THESIS

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

by

Ronald E. Pruett
David P. Haeusler

March 1984

Thesis Advisor: Norman R. Lyons

Approved for public release; distribution unlimited

84 12 11 001
An Automated Individual Training Record Management System (Prototype), United States Marine Corps

Ronald E. Pruiett
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, Base II

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.
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

Seval R. Balk  
Second Reader

Richard P. Jelinek  
Chairman, Department of Administrative Sciences

Karl T. Mundahl  
Dean of Information and Policy Sciences
ABSTRACT

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 ........................................... 11
   A. GENERAL DESCRIPTION .................................... 11
   E. BACKGROUND ........................................... 11
   C. OBJECTIVES ........................................... 12
   D. ENVIRONMENT .......................................... 13
   E. SYSTEM CHARACTERISTICS ................................ 13
      1. User friendly ....................................... 13
      2. Reliability ......................................... 13
      3. Cost .............................................. 14
      4. Expandability of data base .......................... 14
      5. Expandability of functions .......................... 14
      6. Speed of operation .................................. 14
   F. SUMMARY .............................................. 14
      1. Scope of Development ............................... 14
      2. Concept ........................................... 15

II. SYSTEM DESCRIPTION AND REQUIREMENTS ................... 16
   A. INPUT INFORMATION ..................................... 16
   E. SYSTEM SPECIFICATIONS (FUNCTIONAL) .................. 16
      1. Training Data ....................................... 16
      2. Data Base Maintenance .............................. 17
      3. Report Formats ..................................... 17
      4. Data Retrieval ...................................... 17
      5. Standard Reports .................................... 18
      6. Decision Support Subsystem .......................... 20
      7. Nonstandard Reports ................................ 20
      8. System Controls .................................... 20
      9. Output Mode ........................................ 21
   C. REQUIRED SPECIFICATIONS (NON-FUNCTIONAL) ........... 21
I. DATA STRUCTURE DEFINITIONS (DATA DICTIONARY) ........................................... 22
   1. Introduction .................................... 22
   2. Data Base Management System .................. 22

III. METHODOLOGY AND DESIGN .............................................................. 24
   A. METHODOLOGY ........................................ 24
      1. Software ........................................ 24
   B. SOFTWARE DESIGN ................................... 26
      1. Structured Programming ....................... 26
      2. Top-down Design ................................ 26
   C. DATA FLOW DIAGRAMS ................................ 27
   D. SYSTEM HIERARCHY .................................. 32
   E. MODULE DESCRIPTIONS ............................... 33
      1. ITRMS Master Control Module ................. 33
      2. Access Module .................................. 35
      3. Command Determiner Module .................... 36
      4. Nonstandard Report Generator Module .......... 37
      5. Standard Report Generator Module ............. 38
      6. Maintenance Module ............................. 39

IV. IMPLEMENTATION ................................................................. 40
   A. CONSTRAINTS ......................................... 40
      1. dBASE II ......................................... 40
      2. Hardware ......................................... 40
      3. Environmental .................................... 41
   B. GENERAL ............................................... 42
      1. ITRMS Master Control Module ................. 42
      2. Access Module .................................. 43
      3. Command Determiner Module .................... 43
      4. Nonstandard Report Generator Module .......... 44
      5. Standard Report Generator Module ............. 46
      6. Maintenance Module ............................. 56
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: TRS 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

7
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

APENDIX 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
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.

B. 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 (MCS) 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 data base 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 data base 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.

II. 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
1. 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 slicker 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.

7. SUMMARY

a. Scope of Development

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.

E. 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. Rank

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 defined set of attributes (e.g., all sgt's 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 cleanup 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, PFT2, 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 database. After a name is entered, the complete individual training record for that individual is displayed. This includes all training data included in the database 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 database.

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 database 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 non-standard 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.
I. 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 users 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 data base 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
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
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 them 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
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 3.4 Data Flow Expansion for Access Function
Figure 3.5 Data Flow Expansion for Command Generators

I. SYSTEM HIERARCHY

The ITBM 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 ITRMS 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. ITRMS Master Control Module
   a. Module Purpose

   The ITRMS 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.

d. 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 data base 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 confired 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 ESS 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 database 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 database 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 database 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

A. 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 EC 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/cut 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 Quit. 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
USMC INDIVIDUAL TRAINING RECORD MANAGEMENT SYSTEM

MAIN MENU

SELECT ONE OF THE FOLLOWING OPTIONS BY ENTERING THE APPROPRIATE LETTER

D...DATA ENTRY AND SYSTEM FUNCTIONS
S...STANDARD REPORTS
N...NONSTANDARD REPORTS
H...HELP
Q...QUIT

SELECT OPTION ===>;

Figure 4.1 Command Determiner Menu

appropriate module and is not designed to explain the functioning of the entire system. The systems 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 E5 (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>ATTRIBUTE</th>
<th>DESCRIPTOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMPANY IS</td>
<td>A CO</td>
</tr>
<tr>
<td>RANK IS</td>
<td>E5</td>
</tr>
<tr>
<td>PLATOON IS</td>
<td>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 (Slctarti.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 (Slctdesc.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 (Stoerpt.prg)

This routine generates the primary menu for standard report selection and stores the user's response in a memory variable. Con.rcl 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...HELP

G...QUIT

SELECT OPTION ===>:

Figure 4.3 Standard Report Menu

t. Select Unit Routine (Slctunit.prg)

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.prg)

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

d. Training Status Routine (Trngstat.prg)

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

"HISTCRI" 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, ZZBHIS, 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 (Cnttest.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 (Trngrost.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 Trngrpt.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, Trngrost.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 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 Boster Format Routine (Trngrpt.prg)

This defines a coded report format which displays COC, HIS, EKS, 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 'Sltctunit'. 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 Sltctunit.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

<table>
<thead>
<tr>
<th>Rifle</th>
<th>48</th>
<th>Pistol</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expert</td>
<td>48</td>
<td>Expert</td>
<td>3</td>
</tr>
<tr>
<td>Sharpshooter</td>
<td>36</td>
<td>Sharpshooter</td>
<td>0</td>
</tr>
<tr>
<td>Marksmen</td>
<td>48</td>
<td>Marksmen</td>
<td>2</td>
</tr>
<tr>
<td>Unqualified</td>
<td>16</td>
<td>Unqualified</td>
<td>0</td>
</tr>
</tbody>
</table>

### PERCENTAGES

<table>
<thead>
<tr>
<th>Rifle</th>
<th>32</th>
<th>Pistol</th>
<th>60</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expert</td>
<td>32</td>
<td>Expert</td>
<td>60</td>
</tr>
<tr>
<td>Sharpshooter</td>
<td>24</td>
<td>Sharpshooter</td>
<td>0</td>
</tr>
<tr>
<td>Marksmen</td>
<td>32</td>
<td>Marksmen</td>
<td>40</td>
</tr>
<tr>
<td>Unqualified</td>
<td>11</td>
<td>Unqualified</td>
<td>0</td>
</tr>
</tbody>
</table>

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 (Quotapad.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.

m. Marksmanship Roster Routine (Mksrcst.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:Mrksrpt) is used to display the Marksmanship Foster Report (see Figure 4.7).

n. 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).

c. MOS Roster Routine (Mosprrt.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 DATE</th>
<th>QUAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>EAKER, SIEVE N.</td>
<td>E4</td>
<td>100184</td>
<td>EX</td>
</tr>
<tr>
<td>CARY, R.V.</td>
<td>E5</td>
<td>050184</td>
<td>MM</td>
</tr>
<tr>
<td>CASTIN, D.P.</td>
<td>E7</td>
<td>050184</td>
<td>UN</td>
</tr>
<tr>
<td>LCNELY, R.W.</td>
<td>E6</td>
<td>DDMYY</td>
<td>**</td>
</tr>
</tbody>
</table>

PRESS 'ENTER' TO CONTINUE

Figure 4.7 Marksmanship Roster Report

MOS STATUS 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

P. 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 Selectunit Menu.

Q. Personnel Data Format Routine (Mosrpt.frm)

This defines a dBASE II report format which has already been determined and is stored in memory.

53
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.dbf 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

- 42 PERCENT COMPLETED... 76
- 84 PERCENT COMPLETED... 151
- 39 PERCENT COMPLETED... 151
- 63 PERCENT COMPLETED... 113
- 8 PERCENT COMPLETED... 14

PRESS 'INTER' TO CONTINUE

Figure 4.10 EST Status Report

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.

v. 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  Swimming Qualification Status Report

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 (Viewitr.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

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).
## Individual Training Record

<table>
<thead>
<tr>
<th>Name: Last, First MI.</th>
<th>Grade:</th>
<th>MOS:</th>
<th>Date: 1/1/1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Birthdate: DD MM YY</td>
<td>SSN:</td>
<td>Joint: DD MM YY</td>
<td>Gas Mask:</td>
</tr>
<tr>
<td>Code:</td>
<td>Join: DD MM YY</td>
<td>Casual: DD MM YY</td>
<td>Date: DD MM YY</td>
</tr>
<tr>
<td>MIL Code of Conduct:</td>
<td>ITT:</td>
<td>Date: DD MM YY</td>
<td></td>
</tr>
<tr>
<td>History of Narcotics:</td>
<td>ITT:</td>
<td>Date: DD MM YY</td>
<td></td>
</tr>
<tr>
<td>Drill:</td>
<td>NBC:</td>
<td>Date: DD MM YY</td>
<td></td>
</tr>
<tr>
<td>Interior Guard:</td>
<td>Regular Service:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>First Aid:</td>
<td>Individual Tactical</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sport Uniform:</td>
<td>Measures:</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| Swim Qual: | Date: DD MM YY | Weight Control: |
| Kill Qual: | Date: DD MM YY | Height: |
| Pistol Qual: | Date: DD MM YY | Height: |

| Electronic Warfare: | Drug Abuse: |
| Old Weather: | Alcohol Abuse: |
| Law of War: | Human Relations: |
| Training Mos.: | Personal Affairs: |
| Unit Sport: | UCMJ: |
| Char & Moral Ed.: | |
| Counts: | |
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: Pers.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 (Updata.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 data base.

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 data base 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 (Wrtcomnt.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 (Crtcomnt.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...INSTALLED 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

62
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 capa-
bility also exists in the Standard Report Generators DSS.

r. System Access Control Routine (Rstaccess.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

1...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.

II. 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.
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 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.

E. NAME

Each data field has a unique name which always refers to that specific information that is contained in the data field (example: PRIMEOS 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. PERS (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</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. E0311)</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, HHS 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</td>
</tr>
<tr>
<td>07</td>
<td>Flatcon</td>
<td>C</td>
<td>09</td>
<td>PLAT, HQTRS</td>
</tr>
<tr>
<td>08</td>
<td>JoinDate</td>
<td>C</td>
<td>06</td>
<td>The date (DDMMYY) member joined the unit.</td>
</tr>
<tr>
<td>09</td>
<td>EAS</td>
<td>C</td>
<td>06</td>
<td>Expiration of Active Service (DDMMYY)</td>
</tr>
<tr>
<td>10</td>
<td>ExtDate</td>
<td>C</td>
<td>06</td>
<td>Date of birth (DDMMYY)</td>
</tr>
<tr>
<td>11</td>
<td>Comment</td>
<td>L</td>
<td>01</td>
<td>A boolean variable (F/T) for which T indicates that a comment record exists.</td>
</tr>
<tr>
<td>12</td>
<td>Height</td>
<td>C</td>
<td>02</td>
<td>Individual's height in inches</td>
</tr>
<tr>
<td>13</td>
<td>Weight</td>
<td>C</td>
<td>03</td>
<td>Individual's weight in pounds</td>
</tr>
<tr>
<td>14</td>
<td>Tncnt</td>
<td>L</td>
<td>01</td>
<td>A boolean variable (F/T) for which T indicates that the individual is on weight control.</td>
</tr>
<tr>
<td>15</td>
<td>Gasmask</td>
<td>C</td>
<td>01</td>
<td>Gas mask size (S, M, L)</td>
</tr>
</tbody>
</table>
2. **IST (Essential Subjects Testing)**

<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>Code of Conduct contains an entry for each training element (A,B,C).</td>
</tr>
<tr>
<td>02</td>
<td>COC</td>
<td>C</td>
<td>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</td>
<td>03</td>
<td>Close Order Drill contains an entry for each training element (A,B,C).</td>
</tr>
<tr>
<td>04</td>
<td>CD</td>
<td>C</td>
<td>03</td>
<td>Interior Guard contains an entry for each training element (A,B,C).</td>
</tr>
<tr>
<td>05</td>
<td>TMT</td>
<td>C</td>
<td>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</td>
<td>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</td>
<td>03</td>
<td>Close Order Drill contains an entry for each training element (A,B,C).</td>
</tr>
<tr>
<td>08</td>
<td>FPR1RAW</td>
<td>N</td>
<td>03</td>
<td>Includes the numeric score for the first annual physical fitness test (RANGE: 00-300).</td>
</tr>
<tr>
<td>09</td>
<td>FPR1DATE</td>
<td>C</td>
<td>06</td>
<td>Date of first physical fitness test (DDMMYY).</td>
</tr>
<tr>
<td>10</td>
<td>FPR1CLSS</td>
<td>C</td>
<td>01</td>
<td>Results of first physical fitness test (1,2,3,4).</td>
</tr>
<tr>
<td>11</td>
<td>FPR2RAW</td>
<td>N</td>
<td>03</td>
<td>Includes the numeric score for the second annual physical fitness test (RANGE: 00-300).</td>
</tr>
<tr>
<td>12</td>
<td>FPR2DATE</td>
<td>C</td>
<td>06</td>
<td>Date of the second physical fitness test.</td>
</tr>
<tr>
<td>13</td>
<td>FPR2CLSS</td>
<td>C</td>
<td>01</td>
<td>Results of second physical fitness test (1,2,3,4).</td>
</tr>
<tr>
<td>14</td>
<td>NBC</td>
<td>C</td>
<td>06</td>
<td>NBC contains an entry for each training element (A,B,C).</td>
</tr>
<tr>
<td>15</td>
<td>MKS</td>
<td>C</td>
<td>06</td>
<td>Marksmanship contains an entry for each training element (A,B,C).</td>
</tr>
<tr>
<td>16</td>
<td>TAC</td>
<td>C</td>
<td>05</td>
<td>Individual Tactical Measures contains an entry for each training element (A,B,C).</td>
</tr>
</tbody>
</table>

3. **QUAL (Qualifications)**

<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>SWMQUAL</td>
<td>C</td>
<td>02</td>
<td>Swimming Qualification (WQ, SW, S2, SM).</td>
</tr>
<tr>
<td>03</td>
<td>SWMDATE</td>
<td>C</td>
<td>06</td>
<td>SWMQUAL date (DDMMYY).</td>
</tr>
<tr>
<td>04</td>
<td>FPLSCORE</td>
<td>C</td>
<td>03</td>
<td>Numeric rifle score (RANGE: 000-250).</td>
</tr>
<tr>
<td>05</td>
<td>FPLQUAL</td>
<td>C</td>
<td>02</td>
<td>Rifle Qualification (EX,SS, MM, UN).</td>
</tr>
<tr>
<td>06</td>
<td>RPLDATE</td>
<td>C</td>
<td>06</td>
<td>Date of RPLQUAL (DDMMYY).</td>
</tr>
<tr>
<td>07</td>
<td>RPLSCORE</td>
<td>C</td>
<td>03</td>
<td>Numeric pistol score (RANGE: 000-300).</td>
</tr>
<tr>
<td>08</td>
<td>PSTQUAL</td>
<td>C</td>
<td>02</td>
<td>Pistol Qualification (EX,SS, MM, UN).</td>
</tr>
<tr>
<td>09</td>
<td>PSTDATE</td>
<td>C</td>
<td>06</td>
<td>Date of PSTQUAL (DDMMYY).</td>
</tr>
</tbody>
</table>
### 4. INFO_MISC (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>FLECWAR</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>LAWAR</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>HUMBREL</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>CHARMORED</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>USEPSS</td>
<td>C</td>
<td>08</td>
<td>Alphanumeric code that uniquely identifies each user.</td>
</tr>
<tr>
<td>03</td>
<td>AUTELEV</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>INFCTX1</td>
<td>C</td>
<td>25</td>
<td>Unstructured comments.</td>
</tr>
<tr>
<td>03</td>
<td>INFCTX2</td>
<td>C</td>
<td>25</td>
<td>Unstructured comments.</td>
</tr>
<tr>
<td>04</td>
<td>INFCTX3</td>
<td>C</td>
<td>25</td>
<td>Unstructured comments.</td>
</tr>
<tr>
<td>05</td>
<td>INFCTX4</td>
<td>C</td>
<td>25</td>
<td>Unstructured comments.</td>
</tr>
<tr>
<td>06</td>
<td>INFCTX5</td>
<td>C</td>
<td>25</td>
<td>Unstructured comments.</td>
</tr>
<tr>
<td>07</td>
<td>INFCTX6</td>
<td>C</td>
<td>25</td>
<td>Unstructured comments.</td>
</tr>
<tr>
<td></td>
<td>INFCTXT7 C</td>
<td>25</td>
<td>Unstructured comments.</td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>------------</td>
<td>----</td>
<td>------------------------</td>
<td></td>
</tr>
<tr>
<td>08</td>
<td>INFCTXT8 C</td>
<td>25</td>
<td>Unstructured comments.</td>
<td></td>
</tr>
<tr>
<td>09</td>
<td>INFCTXT9 C</td>
<td>25</td>
<td>Unstructured comments.</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>INFCTXTO C</td>
<td>25</td>
<td>Unstructured comments.</td>
<td></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 manhours, 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 persons 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

Database 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).

<table>
<thead>
<tr>
<th>VARIABLE NAME</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>QJAN</td>
<td>25</td>
</tr>
<tr>
<td>QFEB</td>
<td>25</td>
</tr>
<tr>
<td>QMAR</td>
<td>25</td>
</tr>
<tr>
<td>QAPR</td>
<td>25</td>
</tr>
<tr>
<td>QMAY</td>
<td>50</td>
</tr>
<tr>
<td>QJUL</td>
<td>50</td>
</tr>
<tr>
<td>QAGS</td>
<td>50</td>
</tr>
<tr>
<td>QSEP</td>
<td>00</td>
</tr>
<tr>
<td>QOCT</td>
<td>50</td>
</tr>
<tr>
<td>QNOV</td>
<td>00</td>
</tr>
<tr>
<td>QDEC</td>
<td>00</td>
</tr>
</tbody>
</table>

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
* Files (opened/closed): 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
  housekeeping 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.

    SET CCNCSLE ON
    SET TALK OFF
    SET ESCAPE OFF

    STORE T TO ABENDLES
    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 CCNCSLE OFF
      WAIT
      SET CCNCSLE ON
      RESTORE FROM A:MEMDISK ADDITIVE

    ENDCO

    DO WHILE ABENDLES
      ERASE
      @ 10 27 SAY "UNITED STATES MARINE CORPS"
      @ 12 34 SAY "(Prototype)"
      LO PAUSE
      @ 23 23 SAY "***** PRESS ANY KEY TO CONTINUE *****"
      SET CCNCSLE OFF
      WAIT
      SET CCNCSLF ON
      DO ACCESS
      RESTORE FROM A:MEMDISK ADDITIVE
      STORE F TO SYSDATE
      DO WHILE .NOT. SYSDATE .AND. AAVLPAS
      ERASE
@ 11,24 SAY "ENTER TODAY'S DATE"
ACCEPT "DATE (DDMMY) ===>
IF VAL($SYSDAY, 1, 2) < 32 AND. VAL($SYSDAY, 3, 2) < 13 AND. VAL($SYSDAY, 5, 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 AA? ?????
DO Cmddet
ENDIF
STORE T TO AAENDLESS
ENDDC
E. PAUSE ROUTINE

* Routine Name: Pause.prg
* Module Name: IRS Master Control Module
* Version: 1.0.0.0.1.0
* Author: D.E. Haeusler
* 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 ROUTINE LISTING

A. ACCESS ROUTINE

* Routine Name: Access.prg
* Module Name: Access Module
* Version: 2.0
* Author: L.P. Haeusler
* Date: 26 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,
  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
  STORE !(idnum) TO idnum
  IF idnum = 'QUIT'
    STORE t TO thru
  ELSE
    ERASE
    USE a:SECURITY INDEX a:SECINDEX
    FIXL &idnum
  IF # = 0
    USE
  ELSE
    STORE TRIM (USERPASS) TO passrec
    STORE AUTHLEV TO aalevel
    STORE t TO validid
  ENDIF
  ENDF
ENDIF
IF .NC. Valid and .NC. 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 .NC. Valid and .NC. Thru .AND. COUNTER = 2
STORE COUNTER+1 TO COUNTER
ENDIF
ENDIF
IF COUNTER = 3
STORE t TO thru
ERASE
@ 12,20 SAY "******************** TERMINATE ACCESS ******************"
@ 14,20 SAY "****************************
ENDIF
ENDDO
STORE 0 TO COUNTER
DO WHILE .NC. AAvalpas .AND. .NC. Thru
ERASE
@ 7,26 SAY "ENTER YOUR PASSWORD OR"
@ 7,26 SAY "QUIT TO TERMINATE THE SESSION"
@ 11,18 SAY "THE PASSWORD WILL NOT BE DISPLAYED"
@ 14,25 SAY "ENTER PASSWORD ===>"
SET CONSOLE OFF
ACCEPT "ENTER PASSWORD ===>" TO password
SET CONSOLE ON
IF !(password) = 'QUIT'
STORE t TO thru
ELSE
IF !(password) = passrec
STORE t TO AAvalpas
ELSE
IF 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
ENDIF
ENDIF
ENDIF
ENDIF
ENDCC
RELEASE COUNTER, pause, 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. Hasusler
* 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
ccontrol 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 "H...HELP"
   @ 13,17 SAY "Q...QUIT"
   ACCEPT "SELECT OPTION ===>" TO cmdopt
   DC CASE
   CASE 1(cmdopt) = 'D'
      DO maratype
         STORE f TO Aafinish
      ENDCASE
   CASE 2(cmdopt) = 'S'
      DO stdrept
         STORE f TO Aafinish
      ENDCASE
   CASE 3(cmdopt) = 'N'
      DO nstdrept
         STORE f TO Aafinish
      ENDCASE
   CASE 4(cmdopt) = 'H'
      STORE t TO HELPPT
      DO HELP2
         STORE f TO Aafinish
      ENDCASE
   CASE 5(cmdopt) = 'Q'
      STORE t TO Aafinish
   ENDCASE
   ENDCASE
RELEASE cmdopt, HELPPT, 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, Cmmdet.prg

Description: This routine gives a brief description of
the commands that can be invoked by the user.

ERASE
TO CASE
CASE HELROPT = 1
TEXT

D...DATA ENTRY ALL 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 press the shift key and the
print 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 infor- mation to make a query. Paper
Copies of the reports can obtained by the same method as in
Standard Reports.

END TEXT
@ 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 individuals name. You only
need to enter enough of the name to be able to distinguish
it from another Marines name.

C...CREATE AN ITR: This module allcs the creation of
ew 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 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 "**** PRESS 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: NOSTEPFT.PRG
* MODULE NAME: NONSTANDARD COMMAND GENERATOR
* VERSION: 4.0
* AUTHOR: R. L. PRUITT
* DATE: 04/04/84
* VARIABLES USED: MTRI1, DESC1, MTRI2, DESC2, MTRI3,
  DESC3, SLCT1, SLCT2, SLCT3
* VARIABLES MODIFIED: MREPEAT, COUNT, PROCEED
* VARIABLES CREATED: MREPEAT, COUNT, PROCEED
* VARIABLES RELEASED: M?????, REPEAT, COUNT,
  ATTTI????, SLCT????, DESC????
* FILES CRENED/CLOSED: A:PEFST, B:QUAL
* FILES CREATED: TEMP, SLECTDATA, TEMP2, TEMP3
* USING SUBROUTINES: COMMAND IDENTIFIER MODULE
* DESCRIPTION: THIS MODULE SEARCHES THE APPROPRIATE
  DATABASE FILES IN CEDER TC LOCATE AND DISPLAY ALL RECORDS
  THAT SATISFY A SET CF FROM ONE TO THREE USER SPECIFIED
  ATTRIBUTES.

STORE T TO MREPEAT
DO WHILE MREPEAT = T
  STORE P TO MREPEAT
  ERASE
  @ 10, 25 SAY "NONSTANDARD COMMAND GENERATOR"
  @ 20, 0 SAY "PRESS 'H' FOR HELP"
  @ 21, 0 SAY "CH"
  ACCEPT "PRESS 'ENTER' TO CONTINUE" TO PROCEED
  IF I (PROCEED) = "H"
  DO A:NCSTHLP
  STORE T TO MREPEAT
  ENDDC
  ERASE
  @ 1, 25 SAY "NONSTANDARD COMMAND GENERATOR"
  @ 3, 2 SAY "WHEN PROTOVED ENTER AN ATTRIBUTE AND A'":
  "DESCRIPTION FOR EACH"
  @ 4, 5 SAY "CONDITION STATEMENT DESIRED."
  @ 6, 10 SAY "LOCATE ALL MARRAY THAT SATISFY THESE:
  CONDITIONS:" IS "<ATTRIBUTE> <DESCRIPTION>
  @ 7, 10 SAY "<ATTRIBUTE> <DESCRIPTION>
  @ 8, 10 SAY "<ATTRIBUTE> <DESCRIPTION>
  ACCEPT "PRESS 'ENTER' TO CONTINUE" TO PROCEED

STORE T TO MREPEAT
STORE 0 TO COUNT
DO WHILE MREPEAT = T AND COUNT <= 3
  STORE COUNT + 1 TO COUNT
  STORE COUNT TO MREPEAT
  DO SLECTATRI
  ERASE
  DO CASE
    CASE COUNT = 1
    @ 1, 0 SAY "<ATTRIBUTE> "
    @ 2, 0 SAY "<DESCRIPTION>"
    @ 4, 6 SAY MTRI1------

...
CASE COUNT = 2
CASE (SLCT1 >= "A" .AND. SLCT1 <= "I")
USE A:PERS
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 E:EST
CCPY TO A:TEMP FIELD SSN, &ATRI1 FOR &ATRI1 =;
"DESC1"
SELECT PRIMARY
USE A:TEMP
SELECT SECONDARY
USE A:PERS
JCIN TO SLCTDATA FOR P.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 EQ:UAL
CCPY TO A:TEMP FIELD SSN, &ATRI1 FOR &ATRI1 =;
"DESC1"
SELECT PRIMARY
USE A:TEMP
SELECT SECONDARY
USE A:PERS
JOIN TO SELECTDATA FOR F.SSN=S.SSN FIELDS;
NAME S.SN, &ATRI1
USE SELECTDATA
DISPLAY ALL
IF EOP
@ 23,50 SAY "SEARCH COMPLETE"
ENDIF
ENCASE
ACCEPT "PRESS 'ENTER' TO CONTINUE" TO PROCEED
ENDIF

IF CCOUNT = 2
DO CASE
CASE (SLCT1 >= "A", AND, SLCT1 <= "I") .AND.:
{SLCT2 >= "A", AND, SLCT2 <= "I"
USE A:PERSONS
COPY TO A:TEMP FIELD NAME, S.SN, &ATRI1, &ATRI2 FOR;
&ATRI1="DESC1"
USE A:TEMP
DISPLAY FOR &ATRI2="DESC2" OFF FIELDS;
NAME S.SN, &ATRI1, &ATRI2
IF EOP
@ 23,50 SAY "SEARCH COMPLETE"
ENDIF
CASE (SLCT1 >= "A", AND, SLCT1 <= "I") .AND.:
{SLCT2 = "J", OR, SLCT2 = "K"
USE B:EST
COPY TO A:TEMP FIELD S.SN, &ATRI2 FOR &ATRI2 =;
"DESC2"
SELECT PRIMARY
USE A:TEMP
SELECT SECONDARY
USE A:PERSONS
JOIN TO SELECTDATA FOR F.SSN=S.SSN FIELDS;
NAME S.SN, &ATRI1, &ATRI2
USE SELECTDATA
DISPLAY FOR &ATRI1 = "DESC1" OFF FIELDS;
NAME S.SN, &ATRI1, &ATRI2
IF EOP
@ 23,50 SAY "SEARCH COMPLETE"
ENDIF
CASE (SLCT1 = "J", OR, SLCT1 = "K") , AND.:
{SLCT2 >= "A", AND, SLCT2 <= "I"
USE A:PERSONS
COPY TO A:TEMP FIELD NAME, S.SN, &ATRI2 FOR &ATRI2 =;
"DESC2"
SELECT PRIMARY
USE A:TEMP
SELECT SECONDARY
USE A:PERSONS
JOIN TO SELECTDATA FOR F.SSN=S.SSN FIELDS;
NAME S.SN, &ATRI1, &ATRI2
USE SELECTDATA
DISPLAY FOR &ATRI1 = "DESC1" OFF FIELDS;
NAME S.SN, &ATRI1, &ATRI2
IF EOP
@ 23,50 SAY "SEARCH COMPLETE"
ENDIF
CASE (SLCT1 = "J", OR, SLCT1 = "K") , AND.:
{SLCT2 >= "A", AND, SLCT2 <= "I"
USE A:PERSONS
COPY TO A:TEMP FIELD NAME, S.SN, &ATRI2 FOR &ATRI2 =;
"DESC2"
SELECT PRIMARY
USE A:TEMP
SELECT SECONDARY
USE A:PERSONS
JOIN TO SELECTDATA FOR F.SSN=S.SSN FIELDS;
NAME S.SN, &ATRI1, &ATRI2
USE SELECTDATA
DISPLAY FOR &ATRI1 = "DESC1" OFF FIELDS;
NAME S.SN, &ATRI1, &ATRI2
IF EOP
@ 23,50 SAY "SEARCH COMPLETE"
ENDIF
CASE (SLCT1 = "J", OR, SLCT1 = "K") , AND.:
{SLCT2 >= "A", AND, SLCT2 <= "I"
USE A:PERSONS
COPY TO A:TEMP FIELD NAME, S.SN, &ATRI2 FOR &ATRI2 =;
"DESC2"
SELECT PRIMARY
USE A:TEMP
SELECT SECONDARY
USE A:PERSONS
JOIN TO SELECTDATA FOR F.SSN=S.SSN FIELDS;
NAME S.SN, &ATRI1, &ATRI2
USE SELECTDATA
DISPLAY FOR &ATRI1 = "DESC1" OFF FIELDS;
NAME S.SN, &ATRI1, &ATRI2
IF EOP
@ 23,50 SAY "SEARCH COMPLETE"
ENDIF

88
JOIN TO SLCTDATA FOR F.SSN=S.SSN FIELDS;
NAME, S.SN, EATRI1, EATRI2
USE SLCTDATA
DISPLAY FOR EATRI1 = "EDESC1" OFF FIELDS;
NAME, S.SN, EATRI1, EATRI2
IF EOF
  @ 23, 50 SAY "SEARCH COMPLETE"
ENDIF
CASE (SLCT1 = "J" .OR. SLCT1 = "K") .AND.;
  (SLCT2 = "J" .OR. SLCT2 = "K")
  USE A:TEMP
  COPY TO A:TEMP FIELD S.SN, EATRI1, EATRI2 FOR EATRI2 =:
    "EDESC2" .AND. EATRI1 = "EDESC1"
  SELECT PRIMARY
  USE A:TEMP
  SELECT SECONDARY
  USE A:TEMP
  JCIN TO SLCTDATA FOR F.SSN=S.SSN FIELDS;
  NAME, S.SN, EATRI1, EATRI2
  USE SLCTDATA
  DISPLAY ALL
  IF EOF
    @ 23, 50 SAY "SEARCH COMPLETE"
  ENDIF
ENDIF
CASE (SLCT1 = "J" .OR. SLCT1 = "K") .AND.;
  (SLCT2 = "J" .OR. SLCT2 = "K")
  USE A:QUAL
  COPY TO A:TEMP FIELD S.SN, EATRI1, EATRI2 FOR EATRI2 =:
    "EDESC2"
  USE B:EST
  COPY TO A:TEMP FIELD S.SN, EATRI1, EATRI2 FOR EATRI1 =;
    "EDESC1"
  SELECT PRIMARY
  USE A:TEMP
  SELECT SECONDARY
  USE A:TEMP
  JCIN TO A:TEMP3 FOR F.SSN=S.SSN FIELDS;
  S.SN, EATRI1, EATRI2
  SELECT PRIMARY
  USE A:TEMP3
  SELECT SECONDARY
  USE A:TEMP
  JCIN TO SLCTDATA FOR F.SSN=S.SSN FIELDS;
  NAME, S.SN, EATRI1, EATRI2
  USE SLCTDATA
  DISPLAY ALL
  IF EOF
    @ 23, 50 SAY "SEARCH COMPLETE"
  ENDIF
ENDIF
CASE (SLCT1 = "J" .OR. SLCT1 = "K") .AND.;
  (SLCT2 = "J" .OR. SLCT2 = "K")
  USE A:TEMP
  COPY TO A:TEMP FIELD NAME, S.SN, EATRI1, EATRI2 FOR EATRI2 =:
    "EDESC2"
  SELECT PRIMARY
  USE A:TEMP
  SELECT SECONDARY
  USE A:TEMP
  JCIN TO SLCTDATA FOR F.SSN=S.SSN FIELDS;
  NAME, S.SN, EATRI1, EATRI2
  USE SLCTDATA
  DISPLAY FOR EATRI1 = "EDESC1" OFF FIELDS;
  NAME, S.SN, EATRI1, EATRI2
  IF EOF
    @ 23, 50 SAY "SEARCH COMPLETE"
  ENDIF
CASE (SLCT1 = "J" .OR. SLCT1 = "K") .AND.;
  (SLCT2 = "J" .OR. SLCT2 = "K")
  USE B:EST

89
COPY TO A:TEMP FIELD SSN,EATRI2 FOR EATRI2 = "DESC2"
USE B:QUAL
COPY TO A:TEMP2 FIELD SSN,EATRI1 FOR EATRI1 = "DESC1"
SELECT PRIMARY
USE A:TEMP
SELECT SECONDARY
USE A:TEMP2
JOIN TO A:TEMP3 FOR F.SSN=S.SSN FIELDS:
SSN,EATRI1,EATRI2
SELECT PRIMARY
USE A:TEMP3
SELECT SECONDARY
USE A:PERSON
JOIN TO SLCTDATA FOR F.SSN=S.SSN FIELDS;
NAME,SSN,EATRI1,EATRI2
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,EATRI1,EATRI2 FOR EATRI1 = "DESC1" .AND. EATRI2 = "DESC2"
SELECT PRIMARY
USE A:TEMP
SELECT SECONDARY
USE A:PERSON
JOIN TO SLCTDATA FOR F.SSN=S.SSN FIELDS;
NAME,SSN,EATRI1,EATRI2
USE SLCTDATA
DISPLAY ALL
IF EOF
@ 23,50 SAY "SEARCH COMPLETE"
ENDIF
ENCASE
ACCEPT "PRESS 'ENTER' TO CONTINUE" TO PROCEED
ENDIF
IF C8INT = 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,EATRI1,EATRI2,EATRI3 FOR EATRI1="DESC1" EATRI2="DESC2" EATRI3="DESC3"
USE A:TEMP
COPY TO A:TEMP2 FIELD NAME,SSN,EATRI1,EATRI2,EATRI3 FOR EATRI1="DESC1" EATRI2="DESC2" EATRI3="DESC3"
USE A:TEMP
DISPLAY FOR EATRI3="DESC3" OFF FIELDS:
NAME,SSN,EATRI1,EATRI2,EATRI3
IF EOF
@ 23,50 SAY "SEARCH COMPLETE"
ENDIF
CASE (SLCT3 = "E" .OR. SLCT3 = "K")
USE F:EST
COPY TO A:TEMP FIELD SSN,EATRI3 FOR EATRI3 = "DESC3"
SELECT PRIMARY
USE A:TEMP
SELECT SECONDARY
USE A:PERSON

90
JOIN TG SLCTDATA FOR P.SSN=S.SSN FIELDS:
NAME SSN &ATRI1, &ATRI2, &ATRI3
USE SLCTDATA
DISPLAY FOR &ATRI1 = "EDESC1" .AND. &ATRI2;
= "EDESC2" OFF FIELDS NAME, SSN, &ATRI1;
&ATRI2, &ATRI3
IF ECF
@ 2, 50 SAY "SEARCH COMPLETE"
ENDIF
CASE (SLCT3 >= "I" .AND. SLCT3 <= "N")
USE E:QUAL
COPY TO A:TEMP FIELD SSN, &ATRI3 FOR &ATRI3;
= "EDESC3"
SELECT PRIMARY
USE A:TEMP
SELECT SECONDARY
USE A:PERSONS
JOIN TO SLCTDATA FOR P.SSN=S.SSN FIELDS:
NAME SSN, &ATRI1, &ATRI2, &ATRI3
USE SLCTDATA
DISPLAY FOR &ATRI1 = "EDESC1" .AND. &ATRI2;
= "EDESC2" OFF FIELDS NAME, SSN, &ATRI1;
&ATRI2, &ATRI3
IF ECF
@ 2, 50 SAY "SEARCH COMPLETE"
ENDIF
CASE (SLCT3 >= "I" .AND. SLCT3 <= "N")
DO CASE
CASE (SLCT3 >= "A" .AND. SLCT3 <= "I")
USE E:QUAL
COPY TO A:TEMP FIELD SSN, &ATRI3 FOR &ATRI3;
= "EDESC3"
SELECT PRIMARY
USE A:TEMP
SELECT SECONDARY
USE A:PERSONS
JOIN TO SLCTDATA FOR P.SSN=S.SSN FIELDS:
NAME SSN, &ATRI1, &ATRI2, &ATRI3
USE SLCTDATA
DISPLAY FOR &ATRI1 = "EDESC1" .AND. &ATRI2;
= "EDESC2" OFF FIELDS NAME, SSN, &ATRI1;
&ATRI2, &ATRI3
IF ECF
@ 2, 50 SAY "SEARCH COMPLETE"
ENDIF
CASE (SLCT3 >= "I" .AND. SLCT3 <= "N")
DO CASE
CASE (SLCT3 >= "A" .AND. SLCT3 <= "I")
USE E:QUAL
COPY TO A:TEMP FIELD NAME SSN, &ATRI1 FOR; &ATRI1 = "EDESC1"
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 &ATRI2 = "EDESC2" .AND. &ATRI3;
= "EDESC3" OFF FIELDS NAME, SSN, &ATRI1;
&ATRI2, &ATRI3
IF ECF
@ 2, 50 SAY "SEARCH COMPLETE"
ENDIF
CASE (SLCT3 = "J" .OR. SLCT3 = ")
USE A:PERSONS
COPY TO A:TEMP FIELD NAME SSN, &ATRI1 FOR; &ATRI1 = "EDESC1"
SELECT PRIMARY
USE A:TEMP

91
SELECT SECONDARY
USE E:QUAL
JOIN TO A:TEMP2 FOR P.SSN=S.SSN FIELDS;
NAME, SSN, &ATRI1, &ATRI2
SELECT PRIMARY
USE A:TEMP2
SELECT SECONDARY
USE E:EST
JOIN TO SLCTDATA FOR P.SSN=S.SSN FIELDS;
NAME, SSN, &ATRI1, &ATRI2, &ATRI3
USE SLCTDATA
DISPLAY FOR &ATRI1 = "$DESC1" .AND. &ATRI3;
&ATR1 = "$DESC2" OFF FIELDS NAME, SSN, &ATRI1;
&ATRI2, &ATRI3
IF ECF
@ 23,50 SAY "SEARCH COMPLETE"
ENDIF
ENDCASE
CASE (SLCT2 = "J" .OR. SLCT2 = "K")
DO CASE
CASE (SLCT3 >= "A" .AND. SLCT3 <= "H")
USE A:EST
COPY TO A:TEMP FIELD SSN,&ATRI2 FOR &ATRI2;
= "$DESC2"
SELECT PRIMARY
USE A:TEMP
SELECT SECONDARY
USE A:PEPS
JOIN TO SLCTDATA FOR P.SSN=S.SSN FIELDS:
NAME, SSN, &ATRI1, &ATRI2, &ATRI3
USE SLCTDATA
DISPLAY FOR &ATR1 = "$DESC1" .AND. &ATRI3;
&ATRI2, &ATRI3
IF ECF
@ 23,50 SAY "SEARCH COMPLETE"
ENDIF
CASE (SLCT3 >= "I" .AND. SLCT3 <= "N")
USE A:PEPS
COPY TO A:TEMP FIELD NAME,SSN,&ATRI1 FOR;
&ATRI1 = "$DESC1"
USE A:EST
COPY TO A:TEMP2 FIELD SSN,&ATRI2 FOR;
&ATRI2 = "$DESC2"
SELECT PRIMARY
USE A:TEMP
SELECT SECONDARY
USE A:TEMP2
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 = "$DESC3" OFF FIELDS;
NAME, SSN, &ATRI1, &ATRI2, &ATRI3
IF ECF
@ 23,50 SAY "SEARCH COMPLETE"
ENDIF
CASE (SLCT3 = "J" .OR. SLCT3 = "K")
USE A:PEPS
COPY TO A:TEMP FIELD NAME,SSN,&ATRI1 FOR;
&ATRI1 = "$DESC1"
SELECT PRIMARY
USE A:TEMP
SELECT SECONDARY
92
USE F:EST
JOIN TO SLCTDATA FOR P.SSN=S.SSN FIELDS:
NAME, SSN, &ATR1, &ATR2, &ATR3
USE SLCTDATA
DISPLAY FOR &ATR2 = "ESDC2" .AND. &ATR13;
= "ESDC3" OFF FIELDS NAME, SSN, &ATR1,
&ATR2, &ATR3
IF ECF
@ 23,50 SAY "SEARCH COMPLETE"
ENCASE
ENDCASE
**BEGIN CUAL / * / *
CASE (SLCT1 >= "I" .AND. SLCT1 <= "N")
**EAGAIN CUAL / 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, &ATR2,
&ATR3 FOR &ATR2="ESDC2" .AND.;
&ATR3="ESDC3"
SELECT PRIMARY
USE A:TEMP
SELECT SECONDARY
USE F:QUAL
JOIN TO SLCTDATA FOR P.SSN=S.SSN FIELDS:
NAME, SSN, &ATR1, &ATR2, &ATR3
USE SLCTDATA
DISPLAY FOR &ATR1="ESDC1" OFF FIELDS;
NAME, SSN, &ATR1, &ATR2, &ATR3
IF ECF
@ 23,50 SAY "SEARCH COMPLETE"
ENDIF
CASE (SLCT3 = "J" .OR. SLCT3 = "K")
USE F:EST
COPY TO A:TEMP FIELD Ssn, &ATR3 FOR &ATR3;
= "ESDC3"
USE F:QUAL
COPY TO A:TEMP2 FIELD Ssn, &ATR1 FOR;
&ATR1="ESDC1"
SELECT PRIMARY
USE A:TEMP
SELECT SECONDARY
USE A:TEMP2
JOIN TO A:TEMP3 FOR P.SSN=S.SSN FIELDS:
SSN, &ATR1, &ATR3
SELECT PRIMARY
USP A:TEMP3
SELECT SECONDARY
USE A:PERS
JOIN TO SLCTDATA FOR P.SSN=S.SSN FIELDS:
NAME, SSN, &ATR1, &ATR2, &ATR3
USE SLCTDATA
DISPLAY FOR &ATR2 = "ESDC2";
OFF FIELDS NAME, Ssn, &ATR1, &ATR2, &ATR3
IF ECF
@ 23,50 SAY "SEARCH COMPLETE"
ENDIF
CASE (SLCT3 >= "L" .AND. SLCT3 <= "N")
USE F:QUAL
COPY TO A:TEMP FIELD Ssn, &ATR1, &ATR3;
FOR &ATR3 = "ESDC3" .AND. &ATR1 =;
"ESDC1"
SELECT PRIMARY
USE A:TEMP
SELECT SECONDARY
USE A:PERS
93
JOIN TO SLCTDATA FOR P.SSN=S.SSN FIELDS:
NAME, SSN, &ATRI1, &ATRI2, &ATRI3
USE SLCTDATA
DISPLAY FOR &ATRI2 = "EDESC2";
OFF FIELDS NAME, SSN, &ATRI1, &ATRI2, &ATRI3
IF ECF
  @ 23,50 SAY "SEARCH COMPLETE"
ENDIF
ENDCASE

CASE (SLCT2 >= "I" .AND. SLCT2 < "N")
DO CASE
  CASE (SLCT3 >= "A" .AND. SLCT3 < "I")
    USE E:QUAL
    COPY TO A:TEMP FIELD SSN, &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, SSN, &ATRI1, &ATRI2, &ATRI3
    USE SLCTDATA
    DISPLAY FOR &ATRI3 = "EDESC3";
    OFF FIELDS NAME, SSN, &ATRI1, &ATRI2, &ATRI3
    IF ECF
      @ 23,50 SAY "SEARCH COMPLETE"
    ENDIF
  ENDIF
  CASE (SLCT3 >= "I" .AND. SLCT3 < "N")
    USE E:QUAL
    COPY TO A:TEMP FIELD SSN, &ATRI1, &ATRI2;
    &ATRI3 FOR &ATRI1 = "EDESC1" .AND. &ATRI2 = "EDESC2"
    .AND. &ATRI3 = "EDESC3"
    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 ALL OFF FIELDS NAME, SSN,:
    &ATRI1, &ATRI2, &ATRI3
    IF ECF
      @ 23,50 SAY "SEARCH COMPLETE"
    ENDIF
  ENDIF
  CASE (SLCT3 = "J" .OR. SLCT3 = "K")
    USE E:QUAL
    COPY TO A:TEMP FIELD SSN, &ATRI1, &ATRI2;
    FOR &ATRI1 = "EDESC1" .AND. &ATRI2 = "EDESC2"
    USE E:EST
    COPY TO A:TEMP2 FIELD SSN, &ATRI3 FOR;
    &ATRI3 = "EDESC3"
    SELECT PRIMARY
    USE A:TEMP
    SELECT SECONDARY
    USE A:TEMP2
    JOIN TO A:TEMP3 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
      @ 23,50 SAY "SEARCH COMPLETE"
    ENDIF
  ENDIF
  ENDIF
  CASE (SLCT3 = "J" .OR. SLCT3 = "K")
    USE E:QUAL
    COPY TO A:TEMP FIELD SSN, &ATRI1, &ATRI2;
    FOR &ATRI1 = "EDESC1" .AND. &ATRI2 = "EDESC2"
    USE E:EST
    COPY TO A:TEMP2 FIELD SSN, &ATRI3 FOR;
    &ATRI3 = "EDESC3"
    SELECT PRIMARY
    USE A:TEMP
    SELECT SECONDARY
    USE A:TEMP2
    JOIN TO A:TEMP3 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
      @ 23,50 SAY "SEARCH COMPLETE"
    ENDIF
  ENDIF
ENDCASE
CASE (SLCT2 = "J", CR, SLCT2 = "K")
  LO CASE
  CASE (SLCT3 >= "A" AND SLCT3 <= "I")
    USE F:EST
    COPY TO A:TEMP FIELD SSN, &ATRI2 FOR &ATRI2 = "&DESC2"
    USE EQUAL
    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:TEMP2
    DISPLAY FOR S.&ATRI1="&DESC1", AND:
    P.SSN=S.SSN OFF FIELDS NAME, SSN, &ATRI1,
    &ATRI2, &ATRI3
    IF ECF @ 22,50 SAY "SEARCH COMPLETE"
  ENDIF CASE
  CASE (SLCT3 >= "I" AND SLCT3 <= "N")
    USE F:EST
    COPY TO A:TEMP FIELD SSN, &ATRI1, &ATRI3;
    FOR &ATRI1 = "&DESC1", AND, &ATRI3 = "&DESC3"
    USE F:EST
    COPY TO A:TEMP2 FIELD SSN, &ATRI2 FOR;
    &ATRI2 = "&DESC2"
    SELECT PRIMARY
    USE A:TEMP
    SELECT SECONDARY
    USE A:TEMP2
    JOIN TO A:TEMP3 FOR P.SSN=S.SSN FIELDS:
    SSN, &ATRI1, &ATRI2, &ATRI3
    SELECT PRIMARY
    USE A:TEMP3
    SELECT SECONDARY
    USE A:TEMP2
    DISPLAY FOR F.SSN=S.SSN OFF FIELDS NAME,
    P.SSN, SSN, &ATRI1, &ATRI2, &ATRI3
    IF ECF @ 22,50 SAY "SEARCH COMPLETE"
  ENDIF CASE
  CASE (SLCT3 = "J", OR, SLCT3 = "K")
    USE F:EST
    COPY TO A:TEMP FIELD SSN, &ATRI2, &ATRI3;
    FOR &ATRI2 = "&DESC2", AND, &ATRI3 = "&DESC3"
    USE EQUAL
    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, &ATRI3
    SELECT PRIMARY
    USE A:TEMP3
    SELECT SECONDARY
    USE A:TEMP2
    DISPLAY FOR F.SSN=S.SSN OFF FIELDS NAME, SSN, &ATRI1, &ATRI2, &ATRI3
    IF ECF @ 22,50 SAY "SEARCH COMPLETE"
  ENDIF
** BEGIN CASE (SLCT1 = "J" OR SLCT1 = "K")
** BEGIN EST / SERS / *
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 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:TEMP2
DISPLAY FOR P.SSN=S.SSN OFF FIELDS NAME,;
SSN, &ATRI1, &ATRI2, &ATRI3
IF ECF
@ 23,50 SAY "SEARCH COMPLETE"
ENDC
CASE (SLCT3 >= "J" .AND. 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.&ATRI2 =;
"DESC2" OFF FIELDS NAME, SSN, &ATRI1,;
&ATRI2, &ATRI3
IF ECF
@ 23,50 SAY "SEARCH COMPLETE"
ENDC
CASE (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:TEMP2
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"
ENDC
ENDCASE
CASE (SLCT2 = "L" .AND. SLCT2 <= "N")
DO CASE
CASE (SLCT3 = "I")
USE E:QUAL
COPY TO A:TEMP FIELD SSN, &ATRI2 FOR &ATRI2;
= "&DESC2"
USE E:EST
COPY TO A:TEMP2 FIELD SSN,&ATRI1 FOR;
&ATRI1="&DESC1"
SELECT PRIMARY
USE A:TEMP
SELECT SECONDARY
USE A:TEMP2
JOIN TO SCLTDATA FOR P.SSN=S.SSN FIELDS;
SSN, &ATRI1, &ATRI2
SELECT PRIMARY
USE SCLTDATA
SELECT SECONDARY
USE A:TEMP3
DISPLAY FOR P.SSN=S.SSN OFF FIELDS NAME, SSN, &ATRI1,
&ATRI2, &ATRI3
IF ECF
@ 23,50 SAY "SEARCH COMPLETE"
ENDIF
CASE (SLCT3 >= "I" .AND. SLCT3 <= "N")
USE E:EQUAL
COPY TO A:TEMP FIELD SSN,&ATRI2,&ATRI3;
FOR &ATRI2="&DESC2", &ATRI3="&DESC3"
USE E:EST
COPY TO A:TEMP2 FIELD SSN,&ATRI1 FOR;
&ATRI1="&DESC1" 
SELECT PRIMARY
USE A:TEMP
SELECT SECONDARY
USE A:TEMP2
JOIN TO SCLTDATA FOR P.SSN=S.SSN FIELDS;
SSN, &ATRI1, &ATRI2, &ATRI3
SELECT PRIMARY
USE SCLTDATA
SELECT SECONDARY
USE A:TEMP3
DISPLAY FOR P.SSN=S.SSN OFF FIELDS NAME, SSN, &ATRI1,
&ATRI2, &ATRI3
IF ECF
@ 23,50 SAY "SEARCH COMPLETE"
ENDIF
CASE (SLCT3 = "J" .OR. SLCT3 = "K")
USE E:EST
COPY TO A:TEMP FIELD SSN,&ATRI1,&ATRI3;
FOR &ATRI1="&DESC1", &ATRI3="&DESC3"
USE E:EQUAL
COPY TO A:TEMP2 FIELD SSN,&ATRI2 FOR;
&ATRI2="&DESC2"
SELECT PRIMARY
USE A:TEMP
SELECT SECONDARY
USE A:TEMP2
JOIN TO A:TEMP3 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
@ 23,50 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:Pers
DISPLAY FOR P.SSN = S.SSN . AND. &ATRI3 =:
"&DESC3" OFF FIELDS NAME, SSN, &ATRI1, ,
&ATRI2, &ATRI3
IF BCE@ 23,50 SAY "SEARCH COMPLETE"
ENDIF
CASE (SLCT3 >= "I" . ANP. SLCT3 <= "N")
USE E: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;
&ATRI2 = "&DESC3"
SELECT PRIMARY
USE A:TEMP
SELECT SECONDARY
USE A:TEMP2
JOIN TO ATLCDATA FOR P.SSN = S.SSN FIELDS:
SSN, &ATRI1, &ATRI2, &ATRI3
SELECT PRIMARY
USE ATLCDATA
SELECT SECONDARY
USE A:Pers
DISPLAY FOR P.SSN = S.SSN OFF FIELDS NAME,;
SSN, &ATRI1, &ATRI2, &ATRI3
IF BCE@ 23,50 SAY "SEARCH COMPLETE"
ENDIF
CASE (SLCT3 = "3" . OR. SLCT3 = "F")
USE A:EST
COPY TO A:TEMP FIELD SSN, &ATRI1, &ATRI2; 
&ATRI3 FOR &ATRI1 = "&DESC1" . AND. &ATRI2 =
"&DESC2" . ANP. &ATRI3 = "&DESC3"
SELECT PRIMARY
USE A:TEMP
SELECT SECONDARY
USE A:Pers
DISPLAY FOR P.SSN = S.SSN OFF FIELDS NAME,;
SSN, &ATRI1, &ATRI2, &ATRI3
IF BCE@ 23,50 SAY "SEARCH COMPLETE"
ENDIF
ENDCASE
ENDCASE
ACCEPT "PRESS 'ENTER' TO CONTINUE" TO PROCEED
ENDIF
USE
RELEASE ALL LIKE M????????
RELEASE REPEAT COUNT
RELEASE ALL LIKE &ATRI????
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: NCRECRT GENERATOR
* VERSION: 4.1.0
* AUTHOR: R.E. PUIETTI
* DATE: 10FEB84
* VARIABLES USED: NONE
* VARIABLES MODIFIED: NONE
* VARIABLES CREATED: PROCEED
* VARIABLES RELEASED: NONE
* FILES OPENED/CLOSED: NONE
* TENT FILES CREATED: NONE
* USING SUBROUTINES: NOSTDRPT.PRG (4.0)
* DESCRIPTION: THIS ROUTINE SUPPLIES A RESPONSE TO THE USER'S REQUEST FOR "HELP" WHILE IN THE NONSTANDARD REPORT MENU.

ERASE
@0,25 SAY "NONSTANDARD COMMAND GENERATOR"
@ 2,0 SAY "LOCATE ALL MARINES THAT SATISFY THE FOLLOWING:
CONDITION:"<ATTRIBUTE> <DESCRIPTOR>"
@ 4,0 SAY " RANK IS E5"
@ 6,0 SAY "UP TO THREE CONDITION STATEMENTS MAY BE;
JOINED"
@ 8,0 SAY "FOR EXAMPLE"
@ 10,0 SAY "LOCATE ALL MARINES THAT SATISFY THE FOLLOWING:
CONDITIONS:"<ATTRIBUTE> <DESCRIPTOR>"
@ 14,0 SAY " RANK IS E5"
@ 16,0 SAY "COMPANY AND IS A CO"
@ 18,0 SAY "PRIMARY MOS IS 0331"
@ 20,0 SAY "JOINING 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"
@ 24,0 ACCEPT "PRESS 'ENTER' TO CONTINUE" TO PROCEED
RETURN
**C. SELECT ATTRIBUTE ROUTINE**

* ROUTINE NAME: SLCTATRL.PRG  
* MODULE 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, MATH1, MATH1,  
* MATH1, MATH2, MATH3, MATH3, MATH3  
* VARIABLES RELEASED: NONE  
* FILES OPENED/CLOSED: NONE  
* USING SUBROUTINES: NCSTDRPT.PRG (4.0)  

**DESCRIPTION:**  
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. 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.

```plaintext
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
ENDIF
IF COUNT = 3
DO CASE
CASE SLCT2 = "A"
@1,1 SAY "*
CASE SLCT2 = "B"
@2,1 SAY "*
CASE SLCT2 = "C"
@3,1 SAY "*
CASE SLCT2 = "D"
@4,1 SAY "*
CASE SLCT2 = "E"
@5,1 SAY "*
CASE SLCT2 = "F"
@6,1 SAY "*
CASE SLCT2 = "G"
@7,1 SAY "*
CASE SLCT2 = "H"
@8,1 SAY "*
CASE SLCT2 = "I"
@9,1 SAY "*
CASE SLCT2 = "J"
@10,1 SAY "*
CASE SLCT2 = "K"
@11,1 SAY "*
CASE SLCT2 = "L"
@12,1 SAY "*
CASE SLCT2 = "M"
@13,1 SAY "*
CASE SLCT2 = "N"
@14,1 SAY "*
```
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...EFT CIE"
& 11,2 SAY "K...PFT 10"
& 12,2 SAY "L...SWIM QUALIFICATION"
& 13,2 SAY "M...FISTCI QUALIFICATION"
& 14,2 SAY "N...PISTCI QUALIFICATION"

ACCEPT "SELECT ATTRIBUTE == >" TO MATRIBUT
IF COUNT = 1
STORE MATRIBUT TO SICT1
DO CASE
CASE ! (SLCT1) = "A"
STORE "RANK" TO MATRIBUT
STORE "RANK" TO MATRIBUT
CASE ! (SLCT1) = "F"
STORE "PRIMERA MOS" TO MATRIBUT
STORE "PRIMERO MOS" TO MATRIBUT
CASE ! (SLCT1) = "C"
STORE "SECONDARY MOS" TO MATRIBUT
STORE "SEGUNDO MOS" TO MATRIBUT
CASE ! (SLCT1) = "E"
STORE "COMPANY" TO MATRIBUT
STORE "COMPANIA" TO MATRIBUT
CASE ! (SLCT1) = "F"
STORE "PLATOON" TO MATRIBUT
STORE "PLATOON" TO MATRIBUT
CASE ! (SLCT1) = "F"
STORE "JOIN DATE" TO MATRIBUT
STORE "FECHA DE INICIO" TO MATRIBUT
CASE ! (SLCT1) = "F"
STORE "BIRTH DATE" TO MATRIBUT

101
ELSE
  STORE T TO REPEAT
ENDCASE
ENDIF

IF CNT = 3
  STORE WATRIBUT TO SLCT3
DO CASE
  CASE !(SLCT3) = "A"
  STORE "RANK" TO ATRI3
  STORE "RANK" TO ATR13
  CASE !(SLCT3) = "E"
  STORE "PRIMARY MOS" TO MATRI3
  STORE "PRIM MOS" TO ATR13
  CASE !(SLCT3) = "C"
  STORE "SECONDARY MOS" TO MATRI3
  STORE "SECMOS" TO ATR13
  CASE !(SLCT3) = "D"
  STORE "COMPANY" TO MATRI3
  STORE "COMPANY" TO ATR13
  CASE !(SLCT3) = "E"
  STORE "PLATCON" TO MATRI3
  STORE "PLATCON" TO ATR13
  CASE !(SLCT3) = "F"
  STORE "JOINATE" TO MATRI3
  STORE "JOINATE" TO ATR13
  CASE !(SLCT3) = "G"
  STORE "EAS" TO MATRI3
  STORE "EAS" TO ATR13
  CASE !(SLCT3) = "H"
  STORE "BIRTH DATE" TO MATRI3
  STORE "BIRTH DATE" TO ATR13
  CASE !(SLCT3) = "I"
  STORE "WEIGHT CONTROL" TO MATRI3
  STORE "WTCON" TO ATR13
  CASE !(SLCT3) = "J"
  STORE "PPT ONE" TO MATRI3
  STORE "PPT1CLASS" TO ATR13
  CASE !(SLCT3) = "K"
  STORE "PPT TWO" TO MATRI3
  STORE "PPT2CLASS" TO ATR13
  CASE !(SLCT3) = "L"
  STORE "SWIM QUALIFICATION" TO MATRI3
  STORE "SWIMQUAL" TO ATR13
  CASE !(SLCT3) = "M"
  STORE "RIFLE QUALIFICATION" TO MATRI3
  STORE "RFLQUAL" TO ATR13
  CASE !(SLCT3) = "N"
  STORE "PISTOL QUALIFICATION" TO MATRI3
  STORE "PSTQUAL" TO ATR13
  OTHERWISE
  STORE T TO REPEAT
ENDCASE
ENDIF
DO SLCTDESC
ENDDC
RETURN
D. SELECT DESCRIPTOR ROUTINE

* Routine Name: SLCTDESC.PRG
* Module Name: NONSTANDARD REPORT GENERATOR
* Version: 4.2.1.0
* Author: R.E. Pruett
* Date: 4/28/84
* Variables Used: COUNT
* Variables Modified: DESC1, DESC2, DESC3
* Variables Created: MKCOMP, MSECMS, MCOMPANY, MPLATOON,
  MJJOINTE, MEAS, MUTCNTL, MPFTONE, MPFTTWO, MSWQUAL,
  MRIFQAL, MISPQAL, MKCOMP, MSECMS, MCOMPANY, MPLAT3,
  MJJOINTE, MEAS, MUTCNTL, MPFTONE, MPFTTWO,
  MSECMS, MCOMPANY, MPLAT3, MJJOINTE, MEAS, MUTCNTL,
  MPFTONE, MPFTTWO, MSECMS, MCOMPANY, MPLAT3,
  MJJOINTE, MEAS, MUTCNTL, MPFTONE, MPFTTWO,
  MSECMS, MCOMPANY, MPLAT3, MJJOINTE, MEAS, MUTCNTL,
  MPFTONE, MPFTTWO, MSECMS, MCOMPANY, MPLAT3,
* Variables Released: NONE
* Files Opened/Closed: NONE
* Tapes Files Created: NONE
* Using Subroutines: SLCTATRI.PRG (4.2.0)
* Description: This routine prompts 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).

IF COUNT = 1
DO CASE
CASE !(SLCT1) = "A" 
STORE ' ' TO MKCOMP
@ 8.0 SAY "TYPE IN DESIRED COMPANY DESCRIPTION";
@ 8.58 GET MKCOMP PICTURE 'XXX'
READ
STORE MKCOMP TO DESC1
CASE !(SLCT1) = "B" 
STORE ' ' TO MSECMS
@ 8.0 SAY "TYPE IN DESIRED SECONDARY MOS DESCRIPTION";
@ 8.58 GET MSECMS PICTURE 'XXX'
READ
STORE MSECMS TO DESC1
CASE !(SLCT1) = "C" 
STORE ' ' TO MPLATOON
@ 8.58 GET MPLATOON PICTURE 'XXXXXXXX'
READ
STORE MPLATOON TO DESC1
END CASE
ELSE
CASE !(SLCT1) = "A" 
STORE ' ' TO MKCOMP
@ 8.0 SAY "TYPE IN DESIRED COMPANY DESCRIPTION (1ST)";
@ 8.58 GET MKCOMP PICTURE 'XXX'
READ
STORE MKCOMP TO DESC1
CASE !(SLCT1) = "B" 
STORE ' ' TO MSECMS
@ 8.0 SAY "TYPE IN DESIRED SECONDARY MOS DESCRIPTION (1ST)";
@ 8.58 GET MSECMS PICTURE 'XXX'
READ
STORE MSECMS TO DESC1
CASE !(SLCT1) = "C" 
STORE ' ' TO MPLATOON
@ 8.0 SAY "TYPE IN DESIRED JOIN DATE DESCRIPTION";

104
"(220183,0183,83,ETC.)"
"8604 GETJOINITE PICTURE 'XXXXX'
READ
STORE JOINITE TC DESC1
CASE 1(SICT1) = "G"
STORE ' ' TO MEAS
@ 8,0 SAY "TYPE IN DESIRED EAS DESCRIPTION";
"(220183,0183,83,ETC.)"
@ 8.55 GET MEAS PICTURE 'XXXXX'
READ
STORE MEAS TO DESC1
CASE 1(SICT1) = "H"
STORE ' ' TO MBIRDATE
@ 8,0 SAY "TYPE IN DESIRED BIRTH DATE DESCRIPTION";
"(220183,0183,83,ETC.)"
@ 8,50 GET MBIRDATE PICTURE 'XXXXX'
READ
STORE MBIRDATE TC DESC1
CASE 1(SICT1) = "I"
STORE ' ' TO MTWTCONTL
@ 8,0 SAY "TYPE IN DESIRED WEIGHT CONTROL DESCRIPTION";
"(YES OR NO)"
@ 8,50 GET MTWTCONTL PICTURE 'XXX'
READ
STORE MTWTCONTL TC DESC1
CASE 1(SICT1) = "J"
STORE ' ' TO MBFTONE
@ 8,0 SAY "TYPE IN DESIRED PFT ONE DESCRIPTION";
"(YES OR NO)"
@ 8,50 GET MBFTONE PICTURE 'X'
READ
STORE MBFTONE TO DESC1
CASE 1(SICT1) = "K"
STORE ' ' TO MBPTTWO
@ 8,0 SAY "TYPE IN DESIRED PFT TWO DESCRIPTION";
"(YES OR NO)"
@ 8,50 GET MBPTTWO PICTURE 'X'
READ
STORE MBPTTWO TO DESC1
CASE 1(SICT1) = "L"
STORE ' ' TO MSWQUAL
@ 8,0 SAY "TYPE IN DESIRED SWIMMING QUALIFICATION";
"DESCRIPTION"
@ 9,0 SAY "(YES OR NO)"
@ 9,17 GET MSWQUAL PICTURE 'XX'
READ
STORE MSWQUAL TC DESC1
CASE 1(SICT1) = "M"
STORE ' ' TO MRIFQUAL
@ 8,0 SAY "TYPE IN DESIRED RIFLE QUALIFICATION";
"DESCRIPTION"
@ 9,0 SAY "(YES OR NO)"
@ 9,18 GET MRIFQUAL PICTURE 'XX'
READ
STORE MRIFQUAL TO DESC1
CASE 1(SICT1) = "N"
STORE ' ' TO MFSQUAL
@ 8,0 SAY "TYPE IN DESIRED PISTOL QUALIFICATION";
"DESCRIPTION"
@ 9,0 SAY "(YES OR NO)"
@ 9,18 GET MFSQUAL PICTURE 'XX'
READ
STORE MFSQUAL TO DESC1
END CASE
END IF
COUNT = 2
TO CASE
CASE 1(SICT2) = "A"
STORE ' ' TO MFRANK2
105
a $8,0$ SAY "TYPE IN DESIRED RANK DESCRIPTION:
(1E85,02,ETC.)"
@ $8,48$ GET MRANK2 PICTURE 'XX'
REAL
STORE MRANK2 TO DESC2
CASE 1(SLCT2) = "E" NO
STORE " " TC PPRIMCS2
a $8,0$ SAY "TYPE IN DESIRED PRIMARY MOS DESCRIPTION:
(031,1,0331,ETC.)"
@ $8,58$ GET PPRIMCS2 PICTURE 'XXXX'
REAL
STORE PPRIMCS2 TC DESC2
CASE 1(SLCT2) = "C" NO
STORE " " TC PSECMS2
a $8,0$ SAY "TYPE IN DESWRED SECONDARY MOS DESCRIPTION:
(0331,0331,ETC.)"
@ $8,68$ GET PSECMS2 PICTURE 'XXXX'
REAL
STORE PSECMS2 TC DESC2
CASE 1(SLCT2) = "X" TO MCOMPY2
a $8,0$ SAY "TYPE IN DESIRED COMPANY DESCRIPTION (A:
COMMS COH&S CC)"
@ $8,58$ GET MCOMPY2 PICTURE 'XXXXXX'
REAL
STORE MCOMPY2 TC DESC2
CASE 1(SLCT2) = "I" TO MELAT2
a $8,0$ SAY "TYPE IN DESIRED PLATOON DESCRIPTION (1ST:
PLATO ONETE ETC.)"
@ $8,58$ GET MELAT2 PICTURE 'XXXXXXX'
REAL
STORE MELAT2 TO DESC2
CASE 1(SLCT2) = "E" NO
STORE " " TC MJNDE2
a $8,0$ SAY "TYPE IN DESIRED JOIN DATE DESCRIPTION:
(22183,0183,ETC.)"
@ $8,68$ GET MJNDE2 PICTURE 'XXXXXX'
REAL
STORE MJNDE2 TC DESC2
CASE 1(SLCT2) = "F" TO MEAS2
a $8,0$ SAY "TYPE IN DESIRED EAS DESCRIPTION:
(22183,0183,ETC.)"
@ $8,55$ GET MEAS2 PICTURE 'XXXXX'
REAL
STORE MEAS2 TO DESC2
CASE 1(SLCT2) = "H" TO MBIRDTE2
a $8,0$ SAY "TYPE IN DESIRED BIRTH DATE DESCRIPTION:
(22183,0183,ETC.)"
@ $8,68$ GET MBIRDTE2 PICTURE 'XXXXXX'
REAL
STORE MBIRDTE2 TC DESC2
CASE 1(SLCT2) = "I" TO MWTCTRL2
a $8,0$ SAY "TYPE IN DESIRED WEIGHT CONTROL DESCRIPTION:
(1E02,ETC.)"
@ $8,55$ GET MWTCTRL2 PICTURE 'XXX'
REAL
STORE MWTCTRL2 TC DESC2
CASE 1(SLCT2) = "J" TO MPFFCNE2
a $8,0$ SAY "TYPE IN DESIRED PFT ONT DESCRIPTION:
(32183,ETC.)"
@ $8,50$ GET MPFFCNE2 PICTURE 'X'
REAL
STORE MPFFCNE2 TC DESC2
CASE 1(SLCT2) = "K"
STORE ' "TC MPITWO2 "
@ 8,0 SAY "TYPE IN DESIRED PFT TWO DESCRIPTION:" (S, 1, 2, 3, F, )
@ 8,50 GET MPITWO2 PICTURE 'X'
REAL
STORE MPITWO2 TC DESC2
CASE 1(SLCT2) = "I"
STORE ' "TO MSWMOAL2 "
@ 8,0 SAY "TYPE IN DESIRED SWIMMING QUALIFICATION; DESCRIPTION"
@ 9,0 SAY " (S1, S2, S3, UN, )"
@ 9,17 GET MSWMOAL2 PICTURE 'XX'
REAL
STORE MSWMOAL2 TC DESC2
CASE 1(SLCT2) = "E"
STORE ' "TO MSIFQAL2 "
@ 8,0 SAY "TYPE IN DESIRED RIFLE QUALIFICATION; DESCRIPTION"
@ 9,0 SAY " (EX, SS, MM, UN, )"
@ 9,18 GET MSIFQAL2 PICTURE 'XX'
REAL
STORE MSIFQAL2 TC DESC2
CASE 1(SLCT2) = "N"
STORE ' "TO MPRIMCS3 "
@ 8,0 SAY "TYPE IN DESIRED PRIMARY MOS DESCRIPTION; (03, 0331, ETC.)"
@ 8,18 GET MPRIMCS3 PICTURE 'XXX'
REAL
STORE MPRIMCS3 TC DESC3
CASE 1(SLCT3) = "C"
STORE ' "TO MSECMCS3 "
@ 8,0 SAY "TYPE IN DESIRED SECONDARY MOS DESCRIPTION; (03, 0331, ETC.)"
@ 8,58 GET MSECMCS3 PICTURE 'XXXX'
REAL
STORE MSECMCS3 TC DESC3
CASE 1(SLCT3) = "C"
STORE ' "TO MCOMPNY3 "
@ 8,0 SAY "TYPE IN DESIRED COMPANY DESCRIPTION (A; CC, LENS CO, H&G CC)"
@ 8,58 GET MCOMPNY3 PICTURE 'XXXXX'
REAL
STORE MCOMPNY3 TC DESC3
CASE 1(SLCT3) = "I"
STORE ' "TO MLPAT3 "
@ 8,0 SAY "TYPE IN DESIRED PLATOON DESCRIPTION (1ST; PLAT, ETC)"
@ 8,60 GET MLPAT3 PICTURE 'XXXXXXXXX'
REAL
STORE MLPAT3 TC DESC3
END CASE
ENDIF
IP CCENT = 3
DO CASE 1(SLCT3) = "A"
STORE ' "TO MFANK3 "
@ 8,0 SAY "TYPE IN DESIRED RANK DESCRIPTION; (E1, E5, O2, ETC.)"
@ 8,18 GET MFANK3 PICTURE 'XX'
REAL
STORE MFANK3 TO DESC3
CASE 1(SLCT3) = "E"
STORE ' "TO MPRIMCS3 "
@ 8,0 SAY "TYPE IN DESIRED PRIMARY MOS DESCRIPTION; (03, 0331, ETC.)"
@ 8,18 GET MPRIMCS3 PICTURE 'XXX'
REAL
STORE MPRIMCS3 TC DESC3
CASE 1(SLCT3) = "C"
STORE ' "TO MSECMCS3 "
@ 8,0 SAY "TYPE IN DESIRED SECONDARY MOS DESCRIPTION; (03, 0331, ETC.)"
@ 8,58 GET MSECMCS3 PICTURE 'XXXX'
REAL
STORE MSECMCS3 TC DESC3
CASE 1(SLCT3) = "C"
STORE ' "TO MCOMPNY3 "
@ 8,0 SAY "TYPE IN DESIRED COMPANY DESCRIPTION (A; CC, LENS CO, H&G CC)"
@ 8,58 GET MCOMPNY3 PICTURE 'XXXXX'
REAL
STORE MCOMPNY3 TC DESC3
CASE 1(SLCT3) = "I"
STORE ' "TO MLPAT3 "
@ 8,0 SAY "TYPE IN DESIRED PLATOON DESCRIPTION (1ST; PLAT, ETC)"
@ 8,60 GET MLPAT3 PICTURE 'XXXXXXXXX'
REAL
STORE MLPAT3 TC DESC3

107
CASE (SLCT3) = "F"
  STORE 'TC MJNDTE3
  @ 8.0 SAY "TYPE IN DESIRED JOIN DATE DESCRIPTION;"
  (226183,0183,83, ETC.)"
  @ 8.64 GET MJNDTE3 PICTURE 'XXXXX'
END
STORE MJNDTE3 TC DESC3
CASE (SLCT3) = "G"
  STORE 'TC MEAS3
  @ 8.0 SAY "TYPE IN DESIRED EAS DESCRIPTION;"
  (226183,0183,83, ETC.)"
  @ 8.35 GET MEAS3 PICTURE 'XXXXX'
END
STORE MEAS3 TC DESC3
CASE (SLCT3) = "H"
  STORE 'TC MBIIRDATE3
  @ 8.0 SAY "TYPE IN DESIRED BIRTH DATE DESCRIPTION;"
  (226183,0183,83, ETC.)"
  @ 8.30 GET MBIIRDATE3 PICTURE 'XXXXX'
END
STORE MBIIRDATE3 TC DESC3
CASE (SLCT3) = "I"
  STORE 'TC EWTCNIL3
  @ 8.0 SAY "TYPE IN DESIRED WEIGHT CONTROL DESCRIPTION;"
  (XX)"
  @ 8.55 GET EWTCNIL3 PICTURE 'XXX'
END
STORE EWTCNIL3 TC DESC3
CASE (SLCT3) = "J"
  STORE 'TC MPFTCNE3
  @ 8.0 SAY "TYPE IN DESIRED PFT ONE DESCRIPTION;"
  (S1,2,3,F,*)"
  @ 8.60 GET MPFTCNE3 PICTURE 'X'
END
STORE MPFTCNE3 TC DESC3
CASE (SLCT3) = "K"
  STORE 'TC MPFTTWO3
  @ 8.0 SAY "TYPE IN DESIRED PFT TWO DESCRIPTION;"
  (S1,2,3,F,*)"
  @ 8.55 GET MPFTTWO3 PICTURE 'X'
END
STORE MPFTTWO3 TC DESC3
CASE (SLCT3) = "L"
  STORE 'TC MSWMCAL3
  @ 8.0 SAY "TYPE IN DESIRED SWIMMING QUALIFICATION;"
  DESCRIPTION"
  @ 9.0 SAY "S1,S2,S3,UN,**"
  @ 9.17 GET MSWMCAL3 PICTURE 'XX'
END
STORE MSWMCAL3 TC DESC3
CASE (SLCT3) = "M"
  STORE 'TC MRFIPQAL3
  @ 8.0 SAY "TYPE IN DESIRED RIFLE QUALIFICATION;"
  DESCRIPTION"
  @ 9.0 SAY "EX,SS,MM,UN,**"
  @ 9.18 GET MRFIPQAL3 PICTURE 'XX'
END
STORE MRFIPQAL3 TO DESC3
CASE (SLCT3) = "N"
  STORE 'TC MFISQAL3
  @ 8.0 SAY "TYPE IN DESIRED PISTOL QUALIFICATION;"
  DESCRIPTION"
  @ 9.0 SAY "EX,SS,MM,UN,**"
  @ 9.18 GET MFISQAL3 PICTURE 'XX'
END
STORE MFISQAL3 TO DESC3
ENDCASE
END IF
RETURN
APPENDIX G
STANDARD COMMAND GENERATOR MODULE LISTINGS

1. STANDARD REPORT ROUTINE

* RUTIINE NAME: STDPRT.PRG
* MODULE NAME: STANDARD REPORT GENERATOR
* VERSION: 5.0
* AUTHOR: B.E. PRUELL
* DATE: 6NOV83
* VARIABLES USED: NONE
* VARIABLES MODIFIED: NONE
* VARIABLES CREATED: AASTDOPT, AADONE, PROCEED
* VARIABLES RELEASED: AASTDOPT, AADONE, ALL EXCEPT AA?????
* FILES OPENED/CLOSED: NONE
* TEMP FILES CREATED: NONE
* USING SUBROUTINES: COMMAND DETERMINER
* DESCRIPTION: GENERATES THE PRIMARY MENU FOR STANDARD REPORT SELECTION AND STORES THE USER'S RESPONSE IN
* AASTDOPT. CONTROL IS THEN PASSED TO THE APPROPRIATE
* SUBROUTINE TO OBTAIN THE NECESSARY DATA FROM THE
* DATABASE. THIS ROUTINE WILL CONTINUE UNTIL THE USER
* RESIGNS WITH A "Q".

STORE P TO AADONE
DO WHILE NOT .AADONE
ERASE

2. PRIMARY MENU FOR STANDARD REPORT SELECTION
   3. SAY "STANDARD REPORT SELECTION"
   4. SAY "SELECT A REPORT BY ENTERING THE APPROPRIATE:
      ENTER "A" FOR "A" UNIT ROSTER
      ENTER "E" FOR "E" TRAINING STATUS REPORT
      ENTER "C" FOR "C" TRAINING ROSTER REPORT
      ENTER "D" FOR "D" INDIVIDUAL TRAINING RECORD
      ENTER "M" FOR "M" MARSHMANSHIP STATUS REPORT
      ENTER "H" FOR "H" MOS ROSTER REPORT
      ENTER "I" FOR "I" PERSONAL DATA REPORT
      ENTER "J" FOR "J" EMT STATUS REPORT
      ENTER "K" FOR "K" RCSTEP REPORT
      ENTER "N" FOR "N" SWIM QUALIFICATION REPORT
      ENTER "Z" FOR "Z" HELP
      ENTER "Q" FOR "Q" QUIT
   5. ACCEPT "SELECT OPTION ========= TO AASTDOPT"

DO CASE
CASE !(AASTDOPT) = 'A'
   DC SLCUNIT
   DC UNITROST
   ACCEPT "PRESS 'ENTER' KEY TO CONTINUE" TO PROCEED
CASE !(AASTDOPT) = 'E'
   DC SLCUNIT
   DO TRNGSTAT
CASE !(AASTDOPT) = 'C'
   DC SLCUNIT
   DO TRNGCST
3 22,0
ACCEPT "PRESS 'ENTER' KEY TO CONTINUE" TO PROCEED

CASE 1(AASTLOPT) = 'E'
  ERASE
  DC VIEWITR
CASE 1(AASTLOPT) = 'E'
  DO SLCUNIT
  DC B:MKSTAT
CASE 1(AASTLOPT) = 'F'
  DC SLCUNIT
  DO MOSRST
CASE 1(AASTLOPT) = 'G'
  DO SLCUNIT
  DO MOSRST
  @ 22,0
  ACCEPT "PRESS 'ENTER' KEY TO CONTINUE" TO PROCEED
CASE 1(AASTLOPT) = 'H'
  DO SLCUNIT
  DO MOSRST
  @ 22,0
  ACCEPT "PRESS 'ENTER' KEY TO CONTINUE" TO PROCEED
CASE 1(AASTLOPT) = 'I'
  DO SLCUNIT
  DC PESDATA
  @ 22,0
  ACCEPT "PRESS 'ENTER' KEY TO CONTINUE" TO PROCEED
CASE 1(AASTLOPT) = 'J'
  DO SLCUNIT
  DC ESISTAT
CASE 1(AASTLOPT) = 'K'
  DO SLCUNIT
  DC ESISTAT
  @ 22,0
  ACCEPT "PRESS 'ENTER' KEY TO CONTINUE" TO PROCEED
CASE 1(AASTLOPT) = 'L'
  DO SLCUNIT
  DC ESISTAT
CASE 1(AASTLOPT) = 'M'
  DO A:FILE
ENDCASE

IF 1(AASTDCPT) = '0'
  STORE 1 TO AADONE
ENDIF

ENDDC
RELEAE AADONE, AASTDCPT
RELEAE ALL EXCEPT A?? ?
USE
RETURN
B. SELECT UNIT ROUTINE

* ROUTINE NAME: SLCTUNIT.PRG
* MODULE NAME: STANDARD REPORT GENERATOR
* VERSION: 5.1.0
* AUTHOR: R.L. PRUITT
* DATE: 6NOV83
* VARIABLES USED: ONECO, TWOCO, THREECO, FOURCO, ONEELT,
  * TWCELT, THREEPIT, FOURPLT
* VARIABLES MODIFIED: NONE
* VARIABLES CREATED: MUNIT, MPLAT
* VARIABLES RELEASED: NONE
* FILES OPENED/CLOSED: NONE
* TEMP FILES CREATED: NONE
* USING PROGRAMS: STERT.PRG (5.0)
* DESCRIPTION: THIS ROUTINE PROMPTS THE USER TO SELECT
  * THE SIZE AND NAME OF THE UNIT (I.E., BATTALION, COMPANY,
  * PLT DESIGNATION). THIS ROUTINE WILL EXECUTE ONCE EACH
  * TIME CALLED.
STORE F TO SLCTDONE
STORE T TO A:UNITMEM ADDITIVE
DO WHILE .NOT. SLCTDONE
ERASE
6,5 SAY "SELECT UNIT"
8,9 SAY "A..."
10,9 SAY "B..."
10,9 SAY "C..."
12,9 SAY "D..."
14,5 SAY "E...BATTALION"
17,5 SAY "H...
18,5 SAY "I...
ACCEPT "SELECT OPTION =========> " TO MUNIT
IF ! (MUNIT) = 'A'.OR. ! (MUNIT) = 'B' .OR. ! (MUNIT) = 'C'.OR.;
  ! (MUNIT) = 'D'.OR. ! (MUNIT) = 'E'
STORE T TO SLCTDONE
ENDIF
ENDDO

STORE F TO SLCTDONE
DO WHILE .NOT. SLCTDONE.AND. !(MUNIT)<>'E'
IF ! (MUNIT) = 'A'.OR. ! (MUNIT) = 'B'.OR. ! (MUNIT) = 'C';
  OR. ! (MUNIT) = 'D'
ERASE
10,5 SAY "SELECT SUBUNIT SIZE"
12,9 SAY "A..."
12,9 SAY "B..."
14,5 SAY "C..."
14,5 SAY "D..."
16,5 SAY "E...BATTALION"
17,5 SAY "F...COMPANY"
18,5 SAY "G...PLT"
18,5 SAY "H...ALL OF UNIT"
20,5 SAY "I..." 
21,5 SAY "J..."
ACCEPT "SELECT OPTION =========> " TO MPLAT
IF ! (MPLAT) = 'A'.OR. ! (MPLAT) = 'B'.OR. ! (MPLAT) = 'C'.OR.;
  !(MPLAT) = 'D'.OR. !(MPLAT) = 'E'
STORE T TO SLCTDONE

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

* ROUTINE NAME: UNITCST.PRG
* MODULE NAME: STANDARD REPORT GENERATOR
* VERSION: 5.2.0
* AUTHOR: R. E. FROUET
* DATE: 6NOV83
* VARIABLES USED: MUNIT, MPLAT, ONECO, TWOCO, THREECO,
  FOURCO, CNEPLT, TWOPLT, THREEPLT, FOURPLT
* VARIABLES MODIFIED: NONE
* VARIABLES CREATED: NONE
* VARIABLES RELEASED: MUNIT, MELAT
* FILES OPENED/CLOSED: A:FEES INDEX A:ALPHERS
* 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 SELECT MENU (5.1.0).

USE A:FEES INDEX A:ALPHERS
DO CASE
  CASE ! (MUNIT) = 'E'
    REPORT FORM B:FCSTRPT
    CASE ! (MUNIT) = 'A'.AND. ! (MPLAT) = 'E'
      REPORT FORM B:FCSTRPT FOR COMPANY = ONECO
      FLATCN = ONEPLT
    CASE ! (MUNIT) = 'B'.AND. ! (MPLAT) = 'E'
      REPORT FORM B:FCSTRPT FOR COMPANY = ONECO .AND.
       FLATCN = TWOPLT
    CASE ! (MUNIT) = 'C'.AND. ! (MPLAT) = 'E'
      REPORT FORM B:FCSTRPT FOR COMPANY = ONECO .AND.
       FLATCN = THREEPLT
    CASE ! (MUNIT) = 'D'.AND. ! (MPLAT) = 'E'
      REPORT FORM B:FCSTRPT FOR COMPANY = ONECO .AND.
       FLATCN = FOURPLT
  CASE ! (MUNIT) = 'E'.AND. ! (MELAT) = 'E'
      REPORT FORM B:FCSTRPT FOR COMPANY = TWOCO
      FLATCN = CNEPLT
    CASE ! (MUNIT) = 'F'.AND. ! (MELAT) = 'E'
      REPORT FORM B:FCSTRPT FOR COMPANY = TWOCO .AND.
       FLATCN = ONEPLT
    CASE ! (MUNIT) = 'G'.AND. ! (MELAT) = 'E'
      REPORT FORM B:FCSTRPT FOR COMPANY = TWOCO .AND.
       FLATCN = TWOPLT
    CASE ! (MUNIT) = 'H'.AND. ! (MELAT) = 'E'
      REPORT FORM B:FCSTRPT FOR COMPANY = TWOCO .AND.
       FLATCN = THREEPLT
    CASE ! (MUNIT) = 'I'.AND. ! (MELAT) = 'E'
      REPORT FORM B:FCSTRPT FOR COMPANY = TWOCO .AND.
       FLATCN = FOURPLT
  CASE ! (MUNIT) = 'C'.AND. ! (MPLAT) = 'C'
    REPORT FORM B:FCSTRPT FOR COMPANY = THREECO
    CASE ! (MUNIT) = 'A'.AND. ! (MELAT) = 'C'
      REPORT FORM B:FCSTRPT FOR COMPANY = THREECO .AND.
       FLATCN = ONEPLT
    CASE ! (MUNIT) = 'B'.AND. ! (MELAT) = 'C'
      REPORT FORM B:FCSTRPT FOR COMPANY = THREECO .AND.
       FLATCN = TWOPLT
    CASE ! (MUNIT) = 'D'.AND. ! (MELAT) = 'C'
      REPORT FORM B:FCSTRPT FOR COMPANY = THREECO .AND.
       FLATCN = THREEPLT
    CASE ! (MUNIT) = 'E'.AND. ! (MELAT) = 'C'
      REPORT FORM B:FCSTRPT FOR COMPANY = THREECO .AND.
       FLATCN = FOURPLT
  CASE ! (MUNIT) = 'F'.AND. ! (MELAT) = 'C'
    REPORT FORM B:FCSTRPT FOR COMPANY = FOURCO
CASE 1 (MUNIT) = 'E' AND MELAT = 'A'
  EFFORT FORM B: FOSTRET FOR COMPANY = FOURCO AND;
  FLATCCN = TWOPLIT
CASE 1 (MUNIT) = 'E' AND MELAT = 'B'
  EFFORT FORM B: FOSTRET FOR COMPANY = FOURCO AND;
  FLATCCN = THREET.
CASE 1 (MUNIT) = 'E' AND MELAT = 'C'
  EFFORT FORM B: FOSTRET FOR COMPANY = FOURCO AND;
  FLATCCN = FOURPT.
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. PRUETT
* DATE: 22DEC83
* VARIABLES USED: MUNIT, MPLAT, ZZACOC, ZZTOTAL, ZZBCOC,
  ZZCCOC, ZZAHIS, ZZCHIS, ZZACOD, ZZBCOD, ZZCCOD,
  ZZAIID, ZZBIID, ZZCIID, ZZAAID, ZZBAID, ZZCAID, ZZIAID,
  ZZJAIID, ZZPAID, ZZGAID, ZZGAID, ZZJAIID, ZZIAID, ZZKAID,
  ZZDUNI, ZZEUIN, ZZREUNI, ZZCUNI, ZZANBC, ZZBNBC, ZZCNBC,
  ZZDNBC, ZZENBC, ZZFBBC, ZZAMRS, ZZBMRS, ZZCMRS, ZZEMRS,
  ZZERMRS, ZZPMRS, ZZATAC, ZZBATAC, ZZCTAC, ZZDTAC, ZZETAC,
  ZZFT1, ZZFT2, ZZSWIM, ZZTFL, ZZPST, ZZCAPST, ZZNASRFL
* VARIABLES MODIFIED: NO GLOBAL VARIABLES MODIFIED
* VARIABLES RELEASED: ZZ??????
* FILES (OPENED/CLOSED): EST, PERS, QUAL (ALL FILES CLOSED)
* TEMP FILES CREATED: TEMP
* USING SUBROUTINES: STDRPT.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" CONSIST OF THREE TRAINING
  ELEMENTS (A, B, C). A PERCENTAGE COMPLETED FIGURE IS
  CALCULATED FOR EACH TRAINING ELEMENT BY DIVIDING THE
  TOTAL ELEMENTS COMPLETED (EX. ZZAHIS, ZZCHIS,
  ZZCCIS) 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 I(MUNIT) = 'E'
    USE B:EST
    DO CNTRNG
  CASE I(MUNIT) = 'A' AND. I(MPLAT) = 'F'
    SELECT PRIMARY
    USE A:PERS
    SELECT SECONDARY
    USE B:EST
    SELECT PRIMARY
    JOIN TO TEMP FCF (P.COMPANY = ONECO .AND. P.PSN=S.SSN)
    FIELDS COC, HIS, COD, INT, AID, UNI, CMRS, PFT1CLS, PFT2CLS, NBC, HRS, TAC
    USE TEMP
    DO CNTRNG
    DELETE FILE TEMP
  CASE I(MUNIT) = 'A' AND. I(MPLAT) = 'B'
    SELECT PRIMARY
    USE A:PERS
    SELECT SECONDARY
    USE B:EST
    SELECT PRIMARY
    JOIN TO TEMP FCF (P.COMPANY = ONECO .AND. P.PLTCON =
    FIELDS COC, HIS, COD, INT, AID, UNI, PFT1CLS
    USE TEMP
    DO CNTRNG
    DELETE FILE TEMP
  CASE I(MUNIT) = 'A' AND. I(MPLAT) = 'B'
    SELECT PRIMARY
    USE A:PERS
    SELECT SECONDARY
    USE B:EST

115
SELECT PRIMARY
JOIN TO TEMP FCR (P.COMPANY = ONECO .AND. P.PLATOCN =;
T.CFIL1 .AND. P.PSN=S.SSN) FIELDS;
COC,HIS,COL,INT,AID,UNI,PFT1CLSS,
PFT2CLSS, NBC, MFS, TAC
USE TEMP
DO CNTNNG
DELETE FILE TEMF
CASE I'MUNIT') = 'A'.AND. (!MPLAT) = 'C'
SELECT PRIMARY
USE APERS
SELECT SECCNDAFY
USE B:EST
SELECT PRIMARY
JOIN TO TEMP FCR (P.COMPANY = ONECO .AND. P.PLATOCN =;
T.CFIL1 .AND. P.PSN=S.SSN) FIELDS;
COC,HIS,COL,INT,AID,UNI,PFT1CLSS,
PFT2CLSS, NBC, MFS, TAC
USE TEMP
DO CNTNNG
DELETE FILE TEMF
CASE I'MUNIT') = 'A'.AND. (!MPLAT) = 'D'
SELECT PRIMARY
USE APERS
SELECT SECCNDAFY
USE B:EST
SELECT PRIMARY
JOIN TO TEMP FCR (P.COMPANY = ONECO .AND. P.PLATOCN =;
T.CFIL1 .AND. P.PSN=S.SSN) FIELDS;
COC,HIS,COL,INT,AID,UNI,PFT1CLSS,
PFT2CLSS, NBC, MFS, TAC
USE TEMP
DO CNTNNG
DELETE FILE TEMF
CASE I'MUNIT') = 'A'.AND. (!MPLAT) = 'E'
SELECT PRIMARY
USE APERS
SELECT SECCNDAFY
USE B:EST
SELECT PRIMARY
JOIN TO TEMP FCR (P.COMPANY = ONECO .AND. P.PLATOCN =;
T.CFIL1 .AND. P.PSN=S.SSN) FIELDS;
COC,HIS,COL,INT,AID,UNI,PFT1CLSS,
PFT2CLSS, NBC, MFS, TAC
USE TEMP
DO CNTNNG
DELETE FILE TEMF
CASE I'MUNIT') = 'A'.AND. (!MPLAT) = 'F'
SELECT PRIMARY
USE APERS
SELECT SECCNDAFY
USE B:EST
SELECT PRIMARY
JOIN TO TEMP FCR (P.COMPANY = ONECO .AND. P.PLATOCN =;
T.CFIL1 .AND. P.PSN=S.SSN) FIELDS;
COC,HIS,COL,INT,AID,UNI,PFT1CLSS,
PFT2CLSS, NBC, MFS, TAC
USE TEMP
DO CNTNNG
DELETE FILE TEMF
CASE I'MUNIT') = 'A'.AND. (!MPLAT) = 'G'
SELECT PRIMARY
USE APERS
SELECT SECCNDAFY
USE B:EST
SELECT PRIMARY
JOIN TO TEMP FCR (P.COMPANY = ONECO .AND. P.PLATOCN =;
T.CFIL1 .AND. P.PSN=S.SSN) FIELDS;
COC,HIS,COL,INT,AID,UNI,PFT1CLSS,
PFT2CLSS, NBC, MFS, TAC
USE TEMP
DO CNTNNG
DELETE FILE TEMF
DELETE FILE TEMP

CASE !(MUNITI) = 'E' .AND. !(MPLAT) = 'C'
USE A:PERS
SELECT PRIMARY
USE E:EST
SELECT PRIMARY
JOIN TO TEMP FCR (P.COMPANY = TWOCO .AND. P.PLATCON =;
THREEPLT .AND. P.SSN=S.SSN) FIELDS COC,HIS,COD,;
INT,AID,UNI,P1TCLASS,PFT2CLASS,NBC,MARK,TAC
USE TEMP
DO CNTRNG
DELETE FILE TEMP

CASE !(MUNIT) = 'E' .AND. !(MPLAT) = 'D'
SELECT PRIMARY
USE A:PERS
SELECT SECONDARY
USE E:EST
SELECT PRIMARY
JOIN TO TEMP FCR (P.COMPANY = TWOCO .AND. P.PLATCON =;
FOURPLT .AND. P.SSN=S.SSN) FIELDS COC,HIS,COD,;
INT,AID,UNI,P1TCLASS,PFT2CLASS,NBC,MARK,TAC
USE TEMP
DO CNTRNG
DELETE FILE TEMP

CASE !(MUNIT) = 'E' .AND. !(MPLAT) = 'E'
SELECT PRIMARY
USE A:PERS
SELECT SECONDARY
USE E:EST
SELECT PRIMARY
JOIN TO TEMP FCR (P.COMPANY = THREECO .AND. P.PLATCON =;
FOURPLT .AND. P.SSN=S.SSN) FIELDS COC,HIS,COD,;
INT,AID,UNI,P1TCLASS,PFT2CLASS,NBC,MARK,TAC
USE TEMP
DO CNTRNG
DELETE FILE TEMP

CASE !(MUNIT) = 'E' .AND. !(MPLAT) = 'A'
SELECT PRIMARY
USE A:PERS
SELECT SECONDARY
USE E:EST
SELECT PRIMARY
JOIN TO TEMP FCR (P.COMPANY = THREECO .AND. P.PLATCON =;
FOURPLT .AND. P.SSN=S.SSN) FIELDS COC,HIS,COD,;
INT,AID,UNI,P1TCLASS,PFT2CLASS,NBC,MARK,TAC
USE TEMP
DO CNTRNG
DELETE FILE TEMP

CASE !(MUNIT) = 'E' .AND. !(MPLAT) = 'B'
SELECT PRIMARY
USE A:PERS
SELECT SECONDARY
USE E:EST
SELECT PRIMARY
JOIN TO TEMP FCR (P.COMPANY = THREECO .AND. P.PLATCON =;
FOURPLT .AND. P.SSN=S.SSN) FIELDS COC,HIS,COD,;
INT,AID,UNI,P1TCLASS,PFT2CLASS,NBC,MARK,TAC
USE TEMP
DO CNTRNG
DELETE FILE TEMP

CASE !(MUNIT) = 'E' .AND. !(MPLAT) = 'C'
SELECT PRIMARY
USE A:PERS
SELECT SECONDARY
USE E:EST
SELECT PRIMARY
JOIN TO TEMP FCR (P.COMPANY = THREECO .AND. P.PLATCON =;
FOURPLT .AND. P.SSN=S.SSN) FIELDS COC,HIS,COD,;
INT,AID,UNI,P1TCLASS,PFT2CLASS,NBC,MARK,TAC
USE TEMP
DO CNTRNG
DELETE FILE TEMP

CASE !(MUNIT) = 'E' .AND. !(MPLAT) = 'C'
SELECT PRIMARY
USE A:PERS
SELECT SECONDARY
USE E:EST
SELECT PRIMARY
JOIN TO TEMP FCR (P.COMPANY = THREECO .AND. P.PLATCON =;
FOURPLT .AND. P.SSN=S.SSN) FIELDS COC,HIS,COD,;
CASE I (MUNIT) = 'C' .AND. !(MPLAT) = 'D'
SELECT PRIMARY
USE A:PEPS
SELECT SECONDARY
USE B:EST
SELECT PRIMARY
JOIN TO TEMP PCR (P.COMPANY = THREECO .AND. P.PLATCON;
= FOURPLT .AND. F.SSN=S.SSN) FIELDS COC, HIS, CCD;
INT.AID, UNI, PFT1CLASS, PFT2CLASS, NBC, MKS, TAC
ISE TEMP
DELEITE PILE TEMP

CASE I (MUNIT) = 'D' .AND. !(MPLAT) = 'E'
SELECT PRIMARY
USE A:PEPS
SELECT SECONDARY
USE B:EST
SELECT PRIMARY
JOIN TO TEMP PCR (P.COMPANY = FOURCO .AND.;
= CHEEPLT .AND. F.SSN=S.SSN) FIELDS COC, HIS, CCD;
INT.AID, UNI, PFT1CLASS, PFT2CLASS, NBC, MKS, TAC
ISE TEMP
DELEITE PILE TEMP

CASE I (MUNIT) = 'E' .AND. !(MPLAT) = 'A'
SELECT PRIMARY
USE A:PEPS
SELECT SECONDARY
USE B:EST
SELECT PRIMARY
JOIN TO TEMP PCR (P.COMPANY = FOURCO .AND. P.PLATCON;
= THREELT .AND. F.SSN=S.SSN) FIELDS COC, HIS, CCD;
INT.AID, UNI, PFT1CLASS, PFT2CLASS, NBC, MKS, TAC
ISE TEMP
DELEITE PILE TEMP

CASE I (MUNIT) = 'F' .AND. !(MPLAT) = 'B'
SELECT PRIMARY
USE A:PEPS
SELECT SECONDARY
USE B:EST
SELECT PRIMARY
JOIN TO TEMP PCR (P.COMPANY = FOURCO .AND. P.PLATCON;
= FOURPLT .AND. F.SSN=S.SSN) FIELDS COC, HIS, CCD;
INT.AID, UNI, PFT1CLASS, PFT2CLASS, NBC, MKS, TAC
ISE TEMP
DELEITE PILE TEMP

CASE I (MUNIT) = 'G' .AND. !(MPLAT) = 'C'
SELECT PRIMARY
USE A:PEPS
SELECT SECONDARY
USE B:EST
SELECT PRIMARY
JOIN TO TEMP PCR (P.COMPANY = FOURCO .AND. P.PLATCON;
= CHEEPLT .AND. F.SSN=S.SSN) FIELDS COC, HIS, CCD;
PFT1CLASS, NBC, MKS, TAC
ISE TEMP
DELEITE PILE TEMP

CASE I (MUNIT) = 'H' .AND. !(MPLAT) = 'D'
SELECT PRIMARY
USE A:PEPS
SELECT SECONDARY

118
USE TEST
SELECT PRIMARY
JOIN TO TEMP PCS (P.CMPANY =FOURCO AND P.FLATCCN;
= FOURPLT AND P.SSN=S.SSN) FIELDS;
CPC, HISC, COD, INT, AID, UNI, PFTYCLSS;
PFTYCLSS, SBC, MFS, TAC
USE TEMP
DO CASE
DELETE FILE TEMP
ENDCASE
ERASE
0,25 SAY "TRAINING STATUS REPORT"
2,25 SAY "CODE CP CC DUCT..."
2,30 SAY (ZZACOC/ZZTOTAL)*100 USING '999'
2,52 SAY ZZACOC USING '999'
3,25 SAY (ZZBCOC/ZZTOTAL)*100 USING '999'
3,30 SAY ZZBCOC USING '999'
4,25 SAY (ZZCCOC/ZZTOTAL)*100 USING '999'
4,30 SAY ZZCCOC USING '999'
6,30 SAY "MARINE COFFS HISTORY..."
6,25 SAY (ZZAHIS/ZZTOTAL)*100 USING '999'
6,30 SAY ZZAHIS USING '999'
7,25 SAY (ZZBHIS/ZZTOTAL)*100 USING '999'
7,30 SAY ZZBHIS USING '999'
8,25 SAY (ZZCHIS/ZZTOTAL)*100 USING '999'
8,30 SAY ZZCHIS USING '999'
10,30 SAY "CLOSE ORDE DRILL..."
10,25 SAY (ZZACOD/ZZTOTAL)*100 USING '999'
10,30 SAY ZZACOD USING '999'
11,25 SAY (ZZBCOD/ZZTOTAL)*100 USING '999'
11,30 SAY ZZBCOD USING '999'
12,25 SAY (ZZCCOD/ZZTOTAL)*100 USING '999'
12,30 SAY ZZCCOD USING '999'
14,30 SAY "INTERIOR GUARD..."
14,25 SAY (ZZAINT/ZZTOTAL)*100 USING '999'
14,30 SAY ZZAINT USING '999'
15,25 SAY (ZZBINT/ZZTOTAL)*100 USING '999'
15,30 SAY ZZBINT USING '999'
16,25 SAY (ZZCINT/ZZTOTAL)*100 USING '999'
16,30 SAY ZZCINT USING '999'
18,30 SAY "FIRST AID..."
18,25 SAY (ZZAIAID/ZZTOTAL)*100 USING '999'
18,30 SAY ZZAIAID USING '999'
20,25 SAY (ZZBIAID/ZZTOTAL)*100 USING '999'
20,30 SAY ZZBIAID USING '999'
22,54 SAY "PERCENT COMPLETED B..."
22,76 SAY ZZBIAID USING '999'
24,25 SAY (ZZCIAID/ZZTOTAL)*100 USING '999'
24,30 SAY ZZCIAID USING '999'
26,54 SAY "PERCENT COMPLETED C..."
26,76 SAY ZZCIAID USING '999'
28,25 SAY (ZZDIAID/ZZTOTAL)*100 USING '999'
28,30 SAY ZZDIAID USING '999'
30,54 SAY "PERCENT COMPLETED D..."
30,76 SAY ZZDIAID USING '999'
32,25 SAY (ZZEIAID/ZZTOTAL)*100 USING '999'
32,30 SAY ZZEIAID USING '999'
34,25 SAY (ZZFIAID/ZZTOTAL)*100 USING '999'
34,30 SAY ZZFIAID USING '999'
36,54 SAY "PERCENT COMPLETED E..."
36,76 SAY ZZFIAID USING '999'
SAY "PERCENT COMPLETED F..."
SAY ZZFAID USING '999'
SAY (ZZGFAID/ZZTOTAL) *100 USING '999'
SAY "PERCENT COMPLETED G..."
SAY ZZGFAID USING '999'
SAY (ZZHFAID/ZZTOTAL) *100 USING '999'
SAY "PERCENT COMPLETED H..."
SAY ZZHFAID USING '999'
SAY (ZZHFAID/ZZTOTAL) *100 USING '999'
SAY "PERCENT COMPLETED I..."
SAY ZZIHAID USING '999'
SAY (ZZJHAID/ZZTOTAL) *100 USING '999'
SAY "PERCENT COMPLETED J..."
SAY ZZJHAID USING '999'
SAY (ZZJHAID/ZZTOTAL) *100 USING '999'
SAY "PERCENT COMPLETED K..."
SAY ZZKHAID USING '999'
SAY (ZZLHAID/ZZTOTAL) *100 USING '999'
SAY "PERCENT COMPLETED L..."
SAY ZZLHAID USING '999'
SAY (ZZMHAID/ZZTOTAL) *100 USING '999'
SAY "PERCENT COMPLETED M..."
SAY ZZMHAID USING '999'
SAY (ZZNHAID/ZZTOTAL) *100 USING '999'
SAY "PERCENT COMPLETED N..."
SAY ZZNHAID USING '999'
SAY (ZZPHAID/ZZTOTAL) *100 USING '999'
SAY "PERCENT COMPLETED P..."
SAY ZZPHAID USING '999'
SAY (ZZQHAID/ZZTOTAL) *100 USING '999'
SAY "PERCENT COMPLETED Q..."
SAY ZZQHAID USING '999'
SAY (ZZRHAID/ZZTOTAL) *100 USING '999'
SAY "PERCENT COMPLETED R..."
SAY ZZRHAID USING '999'
SAY (ZZSHAID/ZZTOTAL) *100 USING '999'
SAY "PERCENT COMPLETED S..."
SAY ZZSHAID USING '999'
SAY (ZZTHAID/ZZTOTAL) *100 USING '999'
SAY "PERCENT COMPLETED T..."
SAY ZZTHAID USING '999'
SAY (ZZUHAID/ZZTOTAL) *100 USING '999'
SAY "PERCENT COMPLETED U..."
SAY ZZUHAID USING '999'
SAY (ZZVHAID/ZZTOTAL) *100 USING '999'
SAY "PERCENT COMPLETED V..."
SAY ZZVHAID USING '999'
SAY (ZZWAID/ZZTOTAL) *100 USING '999'
SAY "PERCENT COMPLETED W..."
SAY ZZWAID USING '999'
SAY (ZZXHAID/ZZTOTAL) *100 USING '999'
SAY "PERCENT COMPLETED X..."
SAY ZZXHAID USING '999'
SAY (ZZYHAID/ZZTOTAL) *100 USING '999'
SAY "PERCENT COMPLETED Y..."
SAY ZZYHAID USING '999'
SAY (ZZZHAID/ZZTOTAL) *100 USING '999'
SAY "PERCENT COMPLETED Z..."
SAY ZZZHAID USING '999'
SAY "TRAINING STATUS REPORT"
SAY "EQUIPMENT/UNIFORMS...
SAY (ZZAUNI/ZZTOTAL) *100 USING '999'
SAY "PERCENT COMPLETED A...
SAY ZZAUNI USING '999'
SAY (ZZBUNI/ZZTOTAL) *100 USING '999'
SAY "PERCENT COMPLETED B...
SAY ZZBUNI USING '999'
SAY (ZZCUNI/ZZTOTAL) *100 USING '999'
SAY "PERCENT COMPLETED C...
SAY ZZCUNI USING '999'
SAY (ZZDUNI/ZZTOTAL) *100 USING '999'
SAY "PERCENT COMPLETED D...
SAY ZZDUNI USING '999'
SAY (ZZEUNI/ZZTOTAL) *100 USING '999'
SAY "PERCENT COMPLETED E...
SAY ZZEUNI USING '999'
SAY (ZZFUNI/ZZTOTAL) *100 USING '999'
SAY "PERCENT COMPLETED F...
SAY ZZFUNI USING '999'
SAY (ZZGUNI/ZZTOTAL) *100 USING '999'
SAY "PERCENT COMPLETED G...
SAY ZZGUNI USING '999'
SAY (ZZHUNI/ZZTOTAL) *100 USING '999'
SAY "PERCENT COMPLETED H...
SAY ZZHUNI USING '999'
SAY (ZZIUNI/ZZTOTAL) *100 USING '999'
SAY "PERCENT COMPLETED I...
SAY ZZIUNI USING '999'
SAY (ZZJUNI/ZZTOTAL) *100 USING '999'
SAY "PERCENT COMPLETED J...
SAY ZZJUNI USING '999'
SAY (ZZKUNI/ZZTOTAL) *100 USING '999'
SAY "PERCENT COMPLETED K...
SAY ZZKUNI USING '999'
SAY (ZZLUNI/ZZTOTAL) *100 USING '999'
SAY "PERCENT COMPLETED L...
SAY ZZLUNI USING '999'
SAY (ZZMUNI/ZZTOTAL) *100 USING '999'
SAY "PERCENT COMPLETED M...
SAY ZZMUNI USING '999'
SAY (ZZNUNI/ZZTOTAL) *100 USING '999'
SAY "PERCENT COMPLETED N...
SAY ZZNUNI USING '999'
SAY (ZZPUNI/ZZTOTAL) *100 USING '999'
SAY "PERCENT COMPLETED P...
SAY ZZPUNI USING '999'
SAY (ZZQUNI/ZZTOTAL) *100 USING '999'
SAY "PERCENT COMPLETED Q...
SAY ZZQUNI USING '999'
SAY (ZZRUNI/ZZTOTAL) *100 USING '999'
SAY "PERCENT COMPLETED R...
SAY ZZRUNI USING '999'
SAY (ZZSUNI/ZZTOTAL) *100 USING '999'
SAY "PERCENT COMPLETED S...
SAY ZZSUNI USING '999'
SAY (ZZTUNI/ZZTOTAL) *100 USING '999'
SAY "PERCENT COMPLETED T...
SAY ZZTUNI USING '999'
SAY (ZZUNIT/ZZTOTAL) *100 USING '999'
SAY "PERCENT COMPLETED U...
SAY ZZUNIT USING '999'
SAY (ZZVUNI/ZZTOTAL) *100 USING '999'
SAY "PERCENT COMPLETED V...
SAY ZZVUNI USING '999'
SAY (ZZWUNI/ZZTOTAL) *100 USING '999'
SAY "PERCENT COMPLETED W...
SAY ZZWUNI USING '999'
SAY (ZZXUNI/ZZTOTAL) *100 USING '999'
SAY "PERCENT COMPLETED X...
SAY ZZXUNI USING '999'
SAY (ZZYUNI/ZZTOTAL) *100 USING '999'
SAY "PERCENT COMPLETED Y...
SAY ZZYUNI USING '999'
SAY (ZZZUNI/ZZTOTAL) *100 USING '999'
SAY "PERCENT COMPLETED Z...
SAY ZZZUNI USING '999'
ACCEPT "PRESS 'ENTER' TO CONTINUE" TO PROCEED
ERASE
2.0 SAY "TRAINING STATUS REPORT"
2.30 SAY "PERCENT COMPLETED A..."
2.30 SAY ZZATAC USING '999'
3.25 SAY ZZATAC USING '999'
3.25 SAY PERCENT COMPLETED E...
3.25 SAY ZZATAC USING '999'
4.25 SAY ZZATAC USING '999'
4.25 SAY PERCENT COMPLETED C...
4.25 SAY ZZATAC USING '999'
5.25 SAY ZZATAC USING '999'
5.25 SAY PERCENT COMPLETED D...
5.25 SAY ZZATAC USING '999'
6.25 SAY ZZATAC USING '999'
6.25 SAY PERCENT COMPLETED E...
6.25 SAY ZZATAC USING '999'
6.52 SAY ZZATAC USING '999'
8.0 SAY "FPT1..."
8.24 SAY ((ZZTOTAL-ZZPFT1)/ZZTOTAL)*100 USING '999'
8.6 SAY PERCENT COMPLETED FPT ONE...
10.24 SAY ((ZZTOTAL-ZZPFT2)/ZZTOTAL)*100 USING '999'
10.62 SAY PERCENT COMPLETED FPT TWO...
RESTORE FROM ZZTEMP
DO CASHE
CASE (MUNIT) = 'E'
USE B:QUAL
GO CNTQUAL
CASE !(MUNIT) = 'A'.AND. !(MPLAT) = 'E'
SELECT PRIMARY
USE A:PERS
SELECT SECONDARY
USE B:QUAL
SELECT PRIMARY
JOIN TO TEMP PCR (P.COMPANY =ONECO .AND.:
P.SSN=S.SSN) FIELDS SSN,SWIMQUAL, RFLQUAL,PSIQUAL
USE TEMP
GO CNTQUAL
DELETE FILE TEMP
CASE !(MUNIT) = 'A'.AND. !(MPLAT) = 'A'
SELECT PRIMARY
USE A:PERS
SELECT SECONDARY
USE B:QUAL
SELECT PRIMARY
JOIN TO TEMP PCR (P.COMPANY =ONECO .AND. P.PLATCN =:
CNEEL .AND. P.SSN=S.SSN) FIELDS;
SSN,SWIMQUAL, RFLQUAL,PSIQUAL
USE TEMP
GO CNTQUAL
DELETE FILE TEMP
CASE !(MUNIT) = 'A'.AND. !(MPLAT) = 'B'
SELECT PRIMARY
USE A:PERS
SELECT SECONDARY
USE B:QUAL
SELECT PRIMARY
JOIN TO TEMP PCR (P.COMPANY =ONECO .AND. P.PLATCN =:
TWCFT .AND. P.SSN=S.SSN) FIELDS;
SSN,SWIMQUAL, RFLQUAL,PSIQUAL
USE TEMP
GO CNTQUAL
DELETE FILE TEMP
CASE !(MUNIT) = 'A'.AND. !(MPLAT) = 'C'
SELECT PRIMARY
121
USE A:PERS
SELECT SECONDARY
USE E:QUAL
SELECT PRIMARY
JOIN TO TEMP PCIe (P.COMPANY =ONECO .AND. P.PLATCCN =;
THREELET .AND. P.SSN=S.SSN) FIELDS;
SSN,SWIMQUAL,RFLQUAL,PSTQUAL
USE TEMP
DO CNTQUAL
DELETE FILE TEMP
CASE I(MUNIT) = "A" .AND. !(MPLAT) = "D"
SELECT PRIMARY
USE A:PERS
SELECT SECONDARY
USE E:QUAL
SELECT PRIMARY
JOIN TO TEMP PCIe (P.COMPANY =ONECO .AND. P.PLATCCN =;
THREELET .AND. P.SSN=S.SSN) FIELDS;
SSN,SWIMQUAL,RFLQUAL,PSTQUAL
USE TEMP
DO CNTQUAL
DELETE FILE TEMP
CASE I(MUNIT) = "B" .AND. !(MPLAT) = "D"
SELECT PRIMARY
USE A:PERS
SELECT SECONDARY
USE E:QUAL
SELECT PRIMARY
JOIN TO TEMP PCIe (P.COMPANY =ONECO .AND. P.PLATCCN =;
THREELET .AND. P.SSN=S.SSN) FIELDS;
SSN,SWIMQUAL,RFLQUAL,PSTQUAL
USE TEMP
DO CNTQUAL
DELETE FILE TEMP
CASE I(MUNIT) = "C" .AND. !(MPLAT) = "D"
SELECT PRIMARY
USE A:PERS
SELECT SECONDARY
USE E:QUAL
SELECT PRIMARY
JOIN TO TEMP PCIe (P.COMPANY =ONECO .AND. P.PLATCCN =;
THREELET .AND. P.SSN=S.SSN) FIELDS;
SSN,SWIMQUAL,RFLQUAL,PSTQUAL
USE TEMP
DO CNTQUAL
DELETE FILE TEMP
CASE I(MUNIT) = "D" .AND. !(MPLAT) = "D"
SELECT PRIMARY
USE A:PERS
SELECT SECONDARY
USE E:QUAL
SELECT PRIMARY
JOIN TO TEMP PCIe (P.COMPANY =ONECO .AND. P.PLATCCN =;
THREELET .AND. P.SSN=S.SSN) FIELDS;
SSN,SWIMQUAL,RFLQUAL,PSTQUAL
USE TEMP
DO CNTQUAL
DELETE FILE TEMP
CASE I(MUNIT) = "E" .AND. !(MPLAT) = "D"
SELECT PRIMARY
USE A:PERS
SELECT SECONDARY
USE E:QUAL
SELECT PRIMARY
JOIN TO TEMP PCIe (P.COMPANY =ONECO .AND. P.PLATCCN =;
THREELET .AND. P.SSN=S.SSN) FIELDS;
SSN,SWIMQUAL,RFLQUAL,PSTQUAL
USE TEMP
DO CNTQUAL
DELETE FILE TEMP
CASE I(MUNIT) = "F" .AND. !(MPLAT) = "D"
SELECT PRIMARY
USE A:PERS
SELECT SECONDARY
USE E:QUAL
SELECT PRIMARY
JOIN TO TEMP PCIe (P.COMPANY =ONECO .AND. P.PLATCCN =;
THREELET .AND. P.SSN=S.SSN) FIELDS;
SSN,SWIMQUAL,RFLQUAL,PSTQUAL
USE TEMP
DO CNTQUAL
DELETE FILE TEMP
CASE I(MUNIT) = "G" .AND. !(MPLAT) = "D"
SELECT PRIMARY
USE A:PERS
SELECT SECONDARY
USE E:QUAL
SELECT PRIMARY
JOIN TO TEMP PCIe (P.COMPANY =ONECO .AND. P.PLATCCN =;
THREELET .AND. P.SSN=S.SSN) FIELDS;
SSN,SWIMQUAL,RFLQUAL,PSTQUAL
USE TEMP
DO CNTQUAL
DELETE FILE TEMP
CASE I(MUNIT) = "H" .AND. !(MPLAT) = "D"
SELECT PRIMARY
USE A:PERS
SELECT SECONDARY
USE E:QUAL
SELECT PRIMARY
JOIN TO TEMP PCIe (P.COMPANY =ONECO .AND. P.PLATCCN =;
THREELET .AND. P.SSN=S.SSN) FIELDS;
SSN,SWIMQUAL,RFLQUAL,PSTQUAL
USE TEMP
DO CNTQUAL
DELETE FILE TEMP
CASE I(MUNIT) = "I" .AND. !(MPLAT) = "D"
SELECT PRIMARY
USE A:PERS
SELECT SECONDARY
USE E:QUAL
SELECT PRIMARY
JOIN TO TEMP PCIe (P.COMPANY =ONECO .AND. P.PLATCCN =;
THREELET .AND. P.SSN=S.SSN) FIELDS;
SSN,SWIMQUAL,RFLQUAL,PSTQUAL
USE TEMP
DO CNTQUAL
DELETE FILE TEMP
CASE I(MUNIT) = "J" .AND. !(MPLAT) = "D"
SELECT PRIMARY
USE A:PERS
SELECT SECONDARY
USE E:QUAL
SELECT PRIMARY
JOIN TO TEMP PCIe (P.COMPANY =ONECO .AND. P.PLATCCN =;
THREELET .AND. P.SSN=S.SSN) FIELDS;
SSN,SWIMQUAL,RFLQUAL,PSTQUAL
USE TEMP
DO CNTQUAL
DELETE FILE TEMP
CASE I(MUNIT) = "K" .AND. !(MPLAT) = "D"
SELECT PRIMARY
USE A:PERS
SELECT SECONDARY
USE E:QUAL
SELECT PRIMARY
JOIN TO TEMP PCIe (P.COMPANY =ONECO .AND. P.PLATCCN =;
THREELET .AND. P.SSN=S.SSN) FIELDS;
SSN,SWIMQUAL,RFLQUAL,PSTQUAL
USE TEMP
DO CNTQUAL
DELETE FILE TEMP
CASE I(MUNIT) = "L" .AND. !(MPLAT) = "D"
SELECT PRIMARY
USE A:PERS
SELECT SECONDARY
USE E:QUAL
SELECT PRIMARY
JOIN TO TEMP PCIe (P.COMPANY =ONECO .AND. P.PLATCCN =;
THREELET .AND. P.SSN=S.SSN) FIELDS;
SSN,SWIMQUAL,RFLQUAL,PSTQUAL
USE TEMP
DO CNTQUAL
DELETE FILE TEMP
CASE I(MUNIT) = "M" .AND. !(MPLAT) = "D"
SELECT PRIMARY
USE A:PERS
SELECT SECONDARY
USE E:QUAL
SELECT PRIMARY
JOIN TO TEMP PCIe (P.COMPANY =ONECO .AND. P.PLATCCN =;
THREELET .AND. P.SSN=S.SSN) FIELDS;
SSN,SWIMQUAL,RFLQUAL,PSTQUAL
USE TEMP
DO CNTQUAL
DELETE FILE TEMP
CASE I(MUNIT) = "N" .AND. !(MPLAT) = "D"
SELECT PRIMARY
USE A:PERS
SELECT SECONDARY
USE E:QUAL
SELECT PRIMARY
JOIN TO TEMP PCIe (P.COMPANY =ONECO .AND. P.PLATCCN =;
THREELET .AND. P.SSN=S.SSN) FIELDS;
SSN,SWIMQUAL,RFLQUAL,PSTQUAL
USE TEMP
DO CNTQUAL
DELETE FILE TEMP
CASE I(MUNIT) = "O" .AND. !(MPLAT) = "D"
SELECT PRIMARY
USE A:PERS
SELECT SECONDARY
USE E:QUAL
SELECT PRIMARY
JOIN TO TEMP PCIe (P.COMPANY =ONECO .AND. P.PLATCCN =;
THREELET .AND. P.SSN=S.SSN) FIELDS;
SSN,SWIMQUAL,RFLQUAL,PSTQUAL
USE TEMP
DO CNTQUAL
DELETE FILE TEMP
CASE I(MUNIT) = "P" .AND. !(MPLAT) = "D"
SELECT PRIMARY
USE A:PERS
SELECT SECONDARY
USE E:QUAL
SELECT PRIMARY
JOIN TO TEMP PCIe (P.COMPANY =ONECO .AND. P.PLATCCN =;
THREELET .AND. P.SSN=S.SSN) FIELDS;
SSN,SWIMQUAL,RFLQUAL,PSTQUAL
USE TEMP
DO CNTQUAL
DELETE FILE TEMP
CASE I(MUNIT) = "Q" .AND. !(MPLAT) = "D"
SELECT PRIMARY
USE A:PERS
SELECT SECONDARY
USE E:QUAL
SELECT PRIMARY
JOIN TO TEMP PCIe (P.COMPANY =ONECO .AND. P.PLATCCN =;
THREELET .AND. P.SSN=S.SSN) FIELDS;
SSN,SWIMQUAL,RFLQUAL,PSTQUAL
USE TEMP
DO CNTQUAL
DELETE FILE TEMP
CASE I(MUNIT) = "R" .AND. !(MPLAT) = "D"
SELECT PRIMARY
USE A:PERS
SELECT SECONDARY
USE E:QUAL
SELECT PRIMARY
JOIN TO TEMP PCIe (P.COMPANY =ONECO .AND. P.PLATCCN =;
THREELET .AND. P.SSN=S.SSN) FIELDS;
SSN,SWIMQUAL,RFLQUAL,PSTQUAL
USE TEMP
DO CNTQUAL
DELETE FILE TEMP
CASE I(MUNIT) = "S" .AND. !(MPLAT) = "D"
SELECT PRIMARY
USE A:PERS
SELECT SECONDARY
USE E:QUAL
SELECT PRIMARY
JOIN TO TEMP PCIe (P.COMPANY =ONECO .AND. P.PLATCCN =;
THREELET .AND. P.SSN=S.SSN) FIELDS;
SSN,SWIMQUAL,RFLQUAL,PSTQUAL
USE TEMP
DO CNTQUAL
DELETE FILE TEMP
CASE I(MUNIT) = "T" .AND. !(MPLAT) = "D"
SELECT PRIMARY
USE A:PERS
SELECT SECONDARY
USE E:QUAL
SELECT PRIMARY
JOIN TO TEMP PCIe (P.COMPANY =ONECO .AND. P.PLATCCN =;
THREELET .AND. P.SSN=S.SSN) FIELDS;
SSN,SWIMQUAL,RFLQUAL,PSTQUAL
USE TEMP
DO CNTQUAL
DELETE FILE TEMP
CASE (MUNIT) = 'E' .AND. !(MPLAT) = 'D'
  SELECT PRIMARY
  USE A:PEPS
  SELECT SECONDARY
  USE E:QUAL
  SELECT PRIMARY
  JOIN TO TEMP FCF (P.COMPANY = TWOECO .AND. P.PLATCON = ;
  USE E:QUAL
  JOIN TO TEMP FCF (P.COMPANY = TWOCO .AND. P.PLATCON = ;
  USE E:QUAL
  SELECT PRIMARY
  JOIN TO TEMP FCF (P.COMPANY = TWOECO .AND. P.PLATCON = ;
  USE TEMP
  DO CNTQUAL
  DELETE FILE TEMP
CASE I (MUNIT) = 'C' .AND. !(MPLAT) = 'E'
  SELECT PRIMARY
  USE A:PEPS
  SELECT SECONDARY
  USE E:QUAL
  SELECT PRIMARY
  JOIN TO TEMP FCF (P.COMPANY = THREECO .AND. ;
  USE TEMP
  DO CNTQUAL
  DELETE FILE TEMP
CASE II (MUNIT) = 'C' .AND. !(MPLAT) = 'A'
  SELECT PRIMARY
  USE A:PEPS
  SELECT SECONDARY
  USE E:QUAL
  SELECT PRIMARY
  JOIN TO TEMP FCF (P.COMPANY = THREECO .AND. ;
  USE TEMP
  DO CNTQUAL
  DELETE FILE TEMP
CASE III (MUNIT) = 'C' .AND. !(MPLAT) = 'B'
  SELECT PRIMARY
  USE A:PEPS
  SELECT SECONDARY
  USE E:QUAL
  SELECT PRIMARY
  JOIN TO TEMP FCF (P.COMPANY = THREECO .AND. ;
  USE TEMP
  DO CNTQUAL
  DELETE FILE TEMP
CASE IV (MUNIT) = 'C' .AND. !(MPLAT) = 'C'
  SELECT PRIMARY
  USE A:PEPS
  SELECT SECONDARY
  USE E:QUAL
  SELECT PRIMARY
  JOIN TO TEMP FCF (P.COMPANY = THREECO .AND. ;
  USE TEMP
  DO CNTQUAL
  DELETE FILE TEMP
CASE V (MUNIT) = 'C' .AND. !(MPLAT) = 'D'
  SELECT PRIMARY
  USE A:PEPS
  SELECT SECONDARY
  USE E:QUAL
  SELECT PRIMARY
  JOIN TO TEMP FCF (P.COMPANY = THREECO .AND. ;
  USE TEMP
  DO CNTQUAL
  DELETE FILE TEMP
DO CNTQUAL
DELETE FILE TEMP
CASE I (MUNIT) = 'D' AND (MPLAT) = 'E'
SELECT PRIMARY
USE A:PERS
SELECT SECCNDARY
USE E:QUAL
SELECT PRIMARY
JOIN TO TEMP PCR (P.CCMAPNY =FOURCO .AND. P.SSN=S.SSN) FIELDS SSN,SWIMQUAL,RFLQUAL,PSTQUAL
USE TEMP
DO CNTQUAL
DELETE FILE TEMP
CASE I (MUNIT) = 'E' AND (MPLAT) = 'A'
SELECT PRIMARY
USE A:PERS
SELECT SECCNDARY
USE E:QUAL
SELECT PRIMARY
JOIN TO TEMP PCR (P.CCMAPNY =FOURCO .AND. P.PLATCCN; = CNELT .AND. P.SSN=S.SSN) FIELDS;
SSN,SWIMQUAL,RFLQUAL,PSTQUAL
USE TEMP
DO CNTQUAL
DELETE FILE TEMP
CASE I (MUNIT) = 'E' AND (MPLAT) = 'B'
SELECT PRIMARY
USE A:PERS
SELECT SECCNDARY
USE E:QUAL
SELECT PRIMARY
JOIN TO TEMP PCR (P.CCMAPNY =FOURCO .AND. P.PLATCCN; = CNELT .AND. P.SSN=S.SSN) FIELDS;
SSN,SWIMQUAL,RFLQUAL,PSTQUAL
USE TEMP
DO CNTQUAL
DELETE FILE TEMP
CASE I (MUNIT) = 'E' AND (MPLAT) = 'C'
SELECT PRIMARY
USE A:PERS
SELECT SECCNDARY
USE E:QUAL
SELECT PRIMARY
JOIN TO TEMP PCR (P.CCMAPNY =FOURCO .AND. P.PLATCCN; = CNELT .AND. P.SSN=S.SSN) FIELDS;
SSN,SWIMQUAL,RFLQUAL,PSTQUAL
USE TEMP
DO CNTQUAL
DELETE FILE TEMP
CASE I (MUNIT) = 'E' AND (MPLAT) = 'D'
SELECT PRIMARY
USE A:PERS
SELECT SECCNDARY
USE E:QUAL
SELECT PRIMARY
JOIN TO TEMP PCR (P.CCMAPNY =FOURCO .AND. P.PLATCCN; = CNELT .AND. P.SSN=S.SSN) FIELDS;
SSN,SWIMQUAL,RFLQUAL,PSTQUAL
USE TEMP
DO CNTQUAL
DELETE FILE TEMP
ENDCASE
@ 12,6 SAY "SHIMQUAL..."
@ 12,14 SAY ((ZZTOTAL-ZZSWIM)/ZZTOTAL)*100 USING '999'
@ 12,31 SAY "PERCENT SWIM QUALIFIED..."
@ 12,60 SAY (ZZTOTAL-ZZSWIM) USING '999'
@ 14,0 SAY "Rifle Qual...
@ 14,24 SAY ((ZZTOTAL-ZZRFL-ZZRPL)/ZZTOTAL)*100 USING;
14.31 SAY "PERCENT RIFLE QUALIFIED..."
14.60 SAY (Z2TOTAL-Z2RFL-Z2NARFL) USING '999'
16.0  SAY "PISTOL QUAL..."
16.24 SAY ((Z2TOTAL-Z2EST-Z2NAPST)/Z2TOTAL) *100 USING '999'
16.31 SAY "PERCENT PISTOL QUALIFIED..."
16.60 SAY (Z2TOTAL-Z2PST-Z2NAPST) USING '999'
20.0 ACCEPT "PRESS 'ENTER' TO CONTINUE" TO PROCEED
RELEASE ALL LIKE ZZ???? ??
RETURN
I. TRAINING COUNT ROUTINE

* ROUTINE NAME: CNTTEG.PRG
* MODULE NAME: STANDARD REPORT GENERATOR
* VERSION: 5.3.4.0
* AUTHOR: R.E. PRUITT
* DATE: 6NOV83
* VARIABLES USED: ONICO, TWCCC, THREECO, FOURCO, ONEFILT, *TWOFL*THREEFL, FOURFL
* VARIABLES MODIFIED: NONE
* VARIABLES RELEASED: NONE
* FILES OPENED/CLOSED: NONE
* TEMP FILES CREATED: ZZTEMF (SAVES ALL VARIABLES UPON ENTERING THIS ROUTINE)
* FILES INTERFACED: TR.ENGSTAT.ERG
* DESCRIPTION: THIS ROUTINE COUNTS THE NUMBER OF ESSENTIAL SUBJECT TRAINING ELEMENTS THAT HAVE BEEN COMPLETED.

SAVE TO ZZTEMF
RELEASE ALL
COUNT TO ZZTOTAL
COUNT FOR 'A' (CO) TC ZZACOC
COUNT FOR 'E' (CO) TC ZZBCOC
COUNT FOR 'C' (CO) TC ZZCCOC
COUNT FOR 'H' (IS) TO ZZAHIS
COUNT FOR 'E' (IS) TO ZZBHIS
COUNT FOR 'C' (IS) TO ZZCHIS
COUNT FOR 'A' (CD) TC ZZACCD
COUNT FOR 'E' (CD) TC ZZBCCD
COUNT FOR 'C' (CD) TC ZZCCCD
COUNT FOR 'A' (INT) TO ZZAITNT
COUNT FOR 'E' (INT) TO ZZBIINT
COUNT FOR 'C' (INT) TO ZZCINT
COUNT FOR 'A' (ATD) TO ZZAAID
COUNT FOR 'E' (ATD) TO ZZBAID
COUNT FOR 'C' (ATD) TO ZZCAID
COUNT FOR 'D' (ATD) TO ZZDAID
COUNT FOR 'E' (ATD) TO ZZEAID
COUNT FOR 'P' (ATD) TO ZZFAID
COUNT FOR 'A' (AD) TO ZZAAID
COUNT FOR 'G' (AD) TO ZZGAID
COUNT FOR 'H' (AD) TO ZZHAID
COUNT FOR 'I' (AD) TO ZZIAID
COUNT FOR 'J' (AD) TO ZZJAID
COUNT FOR 'K' (AD) TO ZZKAID
COUNT FOR 'A' (UNI) TO ZZAUUNI
COUNT FOR 'E' (UNI) TO ZZBUUNI
COUNT FOR 'C' (UNI) TO ZZCUUNI
COUNT FOR 'A' (NEC) TO ZZANEC
COUNT FOR 'E' (NEC) TO ZZBNEC
COUNT FOR 'C' (NEC) TO ZZCNEC
COUNT FOR 'I' (NEC) TO ZZDNEC
COUNT FOR 'F' (NEC) TO ZZENEC
COUNT FOR 'A' (MKS) TO ZZAMKS
COUNT FOR 'E' (MKS) TO ZZBMKS
COUNT FOR 'C' (MKS) TO ZZCMKS
COUNT FOR 'D' (MKS) TO ZZDMKS
COUNT FOR 'A' (TAC) TO ZZATAC
COUNT FOR 'E' (TAC) TO ZZTAC

126
COUNT PCR 'C'$(TAC) TC ZZCTAC
COUNT FOR 'D'$(TAC) TC ZZDTAC
COUNT FOR 'E'$(TAC) TC ZZETAC
COUNT FOR 'F'$(PFT1CLASS) TO ZZPFT1
COUNT FOR 'G'$(PFT2CLASS) TO ZZPFT2
RETURN
F. QUALIFICATION COUNT ROUTINE

* ROUTINE NAME: CNTQUAL.PRG
* MODULE NAME: STANDARD REPORT GENERATOR
* VERSION: 5.3.2.0
* AUTHOR: R.E. RUIETT
* DATE: CODEC83
* VARIABLES USED: NONE
* VARIABLES MODIFIED: NONE
* VARIABLES CREATED: ZZSWIM, ZZRFL, ZZNARFL, ZZPST, ZZNAPST
* VARIABLES RELEASED: NONE
* FILES OPENED/CLOSED: NONE
* TIME FILES CREATED: NONE
* USING SUBROUTINES: TRNGSTAT.PRG
* DESCRIPTION: THIS ROUTINE FIRST SAVES ALL MEMORY
  * VARIABELES IN A MEMORY FILE (ZZTOTAL) AND THEN COUNTS THE
  * NUMBER OF PERSONS THAT HAVE NOT COMPLETED SWIMQUAL,
  * RELQUAL, ESTQUAL AND THOSE PERSONS NOT REQUIRED TO
  * QUALIFY WITH THE RIFFLE OR PISTOL (NA).
  * COUNT TO ZZTOTAL
  * COUNT FOR '**$SWIMQUAL' TO ZZSWIM
  * COUNT FOR '**$RELQUAL' TO ZZREL
  * COUNT FOR 'NA'$ESTQUAL TO ZZEST
  * RETURN
G. TRAINING ROSTER ROUTINE

* ROUTINE NAME: TRNGCST.PRG
* MODULE NAME: STANDARD REPORT GENERATOR
* VERSION: 5.4.0
* AUTHOR: B. E. FRUITZ
* DATE: 202083
* VARIABLES USED: MUNIT, MPLAT, CNECO, TWOPLT, THREEPLT, FOURPLT
* VARIABLES MODIFIED: NONE
* VARIABLES CREATED: COUNT, PROCEED
* VARIABLES RELEASED: NONE
* FILES (OPENED/CLOSED): A: PERS INDEX A: ALPHAPERS (ALL FILES CLOSING)
* TEXT FILES CREATED: NONE
* USING SUBROUTINES: SIDBPT.PRG (5.0)
* DESCRIPTIVE: This routine provides the "Training Roster" report. It starts at the top of the PERS.DBF and locks at each record individually. If a record corresponds to the selected by the user, the routine TRNGCST.PRG is called. Each collects and displays the training data that belongs to the current PERS.DBF record (e.g., if the user requests training data on every member of 1st PLAT, a Co., TRNGCST.PRG locates each record in the PERS.DBF which belong to 1st PLAT, a Co. Then, TRNGCST.PRG then displays the training data from the appropriate database files.). Information on one individual is displayed in six rows and four sets of data are displayed per screen. The user presses the enter key to scroll forward to each succeeding set of individual data.

ERASE
SAY "TRAINING ROSTER REPORT"
ACCEPT "PRESS "ENTER" KEY TO CONTINUE" TO PROCEED
ERASE
SELECT PRIMARY
USE A: PERS INDEX A: ALPHAPERS
GOTC TOP
STORE 0 TO COUNT
DO WHILE NOT. ECP
CASE
CASE 1 (MUNIT) = 'E'
DO TRNGCST
STORE COUNT + 1 TO COUNT
CASE 1 (MUNIT) = 'A'.AND. 1(MPLAT) = 'E'
IF COMPANY = ONECO
DO TRNGCST
STORE COUNT + 1 TO COUNT
ENDIF
CASE 1 (MUNIT) = 'A'.AND. 1(MPLAT) = 'A'
IF COMPANY = ONECO .AND. PLATOON = ONEPLT
DO TRNGCST
STORE COUNT + 1 TO COUNT
ENDIF
CASE 1 (MUNIT) = 'A'.AND. 1(MPLAT) = 'B'
IF COMPANY = ONECO .AND. PLATOON = TWOPLT
DO TRNGCST
STORE COUNT + 1 TO COUNT
ENDIF
CASE 1 (MUNIT) = 'A'.AND. 1(MPLAT) = 'C'
IF COMPANY = ONECO .AND. PLATOON = THREEPLT
DO TRNGCST
STORE COUNT + 1 TO COUNT
ENDIF
CASE 1 (MUNIT) = 'A'.AND. 1(MPLAT) = 'D'
IF COMPANY = ONECO .AND. PLATOON = FOURPLT
DO TRNGCST
129
CASE !( MUNIT ) = ' E' . AND. !( MPLAT ) = ' E'
  IF COMPANY = TWOCCO
  DO TRNGRPT
  STORE COUNT + 1 TO CCUNT
  ENIF

CASE !( MUNIT ) = ' E' . AND. !( MPLAT ) = ' A'
  IF COMPANY = TWOCCO . AND. PLATOON = ONEPLT
  DO TRNGRPT
  STORE CCUNT + 1 TO CCUNT
  ENIF

CASE !( MUNIT ) = ' E' . AND. !( MPLAT ) = ' B'
  IF COMPANY = TWOCCO . AND. PLATOON = TWOPLT
  DO TRNGRPT
  STORE COUNT + 1 TO CCUNT
  ENIF

CASE !( MUNIT ) = ' E' . AND. !( MPLAT ) = ' C'
  IF COMPANY = THREECCO
  DO TRNGRPT
  STORE CCUNT + 1 TO CCUNT
  ENIF

CASE !( MUNIT ) = ' E' . AND. !( MPLAT ) = ' D'
  IF COMPANY = THREECCO . AND. PLATOON = FOURPLT
  DO TRNGRPT
  STORE CCUNT + 1 TO CCUNT
  ENIF

CASE !( MUNIT ) = ' E' . AND. !( MPLAT ) = ' E'
  IF COMPANY = THREECCO . AND. PLATOON = THREEPET
  DO TRNGRPT
  STORE CCUNT + 1 TO CCUNT
  ENIF

CASE !( MUNIT ) = ' F' . AND. !( MPLAT ) = ' A'
  IF COMPANY = THREECCO . AND. PLATOON = ONEPLT
  DO TRNGRPT
  STORE CCUNT + 1 TO CCUNT
  ENIF

CASE !( MUNIT ) = ' F' . AND. !( MPLAT ) = ' B'
  IF COMPANY = THREECCO . AND. PLATOON = TWOPLT
  DO TRNGRPT
  STORE CCUNT + 1 TO CCUNT
  ENIF

CASE !( MUNIT ) = ' F' . AND. !( MPLAT ) = ' C'
  IF COMPANY = THREECCO . AND. PLATOON = THREEPET
  DO TRNGRPT
  STORE CCUNT + 1 TO CCUNT
  ENIF

CASE !( MUNIT ) = ' F' . AND. !( MPLAT ) = ' D'
  IF COMPANY = THREECCO . AND. PLATOON = FOURPLT
  DO TRNGRPT
  STORE CCUNT + 1 TO CCUNT
  ENIF

CASE !( MUNIT ) = ' F' . AND. !( MPLAT ) = ' E'
  IF COMPANY = PCURCO
  DO TRNGRPT
  STORE CCUNT + 1 TO CCUNT
  ENIF

CASE !( MUNIT ) = ' G' . AND. !( MPLAT ) = ' A'
  IF COMPANY = PCURCO . AND. PLATOON = ONEPLT
  DO TRNGRPT
  STORE CCUNT + 1 TO CCUNT
  ENIF

CASE !( MUNIT ) = ' G' . AND. !( MPLAT ) = ' B'
  IF COMPANY = PCURCO . AND. PLATOON = TWOPLT
  DO TRNGRPT
  STORE CCUNT + 1 TO CCUNT
  ENIF

CASE !( MUNIT ) = ' G' . AND. !( MPLAT ) = ' C'
  IF COMPANY = PCURCO . AND. PLATOON = THREEPET

130
DO TRNGRPT
STORE CCUNT + 1 TO CCUNT
ENDIF
CASE !(HUNIT) = 'C' AND !(MFLAT) = 'D'
IF COMPANY = FCLRCO AND PLATOON = FOURPLT
DC TRNGRPT
STORE CCUNT + 1 TO CCUNT
ENDIF
ENDCASE
IF CCUNT = 4
@ 22,0
ACCEPT "PRESS 'ENTER' TO CONTINUE" TO PROCEED
STORE 0 TO CCUNT
ENDIF
SELECT PRI MARY
SKIP
ENDDC
RELEASE CCUNT, PROCE ELMSSM
ELSE
RETURN
B. TRAINING ROSTER FORMAT ROUTINE

* ROUTINE NAME: STANDARD REPORT GENERATOR
* MODULE NAME: STANDARD REPORT GENERATOR
* VERSION: 5.4.1.0
* AUTHOR: R.E. PRUIET
* DATE: 15DEC83
* VARIABLES USED: COUNT, PLATNAME, UNITNAME, MPLAT, MUNIT
* VARIABLES MODIFIED: NONE
* VARIABLES CREATED: FCW, MSSN
* VARIABLES RELEASED: NONE
* FILES OPENED/CLOSED: E:QUAL
* INDEX E:QUALSSN, B:EST INDEX B:ESTSSN
* USING SUBROUTINES: TRNGROST.PRG
* DESCRIPTION: THIS ROUTINE PROVIDES THE FORMAT AND DISPLAYS THE "TRAINING ROSTER REPORT".

STORE (CCOUNT * 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,52 SAY "RIFLE QUAL"
ROW,63 SAY "PISTOL QUAL"
ROW,78 SAY "PSTQUAL"
SELECT SECONDARY
USE B:EST INDEX E:ESTSSN
FIND &MSSN
ROW + 1,0 SAY "CODE CP CONDUCT"
ROW + 1,16 SAY "COC"
ROW + 1,22 SAY "MARKS COR HIS"
ROW + 1,32 SAY "MARKSMANSHIP"
ROW + 1,43 SAY "MARKSMANSHIP"
ROW + 1,56 SAY "MKS"
ROW + 1,67 SAY "DRILL"
ROW + 1,74 SAY "COD"
ROW + 2,0 SAY "INTEOR GUARD"
ROW + 2,16 SAY "INT"
ROW + 2,22 SAY "MARKSMANSHIP"
ROW + 2,35 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 "FT T1"
ROW + 3,16 SAY "FT1CLSS"
ROW + 3,10 SAY "PFT2"
ROW + 3,16 SAY "PFT2CLSS"
ROW + 3,24 SAY "EQIP/UNIFORM"
ROW + 3,36 SAY "UNIT"
ROW + 3,43 SAY "FIRST ID"
RETUEN
I.  MARKSMANSHIP STATUS ROUTINE

* ROUTINE NAME: KSSSTAT.FRG
* MODULE NAME: STANDARD REPORT GENERATOR
* VERSION: 5.6.0
* AUTHOR: R.E. FRUDET
* DATE: 28NOV83
* VARIABLES USED: UNITNAME, PLATNAME
* VARIABLES MODIFIED: NO GLOBAL VARIABLES MODIFIED
* VARIABLES CREATED: RFLEX, RFLSS, RFLMM, RFLUN, PSTEX, PSTSS, PSTUN, RFLSHOT, PCTEX, PCTSS, PCTMM, PCTUN
* VARIABLES RELEASED: EST?????, PCT?????, RFLEX, RFLSS,
* RFLMM, RFLUN, PROCEED, ANALYZE6
* FILES OPENED/CLOSED: ALL CLOSED
* TEMP FILES CREATED: MKRSTEMP
* 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 TOTALLED BY QUALIFICATION
* CATEGORY FOR THE RIFLE AND PISTOL. PERCENTAGES ARE
* CALCULATED AND DISPLIED IN A SUMMARY STATUS REPORT WHICH
* IS AUTOMATICALLY DISPLAYED ON THE SCREEN.

ERASE LO CVRUNIT
DO CASE
    CASE E:(MUNIT)='E'
        USE E:QUAL
    CASE E:(MUNIT)<>'E' .AND. ! (MPLAT)='E'
        USE A:PERSON
        COPY TO MKRSTEMP ALL FIELD SSN FOR COMPANY=UNITNAME
        USE MKRSTEMP
        SELECT SECONDARY USE E:QUAL
        SELECT PRIMARY JOIN TO MKRSTEMP FOR E.FSN = S.FSN FIELDS RFLQUAL:
        ;
        PCTEX USE MKRSTEMP
    CASE E:(MUNIT)<>'E' .AND. ! (MPLAT)<>'E'
        USE A:PERSON
        COPY TO MKRSTEMP ALL FIELD SSN FOR COMPANY=UNITNAME;
        .AND. PLATOON = PLATNAME
        USE MKRSTEMP
        SELECT SECONDARY USE E:QUAL
        SELECT PRIMARY JOIN TO MKRSTEMP FOR E.FSN = S.FSN FIELDS RFLQUAL:
        ;
        PCTEX USE MKRSTEMP
    ENDCASE
COUNT FOR RFLQUAL = 'EX' TO RFLEX
COUNT FOR RFLQUAL = 'SS' TO RFLSS
COUNT FOR RFLQUAL = 'EM' TO RFLMM
COUNT FOR RFLQUAL = 'UN' TO RFLUN
COUNT FOR PSTQUAL = 'EX' TO PSTEX
COUNT FOR PSTQUAL = 'SS' TO PSTSS
COUNT FOR PSTQUAL = 'EM' TO PSTMM
COUNT FOR PSTQUAL = 'UN' TO PSTUN
ERASE
STORE (RFLEX+RFLSS+RFLMM+RFLUN) TO RFLSHOT
IF RFLSHOT > 0
    STOR (RFLEX/RFLSHOT*100) TO PCTEX
    STORE (RFLSS/RFLSHOT*100) TO PCTSS
    STORE (RFLMM/RFLSHOT*100) TO PCTMM
    STORE (RFLUN/RFLSHOT*100) TO PCTUN
ENDIF
STORE (PSTEX+PSTSS+PSTMM+PSTUN) TO PSTSHOT

133
IF PSTSHCT > 0
  STCIE (PSTEX/PSTSHCT*100) TO PCTEXPST
  STCIE (PSTISS/PSTSHCT*100) TO PCTSSPST
  STCIE (PSTM/PSTSHCT*100) TO PCTMPST
  STCIE (PSTUN/PSTSHCT*100) TO PCTUNPST
ENDIF

1,10 SAY "MARKSMANSHIP STATUS REPORT"
2,1 SAY "RIFLE EXPERTS............"
3,1 SAY "PISTOL EXPERTS............."
4,1 SAY "RIFLE SHARPSHOOTERS.......
5,1 SAY "PISTOL SHARPSHOOTERS.......
6,1 SAY "RIFLE MARLSMEN............
7,1 SAY "PISTOL MARLSMEN............
8,1 SAY "UNQUALIFIED................
9,1 SAY "RIFLE EXPERTS............."
10,1 SAY "PISTOL EXPERTS............."
11,1 SAY "RIFLE SHARPSHOOTERS.......
12,1 SAY "PISTOL SHARPSHOOTERS.......
13,1 SAY "RIFLE MARLSMEN............
14,1 SAY "PISTOL MARLSMEN............
15,1 SAY "UNQUALIFIED................
16,1 SAY "RIFLE EXPERTS............."
17,1 SAY "PISTOL EXPERTS............."
18,1 SAY "RIFLE SHARPSHOOTERS.......
19,1 SAY "PISTOL SHARPSHOOTERS.......
20,1 SAY "RIFLE MARLSMEN............
21,1 SAY "PISTOL MARLSMEN............
22,1 SAY "UNQUALIFIED................
23,1 SAY "RIFLE EXPERTS............."
24,1 SAY "PISTOL EXPERTS............."
25,1 SAY "RIFLE SHARPSHOOTERS.......
26,1 SAY "PISTOL SHARPSHOOTERS.......
27,1 SAY "RIFLE MARLSMEN............
28,1 SAY "PISTOL MARLSMEN............
29,1 SAY "UNQUALIFIED................
30,1 SAY "RIFLE EXPERTS............."
31,1 SAY "PISTOL EXPERTS............."
32,1 SAY "RIFLE SHARPSHOOTERS.......
33,1 SAY "PISTOL SHARPSHOOTERS.......
34,1 SAY "RIFLE MARLSMEN............
35,1 SAY "PISTOL MARLSMEN............
36,1 SAY "UNQUALIFIED................"
J. CONVERT UNIT ROUTINE

* ROUTINE NAME: CVRTUNIT.PRG
* MODULE NAME: STANDARD REPORT GENERATOR
* VERSION: 5.6.1.0
* AUTHOR: R.E. PROJETT
* DATE: 28NVCV3
* VARIABLES USED: MUNIT, MPIAT, ONECO, TWOCC, THREECC, FOURCC, CNPIAT, TWCC, THREEPLT, FOURPLT
* VARIABLES MODIFIED: NONE
* VARIABLES CREATED: UNITNAME, PLATNAME
* VARIABLES RELEASED: NONE
* FILES (OPENED/CLOSED): NONE
* TEOF FILES CREATED: NONE
* USING SUBROUTINES: MNSSTAT.PRG (5.6.0) AND MNSSTAT.PRG
* (5.6.1)
* DESCRIPTION: THIS ROUTINE STORES THE NAME OF THE COMPANY AND PLATCON SELECTED BY THE USER DURING ROUTINE
* SLCRTUNIT. 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
* SLCRTUNIT.PRG (A, B, C, D, E). SOME SCREEN DISPLAYS
* REQUIRE THE UNIT NAME (E.G., "A CO" VICE "A") SO A
* CONVERSION IS NECESSARY.

DO CASE
   CASE I(MUNIT) = "A"
       STORE ONECO TO UNITNAME
   CASE I(MUNIT) = "B"
       STORE TWOCC TO UNITNAME
   CASE I(MUNIT) = "C"
       STORE THREECC TO UNITNAME
   CASE I(MUNIT) = "D"
       STORE FOURCC TO UNITNAME
   IF I(MUNIT) <> "E"
      GO CASE
ENDCASE
   CASE I(PLAT) = "A"
       STORE ONEPLT TO PLATNAME
   CASE I(PLAT) = "B"
       STORE TWOPLT TO PLATNAME
   CASE I(PLAT) = "C"
       STORE THREEPLT TO PLATNAME
   CASE I(PLAT) = "D"
       STORE FOURPLT TO PLATNAME
ENDCASE
RETURN
K. QUOTA STUDY ROUTINE

* ROUTINE NAME: QTA STUDY PRG
* MODULE NAME: STANDARD REPORT GENERATOR
* VERSION: 5.6.2.0
* AUTHOR: E.E. PRUITT
* DATE: 15 SEP 83
* VARIABES USED: UNITNAME, FLATNAME, RPLSHOT, RFLEX,
* RFLSS, RFHMM, FLUN, MUNIT, QPEB, QMAR,
* ..
* VARIABES MODIFIED: NO GLCEAI VARIABLES MODIFIED
* VARIABLES CREATED: UNISNAME
* ..
* VARIABLES RELEASED: RCNT, MUNIT, TOTALREC, MLEFT,
* MLEFT, TOTALREC, RFHNA, QFEB, MAR,
* ..
* FILES (OPENED/CLOSED): NONE
* TEMPE PILPS CREATED: NONE
* USING SUBROUTINES: PKSSTAT PRG
* DESCRIPTION: THIS MODUOE IS USED BY PKSSTAT 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-
* SHIIP BETWEEN REMAINING RANGE QUOTA REQUIREMENTS AND
* PUBLICATIONS RANGE QUOTA ALLOCATIONS. UPON CONCLUSION OF
* THE MODUOE THE USER MAY RETURN TO THE STANDARD REPORT
* MENU OR TO A SCRATCH PGM TO REVISE QUOTA ASSIGNMENTS.
* ..
* ELSE QUOTA ASSIGNMENTS FROM THE SCRATCH PGM CAN EITHER
* BE SAVED OR IGNORED.

CASE
  CASE !(MUNIT) <> 'E' ANDEL !(MPLAT) = 'E'
    1,15 SAY "RANGE QUOTA ANALYSIS FOR "
    1,40 SAY UNITNAME
  CASE !(MUNIT) <> 'E' ANDEL !(MPLAT) <> 'E'
    1,15 SAY "RANGE QUOTA ANALYSIS FOR "
    1,40 SAY UNITNAME
    1,46 SAY ""
    1,48 SAY FLATNAME
  CASE !(MUNIT) = 'E'
    1,15 SAY "RANGE QUOTA ANALYSIS FOR BATTALION"
ENDCASE
COUNT FC RFLQUAL = 'NA' TO RFLNA
STORE $ (ADATE, 3, 2) TC MUNIT
GOTO EXTRACT
STORE 0 TC TOTALREC
4,1 SAY "NUMBER OF MARINES FIRED FOR QUALIFICATION; DURING"
5,1 SAY "CURRENT; YEAR "
5,56 SAY "RPLSHOT"
6,1 SAY "NUMBER OF MARINES FIRED FOR QUALIFICATION; DURING"
7,1 SAY "CURRENT YEAR BUT DID NOT; QUALIFY"
7,30 SAY "RFLEX"
8,1 SAY "NUMBER OF MARINES NOT REQUIRED TO QUALIFY; DURING"
9,1 SAY "CURRENT; YEAR"
9,55 SAY "RFLNA"
STORE (TOTALREC - RFLNA - RFLEX - RFLSS - RFLMM) TO MLEFT
10,1 SAY "MARINES LEFT TO QUALIFY DURING CURRENT; YEAR"
10,50 SAY MLEFT
ENDCASE
EO CASE

136
CASE MUNIT = 'A'
    RESTORE FROM A: ARNGQTAS ADDITIVE
ENDCASE
CASE MUNIT = 'C'
    RESTORE FROM A: ENRNGQTAS ADDITIVE
ENDCASE
CASE MUNIT = 'E'
    RESTORE FROM A: ERNGQTAS ADDITIVE
ENDCASE
CASE MUNIT = 'G'
    RESTORE FROM A: GRNGQTAS ADDITIVE
ENDCASE
CASE MUNIT = 'H'
    RESTORE FROM A: HNGQTAS ADDITIVE
ENDCASE

CASE MUNIT = 'I'
    RESTORE FROM A: INGQTAS ADDITIVE
ENDCASE
CASE MUNIT = 'J'
    RESTORE FROM A: JNGQTAS ADDITIVE
ENDCASE
CASE MUNIT = 'K'
    RESTORE FROM A: KNGQTAS ADDITIVE
ENDCASE
CASE MUNIT = 'L'
    RESTORE FROM A: LNGQTAS ADDITIVE
ENDCASE
CASE MUNIT = 'M'
    RESTORE FROM A: MNGQTAS ADDITIVE
ENDCASE
CASE MUNIT = 'N'
    RESTORE FROM A: NNGQTAS ADDITIVE
ENDCASE
CASE MUNIT = 'O'
    RESTORE FROM A: ONGQTAS ADDITIVE
ENDCASE
CASE MUNIT = 'P'
    RESTORE FROM A: PINGQTAS ADDITIVE
ENDCASE

CASE MUNIT = 'Q'
    RESTORE FROM A: QINGQTAS ADDITIVE
ENDCASE
CASE MUNIT = 'R'
    RESTORE FROM A: RINGQTAS ADDITIVE
ENDCASE
CASE MUNIT = 'S'
    RESTORE FROM A: SINGQTAS ADDITIVE
ENDCASE
CASE MUNIT = 'T'
    RESTORE FROM A: TINGQTAS ADDITIVE
ENDCASE
CASE MUNIT = 'U'
    RESTORE FROM A: UINGQTAS ADDITIVE
ENDCASE
CASE MUNIT = 'V'
    RESTORE FROM A: VINGQTAS ADDITIVE
ENDCASE
CASE MUNIT = 'W'
    RESTORE FROM A: WINGQTAS ADDITIVE
ENDCASE
CASE MUNIT = 'X'
    RESTORE FROM A: XINGQTAS ADDITIVE
ENDCASE
CASE MUNIT = 'Y'
    RESTORE FROM A: YINGQTAS ADDITIVE
ENDCASE
CASE MUNIT = 'Z'
    RESTORE FROM A: ZINGQTAS ADDITIVE
ENDCASE

CASE MUNTE = '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 (CAUG,3)
    14,34 SAY STR (CQJUL,3)
    16,4 SAY STR (CQJUL,3)
    14,40 SAY STR (CQSEP,3)
    14,46 SAY STR (COCT,3)
    16,4 SAY STR (CQDEC,3)
STCRE (QFEB+ QMAR+ QAPR+ QMAY+ QJUN+ QJUL+ QAUG+ QSEP+ QOCT+;
    QCCT+ QCNV+ QQEC) TO MSUM
    16,4 SAY "A TOTAL OF "
    16,14 SAY STR (MSUM,3)
    CASE MUNTE = '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 (CAUG,3)
    14,28 SAY STR (CQJUL,3)
    14,34 SAY STR (CQSEP,3)
    16,4 SAY STR (CQDEC,3)
STCRE (QMAR+ QAPR+ QMAY+ QJUN+ QJUL+ QAUG+ QSEP+ QOCT+;
    QCCT+ QCNV+ QQEC) TO MSUM
    16,4 SAY "A TOTAL OF ";
    16,14 SAY STR (MSUM,3)
    CASE MUNTE = '03'
    12,4 SAY "MAR MAY JUN JUL AUG SEP OCT NCV LEC"
    14,4 SAY STR (CMAY,3)
    14,10 SAY STR (CJUN,3)
    14,16 SAY STR (CAUG,3)
    14,22 SAY STR (CQJUL,3)
    14,28 SAY STR (CQSEP,3)
    14,34 SAY STR (CQDEC,3)
STCRE (CAPR+ QMAY+ QJUN+ QJUL+ QAUG+ QSEP+ QOCT+;
    QCCT+ QCNV+ QQEC) TO MSUM
    16,4 SAY "A TOTAL OF ";
    16,14 SAY STR (MSUM,3)
    CASE MUNTE = '04'
    12,4 SAY "MAY JUN JUL AUG SEP OCT NCV LEC"
137
CASE ACNTH = '06'

CASE ACNTH = '07'

CASE ACNTH = '08'

CASE ACNTH = '09'

CASE ACNTH = '10'
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(MSU, 3)
16, 19 SAY "QUOTAS REMAIN FOR THE CURRENT YEAR."
CASE MONTH = '1'
12, 4 SAY "DEC"
14, 4 SAY  STR(QDEC, 3)
STORE (QDEC) TO MSU
16, 4  SAY "A TOTAL OF"
16, 15 SAY  STR(MSU, 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
DO CASE
CASE (MSUM = MLEFT)
  18, 4 SAY "REMAINING QUOTAS EQUAL NUMBER OF MARINES;"
  18, 4 SAY  STR(MSUM, 3)
CASE (MLEFT > MSUM)
  18, 4 SAY "YOU NEED "
  18, 4 SAY  STR(MLEFT - MSUM)
CASE (MSUM = MLEFT)
  18, 4 SAY "YOU HAVE "
  18, 4 SAY  STR(MSUM - MLEFT) TO EXTRA
  18, 13 SAY  STR(EXTRA, 3)
ENDCASE
18, 18 SAY "EXTRA QUOTAS."
ENDCASE
20, 1 SAY "SELECT ONE"
21, 1 SAY "A...CHANGE QUOTAS ON SCRATCH PAD"
21, 1 SAY "B...RETURN TO STANDARD REPORT MENU"
ACCEPT "SELECT OPTION ===> " TO WHATIF
IF I (WHATIF) = 'A'
DO EX: QUOTAPAD
ENDIF
RELEASE MNCNT, WHATIF, EXTRA, NEEDED, MSUM, MLEFT, TOTALREC, RFNAM
RELEASE ALL LIKE Q???
RETURN
I. QUOTA SCRATCH PAD ROUTINE

ROUTINE NAME: QUOTAED.PRG
MODULE NAME: STANDARD REPORT GENERATOR
VERSION: 5.6.2.1.0
AUTHOR: R. E. PRIEETT
DATE: 5DEC3
VARIABLES USED: MLEFT, MUNIT
VARIABLES MODIFIED:
VARIABLES CREATED: PICKMNTH, TEMPSUM, WHICHWAY, CONTINUE, QUOTA, MLEFT, EXTRACT, PICKMNTH, CONTINUE, EXTRACT
FILES [OPENED/CLOSED]: NONE
TIME FILES CREATED: NONE
USING SUBROUTINES: CTAStudy.ERG
DESCRIPTION: THIS MODULE GIVES THE USER AN OPPORTUNITY TO歷
INTER-ATIVELY 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 I TO PICKMNTH
SET COUNTER OFF

DO CASE
CASE MONT = '01'
DO WHILE PICKMNTH = T
  TO 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:BNRNGQTAS ADDITIVE
  ENDCASE
ENDCASE
ENDCASE
ENDCASE
2 2.5 SAY "FILE RANGE QUOTA SCRATCH PAD"
2 4.5 SAY "SPACE BELOW THIS LINE IS FOR:
CALCULATIONS AND WILL BE SAVED"
2 5.5 SAY "ONLY IF YOU SPECIFY THE "REPLACE;
CURRENT QUOTAS" OPTION"
2 6.0 SAY ""**"**"**"**"**"**
2 8.5 SAY "FEB MAR APR MAY JUN JUL:
AUG SEP OCT NOV DEC"
2 10.4 GET QFEB PICTURE '999
2 10.10 GET QMAR PICTURE '999
2 10.22 GET QAPR PICTURE '999
2 10.22 GET QJUN PICTURE '999
2 10.34 GET QJUL PICTURE '999
2 10.40 GET CAUG PICTURE '999
2 10.46 GET CSEP PICTURE '999
2 10.52 GET CCTJ PICTURE '999
2 10.58 GET CNOV PICTURE '999
2 10.64 GET CDEC PICTURE '999
REAL
2 12.4 SAY "A TOTAL OF"
STORE (QFEB+QMAR+QAPR+QJUN+QJUL+;
CAUG+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..."
  @ 14, 53 SAY STR(TEMPSUM, 3)
CASE (MLEFT > TEMPSUM)
  @ 14, 4 SAY "YOU NEED "
  STORE (MLEFT - TEMPSUM) TO NEEDED
  @ 14, 13 SAY STR(NEEEDED, 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, 19 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 (WHICHWAY) = 'B'
  STORE F TO PICKMNTH
CASE (WHICHWAY) = 'C'
  ACCEPT "PC YOU WANT TO PERMANENTLY REPLACE THE; ORIGINAL SET OF QUOTAS (Y/N)?"; TO CONTINUE
  IF CONTINUE
    DC 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:ERNQTA ALL LIKE Q???
    ENDCASE
    STORE F TO PICKMNTH
  ELSE
    STORE F TO PICKMNTH
  ENDIF
ENDCASE
ENDDC

CASE MONTH = '02'
DO WHILE PICKMNTH = T
  DO CASE
    CASE (MUNIT) = 'A'
      RESTORE FCN A:ARNGQTAS ADDITIVE
    CASE (MUNIT) = 'B'
      RESTORE FCN A:BRNGQTAS ADDITIVE
    CASE (MUNIT) = 'C'
      RESTORE FCN A:CRNGQTAS ADDITIVE
    CASE (MUNIT) = 'D'
      RESTORE FCN A:DRNGQTAS ADDITIVE
    CASE (MUNIT) = 'E'
      RESTORE FCN A:ERNQTA ADDITIVE
  ENDCASE
ENDCASE
@ 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 SEP OCT NOV DEC"
@ 10,10 GET CHAR PICTURE '999'
@ 10,16 GET CHAR PICTURE '999'
@ 10,22 GET CHAR PICTURE '999'
@ 10,28 GET CHAR PICTURE '999'
@ 10,34 GET CHAR PICTURE '999'
@ 10,40 GET CHAR PICTURE '999'
@ 10,46 GET CHAR PICTURE '999'
READ
@ 12,4 SAY "TOTAL OF "
STORE CHAR*CHAR+CHAR+CHAR+CHAR+CHAR+CHAR+CHAR+CHAR+CHAR TO TEMP\SUM
@ 12,15 SAY STR(TEMP\SUM,3)
@ 12,19 SAY "QUOTAS REMAIN FOR THE CURRENT YEAR."
DO CASE
CASE (TEMP\SUM = MLEFT)
   @ 14,4 SAY "QUOTAS EQUAL NUMBER OF MARINES;
   LEFT TO QUALIFY."
   @ 14,53 SAY STR(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 WHICHWAY
DO CASE
CASE ! (WHICHWAY) = 'B'
   STORE P TO PICKMYTH
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:RANGQTAS ALL LIKE Q???
   CASE ! (MUNIT) = 'B'
      SAVE TO A:BANGQTAS 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
STORE F TO PICK\( MNTH \)
ELSE
STORE F TO PICK\( MNTH \)
ENDIF
ENDCASE
ENDDO
CASE \( MCNTH = '03' \)
DO WHILE PICK\( MNTH = T \)
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
\( \text{say} \) \"RIFLE RANGE QUOTA SCRATCH PAD" \( \text{say} \) \"SPACE BELOW THIS LINE IS FOR; CALCULATIONS AND WILL BE SAVED" \( \text{say} \) \"OK IF YOU SPECIFY THE 'REPLACE; CURRENT QUOTAS' OPTION \"
\( \text{say} \) \"**************************; "
\( \text{say} \) \"APR MAY JUN JUL; AUG SEP COT MCV DEC; "
\( \text{GET APR PICTURE} \ '999'; \) \( \text{GET MAY PICTURE} \ '999'; \) \( \text{GET JUN PICTURE} \ '999'; \) \( \text{GET JUL PICTURE} \ '999'; \) \( \text{GET AUG PICTURE} \ '999'; \) \( \text{GET SEP PICTURE} \ '999'; \) \( \text{GET OCT PICTURE} \ '999'; \) \( \text{GET NOV PICTURE} \ '999'; \) \( \text{GET DEC PICTURE} \ '999'; \)
\( \text{ERASE} \)
\( \text{say} \) \"A TOTAL OF \"
STORE \( T(\text{APR} + \text{MAY} + \text{JUN} + \text{JUL}; \) \( \text{CASE} \text{OCT} + \text{NOV} + \text{DEC}) \) TO TEMPSUM
\( \text{say} \) \"ST(TEMPSUM, 3); \) \( \text{say} \) \"QUOTAS REMAIN FOR THE CURRENT YEAR."
ENDCASE
CASE \( \text{TEMPSUM} = \text{MLEFT} \)
\( \text{say} \) \"QUOTAS EQUAL NUMBER OF MARINES; LEFT TO QUALIFY..." \( \text{say} \) \"ST(TEMPSUM, 3); \)
CASE \( \text{MLEFT} > \text{TEMPSUM} \)
\( \text{say} \) \"YOU NEED "
STORE \( \text{MLEFT} - \text{TEMPSUM} \) TO NEEDED
\( \text{say} \) \"ST(NEEDED, 3); \) \( \text{say} \) \"MORE QUOTAS." \( \text{say} \) \"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;"
@ SEIECtION MENU
@ 18,18 SAY "C...PERMANENTLY REPLACE ORIGINAL QUOTAS"
ACCEPT "SELECT ONE ==> " TO WHICHWAY

TO CASE
CASE 1 (WHICHWAY) = 'B'
STORE F TO PICKMNTH
CASE 1 (WHICHWAY) = 'C'
ACCEPT "EC YOU WANT TO PERMANENTLY REPLACE THE; ORIGIinal 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:BRNGQTAS ALL LIKE Q??
ENDCASE
STORE F TO PICKMNTH
ELSE
STORE F TO PICKMNTH
ENDIF
ENDCASE
ENDDC

CASE MCNTF = '04'
DO WHILE PICKMNTH = T
TO CASE
CASE 1 (MUNIT) = 'A'
RESTORE PCM A:ARNGQTAS ADDITIVE
CASE 1 (MUNIT) = 'B'
RESTORE PCM A:BRNGQTAS ADDITIVE
CASE 1 (MUNIT) = 'C'
RESTORE PCM A:CRNGQTAS ADDITIVE
CASE 1 (MUNIT) = 'D'
RESTORE PCM A:DRNGQTAS ADDITIVE
CASE 1 (MUNIT) = 'E'
RESTORE PCM A:BRNGQTAS ADDITIVE
CASE 1 (MUNIT) = 'F'
RESTORE PCM A:BRNGQTAS ADDITIVE
ENDCASE
ERASE
2 2,15 SAY "FILE RANGE QUOTA SCRATCH PAD"
2 4,5 SAY "SPACE BELOW THIS LINE IS FOR;
2 CALCULATIONS AND WILL BE SAVED"
2 5,5 SAY "ONLY IF YOU SPECIFY THE 'REPLACE;
2 CURRENT QUOTAS' OPTION "
2 6,0 SAY "*****************************
2 "
2 8,5 SAY "MAY JUN JUL;"
AUG SEP
2 10,22 GET GAUG PICTURE '999'
2 10,28 GET GCJUN PICTURE '999'
2 10,34 GET GCJUL PICTURE '999'
2 10,40 GET GCSEP PICTURE '999'
2 10,46 GET GCSEP PICTURE '999'
2 10,52 GET GCSEP PICTURE '999'
2 10,58 GET GCSEP PICTURE '999'
2 10,64 GET GCDEC PICTURE '999'
READ

144
STOR (QAY+QJUN+QJUL+QAUG+QSEP+QOCT+QNOV+QDEC) TO TEMPSUM.

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"
  @ 14, 13 SAY STR(NEEDED,3)
CASE (TEMPSUM > MLEFT)
  @ 14, 4 SAY "YOU HAVE"
  @ 14, 13 SAY STR(EXTRA,3)
ENDCASE

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'
      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:ERNGQTAS ALL LIKE Q??
  ENDCASE
  STORE F TO PICKMNTH
ELSE
  STORE F TO PICKMNTH
ENDIF
ENDCASE

CASE MCNTE = '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 !(MUNIT) = 'E'
        RESTORE FROM A:ERNGQTAS ADDITIVE
ENDCASE
ENDDC
CASE I (NUNIT) = 'E'
RESTORE PGM A:BNGQTA ADDITIVE
ENDCASE
CASE
CASE 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 SEPT OCT NOV DEC"
@ 10, 28 GET CJUN picture '999'
@ 10, 34 GET CJUL picture '999'
@ 10, 40 GET CAUG picture '999'
@ 10, 46 GET CSEP picture '999'
@ 10, 52 GET CCCT picture '999'
@ 10, 64 GET CNOV picture '999'
@ 10, 64 GET CDEC picture '999'
END
CASE
CASE 12, 4 SAY "A TOTAL OF"
STORE (CJUN + CJUL +
CAUG + CSEP + CCCT + CNOV + CDEC) TO TEMPSUM
CASE 12, 19 SAY STR(TEMPSUM, 3)
CASE 12, 19 SAY "QUOTAS REMAIN FOR THE CURRENT YEAR."
CASE (TEMPSUM = MLEFT)
CASE @ 14, 4 SAY "QUOTAS EQUAL NUMBER OF MARINES;
LEFT TO QUALIFY...
CASE @ 14, 53 SAY STR(TEMPSUM, 3)
CASE (MLEFT > TEMPSUM)
CASE @ 14, 4 SAY "YOU NEED 
STORE (MLEFT - TEMPSUM) TO NEEDED
CASE @ 14, 13 SAY STR(NEEDED, 3)
CASE @ 14, 19 SAY "MORE QUOTAS."
CASE (TEMPSUM > MLEFT)
CASE @ 14, 4 SAY "YOU HAVE 
STORE (TEMPSUM - MLEFT) TO EXTRA
CASE @ 14, 13 SAY STR(EXTRA, 3)
CASE @ 14, 18 SAY "EXCESS QUOTAS."
ENDCASE
CASE 15, 24 SAY "SELECT ONE OPTION"
CASE 16, 18 SAY "A...START AGAIN W/ THE ORIGINAL QUOTAS"
CASE 17, 18 SAY "E...RETURN TO STANDARD REPORT:
SELECT MENU"
CASE 18, 18 SAY "C...PERMANENTLY REPLACE ORIGINAL QUOTAS"
CASE 18, 18 SAY "SELECT ONE ====> " TO WHICHWAY
CASE !(WHICEWAY) = 'B'
CASE !(WHICEWAY) = 'C'
CASE !(WHICEWAY) = 'A'
STORE F TO PICKMTH
CASE !(WHICEWAY) = 'B'
ACCEPT "DC YOU WANT TO PERMANENTLY REPLACE THE:
ORIGINAL SET OF QUOTAS (Y/N)?" TO CONTINUE
IF CONTINUE
DO CASE
CASE !(MUNIT) = 'A'
CASE !(MUNIT) = 'B'
CASE !(MUNIT) = 'C'
CASE !(MUNIT) = 'D'
CASE !(MUNIT) = 'E'
146
SAVE TO A:BRNGQTA ALL LIKE Q???
ENDCASE
STORE F TO PICKMNTH
ELSE
STORE F TO PICKMNTH
ENDIF
ENDCASE

CASE MCNTH = '06'
WHILE PICKMNTH = T
DO CASE
CASE !(MUNIT) = 'A'
RESTORE PFCM A:BRNGQTA ADDITIVE
CASE !(MUNIT) = 'B'
RESTORE PFCM A:BRNGQTA ADDITIVE
CASE !(MUNIT) = 'C'
RESTORE PFCM A:CRNGQTA ADDITIVE
CASE !(MUNIT) = 'D'
RESTORE PFCM A:DRNGQTA ADDITIVE
CASE !(MUNIT) = 'E'
RESTORE PFCM A:BRNGQTA ADDITIVE
ENDCASE
ERASE
1 2,5 SAY "PIECE RANGE QUOTA SCRATCH PAD"
2 4 SAY "SPACE BELOW THIS LINE IS FOR CACULATIONS AND WILL BE SAVEd"
2 5 SAY "ONLY IF YOU SPECIFY THE REPLACE;
CURRENT QUOTAS" OPTION"
6 0 SAY "***********
***********
***********
6 5 SAY "JUL;
AUG SEP OCT NOV DEC"
10,34 GET CJUL PICTURE '999'
10,40 GET CAUG PICTURE '999'
10,46 GET CSPT PICTURE '999'
10,52 GET CCCT PICTURE '999'
10,58 GET CRVI PICTURE '999'
10,64 GET CECC PICTURE '999'
BUILD
12 4 SAY "A TOTAL OF"
STORE (CJUL):
(CAUG+CSPT+CSCE+CTSCT+CTCV+CTCE) TO TEMPSUM
12 13 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 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"
ACCEP "SELECT ONE ====> " TO WHICHWAY

DC CASE
CASE 1 (WHICFWAY) = 'B'
STORE F TO PICKMNTH
CASE 1 (WHICFWAY) = 'C'
ACCEP "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 F:ARNGQTAS ALL LIKE Q???
CASE 1 (MUNIT) = 'B'
SAVE TO F:BRNGQTAS ALL LIKE Q???
CASE 1 (MUNIT) = 'C'
SAVE TO F:CRNGQTAS ALL LIKE Q???
CASE 1 (MUNIT) = 'D'
SAVE TO F:DRNGQTAS ALL LIKE Q???
CASE 1 (MUNIT) = 'E'
SAVE TO F:BNRNGQTAS ALL LIKE Q???
ENDCASE
STORE F TO PICKMNTH
ELSE
STORE F TO PICKMNTH
ENDIF
ENDCASE
ENDDO
CASE MCNTE = '07'
DO WHILE PICKMNTH = T
TO CASE
CASE 1 (MUNIT) = 'A'
RESTORE FROM A:ARNGQTAS ADDITIVE
CASE 1 (MUNIT) = 'B'
RESTORE FROM A:BRNGQTAS ADDITIVE
CASE 1 (MUNIT) = 'C'
RESTORE FROM A:CRNGQTAS ADDITIVE
CASE 1 (MUNIT) = 'D'
RESTORE FROM A:DRNGQTAS ADDITIVE
CASE 1 (MUNIT) = 'E'
RESTORE FROM A:BNRNGQTAS ADDITIVE
ENDCASE
ERASE
@ 18 SAY "FILE RANGE QUOTA SCRATCH PAD"
@ 19 SAY "SPACE BELOW THIS LINE IS FOR:
CALCULATIONS AND WILL BE SAVED"
@ 20 SAY "ONLY IF YOU SPECIFY THE 'REPLACE;
CURRENT QUOTAS' OPTION"
@ 21 SAY "**************************************************************************;
***************************************************************************
@ 22 SAY "**************************************************************************
@ 23 SAY "AUG SEP OCT NOV DEC"
@ 24 GET CAUG PICTURE '999'
@ 25 GET CSCP PICTURE '999'
@ 26 GET CCCT PICTURE '999'
@ 27 GET CNV PICTURE '999'
@ 28 GET CDEC PICTURE '999'
FINAL
STORE (CAUG+CSEP+CCCT+CNOV+CDEC) TO TEMPSUM
@ 31 SAY "A TOTAL OF "
@ 32 SAY "QUOTAS REMAIN FOR THE CURRENT YEAR."
DC CASE
CASE (TEMPSUM = MLEFT) TO CASE
@ 34 SAY "QUOTAS EQUAL NUMBER OF MARINES;
148
LEFT TO QUALIFY...
CASE (LEFT > TEMPSUM)
  @ 14,53 SAY STR(TEMPSUM, 3)
  @ 14,54 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"
@ 18,18 SAY "C...PERMANENTLY REPLACE ORIGINAL QUOTAS"
ACCEPT "SELECT ONE ===> " TO WHICHWAY
DO CASE
  CASE I(WHICHEWAY) = 'B'
  STORE F TO PICKNTH
  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:ENFGQTAS ALL LIKE Q???
  ENDCASE
  STORE F TO PICKNTH
  ELSE
  STORE F TO PICKNTH
  ENDIF
ENDCASE
ENDDC
CASE MCMTE = '08'
DO WHILE PICKNTH = 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:ENFGQTAS ADDITIVE
  ENDCASE
  ERASE
@ 15,15 SAY "FULL RANGE QUOTA SCRATCH PAD"
@ 15,5 SAY "SPACE BELOW THIS LINE IS FOR CALCUATIONS AND WILL BE SAVED"
@ 5,5 SAY "ONLY IF YOU SPECIFY THE 'REPLACE CURRENT QUOTAS' OPTION"
@ 6,0 SAY "*******************************************************************
145
***************
@ 8,5 SAY " 
SET OCT NCV DEC"
@ 10,6 GET CSEP PICTURE '999'
@ 10,12 GET CCCT PICTURE '999'
@ 10,18 GET CDEC PICTURE '999'
@ 10,24 GET CVC PICTURE '999'
REAL

@ 12,4 SAY "A TOTAL OF 
STORE (QSEP+CCCT+QNCV+CDEC) TO TEMPSUM
@ 12,19 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;
  LEFT TO QUALIFY...
CASE (MLEFT > TEMPSUM)
  @ 14,4 SAY "YOU NEED 
STORE (MLEFT - TEMPSUM) TO NEEDED
  @ 14,13 SAY STR(NEED,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 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'
        SAVE TO A:ARNGQT AS ALL LIKE Q???
      CASE !(MUNIT) = 'B'
        SAVE TO A:BNGQTAS ALL LIKE Q???
      CASE !(MUNIT) = 'C'
        SAVE TO A:CNNGQTAS ALL LIKE Q???
      CASE !(MUNIT) = 'D'
        SAVE TO A:CRNGQTAS ALL LIKE Q???
      CASE !(MUNIT) = 'E'
        SAVE TO A:DRNGQTAS ALL LIKE Q???
    END CASE
    STORE $ TO PICKMNTH
  ELSE
    STORE F TO PICKMNTH
  END CASE
ENDIF
ENDCASE
ENDDO

CASE MNTF = '09'
DO WHILZ PICKMNTH = T
TO CASE
CASE !(MUNIT) = 'A'

150
RESTORE FROM A:APNQGTAS ADDITIVE
CASE ! (MUNIT) = 'A'
RESTORE FROM A:APNQGTAS ADDITIVE
CASE ! (MUNIT) = 'B'
RESTORE FROM A:APNQGTAS ADDITIVE
CASE ! (MUNIT) = 'C'
RESTORE FROM A:APNQGTAS ADDITIVE
CASE ! (MUNIT) = 'D'
RESTORE FROM A:APNQGTAS ADDITIVE
CASE ! (MUNIT) = 'E'
RESTORE FROM A:APNQGTAS ADDITIVE
ENDCASE

BASE
@ 2.75 SAY "FIPLE RANG O TA SCRATCH PAD"
@ 4.5 SAY "SPACE BELOW "H" LINE IS FOR;
CALCULATIONS AND WILL U SAVED"
@ 5.5 SAY "ONLY IF YOU SPECIFY THE 'REPLACE;
CURRENT QUOTAS' OPTION"
@ 6.0 SAY "***************************;
*********------------------*"
@ 8.5 SAY ";
CCT NOV DEC"
@ 10.5 GET (CCT PICTURE '999');
@ 10.56 GET (NOV PICTURE '999');
@ 10.64 GET (DEC PICTURE '999');
REAL

@ 12.4 SAY "A TOTAL OF "
STORE QCT+NOV+DEC TO TEMPSUM
@ 12.15 SAY STR(TEMPSUM,2)
@ 12.19 SAY "QUOTAS REMAIN FOR THE CURRENT YEAR."

LO CASE
CASE {TEMPSUM = MLEFT)
@ 14.1 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)
@ 14.19 SAY "EXCESS QUOTAS."
ENDCASE

@ 15.24 SAY "SELECT ONE OPTION"
@ 15.18 SAY "A... START AGAIN WITH ORIGINAL QUOTAS"
@ 15.18 SAY "B... RETURN TO STANDARD REPORT;
SELECT MENU"
@ 15.18 SAY "C... PERMANENTLY REPLACE ORIGINAL QUOTAS"
ACCEPT "SELECT THE >> " TO WHICHWAY
DO CASE
CASE !(WHICHWAY) = 'A'
STORE P TC PICTURE
CASE !(WHICHWAY) = 'B'
ACCEPT "DO YOU WANT TO PERMANENTLY REPLACE THE;
ORIGINAL LINE?
IF 'Y' CASE
CASE ! (MUNIT) = 'A'
SAVE TO A:APNQGTAS ALL LIKE Q???
CASE ! (MUNIT) = 'B'
SAVE TO A:APNQGTAS ALL LIKE Q???
CASE ! (MUNIT) = 'C'
SAVE TO A:APNQGTAS ALL LIKE Q???
CASE ! (MUNIT) = 'D'
SAVE TO A:APNQGTAS ALL LIKE Q???
CASE (MUNIT) = 'B'
  SAVE TO A:BNRNGQTA ALL LIKE Q???
ENDCASE
STORE F TO PICKMnth
ELSE
STORE F TO PICKMnth
ENDIF
ENDCASE
ENDDO
CASE Mnth = '10'
DO WHILE PICKMnth = T
  DO CASE
    CASE ! (MUNIT) = 'A'
      RESTORE FCm A:ABNGCTAS ADDITIVE
    CASE ! (MUNIT) = 'B'
      RESTORE FCm A:BRNGCTAS ADDITIVE
    CASE ! (MUNIT) = 'C'
      RESTORE FCm A:CRNGCTAS ADDITIVE
    CASE ! (MUNIT) = 'D'
      RESTORE FCm A:DNMGCTAS ADDITIVE
    CASE ! (MUNIT) = 'E'
      RESTORE FCm A:BNRNGQTA ADDITIVE
ENDCASE
ERASE
3 1,15 SAY "BIFLE 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 "

9 10,58 GET Cncv picture '999'
@ 10,64 GET CDEc picture '999'
READ
2 1:4 SAY "A TOTAL OF"
@ 2,12,15 SAY STR(tempsum,3)
@ 12,19 SAY "QUOTAS REMAIN FOR THE CURRENT YEAR."
DO CASE
  CASE (tempsum = Mleft)
    @ 4,4 SAY "QUOTAS EQUAL NUMBER OF MARINES;
    LEFT TO QUALIFY..."
    @ 14,53 SAY STR(Mleft,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"
@ 17,18 SAY "E...RETURN TO STANDARD REPORT;
       SELECTION MENU"
@ 16,18 SAY "C...PERMANENTLY REPLACE ORIGINAL;
QUOTAS"
ACCEPT "SELECT ONE ====> " TO WHICHWAY

152
CASE I (WHICH WAY) = 'B'
STORE F TO PICKMTH
CASE I (WHICH WAY) = '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: BRNRNGQTA ALL LIKE Q???
ENDCASE
ELSE
STORE F TO PICKMTH
ENDIF
ENDCASE
ENDDC
CASE MUNIT = '11'
DO WHILE PICKMTH = T
DO CASE
CASE I (MUNIT) = 'A'
RESTORE PFCM A: ARNGQTAS ADDITIVE
CASE I (MUNIT) = 'B'
RESTORE PFCM A: BRNGQTAS ADDITIVE
CASE I (MUNIT) = 'C'
RESTORE PFCM A: CRNGQTAS ADDITIVE
CASE I (MUNIT) = 'D'
RESTORE PFCM A: DRNGQTAS ADDITIVE
CASE I (MUNIT) = 'E'
RESTORE PFCM A: BRNRNGQTA ADDITIVE
ENDCASE
ERASE
@ 2, 15 SAY "RIFLE RANGE QUOTA SCRATCH PAD"
@ 4, 6 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
RECE
@ 10, 64 GET CIEC PICTURE '999'
REAL
@ 12, 4 SAY "A TOTAL OF "
STORE (QEB+CMAR+QAPF+QWAY+QJUN+QJUL+;
CAQIG+QSEP+QCLI+QHOV+CIEC) 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 WHICH WAY
DC CASE
CASE I(WHICHWAY) = 'B'
STORE P TO PICKMNTH
CASE I(WHICHWAY) = 'C'
ACCEPT "DC 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:BRNGQTAS ALL LIKE Q???
ENDCASE
STORE F TO PICKMNTH
ELSE
STORE F TO PICKMNTH
ENDIF
INICASE
ENDCASE
CASE MMONTH = '12'
ERASE
\* 4.15 SAY "BRING RANGE QUOTA SCRATCH PAD"
\* 5.5 SAY "SPACE BELOW THIS LINE IS FOR CALCULATIONS; AND WILL BE SAVED"
\* 6.0 SAY "ONLY IF YOU SPECIFY THE 'REPLACE CURRENT; QUOTAS' OPTION"
\* 6.1 SAY "*******************************;
*****" 
\* 8.5 SAY "THIS IS THE LAST MONTH OF THE CURRENT; YEAR."
ENDCASE
ENDCASE
SET CLCN ON
RELEASE ALL LIKE Q???
RELEASE MLEFT, TEMPSUM, WHICHWAY, PICKMNTH, CONTINUE
RETURN
N. MARKSMANSHIP ROSTER ROUTINE

* ROUTINE NAME: MKSRRCST.PRG
* MODULE NAME: STANDARD REPORT GENERATOR
* VERSION: 5.7.0
* AUTHOR: R. E. TRUETT
* DATE: 28 DEC 83
* VARIABLES USED: MUNIT, MPLAT
* VARIABLES MODIFIED: MUNIT, MPLAT
* VARIABLES CReATED: MUNIT, MPLAT
* FILES OPENED/CLOSED: PERS, QUAL (ALL FILES CLOSED)
* TIME FILES UPDATED: MKSDATA (CREATED AND DELETED)
* USING SUBROUTINES: STDRT.PRG
* DESCRIPTION: THIS ROUTINE CREATES A TEMPORARY FILE WHICH
  INCLUDES NAME, RFICUAL, RFLDATE, PSTQUAL, PSTDATE FOR
  EACH MEMBER IN THE SELECTED UNIT. A STANDARD REPORT
  FORMAT (B: MKSRPT) IS USED TO DISPLAY THE "MARKSMANSHIP
  ROSTER REPORT".

USE PERS
SELECT SECONDARY
USE QIAL
JOIN TO MKSDATA FOR E.SSN = S.SSN FIELD;
NAME, RFICUAL, RFLDATE, PSTQUAL, PSTDATE
USE MKSRDATA
ERASE
DO CASE
  CASE I (MUNIT) = 'A'
    REPORT FORM B: MKSRPT
  CASE I (MUNIT) = 'B' .AND. ! (MPLAT) = 'A'
    REPORT FORM B: MKSRPT FOR COMPANY = ONECO
  CASE I (MUNIT) = 'B' .AND. ! (MPLAT) = 'B'
    FLATCN = ONEPIT
  CASE I (MUNIT) = 'B' .AND. ! (MPLAT) = 'C'
    FLATCN = TWOPII
  CASE I (MUNIT) = 'B' .AND. ! (MPLAT) = 'D'
    FLATCN = THREPIT
  CASE I (MUNIT) = 'B' .AND. ! (MPLAT) = 'E'
    FLATCN = FOURPIT
  CASE I (MUNIT) = 'C' .AND. ! (MPLAT) = 'A'
    REPORT FORM B: MKSRPT FOR COMPANY = TWOCC
  CASE I (MUNIT) = 'C' .AND. ! (MPLAT) = 'B'
    REPORT FORM B: MKSRPT FOR COMPANY = TWOCC
  CASE I (MUNIT) = 'C' .AND. ! (MPLAT) = 'E'
    FLATCN = TWOPIT
  CASE I (MUNIT) = 'C' .AND. ! (MPLAT) = 'D'
    FLATCN = THREPIT
  CASE I (MUNIT) = 'C' .AND. ! (MPLAT) = 'E'
    REPORT FORM B: MKSRPT FOR COMPANY = TWOCC
  CASE I (MUNIT) = 'D' .AND. ! (MPLAT) = 'A'
    REPORT FORM B: MKSRPT FOR COMPANY = THREECO
  CASE I (MUNIT) = 'D' .AND. ! (MPLAT) = 'B'
    REPORT FORM B: MKSRPT FOR COMPANY = THREECO
  CASE I (MUNIT) = 'D' .AND. ! (MPLAT) = 'C'
    FLATCN = CNEPIT
  CASE I (MUNIT) = 'D' .AND. ! (MPLAT) = 'E'
    REPORT FORM B: MKSRPT FOR COMPANY = THREECO
  CASE I (MUNIT) = 'E' .AND. ! (MPLAT) = 'A'
    REPORT FORM B: MKSRPT FOR COMPANY = THREECO
  CASE I (MUNIT) = 'E' .AND. ! (MPLAT) = 'B'
    FLATCN = TWPIT
  CASE I (MUNIT) = 'E' .AND. ! (MPLAT) = 'C'
    REPORT FORM B: MKSRPT FOR COMPANY = THREECO
  CASE I (MUNIT) = 'E' .AND. ! (MPLAT) = 'D'
    FLATCN = TWPIT
  CASE I (MUNIT) = 'E' .AND. ! (MPLAT) = 'E'
    REPORT FORM B: MKSRPT FOR COMPANY = THREECO

135
CASE 1 (MUNIT) = 'E' .AND. !(MPLAT) = 'E'
EFFECT FORM B: MRKSRPT FOR COMPANY = THREECO .AND;
ELATCCN = FOURFIT
CASE 1 (MUNIT) = 'E' .AND. !(MPLAT) = 'A'
EFFECT FORM B: MRKSRPT FOR COMPANY = FOURCO
CASE 1 (MUNIT) = 'E' .AND. !(MPLAT) = 'B'
EFFECT FORM B: MRKSRPT FOR COMPANY = FOURCO .AND;
ELATCCN = CNEPIT
CASE 1 (MUNIT) = 'E' .AND. !(MPLAT) = 'C'
EFFECT FORM B: MRKSRPT FOR COMPANY = FOURCO .AND;
ELATCCN = TWOPI
CASE 1 (MUNIT) = 'E' .AND. !(MPLAT) = 'D'
EFFECT FORM B: MRKSRPT FOR COMPANY = FOURCO .AND;
ELATCCN = THREEFL
CASE 1 (MUNIT) = 'E' .AND. !(MPLAT) = 'E'
EFFECT FORM B: MRKSRPT FOR COMPANY = FOURCO .AND;
ELATCCN = FOURFIT
ENDCASE USE
RELEASE MUNIT, MPLAT
DELETE FILE MAKSDATA
RETURN
N. MCS STATUS ROUTINE

* ROUTINE NAME: MOSSSTAT.PRG
* MODULE NAME: STANDARD REPORT GENERATOR
* VERSION: 5.8.0
* AUTHOR: R.E. PRUITT
* DATE: JUN 83
* VARS/IS USED: ONECO, TWOCO, THREECO, FOURCO, ONEPLT,
  TWCEL, THREEPLT, FOURPLT, MUNIT, PLATNAME, UNITNAME,
* PLATNAME
* VARS/IS MODIFIED: NONE
* VARS/IS CREATED: COUNT, MOS, RK, PROCEED
* VARS/IS RELEASED: PROCEED, RK, COUNT, UNITNAME,
* PLATNAME
* FILES CLOSED/CLOSED: A:PEERS INDEX A:MOSSNDX (ALL FILES
  CLOSED)
* TIMES 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 TOTALLED, REGARDLESS OF THE TYPE OR NUMBER OF
  DIFFERENT MOS'S IN THE PARTICULAR UNIT. THE RESULTS ARE
  THEN DISPLAYED BY MCS AND BY RANK.

ERASE!
EO CVFIUNIT
USE A:PEERS INDEX A:MOSSNDX
4.20 SAY "MOS STATUS REPORT"
IF ! (MUNIT) = 'E'
  @ 0.36 SAY "BATTALION"
ENDIF
IF (MUNIT) <> 'E'
  @ 0.36 SAY UNITNAME
ENDIF
IF ! (PLATNAME) = 'E'. AND. ! (MUNIT) <> 'E'
  @ 0.46 SAY PLATNAME
ENDIF
2.10 SAY "MOS"
2.30 SAY "RANK"
2.55 SAY "TOTAL"
STORE 4 TO RK
EO WHILE NOT. EOF
STORE PRIMEMOS TO MOS
STORE RANK TO RK
STORE 0 TO COUNT
EO CASE
  ! (MUNIT) = 'E'
  EO WHILE PRIMEMOS=MOS .AND. RANK=RK .AND. .NOT.
  EOP
  STORE COUNT+1 TO COUNT
  SKIP
ENDCASE
  ! (PLATNAME) = 'E'
  EO WHILE PRIMEMOS=MOS .AND. RANK=RK .AND. .NOT.
  EOP
  IF COMPANY=UNITNAME
  STORE COUNT+1 TO COUNT
  ENDIF
  SKIP
ENDCASE
OTHERWISE
  EO WHILE PRIMEMOS=MOS .AND. RANK=RK .AND. .NOT.
  EOP
  IF COMPANY=UNITNAME .AND. PLATNAME
  STORE COUNT+1 TO COUNT
  ENDIF
  SKIP
ENDCASE
ENCASE IF CCNT > 0 THEN
  @ ROW, 30 SAY MCS
  @ ROW, 30 SAY RF
  @ FCW, 50 SAY CCNT
  STORE ROW+2 TO FCW
ENDIF
IF ROW > 20 THEN
  @ 23, 0 ACCEPT "PRESS 'ENTER' TO CONTINUE" TO PROCEED
  STORE 4 TO ROW
  @ 0, 20 SAY "MOS STATUS REPORT"
  IF I(MUNIT) = 'E'
    @ 0, 38 SAY "EATTALION"
  ENDIF
  IF I(MUNIT) <> 'F'
    @ C, 38 SAY UNITNAME
  ENDIF
  IF I(MUNIT) <> 'E'.AND. ! (MUNIT) <> 'E'
    @ 0, 46 SAY PLATNAME
  ENDIF
  @ 2, 10 SAY "MOS"
  @ 2, 30 SAY "RANK"
  @ 4, 55 SAY "TOTAL"
ENDIF
ENDDO
RELEASE PROCEED, ROW, MCS, RF, 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: 6/08/82
* VARIABLES USED: MUNIT, MPLAT, ONECO, TWOCO, THREECO,
  FOURCO, ONEPLT, TWOPLT, THREEPLT, FOURPLT
* VARIABLES MODIFIED: NONE
* VARIABLES USED: MUNIT, MPLAT
* FILES (OPENED/CLOSED): A:EFFS INDEX A:MOSENDX
* TEMP FILES CREATED: NONE
* USING SUBROUTINES: STDRPT.PRG
* DESCRIPTION: THIS IS A CONTROLL 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:EFFS INDEX A:MOSENDX
DO CASE
CASE ! (MUNIT) = 'E'
REPORT FORM B:MCSRPT FOR COMPANY = ONECO .AND.
MPLAT = 'E'
ENDCASE
CASE ! (MUNIT) = 'A'. AND. !(MPLAT) = 'E'
REPORT FORM B: MCSRPT FOR COMPANY = ONECO .AND.
ENDCASE
CASE ! (MUNIT) = 'A'. AND. !(MELAT) = 'B'
REPORT FORM B: MCSRPT FOR COMPANY = ONECO .AND.
ENDCASE
CASE ! (MUNIT) = 'A'. AND. !(MELAT) = 'C'
REPORT FORM B: MCSRPT FOR COMPANY = ONECO .AND.
ENDCASE
CASE ! (MUNIT) = 'A'. AND. !(MELAT) = 'D'
REPORT FORM B: MCSRPT FOR COMPANY = ONECO .AND.
ENDCASE
CASE ! (MUNIT) = 'E'. AND. !(MELAT) = 'E'
REPORT FORM B: MCSRPT FOR COMPANY = TWOCO .AND.
ENDCASE
CASE ! (MUNIT) = 'E'. AND. !(MELAT) = 'F'
REPORT FORM B: MCSRPT FOR COMPANY = TWOCO .AND.
ENDCASE
CASE ! (MUNIT) = 'E'. AND. !(MELAT) = 'G'
REPORT FORM B: MCSRPT FOR COMPANY = TWOCO .AND.
ENDCASE
CASE ! (MUNIT) = 'E'. AND. !(MELAT) = 'H'
REPORT FORM B: MCSRPT FOR COMPANY = TWOCO .AND.
ENDCASE
CASE ! (MUNIT) = 'E'. AND. !(MELAT) = 'I'
REPORT FORM B: MCSRPT FOR COMPANY = TWOPLT .AND.
ENDCASE
CASE ! (MUNIT) = 'E'. AND. !(MELAT) = 'J'
REPORT FORM B: MCSRPT FOR COMPANY = TWOPLT .AND.
ENDCASE
CASE ! (MUNIT) = 'E'. AND. !(MELAT) = 'K'
REPORT FORM B: MCSRPT FOR COMPANY = TWOPLT .AND.
ENDCASE
CASE ! (MUNIT) = 'E'. AND. !(MELAT) = 'L'
REPORT FORM B: MCSRPT FOR COMPANY = TWOPLT .AND.
ENDCASE
CASE ! (MUNIT) = 'E'. AND. !(MELAT) = 'M'
REPORT FORM B: MCSRPT FOR COMPANY = THREECO .AND.
ENDCASE
CASE ! (MUNIT) = 'E'. AND. !(MELAT) = 'N'
REPORT FORM B: MCSRPT FOR COMPANY = THREECO .AND.
ENDCASE
CASE ! (MUNIT) = 'E'. AND. !(MELAT) = 'O'
REPORT FORM B: MCSRPT FOR COMPANY = THREECO .AND.
ENDCASE
CASE ! (MUNIT) = 'E'. AND. !(MELAT) = 'P'
REPORT FORM B: MCSRPT FOR COMPANY = THREECO .AND.
ENDCASE
CASE ! (MUNIT) = 'E'. AND. !(MELAT) = 'Q'
REPORT FORM B: MCSRPT FOR COMPANY = FOURCO .AND.
ENDCASE
CASE ! (MUNIT) = 'E'. AND. !(MELAT) = 'R'
REPORT FORM B: MCSRPT FOR COMPANY = FOURCO .AND.
ENDCASE
CASE ! (MUNIT) = 'E'. AND. !(MELAT) = 'S'
REPORT FORM B: MCSRPT FOR COMPANY = FOURCO .AND.
ENDCASE
CASE ! (MUNIT) = 'E'. AND. !(MELAT) = 'T'
REPORT FORM B: MCSRPT FOR COMPANY = FOURCO .AND.
ENDCASE
CASE ! (MUNIT) = 'E'. AND. !(MELAT) = 'U'
REPORT FORM B: MCSRPT FOR COMPANY = FOURCO .AND.
ENDCASE
CASE ! (MUNIT) = 'E'. AND. !(MELAT) = 'V'
REPORT FORM B: MCSRPT FOR COMPANY = FOURCO .AND.
ENDCASE
CASE ! (MUNIT) = 'E'. AND. !(MELAT) = 'W'
REPORT FORM B: MCSRPT FOR COMPANY = FOURCO .AND.
ENDCASE
CASE ! (MUNIT) = 'E'. AND. !(MELAT) = 'X'
REPORT FORM B: MCSRPT FOR COMPANY = FOURCO .AND.
ENDCASE
CASE ! (MUNIT) = 'E'. AND. !(MELAT) = 'Y'
REPORT FORM B: MCSRPT FOR COMPANY = FOURCO .AND.
ENDCASE
CASE ! (MUNIT) = 'E'. AND. !(MELAT) = 'Z'
REPORT FORM B: MCSRPT FOR COMPANY = FOURCO .AND.
ENDCASE
159
BEFORE FORM B:MSCRPT FOR COMPANY = FOURCO .AND.;
FLATCCN = ONEPIT
CASE I(MUNIT) = 'L' .AND. I(MFLAT) = 'B'
BEFORE FORM B:MSCRPT FOR COMPANY = FOURCO .AND.;
FLATCCN = TWOPIT
CASE I(MUNIT) = 'L' .AND. I(MFLAT) = 'C'
BEFORE FORM B:MSCRPT FOR COMPANY = FOURCO .AND.;
FLATCCN = THREEILT
CASE I(MUNIT) = 'L' .AND. I(MFLAT) = 'D'
BEFORE FORM B:MSCRPT FOR COMPANY = FOURCO .AND.;
FLATCCN = FOURILT
ENDCASE
USE
RELEASE MUNIT, MFLAT
RETURN
F. PERSCHMEL DATA ROUTINE

* ROUTINE NAME: PERSLATA.PRG
* MODULE NAME: STANDARD REPORT GENERATOR
* VERSION: 5.10.0
* AUTHOR: R. E. PRUETT
* DATE: 2 DEC 83
* VARIABLES USED: MUNIT, MPLAT, CNECO, TWOPLT, THREEPLT,
  FOURPLT, CNEPLT, TWOPLT, THREEPLT
* VARIABLES MODIFIED: NONE
* VARIABLES CREATED: COUNT, PROCEED
* VARIABLES RELEASED: COUNT, PROCEED
* FILES (OPENED/CLOSED): A: PERS INDEX A: ALPHPERS
* TEMP FILES CREATED: NONE
* USING SUBROUTINES: SIDRPT.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).

ERASE
USE A: PERS INDEX A: ALPHPERS
GOTO TOP
STORE 0 TO COUNT
DO WHILE .NOT. EOF
TO CASE
CASE 1 (MUNIT) = 'E'
  TO PERSRPT
  STORE COUNT + 1 TO COUNT
CASE 1 (MUNIT) = 'A' AND (MPLAT) = 'E'
  IF COMPANY = ONECO
    TO PERSRPT
    STORE COUNT + 1 TO COUNT
  ENDIF
CASE 1 (MUNIT) = 'A' AND (MPLAT) = 'A'
  IF COMPANY = ONECO AND PLATOON = ONEPLT
    TO PERSRPT
    STORE COUNT + 1 TO COUNT
  ENDIF
CASE 1 (MUNIT) = 'A' AND (MPLAT) = 'B'
  IF COMPANY = ONECO AND PLATOON = TWOPLT
    TO PERSRPT
    STORE COUNT + 1 TO COUNT
  ENDIF
CASE 1 (MUNIT) = 'A' AND (MPLAT) = 'C'
  IF COMPANY = ONECO AND PLATOON = THREEPLT
    TO PERSRPT
    STORE COUNT + 1 TO COUNT
  ENDIF
CASE 1 (MUNIT) = 'A' AND (MPLAT) = 'D'
  IF COMPANY = ONECO AND PLATOON = FOURPLT
    TO PERSRPT
    STORE COUNT + 1 TO COUNT
  ENDIF
CASE 1 (MUNIT) = 'E' AND (MPLAT) = 'E'
  IF COMPANY = TWOCCO
    TO PERSRPT
    STORE COUNT + 1 TO COUNT
  ENDIF
CASE 1 (MUNIT) = 'E' AND (MPLAT) = 'A'
  IF COMPANY = TWOCCO AND PLATOON = ONEPLT
    TO PERSRPT
    STORE COUNT + 1 TO COUNT
  ENDIF
CASE 1 (MUNIT) = 'E' AND (MPLAT) = 'B'
  IF COMPANY = TWOCCO AND PLATOON = TWOPLT
    TO PERSRPT
    STORE COUNT + 1 TO COUNT
  ENDIF

161
CASE !(MUNIT) = 'E' .AND. !(MPLAT) = 'C'
  IF COMPANY = TCC0 .AND. PLATOON = THREEPLT
  DO PERSRPT
  STORE COUNT + 1 TO CCOUNT
  ENDP
CASE !(MUNIT) = 'E' .AND. !(MPLAT) = 'D'
  IF COMPANY = TCC0 .AND. PLATOON = FOURPLT
  DO PERSRPT
  STORE COUNT + 1 TO CCOUNT
  ENDP
CASE !(MUNIT) = 'C' .AND. !(MPLAT) = 'E'
  IF COMPANY = TCC0
  DO PERSRPT
  STORE COUNT + 1 TO CCOUNT
  ENDP
CASE !(MUNIT) = 'C' .AND. !(MPLAT) = 'A'
  IF COMPANY = TCC0 .AND. PLATOON = ONEPLT
  DO PERSRPT
  STORE CCPUNT + 1 TO CCOUNT
  ENDP
CASE !(MUNIT) = 'C' .AND. !(MPLAT) = 'B'
  IF COMPANY = TCC0 .AND. PLATOON = TWOPLT
  DO PERSRPT
  STORE CCOUNT + 1 TO CCOUNT
  ENDP
CASE !(MUNIT) = 'C' .AND. !(MPLAT) = 'C'
  IF COMPANY = TCC0 .AND. PLATOON = THREEPLT
  DO PERSRPT
  STORE CCOUNT + 1 TO CCOUNT
  ENDP
CASE !(MUNIT) = 'C' .AND. !(MPLAT) = 'D'
  IF COMPANY = TCC0 .AND. PLATOON = FOURPLT
  DO PERSRPT
  STORE CCOUNT + 1 TO CCOUNT
  ENDP
CASE !(MUNIT) = 'C' .AND. !(MPLAT) = 'E'
  IF COMPANY = TCC0 .AND. PLATOON = THREEPLT
  DO PERSRPT
  STORE CCOUNT + 1 TO CCOUNT
  ENDP
CASE !(MUNIT) = 'C' .AND. !(MPLAT) = 'A'
  IF COMPANY = TCC0 .AND. PLATOON = ONEPLT
  DO PERSRPT
  STORE CCOUNT + 1 TO CCOUNT
  ENDP
CASE !(MUNIT) = 'C' .AND. !(MPLAT) = 'B'
  IF COMPANY = TCC0 .AND. PLATOON = TWOPLT
  DO PERSRPT
  STORE CCOUNT + 1 TO CCOUNT
  ENDP
CASE !(MUNIT) = 'C' .AND. !(MPLAT) = 'C'
  IF COMPANY = TCC0 .AND. PLATOON = THREEPLT
  DO PERSRPT
  STORE CCOUNT + 1 TO CCOUNT
  ENDP
CASE !(MUNIT) = 'E' .AND. !(MPLAT) = 'C'
  IF COMPANY = TCC0 .AND. PLATOON = THREEPLT
  DO PERSRPT
  STORE CCOUNT + 1 TO CCOUNT
  ENDP
ENDCASE
IF CCOUNT = 4
  @ 22.0
  ACCEPT "PRESS 'ENTER' TO CONTINUE" TO PROCEED
  STORE 0 TO CCOUNT
  END
ENDDC
RELEASE COUNT, PROCEED
GSP
RETURN
Q. PERSONNEL DATA FORMAT ROUTINE

* ROUTINE NAME: PERSDATA.PRG
* MODULE NAME: STANDARD REPORT GENERATOR
* VERSION: 5.10.1.0
* DATE: 21NOV83
* VARIABLES USED: COUNT
* VARIABLES MODIFIED: NONE
* VARIABLES CREATED: NONE
* VARIABLES RELEASED: NONE
* FILES (OPENED/CLOSED): NONE
* USING SUBROUTINES: FERSDATA.EFG
* 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.EFG.

COUNT 6.0 SAY "NAME"
5 SAY NAME
42 SAY "SSN"
47 SAY SSN
5.0 SAY "RANK"
5.0 SAY RANK
5.0 SAY "PRIMARY MOS"
3 SAY PRIMEMOS
5.0 SAY "SECONDARY MOS"
4.0 SAY SECMSOS
1.0 SAY "COMPANY"
8 SAY COMPANY
7.0 SAY "PLATOON"
7.0 SAY PLATOON
4.0 SAY "JOIN DATE"
6.2 SAY JOINDATE
6.0 SAY "EAS"
6.0 SAY EAS
1.0 SAY "BIRTHDATE"
7 SAY BIRTHDATE
2 SAY "HEIGHT"
2 SAY "WEIGHT"
5.0 SAY WEIGHT
6.2 SAY "COMMENTS:"
E. EST STATUS ROUTINE

* ROUTINE NAME: ESTSTAT.FRG
* MODULE NAME: STANDARD REPORT GENERATOR
* VERSION: 5.1.0
* AUTHORS: R. E. PRUITT
* DATE: 22DEC83
* VARIABLES USED: MUNIT, MPLAT, ONECO, TWOCO, THREECO,
*                  POUCO, GFITL, TWOCOFL, THREECOFL, FOURPLT, ZZTOTAL, ZZCOC,
*                  ZZHIS, ZZCOD, ZZINT, ZZAID, ZZUNI, ZZNBC, ZZNKS, ZTTAC,
*                  ZETPT, ZZPT2
* VARIABLES MODIFIED: NONE
* VARIABLES CREATED: PROCEED
* VARIABLES RELEASED: ZZ??????
* FILES (OPENED/CLOSED): EST, A:EPERS (ALL FILES CLOSED)
* TEMP FILES CREATED: TEMP (CREATED AND DELETED)
* USING SUBROUTINES: STDERR.FRG
* DESCRIPTION: THIS ROUTINE CREATES A TEMPORARY FILE THAT
* INCLUDES THE ALL ESSENTIAL SUBJECT AND FET FIELDS FROM
* THE EST REP FOR MEMBERS IN A USER SPECIFIED UNIT. FROM
* THIS FILE IT COUNTS THE TOTAI NUMBER OF 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.

ERASE
DO CASE
  CASE !(MUNIT) = 'E'
    USE P:EST
    DC CNTEST
  CASE !(MUNIT) = 'A'.AND. !(MPLAT) = 'E'
    SELECT PRIMARY
    USE A:EPERS
    SELECT SECONDARY
    USE P:EST
    SELECT PRIMARY
    JOIN TO TEMP PCE (P:COMPANY=ONECO .AND. P:SSN=SSN):
    FIELDS COC, HIS, COD, INT, AID, UNI, PFT1CLSS, PFT2CLSS,
    NEC, MKS, TAC
    USE TEMP
    DC CNTEST
    DELETE FILE TEMP
    CASE !(MUNIT) = 'A'.AND. !(MPLAT) = 'A'
    SELECT PRIMARY
    USE A:EPERS
    SELECT SECONDARY
    USE P:EST
    SELECT PRIMARY
    JOIN TO TEMP PCE (P:COMPANY =ONECO .AND. P:PLACCN =:
    CNEPTL .AND. P:SEN=SSN) FIELDS COC, HIS, COD, INT, AID,:
    UNI, PFT1CLSS, PFT2CLSS, NBC, MKS, TAC
    USE TEMP
    DC CNTEST
    DELETE FILE TEMP
    CASE !(MUNIT) = 'A'.AND. !(MPLAT) = 'B'
    SELECT PRIMARY
    USE A:EPERS
    SELECT SECONDARY
    USE P:EST
    SELECT PRIMARY
    JOIN TO TEMP PCE (P:COMPANY =ONECO .AND. P:PLACCN =:
    TWOCOFL .AND. P:SSN=S.SSNN FIELDS COC, HIS, COD, INT,:
    UNI, PFT1CLSS, PFT2CLSS, NBC, MKS, TAC
    USE TEMP
    DC CNTEST
    DELETE FILE TEMP
    CASE !(MUNIT) = 'A'.AND. !(MPLAT) = 'C'
DO CN'TEST
DELETE FILE TEMP
CASE 1 (MUNIT) = 'E' .AND. 1 (MFLAT) = 'D'
SELECT PRIMARY
USE A: PERS
SELECT SECONDARY
USE B: EST
SELECT PRIMARY
JOIN TO TEMP PCE (P. COMPANY = TWOECO .AND. P. PLATOON = :
TWOPLT . AND. F. SSN=S.SSN) FIELDS COC, HIS, COD, INT;
AIC, UNI, PPT1CLSS, PPT2CLSS, NBC, MKS, TAC
USE TEMP
DO CN'TEST
DELETE FILE TEMP
CASE 1 (MUNIT) = 'C' .AND. 1 (MFLAT) = 'E'
SELECT PRIMARY
USE A: PERS
SELECT SECONDARY
USE B: EST
SELECT PRIMARY
JOIN TO TEMP PCE (P. COMPANY = THREECO .AND. P. SSN=:
THREEPLT . AND. F. SSN=S.SSN) FIELDS COC, HIS, COD, INT;
AIC, UNI, PPT1CLSS, PPT2CLSS, NBC, MKS, TAC
USE TEMP
DO CN'TEST
DELETE FILE TEMP
CASE 1 (MUNIT) = 'C' .AND. 1 (MFLAT) = 'A'
SELECT PRIMARY
USE A: PERS
SELECT SECONDARY
USE B: EST
SELECT PRIMARY
JOIN TO TEMP PCE (P. COMPANY = THREECO .AND. P. PLATOON;
THREEPLT . AND. F. SSN=S.SSN) FIELDS COC, HIS, COD, INT;
AIC, UNI, PPT1CLSS, PPT2CLSS, NBC, MKS, TAC
USE TEMP
DO CN'TEST
DELETE FILE TEMP
CASE 1 (MUNIT) = 'C' .AND. 1 (MFLAT) = 'B'
SELECT PRIMARY
USE A: PERS
SELECT SECONDARY
USE B: EST
SELECT PRIMARY
JOIN TO TEMP PCE (P. COMPANY = THREECO .AND. P. PLATOON;
THREEPLT . AND. F. SSN=S.SSN) FIELDS COC, HIS, COD, INT;
AIC, UNI, PPT1CLSS, PPT2CLSS, NBC, MKS, TAC
USE TEMP
DO CN'TEST
DELETE FILE TEMP
CASE 1 (MUNIT) = 'C' .AND. 1 (MFLAT) = 'C'
SELECT PRIMARY
USE A: PERS
SELECT SECONDARY
USE B: EST
SELECT PRIMARY
JOIN TO TEMP PCE (P. COMPANY = THREECO .AND. P. PLATOON;
THREEPLT . AND. F. SSN=S.SSN) FIELDS COC, HIS, COD, INT;
AIC, UNI, PPT1CLSS, PPT2CLSS, NBC, MKS, TAC
USE TEMP
DO CN'TEST
DELETE FILE TEMP
CASE 1 (MUNIT) = 'C' .AND. 1 (MFLAT) = 'D'
SELECT PRIMARY
USE A: PERS
SELECT SECONDARY
USE B: EST
SELECT PRIMARY
JOIN TO TEMP PCE (P. COMPANY = THREECO .AND. P. PLATOON;
= $FOUEPLT .AND. P.SSN=S.SSN) FIELDS COC,HIS,CCD,;
HIS,CCD,INT,AID,UNI,PFT1CLASS,PFT2CLASS,NBC,MLS,TAC
USE TEMP
DO CNTST
DELETE FILE TEMP
CASE I(MUNIT) = 'C' .AND. !(MPLAT) = 'E'
SELECT PRIMARY
USE A:PERS
SELECT SECONDARY
USE E:EST
SELECT PRIMARY
JOIN TO TEMP PGR (P.COMPANY =FOURCO .AND.;
P.SSN=S.SSN) FIELDS COC,HIS,CCD,INT,AID,UNI,PFT1CLASS,;
PFT2CLASS,NBC,MLS,TAC
USE TEMP
DO CNTST
DELETE FILE TEMP
CASE I(MUNIT) = 'C' .AND. !(MPLAT) = 'A'
SELECT PRIMARY
USE A:PERS
SELECT SECONDARY
USE E:EST
SELECT PRIMARY
JOIN TO TEMP PGR (P.COMPANY =FOURCO .AND. P.PLATCCN;
=FOUPRT. .AND. P.SSN=S.SSN) FIELDS COC,HIS,CCD,INT;
AID,UNI,PFT1CLASS,PFT2CLASS,NBC,MLS,TAC
USE TEMP
DO CNTST
DELETE FILE TEMP
CASE I(MUNIT) = 'C' .AND. !(MPLAT) = 'B'
SELECT PRIMARY
USE A:PERS
SELECT SECONDARY
USE E:EST
SELECT PRIMARY
JOIN TO TEMP PGR (P.COMPANY =FOURCO .AND. P.PLATCCN;
=FOUPRT. .AND. P.SSN=S.SSN) FIELDS COC,HIS,CCD,;
INT,AID,UNI,PFT1CLASS,PFT2CLASS,NBC,MLS,TAC
USE TEMP
DO CNTST
DELETE FILE TEMP
CASE I(MUNIT) = 'C' .AND. !(MPLAT) = 'C'
SELECT PRIMARY
USE A:PERS
SELECT SECONDARY
USE E:EST
SELECT PRIMARY
JOIN TO TEMP PGR (P.COMPANY =FOURCO .AND. P.PLATCCN;
=FOUPRT. .AND. P.SSN=S.SSN) FIELDS COC,HIS,CCD,;
INT,AID,UNI,PFT1CLASS,PFT2CLASS,NBC,MLS,TAC
USE TEMP
DO CNTST
DELETE FILE TEMP
CASE I(MUNIT) = 'C' .AND. !(MPLAT) = 'D'
SELECT PRIMARY
USE A:PERS
SELECT SECONDARY
USE E:EST
SELECT PRIMARY
JOIN TO TEMP PGR (P.COMPANY =FOURCO .AND. P.PLATCCN;
=FOUPRT. .AND. P.SSN=S.SSN) FIELDS COC,HIS,CCD,;
INT,AID,UNI,PFT1CLASS,PFT2CLASS,NBC,MLS,TAC
USE TEMP
DO CNTST
DELETE FILE TEMP
ENDCASE
0,0 SAY "EST STATES REPORT"
2,0 SAY "CODE OF CONDUCT..."
2.52 SAY (ZZTOTAL-ZZCOD)/ZZTOTAL*100 USING '999'
3.30 SAY "PERCENT COMPLETED..."
4.25 SAY (ZZTOTAL-ZZHIS)/ZZTOTAL*100 USING '999'
5.52 SAY "PERCENT COMPLETED..."
6.30 SAY "CLOSER ORDER DRILL..."
5.52 SAY (ZZTOTAL-ZZCOD)/ZZTOTAL*100 USING '999'
6.30 SAY "PERCENT COMPLETED..."
7.0 SAY "INTERIOR GUARD..."
8.25 SAY (ZZTOTAL-ZZINT)/ZZTOTAL*100 USING '999'
9.30 SAY "PERCENT COMPLETED..."
10.52 SAY ZZTOTAL-ZZINT USING '999'
11.0 SAY "FIRST AID...
11.25 SAY (ZZTOTAL-ZZAID)/ZZTOTAL*100 USING '999'
12.0 SAY "PERCENT COMPLETED..."
12.25 SAY ZZTOTAL-ZZAID USING '999'
12.52 SAY "EQUIPMENT/UNIFORMS..."
13.30 SAY "PERCENT COMPLETED..."
13.62 SAY ZZTOTAL-ZZUNI USING '999'
14.0 SAY "NBC..."
14.25 SAY (ZZTOTAL-ZZNBC)/ZZTOTAL*100 USING '999'
14.52 SAY ZZTOTAL-ZZNBC USING '999'
15.62 SAY "MARKSMANSHIP..."
16.25 SAY (ZZTOTAL-ZZMKS)/ZZTOTAL*100 USING '999'
16.52 SAY ZZTOTAL-ZZMKS USING '999'
17.25 SAY "INDIVID-TAC MEASURES..."
18.30 SAY "PERCENT COMPLETED..."
18.62 SAY ZZTOTAL-ZZTAC USING '999'
20.0 SAY "PFT1...
20.24 SAY ((ZZTOTAL-ZZPFT1)/ZZTOTAL)*100 USING '999'
20.52 SAY "PERCENT COMPLETED PFT ONE..."
22.62 SAY (ZZTOTAL-ZZPFT1) USING '999'
22.8 SAY "PFT2...
22.24 SAY ((ZZTOTAL-ZZPFT2)/ZZTOTAL)*100 USING '999'
22.52 SAY "PERCENT COMPLETED PFT TWO..."
22.8 SAY ZZTOTAL-ZZPFT2 USING '999'
ACCEPT "PRESS ENTER TO CONTINUE" TO PROCEED
RELEASE ALL LIKE ZZ???????
USING '999'
RETURN
S. EST COUNT ROUTINE

* ROUTINE NAME: CNTESI.PRG
* MODULE NAME: STANDARD REPORT GENERATOR
* VERSION: 5.11.1.0
* AUTHOR: B.E. PEUIETT
* DATE: 12DEC83
* VARIABLES USED: ZZTCTAL, ZZCCC, ZZHIS, ZZCOD, ZZINT,
  ZZAIIT, 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. COUNTS
  THE NUMBER OF TRAINING CATEGORIES THAT CURRENTLY INCLUDE
  ONE OR MORE TRAINING ELEMENTS THAT HAVE NOT BEEN
  COMPLETED.

COUNT TO ZZTOTAL
COUNT FOR "**$ (COC) TO ZZCO C
COUNT FOR "**$ (HIS) TO ZZHIS
COUNT FOR "**$ (COD) TO ZZCOD
COUNT FOR "**$ (INT) TO ZZINT
COUNT FOR "**$ (AID) TO ZZAIIT
COUNT FOR "**$ (UNI) TO ZZUNI
COUNT FOR "**$ (NEC) TO ZZNBC
COUNT FOR "**$ (MKS) TO ZZMKs
COUNT FOR "**$ (TAC) TO ZZTAC
COUNT FOR "**$ (PFT1CLSS) TO ZZPFT1
COUNT FOR "**$ (PFT2CLSS) TO ZZPFT2
RETURN
I. EST ESTER ROUTINE

* ROUTINE NAME: ESTESTR.CRT.PRG
* MODULE NAME: STANDARD REPORT GENERATOR
* VERSION: 5.12.0
* AUTHOR: R. E. PROVIT
* DATE: 20DEC83
* VARIABLES USED: MUNIT, MPLAT, ONECO, TWOCO, THREECO, FOURCO
* VARIABLES MODIFIED: NONE
* VARIABLES RELEASED: COUNT, PROCEED
* FILES (OPENED/CLOSED): PRIMARY INDEX A:ALPHAPERS, E:EST
* INDEX E:ESTSSN (ALL FILES ARE CLOSED).
* TEMP FILES CREATED: VEHICLE CONSTRUCTION, VEHICLE DESTROYED
* USING SUBROUTINES: STDRPT.PRG
* DESCRIPTION: THIS IS A CONTROL ROUTINE WHICH CREATES THE PERSONNEL POSTER FOR THE UNIT SPECIFIED BY THE USER'S RESPONSE TO THE SELECT UNIT MENU (5.1.0).

ERASE
SELECT PRIMARY
USE A:EST INDEX A:ALPHAPERS
SELECT SECONDARY
USE E:EST INDEX E:ESTSSN
SELECT PRIMARY
GOTO TOP
STORE COUNT TO COUNT
DO WHILE .NOT. EOF
DO CASE
CASE I (MUNIT) = 'E'
DO ESTRT
STORE COUNT + 1 TO COUNT
ENDIP
CASE I (MUNIT) = 'A' .AND. I (MPLAT) = 'E'
IF COMPANY = ONECO
DO ESTRT
STORE COUNT + 1 TO COUNT
ENDIP
CASE I (MUNIT) = 'A' .AND. .NOT. (MPLAT) = 'A'
IF COMPANY = ONECO .AND. PLATOON = ONEPLT
DO ESTRT
STORE COUNT + 1 TO COUNT
ENDIP
CASE I (MUNIT) = 'A' .AND. .NOT. (MPLAT) = 'B'
IF COMPANY = ONECO .AND. PLATOON = TWOPLT
DO ESTRT
STORE COUNT + 1 TO COUNT
ENDIP
CASE I (MUNIT) = 'A' .AND. .NOT. (MPLAT) = 'C'
IF COMPANY = ONECO .AND. PLATOON = THREEPLT
DO ESTRT
STORE COUNT + 1 TO COUNT
ENDIP
CASE I (MUNIT) = 'A' .AND. .NOT. (MPLAT) = 'D'
IF COMPANY = ONECO .AND. PLATOON = FOURPLT
DO ESTRT
STORE COUNT + 1 TO COUNT
ENDIP
CASE I (MUNIT) = 'E' .AND. .NOT. (MPLAT) = 'E'
IF COMPANY = TWOCCO
DO ESTRT
STORE COUNT + 1 TO COUNT
ENDIP
CASE I (MUNIT) = 'E' .AND. .NOT. (MPLAT) = 'A'
IF COMPANY = TWOCCO .AND. PLATOON = ONEPLT
DO ESTRT
STORE COUNT + 1 TO COUNT
ENDIP

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

172
STORE 0 TO COUNT
INDIF
SELECT PRIMARY
SKIP
ENDDC
RELEASE COUNT, PROCEED
ELSE
RETURN
U. EST REPORT FORMAT ROUTINE

* ROUTINE NAME: ESTREP.PRG
* MODULE NAME: STANDARD REPORT GENERATOR
* VERSION: 5.12.1.0
* AUTHOR: R.L. PEULIT
* DATE: 14NCV83
* VARIABLES USED: NONE
* VARIABLES MODIFIED: NONE
* VARIABLES CREATED: ROW, Mسس
* VARIABLES RELEASED: ROW
* FILES OPENED/CLOSED: NONE
* TEMP FILES CREATED: NONE
* USING SUBROUTINES: ESTROST.PRG
* DESCRIPTION: THIS ROUTINE USES FORMATS AND DISPLAYS THE EST ROUTINE REPORT.
* STORE (CCOUNT * 6) TO RCW
  @ ROW, 0 SAY NAME
  @ ROW + 30 SAY RANK
  @ ROW + 33 SAY COMPANY
  STORE SSN TC Mسس
  SELECT SECONDARY
  FIND Mسس
  @ ROW + 1, 0 SAY "CODE OF CONDUCT"
  @ ROW + 1, 16 SAY "CONDUCT"
  @ ROW + 1, 22 SAY "MANCOR HIS"
  @ ROW + 1, 35 SAY "HIS"
  @ ROW + 1, 43 SAY "MARKSMANSHIP"
  @ ROW + 1, 56 SAY "MKS"
  @ ROW + 1, 67 SAY "DRILL"
  @ ROW + 1, 74 SAY "COD"
  @ ROW + 4, 0 SAY "INTERIOR GUARD"
  @ ROW + 4, 16 SAY "INT"
  @ ROW + 4, 22 SAY "MARKSMANSHIP"
  @ ROW + 4, 35 SAY "MKS"
  @ ROW + 4, 43 SAY "TAC MEASURES"
  @ ROW + 4, 56 SAY "TAC"
  @ ROW + 4, 67 SAY "MBC"
  @ ROW + 4, 74 SAY "MBC"
  @ ROW + 3, 0 SAY "PFT1"
  @ ROW + 3, 6 SAY "PFT1CISS"
  @ ROW + 3, 10 SAY "PFT2"
  @ ROW + 3, 16 SAY "PFT2CISS"
  @ ROW + 4, 0 SAY "EQUIP/UNIFORM"
  @ ROW + 4, 36 SAY "UNI"
  @ ROW + 3, 4 SAY "FIRST AID"
  RELEASE ROW
  RETURN
V. SWIM QUALIFICATION ROUTINE

* ROUTINE NAME: SWIMQUAL.PRG
* MODULF NAME: STANDARD REPORT GENERATOR
* VERSION: 5.13.0
* AUTHOR: R. E. Pruitt
* DATE: 20DEC83
* VARIABLES USED: MUNIT, MPLAT, ONECO, TWOPLT, THREEPLT, FOURPLT
* VARIABLES MODIFIED: NONE
* VARIABLES CREATED: PROCEED, COUNT, S1TOTAL, S2TOTAL,
* S3TOTAL, UQTOTAL
* VARIABLES RELEASED: COUNT, PROCEED, MSN, S1TOTAL, S2TOTAL,
* S3TOTAL, UQTOTAL, RCW
* FILES OPended/CLOsed: A:INDEX A:ALPHERS
* 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 STATIS. AT THE END OF THE REPORT, SUMMARY
* DATA IS DISPLAYED WHICH INCLUDES THE TOTALS FOR EACH
* QUALIFICATION CARY AND THE OVERALL TOTAL FOR THE
* UNIT.

ERASE
@ 12.27 SAY "SWIM QUALIFICATION REPORT"
@ 22.27 ACCEPT "PRESS 'ENTER' KEY TO CONTINUE" TO PROCEED
ERASE
SELECT PRI\MARY
USE A:INDEX A:ALPHERS
CUTO TC
STORE 0 TO S1TOTAL
STORE 0 TO S2TOTAL
STORE 0 TO S3TOTAL
STORE 0 TO UQTOTAL
STORE 0 TO COUNT
LO WHILE .NCT. EOF
DO CASE
CASE !(MUNIT) = 'F'
  DC SWIMRPT
  STORE COUNT + 1 TO COUNT
CASE !(MUNIT) = 'A'. AND. !(MPLAT) = 'F'
  IF COMPANY = ONECO
  DO SWIMRPT
  STORE COUNT + 1 TO COUNT
  ENDF
CASE !(MUNIT) = 'A'. AND. !(MPLAT) = 'A'
  IF COMPANY = ONECO .AND. PLATOON = ONEPLT
  DO SWIMRPT
  STORE COUNT + 1 TO COUNT
  ENDF
CASE !(MUNIT) = 'A'. AND. !(MPLAT) = 'B'
  IF COMPANY = ONECO .AND. PLATOON = TWOPLT
  DO SWIMRPT
  STORE COUNT + 1 TO COUNT
  ENDF
CASE !(MUNIT) = 'A'. AND. !(MPLAT) = 'C'
  IF COMPANY = ONECO .AND. PLATOON = THREEPLT
  DO SWIMRPT
  STORE COUNT + 1 TO COUNT
  ENDF
CASE !(MUNIT) = 'A'. AND. !(MPLAT) = 'D'
  IF COMPANY = ONECO .AND. PLATOON = FOURPLT
  DO SWIMRPT
  STORE COUNT + 1 TO COUNT
ENDF

175
CASE I(MUNIT) = 'E' .AND. !(MPLAT) = 'E'
  IF COMPANY = TWCOCO
    DO SWIMRPT
    STORE CCUNT + 1 TO CCUNT
  ENDFP
CASE I(MUNIT) = 'E' .AND. !(MPLAT) = 'A'
  IF COMPANY = TWCOCO .AND. PLATOON = ONEPLT
    DO SWIMRPT
    STORE CCUNT + 1 TO CCUNT
  ENDFP
CASE I(MUNIT) = 'E' .AND. !(MPLAT) = 'B'
  IF COMPANY = TWCOCO .AND. PLATOON = TWOPLT
    DO SWIMRPT
    STORE CCUNT + 1 TO CCUNT
  ENDFP
CASE I(MUNIT) = 'E' .AND. !(MPLAT) = 'C'
  IF COMPANY = TWCOCO .AND. PLATOON = THREEPLT
    DO SWIMRPT
    STORE CCUNT + 1 TO CCUNT
  ENDFP
CASE I(MUNIT) = 'E' .AND. !(MPLAT) = 'D'
  IF COMPANY = TWCOCO .AND. PLATOON = FOURPLT
    DO SWIMRPT
    STORE CCUNT + 1 TO CCUNT
  ENDFP
CASE I(MUNIT) = 'E' .AND. !(MPLAT) = 'E'
  IF COMPANY = TWCOCO
    DO SWIMRPT
    STORE CCUNT + 1 TO CCUNT
  ENDFP
CASE I(MUNIT) = 'C' .AND. !(MPLAT) = 'A'
  IF COMPANY = TWCOCO .AND. PLATOON = ONEPLT
    DO SWIMRPT
    STORE CCUNT + 1 TO CCUNT
  ENDFP
CASE I(MUNIT) = 'C' .AND. !(MPLAT) = 'B'
  IF COMPANY = TWCOCO .AND. PLATOON = TWOPLT
    DO SWIMRPT
    STORE CCUNT + 1 TO CCUNT
  ENDFP
CASE I(MUNIT) = 'C' .AND. !(MPLAT) = 'C'
  IF COMPANY = TWCOCO .AND. PLATOON = THREEPLT
    DO SWIMRPT
    STORE CCUNT + 1 TO CCUNT
  ENDFP
CASE I(MUNIT) = 'C' .AND. !(MPLAT) = 'D'
  IF COMPANY = TWCOCO .AND. PLATOON = FOURPLT
    DO SWIMRPT
    STORE CCUNT + 1 TO CCUNT
  ENDFP
CASE I(MUNIT) = 'C' .AND. !(MPLAT) = 'E'
  IF COMPANY = TWCOCO
    DO SWIMRPT
    STORE CCUNT + 1 TO CCUNT
  ENDFP
CASE I(MUNIT) = 'C' .AND. !(MPLAT) = 'A'
  IF COMPANY = TWCOCO .AND. PLATOON = ONEPLT
    DO SWIMRPT
    STORE CCUNT + 1 TO CCUNT
  ENDFP
CASE I(MUNIT) = 'C' .AND. !(MPLAT) = 'B'
  IF COMPANY = TWCOCO .AND. PLATOON = TWOPLT
    DO SWIMRPT
    STORE CCUNT + 1 TO CCUNT
  ENDFP
CASE I(MUNIT) = 'C' .AND. !(MPLAT) = 'C'
  IF COMPANY = TWCOCO .AND. PLATOON = THREEPLT
    DO SWIMRPT
    STORE CCUNT + 1 TO CCUNT
  ENDFP
CASE I(MUNIT) = 'C' .AND. !(MPLAT) = 'D'
  IF COMPANY = TWCOCO .AND. PLATOON = FOURPLT
    DO SWIMRPT
    STORE CCUNT + 1 TO CCUNT
  ENDFP
CASE I(MUNIT) = 'C' .AND. !(MPLAT) = 'E'
  IF COMPANY = TWCOCO
    DO SWIMRPT
    STORE CCUNT + 1 TO CCUNT
  ENDFP
CASE I(MUNIT) = 'C' .AND. !(MPLAT) = 'A'
  IF COMPANY = TWCOCO .AND. PLATOON = ONEPLT
    DO SWIMRPT
    STORE CCUNT + 1 TO CCUNT
  ENDFP
CASE I(MUNIT) = 'C' .AND. !(MPLAT) = 'B'
  IF COMPANY = TWCOCO .AND. PLATOON = TWOPLT
    DO SWIMRPT
    STORE CCUNT + 1 TO CCUNT
  ENDFP
CASE I(MUNIT) = 'C' .AND. !(MPLAT) = 'C'
  IF COMPANY = TWCOCO .AND. PLATOON = THREEPLT
    DO SWIMRPT
    STORE CCUNT + 1 TO CCUNT
  ENDFP
CASE I(MUNIT) = 'C' .AND. !(MPLAT) = 'D'
  IF COMPANY = TWCOCO .AND. PLATOON = FOURPLT
    DO SWIMRPT
    STORE CCUNT + 1 TO CCUNT
  ENDFP
CASE I(MUNIT) = 'C' .AND. !(MPLAT) = 'E'
  IF COMPANY = TWCOCO
    DO SWIMRPT
    STORE CCUNT + 1 TO CCUNT
  ENDFP
CASE I (MUNIT) = 'L' .AND. ! (MPLAT) = 'D'
  IF COMPANY = PCURCO .AND. PLATOON = FOURPLT
     DO SWIMRPT
       STORE CCOUNT + 1 TO CCOUNT
   ENDP
  IF CCOUNT = 7
     4 22 C
     ACCEPT "PRESS 'ENTER' TO CONTINUE" TO PROCEED
     STORE 0 TO CCOUNT
   ENDP
SELECT PMARY
ENDDO
22 C
ACCEPT "PRESS 'ENTER' TO CONTINUE" TO PROCEED
END
4 2 C SAY "SUMMARY DATA"
4 1 SAY "SWIM QUALIFICATION"
6, 25 SAY "TOTAL"
6, 1 SAY "S1"
6, 25 SAY S1TOTAL USING '999'
7, 1 SAY "S2"
7, 25 SAY S2TOTAL USING '999'
8, 1 SAY "S3"
8, 25 SAY S3TOTAL USING '999'
9, 1 SAY "UNQUALIFIED"
9, 25 SAY UQTOTAL USING '999'
10, 25 SAY S1TOTAL+S2TOTAL+S3TOTAL+UQTOTAL USING '999'
12 0
ACCEPT "PRESS 'ENTER' TO CONTINUE" TO PROCEED
RELEASE CCOUNT, PROCEED, MSSTN, S1TOTAL, S2TOTAL, S3TOTAL, UQTOTAL, FROW
USE
RETURN
G. SWIM REPORT ROUTINE

* ROUTINE NAME: SWIMREP.PRG
* MODULE NAME: STANDARD REPORT GENERATOR
* VERSION: E.13.1.0
* AUTHOR: R.E. PRUIETT
* DATE: 12DEC83
* VARIABLES USED:NONE
* VARIABLES MODIFIED:NONE
* VARIABLES CREATED: RCW, MSSN, S1TOTAL, S2TOTAL, S3TOTAL, UQTCTAL
* VARIABLES RELEASED:NONE
* FILES OPENED/CLOSED:QUAL INDEX QUALSSN
* TEMT 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.

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

* ROUTINE NAME: HELP.FRG
* MODULE NAME: STANDARD REPORT GENERATOR
* VERSION: 5.14.0
* AUTHOB: F.E. PRUIETT
* DATE: 18JAN84
* VARIABLES USED: NONE
* VARIABLES MODIFIED: NONE
* VARIABLES CREATED: HELP, PROCEED
* VARIABLES RELEASED: NONE
* FILES OPENED/CLOSED: NONE
* USING ROUTINES: STANDRAD
* 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 INDIVIDUAL TRAINING RECORD"
@ 9,20 SAY "D...HELP MARKSMANSHIP STATUS REPORT"
@ 10,20 SAY "E...HELP MOS STATUS REPORT"
@ 11,20 SAY "F...HELP MCS ROSTER REPORT"
@ 12,20 SAY "G...HELP PERSONAL DATA REPORT"
@ 13,20 SAY "H...HELP EST STATUS REPORT"
@ 14,20 SAY "I...HELP EST ROSTER REPORT"
@ 15,20 SAY "J...HELP SWIM QUALIFICATION REPORT"
@ 22,20 SAY "---------"
ACCEPT "SELECT OPTION ======> " TO HELPCT
ERASE
DO CASE
CASE !(HELPCT) = 'A'
TEXT

OPTION A...UNIT ROSTER
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-E), 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 UNIT ROSTER REPORT will display in alphabetical order
the name, rank, and PRIMARY MOS for each member of the
selected unit or subunit. The screen will scroll
automatically unless stopped from the terminal.

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

CASE !(HELPCT) = 'B'
TEXT

OPTION 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, EFT2, SWIMQUAL, RIFLE QUAL, and PISTOL QUAL.

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

CASE 1(FILPCPT) = 'C'
ENDTEXT

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.

ENDTEXT

CASE 1(FILPCPT) = 'D'
ENDTEXT

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**EF" 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.

CASE 1(FILPCPT) = 'E'
ENDTEXT
OPTION D...INDIVIDUAL TRAINING 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 UNIF).

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 E...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 UNIF).

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., PISTOL, SNIPER) 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

4 22 0

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

END TEXT

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.
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.

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...PERMENATLY 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.

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

CASE !(EFLPCPT) = 'F'

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., BATTALICON) 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.

CASE !(EFLPCPT) = 'G'

OPTION G...MCS 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., BATTALICON) 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.

ENDTEXT

CASE I(FELPCPT) = '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.

ENDTEXT

CASE I(FELPCPT) = '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, BIRTHDATE, HEIGHT, 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.

ENDTEXT

CASE I(FELPCPT) = '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
CASE 23 C
ACCE 2
"PRESS 'ENTER' TO CONTINUE" TO PROCEED
ENDCASE
RETURN
APPENDIX H
MAINTENANCE MODULE LISTING

A. MAINTAIN CONTROL ROUTINE

* Routine Name: Maintain.Prg
* Module Name: Maintenance Module
* Version: 6.0
* Author: L.P. Hauser
* Date: 30 Sep 83
* Variables Used: mdone, mntnopt
* Variables Modified: mdone, mntnopt
* Variables Released: mdone, mntnopt
* Files (opened/closed): none
* Temporary Files Created: none
* Using Subroutines: Cmddet

Description: This routine produces a module which gives
the user the option of selecting maintenance functions to
activate. Each function has an associated level
associated with and only appropriate users may activate
certain modules.

STORE f TO AAAone
DO WHILE .NOT. AAAone
    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 updtitr
        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

186
ELSE
ERASE 8,31 SAY "ACCESS UNAUTHORIZED"
DO pause
ENDIF
CASE !(mntnopt) = 'S'
IF AALevel = 1
DO SYsyst
ELSE
ERASE 8,31 SAY "ACCESS UNAUTHORIZED"
DO pause
ENDIF
CASE !(mntnopt) = 'H'
STORE 2 TO HELP0PT
DO B:HELP2
CASE !(mntnopt) = 'Q'
STORE t TO AAdone
ERASE
ENDCASE
ENDDO
RELEASE mntnopt, HELPCPT
RELEASE AALCNE
RETURN
E. VIEW CONTROL ROUTINE

* Routine Name: Viewitr.prg
* Module Name: Maintenance Module
* Version: 6.1.0
* Author: L.P. Hauser
* Date: 1 Dec 83
* Variables Used: vicne, vfinish, vname, vcomp, again,
  vsn, vend, view
* Variables Modified: same as above
* Variables Created: same as above
* Variables Released: all like v??????
* 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 P TC VNOFIND
STORE P TC VDONE
DO WHILE .NOT. VDONE
  STORE P TO VFINISH
  DC WHILE .NOT. VFINISH .AND. .NOT. VDONE
  ERASE
  @ 8,31 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:ALFPERS
  FIND VNAME
  IF # = 0
    ERASE
    @ 8,31 SAY "INDIVIDUAL IS NOT IN DATA BASE"
    STORE P TO VCOMP
    DO WHILE .NOT. VCCMP
      ACCEPT "DO YOU WANT TO TRY AGAIN (Y/N)"
      TO VAGAIN
      STORE !VAGAIN TO VAGAIN
      DO CASE
        CASE VAGAIN = 'N'
          STORE T TC VICNE
          STORE T TO VCOMP
          STORE T TO VFINISH
        CASE VAGAIN = 'Y'
          STORE T TC VCCMP
          STORE T TO VNOFIND
      ENDCASE
    ENDDO
    ELSE
      STORE SSN TO ASSN
      STORE COMMENT TO STCCMNT
      USE
      STORE P TO VEND
      DO WHILE .NOT. VEND
        STORE T TC VEND
        DO ITRSCN
        DO GZDATA
          @ 24,24 SAY "** PRESS ANY KEY TO CONTINUE **"
          SET CONCILE OFF
          WAIT
          SET CONCILE ON
        STORE P TC VIEW
      ENDDO
      ERASE
IF STCOMENT
STORE F TO VCOMP
DC WHILE NOT VCOMP
ACCEPT " DO YOU WANT TO VIEW COMMENTS:
(Y/N)?" TO 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
ENDCASE
ENDDO
IF VIEW
DO GETCMNT
ENDIF
ENDIF
@ 24,0
STORE T TO VFINISH
STORE P TO VCOMP
DO WHILE .NOT. VCOMP AND VIEW
ACCEPT " DO YOU WANT TO VIEW IT AGAIN:
(Y/N)?" TO VAGAIN
DC CASE
CASE !(VAGAIN) = 'Y'
STORE T TO VCOMP
STORE F TO VEND
CASE !(VAGAIN) = 'N'
STORE T TO VCOMP
STORE T TO VEND
ENDCASE
ENDDO
ENDIF
ENDCASE
ENDDO
ENDCASE
ENDDO
RELEASE ALL EXCEPT AA??????
RETURN
C. ITF SCREEN FORMAT ROUTINE

* Routine Name: Iterscrn.prg
* Module Name: Maintenance Module
* Version: 6.1.0.1
* Author: C.P. Haesler
* 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"
1,66 SAY "NAME:"
1,46 SAY "GRADE:"
1,57 SAY "MOS:"
2,66 SAY "/:
2,8 SAY "BIRTHDATE.....:""""
2,40 SAY "SSN....:""
3,8 SAY "CO:""""
3,18 SAY "PLAT:"
3,33 SAY "JOIN:""""
3,46 SAY "EAS:""""
3,56 SAY "GAS MASK:""""
4,8 SAY "KIL CODE OF CONDUCT:""""
4,40 SAY "PFT1........:"
4,55 SAY "/:
4,61 SAY "DATE:"
5,8 SAY "HISTORY OF MARCOR:""""
5,55 SAY "PFT2........:"
5,61 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.......:""""
10,40 SAY "MEASURES............:""""
11,40 SAY "SKIN QUAL....:""""
11,24 SAY "DATE:"
11,40 SAY "WEIGHT CONTROL......:""""
12,8 SAY "RIFLE QUAL...:""""
12,25 SAY "/:
12,31 SAY "DATE:"
12,49 SAY "HEIGHT......:""""
13,8 SAY "PISTOL QUAL.:"
13,25 SAY "/:
13,31 SAY "DATE:"
13,49 SAY "HEIGHT.......:""""
15,8 SAY "ELECTRONIC WARFARE.......:""""
15,40 SAY "DRUG ABUSE...........:""""
15,8 SAY "DLD WEATHER...........:""""
16,40 SAY "ALCOHOL ABUSE...........:""""
17,40 SAY "LAW O' WAR...........:""""
17,40 SAY "HUMAN RELATIONS...........:""""
18,8 SAY "TRAINING MOS...........
18,4C SAY "PERSONAL AFFAIRS.....:""""
19,8 SAY "LEADERSHIP...........:""""
19,40 SAY "UCMJ...........:""""
@ 20, & SAY "CHAR & MCRAL ED.....:"
@ 21, & SAY "CMMENTS...:"

RETURN
SCREEN DATA RETRIEVAL ROUTINE

* Routine Name: Getdata.prg
* Module Name: Maintenance Module
* Version: 6.1.0.2
* Author: D. P. Haebeler
* 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)
* Temporary Files Created: NONE
* Using Subroutines: Viewitr, Updtitr, Deletitr
* Description: This routine retrieves the information stored in the data base for the individual specified by vssn. It then arranges it in the screen format created by Itscreen.prg.

STORE AASSN TO VSSN
USE A:ESTS INDEX A:PEISSN
FIND &VSSN

1,13 SAY NAME
1,12 SAY BANK
1,61 SAY FRIMEMOS
1,56 SAY SECMMOS
2,23 SAY FERTDATE
2,49 SAY SSN
3,11 SAY COMPANY
3,23 SAY ILAATOON
3,38 SAY JOINDATE
3,51 SAY FAS
3,65 SAY GASMASK
4,26 SAY OCC
4,6 SAY FT1RAW
4,57 SAY FT1CIS
4,66 SAY FT1DATE
5,28 SAY HIS
5,51 SAY FT2RAW
5,57 SAY FT2CIS
5,62 SAY FT2DATE
6,28 SAY COD
6,49 SAY NEC
6,28 SAY INT
7,59 SAY MRS
8,28 SAY AID
8,59 SAY UNI
9,59 SAY TAG

USE B:EST INDEX B:ESISSN
FIND &VSSN

2,6 SAY CIC
4,26 SAY OFF1RAW
4,6 SAY OFF1CIS
4,66 SAY OFF1DATE
5,28 SAY HIS
5,51 SAY OFF2RAW
5,57 SAY OFF2CIS
5,62 SAY OFF2DATE
6,28 SAY COD
6,49 SAY NEC
6,28 SAY INT
7,59 SAY MRS
8,28 SAY AID
8,59 SAY UNI
9,59 SAY TAG

USE B:QUAL INDEX B:QUALSSN
FIND &VSSN

11,20 SAY SWIMQUAL
11,59 SAY SWIMDATE
12,11 SAY RFLSCORE
12,27 SAY RPLQUAL
USE B:INFCMISC INDEX E:INFOSSN
FIND EVSS
USE RETURN

E. COMMENT RETRIEVAL ROUTINE

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

ERASE
@ 5,33 SAY "COMMENT SECTION"
USE A:CMNT INDEX A:CMNTSSN
FIND EAAAA
@ 8,2 SAY INFOTXT1
@ 8,4 SAY INFOTXT2
@ 8,6 SAY INFOTXT3
@ 8,8 SAY INFOTXT4
@ 10,2 SAY INFOTXT5
@ 10,4 SAY INFOTXT6
@ 11,2 SAY INFOTXT7
@ 11,4 SAY INFOTXT8
@ 12,2 SAY INFOTXT9
@ 12,4 SAY INFOTXT10
USE
RETURN
P. UPDATE CONTROL Routines

- Routine Name: Updtitr.prg
- Module Name: Maintenance Module
- Version: 6.2.0
- Author: D.P. Haeusler
- Date: 4 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
- Temporary Files Created: None
- 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" @ 9.23 SAY "UPDATED OR CORRECTED?"
   @ 10.23 SAY "NAME (LAST, FIRST MI.)"
   ACCEPT "NAME ===>" TO UPNAME
   STORE ! (UPNAME) TC UPNAME
   USE A:FILES INDEX A:ALPHABET
   FIND UPNAME
   IF 1 = 0
      ERASE
      @ 6.24 SAY "INDIVIDUAL IS NOT IN THE DATABASE" @ 9.23 SAY "OR YOU HAVE ENTERED AN INCORRECT NAME"
      STORE F TO UPComp
   DO WHILE .NOT. DECOMP
      ACCEPT "DO YOU WANT TO TRY AGAIN (Y/N)?": TO UPAGAIN
      DO CASE
         CASE ! (UPAGAIN) = 'Y'
            STORE 1 TO UPComp
            CASE ! (UPAGAIN) = 'N'
            STORE 1 TO UPComp
            STORE 1 TO AAADONE
      ENDCASE
   ELSE
      STORE SSN TO AASSN
      USE
      DC ITSCREEN
      TO GETDATA
      DC UPTDATA
      STORE F TO UPComp
      DO WHILE .NOT. DECOMP
         ERASE
         ACCEPT "DO YOU WANT TO UPDATE ANOTHER ITR: (Y/N)?": TO UPAGAIN
         DO CASE
            CASE ! (UPAGAIN) = 'Y'
               STORE 1 TO UPComp
            CASE ! (UPAGAIN) = 'N'
               STORE 1 TO UPComp
      ENDCASE
   END ELSE
STORE I TO AAADONE
ENDCASE
ENDDO
ENDDIF
ENDDDC
RELEASE ALL EXCEPT AA??????
RELEASE AASSN, AATHRU, AACORE, AAADONE
RETURN
G. DATA UPDATE ROUTINE

* Routine Name: Update.prg
* Module Name: Maintenance Module
* Version: 6.2.0.1
* Author: D. F. Haeusler
* Date: 8 Dec 83

* Variables Used: aatbru, aacorr, upopt, inname, inrank, upccmc5, inptme, incsc, inbtrh, insm, errcpt, inccmp, inplmt, inndate, iness, ingas, incode, inpt1, calccpt, wthru, wdate, inprestick, inprestick2, inswrt, inplmt, inprestick, inhis, inpt2, indrill

* Variables Modified: same as above
* Variables Created: same as above
* Variables Released: all like in???????

* Files (opened/closed): a:pers (opened/closed)
  b:qual (opened/closed)
  b:est (opened/closed)
  b:inmcisc (opened/closed)

* Temporary Files Created: none
* Using Subroutines: updtitr.prg
* Description: This routine prompts the user to choose a specific field of the ITR to update. It then locates that field in the database and replaces the old data with new data. It also calls error checking routines and a calculate function.

STORE E TO AATHRU
DO WHILE .NOT. AATHRU
  STORE E TO AACORR
  RELEASE ALL LIKE IN???????
  STORE " " TO UPOPT
  @ 44,0
  @ 44,0
  @ 44,0
  @ 22,1
  SAY " OPTION: FIELD TO BE CHANGED OR 'Q'; TO QUIT."
  @ 23,1
  SAY " OPTION = >>" GET UPOPT PICTURE "XXXXXXX; XXXXXXXXXXXXXXXXXXXXXXX"
  READ
  SICBE ! (UPCPT) TO UPOFT
  LO CASE
  CASE UPCPT = 'NAME'
    STORE T TO AACORR
    USE A:PEBS INDEX A:PEBSN, A:ALPHPERS
    FIND &AASSN
    STORE NAME TO INNAME
    @ 7,12 GET INNAME PICTURE "XXXXXXXXXXXXXXXXXXXXX;
    XXXXXX11111111"
    READ
    REPLACE NAME WITH ! (INNAME)
    USE
  CASE UPCPT = 'GRADE'
    STORE T TO AACORR
    USE A:PEBS INDEX A:PEBSN
    FIND &AASSN
    STORE RANK TO INRANK
    @ 1,1 GET INRANK PICTURE "XX"
    READ
STORE 1 TO ERROR
DO ERROR
REPLACE BANK WITH !(INRANK)
USE

CASE UPOPT = 'NOS'
STORE TC TO AACORR
@ 23,0
@ 24,0
STORE P TO UPCROMOS
DO WHILE NOT. UPCROMOS
@ 23,1 SAY "PRIMARY OR SECONDARY?"
ACCEPT "ENTER P OR S ==>>" TO UPAGAIN
DO CASE
CASE !(UPAGAIN) = 'P'
STORE T TO UPCROMOS
USE A:PERSS INDEX A:PERSN , A:MOSNEX
FIND &AASSN
STORE PRIMEMOS TO INPRIME
@ 1,60 GET INPRIME PICTURE "9999"
READ
REPLACE PRIMEMOS WITH !(INPRIME)
USE

CASE !(UPAGAIN) = 'S'
STORE T TO UPCROMOS
USE A:PERSS INDEX A:PERSN
FIND &AASSN
STORE SECMS TO INSEC
@ 1,67 GET INSEC PICTURE "9999"
READ
REPLACE SECMS WITH INSEC
USE
ENDCASE
ENDDO

CASE UPOPT = 'BIRTHDATE'
STORE T TO AACORR
USE A:PERSS INDEX A:PERSN
FIND &AASSN
STORE BIRTHDATE TO INBIRTH
@ 2,22 GET INBIRTH PICTURE "XXXXX"
READ
STORE 2 TO ERROR
DO ERROR
REPLACE ERROR WITH INBIRTH
USE

CASE UPOPT = 'SSN'
STORE T TO AACORR
USE A:PERSS INDEX A:PERSN
FIND &AASSN
STORE SSN TO INSSN
@ 2,48 GET INSSN PICTURE "999999999"
READ
REPLACE SSN WITH INSSN
USE
USE A:CMNT INDEX A:CMNMTSSN
FIND &AASSN
REPLACE SSN WITH INSSN
USE B:EST INDEX B:ESTSSN
FIND &AASSN
REPLACE SSN WITH INSSN
USE B:QUAL INDEX B:QUALSSN
FIND &AASSN
REPLACE SSN WITH INSSN
USE B:INFO INDEX B:INFOSSN
FIND &AASSN
REPLACE SSN WITH INSSN

198
STORE INSSN TO AASSN
USE

CASE UPCPT = 'CO'
STORE T TC AACORR
USE A: PERS INDEX A: PERSSN
FIND GAAASSN
STORE COMPANY TC INCCMP
@ 3, 10 GET INCCMP PICTURE "XXXXXX"
READ
REPLACE COMPANY WITH !(INCCMP)
USE

CASE UPOPT = 'PLAT'
STORE T TC AACORR
USE A: PERS INDEX A: PERSSN
FIND GAAASSN
STORE PLAT TO INPLAT
@ 3, 22 GET INPLAT PICTURE "XXXXXXXXX"
READ
REPLACE PLATTOON WITH !(INPLAT)
USE

CASE UPCPT = 'JCIN'
STORE T TC AACORR
USE A: PERS INDEX A: PERSSN
FIND GAAASSN
STORE JICINDATE TO INJNDATE
@ 3, 37 GET INJNDATE PICTURE "XXXXXX"
READ
STORE 3 TC ERCOPT
DO ERROR
REPLACE JCINDATE WITH INJNDATE
USE

CASE UPOPT = 'EAS'
STORE T TC AACORR
USE A: PERS INDEX A: PERSSN
FIND GAAASSN
STORE EAS TO INEAS
@ 3, 50 GET INEAS PICTURE "XXXXXX"
READ
REPLACE EAS WITH INEAS
USE

CASE UPOPT = 'GASMASK'
STORE T TC AACORR
USE A: PERS INDEX A: PERSSN
FIND GAAASSN
STORE GASMASK TC INGAS
@ 3, 68 GET INGAS PICTURE "X"
READ
STORE 4 TC ERCOPT
DO ERROR
REPLACE GASMASK WITH !(INGAS)
USE

CASE UPOPT = 'MIL CODE OF CONDUCT'
STORE T TC AACORR
USE B: ESI INDEX B: ESISSN
FIND GAAASSN
STORE COC TO INCODE
@ 4, 27 GET INCODE PICTURE "XXX"
READ
REPLACE CCC WITH !(INCODE)
USE

CASE UPCPT = 'FPT1'
STORE T TC AACORR
USE B:EST INDEX E:ESTSSN
FIND &AASSN
STORE PFT1DATE TO INPFT1
@4,50 GET INPFT1 PICTURE "999"
READ
STORE 5 TO ERROPT
DO ERROR
REPLACE PFT1RAW WITH INPFT1
USE
STORE 1 TO CALCCT
DO CALC
CASE UPOPT = 'DATE'
STORE E TO WTHRU
STORE T TO AACORR
DO WHILE .NOT. WTHRU
@23,0
@24,0
@23,1 SAY "A..PFT1, B..PFT2, C..SWIM,
D..PIFLE, 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 INPFT1
@4,50 GET INPFT1 PICTURE "XXXXX"
READ
STORE 6 TO ERROPT
EC ERROR
REPLACE PFT1DATE WITH INPFT1
USE
CASE WDATE = 'B'
STORE T TO WTHRU
USE B:EST INDEX B:ESTSSN
FIND &AASSN
STORE PFT2DATE TO INPFT2
@5,65 GET INPFT2 PICTURE "XXXXX"
READ
STORE 8 TO ERROPT
EC ERROR
REPLACE PFT2DATE WITH INPFT2
USE
CASE WDATE = 'C'
STORE T TO WTHRU
USE B:QUAL INDEX B:QUALSSN
FIND &AASSN
STORE SWIMDATE TO INSADT
@11,28 GET INSADT PICTURE "XXXXX"
READ
STORE 10 TO ERROPT
EC ERROR
REPLACE SWIMDATE WITH INSADT
USE
CASE WDATE = 'D'
STORE T TO WTHRU
USE B:QUAL INDEX B:QUALSSN
FIND &AASSN
STORE RFLDATE TO INSFLD
@12,35 GET INSFLD PICTURE "XXXXX"
READ
STORE 12 TO ERROPT
EC ERROR
REPLACE FLDATA WITH INPSTD
USE

CASE WDATE = 'E'
SIGN_T TC WTHRU
(DB:QUAL INDEX B:QUALSSN
FIND &ASSN
STORE FLDATA TO INPSTD
@13,35 GET INPSTD PICTURE "XXXXXX"
READ
STORE 15 TC ERROPT
TC ERROR
REPLACE FLDATA WITH INPSTD
USE

ENDCASE
ENDDO

CASE UPCPT = 'HISTORY OF MARCOR'
STORE T TC AACOR
USE B:EST INDEX E:ESTSSN
FIND &ASSN
STORE HIS TO INHIS
@5,27 GET INHIS PICTURE "XXX"
READ
REPLACE HIS WITH INHIS
USE

CASE UPCPT = 'FPT2'
STORE T TC AACOR
USE B:EST INDEX E:ESTSSN
FIND &ASSN
STORE FPT2RAW TO INPFT2
@5,50 GET INPFT2 PICTURE "999"
READ
STORE 7 TC ERROPT
DO ERROR
REPLACE FPT2RAW WITH INPFT2
USE
STORE 2 TC CALCFCPT
DO CALCULAT

CASE UPCPT = 'DRILL'
STORE T TC AACOR
USE B:EST INDEX E:ESTSSN
FIND &ASSN
STORE CCD TO INDFILL
@6,27 GET INDFILL PICTURE "XXX"
READ
REPLACE CCD WITH INDFILL
USE

CASE UPCPT = 'NBC'
STORE T TC AACOR
USE B:EST INDEX E:ESTSSN
FIND &ASSN
STORE NEC TO INNBC
@6,47 GET INNBC PICTURE "XXXXXX"
READ
REPLACE NEC WITH INNBC
USE

CASE UPCPT = 'INTERFACE GUARD'
STORE T TC AACOR
USE B:EST INDEX E:ESTSSN
FIND &ASSN
STORE INT TO ININT
@7,27 GET ININT PICTURE "XXX"
READ
REPLACE INT WITH ININT
USE
CASE UPOPT = 'MARKSMANSHIP'
STORE T TO AACORR
USE B:EST INDEX B:ESTSSN
FIND &AASSN
STORE IMKS TO INMKS
@ 7, 58 GET INMKS PICTURE "XXXXXX"
READ
REPLACE IMKS WITH INMKS
USE
CASE UPOPT = 'FIRST AID'
STORE T TO AACORR
USE B:EST INDEX B:ESTSSN
FIND &AASSN
STORE AID TO INAID
@ 8, 27 GET INAID PICTURE "XXXXXXXXXXX"
READ
REPLACE AID WITH INAID
USE
CASE UPOPT = 'EQUIP & UNIFORM'
STORE T TO AACORR
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 AACORR
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 AACORR
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 AACORR
USE B:QUAL INDEX B:QUALSSN
FIND &AASSN
STORE RIFSCORE TO INRFLSC
@ 12, 20 GET INRFLSC PICTURE "999"
READ
STORE 11 TO ERRCPT
DO ERROR
REPLACE RIFSCORE WITH INRFLSC
USE
STORE 3 TO CALCACPT
DO CALCULAT
CASE UPOPT = 'WEIGHT'

202
STORE T TC AACORR
USE A:PESS INDEX A:PESSN
FIND EASSN
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 CALCCPT
DO CALCULAT

CASE UPOPT = 'PISTCL QUAL'
STORE T TC AACORR
USE B:QUAL INDEX B:QUALSSN
FIND EASSN
STORE PSTSCORE TO INPSTSC
@ 13,20 GET INPSTSC PICTURE "999"
READ
STORE 14 TO ERRCPT
DO ERROR
REPLACE PSTSCORE WITH INPSTSC
USE
STORE 4 TO CALCCPT
DO CALCULAT

CASE UPOPT = 'HEIGHT'
STORE T TC AACORR
USE A:PESS INDEX A:PESSN
FIND EASSN
STORE HEIGHT TO INHGHT
@ 13,58 GET INHGHT PICTURE "999"
READ
STORE 16 TO ERRCPT
DO ERROR
REPLACE HEIGHT WITH INHGHT
USE
STORE 5 TO CALCCPT
DO CALCULAT

CASE UPOPT = 'ELECTRCN WRAFAE'
STORE T TC AACORR
USE B:INFCMISC INDEX B:INFOSSN
FIND EASSN
STORE PLECWHR TO INLEC
@ 15,28 GET INLEC PICTURE "XXXXXX"
READ
REPLACE PLECWHR WITH INLEC
USE

CASE UPOPT = 'DRUG ABUSE'
STORE T TC AACORR
USE B:INFCMISC INDEX B:INFOSSN
FIND EASSN
STORE DRUG TO INDRUG
@ 15,57 GET INDRUG PICTURE "XXXXXX"
READ
REPLACE DRUG WITH INDRUG
USE

CASE UPOPT = 'CLD WEATHER'
STORE T TC AACORR
USE B:INFCMISC INDEX B:INFOSSN
FIND EASSN
STORE CLDWTHR TO INCLDWR
@ 16,28 GET INCLDWR PICTURE "XXXXXX"
READ
REPLACE CLDWTHR WITH INCLDWR

203
REPLACE CCMJ WITH INUCMJ
USE

CASE UPCT = 'CHAR & MCRAL ED'
STORE T TO AACORR
USE E:INVMISC INDEX B:INPOSSN
FIND &AASSN
STORE CHECKED TO INCHRMOR
@ 20, 28 GET INCHRMOR PICTURE "XXXXX"
READ
REPLACE CERMORED WITH INCHRMOR
USE

CASE UPCT = 'COMMENTS'
STORE T TO AACORR
USE A:PERSE INDEX A:PERSSN
FIND &AASSN
IF COMMENT
   DC WRITCMNT
   DC ITSCRN
   DC GETDATA
ELSE
   STORE & TO INCCMMNT
   @ 21, 16 GET INCCMMNT PICTURE "X"
   READ
   REPLACE COMMENT WITH INCCMNT
   USE
   IF INCCMMNT
      DO CRITCMNT
      DO WRITCMNT
      DO ITSCRN
      DO GETDATA
   ENDIF
ENDIF

CASE UPCT = 'O'
STORE T TO AACORR
STORE T TO AATHRU
ENDCASE
ENDDC
ENDC
RETURN
B. ITR CREATION CONTROL ROUTINE

* Routine Name: Creatitr.prg
* Module Name: Maintenance Module
* Version: 6.3.0
* Author: D.F. HAEUSLER
* Date: 7 Dec 83
* Variables Used: crccmp, 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 ITESCN
  EC INITATA
  STORE F TO CRCCMP
  DC WHILE .NOT. CRCCMP
  ERASE
  ACCEPT "DO YOU WANT TO CREATE ANOTHER ITR (Y/N)?"; TO CRAFTAIN
  DO CASE
    CASE ! (CRAFTAIN) = 'Y'
      STORE T TO CRCCMP
    CASE ! (CRAFTAIN) = 'N'
      STORE T TO CRCCMP
      STORE T TO AAFINI
  ENDCASE
ENDDO
ENDDC
RELEASE ALL LIKE CR??????
RELEASE AAFINI
USP
RETURN
I. THE DATA INITIALIZATION ROUTINE

* Routine Name: Indata.prg
* Module Name: Maintenance Module
* Version: 6.3.1
* Author: D.P. Mausler
* 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:ilincmisc(O/C), b:qual(O/C)
* Temporary Files Created: None
* Using Subroutines: Creatitr.prg
* Description: This routine inserts the data in 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 AADATE TO INDATE
STORE "LAST FIRST MI. TO INNAME
STORE "" TO INRANK
STORE "" TO INPREN
STORE "" TO INBIMOS
STORE "DMY" TO INNETH
STORE "" TO INSSN
STORE "" TO INCMP
STORE "" TO INILAT
STORE "DMY" TO INJDATE
STORE "DMY" TO INHASP
STORE "" TO INNGAS
STORE "***" TO INCODE
STORE "000" TO INPIFT1
STORE "DMY" TO INNFDT1
STORE "" TO INLIST
STORE "000" TO INPIFT2
STORE "DMY" TO INNFDT2
STORE "***" TO INDRILL
STORE "***" TO INREC
STORE "***" TO ININT
STORE "***" TO INPKS
STORE "***" TO INNAC
STORE "***" TO INDUC
STORE "" TO INSWQO
STORE "LEMMY" TO INSWTD
STORE "000" TO INPLSIC
STORE "LEMMY" TO INPLDT
STORE "000" TO INWEIGHT
STORE "000" TO INPSIC
STORE "DMYYY" TO INFSDT
STORE "00" TO INPHT
STORE "***" TO INPEIC
STORE "***" TO INPBUG
STORE "***" TO INICDTR
STORE "***" TO INIAIC
STORE "***" TO INIMUM
STORE "***" TO INmos
STORE "***" TO INERSAP
STORE "***" TO INILER
STORE "***" TO INUCMJS
STORE "***" TO INCHRMOR
STORE "" TO INCOMMT
STORE "" TO INFTCCTH
STORE "" TO INPFT1C1
STORE "" TO INPFT2C1
```
STORE "" TO INFLOU
STORE "" TO INFSQU

1 12 GET INNAME PICTURE "XXXXXXXXXXXXXXXXXXXXXXXXX"
1 13 GET INRANK PICTURE "XX"
1 16 GET INPRIM2 PICTURE "9999"
1 17 GET INSECMS PICTURE "9999"
2 18 GET INBIRT PICTURE "XXXXX"
2 19 GET INCOME PICTURE "XXXXX"
2 20 GET INFLAT PICTURE "XXXXX"
3 21 GET INJNDATE PICTURE "XXXXX"
3 22 GET INPRIM2-M PICTURE "9999"
4 23 GET INSECMOS PICTURE "XXXXX"
4 24 GET INCODE PICTURE "XX"
5 25 GET INPFTDT1 PICTURE "XXXXX"
5 26 GET INRFLD PICTURE "XX"
6 27 GET INRFLDT PICTURE "XXXXX"
6 28 GET INRFLSC PICTURE "9999"
7 29 GET INRFLDT PICTURE "XXXXX"
7 30 GET INPSTSC PICTURE "9999"
8 31 GET INPSTDT PICTURE "XXXXX"
8 32 GET INPSTSC PICTURE "9999"
9 33 GET INNAME PICTURE "XXXXX"
9 34 GET INNAME PICTURE "XXXXX"
10 35 GET INRANK PICTURE "XX"
10 36 GET INRANK PICTURE "XX"
11 37 GET INCODE PICTURE "XX"
11 38 GET INCODE PICTURE "XX"
12 39 GET INNAME PICTURE "XXXXX"
12 40 GET INNAME PICTURE "XXXXX"
13 41 GET INRANK PICTURE "XX"
13 42 GET INRANK PICTURE "XX"
14 43 GET INRANK PICTURE "XX"
14 44 GET INRANK PICTURE "XX"
15 45 GET INRANK PICTURE "XX"
15 46 GET INRANK PICTURE "XX"
16 47 GET INRANK PICTURE "XX"
16 48 GET INRANK PICTURE "XX"
17 49 GET INRANK PICTURE "XX"
17 50 GET INRANK PICTURE "XX"
18 51 GET INRANK PICTURE "XX"
18 52 GET INRANK PICTURE "XX"
19 53 GET INRANK PICTURE "XX"
19 54 GET INRANK PICTURE "XX"
20 55 GET INRANK PICTURE "XX"
20 56 GET INRANK PICTURE "XX"
21 57 GET INRANK PICTURE "XX"
21 58 GET INRANK PICTURE "XX"
22 59 GET INRANK PICTURE "XX"
22 60 GET INRANK PICTURE "XX"

READ
STORE INSSN TO ASSN
STORE 1 TO ERROPT
TO WHILE ERROPT < 17
 2 24.0 SAY "ERROR CHECKING IN PROGRESS"
 2 25.0 DC ERROR
 2 26.0 STOE (ERROPT + 1) TO ERROPT
END DC
IF INCCMNT
 2 61.0 SAY "INFILE ALL LIKE IN??????
 2 62.0 RELEASE ALL LIKE IN??????
 2 63.0 RELEASE ALL EXCEPT AA??????
 2 64.0 DC CRITCMNT
 2 65.0 DC WRITCMNT
 2 66.0 RESTORE FROM A:INFILF ADDITIVE
 2 67.0 DO ITSCRN
END IF

1 15.0 SAY INNAME
1 16.0 SAY INRANK
```
1.61 SAY INPRIMEM
1.66 SAY INSECMOS
2.22 SAY INBIRTH
2.49 SAY INSSN
3.11 SAY INCOMP
3.23 SAY INPLAT
3.39 SAY INJNDATE
3.51 SAY INEAS
3.69 SAY INGAS
4.26 SAY INCODE
4.51 SAY INPPT1 USING "999"
4.66 SAY INPPTDT1
5.28 SAY INHIS
5.51 SAY INPPT2 USING "999"
5.66 SAY INPPTDT2
6.28 SAY INBRIIL
6.48 SAY INNBC
7.28 SAY ININIT
7.59 SAY INMKS
8.28 SAY INHAID
8.59 SAY INUNID
9.59 SAY INTAC
11.0 SAY INPLAT
11.6 SAY INJNDATE
13.21 SAY INWEIGHT
13.59 SAY INPSTSC
14.29 SAY INPSTDT
14.59 SAY INNIGHT
15.29 SAY INELEC
15.59 SAY INDRUG
16.29 SAY INCLCWE
16.59 SAY INALCH
17.29 SAY INHUM
18.29 SAY INMOS
18.59 SAY INPERSAF
19.29 SAY INLD
19.59 SAY INUCMJ
21.19 SAY INCHMOR
21.49 SAY INCOMMNT
INDI
24.1 SAY "STORING DATA IN THE SYSTEM ON DISK"
USE A:INDEX A:PIESSN, A:ALPHPERS, A:MOSNDX
APPEND BLANK
REPLACE NAME WITH ! (INNAME)
REPLACE SSN WITH INSS
REPLACE RANK WITH INBIRTH, PRIMEMOS WITH INPRIMEM
REPLACE SECMSOS WITH INSECMSOS, COMPANY WITH !(INCOMP)
REPLACE PLATOON WITH !(INPLAT), JOINDATE WITH INJNDATE
REPLACE EAS WITH INEAS, BIRTHDATE WITH INBIRTH
REPLACE COMMENT WITH INCOMMNT
REPLACE WEIGHT WITH INWEIGHT, WEIGHT WITH INWEIGHT
REPLACE WTCNT WITH INWEIGHT, GASMASK WITH ! (INGAS)
USE B:INDEX B:ESISSN
APPEND BLANK
REPLACE SSN WITH INSSN, COG WITH !(INCODE)
REPLACE HIS WITH ! (INHIS), COD WITH !(INDRILL)
REPLACE INT WITH ! (ININT), AID WITH !(INALID)
REPLACE UNID WITH !(INUNI)
REPLACE E1T1RAW WITH INPPT1, E1T1DATE WITH INPPTDT1
REPLACE E1T2RAW WITH INPPT2, E1T2DATE WITH INPPTDT2
REPLACE NEC WITH !(INNBC)
REPLACE E1KS WITH !(INMKS), TAC WITH !(INTAC)
USE B:INDEX B:QUALISSN
APPEND BLANK
209
REPLACE SSN WITH INSSN, SWINQUAL WITH !(INSWQU)
REPLACE SWINDATE WITH INSWDT, FFLDATE WITH INFRFLDT
REPLACE SSRFScore WITH INRPLSC
REPLACE FISTScore WITH INPSTSC
REPLACE ESTATE WITH INFSTDT

USE E:INFMISC INDEX E:INFOSSN
APPEND BLANK
REPLACE SSN WITH INSSN, ELECTWAR WITH !(INELEC)
REPLACE CLDWTHR WITH !(INCLNEWTS)
REPLACE LAWWAR WITH !(INLAW), MOS WITH !(INMOS)
REPLACE DRUG WITH !(INDRUG), LERSHP WITH INLDR
REPLACE ALCOHOL WITH !(INILCH), HUMREL WITH !(INHUM)
REPLACE ERSAPFR WITH !(INERSATF), UCMI WITH !(INUCM)
REPLACE CHRMRED WITH !(INCHRMCR)
STORE 1 TO CALCOPT
DO WHILE CALCOPT < 6
   DO CALCUL
      STCRE (CALCOPT + 1) TO CALCPT
   ENDDC
   RELEASE ALL EXCEPT AA???????
REPLACE AASSN TO VSSN
IO ITESCN
DO GETDATA

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

* Routine Name: Error.prg
* Module Name: Maintenance Module
* Version: 6.3.1.1
* Author: D.P. Haessler
* 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: Creatitz.prg, update.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
  a prototype system but can be increased if the need
  arises.

STORE ADATE TO INDATE
STORE T TO INERROR
DO WHILE INERROR TO INERROR
  STORE F TO INERROR
  DC CASE
  CASE ERRORPT = 1
    IF NOT ( $ (INRANK 1 1) = 'E' OR $ (INRANK 1 1) = 'O' )
      STORE T TO INERROR
      @ 24,0 SAY "1.51 GET INRANK PICTURE "X9"
    ENDIF
  CASE ERRORPT = 2
    IF NOT ( VAL($ (INRTH 1 2)) < 31 AND
             VAL($ (INRTH 3 2)) < 13 AND
             VAL($ (INRTH 5 2)) <= VAL($ (INDATE 5 2)) )
      STORE T TO INERROR
      @ 24,0 SAY "1.51 GET INRANK PICTURE "X9"
    ENDIF
  CASE ERRORPT = 3
    IF NOT ( VAL($ (INJNATE 1 2)) < 31 .AND.
             VAL($ (INJNATE 2 2)) < 13 .AND.
             VAL($ (INJNDATE 2 2)) <= VAL($ (INDATE 2 2)) )
      STORE T TO INERROR
      @ 24,0 SAY "1.51 GET INJNDATE PICTURE "X9"
    ENDIF
  CASE ERRORPT = 4
    IF NOT (INGAS) = 'S' OR (INGAS) = 'M' OR ;
      STORE T TO INERROR
      @ 24,0 SAY "1.51 GET INGAS PICTURE "X"
    ENDIF
  CASE ERRORPT = 6
    IF NOT (INPT1 < 30)
      STORE T TO INERROR
      @ 24,0 SAY "1.51 SAY "IMPROPER PFT SCORE -- REENTER"
    ENDIF

211
@ 450 GET INPFT1 PICTURE "999"

READ

ENDIF

CASE ERROPT = 6
  IF .NOT. $(INPFTT1, 1, 2)) < 31 .AND.: VAL($INPFTDT1, 3, 2) < 13 .AND.: VAL($INPFTDT1, 5, 2) = VAL($INDATE, 5, 2) .CR.: INPFTDT1 = 'DDMMYY'
  STORE TC INERROR
  @ 24.0 SAY "IMPROPER PPT DATE -- REENTER"
  @ 465 GET INPFTDT1 PICTURE "999999"

READ

ENDIF

CASE ERROPT = 7
  IF .NOT. (INPFTT2 < 301) STORE TC INERROR
  @ 24.0 SAY "IMPROPER PPT SCORE -- REENTER"
  @ 565 GET INPFT2 PICTURE "999"

READ

ENDIF

CASE ERROPT = 8
  IF .NOT. $(INPFTT2, 1, 2)) < 31 .AND.: VAL($INPFTDT2, 3, 2) < 13 .AND.: VAL($INPFTDT2, 5, 2) = VAL($INDATE, 5, 2) .CR.: INPFTDT2 = 'DDMMYY'
  STORE TC INERROR
  @ 24.0 SAY "IMPROPER PPT DATE -- REENTER"
  @ 565 GET INPFTDT2 PICTURE "999999"

READ

ENDIF

CASE ERROPT = 9
  IF .NOT. ($INSWQU, 1, 1) = 'S'. OR. $(INSWQU, 1, 1) = 'S'. OR. INSWQU = 'WQ'. OR. INSWQU = 'WQ'
  STORE TC INERROR
  @ 24.0 SAY "IMPROPER ENTRY -- REENTER"
  @ 11.19 GIT INSWQU PICTURE "XX"

READ

ENDIF

CASE ERROPT = 10
  IF .NOT. $(INSWD, 1, 2)) < 31 .AND.: VAL($INSWD, 3, 2) < 13 .AND.: VAL($INSWD, 5, 2) = VAL($INDATE, 5, 2) .CR.: INSWDT = 'DDMMYY'
  STORE TC INERROR
  @ 24.0 SAY "IMPROPER DATE -- REENTER"
  @ 21.2 GIT INSWD PICTURE "999999"

READ

ENDIF

CASE ERROPT = 11
  IF .NOT. ($INFLSC < '251') STORE TC INERROR
  @ 24.0 SAY "IMPROPER RIFLE SCORE --- REENTER"
  @ 12.30 GIT INFLSC PICTURE "999"

READ

ENDIF

CASE ERROPT = 12
  IF .NOT. $(INRFLDT, 1, 2)) < 31 .AND.: VAL($INRFLDT, 3, 2) < 13 .AND.: VAL($INRFLDT, 5, 2) = VAL($INDATE, 5, 2) .CR.: INRFLDT = 'DDMMYY'
  STORE TC INERROR
  @ 24.0 SAY "IMPROPER DATE -- REENTER"
@ 12.35 GET INRFLDI PICTURE "999999"
READ
ENDIF
CASE ERROR = 13
IF INHEIGHT > '300' .CR. INHEIGHT < '070'
STORE T TO INERROR
@ 24,0 SAY "" @ 24,1 SAY "IMPROPER WEIGHT -- REENTER"
@ 12.58 GET INHEIGHT PICTURE "999"
READ
ENDIF
CASE ERROR = 14
IF .NOT. (INESTSC < '301')
STORE T TO INERROR
@ 24,0 SAY "" @ 24,1 SAY "IMPROPER PISTOL SCORE -- REENTER"
@ 13.20 GET INPSTSC PICTURE "999"
READ
ENDIF
CASE ERROR = 15
IF .NOT. (VAL($(INPSTDT,1,2)) < 31 .AND.
VAL($(INPSTDT,3,2)) < '13' .AND.
VAL($(INPSTDT,5,2)) <= VAL($(INDATE,5,2)) .CR.
INPSTDT = 'DDMMYY')
STORE T TO INERROR
@ 24,0 SAY "" @ 24,1 SAY "IMPROPER DATE -- REENTER"
@ 13.35 GET INPSTLT PICTURE "999999"
READ
ENDIF
CASE ERROR = 16
IF .NOT. (INHGHT < '78' .AND. INHGHT > '64')
STORE T TO INERROR
@ 24,0 SAY "" @ 24,1 SAY "IMPROPER HEIGHT -- REENTER"
@ 13.58 GET INHGHT PICTURE "99"
READ
ENDIF
ENDCASE
ENDDC
RETURN
K. CALCULATION ROUTINE

* Routine Name: Calculate.prg
* Module Name: Maintenance Module
* Version: 6.3.1.2
* Author: D. E. Haeseler
* Date: 2 Dec 83
* Variables Used: calcept, aassn, aadate, birthdate, age, mm, in*
* Variables Modified: age, mm, in*
* Variables Released: all except age********
* 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.

CASE
CASE CALCOPT = 1
USE A:PERSON INDEX A:PERSONS
FIND &AASSN
IF IPFT1RAW > 0
STORE (VAL(2(AADATE,5,2)) - 1)
STORE (VAL(3(BIRTHDATE,5,2))) TO AGE
STORE (VAL(3(BIRTHDATE,3,2))) TO MM
IF MM < 0
STORE AGE - 1 TC AGE
ENDIF
USE B:EST INDEX B:ESTSSN
FIND &AASSN
IF IPFT1RAW >= 285
STORE 'S' TO INPFT1CL
REPLACE PFT1CLASS WITH INPFT1CL
@ 4,57 SAY INPFT1CL
RELEASE INPFT1CL
USE
ELSE
IF AGE <= 26
STORE IPFT1RAW TO IPFT1RAW
ENDIF
IF AGE < 40 AND AGE > 26
STORE (IPFT1RAW + 25) TO IPFT1RAW
ENDIF
IF AGE > 39
STORE (IPFT1RAW + 50) TO IPFT1RAW
ENDIF
IF IPFT1RAW <= 134
STORE 'U' TO INPFT1CL
REPLACE PFT1CLASS WITH INPFT1CL
@ 4,57 SAY INPFT1CL
RELEASE INPFT1CL
USE
ENDIF
IF IPFT1RAW <= 174 AND IPFT1RAW > 134
STORE '3' TO INPFT1CL
REPLACE PFT1CLASS WITH INPFT1CL
@ 4,57 SAY INPFT1CL
RELEASE INPFT1CL
USE
ENDIF
IF IPFT1RAW <= 224 AND IPFT1RAW > 174
STORE '7' TO INPFT1CL
REPLACE PFT1CLASS WITH INPFT1CL
@ 4,57 SAY INPFT1CL
214
RELEASE INPFT1CL
USE
ENDIF
IF IPFT1FAW > 224
STORE '1' TO INPFT1CL
REPLACE PFT1CLASS WITH INPFT1CL
@ 4,57 SAY INPFT1CL
RELEASE INPFT1CL
USE
ENDIF

ENDIF
CASE CALCLOPT = 2
USE A:PERS INDEX A:PERSSN
FIND &AASSN
IF IPFT2 > 0
STORE (VAL($AADATE,5,2)) -;
VAL($BIRTHDATE,5,2)) TO AGE
STORE (VAL($AADATE,5,2)) -;
VAL($BIRTHDATE,3,2)) TO MM
IF MM < 0
STORE AGE = 1 TC AGE
ENDIF
USE B:EST INDEX B:ESTSSN
FIND &AASSN
IF PFT2RAW >= 285
STORE 'S' TO INPFT2CL
REPLACE PFT2CLASS WITH INPFT2CL
@ 5,57 SAY INPFT2CL
RELEASE INPFT2CL
USE
ELSE
IF AGE <= 26:
STORE PFT2RAW TC IPFT2RAW
ENDIF
IF AGE < 40 .AND. AGE > 26
STORE (PFT2RAW + 25) TO IPFT2RAW
ENDIF
IF AGE > 39
STORE (PFT2RAW + 50) TO IPFT2RAW
ENDIF
IF IPFT2FAW <= 134
STORE 'U' TO INPFT2CL
REPLACE PFT2CLASS WITH INPFT2CL
@ 5,57 SAY INPFT2CL
RELEASE INPFT2CL
USE
ENDIF
IF IPFT2FAW <= 174 .AND. IPFT2RAW > 134
STORE '3' TO INPFT2CL
REPLACE PFT2CLASS WITH INPFT2CL
@ 5,57 SAY INPFT2CL
RELEASE INPFT2CL
USE
ENDIF
IF IPFT2FAW <= 224 .AND. IPFT2RAW > 174
STORE '2' TO INPFT2CL
REPLACE PFT2CLASS WITH INPFT2CL
@ 5,57 SAY INPFT2CL
RELEASE INPFT2CL
USE
ENDIF
IF IPFT2RAW > 224
STORE '1' TO INPFT2CL
REPLACE PFT2CLASS WITH INPFT2CL
@ 5,57 SAY INPFT2CL
RELEASE INPFT2CL
USE
ENDIF
ENDIF
ENDIF
CASE CALCOPT = 3
USE E:QUAL INDEX B:QUALSSN
FIND E:ASSN
IF RELSCORE > '0'
IF RELSCORE < '190'
STORE 'UN' TO INRFLQU
REPLACE RELQUAL WITH INRFLQU
@ 12.27 SAY INRFLQU
RELEASE INRFLQU
ENDIF
IF RELSCORE < '210' .AND. RELSCORE >= '190'
STORE 'NM' TO INRFLQU
REPLACE RELQUAL WITH INRFLQU
@ 12.27 SAY INRFLQU
RELEASE INRFLQU
ENDIF
IF RELSCORE < '220' .AND. RELSCORE >= '210'
STORE 'SS' TO INRFLQU
REPLACE RELQUAL WITH INRFLQU
@ 12.27 SAY INRFLQU
RELEASE INRFLQU
ENDIF
ENDIF
USE
ENDIF
CASE CALCOPT = 4
USE B:QUAL INDEX B:QUALSSN
FIND &ASSN
IF PSTSCORE > '0'
IF PSTSCORE < '180'
STORE 'UN' TO INPSTQU
REPLACE PSTQUAL WITH INPSTQU
@ 13.27 SAY INPSTQU
RELEASE INPSTQU
ENDIF
IF PSTSCORE < '210' .AND. PSTSCORE >= '180'
STORE 'NM' TO INPSTQU
REPLACE PSTQUAL WITH INPSTQU
@ 13.27 SAY INPSTQU
RELEASE INPSTQU
ENDIF
IF PSTSCORE < '250' .AND. PSTSCORE >= '210'
STORE 'SS' TO INPSTQU
REPLACE PSTQUAL WITH INPSTQU
@ 13.27 SAY INPSTQU
RELEASE INPSTQU
ENDIF
ENDIF
USE
ENDIF
CASE CALCOPT = 5
USE A:PERS INDEX A:PERSSN
FIND &ASSN
STORE T TO INWTCONT
IF HEIGHT = '64' .AND. WEIGHT <= '160' .AND.: WEIGHT >= '165'
STORE P TO INWTCONT
216
REPLACE WTCCNT WITH INWTCONT
ENDEF
IF HEIGHT = '66'. AND. WEIGHT <= '165'. AND.;
    WEIGHT >= '106'
    STORE P TO INWTCONT
    REPLACE WTCCNT WITH INWTCONT
@ 11,59 SAY INWTCONT
ENLIF
IF HEIGHT = '66'. AND. WEIGHT <= '170'. AND.;
    WEIGHT >= '107'
    STORE P TO INWTCONT
    REPLACE WTCCNT WITH INWTCONT
@ 11,59 SAY INWTCONT
ENDEF
IF HEIGHT = '66'. AND. WEIGHT <= '175'. AND.;
    WEIGHT >= '111'
    STORE P TO INWTCONT
    REPLACE WTCCNT WITH INWTCONT
@ 11,59 SAY INWTCONT
ENLIF
IF HEIGHT = '66'. AND. WEIGHT <= '181'. AND.;
    WEIGHT >= '115'
    STORE P TO INWTCONT
    REPLACE WTCCNT WITH INWTCONT
@ 11,59 SAY INWTCONT
ENDEF
IF HEIGHT = '66'. AND. WEIGHT <= '186'. AND.;
    WEIGHT >= '119'
    STORE P TO INWTCONT
    REPLACE WTCCNT WITH INWTCONT
@ 11,59 SAY INWTCONT
ENLIF
IF HEIGHT = '70'. AND. WEIGHT <= '192'. AND.;
    WEIGHT >= '122'
    STORE P TO INWTCONT
    REPLACE WTCCNT WITH INWTCONT
@ 11,59 SAY INWTCONT
ENDEF
IF HEIGHT = '70'. AND. WEIGHT <= '197'. AND.;
    WEIGHT >= '127'
    STORE P TO INWTCONT
    REPLACE WTCCNT WITH INWTCONT
@ 11,59 SAY INWTCONT
ENLIF
IF HEIGHT = '72'. AND. WEIGHT <= '203'. AND.;
    WEIGHT >= '131'
    STORE P TO INWTCONT
    REPLACE WTCCNT WITH INWTCONT
@ 11,59 SAY INWTCONT
ENDEF
IF HEIGHT = '72'. AND. WEIGHT <= '209'. AND.;
    WEIGHT >= '135'
    STORE P TO INWTCONT
    REPLACE WTCCNT WITH INWTCONT
@ 11,59 SAY INWTCONT
ENLIF
IF HEIGHT = '74'. AND. WEIGHT <= '214'. AND.;
    WEIGHT >= '139'
    STORE P TO INWTCONT
    REPLACE WTCCNT WITH INWTCONT
@ 11,59 SAY INWTCONT
ENDEF
IF HEIGHT = '76'. AND. WEIGHT <= '219'. AND.;
    WEIGHT >= '143'
    STORE P TO INWTCONT
    REPLACE WTCCNT WITH INWTCONT
@ 11,59 SAY INWTCONT
ENLIF
IF \text{HEIGHT} = '76' \text{AND. WEIGHT} \leq '225' \text{AND.}: \\
\text{WEIGHT} \geq '147' \\
\text{STORE F TO INWTCONT} \\
\text{REPLACE WTCONT WITH INWTCONT} \\
@ 11,59 \text{SAY INWTCONT} \\
ENDIF \\
IF \text{HEIGHT} = '77' \text{AND. WEIGHT} \leq '230' \text{AND.}: \\
\text{WEIGHT} \geq '151' \\
\text{STORE F TO INWTCONT} \\
\text{REPLACE WTCONT WITH INWTCONT} \\
@ 11,59 \text{SAY INWTCONT} \\
ENDIF \\
IF \text{HEIGHT} = '76' \text{AND. WEIGHT} \leq '235' \text{AND.}: \\
\text{WEIGHT} \geq '153' \\
\text{STORE F TO INWTCONT} \\
\text{REPLACE WTCONT WITH INWTCONT} \\
@ 11,59 \text{SAY INWTCONT} \\
ENDIF \\
IF \text{INWTCONT} \\
\text{REPLACE WTCONT WITH INWTCONT} \\
@ 11,59 \text{SAY INWTCONT} \\
ENDIF \\
USE \\
ENCASE \\
RETURN
L. COMMENT UPDATE ROUTINE

* Routine Name: WRTCUNIT.prg
* Module Name: Maintenance Module
* Version: 6.3.1.3
* Author: D.P. Haeusler
* Date: 19 Jan 84
* Variables Used: assn, intxt*
* Variables Modified: intxt*
* Variables Created: intxt*
* Variables Released: all like intxt*
* Files (opened/closed): a:cmnt (o/c)
* Temporary Files Created: none
* Using Subroutines: indata.prg, updata.prg
* Description: This routine updates the information stored in the database on an individual.

ERASE
** 3.3 SAY "COMMENT SECTION"
USE A:CMNT INDEX A:CMNTSS
FIND &ASSN
TP, NCT, # = 0
@ 8,7 SAY INFOTXT1
@ 8,1 SAY INFOTXT2
@ 9,1 SAY INFOTXT3
@ 9,2 SAY INFOTXT4
@ 10,2 SAY INFOTXT5
@ 10,1 SAY INFOTXT6
@ 11,2 SAY INFOTXT7
@ 11,1 SAY INFOTXT8
@ 12,1 SAY INFOTXT9
STORE INFOTXT1 TO INTXT1
STORE INFOTXT2 TO INTXT2
STORE INFOTXT3 TO INTXT3
STORE INFOTXT4 TO INTXT4
STORE INFOTXT5 TO INTXT5
STORE INFOTXT6 TO INTXT6
STORE INFOTXT7 TO INTXT7
STORE INFOTXT8 TO INTXT8
STORE INFOTXT9 TO INTXT9
STORE INTXT10 TO INTXT10
@ 8,1 GET INFOTXT1 PICTURE "XXXXXXXXXXXXXXXXXXXXXXXXX"
@ 8,50 GET INFOTXT2 PICTURE "XXXXXXXXXXXXXXXXXXXXXXXXX"
@ 9,1 GET INFOTXT3 PICTURE "XXXXXXXXXXXXXXXXXXXXXXXXX"
@ 9,50 GET INFOTXT4 PICTURE "XXXXXXXXXXXXXXXXXXXXXXXXX"
@ 10,1 GET INFOTXT5 PICTURE "XXXXXXXXXXXXXXXXXXXXXXXXX"
@ 10,50 GET INFOTXT6 PICTURE "XXXXXXXXXXXXXXXXXXXXXXXXX"
@ 11,1 GET INFOTXT7 PICTURE "XXXXXXXXXXXXXXXXXXXXXXXXX"
@ 11,50 GET INFOTXT8 PICTURE "XXXXXXXXXXXXXXXXXXXXXXXXX"
@ 12,1 GET INFOTXT9 PICTURE "XXXXXXXXXXXXXXXXXXXXXXXXX"
@ 1,0 GET INFOTXT10 PICTURE "XXXXXXXXXXXXXXXXXXXXXXXXX"
REAL
REPLACE INFOTXT1 WITH INTXT1
REPLACE INFOTXT2 WITH INTXT2
REPLACE INFOTXT3 WITH INTXT3
REPLACE INFOTXT4 WITH INTXT4
REPLACE INFOTXT5 WITH INTXT5
REPLACE INFOTXT6 WITH INTXT6
REPLACE INFOTXT7 WITH INTXT7
REPLACE INFOTXT8 WITH INTXT8
REPLACE INFOTXT9 WITH INTXT9
REPLACE INFOTXT10 WITH INTXT10
ENDIP
USE
RELEASE ALL LIKE INTXT?
RETURN
M. COMMENT CREATION ROUTINE

* Routine Name: CrtCmnt.prq  
* Module Name: Maintenance Module  
* Version: 6.3.1.4  
* Author: D.P. Hausler  
* Date: 10 Jan 84  
* Variables Used: newtext, aassn  
* Variables Modified: newtext  
* Variables Created: newtext  
* Variables Released: newtext  
* Files (opened/closed): a:cmnt (o/c)  
* Temporary Files Created: none  
* Using Subroutines: indata.prq, updata.prq  
* 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, INFOTXT1 WITH NEWTXT  
REPLACE INFOTXT2 WITH NEWTXT, INFOTXT3 WITH NEWTXT  
REPLACE INFOTXT5 WITH NEWTXT, INFOTXT6 WITH NEWTXT  
REPLACE INFOTXT7 WITH NEWTXT, INFOTXT8 WITH NEWTXT  
REPLACE INFOTXT9 WITH NEWTXT, INFOTXT10 WITH NEWTXT  
REPLACE INFOTXT11 WITH NEWTXT  
LSP  
RELEASE NEWTXT  
RETURN

220
N. DELETE ITR ROUTINE

* Routine Name: Deletrn. prg
* 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 (c/c), b: informisc (c/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: ALPH PerS, a: PERSSN, a: MOSN
FIND DELNAME
IF .NCT. # = 0
SICSE SSN TO AASSN
LC ITSCAN
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.:0 SAY "DELETING ITR"
STORE T TO DELTHRU
USE a: ALPH PerS, a: PERSSN, a: MOSN
FIND &AASSN
IF .NCT. # = 0
DELETE PACK
ENDIF
USE a: CMNNT INDEX a: CMNNTSSN
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: INFORMISC INDEX a: INFSN

221
FIND SAASSN
IF .NCT. # = 0
DELETE PACK
ENDIF
ENDIF
ENDDO
END IF
SICRE P TO DEDONE
DC WHILE .NOT. DEDONE
ERASE @ 12, 20 SAY " DO you WANT TO DELETE ANOTHER RECORD? "
ACCEPT " ENTER YES OR NO =========> " TO DELCHK
IF I(DELCHK) = 'NO'
STORE T TO DEDONE
STORE T TO TLELPINT
ELSE
IF I(DELCHK) = 'YES'
STORE T TO DEDONE
ENDIF
ENDIF
ENDO
ENDDC
PLEASE ALL EXCEPT AA??????
RETURN
C. SYSTEM FUNCTION CONTROL ROUTINE

* Routine Name: System.prg
* Module Name: Maintenance Module
* Version: 6.5.0
* Author: D.P. Haeusler
* Date: 30 Jan 1984
* Variables Used: aacomple, sysopt
* Variables Modified: aacomple, sysopt
* Variables Created: aacomple, sysopt
* Variables Released: 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 "
  6,24 SAY " SELECT ONE OF THE FOLLOWING OPTIONS"
  10,31 SAY " I...INSTALL SYSTEM"
  11,31 SAY " S...SYSTEM RESET (YEARLY)"
  12,31 SAY " A...ACCESS LIST MAINTENANCE"
  13,31 SAY " C...QUIT TO MAINTENANCE MENU"
  ACCEPT " ENTER OPTION ==>" TO SYSOPT
  DC CASE
  CASE 1(SYSOPT) = " I"
      STORE P TO AACOMPLE
      DO B: INSTALL
  CASE 1(SYSOPT) = " S"
      STORE P TO AACOMPLE
      DO B: SYSRESFY
  CASE 1(SYSOPT) = " A"
      STORE P TO AACOMPLE
      DC B: RSTACCESS
  CASE 1(SYSOPT) = " Q"
      STORE T TO AACOMPLE
  ENDCASE
ENDDC
RELEASE ALL EXCEPT AA??????
RELEASE AACOMPLE
RETURN

222
E. SYSTEM INSTALLATION ROUTINE

Routine Name: Install.prg
Module Name: Maintenance Module
Version: 6.5.1
Author: D.P. Haeusler
Date: 7 Feb 84
Variables Used: *cc, *plt, cntyr
Variables Modified: *cc, *plt, cntyr
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
ERASE FROM A:UNITMEM
2, 3C SAY "SYSTEM INSTALLATION"
3, 15 SAY "ENTER THE NAMES OF YOUR COMPANY SIZE UNITS"
3, 34 GET CNECO PICTURE "XXXXXX"
3, 34 GET TWOCO PICTURE "XXXXXX"
3, 34 GET THREECO PICTURE "XXXXXX"
3, 34 GET FOURCO PICTURE "XXXXXX"
READ
STORE I(CNECO) TO ONECO
STORE I(TWOCO) TO TWCCO
STORE I(THREECO) TO THREECO
STORE I(FOURCO) TO FOURCO
ERASE
2, 3C SAY "SYSTEM INSTALLATION"
3, 15 SAY "ENTER THE NAMES OF PLATOON SIZED UNITS"
3, 26 GET CNEPLT PICTURE "XXXXXXXXXX"
3, 26 GET TWOPLT PICTURE "XXXXXXXXXX"
3, 26 GET THREEPLT PICTURE "XXXXXXXXXX"
3, 26 GET FOURPLT PICTURE "XXXXXXXXXX"
READ
STORE I(CNEPLT) TO ONEPLT
STORE I(TWOPLT) TO TWPLT
STORE I(THREEPLT) TO THREEPLT
STORE I(FOURPLT) TO FOURPLT
SAVE TO UNITMEM
ERASE
RELEASE ALL
RESTORE FROM A:MEMDISK
2, 12 SAY "ENTER THE CURRENT YEAR"
ACCEPT "YEAR (YY) ===>" TO CNTYR
SAVE TO A:MEMDISK
RELEASE ALL
RESTORE FROM TEMP
RETURN
```
Q. SYSTEM RESET ROUTINE

* Routine Name: Sysreset.pro
* Module Name: Maintenance Module
* Version: 6.5.2
* Author: E.P. Haeusler
* Date: 4 Feb 84
* Variables Used: crntyr, reset, rgreset, done, q*, *cc
* Variables Modified: crntyr, reset, rgreset, done
* Variables Created: reset, rgreset, done
* Variables Released: all
* Files (opened/closed): a:memdisk.mem, a:unitmem.mwm,
* Temporary Files Created: temp.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 TEMP
RELEASE ALL
RESTORE FROM A:MEMDISK ADDITIVE
ACCEP "ENTER THE CURRENT YEAR (YY) ===>" TO CRNTYR
SAVE TO MEMDISK
RELEASE ALL
STORE F TO DONE
DO WHILE NOT DCNE
  IF 10 TO 15 SAY "DO YOU WANT TO RESET RANGE QUOTAS FOR:"
  ACCEPT "THE YEAR?"
  IF I(RES) = 'Y' THEN
    STORE T TO DONE
    STORE T TO RGRESET
  ELSE
    IF I(RES) = 'N' THEN
      STORE T TO LGNE
      STORE P TO RGRESET
  ENDIF
ENDDO
IF RGRESET THEN
  ERASE
  RESTORE FROM A:ARNGQTAS
  RESTORE FROM A:UNITMEM ADDITIVE
  @ 3,15 SAY "ENTER RANGE QUOTAS AS PROJECTED;"
  FOR THE YEAR"
  @ 4,15 SAY "FOR UNIT:"
  @ 4,25 SAY CHECO
  @ 6,15 SAY "JANUARY"
  @ 6,23 GET QJAN PICTURE "999"
  @ 7,16 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 QAEB PICTURE "999"
  @ 10,15 SAY "MAY"
  @ 10,19 GET QMAY 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"
READ
RELEASE ALL EXCEPT Q???
SAVE TO A:ARNGQTAS
RELEASE ALL
RESTORE FROM A:BRNGQTAS
ERASE
3.15 SAY "ENTER RANGE QUOTAS AS PROJECTED, ;
FOR THE YEAR, ;
4.15 SAY "FOR UNIT: ",
4.25 SAY QCCO
5.15 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 QMAY 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:ARNGQTAS
RELEASE ALL
ERASE
3.15 SAY "ENTER RANGE QUOTAS AS PROJECTED, ;
FOR THE YEAR, ;
4.15 SAY "FOR UNIT: ",
4.25 SAY QCCO
5.15 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 QMAY 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:CRNGQTAS
RELEASE ALL
RELEASE
RESTORE FROM A:DREGQTAS
RESTORE FROM UNITM IN ADDITIVE
3 15 SAY "ENTER RANGE QUCTAS AS PROJECTED, :"
 FOR THE YEAR:
4 15 SAY "FOR UNIT: "
4 25 SAY FOURCO
6 15 SAY "JANUARY"
7 15 SAY "FEBRUARY"
7 24 SAY QFEB PICTURE "999"
8 21 SAY QMAR PICTURE "999"
9 15 SAY "APRIL"
10 15 SAY "MAY"
10 19 SAY QMAT PICTURE "999"
11 15 SAY "JUNE"
11 20 SAY QJUN PICTURE "999"
12 15 SAY "JULY"
12 20 SAY QJUL PICTURE "999"
13 15 SAY "AUGUST"
13 22 SAY QAUG PICTURE "999"
14 15 SAY "SEPTEMBER"
14 25 SAY QSEP PICTURE "999"
15 15 SAY "OCTOBER"
15 22 SAY QCCT PICTURE "999"
16 15 SAY "NOVEMBER"
16 24 SAY 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 T TO DONE
DO WHILE .NOT. DONE
ERASE
3 15 SAY "DO YOU WANT TO RESET YEARLY TRAINING; VALUES:
3 24 SAY "IN THE DATA BASE FOR THE BATTALION?"
ACCEPT " "
IF "RESET" = "Y"
STORE T TO DONE
USE EST
REPLACE ALL COG WITH "***"
REPLACE ALL HIS WITH "***"
REPLACE ALL COD WITH "***"
REPLACE ALL INT WITH "***********"
REPLACE ALL AIL WITH "**********"
REPLACE ALL UNI WITH "***"
REPLACE ALL PETRAN WITH "***"
REPLACE ALL PPTDATE WITH "DDMMYY"
227
REPLACE ALL PPT1CLS WITH "*"
REPLACE ALL PPT2RAW WITH 000
REPLACE ALL PPT2DATE WITH "DDMMYY"
REPLACE ALL PPT2CLS WITH "*"
REPLACE ALL NBC WITH "*******"
REPLACE ALL MKS WITH "*******"
REPLACE ALL TAC WITH "*******"
USE EQUAL
REPLACE ALL RPISCORE WITH "***"
REPLACE ALL RFICIAL WITH "***"
REPLACE ALL RFSSTATE WITH "DDMMYY"
REPLACE ALL PSISCORE WITH "***"
REPLACE ALL PSICUAL WITH "***"
REPLACE ALL PSTSTATE WITH "DDMMYY"
USE
ELSE
IF I(RESET) = 'K'
 STOR T TO LOOUE
 ENDE
 ENDE
 ENDC
 RELEASE ALL EXCEPT AA???????
RETURN
E. SYSTEM ACCESS CONTROL RUTINE

* 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
* Variables Released: all
* 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"
   @ 10, 28 SAY " L...LIST ALL USERS"
   @ 11, 28 SAY " A...ADD TO ACCESS LIST"
   @ 12, 28 SAY " D...DELETE FROM ACCESS LIST"
   @ 13, 28 SAY " Q...QUIT TO SYSTEM FUNCTION MENU"
   ACCEPT " ENTER OPTION ===>' TO ACCEPT
   DC CASE
CASE 1 (ACCOPT) = "L"
   USE A:SECURITY INDEX A:ACCNAME
   DO WHILE .NOT. EOF
   ERASE
   @ 3, 24 SAY " SECURITY ACCESS LIST"
   @ 4, 24 SAY " NAME USER NO."
   PASSWORD LEVEL
   STORE LINES
   DO WHILE .NOT. EOF .AND. LINES < 22
   @ LINES, 1 SAY NAME
   @ LINES, 20 SAY USERID
   @ LINES, 31 SAY USERPASS
   @ LINES, 56 SAY AUTHLEV
   SKIP
   STORE (LINES + 2 ) TO LINES
   CONTINUE ***
   SET CONSOLE CFF
   WAIT
   SET CONSOLE CN
   ENDIF
ENDC
ENDDO
@ 23, 15 SAY "*** PRESS ANY KEY TO CONTINUE ***"
SET CONSOLE CFF
WAIT
SET CONSOLE CN
USE
CASE 1 (ACCOPT) = "A"
STORE T TO ACNAME
DO WHILE MORE
   ERASE
   STORE " TO ACNAME
   STORE " TO ACCUSER
   STORE " TO ACCPASS
   STORE TC ACCLEV
   @ 10, 25 SAY " NAME OF USER"
   @ 10, 30 GET ACNAME PICTURE "XXXXXXXXXXXXXXXX"
   @ 12, 25 SAY "USER ID"
12.33 GET ACCUSER PICTURE "XXXXX"
14.25 SAY "PASSWORD"
14.44 GET ACCEPASS PICTURE "XXXXXXX"
16.25 SAY "AUTHORIZATION LEVEL"
16.45 GET ACCELEV PICTURE "9"
READ
STORE T TO NOGOCD
DO WHILE NOGOCD
  IF (ACCELEV < 1) .OR. (ACCELEV > 3)
    @ 23,1 SAY "ILLEGAL AUTHORIZATION LEVEL"
  ELSE
    STORE F TO NOGOCD
  ENDIF
ENDDO
USE A:SECURITY INDEX A:ACCNAME, A:SECINDX
APPEND BLANK
REPLACE NAME WITH !(ACNAME)
AUTHLEV WITH ACCELEV
REPLACE USERID WITH !(ACUSER)
USERPASS WITH !(ACCEPASS)
USE
STORE P TO TC DONE
DO WHILE NOT. DONE
  ERASE
  @ 10.25 SAY "DO YOU WANT TO ENTER ANOTHER USER?"
  ACCEPT "" ENTER (Y OR N ) ===>"
  IF !(CONT) = "Y"
    STORE T TO TC DONE
  ELSE
    IF !(CONT) = "N"
      STORE T TO DONE
      STORE P TO TC MORE
    ENDIF
  ENDIF
ENDDO
ENDDO
CASE !(ACCEPT) = "D"
STORE T TO TC MORE
DO WHILE NOT. MORE
  ERASE
  @ 12.24 SAY "ENTER NAME OF THE USER TO BE DELETED"
  STORAGE NAME TO ACNAME
  STORE !(ACNAME) TO ACNAME
  IF ACNAME = "QUIT"
    STORE F TO TC MORE
  ELSE
    USE A:SECURITY INDEX A:ACCNAME, A:SECINDX
    FIND &ACNAME
    IF # = 0
      @ 22.28 SAY "INDIVIDUAL NOT FOUND IN FILE"
    ELSE
      USE A:SECURITY INDEX A:ACCNAME, A:SECINDX
      FILE
      ENDIF
    DELETE
    ENDU
  ELSE
    SET CONSOLE OFF
  WAIT
  SET CONSOLE ON
  ENDIF
  USE
ENDDO
ENDU
USE A:SECURITY INDEX A:ACCNAME, A:SECINDX
PACK
USE
ENDDO
CASE 1(ACCEPT) = "Q"
STORE T TO THRU
ENDCASE
ENDDO
RELEASE ALL EXCEPT AA??????
RETURN
BIBLIOGRAPHY


<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>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. Pruiett, 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. Haeusler, USMC</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>12112 F. Craig Dr.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Woodbridge, Virginia 22192</td>
<td></td>
</tr>
</tbody>
</table>