ALCOHOL
a 17,29 SAY LAWWAR
a 17,59 SAY HUMREI
a 18,29 SAY MOS
a 18,59 SAY PERSAFFR
a 19,29 SAY LDRSHP
a 19,59 SAY LDRSHP
a 20,29 SAY LDRSHP

*USE 19
E. COMMENT RETRIEVAL ROUTINE

* Routine Name: Getcomnt.prg
* Module Name: Maintenance Module
* Version: 6.1.1
* Author: D.P. Haessler
* Date: 10 Jan 84
* Variables Used: none
* Variables Modified: none
* Variables Created: none
* Variables Released: none
* Files (opened/closed): a:comnt(c/c)
* Temporary Files Created: none
* Using Subroutines: Viewit.prg
* Description: This routine displays the contents of the
  comments data base file.

ERASE
A:CMNT INDEX A:CMNTSSN
FIND &ASSN
8,2 SAY INFOTXT1
8,3 SAY INFOTXT2
9,2 SAY INFOTXT3
9,5 SAY INFOTXT4
10,2 SAY INFOTXT5
10,5 SAY INFOTXT6
11,2 SAY INFOTXT7
11,5 SAY INFOTXT8
12,3 SAY INFOTXT9
12,4 SAY INFOTXT0
USE
RETURN
F. UPDATE CONTROL ROUTINE

* Routine Name: Updtitr.fng
* Module Name: Maintenance Module
* Version: 6.2.0
* Author: D.P. Haesler
* Date: 2 Dec 83
* Variables Used: aaadone, upname, upcomp, upagain, aassn
* Variables Modified: aaadone, upname, upcomp, upagain, aassn
* Variables Created: aaadone, upname, upcomp, upagain, aassn
* Variables Released: all like up?????, aassn, aaadone
* athru, aaccr
* Files (open/closed): a:pe:s (OPENED/CLOSED)
* Temporary Files Created: NONE
* Using Subroutines: Maintain.fng
* Description: This routine finds the ITR to be updated
* and controls the update routines.

STORE F TO AAADONE

DO WHILE .NOT. AAADONE
ERASE
@ 6,24 SAY "WHICH ITR IS TO BE"
@ 5,23 SAY "UPDATED OR CORRECTED?"
@ 10,23 SAY "NAME (LAST, FIRST MI.)"
ACCEPT "NAME
" TO UPNAME

STORE ! (UPNAME) TC UPNAME
USE A:PPERS INDEX a:ALPHAPERS
FIND EUPNAME

IF @ = 0
ERASE
@ 6,24 SAY "INDIVIDUAL IS NOT IN THE DATABASE"
@ 5,23 SAY "OR YOU HAVE ENTERED AN INCORRECT NAME"
STORE F TO UPComp

DO WHILE .NOT. UICOMP
ACCEPT "DO YOU WANT TO TRY AGAIN (Y/N)?":
TO UPAGAIN
DO CASE
CASE I (UPAGAIN) = 'Y'
STORE U TO UPComp

CASE I (UPAGAIN) = 'N'
STORE U TO UPComp
STORE U TO AAADONE
ENICASE
ENDDO
ELSE
STORE SSN TO AASSN
USE
DC ITSCRN
TO GETDATA
DC UPTDATA
STORE F TO UPcomp
DO WHILE .NOT. UICOMP
ERASE
ACCEPT "DO YOU WANT TO UPDATE ANOTHER ITR:
(Y/N)?":
TO UPAGAIN
DO CASE
CASE I (UPAGAIN) = 'Y'
STORE U TO UPTDATA
CASE I (UPAGAIN) = 'N'
STORE U TO UPTDATA
ENICASE
ENDDO

195
G. DATA UPDATE ROUTINE

* Routine Name: Update.prg
* Module Name: Maintenance Module
* Version: 6.2.0.1
* Author: D. P. Haesler
* Date: 8 Dec 83
* Variables Used: aathru, aacorr, upopt, inname, inrank, upcrams, inplace, insec, inbirth, insex, errcpt, incorp, inplate, injdate, iness, innum, incode, inptl, calcseq, wthru, wdate, inrptot, in =============================================================================

STORE E TO AATHRU
DO WHILE .NOT. AATHRU
    STORE E TO AACORR
    RELEASE ALL LIKE IN???????
    STORE "TO UPOPT
    @ 44, 0
    @ 22, 1 SAY "OPTION: FIELD TO BE CHANGED OR 'Q';
    TO QUIT,
    0 23, 1 SAY "OPTION ==>> GET UPOPT PICTURE "XXXXXX;
    XXXXXXXXXX
    XXXXXXXXXX
    READ
    SICRE !(UPCPT) TC UPOFT
    LO CASE
    CASE UPCPT = 'NAME'
    STORE T TC AACORR
    USE A:PEBS INDEX A:PERSON, A:ALPHPER
    FIND &AAPP
    STORE NAME TO INNAME
    @ 1, 12 GET INNAME PICTURE "XXXXXXXXXXXXXXXXXXXXX;
    XXXXXXXXXX
    XXXXXXXXXX
    READ
    REPLACE NAME WITH !(INNAME)
    USE
    CASE UPCPT = 'GRADE'
    STORE T TO AACORR
    USE A:PEBS INDEX A:PERSON
    FIND &AAPP
    STORE RANK TO INRANK
    @ 1, 1 GET INRANK PICTURE "XX" READ
STORE 1 TO ERROPT
DO ERROR
REPLACE RANK WITH !(INRANK)
USE

CASE UPLOPT = 'MOS'
STORE T TO AACORR
@ 23.0
@ 24.0
STORE P TO UPCORMOS
DO WHILE .NOT. UPCORMOS
@ 23.1 SAY "PRIMA Y OR SECONDARY?"
ACCEPT ENTER 1 OR 2 ====> TO UPAGAIN
DO CASE
CASE ! (UPAGAIN) = 'P'
STORE T TO UPCORMOS
USE A: PERS INDEX A: PERSIN, A: MCINDEX
FIND & AASSN
STORE PRIMEMOS TO INPRIME
@ 1.60 GET INPRIME PICTURE "99999"
READ
REPLACE PRIMEMOS WITH !(INPRIME)
USE
CASE ! (UPAGAIN) = 'S'
STORE T TO UPCORMOS
USE A: PERS INDEX A: PERSIN
FIND & AASSN
STORE SECMOS TO INSEC
@ 1.67 GET INSEC PICTURE "99999"
READ
REPLACE SECMOS WITH INSEC
USE
ENDCASE
ENDDO

CASE UPLOPT = 'BIRTHDATE'
STORE T TO AACORR
USE A: PERS INDEX A: PERSIN
FIND & AASSN
STORE BIRTHDATE TO INBIRTH
@ 2.42 GET INBIRTH PICTURE "XXXXXX"
READ
STORE 2 TO ERPLOPT
DO ERROR
REPLACE BIRTHDATE WITH INBIRTH
USE

CASE UPLOPT = 'SSN'
STORE T TO AACORR
USE A: PERS INDEX A: PERSIN
FIND & AASSN
STORE SSN TO INSSN
@ 2.48 GET INSSN PICTURE "999999999"
READ
REPLACE SSN WITH INSSN
USE
USE A: CMT INDEX A: CMNTSSN
FIND & AASSN
REPLACE SSN WITH INSSN
USE B: EST INDEX B: ESTSSN
FIND & AASSN
REPLACE SSN WITH INSSN
USE E: QUIL INDEX E: QUALSSN
FIND & AASSN
REPLACE SSN WITH INSSN
USE R: INCMSC INDEX R: INFOSSN
FIND & AASSN
REPLACE SSN WITH INSSN
STORE INSSN TO AASSN
USE

CASE UPOPT = "CO"
STORE T TC AACORR
USE A:PERSSN INDEX A:PERSSN
FIND GAASSN
STORE COMPANY TO INCOMP
@ 3,10 GET INCOMP PICTURE "XXXXXX"
READ
REPLACE COMPANY WITH !(INCOMP)
USE

CASE UPOPT = "PLAT"
STORE T TC AACORR
USE A:PERSSN INDEX A:PERSSN
FIND GAASSN
STORE PLAT TO INPLAT
@ 3,22 GET INPLAT PICTURE "XXXXXXXXX"
READ
REPLACE PLAT WITH !(INPLAT)
USE

CASE UPOPT = "JGIN"
STORE T TC AACORR
USE A:PERSSN INDEX A:PERSSN
FIND GAASSN
STORE JCINDATE TO INJNDATE
@ 3,37 GET INJNDATE PICTURE "XXXXXX"
READ
STORE 3 TC ERROPT
DO ERROR
REPLACE JCINDATE WITH INJNDATE
USE

CASE UPOPT = "EAS"
STORE T TC AACORR
USE A:PERSSN INDEX A:PERSSN
FIND GAASSN
STORE EAS TO INEAS
@ 3,50 GET INEAS PICTURE "XXXXXX"
READ
REPLACE EAS WITH INEAS
USE

CASE UPOPT = "GAS MASK"
STORE T TC AACORR
USE A:PERSSN INDEX A:PERSSN
FIND GAASSN
STORE GASMASK TC INGAS
@ 3,68 GET INGAS PICTURE "X"
READ
STORE 4 TC ERROPT
DO ERROR
REPLACE GASMASK WITH !(INGAS)
USE

CASE UPOPT = "MIL CODE OF CONDUCT"
STORE T TC AACORR
USE B:ESTSSN INDEX B:ESTSSN
FIND GAASSN
STORE CODE TO INCODE
@ 4,27 GET INCODE PICTURE "XXX"
READ
REPLACE CODE WITH !(INCODE)
USE

CASE UPOPT = "FPT1"
STORE T TC AACORR
USE B:EST INDEX E:ESTSSN
FIND &AASSN
STORE PFT1RAW TC INPF1
@ 4, 50 GET INPF1 PICTURE "999"
READ
STORE 5 TC ERROPT
DO ERROR
REPLACE PFT1RAW WITH INPF1
USE
STORE 1 TC CALCCPT
DO CALCUL
CASE UPORP = 'DATE'
STORE F TC WTHRU
STORE T TC AACORR
DO WHILE .NOT. WTHRU
  @ 23, 0
  @ 24, 0
  @ 25, 1 SAY " A..PFT1, B..PFT2, C..SWIM."
  D..RIFLE, E..PISTOL "WHICH DATE IS TO BE ENTERED ==> ";
  TO WDATE
  STORE 1(WDATE) TO WDATE
  DO CASE
  CASE WDATE = 'A'
    STORE T TO WTHRU
    USE B:EST INDEX B:ESTSSN
    FIND &AASSN
    STORE PFT1DATE TO INPF1DT1
    @ 4, 65 GET INPF1DT1 PICTURE "XXXXXX"
    READ
    STORE 6 TC ERROPT
    EC ERROR
    REPLACE PFT1DATE WITH INPF1DT1
    USE
  CASE WDATE = 'B'
    STORE T TO WTHRU
    USE B:EST INDEX B:ESTSSN
    FIND &AASSN
    STORE PFT2DATE TO INPF2DT2
    @ 5, 65 GET INPF2DT2 PICTURE "XXXXXX"
    READ
    STORE 8 TO ERROPT
    EC ERROR
    REPLACE PFT2DATE WITH INPF2DT2
    USE
  CASE WDATE = 'C'
    STORE T TO WTHRU
    USE B:QUAL INDEX B:QUALSSN
    FIND &AASSN
    STORE SWIMDATE TO INSWDT
    @ 11, 28 GET INSWDT PICTURE "XXXXXX"
    READ
    STORE 10 TC ERROPT
    EC ERROR
    REPLACE SWIMDATE WITH INSWDT
    USE
  CASE WDATE = 'D'
    STORE T TO WTHRU
    USE B:QUAL INDEX B:QUALSSN
    FIND &AASSN
    STORE RFLDATE TO INRFLDT
    @ 12, 35 GET INRFLDT PICTURE "XXXXXX"
    READ
    STORE 12 TO ERROPT
    EC ERROR

206
PLACE PIDATE WITH INPSTDT
USE

CASE WDATE = "E"
STORE T TC WTHRU
USE 3:QUAL INDEX 3:QUALSN
FIND &AASSN
STORE PIDATE TO INPSTDT
@ 13,35 GET INPSTDT PICTURE "XXXXXX"
READ
STORE 15 TC ERROPT
TC ERROR
REPLACE FRIDATE WITH INPSTDT
USE

ENDCASE
ENDDO

CASE UPOPT = 'HISTORY OF MARCOR'
STORE T TC AARCOR
USE 6:EST INDEX 6:ESTSN
FIND &AASSN
STORE HIS TO INHIS
@ 5, 27 GET INHIS PICTURE "XXX"
READ
REPLACE HIS WITH INHIS
USE

CASE UPOPT = 'PPT2'
STORE T TC AARCOR
USE 7:EST INDEX 7:ESTSN
FIND &AASSN
STORE PPT2RAW TO INPPT2
@ 5, 50 GET INPPT2 PICTURE "999"
READ
STORE 7 TC ERROPT
DO ERROR
REPLACE PPT2RAW WITH INPPT2
USE
STORE 2 TC CALCCEPT
DO CALCULATE

CASE UPOPT = 'DRILL'
STORE T TC AARCOR
USE 6:EST INDEX 6:ESTSN
FIND &AASSN
STORE CCL TO INDRILL
@ 6, 27 GET INDRILL PICTURE "XXX"
READ
REPLACE CCL WITH INDRILL
USE

CASE UPOPT = 'NBC'
STORE T TC AARCOR
USE 6:EST INDEX 6:ESTSN
FIND &AASSN
STORE RFC TO INNBC
@ 6, 47 GET INNBC PICTURE "XXXXXX"
READ
REPLACE RFC WITH INNBC
USE

CASE UPOPT = 'INTERFACE GUARD'
STORE T TC AARCOR
USE 7:EST INDEX 7:ESTSN
FIND &AASSN
STORE INT TO ININT
@ 7, 27 GET ININT PICTURE "XXX"
READ
REPLACE INT WITH ININT

201
USE
CASE UPOPT = 'MARKSMANSHIP'
STORE T TO A:ACORR
USE B:EST INDEX B:ESTSSN
FIND &AASSN
STORE MKS TO INMKS
@ 7,58 GET INMKS PICTURE "XXXXX"
READ
REPLACE MKS WITH INMKS
USE
CASE UPOPT = 'FIRST AID'
STORE T TO A:ACORR
USE B:EST INDEX B:ESTSSN
FIND &AASSN
STORE AID TO INAID
@ 8,27 GET INAID PICTURE "XXXXXXXXX"
READ
REPLACE AID WITH INAID
USE
CASE UPOPT = 'EQUIP & UNIFORM'
STORE T TO A:ACORR
USE B:EST INDEX B:ESTSSN
FIND &AASSN
STORE UNI TO INUNI
@ 9,27 GET INUNI PICTURE "XXX"
READ
REPLACE UNI WITH INUNI
USE
CASE UPOPT = 'INDIVIDUAL TACTICAL MEASURES'
STORE T TO A:ACORR
USE B:EST INDEX B:ESTSSN
FIND &AASSN
STORE TAC TO INTAC
@ 9,58 GET INTAC PICTURE "XXXX"
READ
REPLACE TAC WITH INTAC
USE
CASE UPOPT = 'SWIM QUAL'
STORE T TO A:ACORR
USE B:QUAL INDEX B:QUALSSN
FIND &AASSN
STORE SWIMQUAL TO INSWQU
@ 11,19 GET INSWQU PICTURE "XX"
READ
STORE 9 TO ERRCPT
DO ERROR
REPLACE SWIMQUAL WITH INSWQU
USE
CASE UPOPT = 'RIFLE QUAL'
STORE T TO A:ACORR
USE B:QUAL INDEX B:QUALSSN
FIND &AASSN
STORE RFISCOPE TO INRFSLSC
@ 12,20 GET INRFSLSC PICTURE "999"
READ
STORE 11 TO ERRCPT
DO ERROR
REPLACE RFISCOPE WITH INRFSLSC
USE
STORE 3 TO CALCPT
DO CALCULAT
CASE UPOPT = 'WEIGHT'

202
STORE T TC AACORR
USE A:PERSSN INDEX A:PERSSN
FIND &AASSN
STORE WEIGHT TO INWEIGHT
@ 12,58 GET INWEIGHT PICTURE "999"
READ
STORE 13 TO ERRCPT
DO ERROR
REPLACE WEIGHT WITH INWEIGHT
USE
STORE 5 TO CALCCEPT
DO CALCULAT

CASE UPLOPT = 'PISTOL QUAL'
STORE T TC AACORR
USE B:QUALSSN INDEX B:QUALSSN
FIND &BSSN
STORE PSTSCORE TC INPSTSC
@ 13,20 GET INPSTSC PICTURE "999"
READ
STORE 14 TO ERRCPT
DO ERROR
REPLACE PSTSCORE WITH INPSTSC
USE
STORE 4 TO CALCCEPT
DO CALCULAT

CASE UPLOPT = 'HEIGHT'
STORE T TC AACORR
USE A:PERSSN INDEX A:PERSSN
FIND &AASSN
STORE HEIGHT TO INHEIGHT
@ 13,58 GET INHEIGHT PICTURE "999"
READ
STORE 16 TO ERRCPT
DO ERROR
REPLACE HEIGHT WITH INHEIGHT
USE
STORE 5 TO CALCCEPT
DO CALCULAT

CASE UPLOPT = 'ELECTRONIC WARFARE'
STORE T TC AACORR
USE B:INFOMISC INDEX B:INFOSSN
FIND &BASSN
STORE ELECWAR TC INELEC
@ 15,28 GET INELEC PICTURE "XYYYYY"
READ
REPLACE ELECWAR WITH INELEC
USE

CASE UPLOPT = 'DRUG ABUSE'
STORE T TC AACORR
USE B:INFOMISC INDEX B:INFOSSN
FIND &BASSN
STORE DRUG TO INDRUG
@ 15,57 GET INDRUG PICTURE "XYYYYY"
READ
REPLACE DRUG WITH INDRUG
USE

CASE UPLOPT = 'COLD WEATHER'
STORE T TC AACORR
USE B:INFOMISC INDEX B:INFOSSN
FIND &BASSN
STORE CLDWTHR TC INCLDWR
@ 16,28 GET INCLDWR PICTURE "XYYYYY"
READ
REPLACE CLDWTHR WITH INCLDWR

203
CASE UPOPT = 'ALCOHOL ABUSE'
STORE T TO AACORR
USE B:INFCMISC INDEX B:INFOSSN
FIND &AASSN
STORE ALCOHOL TO INALCH
@ 16,58 GET INALCH PICTURE "XXXXXX"
READ
REPLACE ALCOHOL WITH INALCH
USE

CASE UPOPT = 'LAW OF WAR'
STORE T TO AACORR
USE B:INFCMISC INDEX B:INFOSSN
FIND &AASSN
STORE LAWWAR TO INLAW
@ 17,28 GET INLAW PICTURE "XXXXXX"
READ
REPLACE LAWWAR WITH INLAW
USE

CASE UPOPT = 'HUMAN RELATIONS'
STORE T TO AACORR
USE B:INFCMISC INDEX B:INFOSSN
FIND &AASSN
STORE HUMREL TO INHUM
@ 17,58 GET INHUM PICTURE "XXXXXX"
READ
REPLACE HUMREL WITH INHUM
USE

CASE UPOPT = 'TRAINING MOS'
STORE T TO AACORR
USE B:INFCMISC INDEX B:INFOSSN
FIND &AASSN
STORE MCS TO INMOS
@ 18,27 GET INMOS PICTURE "XXXX"
READ
REPLACE MCS WITH INMOS
USE

CASE UPOPT = 'PERSONAL AFFAIRS'
STORE T TO AACORR
USE B:INFCMISC INDEX B:INFOSSN
FIND &AASSN
STORE PERSAFFR TO INPERSAF
@ 18,58 GET INPERSAF PICTURE "XXXXXX"
READ
REPLACE PERSAFFR WITH INPERSAF
USE

CASE UPOPT = 'LEADERSHIP'
STORE T TO AACORR
USE B:INFCMISC INDEX B:INFOSSN
FIND &AASSN
STORE LDRSHIP TO INLDR
@ 19,28 GET INLDR PICTURE "XXXXXX"
READ
REPLACE LDRSHIP WITH INLDR
USE

CASE UPOPT = 'UCMJ'
STORE T TO AACORR
USE B:INFCMISC INDEX B:INFOSSN
FIND &AASSN
STORE UCMJ TO INUCMJ
@ 19,58 GET INUCMJ PICTURE "XXXXXX"
READ
REPLACE CCMJ WITH INUCMJ
USE

CASE UPCPT = 'CHAPTER & McGraw Ed'
STORE T TO AAOORR
USE B:INCMISC INDEX B:INPOSSN
FIND 8AASSN
STORE CHREESD TO INCHRMOR
@ 20,28 GET INCHRMOR PICTURE "XXXXXX"
READ
REPLACE CERMORED WITH INCHRMOR
USE

CASE UPCPT = 'COMMENTS'
STORE T TO AAOORR
USE A:PERSON INDEX A:PERSON
FIND 8AASSN
IF COMMENT
DC WRTCMMT
DC ITSCRN
DC GETDATA
ELSE
STORE & TO INCCMMNT
@ 21,16 GET INCCMMNT PICTURE "X"
READ
REPLACE COMMENT WITH INCCMMNT
USE
IF INCCMMNT
DO CRTCMMT
DO WRTCMMT
DO ITSCRN
DO GETDATA
ENDIF

CASE UPCPT = 'Q'
STORE T TO AAOORR
ENDCASE
ENDDC
ENDC
RETURN
E. ITR CREATION CONTROL ROUTINE

* Routine Name: Creatitr.prg
* Module Name: Maintenance Module
* Version: 6.3.0
* Author: D.F. Haesler
* Date: 7 Dec 83
* Variables Used: crccomp, aafini, cragain
* Variables Modified: same as above
* Variables Created: same as above
* Variables Released: same as above
* Files (opened/closed): NONE
* Temporary Files Created: NONE
* Using Subroutines: Maintain.prg
* Description: This routine controls the creation of new ITR's.

STORE F TO AAFINI
DO WHILE .NOT. AAFINI
ERASE
DC ITESCCN
LC INIATA
STORE F TO CRCCMP
ENDDO
ENDDC
RELEASE ALL LIKE CR??????
RELEASE AAFINI
USP

RETURN
I. ITB DATA_INITIALIZATION ROUTINE

* Routine Name: Indata.prg
* Module Name: Maintenance Module
* Version: 6.3.1
* Author: D.P. Kaeusler
* Date: 2 Dec 83
* Variables Used: aadate, vssr, in*, aassn
* Variables Modified: in*, aassn
* Variables Created: in*, aassn
* Files (opened/closed): a:pers(O/C), b:est(O/C),
  b:quals(O/C), b:trans(O/C), b:incsc(O/C)
* Temporary Files Created: None
* Using Subroutines: Creatitr.prg
* Description: This routine inserts the data into the
  training record format on the screen and checks for
  errors in the data entry. The data is then inserted in
  the data base.

STORE AALAI TO INDAIE
STORE "LAST, FIRST Mi."
STORE "TO INRANK
STORE ": TO INPSC
STORE "TO INSCMOS
STORE "DMMYY" TO INRETH
STORE "" TO INSSN
STORE " TO INCMP
STORE "DMMYY" TO INNDATE
STORE "DMMY" TO INNASS
STORE " TO INCCMP
STORE *** TO INCODE
STORE 000 TO INP1T1
STORE "DMMYY" TO INFRSTDT1
STORE "***" TO INHAS
STORE 000 TO INP1T2
STORE "DMMYY" TO INFRSTDT2
STORE "***" TO INDRILL
STORE "***" TO INREC
STORE "***" TO ININT
STORE "***" TO INPKS
STORE "***" TO INCAID
STORE "***" TO INNRT
STORE "***" TO INTAC
STORE "S " TO INSWRT
STORE "DMMYY" TO INSDAT
STORE "000" TO INP3LSC
STORE "DMMYY" TO INFLDT1
STORE "000" TO INWEIGHT
STORE "000" TO INP3TSC
STORE "DMMYY" TO INFLDT2
STORE "00" TO INGHT
STORE "***" TO INREC
STORE "***" TO INREX
STORE "***" TO INRMFP
STORE "***" TO INREM
STORE "***" TO INMD
STORE "***" TO INFRSAP
STORE "***" TO INLCR
STORE "***" TO INCMJ
STORE "***" TO INCMR
STORE " TO INCOMPT
STORE " TO INTPCCHR
STORE " TO INPT1C1
STORE " TO INPT2C1

207
STORE " " TO INFLQOU
STORE " " TO INFLSTOU

1,12 GET INNAME PICTURE "XXXXXXXXXXXXXXXXXXXXXX"  
1,60 GET INRANK PICTURE "XX"  
1,67 GET INSECNAME PICTURE "9999"  
2,22 GET INBRTH PICTURE "XXXXXX"  
3,10 GET INCOME PICTURE "XXXXXX"  
3,24 GET INFLAT PICTURE "XXXXXXX"  
3,50 GET INJNAME PICTURE "XXXXXX"  
3,69 GET INRFLQOU TO INFLSTOU  
4,62 GET INPRIM2-M PICTURE "9999"  
4,67 GET INSECMOS PICTURE "9599"  
5,51 GET INNAME PICTURE "XXXXXXXXXXXXXXXXXXXXXX"  
5,60 GET INRANK PICTURE "XX"  
5,67 GET INSECNAME PICTURE "9999"  
6,27 GET INPRIM2-M PICTURE "9999"  
6,60 GET INRFLQOU TO INFLSTOU  
7,51 GET INNAME PICTURE "XXXXXXXXXXXXXXXXXXXXXX"  
7,60 GET INRANK PICTURE "XX"  
7,67 GET INSECNAME PICTURE "9999"  
8,27 GET INNAME PICTURE "XXXXXXXXXXXXXXXXXXXXXX"  
8,60 GET INRANK PICTURE "XX"  
8,67 GET INSECNAME PICTURE "9999"  
9,51 GET INNAME PICTURE "XXXXXXXXXXXXXXXXXXXXXX"  
9,60 GET INRANK PICTURE "XX"  
9,67 GET INSECNAME PICTURE "9999"  
10,27 GET INNAME PICTURE "XXXXXXXXXXXXXXXXXXXXXX"  
10,60 GET INRANK PICTURE "XX"  
10,67 GET INSECNAME PICTURE "9999"  
11,27 GET INNAME PICTURE "XXXXXXXXXXXXXXXXXXXXXX"  
11,60 GET INRANK PICTURE "XX"  
11,67 GET INSECNAME PICTURE "9999"  
12,27 GET INNAME PICTURE "XXXXXXXXXXXXXXXXXXXXXX"  
12,60 GET INRANK PICTURE "XX"  
12,67 GET INSECNAME PICTURE "9999"  
13,27 GET INNAME PICTURE "XXXXXXXXXXXXXXXXXXXXXX"  
13,60 GET INRANK PICTURE "XX"  
13,67 GET INSECNAME PICTURE "9999"  
14,27 GET INNAME PICTURE "XXXXXXXXXXXXXXXXXXXXXX"  
14,60 GET INRANK PICTURE "XX"  
14,67 GET INSECNAME PICTURE "9999"  
15,27 GET INNAME PICTURE "XXXXXXXXXXXXXXXXXXXXXX"  
15,60 GET INRANK PICTURE "XX"  
15,67 GET INSECNAME PICTURE "9999"  
16,27 GET INNAME PICTURE "XXXXXXXXXXXXXXXXXXXXXX"  
16,60 GET INRANK PICTURE "XX"  
16,67 GET INSECNAME PICTURE "9999"  
17,27 GET INNAME PICTURE "XXXXXXXXXXXXXXXXXXXXXX"  
17,60 GET INRANK PICTURE "XX"  
17,67 GET INSECNAME PICTURE "9999"  
18,27 GET INNAME PICTURE "XXXXXXXXXXXXXXXXXXXXXX"  
18,60 GET INRANK PICTURE "XX"  
18,67 GET INSECNAME PICTURE "9999"  
19,27 GET INNAME PICTURE "XXXXXXXXXXXXXXXXXXXXXX"  
19,60 GET INRANK PICTURE "XX"  
19,67 GET INSECNAME PICTURE "9999"  
20,27 GET INNAME PICTURE "XXXXXXXXXXXXXXXXXXXXXX"  
20,60 GET INRANK PICTURE "XX"  
20,67 GET INSECNAME PICTURE "9999"  
21,16 GET INNAME PICTURE "XXXXXXXXXXXXXXXXXXXXXX"  
21,60 GET INRANK PICTURE "XX"  
21,67 GET INSECNAME PICTURE "9999"  

READ  
STORE INSSN TO AASSN  
STORE 1 TO ERROPT  
TO WHILE ERROPT < 17  
3 24,0 SAY "ERROR CHECKING IN PROGRESS"  
DC ERROPT  
STCR (ERROPT + 1) TO ERROPT  
ENDDC  
IF INCCMNT  
SAVE TO A:INFILE ALL LIKE IN??????  
RELEASE ALL LIKE IN??????  
RELEASE ALL EXCEPT AA??????  
DC CRICMMNT  
DC WRICMMNT  
RESTOR FROM A:INFILE ADDITIVE  
DO ITSCRN  
1,13 SAY INNAME  
1,62 SAY INRANK
say "storing data in the system on disk"

use a: index a: ssn, a: alphpers, a: mnsndx
append blank
replace name with 'inname"
replace ssn with inssn
replace rank with inrank, primenos with inprimem
replace secmos with insecmos, company with "(incomp)
replace platoon with "(inplat) join date with injndate
replace las with inlas, birthdate with inbrth
replace comment with incomment
replace height with inheight, weight with inweight
replace wtcnt with inwtcnt, gasmask with "(ingas)

use b:est index b: esissn
append blank
replace ssn with inssn, cog with "(icode)
replace his with "(inhis), cod with "(indrill)
replace int with "(iint), aid with "(inaid)
replace uni with "(inuni)
replace ft1 with "(inft1), ft1date with inft1dt1
replace ft2 with "(inft2), ft2date with inft2dt2
replace nec with "(innbc)
replace nks with "(innks), tac with "(intac)

use b:qual index b:qualssn
append blank

209
REPLACE SSN WITH INSSN, SWIMQUAL WITH !1INSWQU)
REPLACE SWIMDATE WITH INSWDT, FLFLDATE WITH INFLFLDT
REPLACE FFLSCORE WITH INFLPLSC
REPLACE FSTSCORE WITH INPSTSC
REPLACE FSTDATE WITH INFSTDT

USE E:INECMISC INDEX E:INFOSSN
APPEND BLANK
REPLACE SSN WITH INSSN, ELECWAR WITH !1INELEC)
REPLACE CLDWTHR WITH !1INCLUDT5)
REPLACE CANDWAR WITH !1INLAW), MOS WITH !1INMOS)
REPLACE DRUG WITH !1INDRUG), LERSHP WITH INLDR
REPLACE ALCCOL WITH !1INLCLCH), HUMREL WITH !1INHUM)
REPLACE EERSAFRF WITE !1INPERSAF), UCMJ WITH !1INUCMJ)
REPLACE CHRMRED WITH !1INCHRMCR)
STORE 1 TO CALCOPT
DO WHILE CALCOPT < 6
  DO CALCULAT
  SICRE (CALCOPT + 1) TO CALCPT
ENDDC

RELEASE ALL EXCEPT AA???????
ERASE
STORE AAASSN TO VSSN
IO ITESCEN
IO GETDATA

STORE F TO INDONE
DO WHILE .NOT. INDONE
  ACCEPT "IS THIS DATA CORRECT (Y/N) ?" TO INAGAIN
  DC CASE
  CASE !(INAGAIN) = 'Y'
    STORE T TO INDONE
  CASE !(INAGAIN) = 'N'
    DO UPDATA
    STORE T TO INDONE
  ENCASE
ENDDC
RELEASE ALL EXCEPT AA???????
RETURN
J. DATA ERROR CHECKING ROUTINE

* Routine Name: Error.prg
* Module Name: Maintenance Module
* Version: 6.3.1.1
* Author: D.P. Haeusler
* Date: 2 Dec 83
* Variables Used: aadate, in*
* Variables Modified: in*
* Variables Created: inerror
* Variables Released: none
* Files (opened/closed): none
* Temporary Files Created: none
* Using subroutines: Creatit.prg, updata.prg
* Description: This routine does the error checking for the information entered in each of the appropriate fields.
* This error checking is not very extensive because this is an prototype system but can be increased if the need arises.

STORE AALATE TO INDATE
STORE T TC INERROR
DO WHILE INERROR
STORE F TO INERROR
DC CASE
CASE ERROPT = 1
IF .NOT. ( $(INRANK, 1, 1) = 'E' .OR. $(INRANK, 1, 1) = 'O' )
STORE T TC INERROR
@ 24,0 SAY "@ 24,1 SAY "IMPROPER GRADE -- REENTER"
@ 4,51 GET INRANK PICTURE "99"
READ
ENDIF
CASE ERROPT = 2
IF .NOT. ( VAL($(INRTH, 1, 2)) < 31 ) .AND.
VAL($(INRTH, 2, 2)) < 13 ) .AND.
VAL($(INRTH, 2, 2)) <= VAL($(INDATE, 5, 2)) )
STORE T TC INERROR
@ 24,0 SAY "@ 24,1 SAY "IMPROPER DATE -- REENTER"
@ 2,22 GET INRTH PICTURE "999999"
READ
ENDIF
CASE ERROPT = 3
IF .NOT. ( VAL($(INJNDATE, 1, 2)) < 31 ) .AND.
VAL($(INJNDATE, 2, 2)) < 13 ) .AND.
VAL($(INJNDATE, 2, 2)) <= VAL($(INDATE, 5, 2)) )
STORE T TC INERROR
@ 24,0 SAY "@ 24,1 SAY "INERGIER DATE -- REENTER"
@ 4,37 GET INJNDATE PICTURE "999999"
READ
ENDIF
CASE ERROPT = 4
IF .NOT. ( $(INGAS) = 'S' .OR. $(INGAS) = 'M' .OR.
$(INGAS) = 'L' )
STORE T TC INERROR
@ 24,0 SAY "@ 24,1 SAY "IMPROPER GAS MASK SIZE -- REENTER"
@ 3,6B GET INGAS PICTURE "X"
READ
ENDIF
CASE ERROPT = 5
IF .NOT. ( INPT1 < 301)
STORE T TC INERROR
@ 24,0 SAY "@ 24,1 SAY "IMPROPER PFT SCORE -- REENTER"
@ 450 GET INPFT1 PICTURE "999"
READ
ENDIF
CASE ERROPT = 6
IF . NOT. (VAL($(INFEFT1, 1,2)) < 31 . AND.: 
   VAL($(INFEFT1, 2,2)) < 13 . AND.: 
   VAL($(INFEFT1, 5,2)) = VAL($(INDATE, 5,2)) . CR.:  
   INPFT1 = 'DDMMYY'
STORE T TC TNC TNC
   @ 24,0 SAY ""IMPROPER PFT DATE -- REENTER"
   @ 4,50 GET INPFT1 PICTURE "999999"
READ
ENDIF
CASE ERROPT = 7
IF . NOT. (INFT2 < 301) 
   STORE T TC TNC TNC
   @ 24,0 SAY ""IMPROPER PFT SCORE -- REENTER"
   @ 5,65 GET INPFT2 PICTURE "999"
READ
ENDIF
CASE ERROPT = 6
IF . NOT. (VAL($(INFEFT2, 1,2)) < 31 . AND.: 
   VAL($(INFEFT2, 2,2)) < 13 . AND.: 
   VAL($(INFEFT2, 5,2)) = VAL($(INDATE, 5,2)) . CR.: 
   INPFTDT2 = 'DDMMYY'
STORE T TC TNC TNC
   @ 24,0 SAY """IMPROPER PFT DATE -- REENTER"
   @ 5,65 GET INPFTDT2 PICTURE "999999"
READ
ENDIF
CASE ERROPT = 5
IF . NOT. ($INSWQU = 'S') . OR. $(INSWQU = 'WQ' . OR. INSWQU = 'WQ' . 
   STORE T TC TNC TNC
   @ 24,0 SAY ""IN MEEPER ENTRY -- REENTER"
   @ 11,19 GET INSWQU PICTURE "XX"
READ
ENDIF
CASE ERROPT = 10
IF . NOT. (VAL($(INSWD, 1,2)) < 31 . AND.: 
   VAL($(INSWD, 2,2)) < 13 . AND.: 
   VAL($(INSWD, 5,2)) = VAL($(INDATE, 5,2)) . CR.: 
   INSWDT = 'DDMMYY'
STORE T TC TNC TNC
   @ 24,0 SAY ""IMPROPER DATE -- REENTER"
   @ 11,28 GET INSWDT PICTURE "999999"
READ
ENDIF
CASE ERROPT = 11
IF . NOT. (INFLSC < '251') 
   STORE T TC TNC TNC
   @ 24,0 SAY ""IMPROPER RIFLE SCORE -- REENTER"
   @ 12,20 GET INFLSC PICTURE "999"
READ
ENDIF
CASE ERROPT = 12
IF . NOT. (VAL($(INRFLDT, 1,2)) < 31 . AND.: 
   VAL($(INRFLDT, 2,2)) < 13 . AND.: 
   VAL($(INRFLDT, 5,2)) = VAL($(INDATE, 5,2)) . CR.: 
   INRFLDT = 'DDMMYY'
STORE T TC TNC TNC
   @ 24,0 SAY ""IMPROPER DATE -- REENTER"
@ 12,35 GET INRLDI PICTURE "999999"
READ
ENDP
CASE ERRORT = 13
IF INWEIGHT > '300' .CR. INWEIGHT < '070'
STORE T TC INERROR
@ 24,0 SAY " "
@ 24,1 SAY " IMPROPER WEIGHT -- REENTER"
@ 12,58 GET INWEIGHT PICTURE "999"
READ
ENDP
CASE ERRORT = 14
IF (INESTSC < '301')
STORE T TC INERROR
@ 24,0 SAY " "
@ 24,1 SAY " IMPROPER PISTOL SCORE -- REENTER"
@ 13,20 GET INPSTSC PICTURE "999"
READ
ENDP
CASE ERRORT = 15
IF NOT. (VAL $(INPSTD,1,2)) < 31 .AND.:
VAL $(INPSTD,3,2)) < '13' .AND.:
VAL $(INPSTD,5,2)) <= VAL $(INDATE,5,2)) .CR.:
INPSTD = 'DDMMYY'
STORE T TC INERROR
@ 24,0 SAY " "
@ 24,1 SAY " IMPROPER DATE -- REENTER"
@ 13,35 GET INPSTLT PICTURE "999999"
READ
ENDP
CASE ERRORT = 16
IF NOT. (INHGHT < '78' .AND. INHGHT > '64')
STORE T TC INERROR
@ 24,0 SAY " "
@ 24,1 SAY " IMPROPER HEIGHT -- REENTER"
@ 13,58 GET INHGHT PICTURE "99"
READ
ENDP
ENCASE
ENDDC
RETURN
K. CALCULATION ROUTINE

* Routine Name: Calculate.prg
* Module Name: Maintenance Module
* Version: 6.3.1.2
* Author: D. F. Hausler
* Date: 2 Dec 83
* Variables Used: calcept, aassn, aadate, birthdate, age, mm, in
* Variables Modified: age, mm, in
* Variables Released: all except a??????
* Files (c/c): a:pers (c/c), b:qual (o/c), b:est (o/c)
* Temporary Files Created: none
* Using Subroutines: Update.prg, Indata.prg
* Description: This routine does the calculations that are triggered by the entry of data into certain fields of the ITR. These calculations reconstruct the tables found in the training orders.

DO CASE
  CASE CALCOPT = 1
    USE A:PERSON INDEX A:PERSSSN
    FIND &AASSN
    IF INPUT1 > 0
      STORE (VAL($(@AADATE,5,2)) -
      (VAL($(@BIRTHDATE,5,2))) TO AGE
      STORE (VAL($(@BIRTHDATE,3,2)) -
      (VAL($(@BIRTHDATE,3,2))) TO MM
      IF MM < 0
        STORE AGE - 1 TC AGE
      ENDLIF
      USE B:EST INDEX B:ESTSSN
      FIND &AASSN
      IF FFT1RAW >= 285
        STORE 'S' TO INPT1CL
        REPLACE INPT1CLSS WITH INPT1CL
        @ 4,57 SAY INPT1CL
        RELEASE INPT1CL
        USE
      ELSE
        IF AGE <= 26
          STORE FFT1RAW TC IPFT1RAW
        ENDFI
        IF AGE < 40 AND AGE > 26
          STORE (FFT1RAW + 25) TO IPFT1RAW
        ENDFI
        IF AGE > 39
          STORE (FFT1RAW + 50) TO IPFT1RAW
        ENDFI
        IF IPFT1RAW <= 134
          STORE 'U' TO INPT1CL
          REPLACE INPT1CLSS WITH INPT1CL
          @ 4,57 SAY INPT1CL
          RELEASE INPT1CL
          USE
          ENDIF
          IF IPFT1RAW <= 174 AND IPFT1RAW > 134
            STORE 'T' TO INPT1CL
            REPLACE INPT1CLSS WITH INPT1CL
            @ 4,57 SAY INPT1CL
            RELEASE INPT1CL
            USE
            ENDIF
            IF IPFT1RAW <= 224 AND IPFT1RAW > 174
              STORE 'Z' TO INPT1CL
              REPLACE INPT1CLSS WITH INPT1CL
              @ 4,57 SAY INPT1CL
              USE
            ENDIF
RELEASE INPF1CL
USE END IF
IF IPFT1RAW > 224
STORE Il' TO INPFT1CL
REPLACE PF"1T1CLS WITH NPF1T1CLS
a 4 57
SAY INPFT1CL
RELEASE INPF1CL
USE ENrif
CASE CALCOPT = 2
USE A:PERS INDEX
FIND ZAASSN
IF IWFFT2RAW > 0
SICRE (VAL(fAkDATc5,2))
VAL($BRI8HDATA,5 ~
SICRE (VAL($ AADTfrD))-
VA1
S (B RTHDAE,302)
TM M
STORE AGE - TC AGE
EN LIP
USE B:EST IbEEX
B:ESTSSN
FIND &AASSN
IF PFT2RAW = 285
STORE 'S' TO INPFT2CL
REPLACE E172CLSS WITH INPFT2CL
@ 5,57 SAY INPFT2CL
REL ASE INPFT2CI
USE ELSE
IF AGE <= 26
STORE EFT2RAW = IC
PFT2RAW END IF
AGE < 40 .ANT. AGE > 26
A.STORE (PF'T2RAW + 25) TO IPFT2RAW
END IF
AGE > ~ 9
STORE IPFT2RAW + 50) TO IPFT2RAW
END IF
IPFT2RAW <= 134
STORE 'U' TO INPFT2CL
REPLACE PFT2CLSS WITH INPFT2CL
a 5 57SAY INPFT2CL
RELEASE INPF1CL
USE Ell
DI F
IF IEFT2FAW <= 224 .AND.
IPFT2RAW > 174
STORE '2' TO INPFT2CL
REPLACE PFT2CLSS WITH INPFT2CL
a 5 51
SAY INPFT2CL
RELEASE INPF1CL
USE Y:N
DI F
215
ENDIF
ENDIF
CASE CALCOPT = 3
USE B:QUAL INDEX B:QUALSSN
FIND B:ASSN
IF RFLSCORE > '0'
IF RFLSCORE < '190'
STORE 'UN' TC INRFLQU
REPLACE RFLQUAL WITH INRFLQU
@ 12.27 SAY INRFLQU
RELEASE INRFLQU
ENDIF
IF RFLSCORE < '210' .AND. RFLSCORE >= '190'
STORE 'IN' TC INRFLQU
REPLACE RFLQUAL WITH INRFLQU
@ 12.27 SAY INRFLQU
RELEASE INRFLQU
ENDIF
IF RFLSCORE < '220' .AND. RFLSCORE >= '210'
STORE 'EX' TC INRFLQU
REPLACE RFLQUAL WITH INRFLQU
@ 12.27 SAY INRFLQU
RELEASE INRFLQU
ENDIF
IF RFLSCORE >= '220'
STORE 'ELL TO INRF1QU
REPLACE FELQUAL WITH INRFLQU
@ 12.27 SAY INRFLQU
RELEASE INRFLQU
ENDIF
USE
ENDIF
CASE CALCOPT = 4
USE B:QUAL INDEX B:QUALSSN
FIND B:ASSN
IF PSTSCORE > '0'
IF PSTSCORE < '180'
STORE 'UN' TC INPSTQU
REPLACE PSTQUAL WITH INPSTQU
@ 13.27 SAY INPSTQU
RELEASE INPSTQU
ENDIF
IF PSTSCORE < '210' .AND. PSTSCORE >= '180'
STORE 'IN' TC INPSTQU
REPLACE PSTQUAL WITH INPSTQU
@ 13.27 SAY INPSTQU
RELEASE INPSTQU
ENDIF
IF PSTSCORE < '250' .AND. PSTSCORE >= '210'
STORE 'EX' TC INPSTQU
REPLACE PSTQUAL WITH INPSTQU
@ 13.27 SAY INPSTQU
RELEASE INPSTQU
ENDIF
IF PSTSCORE >= '250'
STORE 'ELL TO INPSTQU
REPLACE FSTQUAL WITH INPSTQU
@ 13.27 SAY INPSTQU
RELEASE INPSTQU
ENDIF
USE
ENDIF
CASE CALCOPT = 5
USE A: PERS INDEX A: PERSSSN
FIND A:ASSN
STORE T TO INWTCONT
IF HEIGHT = '64' .AND. WEIGHT <= '160' .AND.:
WEIGHT >= '105'
STORE P TO INWTCONT

216
REPLACE WTCCNT WITH INWTCONT
@ 11,59 SAY INWTCONT
ENDIF

IF HEIGHT = "66", AND. WEIGHT <= "165", AND.
WEIGHT >= "106"
STORE P TO INWTCONT
REPLACE WTCCNT WITH INWTCONT
@ 11,59 SAY INWTCONT
ENDIF

IF HEIGHT = "66", AND. WEIGHT <= "170", AND.
WEIGHT >= "107"
STORE P TO INWTCONT
REPLACE WTCCNT WITH INWTCONT
@ 11,59 SAY INWTCONT
ENDIF

IF HEIGHT = "67", AND. WEIGHT <= "175", AND.
WEIGHT >= "111"
STORE P TO INWTCONT
REPLACE WTCCNT WITH INWTCONT
@ 11,59 SAY INWTCONT
ENDIF

IF HEIGHT = "68", AND. WEIGHT <= "181", AND.
WEIGHT >= "115"
STORE P TO INWTCONT
REPLACE WTCCNT WITH INWTCONT
@ 11,59 SAY INWTCONT
ENDIF

IF HEIGHT = "69", AND. WEIGHT <= "186", AND.
WEIGHT >= "119"
STORE P TO INWTCONT
REPLACE WTCCNT WITH INWTCONT
@ 11,59 SAY INWTCONT
ENDIF

IF HEIGHT = "70", AND. WEIGHT <= "192", AND.
WEIGHT >= "122"
STORE P TO INWTCONT
REPLACE WTCCNT WITH INWTCONT
@ 11,59 SAY INWTCONT
ENDIF

IF HEIGHT = "71", AND. WEIGHT <= "197", AND.
WEIGHT >= "127"
STORE P TO INWTCONT
REPLACE WTCCNT WITH INWTCONT
@ 11,59 SAY INWTCONT
ENDIF

IF HEIGHT = "72", AND. WEIGHT <= "203", AND.
WEIGHT >= "131"
STORE P TO INWTCONT
REPLACE WTCCNT WITH INWTCONT
@ 11,59 SAY INWTCONT
ENDIF

IF HEIGHT = "73", AND. WEIGHT <= "209", AND.
WEIGHT >= "135"
STORE P TO INWTCONT
REPLACE WTCCNT WITH INWTCONT
@ 11,59 SAY INWTCONT
ENDIF

IF HEIGHT = "74", AND. WEIGHT <= "214", AND.
WEIGHT >= "139"
STORE P TO INWTCONT
REPLACE WTCCNT WITH INWTCONT
@ 11,59 SAY INWTCONT
ENDIF

IF HEIGHT = "75", AND. WEIGHT <= "219", AND.
WEIGHT >= "143"
STORE P TO INWTCONT
REPLACE WTCCNT WITH INWTCONT
@ 11,59 SAY INWTCONT
ENDIF
IF HEIGHT = '76' .AND. WEIGHT <= '225' .AND.:
    WEIGHT >= '147'
STORE P TO INWTCONT
REPLACE WTCCNT WITH INWTCONT
@ 11,59 SAY INWTCONT
ENDIF
IF HEIGHT = '77' .AND. WEIGHT <= '230' .AND.:
    WEIGHT >= '151'
STORE P TO INWTCONT
REPLACE WTCCNT WITH INWTCONT
@ 11,59 SAY INWTCONT
ENDIF
IF HEIGHT = '78' .AND. WEIGHT <= '235' .AND.:
    WEIGHT >= '153'
STORE P TO INWTCONT
REPLACE WTCCNT WITH INWTCONT
@ 11,59 SAY INWTCONT
ENDIF
IF INWTCONT
    REPLACE WTCCNT WITH INWTCONT
@ 11,59 SAY INWTCONT
ENDIF
USE
ENDCASE
RETURN
L. COMMENT UPDATE ROUTINE

* Routine Name: Wrtcment.prg
* Module Name: Maintenance Module
* Version: 6.3.1.3
* Author: D.F. Haaseler
* Date: 19 Jan 84
* Variables Used: assn, intxt*
* Variables Modified: intxt*
* Variables Created: intxt*
* Variables Released: all like intxt*
* Files (opened/closed): a:cmmt (o/c)
* Temporary Files Created: none
* Using Subroutines: indata.prg, update.prg
* Description: This routine updates the information stored in the data base on an individual.

ERASE
89.13 SAY "COMMENT SECTION"
USE A:CMMT INDEX A:CMMTSS
FIND &ASSN
TP, NCT, # = 0
8.1 SAY INXTXT1
8.2 1 SAY INXTXT2
8.3 1 SAY INXTXT3
8.4 1 SAY INXTXT4
10.2 SAY INXTXT5
10.1 SAY INXTXT6
11.2 SAY INXTXT7
11.1 SAY INXTXT8
12.2 SAY INXTXT9
12.1 SAY INXTXT0
STORE INXTXT1 TO INTXT1
STORE INXTXT2 TO INTXT2
STORE INXTXT3 TO INTXT3
STORE INXTXT4 TO INTXT4
STORE INXTXT5 TO INTXT5
STORE INXTXT6 TO INTXT6
STORE INXTXT7 TO INTXT7
STORE INXTXT8 TO INTXT8
STORE INXTXT9 TO INTXT9
STORE INXTXT0 TO INTXT0
11.1 GET INXTXT1 PICTURE "XXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
11.2 GET INXTXT2 PICTURE "XXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
12.1 GET INXTXT3 PICTURE "XXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
12.2 GET INXTXT4 PICTURE "XXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
10.1 GET INXTXT5 PICTURE "XXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
10.2 GET INXTXT6 PICTURE "XXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
11.1 GET INXTXT7 PICTURE "XXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
11.2 GET INXTXT8 PICTURE "XXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
12.1 GET INXTXT9 PICTURE "XXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
12.2 GET INXTXT0 PICTURE "XXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
REPLACE INXTXT1 WITH INTXT1
REPLACE INXTXT2 WITH INTXT2
REPLACE INXTXT3 WITH INTXT3
REPLACE INXTXT4 WITH INTXT4
REPLACE INXTXT5 WITH INTXT5
REPLACE INXTXT6 WITH INTXT6
REPLACE INXTXT7 WITH INTXT7
REPLACE INXTXT8 WITH INTXT8
REPLACE INXTXT9 WITH INTXT9
REPLACE INXTXT0 WITH INTXT0
ENDIP
USE
RELEASE ALL LIKE INTXT?
RETURN
M. COMMENT CREATION ROUTINE

* Routine Name: CrtCmnt.prg
* Module Name: Maintenance Module
* Author: D.F. Haeusler
* Date: 10 Jan 84
* Variables Used: newtext, aassn
* Variables Modified: newtext
* Variables Created: newtext
* Temporary Files Created: none
* Using Subroutines: indata.prg, update.prg
* Description: This routine creates a comment data base
* file for the individual that corresponds to the aassn.
* This file when created has 10 fields of 25 characters
* each and is filled with asterisks.

STORE "************" TO NEWTXT
LSP A:CMNT INDEX A:CMNTSSN
APPEND BLANK
REPLACE SSN WITH AASSN, INFTEXT1 WITH NEWTXT
REPLACE INFTEXT2 WITH NEWTXT, INFTEXT3 WITH NEWTXT
REPLACE INFTEXT5 WITH NEWTXT, INFTEXT6 WITH NEWTXT
REPLACE INFTEXT7 WITH NEWTXT, INFTEXT8 WITH NEWTXT
REPLACE INFTEXT9 WITH NEWTXT, INFTEXT10 WITH NEWTXT
REPLACE INFTEXT1 WITH NEWTXT
LSP
RELEASE NEWTXT
RETURN
N. DELETE ITR ROUTINE

* Routine Name: Deltrim.prq
* Module Name: Maintenance Module
* Version: 6.4.0
* Author: D.P. Haeusler
* Date: 20 Jan 84
* Variables Used: del*, assn
* Variables Modified: del*, assn
* Variables Created: del*
* Variables Released: del*
* Files (opened/closed): a:pers(o/c); b:est(o/c)
  b:qual(o/c); b:infomisc(o/c)
* Temporary Files Created: none
* Using Subroutines: maintain.prg
* Description: This routine deletes records in the data
  base. It first checks to see if the record is in the
  data base, displays the record, queries the user if this
  is the record to be deleted, and deletes the record.

STORE F TO DELPINI
DO WHILE .NOT. DELPINI
  ERASE
  @ 12, 30 SAY "ITR TO BE DELETED" 
  ACCEPT " NAME (LAST, FIRST MI.) ==>> " TO DELNAME
  USE A:APERS INDEX A:ALPHPERS, A:PERSSN, A:MOSSNOX
  FIND DELNAME
  IF .NCT. # = 0
    SIGE SSN TO AASSN
    LC ITFSRN
    DO GETDATA
    STORE F TO DELTHRU
    DO WHILE .NOT. DELTHRU
      @ 23, 0 SAY "IS THIS ITR TO BE DELETED? 
      ACCEPT " ENTER YES OR NO ==>> " TO DELCHK
      IF !(DELCHK) = 'NO'
        STORE T TO DELTHRU
        ELSE
          IF !(DELCHK) = 'YES'
            ERASE
            @ 12, 30 SAY "DELETING ITR"
            STORE T TO DELTHRU
            USE A:APERS INDEX A:ALPHPERS, A:PERSSN,
            A:MOSSNOX
            FIND &AASSN
            IF .NCT. # = 0
              DELETE PACK
            ENDIF
            USE A:CMNMT INDEX A:CMNMTSSN
            FIND &AASSN
            IF .NCT. # = 0
              DELETE PACK
            ENDIF
            USE B:QUAL INDEX B:QUALSSN
            FIND &AASSN
            IF .NCT. # = 0
              DELETE PACK
            ENDIF
            USE B:EST INDEX B:ESTSSN
            FIND &AASSN
            IF .NCT. # = 0
              DELETE PACK
            ENDIF
            USE B:INFOMISC INDEX A:INFOSSN
FIND SAASSN
IF NCT. # = 0
DELETE PACK
ENDIF
ENDIF
ENDIF
ENDDO
ENDDO
ENDDO
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDDO
ENDDC
RELEASE ALL EXCEPT AA
RETURN
C. SYSTEM FUNCTION CONTROL ROUTINE

* Routine Name: System.prg
* Module Name: Maintenance Module
* Versions: 6.5.0
* Author: D.P. Haeusler
* Date: 30 Jan 1984
* Variables Used: aacomple, sysopt
* Variables Modified: aacomple, sysopt
* Variables Created: aacomple, sysopt
* Files (opened/closed): none
* Temporary Files Created: none
* Using Subroutines: Maintain.prg
* Description: This module creates the menu for all the system function modifications. It is a restricted module and can only be accessed by a user with a authorization level of 1. The module then can choose the appropriate subroutine corresponding to the option selected.

STORE P TO AACOMPLE
DO WHILE .NOT. AACOMPLE
  ERASE
  6,29 SAY " SYSTEM FUNCTION MENU"
  8,24 SAY " SELECT ONE OF THE FOLLOWING OPTIONS"
  10,31 SAY " A. INSTALL SYSTEM"
  11,31 SAY " A. SYSTEM RESET (YEARLY)"
  12,31 SAY " A. ACCESS LIST MAINTENANCE"
  13,31 SAY " A. CUIT TO MAINTENANCE MENU"
  ACCEPT " ENTER OPTION ===>" TO SYSOPT
  DC CASE
   CASE I(SYSOPT) = "A"
      STORE P TO ACOMPL
      DO: INSTALL
   CASE I(SYSOPT) = "A"
      STORE P TO ACOMPL
      DO: SYSRESFY
   CASE I(SYSOPT) = "A"
      STORE P TO ACOMPL
      DO: RSTACCESS
   CASE I(SYSOPT) = "A"
      STORE T TO ACOMPL
ENDCASE
RELEASE ALL EXCEPT AA??????
RELEASE AACOMPLE
RETURN
F. SYSTEM INSTALLATION ROUTINE

* Routine Name: Install.prg
* Module Name: Maintenance Module
* Version: 6.5.1
* Author: D.P. Heusler
* Date: 7 Feb 84
* Variables Used: *cc, *plt, crnty;
* Variables Modified: *cc, *plt, crnty;
* Variables Created: none
* Variables Released: all
* Files (opened/closed): a:unitmem.mem, a:memdisk.mem
* Temporary Files Created: temp mem
* Using Subroutines: System.prg
* Description: This routine stores the appropriate values in the memory files that the system uses to describe the units that are in the battalion. It also establishes the current year that is used for error checking and calculations.

SAVE TO TEMP
RELEASE ALL
ERASE
ESTORE DOCM A:UNITMEM
2. 30 SAY "SYSTEM INSTALLATION"
3. 15 SAY "ENTER THE NAMES OF YOUR COMPANY SIZE UNITS"
4. 34 GET CNECO PICTURE "XXXXXXX"
5. 15 SAY "COMPANY/EATTERY #1"
6. 34 GET TWOCO PICTURE "XXXXXXX"
7. 15 SAY "COMPANY/EATTERY #2"
8. 34 GET THREECO PICTURE "XXXXXXX"
9. 15 SAY "COMPANY/EATTERY #3"
10. 34 GET FOURCO PICTURE "XXXXXXX"
READ
STORE 1(CNECO). TO ONECO
STORE 1(TWOCO) TO TWCCO
STORE 1(THREECO) TO THREECO
STORE 1(FOURCO) TO FOURCO
ERASE
2. 30 SAY "SYSTEM INSTALLATION"
3. 15 SAY "ENTER THE NAMES OF PLATOON SIZED UNITS"
4. 26 GET CNEPLT PICTURE "XXXXXXXXX"
5. 15 SAY "PLATOON #1"
6. 26 GET TWOPL PICTURE "XXXXXXXXX"
7. 15 SAY "PLATOON #2"
8. 26 GET THREEPL PICTURE "XXXXXXXXX"
9. 15 SAY "PLATOON #3"
10. 26 GET FOURPL PICTURE "XXXXXXXXX"
READ
STORE 1(CNEPLT) TO ONEPLT
STORE 1(TWOPL) TO TWPLT
STORE 1(THREEPL) TO THREEPLT
STORE 1(FOURPLT) TO FOURPLT
SAVE TO UNITMEM
ERASE
RELEASE ALL
RESTORE FROM A:MEMDISK
4. 15 SAY "ENTER THE CURRENT YEAR"
ACCEPT "YEAR (YY) =="" TO CRNTYR
SAVE TO A:MEMDISK
RELEASE ALL
RESTORE FROM TEMP
RETURN
Q. SYSTEM RESET ROUTINE

* Routine Name: Sysreset.prq
* Module Name: Maintenance Module
* Version: 6.5.2
* Author: C.P. Hauesler
* Date: 4 Feb 84
* Variables Used: crnty, reset, rgreset, done, q*, *cc
* Variables Modified: crnty, reset, rgreset, done
* Variables Created: reset, rgreset, done
* Variables Released: all
* Files opened/closed): a:memdisk.mem, a:unitmem.mwm,
* Temporary Files Created: tez,..mem
* Using Subroutines: System.prq
* Description: This routine sets all the values to the
* default value for the fields that are reset yearly. The
* information is then stored in the data base for each
* individual. It also stores the yearly projections on
* the range quotas to a memory file that can be recalled by
* routines that use these values for calculations.

SAVE TO TEM
RELEASE ALL
RESTC "A:MEMDISK ADDITIVE"
ACCE "ENTER THE CURRENT YEAR" "ENTER THE CURRENT YEAR (YY) ==>" TO CRNTYR

SAVE TO a:MEMDISK
RELEASE ALL
STORE F TO DONE
DO WHILE .NOT. DONE
IF RESET
SAY "DO YOU WANT TO RESET RANGE QUOTAS FOR:
THE YEAR?"
ACCEPT "ENTER (Y OR N) ==>" TO RESET
IF (RESET) = 'Y'
STORE T TO DONE
STORE T TO RGRESET
ELSE
IF (RESET) = 'N'
STORE T TO LCNE
STORE P TO RGRESET
ENDIF
ENDIF
ENDDO
IF RCRESET
ERASE
RESTORE FROM A:RNGQTAS
RESTORE FROM A:UNITMEM ADDITIVE
SAY "ENTER RANGE QUOTAS AS PROJECTED;"
FOR THE YEAR"
1 TO 15 SAY "FOR UNIT: "
1 TO 25 SAY CNECO
6 TO 15 SAY "JANUARY"
6 TO 23 GET QJAN PICTURE "999"
7 TO 15 SAY "FEBRUARY"
7 TO 24 GET QFEB PICTURE "999"
8 TO 15 SAY "MARCH"
8 TO 21 GET QMAR PICTURE "999"
9 TO 15 SAY "APRIL"
9 TO 21 GET QA04 PICTURE "999"
10 TO 15 SAY "MAY"
10 TO 19 GET QMAY PICTURE "999"
11 TO 15 SAY "JUNE"
11 TO 20 GET QJUN PICTURE "999"
12 TO 15 SAY "JULY"
12 TO 20 GET QJUL PICTURE "999"
13 TO 15 SAY "AUGUST"
READ
RELEASE ALL EXCEPT Q???
SAVE TO A:BRNGQTAS
RELEASE ALL
RESTORE FROM A:BRNGQTAS
RESTORE FROM UNITMEM ADDITIVE

3.15 SAY "ENTER RANGE QUOTAS AS PROJECTED;"
FOR THE YEAR,
4.15 SAY "FOR UNIT:"
4.25 SAY "JANUARY"
5.23 GET QJAN PICTURE "999"
6.15 SAY "FEBRUARY"
7.24 GET QFEB PICTURE "999"
8.15 SAY "MARCH"
9.21 GET QMAR PICTURE "999"
10.15 SAY "APRIL"
10.19 GET QAPR PICTURE "999"
11.15 SAY "MAY"
12.20 GET QJUN PICTURE "999"
12.15 SAY "JUNE"
12.20 GET QJUL PICTURE "999"
13.15 SAY "JULY"
13.22 GET QAUG PICTURE "999"
14.19 SAY "AUGUST"
14.19 SAY "SEPTEMBER"
14.25 GET QSEP PICTURE "999"
15.15 SAY "OCTOBER"
15.23 GET QCT PICTURE "999"
16.15 SAY "NOVEMBER"
16.24 GET QNOV PICTURE "999"
17.15 SAY "DECEMBER"
17.24 GET QDEC PICTURE "999"

READ
RELEASE ALL EXCEPT Q???
SAVE TO A:BRNGQTAS
RELEASE ALL
RESTORE FROM A:BRNGQTAS
RESTORE FROM UNITMEM ADDITIVE
3.15 SAY "ENTER RANGE QUOTAS AS PROJECTED;"
FOR THE YEAR,
4.15 SAY "FOR UNIT:"
4.25 SAY "JANUARY"
5.23 GET QJAN PICTURE "999"
6.15 SAY "FEBRUARY"
7.24 GET QFEB PICTURE "999"
8.15 SAY "MARCH"
9.21 GET QMAR PICTURE "999"
10.15 SAY "APRIL"
10.19 GET QAPR PICTURE "999"
11.15 SAY "MAY"
11.20 GET QJUN PICTURE "999"
12.15 SAY "JUNE"
12.20 GET QJUL PICTURE "999"
13.15 SAY "JULY"
13.22 GET QAUG PICTURE "999"
14.19 SAY "AUGUST"
14.19 SAY "SEPTEMBER"
14.25 GET QSEP PICTURE "999"
15.15 SAY "OCTOBER"
15.23 GET QCT PICTURE "999"
16.15 SAY "NOVEMBER"
16.24 GET QNOV PICTURE "999"
17.15 SAY "DECEMBER"
17.24 GET QDEC PICTURE "999"

READ
RELEASE ALL EXCEPT Q???
SAVE TO A:BRNGQTAS
RELEASE ALL
RESTORE FROM A:BRNGQTAS
RESTORE FROM UNITMEM ADDITIVE
3.15 SAY "ENTER RANGE QUOTAS AS PROJECTED;"
FOR THE YEAR,
4.15 SAY "FOR UNIT:"
4.25 SAY "JANUARY"
5.23 GET QJAN PICTURE "999"
6.15 SAY "FEBRUARY"
7.24 GET QFEB PICTURE "999"
8.15 SAY "MARCH"
9.21 GET QMAR PICTURE "999"
10.15 SAY "APRIL"
10.19 GET QAPR PICTURE "999"
11.15 SAY "MAY"
11.20 GET QJUN PICTURE "999"
12.15 SAY "JUNE"
12.20 GET QJUL PICTURE "999"
13.15 SAY "JULY"
13.22 GET QAUG PICTURE "999"
14.19 SAY "AUGUST"
14.19 SAY "SEPTEMBER"
14.25 GET QSEP PICTURE "999"
15.15 SAY "OCTOBER"
15.23 GET QCT PICTURE "999"
16.15 SAY "NOVEMBER"
16.24 GET QNOV PICTURE "999"
17.15 SAY "DECEMBER"
17.24 GET QDEC PICTURE "999"

READ
RELEASE ALL EXCEPT Q???
SAVE TO A:BRNGQTAS
RELEASE ALL
RESTORE FROM A:BRNGQTAS
RESTORE FROM UNITMEM ADDITIVE
3.15 SAY "ENTER RANGE QUOTAS AS PROJECTED;"
FOR THE YEAR,
4.15 SAY "FOR UNIT:"
4.25 SAY "JANUARY"
5.23 GET QJAN PICTURE "999"
6.15 SAY "FEBRUARY"
7.24 GET QFEB PICTURE "999"
8.15 SAY "MARCH"
9.21 GET QMAR PICTURE "999"
10.15 SAY "APRIL"
10.19 GET QAPR PICTURE "999"
11.15 SAY "MAY"
11.20 GET QJUN PICTURE "999"
12.15 SAY "JUNE"
12.20 GET QJUL PICTURE "999"
13.15 SAY "JULY"
13.22 GET QAUG PICTURE "999"
14.19 SAY "AUGUST"
14.19 SAY "SEPTEMBER"
14.25 GET QSEP PICTURE "999"
15.15 SAY "OCTOBER"
15.23 GET QCT PICTURE "999"
16.15 SAY "NOVEMBER"
16.24 GET QNOV PICTURE "999"
17.15 SAY "DECEMBER"
17.24 GET QDEC PICTURE "999"
13, 15  SAY "AUGUST"
13, 22  GET QAUG PICTURE "999"
14, 15  SAY "SEPTEMBER"
14, 25  GET QSEP PICTURE "999"
15, 15  SAY "OCTOBER"
15, 23  GET QCCT PICTURE "999"
16, 15  SAY "NOVEMBER"
16, 24  GET QNOV PICTURE "999"
17, 15  SAY "DECEMBER"
17, 24  GET QDEC PICTURE "999"
READ
RELEASE ALL EXCEPT Q???
SAVE TO A:CRNGQTAS
RELEASE ALL
RESTORE FROM A:DRBGQTAS
RESTORE FROM UNITMEM ADDITIVE
3, 15  SAY "ENTER RANGE COUNT AS PROJECTED:
FOR THE YEAR:
4, 15  SAY "FOR UNIT:"
4, 24  SAY "FOURCO"
6, 15  SAY "JANUARY"
6, 23  GET QJAN PICTURE "999"
7, 15  SAY "FEBRUARY"
7, 24  GET QFEB PICTURE "999"
8, 15  SAY "MARCH"
8, 21  GET QMAR PICTURE "999"
9, 15  SAY "APRIL"
9, 21  GET QAPR PICTURE "999"
10, 15  SAY "MAY"
10, 19  GET QMAT PICTURE "999"
11, 15  SAY "JUNE"
11, 20  GET QJUN PICTURE "999"
12, 15  SAY "JULY"
12, 20  GET QJUL PICTURE "999"
13, 15  SAY "AUGUST"
13, 22  GET QAUG PICTURE "999"
14, 15  SAY "SEPTEMBER"
14, 25  GET QSEP PICTURE "999"
15, 15  SAY "OCTOBER"
15, 23  GET QCCT PICTURE "999"
16, 15  SAY "NOVEMBER"
16, 24  GET QNOV PICTURE "999"
17, 15  SAY "DECEMBER"
17, 24  GET QDEC PICTURE "999"
READ
RELEASE ALL EXCEPT Q???
SAVE TO A:BRNGQTAS
RELEASE ALL
ENDIF
RESTORE FROM TEMP
STORE F TO DONE
DO WHILE .NOT. DONE
ERASE
3, 15  SAY "DO YOU WANT TO RESET YEARLY TRAINING;
VALUES"
4, 15  SAY "IN THE DATA BASE FOR THE BATTALION?"
ACCEPT "" ENTER (Y OR N) ===>"" TO RESET
IF !(RESET) = 'Y',
STORE T TO DONE
USE F:EST
REPLACE ALL CGC WITH "***"
REPLACE ALL HIS WITH "***"
REPLACE ALL COD WITH "***"
REPLACE ALL INT WITH "*******"
REPLACE ALL ALL WITH "***********"
REPLACE ALL UNI WITH "***"
REPLACE ALL PIRAN WITH "200"
REPLACE ALL PPIT DATE WITH "DDMMYY"
REPLACE ALL PPT1CLSS WITH "*"
REPLACE ALL PPT2RAW WITH 000
REPLACE ALL PPT2DATE WITH "DDMMYY"
REPLACE ALL PPT2CLSS WITH "*"
REPLACE ALL NBC WITH "*********
REPLACE ALL MKS WITH "**********"
REPLACE ALL TAC WITH "**********"
USE E:QUAL
REPLACE ALL BPISCORE WITH "***"
REPLACE ALL RELQUAL WITH "***"
REPLACE ALL RELATE WITH "DDMMYY"
REPLACE ALL PSTSCORE WITH "***"
REPLACE ALL PSTQUAL WITH "***"
REPLACE ALL PSTDATE WITH "DDMMYY"
USE
ELSE
IF I(RESET) = 'K'
STORE T TO LONE
ENDIF
ENDIF
ENDDC
RELEASE ALL EXCEPT AA??????
RETURN
E. SYSTEM ACCESS CONTROL ROUTINE

* Routine Name: Rstaccess.prg
* Module Name: Maintenance Module
* Version: 6.5.3
* Author: D.P. Haeusler
* Date: 8 Feb 84
* Variables Used: thru, ac*, more, lines, done, cont
* Variables Modified: same as above
* Variables Created: same as above
* Files (opened/closed): a:security(o/c)
* Temporary Files Created: none
* Using Subroutines: System.prg
* Description: This routine allows users with an access level of 1 to update the access list.

STORE P TO THRU
TO WHILE .NOT. THRU
ERASE
@ 8,28 SAY " SECURITY ACCESS LIST"
@ 9,28 SAY " CHOOSE OPTION TO BE EXECUTED"
@ 11,24 SAY " L...LIST ALL USERS"
@ 12,24 SAY " A...ADD TO ACCESS LIST"
@ 14,24 SAY " Q...QUIT TO SYSTEM FUNCTION MENU"
ACCEP " ENTER OPTION ===>" TO ACCOPT
DC CASE
CASE ! (ACCOPT) = "L"
USE A:SECURITY INDEX A:ACCNAME
DO WHILE .NOT. EOF
ERASE
@ 3,24 SAY " SECURITY ACCESS LIST"
@ 5,1 SAY " NAME USER NO.
PASSWORD LEVEL"
STORE 7 TO LINES
DO WHILE .NOT. EOF AND. LINES < 22
@ LINES,1 SAY NAME
@ LINES,20 SAY USERID
@ LINES,41 SAY USERPASS
@ LINES,56 SAY AUTHLEV
SKIP
STORE (LINES + 2 ) TO LINES
IF LINES > 22
@ 23,15 SAY "*** PRESS ANY KEY TO CONTINUE ***"
CONTINUE ***
SET CONSOLE CP
WAIT
SET CONSOLE CN
ENDIF
ENDDC
ENDDO
@ 23,15 SAY "*** PRESS ANY KEY TO CONTINUE ***"
SET CONSOLE CP
WAIT
SET CONSOLE CN
USE
CASE ! (ACCOPT) = "A"
STORE T TO PCRE
DC WHILE MORE
ERASE
STORE " " TO ACNAME
STORE " " TO ACCUSER
STORE " " TO ACCPASS
STORE 0 TO ACCLEV
@ 10,25 SAY " NAME OF USER"
@ 10,38 GET ACNAME PICTURE "XXXXXXXXXXXXXXXX"
@ 12,25 SAY "USER ID"
@ 12,33 GET ACCUSER PICTURE "XXXXX"
@ 14,25 SAY "PASSWORD"
@ 14,44 GET ACCESS PICTURE "XXXXX"
@ 16,25 SAY "AUTHORIZATION LEVEL"
@ 16,45 GET ACCLEV PICTURE "9"
READ
STORE T TC NOGOOD
DO WHILE NOGOOD
   IF (ACCLEV < 1) .OR. (ACCLEV > 3)
       @ 23,31 SAY "ILLEGAL AUTHORIZATION LEVEL;"
   @ 16,45 GET ACCLEV PICTURE "9"
   REAL
   ELSE
   STORE F TC NOGOOD
   ENDDP
ENDDO
USE A:SECURITY INDEX A:ACCNAME, A:SECINDEX
APPEND BLANK
REPLACE NAME WITH '(ACNAME) ; ';
AUTHLEV WITH ACCLEV
REPLACE USERID WITH '(ACCUSER) ; ';
USERPASS WITH '(ACCPASS)
USE
STORE P TC DONE
DO WHILE .NOT. DONE
   ERASE
   @ 10,25 SAY " DO YOU WANT TO ENTER ANOTHER;"
   ACCEPT " ENTER (Y OR N ) ===> " TO CONT
   IF ! (CONT) = "Y"
      STORE T TC DONE
   ELSE
      IF ! (CONT) = "N"
         STORE F TC MORE
      ENDDP
   ENDDO
CASE !(ACCOPT) = "D"
STORE T TO MORE
DELETE
   @ 12,24 SAY " ENTER NAME OF THE USER TO BE;"
   ACCEPT " OR QUIT TO REENTER ACCESS MENU " TO ACNAME
   STORE ?(ACNAME) TO ACNAME
   IF ACNAME = "QUIT"
      STORE F TO MORE
   ELSE
      USE A:SECURITY INDEX A:ACCNAME, A:SECINDEX
      FIND &ACNAME
      IF # = 0
         @ 22,28 SAY "INDIVIDUAL NOT FOUND IN:
      FILE"
      CONTINUE ***
      ELSE
      USE
      ENDIF
   USE A:SECURITY INDEX A:ACCNAME, A:SECINDEX
   230
PACK
USE
ENDDO
CASE 1(ACCEPT) = "Q"
   STORE T TO THRU
ENDCASE
ENDDO
RELEASE ALL EXCEPT AA??????
RETURN
BIBLIOGRAPHY


# INITIAL DISTRIBUTION LIST

<table>
<thead>
<tr>
<th>No.</th>
<th>Distribution List</th>
<th>Copies</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Defense Technical Information Center</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>Cameron Station</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Alexandria, Virginia 22134</td>
<td></td>
</tr>
<tr>
<td>2.</td>
<td>Library, Code 0142</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>Naval Postgraduate School</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Monterey, California 93943</td>
<td></td>
</tr>
<tr>
<td>3.</td>
<td>Naval Postgraduate School</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Computer Technologies Curriculum Office</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Code 37</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Monterey, California 93943</td>
<td></td>
</tr>
<tr>
<td>4.</td>
<td>Associate Professor Norman Lyons, Code 54LE</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Department of Administrative Sciences</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Naval Postgraduate School</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Monterey, California 93943</td>
<td></td>
</tr>
<tr>
<td>5.</td>
<td>Lieutenant Colonel Joseph P. Mullane, USMC</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Code 0309</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Marine Corps Representative</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Naval Postgraduate School</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Monterey, California 93943</td>
<td></td>
</tr>
<tr>
<td>6.</td>
<td>Commandant of the Marine Corps (Code CC)</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>Headquarters Marine Corps</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Washington, D.C. 20380</td>
<td></td>
</tr>
<tr>
<td>7.</td>
<td>Lieutenant Colonel Ronald E. Pruett, USMC</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Marine Corps Tactical Systems Support Activity</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Camp Pendleton, California 92055</td>
<td></td>
</tr>
<tr>
<td>8.</td>
<td>Captain David P. Haessler, USMC</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>12112 Ft. Craig Dr.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Waccabridge, Virginia 22192</td>
<td></td>
</tr>
</tbody>
</table>

233