7-1-1984

# A data base for a boy scout troop

Kneungnij Ratanadheb
*Atlanta University*

# A DATA BASE FOR A BOY SCOUT TROOP

A THESIS
SUBMITTED TO THE FACULTY OF ATLANTA UNIVERSITY
FOR FULFILLMENT OF THE REQUIREMENT FOR
THE DEGREE OF MASTER OF SCIENCE

BY

KNEUNGNIJ RATANADHEB
DEPARTMENT OF MATHEMATICAL SCIENCES
ATLANTA, GEORGIA

JULY 1984

R= vii T= 167

# TABLE OF CONTENTS

LIST OF FIGURES .......................................... i

ACKNOWLEDGMENTS.......................................... iii

ABSTRACT ................................................ iv

INTRODUCTION ............................................ v

Chapter

1. WHAT IS A DATA BASE .............................. 1

2. SYSTEM OPERATION ................................. 4

3. THE ENVIRONMENT AND DOCUMENT ASSUMPTION FOR A BOY SCOUT TROOP ................................. 9

4. THE CONCEPTUAL MODEL OF A DATA BASE OF BOY SCOUT TROOP ........................................... 21

APPENDIX A GLOSSARY .................................... 45

APPENDIX B THE SOURCE PROGRAM LISTING .................. 47

LIST OF FOOTNOTES ...................................... 166

BIBLIOGRAPHIES ......................................... 167

## LIST OF FIGURES

1. A Boy Scout Troop System (A.) .................. vi

2. Troop Program Schedule Report (3.1) ............ 10

3. Adult Leadership Roster Report (3.2) ........... 11

4. Troop. Roster Report (3.3) ..................... 13

5. Family Data Report (3.4) ....................... 13

6. Membership Report (3.5) ........................ 14

7. Leadership Report (3.6) ........................ 14

8. Merit Badge Report (3.7) ....................... 16

9. Advancement Report (3.8) ....................... 16

10. Attendance and Dues Report (3.9) .............. 17

11. Camping Log Report (3.10) ..................... 17

12. The Relationship between the data elements in the 'TROOP PROGRAM RECORD' (4.1) ............... 26

13. The relationship between the data elements in the 'ADULT LEADERSHIP ROSTER' (4.2) ............ 26

14. The relationship between the data elements in the 'TROOP ROSTER' (4.3) ........................ 28

15. The relationship between the data elements in the 'FAMILY DATA' (4.4) ......................... 28

16. The relationship between the data elements in the 'MEMBERSHIP DATA' (4.5) .................... 30

17. The relationship between the data elements in

the 'LEADERSHIP RECORD' (4.6) .................. 32

18. The relationship between the data elements in the 'AWARD' report (4.7) ....................... 32

19. The relationship between the data elements in the 'ADVANCEMENT' report (4.8) ................. 34

20. The relationship between the data elements in the 'ATTENDANCE AND DUES' (4.9) ............... 36

21. The relationship between the data elements in The 'OUR CAMPING LOG' (4.10) .................. 38

# ACKNOWLEDGMENTS

I would like to express my gratitude especially to my advisor, Dr. Benjamin J. Martin, who so generously contributed his time, patience and energy to the effectiveness of my research. Furthermore, I would like to extend my respect and my thanks to the following Atlanta University Faculty:

Dr. Bennett Setzer
Dr. Nazir A. Warsi
Dr. Johnny L. Houston
Dr. Ronald Biggers
Dr. G.Scott Owen
Dr. Henry Gore
Dr. Arthur M. Jones
Prof. Grover Simmons

Those instructors have been extremely helpful to me for all the time I have been an Atlanta University student.

# INTRODUCTION

The objective of this data-base management system (DBMS) is to provide a boy scout troop (hereafter referred to as the troop) with a capability of keeping an on-line data bank. This DBMS contains such vital information as troop programs, adult leaders' information, and all information on the scouts.

This system is divided into 3 parts: ADULT, TROOP, and SCOUT as shown in figure A. Each part consists of an application program which has the ability to manage a data bank in such ways as adding, deleting, modifying, listing, and reporting. The 'ADULT' part processes information about adult leadership while 'TROOP' contains records of troop programs and activities, and 'SCOUT' contains information on the scouts themselves.

The output of these 3 parts is provided in 2 forms:

1. Listing of information on a terminal screen.

2. Printing reports in order to fulfill the user's demands.

This thesis consists of 4 chapters. Chapter 1 contains a brief discussion about what a data base is and its function. Chapter 2 contains a review of the dBASE II software package, its limitation and the system requirement

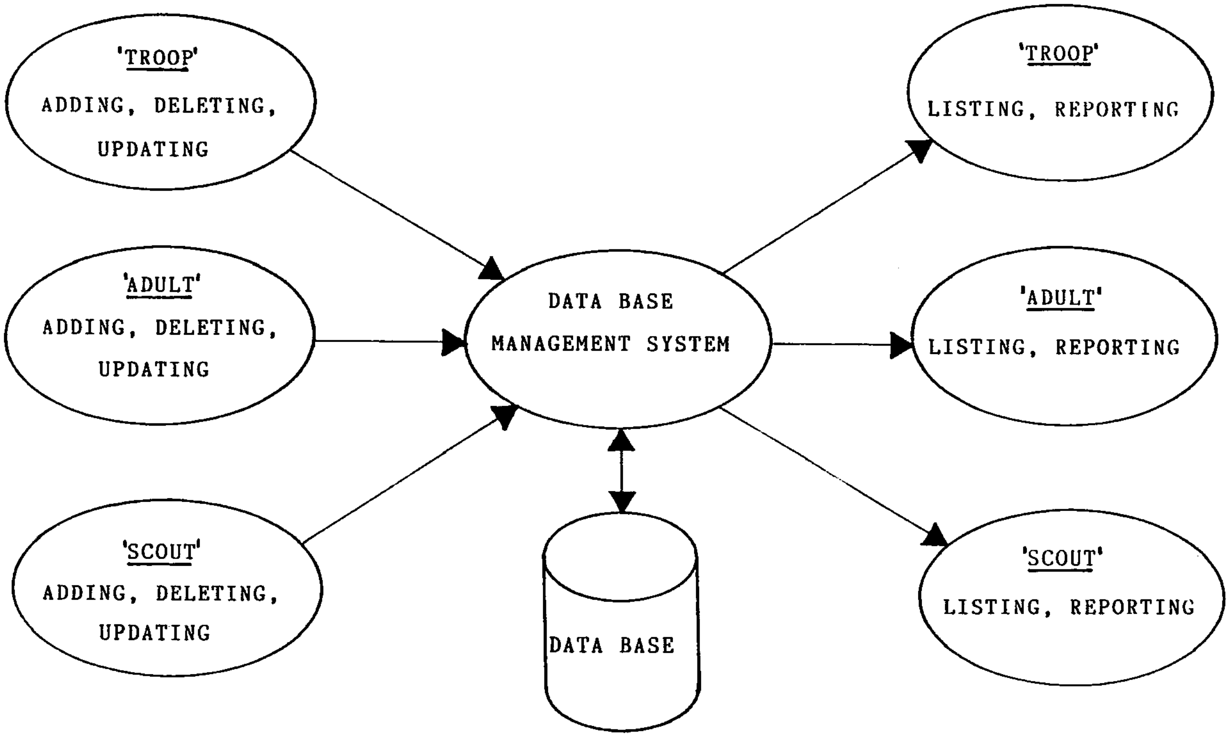


Figure A. A BOY SCOUT TROOP SYSTEM

which is used to operate the dBASE II. Chapter 3 contains an overview of the documents and the environment of the boy scout troop system. Chapter 4 contains the description and the relationship between the data elements referenced in each report. Furthermore, there are Appendices A and B. Appendix A contains a glossary and Appendix B contains the source program listing. Bibliographies are attached at the end.

# CHAPTER 1

# WHAT IS A DATA BASE

What is a data base?

A data base is defined as a collection of related data that are stored together without harmful or unnecessary redundancy, shared and used by multiple applications and users. The data are preceived differently by different users. Each user will see only one view of the data. However, the files that hold the data always have the same structure and appear simple to each user. The data are stored in order to be independent of the programs which use those data. Either data structures or programs can be changed without causing the other to be changed.

A Data Base Management System (DBMS)

A DBMS is a group or collection of routines that connects the users to one or more collections of information.

In the 'perfect' data base management system, the storage and update of the data is invisible to the user. All the data are stored in a single data base. The user is only concerned about information flow. As the system design is altered, very little change is necessary for existing programs.

All the better data base management systems fall into 3 catagories: Hierarchical, Network, or Relational. The hierachical catagory is a special case of the network system.

- In a hierarchical or network system, information is stored in a structure that look very much like an inverted tree. In a hierarchical system, no child can have more than one parent. A network system is similar, except that a child can have more than one parent.
- A relational system is quite different in its organizational structure. The information is stored in 2-dimensional tables that are more conveniently called files. Information in the tables is accessed by the user based on any desired relationships.

A true DBMS also has the following features:

- The data can be used by a large collection of application programs with security control on the information to prevent accesses by unauthorized users.
- Tools for sorting, indexing and reporting data in the data base are provided to minimize programming development.

## The Advantages of a DBMS

- The DBMS helps to prevent the duplication of data by storing everything in a single data base.
- The DBMS reduces program development time. Much of the routine programming work of file management, indexing, sorting and report generating are done automatically with a simple call in a data base management system. Programs can easily evolve to meet the needs of the users.
- The DBMS improves data reliability. The integration of information and the relationship of information within a data base is done automatically by the data base manager. This relieves the programmers of the need to use pointers and pointer chains to locate needed information.

# CHAPTER 2

# SYSTEM OPERATION

## dBASE II

dBASE II is a software package that is marketed by Ashton-Tate, Inc. as a relational data-base management system ( a system that stores information in 2-dimensional tables in which data can be accessed by relationship of information ).

dBASE II is not a true data-base management system in the strictest sense of the word [1]. More accurately, it is like a file management system with an advanced programming language and relational features added. It contains its own programming language, permitting a user with a minimum of experience to develop extremely powerful and complex programs that meet demanding applications. It is also very easy and is 'structured', making it possible for programmers to develop programs much faster than using other high level languages such as BASIC or FORTRAN.

Some of the features of dBASE II include:

- A large degree of program and data independence. Data structures can be altered without the necessity of many program changes.

- Data can be easily added, edited, deleted, sorted, indexed, or reported using a minimum of programmimg.
- Reports can be easily created from data in a data base using mathematical operations like multiplication and division. Subtotals and totals can be easily generated.
- Professional screen formats can be created for data entry. After designing the entire screen, data entry only requires the operator to 'fill in the blanks'.
- The internal language is extremely powerful with a single command for indexing, sorting, and reporting. Programs can be developed at five or ten times the speed of developing equivalent BASIC programs.
- The internal language is a structured language which is far easier to write and update than unstructured programs.

## LIMITATIONS OF dBASE II

dBASE II has some limitations:

- Only 2 files can be used at a time. This creates difficulties with any types of hierarchical processing when there are necessities to get information from more than 2 files.
- dBASE II only permits 32 items of information per record.
- dBASE II uses 48,000 bytes of memory, so there is room for very little of the program in memory. Program lines must be read as they are executed; this means there will be many disk operations during a program execution, which means slower processing. The applications will be slower than some other compiled programs like word processing.
- dBASE II can only respond to a console or a printer. If the dBASE II information must be sent to a remote computer, it must be processed to a separate file and then communications software used to send this file. This limits its use in a real-time world with applications like information networking.
- There have been problems using dBASE II in a batch mode. dBASE II does batch processing by building a submit file; some versions of dBASE II will not permit batch processing on some hardware.

- There is no protection against 2 people writing to the same file at the same time in multi-user systems. The Ashton-Tate,Inc. is working to release a multi-user version of dBASE II in the future.
- The programming language of dBASE II allows only 64 memory variables at a time, and no array variables are available. Due to the fact that dBASE II uses so much computer memory, there is little space left for arrays or memory variables. There are some tricks that allow a user to create an 'array', however, it will take programming development time and a lot of work in order to keep memory variables less than 64 at a time and the program will run slower.

SYSTEM REQUIREMENT

In order to operate dBASE II, at lease one of these 2 system types is needed:

1. A CP/M-80 based system using the 8080,8085 or Z80 microprocessor. These systems are made by several hundred manufacturers, including Compupro, Vector Graphic, Systems Group, and Televideo, and include CP/M-80 version 2.0 or later, one or more floppy disk drives or a hard disk, 48,000 bytes of memory or more, a cursor addressable CRT with a dimensional of at lease 24 lines by 80 columns and a text printer.

2. A CP/M-86 or MS-DOS system like the IBM PC, Rainbow-100 or Victor 9000. This system should include 128K bytes of memory, one or more floopy disk drives or a hard disk, a cursor addressable terminal ( 24 lines by 80 columns ), and a text printer.

dBASE II as purchased is supplied without an operating system. It is important for the user, who wishes to purchase dBASE II, to carefully choose the version of dBASE II that corresponds to the operating system and disk format of the user's hardware

# CHAPTER 3

# THE ENVIRONMENT AND DOCUMENT ASSUMPTION FOR A BOY SCOUT TROOP

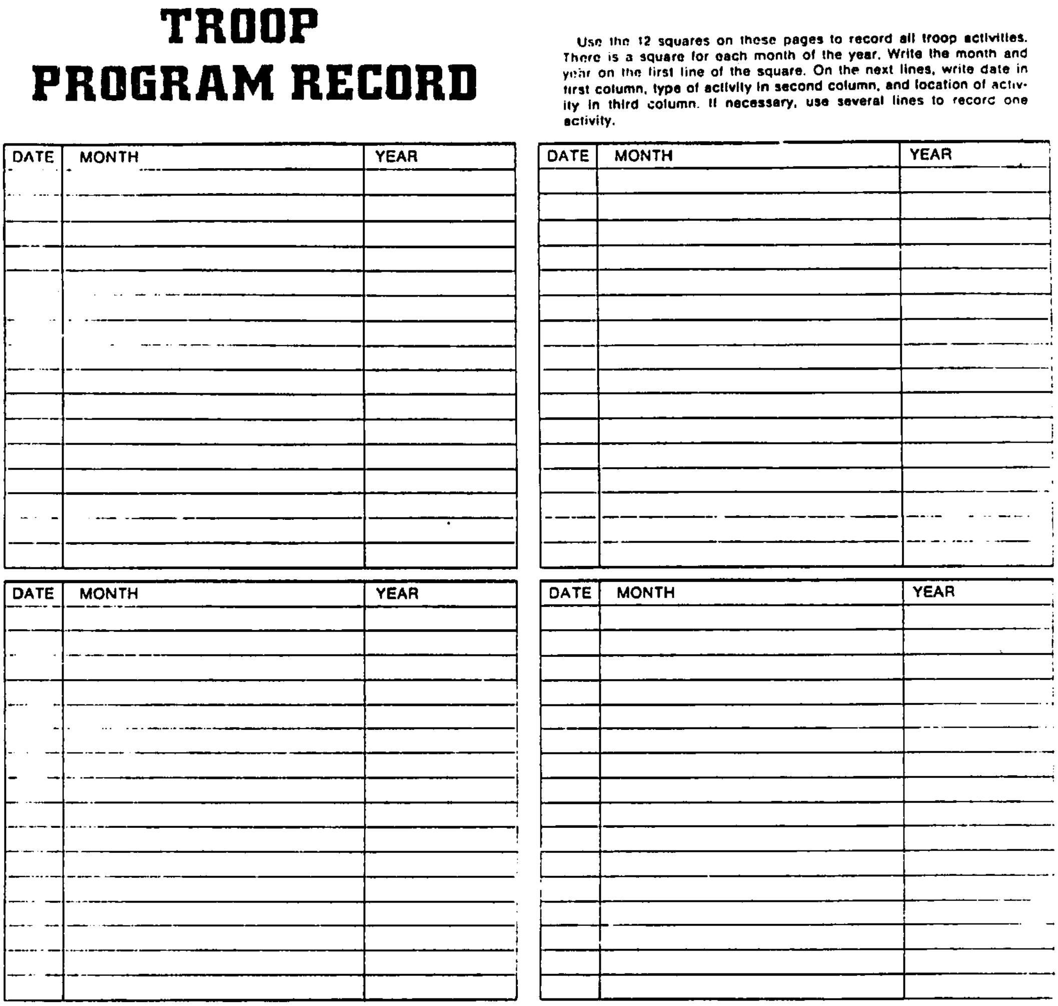
A boy scout troop consists of three important groups of information: the schedule of programs of the troop, the information on adult leaders and the information on the scouts. The data base is designed for the following applications.

## TROOP PROGRAM SCHEDULE (Figure 3.1)

The activities for the boy scout troop may be scheduled in advance. The troop programs tell what types of activities (meeting, camping, etc.), when, and where they will be taking place.

## ADULT LEADERSHIP ROSTER (Figure 3.2)

This is a list of adult leaders who are working for the troop. This list collects the names of these leaders, their titles, their addresses, their sex, their home phone numbers, and their business phone numbers.



# TROOP PROGRAM RECORD

Use the 12 squares on these pages to record all troop activities. There is a square for each month of the year. Write the month and year on the first line of the square. On the next lines, write date in first column, type of activity in second column, and location of activity in third column. If necessary, use several lines to record one activity.

| DATE | MONTH | YEAR |
|---|---|---|
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |

| DATE | MONTH | YEAR |
|---|---|---|
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |

| DATE | MONTH | YEAR |
|---|---|---|
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |

| DATE | MONTH | YEAR |
|---|---|---|
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |

Figure 3.1 TROOP PROGRAM SCHEDULE REPORT

# ADULT LEADERSHIP ROSTER

| TITLE | NAME | ADDRESS | PHONE NO. |
|---|---|---|---|
| Exec. Officer Institution | | | |
| Scouting Coordinator | | | |
| Chairman Committee | | | |
| Troop Committee | | | |
| Troop Committee | | | |
| Troop Committee | | | |
| Troop Committee | | | |
| Troop Committee | | | |
| Troop Committee | | | |
| Troop Committee | | | |
| Scoutmaster | | | |
| Assistant Scoutmaster | | | |
| Assistant Scoutmaster | | | |
| Assistant Scoutmaster | | | |

# OTHER ADULT LEADERSHIP

| | |
|---|---|
| S.M.E.<br>CHAIR PERSON | |
| | |
| | |
| | |
| | |

Figure 3.2 ADULT LEADERSHIP ROSTER REPORT

TROOP ROSTER (Figure 3.3)

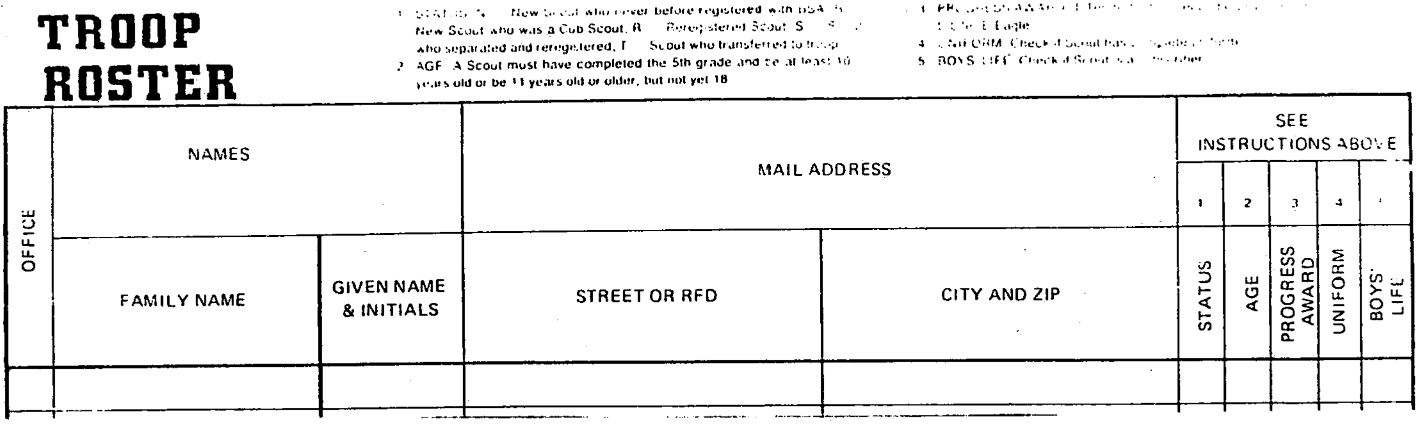
This report is used for collecting the information of the boy scouts. This information includes scouts' name, mailing address, present age, present status, and present rank. The report also states whether each scout has a completed uniform and whether he subscribes to Boy's Life magazine.

FAMILY DATA (Figure 3.4)

Since these scouts are not over 18 yet, they have to have guardian's permission to participate in the activities. This report collects the information on the guardians such as their names, their occupations, their business phone numbers, and their relation to scout. Also, it tells about scouts' church or religion.

MEMBERSHIP (Figure 3.5)

Some types of activities, such as being a cubscout, registered as a scout, or transferred to troop, and when they happened have to be kept as records. This is what the membership records are for. Furthermore, this report keeps the records of scouts' birthdate, their patrol name, whether they have received church award or order of arrow member and when.



# TROOP ROSTER

1 STATUS [illegible] New Scout who never before registered with BSA [illegible] New Scout who was a Cub Scout, R [illegible] Reregistered Scout S [illegible] who separated and reregistered, T [illegible] Scout who transferred to troop

2 AGE A Scout must have completed the 5th grade and be at least 10 years old or be 11 years old or older, but not yet 18

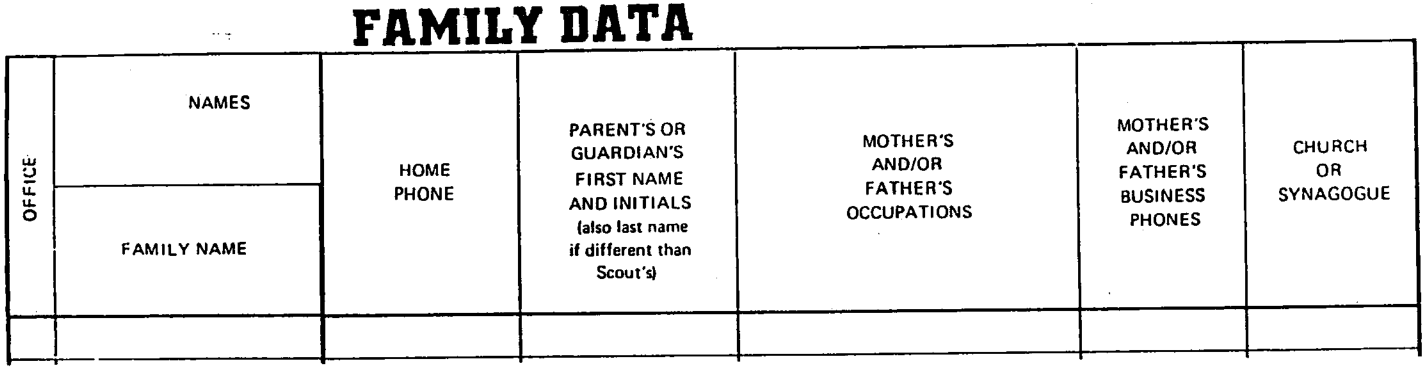
3 PROGRESS AWARD [illegible] Eagle

4 UNIFORM Check if Scout has [illegible]

5 BOYS' LIFE Check if Scout is a [illegible]

| OFFICE | NAMES | | MAIL ADDRESS | | SEE INSTRUCTIONS ABOVE | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | | 1 | 2 | 3 | 4 | 5 |
| | FAMILY NAME | GIVEN NAME & INITIALS | STREET OR RFD | CITY AND ZIP | STATUS | AGE | PROGRESS AWARD | UNIFORM | BOYS' LIFE |
| | | | | | | | | | |

Figure 3.3 TROOP ROSTER REPORT



# FAMILY DATA

| OFFICE | NAMES | HOME PHONE | PARENT'S OR GUARDIAN'S FIRST NAME AND INITIALS (also last name if different than Scout's) | MOTHER'S AND/OR FATHER'S OCCUPATIONS | MOTHER'S AND/OR FATHER'S BUSINESS PHONES | CHURCH OR SYNAGOGUE |
|---|---|---|---|---|---|---|
| | FAMILY NAME | | | | | |
| | | | | | | |

Figure 3.4 FAMILY DATA REPORT

# MEMBERSHIP

| OFFICE | NAMES<br>FAMILY NAME | BIRTH DATE | CUB SCOUT | | BECAME SCOUT | YEARS REGISTERED AS A SCOUT | TRANSFER IN | TRANSFERRED TO ANOTHER TROOP | TRANSFERRED TO EXPLORER UNIT | DROPPED | PATROL NAME | CHURCH AWARD | ORDER OF THE ARROW MEMBER |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | FROM | TO | MONTH & YEAR | | MONTH & YEAR | MONTH & YEAR | MONTH & YEAR | MONTH & YEAR | | | |
| | | | | | | | | | | | | | |

Figure 3.5 MEMBERSHIP REPORT

# LEADERSHIP

| OFFICE | NAMES<br>FAMILY NAME | DEN CHIEF | | LIBRARIAN | | QUARTERMASTER | | SCRIBE | | HISTORIAN | | BUGLER | | CHAPLAIN AIDE | | ASSISTANT PATROL LEADER | | PATROL LEADER | | ASSISTANT SENIOR PATROL LEADER | | SENIOR PATROL LEADER | | LEADERSHIP CORPS | | JUNIOR ASSISTANT SCOUTMASTER | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | FROM | TO | FROM | TO | FROM | TO | FROM | TO | FROM | TO | FROM | TO | FROM | TO | FROM | TO | FROM | TO | FROM | TO | FROM | TO | FROM | TO | FROM | TO |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Figure 3.6 LEADERSHIP REPORT

LEADERSHIP (Figure 3.6)

As a member of the troop, a scout will receive an honor to be a leader of some activities. This report has a purpose of keeping the records of types of the leadership actions and when they happened.

AWARD (Figure 3.7, 3.8)

The purpose of this report is to keep the records of the types of the merit badges and skill awards the scout received and when.

ADVANCEMENT (Figure 3.8)

This report is used to keep the records of the scouts' rank, and the date these records occurred.

ATTENDANCE AND DUES (Figure 3.9)

When the scouts have attended some activities, there are some dues they have to pay. The purpose of this report is to keep the records of scouts' dues, the amount they paid, and their balances.

## MERIT BADGES

| OFFICE | NAMES<br>FAMILY NAME | LAW | LEATHERWORK | LIFESAVING | MACHINERY | MAMMALS | MASONRY | METALS ENGINEERING | METALWORK | MODEL DESIGN AND BUILDING | MOTORBOATING | MUSIC | NATURE | OCEANOGRAPHY | ORIENTEERING | PAINTING | PERSONAL FITNESS | PERSONAL MANAGEMENT | PETS | PHOTOGRAPHY | PIGEON RAISING | PIONEERING | PLANT SCIENCE | PLUMBING | POTTERY | PRINTING | PUBLIC HEALTH | PUBLIC SPEAKING | PULP AND PAPER | RABBIT RAISING | RADIO |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

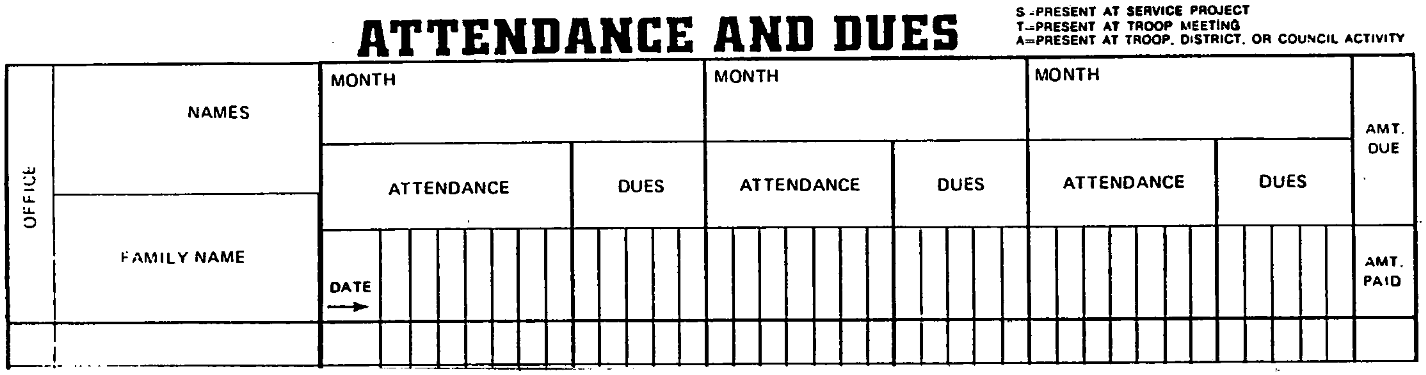
Figure 3.7 MERIT BADGE REPORT

## ADVANCEMENT

Enter date each requirement is completed.

| OFFICE | NAMES<br>FAMILY NAME | BECAME A SCOUT | PERSONAL GROWTH AGREEMENT CONFERENCE NO. 1 | SKILL AWARDS (ENTER DATE) |  |  |  |  |  |  |  |  |  |  |  | TENDERFOOT PROGRESS AWARD |  |  |  |  |  |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
|  |  |  |  | A. CAMPING | B. CITIZENSHIP | C. COMMUNICATION | D. COMMUNITY LIVING | E. CONSERVATION | F. COOKING | G. ENVIRONMENT | H. FAMILY LIVING | I. FIRST AID | J. HIKING | K. PHYSICAL FITNESS | L. SWIMMING | PARTICIPATION | SCOUT SPIRIT | CITIZENSHIP SKILL AWARD | SKILL AWARD (LETTER) | PERSONAL GROWTH CONFERENCE NO. 2 | PROGRESS REVIEW |
|  |  | MONTH & YEAR |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | MONTH & YEAR |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

Figure 3.8 ADVANCEMENT REPORT

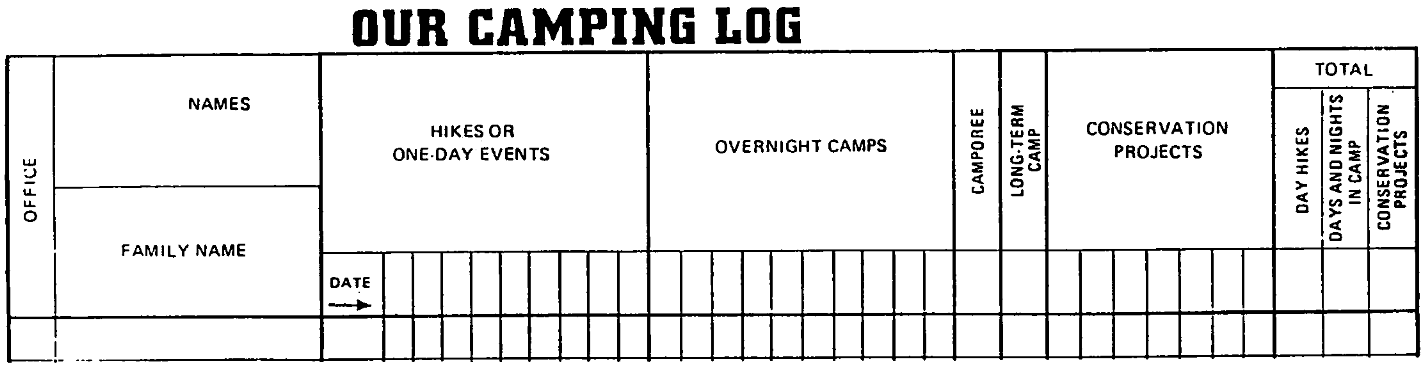


ATTENDANCE AND DUES

S -PRESENT AT SERVICE PROJECT
T=PRESENT AT TROOP MEETING
A=PRESENT AT TROOP, DISTRICT, OR COUNCIL ACTIVITY

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| OFFICE | NAMES | MONTH | | | | | | | | | | | | | MONTH | | | | | | | | | | | | MONTH | | | | | | | | | | | | AMT. DUE |
| | | ATTENDANCE | | | | | | | | DUES | | | | | ATTENDANCE | | | | | | | DUES | | | | | ATTENDANCE | | | | | | | DUES | | | | | |
| | FAMILY NAME | DATE → | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | AMT. PAID |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Figure 3.9 ATTENDANCE AND DUES REPORT



OUR CAMPING LOG

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| OFFICE | NAMES | HIKES OR ONE-DAY EVENTS | | | | | | | | | | OVERNIGHT CAMPS | | | | | | | | | | CAMPOREE | LONG-TERM CAMP | CONSERVATION PROJECTS | | | | | | | TOTAL | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | DAY HIKES | DAYS AND NIGHTS IN CAMP | CONSERVATION PROJECTS |
| | FAMILY NAME | DATE → | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Figure 3.10 OUR CAMPING LOG REPORT

OUR CAMPING LOG (Figure 3.10)

The scouts have to participate in many kinds of activities, many places, and some of these activities take many days. This report is intended to keep records of these actions.

ASSUMPTIONS ABOUT THE ENVIRONMENT OF THE 'BOY SCOUT TROOP [2]'

- For the troop program schedule, some activities can happen in one program day which will be performed by different groups of the scouts.
- A boy scout must have completed at least the 5th grade.
- A boy scout must be at least 10 and a half years old and older but not yet 18.
- Each scout will be assigned to one patrol group.
- The last name and first name together is a unique identification for each scout, which is also used to identify the birthdate, the address, the status, the progress award, the church, the home phone number, and so on.
- The code for the scout who has a completed uniform is 'Y', otherwise 'N'.
- The code for the scout who subscribes to Boy's Life

magazine is 'Y', otherwise 'N'.

- The guardian of the scout can be a guardian of more than one scout.
- Hiking and first aid skill awards are required for the Second class, but can be earned at any time.
- Camping and cooking skill awards are required for the First class, but can be earned at any time.
- First aid merit badge is required for the First class but may be earned at any time.
- A total of eight merit badges, including any four from the required list for the Eagle progress award are required for the Star award.
- A total of 15 merit badges, including any 7 from the required list for the Eagle progress award are required for the Life progress award.
- A total of 24 merit badges including 11 from the required list are required for the Eagle progress award.
- To be eligible to receive the Bronze palm, Gold palm, and Silver palm respectively, the scouts have to eran 5 more additional merit badges at each time.
- Patrol activities (A requirement to earn the National Camping Award): Each patrol of the troop participated in at least 3 of the following activities during the last 12 months: they attended camporee, held a day

hike, did a conservation project, attended a scout retreat, conducted a project for anniversary month, conducted a father-and-son camp-out, and attened a klondike derby.

- Short-term camp (A requirement to earn the National Camping Award): All patrols in the troop were represented in four or more short-term camp-out during the past 12 months.
- Long-term camp (A requirement to earn the National Camping Award): All patrols in the troop were represented in a long-term (6 or more consecutive days and nights) camp and at least 50 percent of the total boy membership of the troop attended.

## CHAPTER 4

## THE CONCEPTUAL MODEL OF A DATA BASE OF BOY SCOUT TROOP

Determine the data elements referenced in every report individually.

AD:ADDR The street of the adult leader's address.

AD:BPHONE The business phone number of the adult leader.

AD:CITY The city in which the adult leader lives.

AD:FIRST The first name of the adult leader.

AD:HPHONE The home phone number of the adult leader.

AD:LAST The last name of the adult leader.

AD:TR:NAME The troop name of the adult leader.

AD:SEX The sex of the adult leader.

AD:ZIP The state and zipcode in which the adult leader lives.

AMT:DUE The amount of money that the scout owes the troop.

AMT:PAID The amount of money that the scout pays according to the dues.

ATT:CODE The attendance code of the scout at the activities.

S - Present at service project

T - Present at troop meeting

A - Present at troop, district or council activity

N - Not attend

AWD:DATE The date the scout passed a skill award.

AWD:NAME The name of merit badges or skill awards.

BALANCE The total amount of money the scout owes the troop.

BOY:LIFE States whether the scout subscibes to Boy's Life magazine.

DATE:ACT The date that the membership activities occurred.

EVENT The activity that is scheduled for the scouts.

G:BPHONE The business phone number of the scout's guardian.

G:NAME The name of the scout's guardian.

G:OCCUP The occupation of the scout's guardian.

G:RELATE The relation of the guardian to the scout (:- mother,etc.).

LEADER The leadership post held by the scout.

LE:DATE The date of the leadership activities.

LOC The location that the activity takes place

MEMBER:ACT The membership activities of the scout (:- become scout, transferred,etc.).

NO:DAY The number of days that the activities takes place.

PATROLNAME The name of the patrol group to which the scout is assigned.

PRE:DATE Today's date

PROG:DATE Program date that is scheduled for the activities of the scouts.

RANK The scouts' progress award :

N - Not rank

T - Tenderfoot award.

2 - Second class award.

1 - First class award.

S - Star award.

L - Life award.

E - Eagle award.

Also : Bronze Palm, Gold Palm, and Silver Palm.

SC:ADDR The street of the scout's address.

SC:BIRTH The birth date of the scout.

SC:CHURCH The church or religion of the scout.

SC:CITY The city where the scout lives in.

SC:FIRST The scout's first name.

SC:HPHONE The scout's home phone number.

SC:LAST The scout's last name.

SC:OFFICE The scout's office if he has one, otherwise blank.

SC:ZIP The state and zipcode in which the scout lives

STATUS The status of the scout :

N - New scout who never before registered with BSA.

B - New scout who was a cub scout.

R - Registered scout.

S - Scout who separated and, reregistered.

T - Scout who transferred to troop.

TITLE The title of the adult leader.

TYPE The type of award that the scout received

B - Merit badge

S - skill award

UNIFORM This indicates whether the scout has a completed uniform.

## THE RELATIONSHIP BETWEEN THE DATA ELEMENTS

This section is to identify the primary key data elements and the nonkey data elements. It also developes the 3rd normal form relations for each set of data elements.

TROOP PROGRAM RECORD (Figure 4.1)

The data elements representing this report (figure 3.1) are :

PROG:DATE, EVENT, LOC

The relationships between the data elements are:

PROG:DATE ◄◄---►► EVENT

EVENT ◄◄---► LOC

For the first relation, there can be more than one EVENT (activity) occurring in one day (PROG:DATE), and for a given EVENT, it can be scheduled for several times in several days. In order to make the primary key of the relation be able to identify the nonkey data element and avoid transitive dependency, the 'EVENT' and the 'PROG:DATE' will be concatenated (combined together) and used as a primary key to identify the field 'LOC'.

Therefore, the relationship between the data elements is

PROG:DATE+EVENT ◄◄---► LOC

ADULT LEADERSHIP ROSTER (Figure 4.2)

The data elements representing this report (figure 3.2) are :

AD:LAST, AD:FIRST, AD:TR:NAME, TITLE, AD:ADDR, AD:CITY, AD:ZIP, AD:SEX, AD:HPHONE, AD:BPHONE

The relationships between the data elements are :

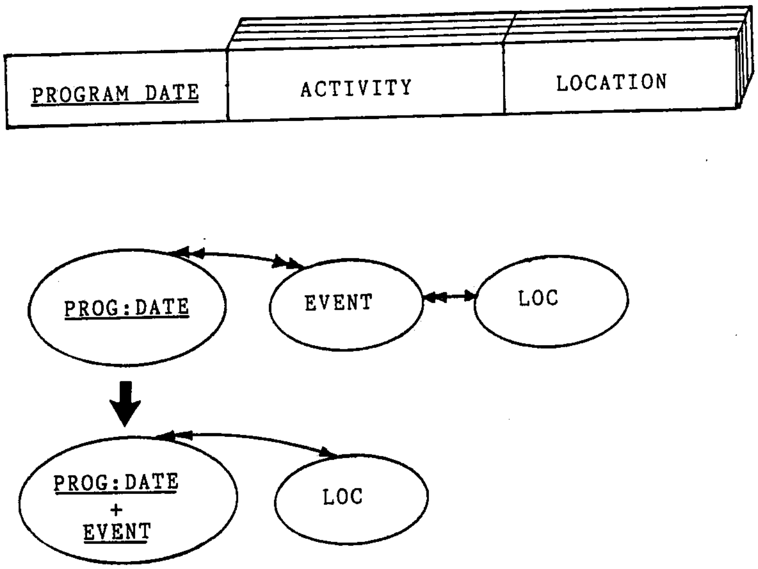


Figure 4.1 The relationship between the data elements in the 'TROOP PROGRAM RECORD'.

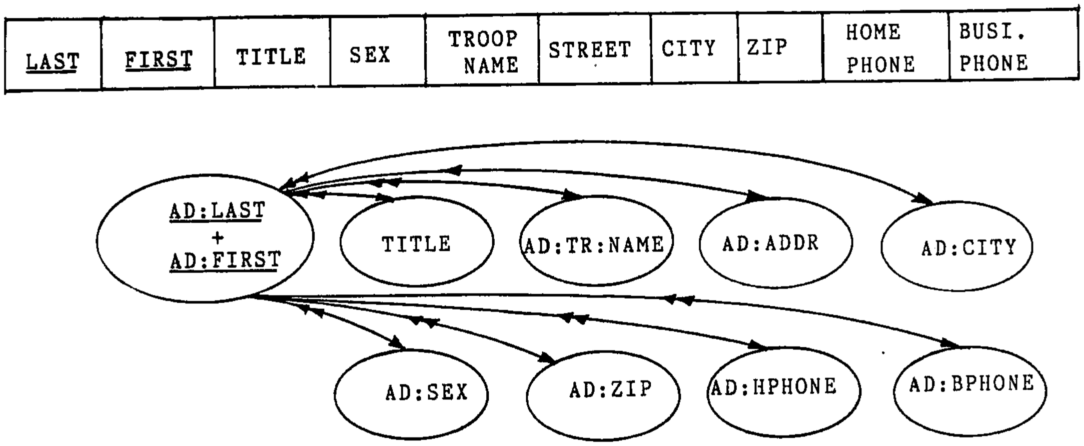


Figure 4.2 The relationship between the data elements in the 'ADULT LEADERSHIP ROSTER'.

AD:LAST+AD:FIRST ◄◄---► AD:TR:NAME, TITLE, AD:CITY,
AD:ZIP,AD:SEX

AD:LAST+AD:FIRST ◄----► AD:ADDR, AD:HPHONE, AD:BPHONE

These relationships are in the 3rd normal form. The primary key can identify the nonkey data elements and there is no transitive dependency.

### TROOP ROSTER (Figure 4.3)

The data elements representing this report (figure 3.3) are

SC:LAST, SC:FIRST, SC:OFFICE, SC:ADDR, SC:CITY, SC:ZIP, STATUS, AGE, RANK, UNIFORM, BOY:LIFE

The 'AGE' of the scout can be calculated from the present date and his birthdate.

The relationship between the elements is

(*)SC:LAST+SC:FIRST ◄◄---► SC:OFFICE, SC:ADDR, SC:CITY,
SC:ZIP, STATUS, RANK, UNIFORM
BOY:LIFE

This relation is in the 3rd normal form and the primary key can identify all nonkey data elements.

### FAMILY DATA (Figure 4.4)

The data elements representing this report (figure 3.4) are :

SC:LAST, SC:FIRST, SC:OFFICE, SC:HPHONE, SC:CHURCH, G:NAME,

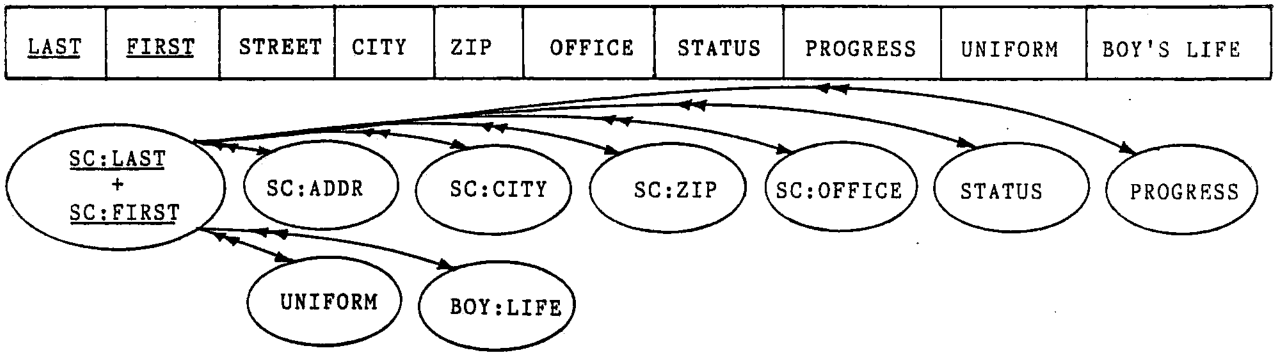


Figure 4.3 The relationship between the data elements in the 'TROOP ROSTER'.

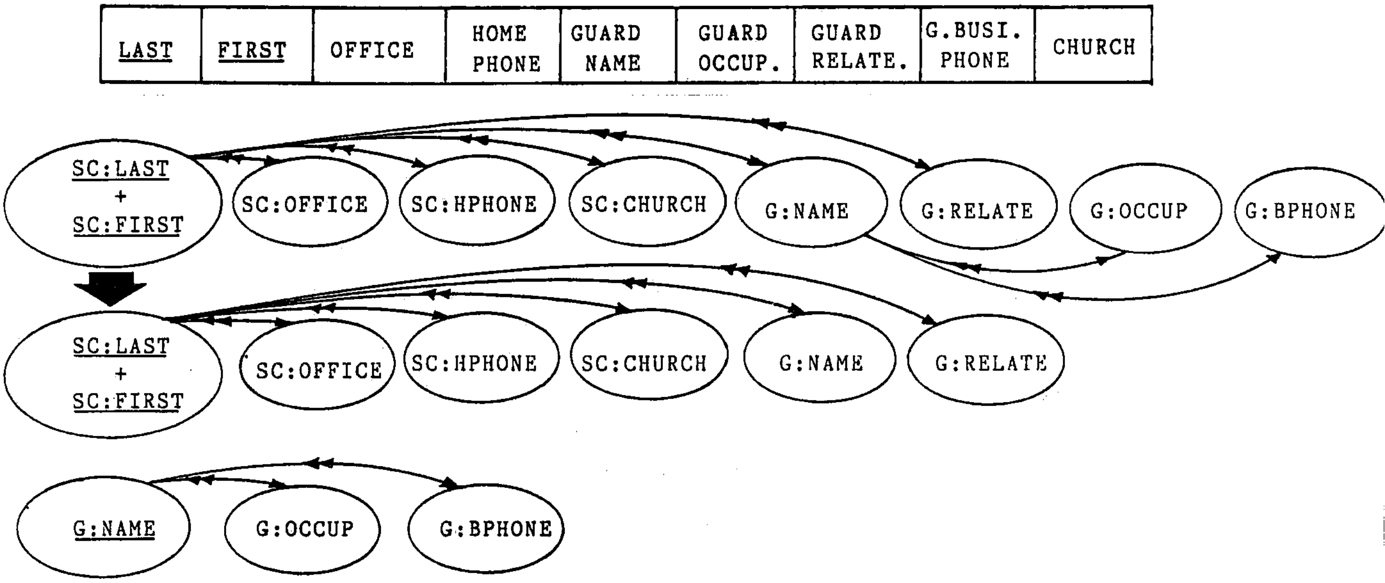


Figure 4.4 The relationship between the data elements in the 'FAMILY DATA'.

G:RELATE, G:OCCUP, G:BPHONE

The relationship between 'SC:LAST'+'SC:FIRST' and 'SC:OFFICE' is already shown in the relation (*).

The relationships of the data elements are :

SC:LAST+SC:FIRST ◀◀---▶ SC:HPHONE, SC:CHURCH, G:NAME,
G:RELATE

G:NAME ◀◀---▶ G:OCCUP, G:BPHONE

These 2 relationships are already in the 3rd normal form.

MEMBERSHIP DATA (Figure 4.5)

The data elements representing this report (figure 3.5) are:

SC:LAST, SC:FIRST, SC:OFFICE, SC:BIRTH, PATROLNAME, STATUS, MEMBER:ACT, DATE:ACT

The relationship between 'SC:LAST'+'SC:FIRST' and 'SC:OFFICE' is already shown in the relation (*).

The relationships between the data elements are

SC:LAST+SC:FIRST ◀◀---▶ SC:BIRTH, PATROLNAME, STATUS

SC:LAST+SC:FIRST ◀◀---▶▶ MEMBER:ACT

STATUS ◀◀---▶▶ MEMBER:ACT

MEMBER:ACT ◀◀---▶▶ DATE:ACT

The first relation is already in the 3rd normal form. However, the second, third, and fourth are not. In order to make them the 3rd normal form, the primary key must be able

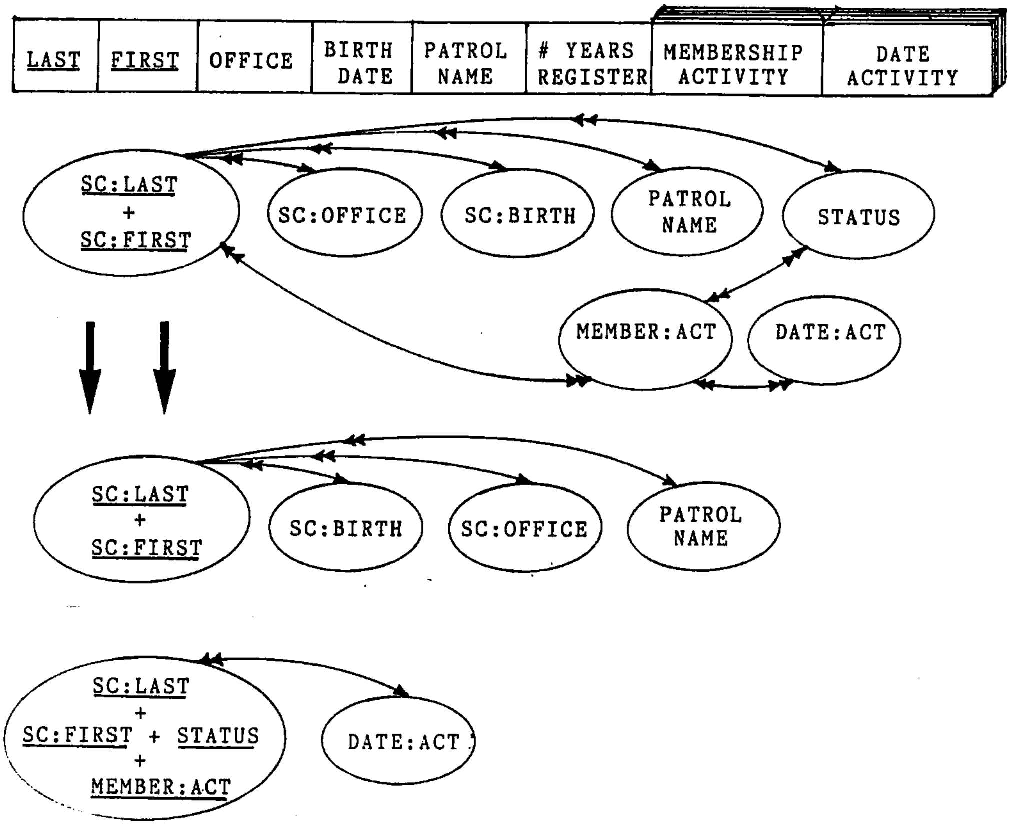


Figure 4.5 The relationship between the data elements in the 'MEMBERSHIP DATA'.

to identify the nonkey field. Some arrangements have been made. The field 'STATUS' and 'MEMBER:ACT' have been concatenated to the primary key 'SC:LAST'+'SC:FIRST' in order to identify 'DATE:ACT'.

The relationships of these data elements are :

SC:LAST+SC:FIRST ◄◄---► SC:BIRTH, PATROLNAME

SC:LAST+SC:FIRST+STATUS+MEMBER:ACT ◄◄---► DATE:ACT

### LEADERSHIP (Figure 4.6)

The data elements representing this report (figure 3.6) are :

SC:LAST, SC:FIRST, SC:OFFICE, LEADER, LE:DATE

The relationship between 'SC:LAST'+'SC:FIRST' and 'SC:OFFICE' is already shown in the relation (*).

The relationships of these data elements are

SC:LAST+SC:FIRST ◄◄---►► LEADER

LEADER ◄◄--►► LE:DATE

These 2 relations are not in the 3rd normal form. The 'LEADER' has been added to the primary key. The new relationship is :

SC:LAST+SC:FIRST+LEADER ◄◄---► LE:DATE

### AWARD (Figure 4.7, 4.8)

The data elements representing these reports (figure 3.6,3.7) are

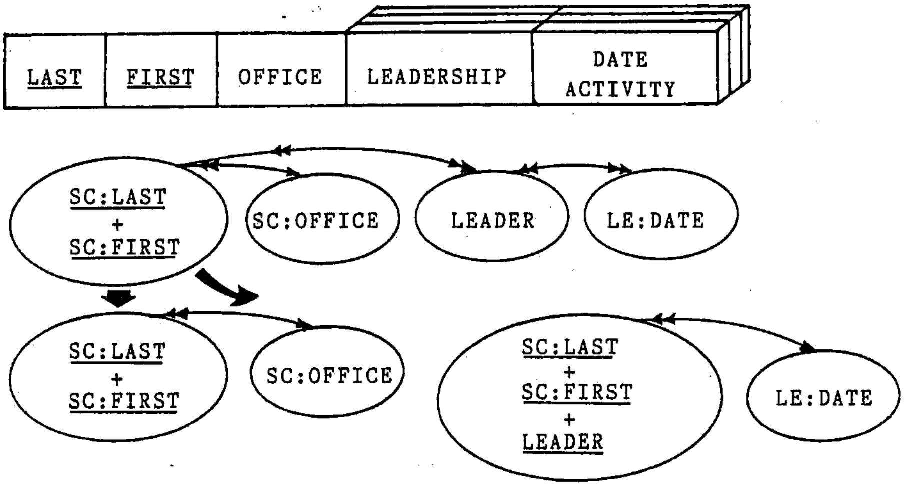


Figure 4.6 The relationship between the data elements in the 'LEADERSHIP RECORD'.

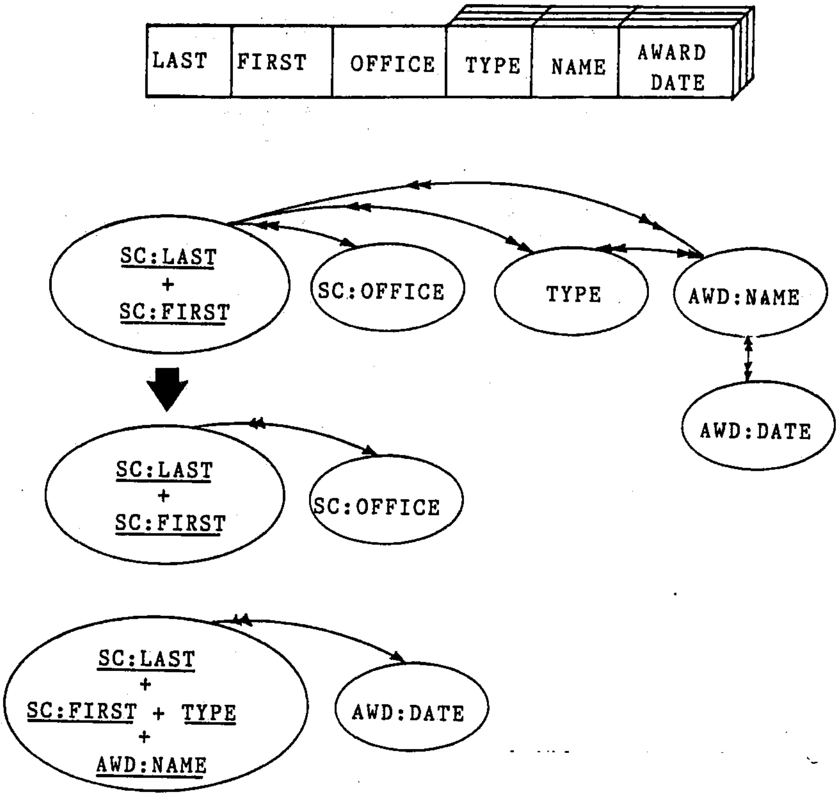


Figure 4.7 The relationship between the data elements in the 'AWARD' report

SC:LAST, SC:FIRST, SC:OFFICE, TYPE, AWD:NAME, AWD:DATE

The relationship between 'SC:LAST'+'SC:FIRST' and 'SC:OFFICE' is already shown in the relation (*).

The relationships between the data elements are

SC:LAST+SC:FIRST ◄◄--►► TYPE, AWD:NAME

AWD:NAME ◄◄--►► AWD:DATE

These 2 relationships are not in the 3rd normal form. The 'TYPE' of award and the 'AWD:NAME' are added to the primary key in order to identify the 'AWD:DATE'. The relation between these data elements is

SC:LAST+SC:FIRST+TYPE+AWD:NAME ◄◄---► AWD:DATE

ADVANCEMENT (Figure 4.8)

The data elements representing this report (figure 3.8) are :

SC:LAST, SC:FIRST, SC:OFFICE, 'BECAME SCOUT', When 'BECAME SCOUT', RANK, RANK:DATE

The relationship between 'SC:LAST'+'SC:FIRST' and 'SC:OFFICE' is already shown in the relation (*).

The 'BECAME SCOUT' and when 'BECAME SCOUT' are represented as 'MEMBER:ACT' and 'DATE:ACT' repectively, and their relation to the scout is shown on figure 4.5.

The relationships between the data elements are :

SC:LAST+SC:FIRST ◄◄--►► RANK

RANK ◄◄--►► RANK:DATE

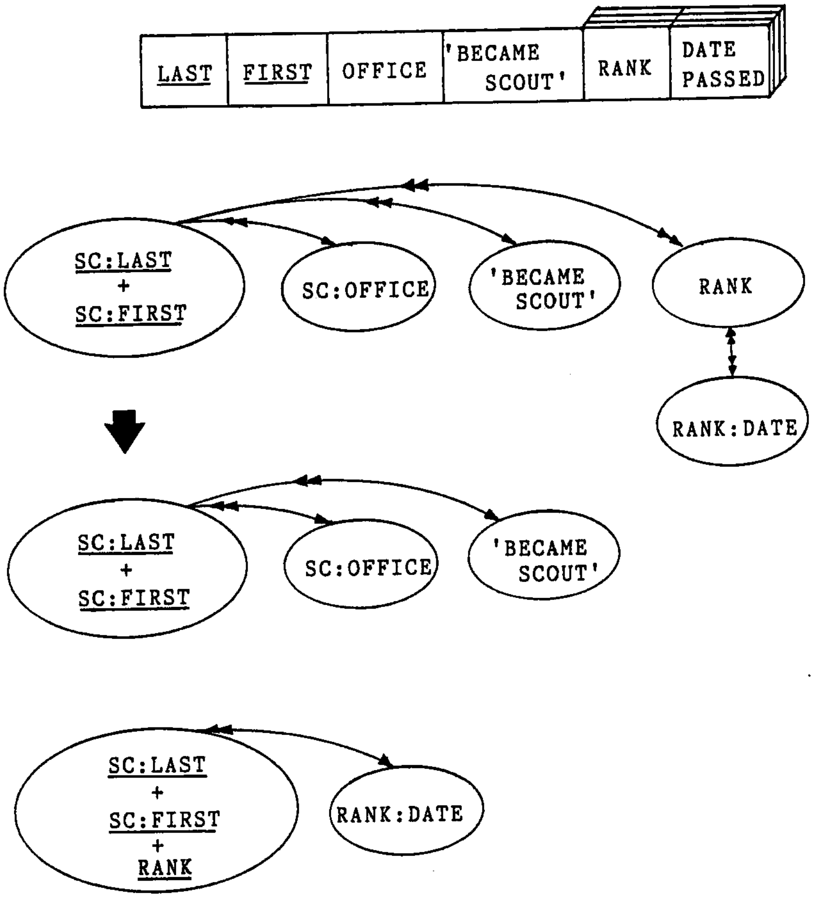


Figure 4.8 The relationship between the data elements in the 'ADVANCEMENT' report

After rearranging the new primary key, the new relationship of these data elements is:

SC:LAST+SC:FIRST+RANK ◄◄---► RANK:DATE

### ATTENDANCE AND DUES (Figure 4.9)

The data elements representing this report (figure 3.9) are :

SC:LAST, SC:FIRST, SC:OFFICE, PROG:DATE, ATT:CODE, AMT:DUE, AMT:PAID, TOTDUE, TOTPAID

The relationship between 'SC:LAST'+'SC:FIRST' and 'SC:OFFICE' is shown in the relation (*).

The data elements 'TOTDUE' and 'TOTPAID' can be calculated.

The relationships between the data elements are

SC:LAST+SC:FIRST+PROG:DATE ◄◄---► ATT:CODE, AMT:DUE,
AMT:PAID,BALANCE

The relationships are in the 3rd normal form.

### OUR CAMPING LOG (Figure 4.10)

The data elements representing this report (figure 3.10) are :

SC:LAST, SC:FIRST, SC:OFFICE, EVENT, PROG:DATE, ATT:CODE, NO:DAY, TOTDAY

The data element 'TOTDAY' can be calculated.

The relationship between 'SC:LAST'+'SC:FIRST' and

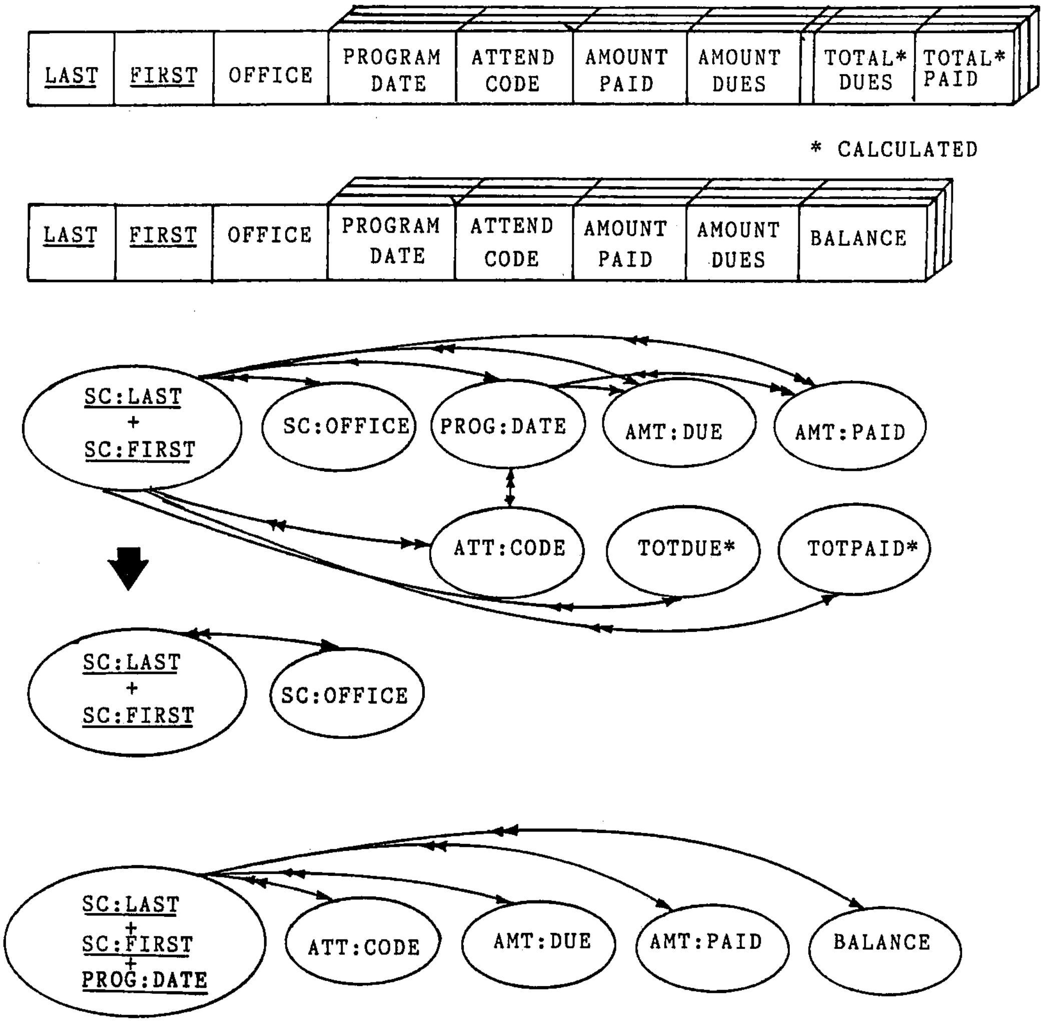


Figure 4.9 The relationship between the data elements in the 'ATTENDANCE AND DUES'.

'SC:OFFICE' is shown in the relation (*).

The relationships between the data elements are

SC:LAST+SC:FIRST ↞---↠ EVENT, PROG:DATE, ATT:CODE

EVENT ↞---↠ PROG:DATE, NO:DAY

In order to change the relationships into the 3rd normal form, the new relationships are :

SC:LAST+SC:FIRST+EVENT+PROG:DATE ↞----▸ ATT:CODE

PROG:DATE+EVENT ↞----▸ NO:DAY

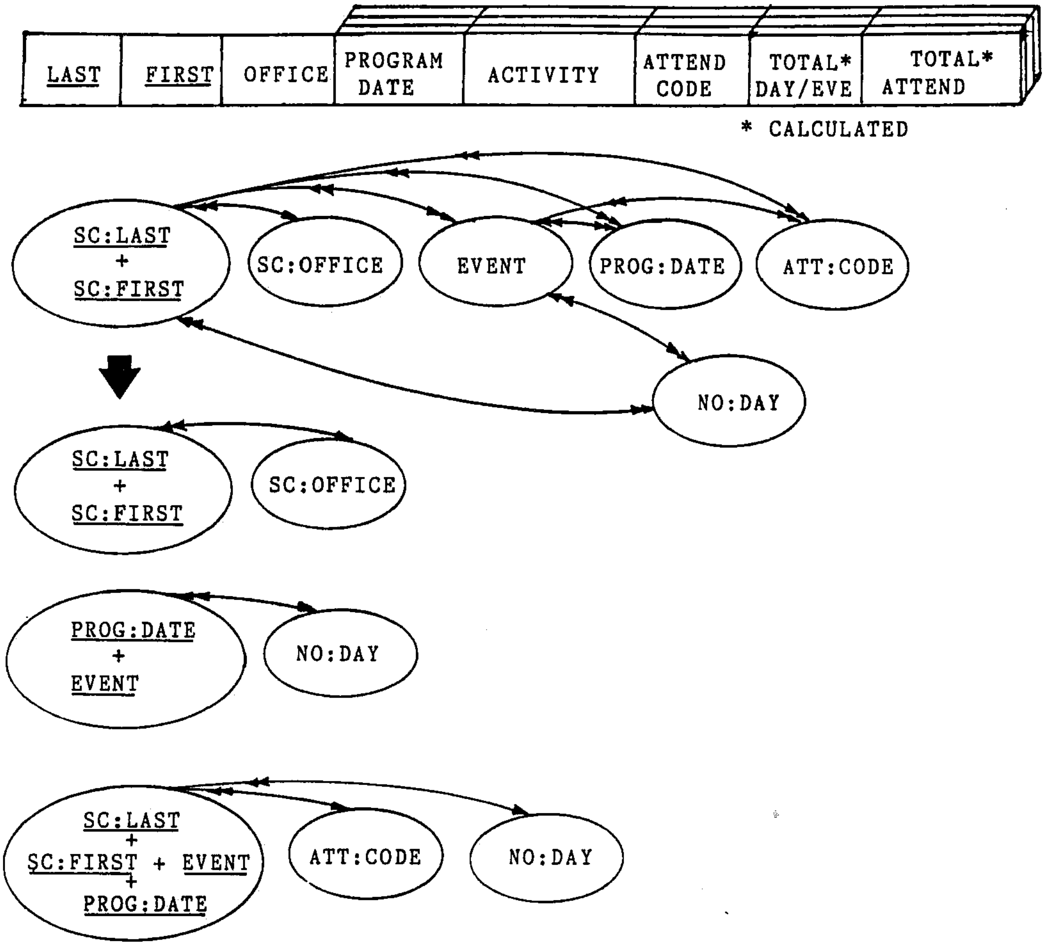


Figure 4.10 The relationship between the data elements in the 'OUR CAMPING LOG'.

The relationship between all the data elements are represented in the total set as follow :

PROG:DATE+EVENT ◀◀---▶ LOC, AMT:DUE, NO:DAY

AD:LAST+AD:FIRST ◀◀---▶ AD:TR:NAME, TITLE, AD:CITY, AD:ZIP, AD:SEX

AD:LAST+AD:FIRST ◀----▶ AD:ADDR, AD:HPHONE, AD:BPHONE

SC:LAST+SC:FIRST ◀◀---▶ SC:OFFICE, SC:ADDR, SC:CITY, SC:ZIP, STATUS, RANK, UNIFORM, BOY:LIFE, SC:BIRTH, PATROLNAME, SC:HPHONE, SC:CHURCH, G:NAME, G:RELATE

SC:LAST+SC:FIRST+STATUS+MERBER:ACT ◀◀---▶ DATE:ACT

SC:LAST+SC:FIRST+LEADER ◀◀---▶ LE:DATE

SC:LAST+SC:FIRST+TYPE+AWD:NAME ◀◀---▶ AWD:DATE

SC:LAST+SC:FIRST+RANK ◀◀---▶ RANK:DATE

SC:LAST+SC:FIRST+PROG:DATE ◀◀---▶ ATT:CODE, AMT:PAID, BALANCE

G:NAME ◀◀---▶ G:OCCUP, G:BPHONE

THE FILE DESCRIPTION ON THE BASIS OF THE RELATIONSHIPS

The following is the structure of the fields of each report :

TROOP PROGRAM RECORD

FILE NAME TRPROG.DBF

| FIELD | NAME | TYPE | WIDTH | DEC |
|---|---|---|---|---|
| 1 | PROG:DATE | C | 008 | |
| 2 | EVENT | C | 025 | |
| 3 | LOC | C | 015 | |
| 4 | AMT:DUE | N | 007 | 002 |
| 5 | NO:DAY | N | 002 | 000 |
| ** TOTAL ** | | | 057 | 002 |

ADULT LEADERSHIP ROSTER

FILE NAME ADLEAD.DBF

| FIELD | NAME | TYPE | WIDTH | DEC |
|---|---|---|---|---|
| 1 | AD:LAST | C | 015 | |
| 2 | AD:FIRST | C | 015 | |
| 3 | TITLE | C | 015 | |
| 4 | AD:TR:NAME | C | 010 | |
| 5 | AD:ADDR | C | 025 | |
| 6 | AD:CITY | C | 015 | |
| 7 | AD:ZIP | C | 008 | |
| 8 | AD:SEX | C | 001 | |

|  |  |  |  |
|---|---|---|---|
| 9 | AD:HPHONE | C | 013 |
| 10 | AD:BPHONE | C | 013 |
| ** TOTAL ** |  |  | 0130 |

TROOP ROSTER

FILE NAME SCRECORD.DBF

| FIELD | NAME | TYPE | WIDTH | DEC |
|---|---|---|---|---|
| 1 | SC:LAST | C | 015 |  |
| 2 | SC:FIRST | C | 015 |  |
| 3 | SC:BIRTH | C | 008 |  |
| 4 | SC:ADDR | C | 025 |  |
| 5 | SC:CITY | C | 015 |  |
| 6 | SC:ZIP | C | 008 |  |
| 7 | SC:HPHONE | C | 013 |  |
| 8 | SC:CHURCH | C | 010 |  |
| 9 | SC:OFFICE | C | 010 |  |
| 10 | PATROLNAME | C | 010 |  |
| 11 | STATUS | C | 001 |  |
| 12 | UNIFORM | C | 001 |  |
| 13 | BOY:LIFE | C | 001 |  |
| 14 | RANK | C | 001 |  |
| 15 | G:NAME | C | 025 |  |
| 16 | G:RELATE | C | 010 |  |
| ** TOTAL ** |  |  | 0163 |  |

MEMBERSHIP

FILE NAME MEMBER.DBF

| FIELD | NAME | TYPE | WIDTH | DEC |
|---|---|---|---|---|
| 1 | SC:LAST | C | 015 | |
| 2 | SC:FIRST | C | 015 | |
| 3 | STATUS | C | 001 | |
| 4 | MEMBER:ACT | C | 020 | |
| 5 | DATE:ACT | C | 008 | |
| ** TOTAL ** | | | 054 | |

FAMILY DATA

FILE NAME GUARDIAN.DBF

| FIELD | NAME | TYPE | WIDTH | DEC |
|---|---|---|---|---|
| 1 | G:NAME | C | 025 | |
| 2 | G:OCCUP | C | 015 | |
| 3 | G:BPHONE | C | 013 | |
| ** TOTAL ** | | | 053 | |

LEADERSHIP

FILE NAME LEADER.DBF

| FIELD | NAME | TYPE | WIDTH | DEC |
|---|---|---|---|---|
| 1 | SC:LAST | C | 015 | |
| 2 | SC:FIRST | C | 015 | |
| 3 | LEADER | C | 020 | |
| 4 | LE:DATE | C | 008 | |

| | | | | |
|---|---|---|---|---|
| ** TOTAL ** | | | 053 | |

MERIT AWARD

FILE NAME AWARD.DBF

| FIELD | NAME | TYPE | WIDTH | DEC |
|---|---|---|---|---|
| 1 | SC:LAST | C | 015 | |
| 2 | SC:FIRST | C | 015 | |
| 3 | TYPE | C | 001 | |
| 4 | AWD:NAME | C | 020 | |
| 5 | AWD:DATE | C | 008 | |
| ** TOTAL ** | | | 054 | |

ADVANCEMENT

FILE NAME ADVANCE.DBF

| FIELD | NAME | TYPE | WIDTH | DEC |
|---|---|---|---|---|
| 1 | SC:LAST | C | 015 | |
| 2 | SC:FIRST | C | 015 | |
| 3 | RANK | C | 001 | |
| 4 | RANK:DATE | C | 008 | |
| ** TOTAL ** | | | 034 | |

ATTENDANCE AND DUES

FILE NAME ACCOUNT.DBF

| FIELD | NAME | TYPE | WIDTH | DEC |
|---|---|---|---|---|
| 1 | SC:LAST | C | 015 | |

```
2          SC:FIRST         C      015
3          PROG:DATE        C      008
4          EVENT            C      025
5          ATT:CODE         C      001
6          AMT:PAID         N      007     002
7          BALANCE          N      009     002
  ** TOTAL ***                     080     004
```

# APPENDIX A

## GLOSSARY

CONCATENATE: To link together

FILE: A collection of related records treated as a unit.

HARDWARE: Physical equipment such as electronic magnetic, and mechanical devices.

KEY: A data item uses to identify or locate a record.

ON-LINE: A system in which the input data enters the computer directly from their point of origin and/or output data are transmitted directly to where they are used.

OPERATING SYSTEM: An organized collection of software that controls the overall operations of a computer

PRIMARY KEY: A key (data item or items) which is used as a unique identification of a record

RECORD: A group of related fields of information treated as a unit by an application program

SOFTWARE: A set of programs, procedures, routines, and documents associated with the operation of a computer system.

SYSTEM: A composite of equipment, skills, techniques, and information capable of performing and/or supporting an operational role in attaining specified management objectives.

# APPENDIX B

# SOURCE PROGRAM LISTING

```
MAIN.CMD
*
*
*
DO setup
STORE t to a
DO WHILE a
   ERASE
   STORE '        ' TO pre:date
   @ 10,11 SAY"ENTER TODAY'S DATE (AS MM/DD/YY)" GET ;
pre:date PICTURE '99/99/99'
   READ
   STORE pre:date TO cdate
   DO ckdate
   IF state = '1'
      @ 23,10 SAY'INVALID DATE ERROR - TRY AGAIN'
      DO delay
      LOOP
   ENDIF
   STORE f TO a
ENDDO
SAVE TO B:memfile
SET DATE TO pre:date
DO WHILE T
   ERASE
   STORE ' ' TO main:sel
   @ 4,25 SAY'MAIN MENU'
   @ 4,60 SAY DATE()
   @ 8,18 SAY'0 - EXIT TO dBASE'
   @ 10,18 SAY'1 - TROOP SCHEDULE'
   @ 11,18 SAY'2 - ADULT LEADERSHIP INFORMATION'
   @ 12,18 SAY"3 - SCOUT'S INFORMATION"
   @ 13,18 SAY'4 - MISCELLANEOUS'
   @ 15,18 SAY'9 - QUIT TO OPERATION SYSTEM'
   @ 20,10 SAY'ENTER YOUR SELECTION' GET main:sel
   READ
   DO CASE
      CASE main:sel = '0'
         RELEASE ALL
         @ 22,10 SAY'THIS IS THE END OF THE PROGRAM.'
         @ 23,10 SAY' YOU ARE ENTERING THE dBASE OPERATION;
 SYSTEM.'
         DO bakup
         CANCEL
      CASE main:sel = '9'
         RELEASE ALL
         @ 23,10 SAY' THIS IS THE END OF THE PROGRAM.'
         CLEAR
         DO bakup
         QUIT
      CASE main:sel = '1'
         DO troop
      CASE main:sel = '2'
         DO adult
      CASE main:sel = '3'
         DO scout
      CASE main:sel = '4'
         DO misc
      OTHERWISE
         @ 23,10 SAY'WRONG CODE - TRY AGAIN'
         DO delay
   ENDCASE
ENDDO
```

```
CKDATE.CMD
*
*
*
STORE ' ' TO state
STORE VAL($(cdate,1,2)) TO mont
STORE VAL($(cdate,4,2)) TO dat
STORE VAL($(cdate,7,2)) TO yr
STORE (yr/4) - INT(yr/4) TO result
IF (mont  1 .OR. mont  12) .OR. (dat=0 .OR. dat 31)
   STORE '1' TO state
ENDIF
IF (mont = 4 .OR. mont = 6 .OR. mont = 9 .OR. mont = 11);
 .AND. (dat=0 .OR. dat  30)
   STORE '1' TO state
ENDIF
IF mont = 2 .AND. result = 0 .AND. (dat=0 .OR. dat29)
   STORE '1' TO state
ENDIF
IF mont = 2 .AND. result  0 .AND. (dat=0 .OR. dat28)
   STORE '1' TO state
ENDIF
RETURN
```

```
BAKUP.CMD
*
*
*
DO setup
? 'The program is copying files for back up.'
USE B:trprog
COPY TO B:btrprog
USE B:adlead
COPY TO B:badlead
USE B:screcord
COPY TO B:bscrecord
USE B:member
COPY TO B:bmember
USE B:leader
COPY TO B:bleader
USE B:award
COPY TO B:baward
USE B:account
COPY TO B:baccount
USE B:advance
COPY TO B:badvance
USE B:guardian
COPY TO B:bguardian
?' The files have already copied. '
DO delay
RETURN
```

```
MONAME.CMD
*
*
*
DO CASE
   CASE mont ='01'
      STORE 'JANUARY' TO mon
   CASE mont ='02'
      STORE 'FEBRUARY' TO mon
   CASE mont ='03'
      STORE 'MARCH' TO mon
   CASE mont ='04'
      STORE 'APRIL' TO mon
   CASE mont ='05'
      STORE 'MAY' TO mon
   CASE mont ='06'
      STORE 'JUNE' TO mon
   CASE mont ='07'
      STORE 'JULY' TO mon
   CASE mont ='08'
      STORE 'AUGUST' TO mon
   CASE mont ='09'
      STORE 'SEPTEMBER' TO mon
   CASE mont ='10'
      STORE 'OCTOBER' TO mon
   CASE mont ='11'
      STORE 'NOVEMBER' TO mon
   CASE mont ='12'
      STORE 'DECEMBER' TO mon
ENDCASE
RETURN
```

```
SETUP.CMD
*
*
*
SET TALK OFF
SELECT PRIMARY
SET FORMAT TO SCREEN
SET PRINT OFF
SET CONSOLE ON
RETURN
```

```
DELAY.CMD
*
*
*
STORE 1 TO xx
DO WHILE xx  60
   STORE xx + 1 TO xx
ENDDO
RETURN
```

```
MBADD.CMD
*
*
*
USE B:member
APPEND BLANK
REPLACE sc:last WITH l:name,sc:first WITH f:name,status ;
WITH n:sta,member:act WITH memact,date:act WITH actdate
RETURN
```

```
ADVADD.CMD
*
*
*
USE B:advance
APPEND BLANK
REPLACE sc:last WITH l:name,sc:first WITH f:name,rank ;
WITH t:rank,rank:date WITH nr:date
RETURN
```

```
TROOP.CMD
*
*
*
DO setup
DO WHILE t
   STORE ' ' TO tr:sel
   ERASE
   @ 3,20 SAY'TROOP INFORMATION MENU'
   @ 3,60 SAY DATE()
   @ 6,15 SAY'0 - Exit'
   @ 8,15 SAY'1 - Add new program for troop'
   @ 10,15 SAY'2 - Delete cancelled program '
   @ 12,15 SAY'3 - Update the program'
   @ 14,15 SAY"4 - List  report troop's program"
   @ 19,10 SAY'Enter your selection ' GET tr:sel
   READ
   DO CASE
      CASE tr:sel ='0'
         RELEASE ALL
         RETURN
      CASE tr:sel ='1'
         DO tradd
      CASE tr:sel ='2'
         DO trdel
      CASE tr:sel ='3'
         DO trupd
      CASE tr:sel ='4'
         DO trlist
      OTHERWISE
         @ 23,10 SAY'WRONG CODE TRY AGAIN'
         DO delay
   ENDCASE
ENDDO
RETURN
```

```
TRADD.CMD
*
*
*
DO setup
DO WHILE t
   STORE t to a
   DO WHILE a
      STORE '        ' TO n:date
      STORE '                         ' TO n:act
      STORE '                    ' TO n:loc
      STORE 0 TO n:due
      STORE 0 TO n:no:day
      ERASE
      @ 5,22 SAY'NEW TROOP PROGRAM'
      @ 5,60 SAY DATE()
      @ 7,11 SAY'PROGRAM DATE (MM/DD/YY) ' GET n:date ;
PICTURE '99/99/99'
      READ
      IF n:date ='  /  /  '
         RELEASE ALL
         RETURN
      ENDIF
      STORE n:date TO cdate
* check valid date
      DO ckdate
      IF state = '1'
         @ 23,10 SAY'INVALID DATE ERROR - TRY AGAIN'
         DO delay
         LOOP
      ENDIF
      STORE f TO a
   ENDDO
   USE B:trprog
   @ 9,11 SAY'ACTIVITY                 ' GET n:act
   READ
   COUNT TO no:rec FOR prog:date = n:date .AND. ;
TRIM(!(event)) = TRIM(!(n:act))
   IF no:rec 0
      @ 23,10 SAY' This program has been already added.'
      DO delay
      LOOP
   ENDIF
   @ 11,11 SAY'LOCATION                 ' GET n:loc
   @ 13,11 SAY'AMOUNT DUE               ' GET n:due PICTURE;
'999.99'
   @ 15,11 SAY'HOW LONG (DAYS)          ' GET n:no:day ;
PICTURE '99'
```

```
   READ
   APPEND BLANK
   REPLACE prog:date WITH n:date,event WITH n:act,loc WITH;
 n:loc,amt:due WITH n:due,no:day WITH n:no:day
ENDDO
```

```
TRDEL.CMD
*
*
*
DO setup
RESTORE FROM B:memfile
STORE 'T' TO first
STORE 'T' TO aa
DO WHILE aa = 'T'
   ERASE
   STORE ' 'TO state
   STORE '        ' TO t:date
   STORE '                         ' TO t:event
   STORE '                    ' TO t:loc
   STORE 0 TO t:due
   STORE 0 TO t:noday
   STORE '        ' TO pdate
   STORE ' ' TO select
   STORE ' ' TO decide
   STORE '                         ' TO act
   STORE 0 TO no
   USE B:trprog
   COUNT TO no:rec
   IF no:rec = 0
      @ 23,10 SAY'The file of the troop program is ;
empty now.'
      DO delay
      RELEASE ALL
      RETURN
   ENDIF
   @ 1,10 SAY' DELETE A PARTICULAR TROOP PROGRAM'
   @ 1,60 SAY DATE()
   @ 3,5 SAY'NOTE: THE PROGRAM MUST BE THE FUTURE ;
SCHEDULE ONLY'
   @ 4,11 SAY'PRESS RETURN TO STOP'
   @ 6,10 SAY'Enter program date' GET pdate PICTURE ;
'99/99/99'
   READ
   IF pdate = '  /  /  '
      IF first = 'T'
         RELEASE ALL
         RETURN
      ENDIF
      STORE 'F' TO aa
      LOOP
   ENDIF
   COUNT TO no:rec FOR prog:date = pdate
   IF no:rec = 0
```

```
      @ 23,10 SAY'There is no program on that day'
      DO delay
      LOOP
   ENDIF
   IF $(pdate,7,2)  $(pre:date,7,2)
      STORE '1' TO state
   ENDIF
   IF $(pdate,7,2) = $(pre:date,7,2) .AND. $(pdate,1,2)  ;
$(pre:date,1,2)
      STORE '1' TO state
   ENDIF
   IF $(pdate,7,2) = $(pre:date,7,2) .AND. $(pdate,1,2) = ;
$(pre:date,1,2) .AND. $(pdate,4,2)  $(pre:date,4,2)
      STORE '1' TO state
   ENDIF
   IF state = '1'
      @ 23,10 SAY'THIS PROGRAM CANNOT BE DELETED NOW, IT ;
IS NOT A FUTURE SCHEDULE'
      DO delay
      LOOP
   ENDIF
   IF no:rec = 1
      LOCATE FOR prog:date = pdate
      STORE prog:date TO t:date
      STORE event TO t:event
      STORE loc TO t:loc
      STORE amt:due TO t:due
      STORE no:day TO t:noday
      @ 7,17 SAY'ACTIVITY        '+TRIM(event)
      @ 8,17 SAY'LOCATION        '+TRIM(loc)
      @ 9,17 SAY'AMOUNT DUE      '+STR(amt:due,6,2)
      @ 10,17 SAY'HOW LONG (DAYS)'+STR(no:day,2,0)
      @ 13,10 SAY'Is this the program you wish to delete? ;
(Y OR N)' GET decide
      READ
      IF !(decide) = 'N'
         LOOP
      ENDIF
      USE B:trbakup
      APPEND BLANK
      REPLACE prog:date WITH t:date,event WITH t:event,;
loc WITH t:loc,amt:due WITH amt:due,no:day WITH t:noday
      USE B:trprog
      DELETE FOR prog:date = pdate
      PACK
      STORE 'F' TO first
   ENDIF  (* end if of no:rec = 1 *)
   IF no:rec 1
      STORE no:rec TO norec
      STORE 'T' TO a
      DO WHILE a = 'T'
         ERASE
         @ 1,3 SAY'There are '+STR(norec,2,0)+;
' activities  on that day'
         @ 3,10 SAY'NO.'
         @ 3,17 SAY"ACTIVITY"
         @ 3,47 SAY'LOCATION'
         @ 3,63 SAY'HOW LONG'
         STORE 4 TO row
         COPY TO B:temp FOR prog:date = pdate
         STORE 1 TO no
         SELECT SECONDARY
         USE B:temp
         GOTO TOP
         DO WHILE .NOT. EOF
            @ row,2 SAY no
            @ row,17 SAY event
            @ row,47 SAY loc
            @ row,63 SAY no:day
            STORE row + 1 TO row
            STORE no+1 TO no
            SKIP
         ENDDO
         STORE row+1 TO row
         @ row,10 SAY'Please specify the activity you ;
wish to delete'
         STORE row+1 TO row
         @ row,10 SAY'(Please type exact activity as ;
listing above)'
         STORE row + 1 TO row
         @ row,10 SAY'ACTIVITY' GET act
         READ
         IF act = '                          '
            STORE 'F' TO a
            STORE '1' TO flag
            LOOP
         ENDIF
         COUNT TO no:rec FOR !(event) = !(act)
         IF no:rec = 0
            @ 23,10 SAY'NOT FOUND PLEASE TRY AGAIN'
            DO delay
            STORE '                          ' TO act
            @ row,2 SAY'                                  ;
                   '
            @ 23,10 SAY'                                   '
            LOOP
         ENDIF
         STORE 'F' TO a
      ENDDO
      IF flag = '1'
```

```
         STORE 'F' TO aa
         LOOP
      ENDIF
      STORE row+1 TO row
      @ row,10 SAY'Please confirm that you wish to ;
delete the program on '+pdate
      STORE row+1 TO row
      @ row,10 SAY"the activity named '"+TRIM(act)+"' - Y ;
OR N -" GET decide
      READ
      IF !(decide) = ' '
         STORE 'F' TO aa
         LOOP
      ENDIF
      IF !(decide) = 'N'
         LOOP
      ENDIF
      USE B:trprog
      LOCATE FOR prog:date = pdate .AND. TRIM(!(event)) = ;
TRIM(!(act))
      STORE prog:date TO t:date
      STORE event TO t:event
      STORE loc TO t:loc
      STORE amt:due TO t:due
      STORE no:day TO t:noday
      USE B:trbakup
      APPEND BLANK
      REPLACE prog:date WITH t:date,event WITH t:event,;
loc WITH t:loc,amt:due WITH t:due,no:day WITH t:noday
      USE B:trprog
      DELETE FOR prog:date = pdate .AND. !(event) = !(act)
      PACK
      STORE 'F' TO first
      DELETE FILE B:temp
   ENDIF (* end if of no:rec  1 *)
ENDDO
* list the records of deletion
ERASE
@                                                        5,10
SAY'****************************************************'
@ 6,10 SAY'*                                             ;
    *'
@ 7,10 SAY'*        PREPARE THE PRINTER READY TO PRINT   ;
    *'
@ 8,10 SAY'*        THE RECORD OF CANCELLED SCHEDULE     ;
    *'
@ 9,10 SAY'*                                             ;
    *'
@ 10,10 SAY'*        WHEN YOU ARE READY PRESS RETURN   ;
     *'
```

```
@ 11,10 SAY'*                                            ;
     *'
@ 12,10 SAY'******************************************;
********'
WAIT
SET FORMAT TO PRINT
SET PRINT ON
SET CONSOLE OFF
STORE 55 TO line
STORE 'EJECT' TO npage
STORE 'trbakup' TO trfile
ERASE
DO tllistl
SET FORMAT TO SCREEN
SET PRINT OFF
SET CONSOLE ON
ERASE
@ 10,10 SAY' THE REPORT ON THE DELETED TROOP SCHEDULE HAS ;
FINISHED'
DO delay
USE B:trbakup
DELETE ALL
PACK
RELEASE ALL
RETURN
```

```
TRUPD.CMD
*
*
*
DO setup
RESTORE FROM B:memfile
SELECT PRIMARY
USE B:trprog
COUNT TO no:rec
IF no:rec =0
   @ 23,10 SAY'I am sorry - This file is empty'
   DO delay
   RETURN
ENDIF
DO WHILE t
   ERASE
   STORE '        ' TO pdate
   STORE 0 TO nodate
   STORE '        ' TO t:date
   STORE '                         ' TO t:act
   STORE '              ' TO t:loc
   STORE 0 TO t:due
   STORE 0 TO t:no:day
   @ 3,20 SAY'UPDATE TROOP PROGRAM'
   @ 3,60 SAY DATE()
   @ 5,6 SAY'NOTE : Enter new informations in the areas ;
you wish to update,'
   @ 6,19 SAY'Otherwise leave them blank'
   @ 8,10 SAY'Enter program date ' GET pdate PICTURE ;
'99/99/99'
   READ
   IF pdate = '  /  /  '
      RELEASE ALL
      RETURN
   ENDIF
   COUNT TO nodate FOR prog:date = pdate
   IF nodate = 0
      @ 23,10 SAY' There is no such record try again'
      DO delay
      LOOP
   ENDIF
   IF nodate  1
      STORE 't' TO a
      DO WHILE a ='t'
         ERASE
         @ 1,10 SAY'There are '+STR(nodate,2,0)+;
' activities in that day:'
         COPY TO B:temp FOR prog:date = pdate
         USE B:temp
         STORE 3 TO row
         STORE 1 TO no
         @ row,12 SAY'NO.'
         @ row,20 SAY'ACTIVITY'
         STORE row+1 TO row
         GOTO TOP
         DO WHILE .NOT. EOF
            @ row,12 SAY no USING '99'
            @ row,20 SAY TRIM(event)
            STORE row+1 TO row
            STORE no+1 TO no
            SKIP
         ENDDO
         @ row,5 SAY'Please specify the activity you wish ;
to update'
         STORE row+1 TO row
         @ row,5 SAY'(Please type exactly as listing ;
above)'
         STORE row+1 TO row
         STORE '                          ' TO act
         @ row,5 SAY'ACTIVITY' GET act
         READ
         IF act = '                          '
            RELEASE ALL
            RETURN
         ENDIF
         USE B:trprog
         LOCATE FOR !(event) = !(act) .AND. prog:date = ;
pdate
         IF !(event)  !(act) .AND. prog:date = pdate
            @ 23,10 SAY'NOT FOUND - TRY AGAIN'
            DELETE FILE B:temp
            DO delay
            LOOP
         ENDIF
         IF !(event) = !(act) .AND. prog:date = pdate
            ERASE
            STORE prog:date TO t:date
            STORE event TO t:act
            STORE loc TO t:loc
            STORE amt:due TO t:due
            STORE no:day TO t:no:day
            @ 6,1 SAY'PROGRAM DATE   '+prog:date
            @ 6,40 SAY'CHANGE TO ' GET t:date PICTURE ;
'99/99/99'
            @ 7,1 SAY'ACTIVITY       '+TRIM(!(event))
            @ 7,40 SAY'CHANGE TO ' GET t:act
            @ 8,1 SAY'LOCATION       '+TRIM(!(t:loc))
            @ 8,40 SAY'CHANGE TO ' GET t:loc
            @ 9,1 SAY'AMOUNT DUE     '+STR(t:due,6,2)
            @ 9,40 SAY'CHANGE TO ' GET t:due
            @ 10,1 SAY'HOW LONG (DAYS)'+STR(t:no:day,2,0)
            @ 10,40 SAY'CHANGE TO ' GET t:no:day
            READ
            IF prog:date  t:date .OR. !(event)  ;
!(t:act)
               USE B:trprog
               COUNT TO no:rec FOR prog:date = t:date ;
.AND. !(event) = !(t:act)
               IF no:rec  0
                  @ 23,10 SAY'CANNOT MAKE CHANGES ;
BECAUSE THE RECORD HAS ALREADY EXISTED'
                  DO delay
                  LOOP
               ENDIF
            ENDIF
            REPLACE prog:date WITH t:date,event WITH ;
t:act,loc WITH t:Loc,amt:due WITH t:due,no:day WITH  ;
t:no:day FOR !(event) = !(act) .AND. prog:date = pdate
         ENDIF
         @ 14,10 SAY'The troop program has been update'
         DO delay
         STORE 'f' TO a
      ENDDO
   ENDIF
   IF nodate = 1
      SELECT PRIMARY
      LOCATE FOR prog:date = pdate
      STORE prog:date TO t:date
      STORE event TO t:act
      STORE loc TO t:Loc
      STORE amt:due TO t:due
      STORE no:day TO t:no:day
      @ 11,1 SAY'PROGRAM DATE   '+prog:date
      @ 11,40 SAY'CHANGE TO ' GET t:date PICTURE '99/99/99'
      @ 12,1 SAY'ACTIVITY       '+TRIM(!(event))
      @ 12,40 SAY'CHANGE TO ' GET t:act
      @ 13,1 SAY'LOCATION       '+TRIM(!(loc))
      @ 13,40 SAY'CHANGE TO ' GET t:Loc
      @ 14,1 SAY'AMOUNT DUE    '+STR(amt:due,6,2)
      @ 14,40 SAY'CHANGE TO ' GET t:due
      @ 15,1 SAY'HOW LONG (DAYS)'+STR(no:day,2,0)
      @ 15,40 SAY'CHANGE TO ' GET t:no:day
      READ
      IF prog:date  t:date .OR. !(event)  !(t:act)
         USE B:trprog
         COUNT TO no:rec FOR prog:date = t:date .AND. ;
!(event) = !(t:act)
         IF no:rec  0
```

```
            @ 23,10 SAY'CANNOT MAKE CHANGES BECAUSE ;
THE RECORD HAS ALREADY EXISTED'
            DO delay
            LOOP
         ENDIF
      ENDIF
      @ 18,10 SAY'The record has been updated'
      REPLACE prog:date WITH t:date,event WITH t:act,loc ;
WITH t:loc,amt:due WITH t:due,no:day WITH t:no:day FOR ;
prog:date = pdate
      DO delay
   ENDIF
ENDDO
```

```
TRLIST.CMD
*
*
*
DO setup
USE B:trprog
COUNT TO no:rec
IF no:rec = 0
   @ 23,10 SAY'This file is empty'
   DO delay
   RETURN
ENDIF
STORE 'trprog' TO trfile
DO WHILE t
   STORE ' ' TO tr:sel
   ERASE
   @ 5,20 SAY'LISTING TROOP PROGRAM MENU'
   @ 5,60 SAY DATE()
   @ 8,15 SAY'0 - Exit'
   @ 10,15 SAY'1 - List all the troop program file'
   @ 12,15 SAY'2 - List troop program by month'
   @ 18,10 SAY'Enter your selection' GET tr:sel
   READ
   DO CASE
      CASE tr:sel ='0'
         RELEASE ALL
         RETURN
      CASE tr:sel ='1'
         STORE 'tllistl' TO tlist
         DO tlistl
      CASE tr:sel ='2'
         DO tlist2
      OTHERWISE
         @ 23,10 SAY'WRONG CODE - TRY AGAIN'
         DO delay
   ENDCASE
ENDDO
RETURN
```

```
TLIST1.CMD
*
*
DO setup
STORE ' ' TO rpt
ERASE
@                                                        5,10
SAY'****************************************************;
****'
@ 6,10 SAY'*                                            ;
          *'
@ 7,10 SAY'*           TROOP PROGRAM LISTING            ;
          *'
@ 8,10 SAY'*                                            ;
          *'
@ 9,10 SAY'*     TO HALT - Press CRTL and S together ;
        *'
@ 10,10 SAY'*     TO CONTINUE - Press any key             ;
         *'
@ 11,10 SAY'*                                             ;
         *'
@ 12,10 SAY'*                                             ;
         *'
@ 13,10 SAY'*     IF you are ready press RETURN to     ;
continue     *'
@ 14,10 SAY'*                                             ;
         *'
@ 15,10 SAY'*********************************************;
***********'
WAIT
STORE 22 TO line
STORE 'ERASE' TO npage
ERASE
DO tlist
IF row  22
   STORE 1 TO row
   ERASE
ENDIF
@ row,10 SAY 'This is the end of the file.'
STORE row+1 TO row
@ row,10 SAY'Do you want a report on this file ? (Y or ;
N)'  GET rpt
READ
IF !(rpt) = 'N'
   RETURN
ENDIF
npage
@ 5,10 SAY"*******************************************"
@ 6,10 SAY'*     PREPARE PRINTER TO PRINT OUT         *'
@ 7,10 SAY'*                                          *'
@ 8,10 SAY'*      PRESS RETURN WHEN READY          *'
@ 9,10 SAY'*******************************************'
WAIT
SET FORMAT TO PRINT
SET CONSOLE OFF
SET PRINT on
STORE 55 TO line
STORE 'EJECT' TO npage
DO tlist
SET FORMAT TO SCREEN
SET PRINT OFF
SET CONSOLE ON
@ 22,10 SAY'This is the end of the file.'
@ 23,10 SAY'Press any key to return to MENU.'
WAIT
RETURN
```

**T1LIST1.CMD**

```
*
*
*
STORE '          ' TO mon
STORE '  ' TO mont
STORE '  ' TO yr
USE B:trfile
INDEX ON $(prog:date,7,2)+$(prog:date,1,2)+;
$(prog:date,4,2)+!(event) TO B:trindx
GOTO TOP
STORE $(prog:date,7,2) TO yr
STORE $(prog:date,1,2) TO mont
DO WHILE (.NOT. EOF)
   DO moname
   @ 1,23 SAY'TROOP PROGRAM RECORD'
   @ 2,21 SAY'IN THE MONTH OF '+TRIM(!(mon))+', 19'+yr
   @ 4,6 SAY'DATE'
   @ 4,25 SAY'ACTIVITY'
   @ 4,57 SAY'LOCATION'
   STORE 6 TO row
   DO WHILE (.NOT. EOF) .AND. rowline .and. ;
$(prog:date,7,2) = yr .AND. $(prog:date,1,2) = mont
      @ row,7 SAY VAL($(prog:date,4,2)) USING '99'
      @ row,19 SAY TRIM(!(event))
      @ row,55 SAY TRIM(!(loc))
      STORE row+1 TO row
      DO delay
      SKIP
   ENDDO
   IF .NOT. EOF
      STORE $(prog:date,7,2) TO yr
      STORE $(prog:date,1,2) TO mont
   ENDIF
   npage
ENDDO
RETURN
```

```
TLIST2.CMD
*
*
*
DO setup
DO WHILE T
   ERASE
   STORE '           ' TO mon
   STORE '  ' TO mont
   STORE '  ' TO yr
   @ 5,10 SAY ' What month do you like to list? (:-01 to ;
12)' GET mont
   READ
   IF mont = '  '
      RELEASE ALL
      RETURN
   ENDIF
   @ 6,10 SAY'and what year? -19' GET yr
   READ
   USE B:trprog
   COUNT TO no:rec FOR $(prog:date,7,2) = yr .AND. ;
$(prog:date,1,2) = mont
   IF no:rec = 0
      @ 23,10 SAY'THERE IS NO RECORD ON THAT MONTH.'
      DO delay
      LOOP
   ENDIF
   DO moname
   STORE ' ' TO confirm
   @ 8,10 SAY'Please confirm that you want the list off;
 all programs'
   @ 9,10 SAY'in the month of '+TRIM(!(mon))+', 19'+yr+;
' ?(Y or N)' GET confirm
   READ
   IF !(confirm) ='N'
      LOOP
   ENDIF
   STORE 't2list2' TO tlist
   DO tlist1
   USE B:trprog
   DELETE FILE B:temp
   RETURN
ENDDO
```

T2LIST2.CMD

```
*
*
*
USE B:trprog
INDEX ON $(prog:date,7,2)+$(prog:date,1,2)+;
$(prog:date,4,2)+!(event) TO B:trindx
COPY TO B:temp FOR $(prog:date,7,2) = yr .AND. ;
$(prog:date,1,2) = mont
SELECT SECONDARY
USE B:temp
GOTO TOP
DO WHILE .NOT. EOF
   @ 1,23 SAY'TROOP PROGRAM RECORD'
   @ 1,60 SAY DATE()
   @ 2,21 SAY'IN THE MONTH OF '+TRIM(mon)+', 19'+yr
   @ 4,6 SAY 'DATE'
   @ 4,25 SAY'ACTIVITY'
   @ 4,57 SAY'LOCATION'
   STORE 6 TO row
   DO WHILE .NOT. EOF .AND. row  line
      @ row,7 SAY VAL($(prog:date,4,2)) USING '99'
      @ row,19 SAY TRIM(!(event))
      @ row,55 SAY TRIM(!(loc))
      STORE row+1 TO row
      DO delay
      SKIP
   ENDDO
   npage
ENDDO
RETURN
```

```
ADULT.CMD
*
*
*
DO setup
DO WHILE t
   STORE " " TO ad:sel
   ERASE
   @ 3,20 SAY'ADULT LEADERSHIP MENU'
   @ 3,60 SAY DATE()
   @ 6,15 SAY'0 - Exit'
   @ 8,15 SAY"1 - Add new adult leader's information"
   @ 10,15 SAY'2 - Delete unused record'
   @ 12,15 SAY'3 - Update record'
   @ 14,15 SAY'4 - List  report adult information'
   @ 19,10 SAY'Enter your selection' GET ad:sel
   READ
   DO CASE
      CASE ad:sel ='0'
         RELEASE ALL
         RETURN
      CASE ad:sel ='1'
         DO adadd
      CASE ad:sel ='2'
         DO addel
      CASE ad:sel ='3'
         DO adupd
      CASE ad:sel ='4'
         DO adlist
      OTHERWISE
         @ 23,10 SAY'WRONG CODE - TRY AGAIN'
         DO delay
   ENDCASE
ENDDO
RETURN
```

```
ADADD.CMD
*
*
*
DO setup
DO WHILE t
   STORE '               ' TO n:last
   STORE '               ' TO n:first
   STORE '                    ' TO n:title
   STORE '          ' TO n:tr:name
   STORE '                         ' TO n:addr
   STORE '               ' TO n:city
   STORE '        ' TO n:zip
   STORE '             ' TO n:hphone
   STORE '             ' TO n:bphone
   STORE ' ' TO n:sex
   ERASE
   @ 3,22 SAY 'NEW ADULT MEMBER'
   @ 3,60 SAY DATE()
   @ 5,10 SAY 'ENTER LAST NAME          ' GET n:last
   READ
   IF n:last = '               '
      RELEASE ALL
      RETURN
   ENDIF
   @ 6,16 SAY 'MIDDLE  FIRST NAME' GET n:first
   READ
   USE b:adlead
   COUNT TO no:rec FOR !(ad:last) = !(n:last) .AND. ;
!(ad:first) = !(n:first)
   IF no:rec  0
      @ 23,10 SAY 'The record of this person has been ;
 added already'
      DO delay
      LOOP
   ENDIF
   @ 7,16 SAY'TITLE              ' GET n:title
   @ 8,16 SAY'SEX                ' GET n:sex
   @ 9,16 SAY'TROOP NAME         ' GET n:tr:name
   @ 10,16 SAY'ADDRESS            ' GET n:addr
   @ 11,16 SAY'CITY               ' GET n:city
   @ 12,16 SAY'STATE  ZIP CODE   ' GET n:zip PICTURE   ;
'AA.99999'
   @ 13,16 SAY'HOME PHONE         ' GET n:hphone PICTURE ;
'(999)999-9999'
   @ 14,16 SAY'BUSINESS PHONE     ' GET n:bphone PICTURE ;
'(999)999-9999'
   READ
   APPEND BLANK
   REPLACE ad:last WITH n:last,ad:first WITH n:first,;
title WITH n:title,ad:sex WITH n:sex,ad:tr:name WITH ;
n:tr:name,ad:addr WITH n:addr,ad:city WITH n:city,;
ad:zip WITH n:zip,ad:hphone WITH n:hphone,ad:bphone ;
WITH n:bphone
ENDDO
```

```
ADDEL.CMD
*
*
*
DO setup
RESTORE FROM B:memfile
STORE 'T' TO first
STORE 'T' TO a
DO WHILE a = 'T'
   STORE '               ' TO t:last
   STORE '               ' TO t:first
   STORE '                    ' TO t:title
   STORE '          ' TO t:trname
   STORE ' ' TO t:sex
   STORE '                         ' TO t:addr
   STORE '               ' TO t:city
   STORE '        ' TO t:zip
   STORE '             ' TO t:hphone
   STORE '             ' TO t:bphone
   STORE '               ' TO l:name
   STORE '               ' TO f:name
   ERASE
   USE B:adlead
   COUNT TO no:rec
   IF no:rec = 0
      @ 23,10 SAY' This adult leadership file is empty'
      DO delay
      RELEASE ALL
      RETURN
   ENDIF
   @ 3,19 SAY"DELETE ADULT LEADERSHIP'S RECORD"
   @ 3,60 SAY DATE()
   @ 4,10 SAY'NOTE: PRESS RETURN TO STOP'
   @ 6,10 SAY'ENTER LAST NAME            ' GET l:name
   READ
   IF l:name = '               '
      IF first = 'T'
         RELEASE ALL
         RETURN
      ENDIF
      STORE 'F' TO a
      LOOP
   ENDIF
   @ 7,16 SAY'MIDDLE  FIRST NAME  ' GET f:name
   READ
   COUNT TO no:rec FOR !(ad:last) = !(l:name) .AND.
!(ad:first) = !(f:name)
   IF no:rec = 0
```

```
      @ 23,10 SAY 'There is no such name in this file'
      DO delay
      LOOP
   ENDIF
   IF no:rec  0
      LOCATE FOR !(ad:Last) = !(l:name) .AND. !(ad:first) ;
= !(f:name)
      STORE ad:last TO t:last
      STORE ad:first TO t:first
      STORE title TO t:title
      STORE ad:tr:name TO t:trname
      STORE ad:sex TO t:sex
      STORE ad:addr TO t:addr
      STORE ad:city TO t:city
      STORE ad:zip TO t:zip
      STORE ad:hphone TO t:hphone
      STORE ad:bphone TO t:bphone
      @ 8,16 SAY'TITLE                '+TRIM(!(title))
      @ 9,16 SAY'TROOP NAME           '+TRIM(!(ad:tr:name))
      @ 10,16 SAY'SEX                  '+!(ad:sex)
      @ 11,16 SAY'MAILING ADDRESS      '+TRIM(!(ad:addr))
      @ 12,16 SAY'CITY                 '+TRIM(!(ad:city))
      @ 13,16 SAY'STATE  ZIP CODE     '+!(ad:zip)
      @ 14,16 SAY'HOME PHONE           '+ad:hphone
      @ 15,16 SAY'BUSINESS PHONE       '+ad:bphone
      STORE ' ' TO decide
      @ 17,10 SAY'Is this the record you want to delete? ;
(Y OR N) :' GET decide
      READ
      IF !(decide) ='N'
         LOOP
      ENDIF
      USE B:adbakup
      APPEND BLANK
      REPLACE ad:last WITH t:last,ad:first WITH t:first,;
title WITH t:title,ad:sex WITH t:sex,ad:tr:name WITH ;
t:trname,ad:addr WITH t:addr,ad:city WITH t:city,ad:zip ;
WITH t:zip,ad:hphone WITH t:hphone,ad:bphone WITH ;
t:bphone
      USE B:adlead
      DELETE FOR !(ad:last) = !(l:name) .AND. !(ad:first) ;
=!(f:name)
      PACK
      @ 18,10 SAY'THE RECORD OF '+TRIM(!(l:name))+', '+;
TRIM(!(f:name))+' HAS BEEN DELETED.'
      STORE 'F' TO first
      DO delay
   ENDIF
ENDDO
ERASE
```

```
@                                                        5,10
SAY'****************************************************'
@                          6,10                          SAY'*
*'
@  7,10  SAY'*           PREPARE THE PRINTER READEY TO PRINT
*'
@  8,10  SAY'*                     THE  RECORD  OF  DELETION
*'
@                          9,10                          SAY'*
*'
@  10,10  SAY'*                WHEN   READY   PRESS   RETURN
*'
@                          11,10                         SAY'*
*'
@                                                       12,10
SAY'****************************************************'
WAIT
SET FORMAT TO PRINT
SET PRINT ON
SET CONSOLE OFF
USE B:adbakup
INDEX ON !(ad:tr:name)+!(ad:last)+!(ad:first) TO B:adindx
GOTO TOP
STORE 'EJECT' TO npage
STORE 55 TO line
STORE 'THE DELETED ADULT LEADER RECORDS' TO adtitle
ERASE
DO adllistl
SET FORMAT TO SCREEN
SET PRINT OFF
SET CONSOLE ON
ERASE
@ 10,15 SAY'THE REPORT HAS BEEN COMPLETELY FINISHED.'
DO delay
DELETE ALL
PACK
RELEASE ALL
RETURN
```

```
ADUPD.CMD
*
*
*
DO setup
USE B:adlead
COUNT TO no:rec
IF no:rec = 0
   @ 23,10 SAY'I am sorry - this file is empty now'
   DO delay
   RELEASE ALL
   RETURN
ENDIF
STORE 'UPDATE ADULT LEADERSHIP RECORD' TO adtitle
DO WHILE t
   ERASE
   STORE '               ' TO l:name
   STORE '               ' TO f:name
   STORE '               ' TO t:last
   STORE '               ' TO t:first
   STORE '                    ' TO t:title
   STORE ' ' TO t:sex
   STORE '          ' TO t:trname
   STORE '                         ' TO t:addr
   STORE '               ' TO t:city
   STORE '        ' TO t:zip
   STORE '             ' TO t:hphone
   STORE '             ' TO t:bphone
   @ 3,20 SAY adtitle
   @ 3,60 SAY DATE()
   @ 4,5 SAY'NOTE : To update - Enter new information in ;
the'
   @ 4,52 SAY' areas you wish to'
   @ 5,22 SAY'change, otherwise leave them blank'
   @ 6,12 SAY'To stop - Press RETURN at the last name'
   @ 9,10 SAY'ENTER LAST NAME ' GET l:name
   READ
   IF l:name = '               '
      RETURN
   ENDIF
   @ 10,16 SAY'FIRST NAME' GET f:name
   READ
   COUNT TO no:rec FOR !(ad:last) = !(l:name) .AND. ;
!(ad:first) =!(f:name)
   IF no:rec = 0
      @ 23,10 SAY'There is no such record'
      DO delay
      LOOP
   ENDIF
   LOCATE FOR !(ad:last) = !(l:name) .AND. !(ad:first) = ;
!(f:name)
   STORE ad:last TO t:last
   STORE ad:first TO t:first
   STORE title TO t:title
   STORE ad:sex TO t:sex
   STORE ad:tr:name TO t:trname
   STORE ad:addr TO t:addr
   STORE ad:city TO t:city
   STORE ad:zip TO t:zip
   STORE ad:hphone TO t:hphone
   STORE ad:bphone TO t:bphone
   ERASE
   @ 3,20 SAY adtitle
   @ 5,4 SAY'LAST NAME      '+!(ad:last)
   @ 5,42 SAY'CHANGE TO ' GET t:last
   @ 6,4 SAY'FIRST NAME     '+!(ad:first)
   @ 6,42 SAY'CHANGE TO ' GET t:first
   @ 7,4 SAY'TITLE          '+TRIM(!(title))
   @ 7,42 SAY'CHANGE TO ' GET t:title
   @ 8,4 SAY'SEX            '+!(ad:sex)
   @ 8,42 SAY'CHANGE TO ' GET t:sex
   @ 9,4 SAY'TROOP NAME     '+TRIM(!(ad:tr:name))
   @ 9,42 SAY'CHANGE TO ' GET t:trname
   @ 10,4 SAY'MAILING ADDRESS'
   @ 10,42 SAY'CHANGE TO '
   @ 11,19 SAY !(ad:addr)
   @ 11,52 GET t:addr
   @ 12,4 SAY'CITY           '+TRIM(!(ad:city))
   @ 12,42 SAY'CHANGE TO ' GET t:city
   @ 13,4 SAY'ZIPCODE        '+!(ad:zip)
   @ 13,42 SAY'CHANGE TO ' GET t:zip PICTURE 'AA.99999'
   @ 14,4 SAY'HOME PHONE     '+ad:hphone
   @ 14,42 SAY'CHANGE TO ' GET t:hphone PICTURE  ;
'(999)999-9999'
   @ 15,4 SAY'BUSINESS PHONE '+ad:bphone
   @ 15,42 SAY'CHANGE TO ' GET t:bphone PICTURE  ;
'(999)999-9999'
   READ
   IF !(ad:last)  !(t:last) .AND. !(ad:first)  ;
!(t:first)
      COUNT TO no:rec FOR !(ad:last)=!(t:last) .AND. ;
!(ad:first)=!(t:first)
      IF no:rec  0
         @ 22,10 SAY'CANNOT MAKE CHANGES BECAUSE THE ;
RECORD HAS ALREADY EXISTED'
         DO delay
         LOOP
      ENDIF
   ENDIF
STORE ' ' TO decide
   @ 20,10 SAY'Are these changes corrected? (Y OR N)' GET ;
decide
   READ
   IF !(decide) = 'N'
      LOOP
   ENDIF
   REPLACE ad:last WITH t:last,ad:first WITH t:first,;
title WITH t:title,ad:sex WITH t:sex,ad:tr:name WITH ;
t:trname,ad:addr WITH t:addr,ad:city WITH t:city,ad:zip ;
WITH t:zip,ad:hphone WITH t:hphone,ad:bphone WITH ;
t:bphone FOR !(ad:last) = !(l:name) .AND. !(ad:first) = ;
!(f:name)
   @ 21,10 SAY'This record has been updated'
   DO delay
ENDDO
```

```
adlist.cmd
*
*
*
DO setup
STORE 'LIST ADULT LEADER INFORMATION' TO adtitle
STORE ' ' TO rpt
USE B:adlead
COUNT TO no:rec
IF no:rec = 0
   ERASE
   @ 23,10 SAY' SORRY THIS FILE IS EMPTY.'
   DO delay
   RELEASE ALL
   RETURN
ENDIF
INDEX ON !(ad:tr:name)+!(ad:last)+!(ad:first)+!(title) TO ;
B:adindx
GOTO TOP
ERASE
@ 5,15 SAY'***********************************************'
@ 6,15 SAY'*       '+adtitle+'         *'
@ 7,15 SAY'*         IN ALPHABETICAL NAME ORDER          *'
@ 8,15 SAY'*                                             *'
@ 9,15 SAY'*   TO HALT - PRESS CTRL AND S TOGETHER   *'
@ 10,15 SAY'*   TO CONTINUE - PRESS ANY KEY               ;
*'
@ 11,15 SAY'*                                             ;
*'
@ 12,15 SAY'*                                             ;
*'
@ 13,15 SAY'*   WHEN YOU ARE READY PRESS RETURN         ;
*'
@ 14,15 SAY'***********************************************'
WAIT
STORE 23 TO line
STORE 'ERASE' TO npage
ERASE
DO adllistl
IF row  20
   ERASE
   STORE 1 TO row
ENDIF
STORE row + 1 TO row
@ row,10 SAY'This is the end of this file'
STORE row + 1 TO row
@ row,10 SAY'Do you want a report of this file? (Y OR N)';
GET rpt
```

```
READ
IF !(rpt) ='N'
   RETURN
ENDIF
ERASE
@ 5,10 SAY'********************************************'
@ 6,10 SAY'*     PREPARE THE PRINTER TO PRINT OUT     *'
@ 7,10 SAY'*                                          *'
@ 8,10 SAY'*     PRESS RETURN WHEN READY            *'
@ 9,10 SAY'********************************************'
WAIT
ERASE
STORE 55 TO line
STORE 'EJECT' TO npage
SELECT PRIMARY
GOTO TOP
SET FORMAT TO PRINT
SET CONSOLE OFF
SET PRINT ON
DO adllistl
SET FORMAT TO SCREEN
SET PRINT OFF
SET CONSOLE ON
@ 10,15 SAY'The report has already finished'
@ 11,15 SAY'Press any key to return to MENU'
WAIT
RETURN
```

```
AD1LIST1.CMD
*
*
*
STORE ad:tr:name TO trname
DO WHILE .NOT. EOF
   @ 1,20 SAY adtitle
   @ 1,60 SAY 'DATE'
   @ 1,65 SAY DATE()
   @ 2,22 SAY 'OF TROOP '+TRIM(!(ad:tr:name))
   @ 4,5  SAY 'NAME'
   @ 4,16 SAY'SEX'
   @ 4,24 SAY 'TITLE'
   @ 4,46 SAY 'ADDRESS'
   @ 4,67 SAY 'PHONE NO.'
   STORE 6 TO row
   DO WHILE (.NOT. EOF) .AND.!(ad:tr:name)=!(trname) .AND.;
 rowline
      @ row,1  SAY TRIM(!(ad:last))+','
      @ row,17 SAY !(ad:sex)
      @ row,19 SAY TRIM(!(title))
      @ row,40 SAY TRIM(!(ad:addr))
      IF ad:hphone  '(   )   -    '
         @ row,65 SAY 'H.'+ad:hphone
      ENDIF
      STORE row+1 TO row
      @ row,1 SAY TRIM(!(ad:first))
      @ row,40 SAY TRIM(!(ad:city))+', '+!(ad:zip)
      IF ad:bphone  '(   )   -    '
         @ row,65 SAY 'B.'+ad:bphone
      ENDIF
      DO delay
      STORE row+2 to row
      SKIP
   ENDDO
   DO delay
   npage
   STORE ad:tr:name TO trname
ENDDO
RETURN
```

```
SCOUT.CMD
*
*
*
DO setup
DO WHILE t
   STORE ' ' TO sc:sel
   ERASE
   @ 3,20 SAY"SCOUT'S INFORMATION MENU"
   @ 3,60 SAY DATE()
   @ 6,15 SAY'0 - Exit'
   @ 8,15 SAY'1 - Add new member'
   @ 10,15 SAY'2 - Delete dropped out scout'
   @ 12,15 SAY"3 - Update scout's record"
   @ 14,15 SAY"4 - List  report scout's information"
   @ 17,10 SAY'Enter your selection  ' GET sc:sel
   READ
   DO CASE
      CASE sc:sel ='0'
         RELEASE ALL
         RETURN
      CASE sc:sel ='1'
         DO scadd
      CASE sc:sel ='2'
         DO scdel
      CASE sc:sel ='3'
         DO scupd
      CASE sc:sel ='4'
         DO sclist
      OTHERWISE
         @ 23,10 SAY'WRONG CODE - TRY AGAIN'
         DO delay
   ENDCASE
ENDDO
RETURN
```

```
SCADD.CMD
*
*
*
DO setup
RESTORE FROM B:memfile
STORE 'ADD NEW MEMBER               ' TO sctitle
DO WHILE T
   STORE '  ' TO grade
   STORE '        ' TO b:date
   STORE '                         ' TO n:add
   STORE '               ' TO n:city
   STORE '        ' TO n:zip
   STORE '             ' TO n:hphone
   STORE '          ' TO church
   STORE '          ' TO office
   STORE '          ' TO n:pat
   STORE ' ' TO unif
   STORE ' ' TO subboy
   STORE ' ' TO n:rank
   STORE '                         ' TO n:gname
   STORE '          ' TO n:grel
   STORE '               ' TO l:name
   STORE '               ' TO f:name
   ERASE
   @ 2,22 SAY sctitle
   @ 4,1 SAY 'ENTER LAST NAME                           ;
 ' GET l:name
   READ
   IF l:name = '               '
      RELEASE ALL
      RETURN
   ENDIF
   @ 5,7 SAY 'FIRST NAME                            ' GET ;
f:name
   READ
   USE B:screcord
   COUNT TO no:rec FOR !(sc:last) = !(l:name) .AND. ;
!(sc:first) = !(f:name)
   IF no:rec  0
      @ 23,10 SAY'This record has already existed !!!'
      DO delay
      LOOP
   ENDIF
   @ 6,7 SAY 'GRADE OF CLASS AT SCHOOL (:-01,05,...)' GET ;
grade
   READ
   IF grade  '05'
```

```
      @ 20,6 SAY'Scout must have finished the fifth grade'
      @ 22,6 SAY"This person's grade is "+TRIM(grade)
      DO delay
      DO delay
      LOOP
   ENDIF
   @ 7,7 SAY'BIRTH DATE (AS MM/DD/YY)              'GET ;
b:date PICTURE '99/99/99'
   READ
   STORE b:date TO cdate
   DO ckdate
   IF state = '1'
      @ 23,10 SAY' INVALID DATE ERROR - TRY AGAIN'
      DO delay
      LOOP
   ENDIF
   STORE VAL($(pre:date,7,2)) TO p:yr
   STORE VAL($(pre:date,1,2)) TO p:mo
   STORE VAL($(b:date,7,2)) TO b:yr
   STORE VAL($(b:date,1,2)) TO b:mo
   STORE p:yr - b:yr TO age
   STORE 0 TO age:mo
   IF age = 10 .AND. p:mo  b:mo
      STORE p:mo - b:mo TO age:mo
   ENDIF
   IF age  10 .OR. (age = 10 .AND. age:mo  6) .OR. age ;
18
      @ 20,6 SAY'Scout must be at least 10 and a half ;
years '
      @ 21,6 SAY'old or older, but not yet 18'
      @ 22,6 SAY'This person is '+STR(age,2,0)+' years ;
old.'
      DO delay
      LOOP
   ENDIF
   @ 8,7 SAY'MAILING ADDRESS                       ' GET ;
n:add
   @ 9,15 SAY'CITY                          ' GET n:city
   @ 10,15 SAY'STATE AND ZIPCODE              ' GET n:zip ;
PICTURE 'AA.99999'
   @ 11,7 SAY'HOME PHONE                            ' GET ;
n:hphone PICTURE '(999)999-9999'
   @ 12,7 SAY'CHURCH OR RELIGION                    ' GET ;
church
   @ 13,7 SAY'OFFICE (IF HAVE)                      ' GET ;
office
   @ 14,7 SAY'PATROL NAME                           ' GET ;
n:pat
   STORE 'T' TO a
   DO WHILE a = 'T'
```

```
      STORE ' ' TO n:sta
      @ 15,7 SAY'STATUE (N,B,R,S,T)                     ' ;
GET n:sta
      READ
      DO CASE
         CASE !(n:sta) = 'N'
            @ 16,14 SAY'New scout never before registered ;
with BSA'
         CASE !(n:sta) = 'B'
            @ 16,14 SAY'New scout who was a cub scout ;
before'
         CASE !(n:sta) = 'R'
            @ 16,14 SAY'Registered scout '
         CASE !(n:sta) = 'S'
            @ 16,14 SAY'Scout who seperated and ;
registered '
         CASE !(n:sta) = 'T'
            @ 16,14 SAY'Scout who transferred to troop'
         OTHERWISE
            @ 23,10 SAY'WRONG CODE - TRY AGAIN'
            DO delay
            @ 23,10 SAY'                         '
            LOOP
      ENDCASE
      STORE 'F' TO a
   ENDDO
   STORE 'T' TO a
   DO WHILE a = 'T'
      @ 17,7 SAY'UNIFORM COMPLETED? (Y/N)               ' ;
GET unif
      READ
      IF !(unif)  'Y' .AND. !(unif) 'N'
         @ 23,10 SAY'WRONG CODE - TRY AGAIN'
         DO delay
         @ 23,10 SAY'                         '
         LOOP
      ENDIF
      STORE 'F' TO a
   ENDDO
   STORE 'T' TO a
   DO WHILE a = 'T'
      @ 18,7 SAY"SUBSCIBE BOY'S LIFE? (Y/N)             " ;
GET subboy
      READ
      IF !(subboy)  'Y' .AND. !(subboy)  'N'
         @ 23,10 SAY'WRONG CODE - TRY AGAIN'
         DO delay
         @ 23,10 SAY'                         '
         LOOP
      ENDIF
      STORE 'F' TO a
   ENDDO
   STORE 'T' TO a
   DO WHILE a = 'T'
      @ 19,7 SAY'RANK (N,T,2,1,S,L,E)                  ' ;
GET n:rank
      READ
      DO CASE
         CASE !(n:rank) = 'N'
            @ 20,14 SAY'UNRANK'
         CASE !(n:rank) = 'T'
            @ 20,14 SAY'TENDER FOOT AWARD'
         CASE !(n:rank) = '2'
            @ 20,14 SAY'SECOND CLASS AWARD'
         CASE !(n:rank) = '1'
            @ 20,14 SAY'FIRST CLASS AWARD'
         CASE !(n:rank) = 'S'
            @ 20,14 SAY'STAR AWARD'
         CASE !(n:rank) = 'L'
            @ 20,14 SAY'LIFE AWARD'
         CASE !(n:rank) = 'E'
            @ 20,14 SAY'EAGLE AWARD'
         OTHERWISE
            @ 23,10 SAY'WRONG CODE - TRY AGAIN'
            DO delay
            @ 23,10 SAY'                       '
            LOOP
      ENDCASE
      STORE 'F' TO a
   ENDDO
   @ 21,7 SAY"GUARDIAN'S NAME                       " GET ;
n:gname
   @ 22,7 SAY"GUARDIAN'S RELATION TO SCOUT          " GET ;
n:grel
   READ
   USE B:screcord
   APPEND BLANK
   REPLACE sc:last WITH l:name,sc:first WITH f:name,;
sc:birth WITH b:date,sc:addr WITH n:add,sc:city WITH;
n:city,sc:zip WITH n:zip,sc:hphone WITH n:hphone,;
sc:church WITH church,sc:office WITH office,patrolname ;
WITH n:pat,status WITH n:sta,uniform WITH unif,boy:life ;
WITH subboy,rank WITH n:rank,g:name WITH n:gname,g:relate ;
WITH n:grel
   USE B:guardian
   COUNT TO no:rec FOR !(g:name) = !(n:gname)
   IF no:rec = 0
      STORE '               ' TO n:goccup
      STORE '             ' TO n:gbphone
      ERASE
      @ 2,22 SAY sctitle
      @ 2,60 SAY'DATE'
      @ 2,65 SAY DATE()
      @ 4,16 SAY"FAMILY INFORMATION OF '"+TRIM(!(l:name));
+', '+TRIM(!(f:name))+"'"
      @ 6,16 SAY"GUARDIAN'S NAME       "+!(n:gname)
      @ 7,16 SAY"GUARDIAN'S OCCUPATION " GET n:goccup
      @ 9,16 SAY'BUSINESS PHONE        ' GET n:gbphone ;
PICTURE '(999)999-9999'
      READ
      APPEND BLANK
      REPLACE g:name WITH n:gname,g:occup WITH n:goccup,;
g:bphone WITH n:gbphone
   ENDIF
   ERASE
   @ 2,22 SAY sctitle
   @ 2,60 SAY'DATE'
   @ 2,65 SAY DATE()
   @ 4,16 SAY"MEMBERSHIP INFORMATION OF '"+TRIM(!(l:name));
+', '+TRIM(!(f:name))+"'"
   STORE 'BECAME SCOUT' TO memact
   STORE '        ' TO be:scout
   @ 6,16 SAY'WHEN BECAME A SCOUT (MM/DD/YY)?   ' GET ;
be:scout PICTURE '99/99/99'
   READ
   STORE be:scout TO actdate
   DO mbadd
   STORE 'T' TO a
   DO WHILE a = 'T'
      STORE ' ' TO becub
      @ 7,16 SAY 'BE A CUB SCOUT BEFORE? (Y OR N)   ' GET ;
becub
      READ
      DO CASE
         CASE !(becub) ='Y'
            STORE 'F' TO a
         CASE !(becub) = 'N'
            STORE 'F' TO a
         OTHERWISE
            @ 23,10 SAY'PLEASE ENTER ONLY [Y] OR [N] - TRY;
AGAIN'
            DO delay
            @ 23,10 SAY'                                  ;
       '
      ENDCASE
   ENDDO
   IF !(becub) = 'Y'
      STORE '        ' TO be:cub
      @ 8,16 SAY'BECAME A CUB SCOUT FROM (MM/DD/YY)' GET ;
be:cub PICTURE  '99/99/99'
```

```
      READ
      STORE '        ' TO to:cub
      @ 9,16 SAY'BE A CUB SCOUT UNTIL (MM/DD/YY)   ' GET ;
to:cub PICTURE '99/99/99'
      READ
      STORE 'BE CUB SCOUT FROM' TO memact
      STORE be:cub TO actdate
      DO mbadd
      STORE 'BE CUB SCOUT TO' TO memact
      STORE to:cub TO actdate
      DO mbadd
   ENDIF
   STORE ' ' TO awd
   @ 6,16 SAY'                                           ;
   '
   @ 7,16 SAY'                                           ;
   '
   @ 8,16 SAY'                                           ;
   '
   @ 9,16 SAY'                                           ;
   '
   @ 6,16 SAY'HAVE YOU EVER RECEIVED CHURCH AWARD? (Y OR ;
N)' GET awd
   READ
   IF !(awd) = 'Y'
      STORE '        ' TO awddate
      @ 7,16 SAY'WHAT DATE?' GET awddate PICTURE '99/99/99'
      READ
      STORE 'CHURCH AWARD' TO memact
      STORE awddate TO actdate
      DO mbadd
   ENDIF
   STORE ' ' TO awd
   @ 6,16 SAY'                                             ;
          '
   @ 7,16 SAY'                                             ;
          '
   @ 8,16 SAY'                                             ;
          '
   @ 6,16 SAY'WHAT ABOUT ORDER OF ARROW MEMBER? (Y/N)' GET;
 awd
   READ
   IF !(awd) = 'Y'
      STORE '        ' TO awddate
      @ 7,16 SAY'WHAT DATE ?' GET awddate PICTURE ;
'99/99/99'
      READ
      STORE 'ORDER - ARROW MEMBER' TO memact
      STORE awddate TO actdate
      DO mbadd
   ENDIF
   @ 6,16 SAY'                                              ;
           '
   @ 7,16 SAY'                                              ;
           '
   STORE ' ' TO more
   @ 6,10 SAY' Are there any more member activities to be ;
added? (Y/N)' GET more
   READ
   IF !(more) = 'Y'
      STORE 'T' TO a
      DO WHILE a = 'T'
         STORE '                    ' TO memact
         STORE '        ' TO actdate
         @ 7,16 SAY'MEMBER ACTIVITY ' GET memact
         READ
         IF memact = '                    '
            STORE 'F' TO a
            LOOP
         ENDIF
         @ 8,16 SAY'WHEN      ' GET actdate PICTURE ;
'99/99/99'
         READ
         USE B:member
         COUNT TO no:rec FOR !(sc:last) = !(l:name) .AND. ;
!(sc:first) = !(f:name) .AND. !(status) = !( n:sta) .AND. ;
!(member:act) = !(memact)
         IF no:rec  0
            @ 23,10 SAY'This record has already existed ;
!!!'
            DO delay
            LOOP
         ENDIF
         DO mbadd
      ENDDO
   ENDIF
   ERASE
   STORE ' ' TO decide
   @ 6,10 SAY'Have any leader acticities to be added? ;
(Y/N) ' GET decide
   READ
   IF !(decide) = 'Y'
      STORE 'T' TO a
      DO WHILE a = "T"
         ERASE
         @ 2,22 SAY sctitle
         @ 2,60 SAY 'DATE'
         @ 2,65 SAY DATE()
         @ 4,16 SAY"LEADERSHIP INFORMATION OF '"+;
TRIM(!(l:name))+", "+TRIM(!(f:name))+"'"
```

```
         STORE '                    ' TO le:title
         STORE '        ' TO ledate
         @ 7,10 SAY'LEADERSHIP TITLE   ' GET le:title
         READ
         IF le:title = '                    '
            STORE 'F' TO a
            LOOP
         ENDIF
         @ 9,10 SAY'WHEN - (MM/DD/YY) ' GET ledate ;
PICTURE '99/99/99'
         READ
         USE B:leader
         COUNT TO no:rec FOR !(sc:last) = !(l:name) .AND.;
 !(sc:first) = !(f:name) .AND. !(leader) = !(le:title) ;
.AND. le:date = ledate
         IF no:rec  0
            @ 23,10 SAY 'This record has already existed ;
!!!'
            DO delay
            LOOP
         ENDIF
         APPEND BLANK
         REPLACE sc:last WITH l:name,sc:first WITH f:name,;
leader WITH le:title,le:date WITH ledate
      ENDDO
   ENDIF
   ERASE
   STORE ' ' TO awd
   @ 6,10 SAY'HAVE RECEIVED ANY AWARDS? (Y/N)' GET awd
   READ
   IF !(awd) = 'Y'
      STORE 'T' TO a
      DO WHILE a = 'T'
         ERASE
         @ 2,22 SAY sctitle
         @ 2,60 SAY'DATE'
         @ 2,65 SAY DATE()
         @ 4,16 SAY"MERIT BADGES AND SKILL AWARD OF '"+;
TRIM(!(l:name))+', '+TRIM(!(f:name))+"'"
         STORE ' ' TO awtype
         STORE '                    ' TO awname
         STORE '        ' TO when
         @ 7,10 SAY'TYPE OF AWARD (B - MERIT BADGE, S - ;
SKILL AWARD) ' GET awtype
         READ
         IF awtype = ' '
            STORE 'F' TO a
            LOOP
         ENDIF
         @ 8,10 SAY'NAME OF THE AWARD ' GET awname
         READ
         USE B:award
         COUNT TO no:rec FOR !(sc:Last) = !(l:name) .AND.;
!(sc:first) = !( f:name) .AND. !(type) = !(awtype) .AND. ;
!(awd:name) = !(awname)
         IF no:rec  0
            @ 23,10 SAY' This record has already existed ;
!!!'
            DO delay
            LOOP
         ENDIF
         @ 9,10 SAY'WHEN RECEIVED - (MM/DD/YY) ' GET when ;
PICTURE '99/99/99'
         READ
         APPEND BLANK
         REPLACE sc:last WITH l:name,sc:first WITH f:name,;
type WITH awtype,awd:name WITH awname,awd:date WITH when
      ENDDO
   ENDIF
   IF !(n:rank)  'N'
      ERASE
      @ 2,22 SAY sctitle
      @ 2,60 SAY"DATE"
      @ 2,65 SAY DATE()
      @ 4,16 SAY'ADVANCEMENT INFORMATION OF '+;
TRIM(!(l:name))+', '+TRIM(!(f:name))
      IF !(n:rank) = 'T' .OR. !(n:rank) = '1' .OR. ;
!(n:rank) = '2' .OR. !(n:rank) = 'S' .OR. !(n:rank) = ;
'L' .OR. !(n:rank) = 'E'
         STORE 'T' TO t:rank
         STORE '        ' TO nr:date
         @ 6,10 SAY'WHEN - PASSED TENDERFOOT AWARD ;
(MM/DD/YY) ' GET nr:date PICTURE '99/99/99'
         READ
         DO advadd
      ENDIF
      IF !(n:rank) = '2' .OR. !(n:rank) = '1' .OR. ;
!(n:rank) = 'S' .OR. !(n:rank) = 'L' .OR. !(n:rank) = ;
'E'
         STORE '2' TO t:rank
         STORE '        ' TO nr:date
         @ 6,10 SAY'WHEN - PASSED THE SECOND CLASS ;
(MM/DD/YY) ' GET nr:date PICTURE '99/99/99'
         READ
         DO advadd
      ENDIF
      IF !(n:rank) = '1' .OR. !(n:rank)  = 'S' .OR. ;
!(n:rank) = 'L' .OR. !(n:rank) = 'E'
         STORE '1' TO t:rank
         STORE '        ' TO nr:date
         @ 6,10 SAY'WHEN - PASSED THE FIRST CLASS ;
(MM/DD/YY)  ' GET nr:date PICTURE '99/99/99'
         READ
         DO advadd
      ENDIF
      IF !(n:rank) = 'S' .OR. !(n:rank) = 'L' .OR. ;
!(n:rank) = 'E'
         STORE 'S' TO t:rank
         STORE '        ' TO nr:date
         @ 6,10 SAY'WHEN - PASSED STAR AWARD (MM/DD/YY) ;
      ' GET nr:date PICTURE '99/99/99'
         READ
         DO advadd
      ENDIF
      IF !(n:rank) = 'L' .OR. !(n:rank) = 'E'
         STORE 'L' TO t:rank
         STORE '        ' TO nr:date
         @ 6,10 SAY'WHEN - PASSED LIFE AWARD (MM/DD/YY) ;
     ' GET nr:date PICTURE '99/99/99'
         READ
         DO advadd
      ENDIF
      IF !(n:rank) = 'E'
         STORE 'E' TO t:rank
         STORE '        ' TO nr:date
         @ 6,10 SAY'WHEN - PASSED EAGLE AWARD (MM/DD/YY) ;
    ' GET nr:date PICTURE '99/99/99'
         READ
         DO advadd
      ENDIF
   ENDIF
   USE B:trprog
   COPY FIELD prog:date,event TO B:temp FOR ;
$(prog:date,7,2) = $(pre:date,7,2) .AND. $(prog:date,1,2);
= $(pre:date,1,2) .AND. $(prog:date,4,2) = ;
$(pre:date,4,2)
   SELECT PRIMARY
   USE B:temp
   GOTO TOP
   DO WHILE .NOT. EOF
      STORE event TO tevent
      STORE prog:date TO pgdate
      SELECT SECONDARY
      USE B:ACCOUNT
      APPEND BLANK
      REPLACE sc:last WITH l:name,sc:first WITH f:name,;
prog:date WITH pgdate,event WITH tevent,amt:paid WITH ;
0,balance WITH 0
      SELECT PRIMARY
      SKIP
   ENDDO
   USE B:screcord
   DELETE FILE B:temp
ENDDO
```

```
SCDEL.CMD
*
*
*
DO setup
RESTORE FROM B:memfile
STORE 'T' TO first
STORE 'T' TO aa
DO WHILE aa = 'T'
   ERASE
   STORE '        ' TO t:birth
   STORE '                         ' TO t:addr
   STORE '               ' TO t:city
   STORE '        ' TO t:zip
   STORE '             ' TO t:hphone
   STORE '          ' TO t:church
   STORE '          ' TO t:patrol
   STORE '          ' TO t:office
   STORE '                         ' TO gdname
   STORE '          ' TO gdrel
   STORE ' ' TO t:sta
   STORE ' ' TO t:rank
   STORE ' ' TO t:unif
   STORE ' 'TO tboylife
   STORE '               ' TO l:name
   STORE '               ' TO f:name
   STORE ' ' TO decide
   USE B:screcord
   COUNT TO no:rec
   IF no:rec = 0
      @ 23,10 SAY'I AM SORRY, THIS FILE IS EMPTY NOW.'
      DO delay
      IF first = 'T'
         RELEASE ALL
         RETURN
      ENDIF
      STORE 'F' TO aa
      LOOP
   ENDIF
   @ 1,60 SAY DATE()
   @ 2,19 SAY'DELETE RECORDS OF DROPPED OUT SCOUT'
   @ 4,10 SAY'ENTER LAST NAME          ' GET l:name
   READ
   IF l:name ='               '
      IF first = 'T'
         RELEASE ALL
         RETURN
      ENDIF
      STORE 'F' TO aa
      LOOP
   ENDIF
   @ 5,16 SAY'MIDDLE  FIRST NAME' GET f:name
   READ
   COUNT TO no:rec FOR !(sc:last) = !(l:name) .AND. ;
!(sc:first) = !(f:name)
   IF no:rec = 0
      @ 23,10 SAY'There is no such record'
      DO delay
      LOOP
   ENDIF
   LOCATE FOR !(sc:last) = !(l:name) .AND. !(sc:first) = ;
 (f:name)
   @ 6,16 SAY'BIRTH DATE           '+sc:birth
   @ 7,16 SAY'ADDRESS              '+TRIM(!(sc:addr))+', '+;
TRIM(!(sc:city))+', '+!(sc:zip)
   @ 8,16 SAY'HOME PHONE           '+sc:hphone
   @ 9,16 SAY'CHURCH               '+!(sc:church)
   @ 10,16 SAY'PATROL NAME          '+!(patrolname)
   @ 11,16 SAY"GUARDIAN'S NAME      "+!(g:name)
   @ 12,16 SAY"GUARDIAN'S RELATE    "+!(g:relate)
   @ 13,16 SAY'PRESENT STATUS'
   DO CASE
      CASE !(status) = 'N'
         @ 13,36 SAY'N - NEW SCOUT WHO NEVER REGISTERED ;
BEFORE'
      CASE !(status) = 'B'
         @ 13,36 SAY'B - NEW SCOUT WHO WAS A CUB SCOUT'
      CASE !(status) = 'R'
         @ 13,36 SAY'R - REGISTERED SCOUT'
      CASE !(status) = 'S'
         @ 13,36 SAY'S - SEPARATED SCOUT AND REGISTERED'
      CASE !(status) = 'T'
         @ 13,36 SAY'T - TRANSFERRED TO TROOP'
   ENDCASE
   @ 14,16 SAY'PRESENT RANK'
   DO CASE
      CASE !(rank) = 'N'
         @ 14,36 SAY'UNRANK'
      CASE !(rank) = 'T'
         @ 14,36 SAY'TENDERFOOT'
      CASE !(rank) = '2'
         @ 14,36 SAY'SECOND CLASS'
      CASE !(rank) = '1'
         @ 14,36 SAY'FIRST CLASS'
      CASE !(rank) = 'S'
         @ 14,36 SAY'STAR'
      CASE !(rank) = 'L'
         @ 14,36 SAY'LIFE'
      CASE !(rank) = 'E'
         @ 14,36 SAY'EAGLE'
   ENDCASE
   STORE ' ' TO decide
      @ 16,10 SAY'Is this the record you want to delete? ;
(Y OR N)' GET decide
      READ
      IF !(decide) = 'N'
         LOOP
      ENDIF
      STORE sc:birth TO t:birth
      STORE sc:addr TO t:addr
      STORE sc:city TO t:city
      STORE sc:zip TO t:zip
      STORE sc:hphone TO t:hphone
      STORE sc:church TO t:church
      STORE sc:office TO t:office
      STORE patrolname TO t:patrol
      STORE status TO t:sta
      STORE rank TO t:rank
      STORE uniform TO t:unif
      STORE boy:Life TO tboylife
      STORE g:name TO gdname
      STORE g:relate TO gdrel
      USE B:scbakup
      APPEND BLANK
      REPLACE sc:last WITH l:name,sc:first WITH f:name,;
sc:birth WITH t:birth,sc:addr WITH t:addr,sc:city WITH ;
t:city,sc:zip WITH t:zip,sc:hphone WITH t:hphone,;
sc:church WITH t:church,sc:office WITH t:office,;
patrolname WITH t:patrol,status WITH t:sta,uniform WITH ;
t:unif,boy:life WITH tboylife,rank WITH t:rank,g:name ;
WITH gdname,g:relate WITH gdrel
      STORE 'F' TO first
      USE B:screcord
      DELETE FOR !(sc:last) = !(l:name) .AND. !(sc:first) ;
= !(f:name)
      PACK
      COUNT TO no:rec FOR !(g:name) =!(gdname)
      STORE '               ' TO tgoccup
      STORE '             ' TO tgbphone
      USE B:guardian
      LOCATE FOR !(g:name) = !(gdname)
      STORE g:occup TO tgoccup
      STORE g:bphone TO tgbphone
      USE B:guardbak
      APPEND BLANK
      REPLACE g:name WITH gdname,g:occup WITH tgoccup,;
g:bphone WITH tgbphone
      IF no:rec = 0
```

```
         USE B:guardian
         DELETE FOR !(g:name) = !(gdname)
         PACK
      ENDIF
      USE B:member
      COUNT TO no:rec FOR !(sc:last) =!(l:name) .AND. ;
!(sc:first) = !(f:name)
      IF no:rec  0
         COPY TO B:temp FOR !(sc:last) = !(l:name) .AND. ;
!(sc:first) = !(f:name)
         USE B:membakup
         APPEND FROM B:temp
         USE B:member
         DELETE ALL FOR !(sc:last) = !(l:name) .AND. ;
!(sc:first) = !(f:name)
         PACK
         DELETE FILE B:temp
      ENDIF
      USE B:leader
      COUNT TO no:rec FOR !(sc:last) = !(l:name) .AND. ;
!(sc:first) = !(f:name)
      IF no:rec  0
         COPY TO B:temp FOR !(sc:Last) = !(l:name) .AND. ;
!(sc:first) = !(f:name)
         USE B:leadbak
         APPEND FROM B:temp
         USE B:leader
         DELETE ALL FOR !(sc:last) = !(l:name) .AND. ;
!(sc:first) = !(f:name)
         PACK
         DELETE FILE B:temp
      ENDIF
      USE B:award
      COUNT TO no:rec FOR !(sc:last) = !(l:name) .AND. ;
!(sc:first) = !(f:name)
      IF no:rec  0
         COPY TO B:temp FOR !(sc:Last) = !(l:name) .AND. ;
!(sc:first) = !(f:name)
         USE B:awdbak
         APPEND FROM B:temp
         USE B:award
         DELETE ALL FOR !(sc:last) = !(l:name) .AND. ;
!(sc:first) = !(f:name)
         PACK
         DELETE FILE B:temp
      ENDIF
      USE B:account
      COUNT TO no:rec FOR !(sc:last) = !(l:name) .AND. ;
!(sc:first) = !(f:name)
      IF no:rec  0
```

```
         COPY TO B:temp FOR !(sc:Last) = !(l:name) .AND. ;
!(sc:first) = !(f:name)
         USE B:accbak
         APPEND FROM B:temp
         USE B:account
         DELETE ALL FOR !(sc:last) = !(l:name) .AND. ;
!(sc:first) = !(f:name)
         PACK
         DELETE FILE B:temp
      ENDIF
      USE B:advance
      COUNT TO no:rec FOR !(sc:last) = !(l:name) .AND. ;
!(sc:first) = !(f:name)
      IF no:rec  0
         COPY TO B:temp FOR !(sc:Last) = !(l:name) .AND.;
!(sc:first) = !(f:name)
         USE B:advbak
         APPEND FROM B:temp
         USE B:advance
         DELETE ALL FOR !(sc:last) = !(l:name) .AND. ;
!(sc:first) = !(f:name)
         PACK
         DELETE FILE B:temp
      ENDIF
   ENDIF
   @ 18,10 SAY' THE RECORD OF '+TRIM(!(l:name))+', '+;
TRIM(!(f:name))+' HAS BEEN DELETED.'
   DO delay
   STORE 'F' TO aa
ENDDO
ERASE
@ 5,10 SAY '**********************************************;
**********'
@ 6,10 SAY'*                                              ;
         *'
@ 7,10 SAY'*      PREPARE THE PRINTER READEY TO PRINT     ;
         *'
@ 8,10 SAY'*      THE RECORD OF DELETED SCOUT             ;
         *'
@ 9,10 SAY'*                                              ;
         *'
@ 10,10 SAY'*        WHEN YOU ARE READY PRESS RETURN  ;
         *'
@ 11,10 SAY'*                                             ;
         *'
@ 12,10 SAY'******************************************;
**************'
WAIT
SET FORMAT TO PRINT
SET PRINT ON
SET CONSOLE OFF
STORE 'EJECT' TO npage
STORE 55 TO line
ERASE
STORE "SCOUT'S PERSONAL IMFORMATION" TO stitle
STORE 'scbakup' TO scfile
DO s1all
STORE 'guardbak' TO gfile
STORE "SCOUT'S FAMILY DATA" TO stitle
DO s2all
STORE "SCOUT'S MEMBERSHIP RECORD" TO stitle
STORE 'membakup' TO mbfile
DO s3all
USE B:leadbak
COUNT TO no:rec
IF no:rec  0
   STORE "SCOUT'S LEADERSHIP RECORD" TO stitle
   STORE 'leadbak' TO lefile
   DO s4all
ENDIF
USE B:awdbak
COUNT TO no:rec
IF no:rec  0
   STORE 'MERIT BADGE AND SKILL AWARDS' TO stitle
   STORE 'awdbak' TO awfile
   DO s5all
ENDIF
STORE 'ADVANCEMENT' TO stitle
STORE 'advbak' TO advfile
DO s6all
USE B:accbak
COUNT TO no:rec
IF no:rec  0
   STORE 'ATTENDANCE AND DUES' TO stitle
   STORE 'accbak' TO accfile
   STORE 'trprog' TO trfile
   DO s7all
   STORE 'OUR CAMPING AND ACTIVITIES LOG' TO stitle
   DO s8all
ENDIF
SET FORMAT TO SCREEN
SET PRINT OFF
SET CONSOLE ON
USE B:scbakup
DELETE ALL
PACK
USE B:guardbak
DELETE ALL
PACK
USE B:membakup
```

```
DELETE ALL
PACK
USE B:lebakup
DELETE ALL
PACK
USE B:awdbak
DELETE ALL
PACK
USE B:advbak
DELETE ALL
PACK
USE B:accbak
DELETE ALL
PACK
RELEASE ALL
RETURN
```

```
SCUPD.CMD
*
*
*
DO setup
USE B:screcord
COUNT TO no:rec
IF no:rec = 0
   @ 23,10 SAY' THIS FILE IS EMPTY NOW'
   DO delay
   RETURN
ENDIF
STORE "UPDATE SCOUT'S RECORD" TO sctitle
STORE 'T' TO a
DO WHILE a = 'T'
   ERASE
   STORE '               ' TO l:name
   STORE '               ' TO f:name
   @ 1,22 SAY sctitle
   @ 3,1 SAY 'ENTER LAST NAME            ' GET l:name
   READ
   IF l:name = '               '
      RELEASE ALL
      RETURN
   ENDIF
   @ 4,7 SAY 'FIRST NAME          ' GET f:name
   READ
   USE B:screcord
   COUNT TO no:rec FOR !(sc:last) = !(l:name) .AND. ;
!(sc:first) = !(f:name)
   IF no:rec = 0
      @ 23,10 SAY 'THIS RECORD DOES NOT EXIST'
      DO delay
      LOOP
   ENDIF
   LOCATE FOR !(sc:Last) = !(l:name) .AND. !(sc:first) = ;
!(f:name)
   @ 5,7 SAY 'BIRTH DATE          '+sc:birth
   @ 6,7 SAY 'MAILING ADDRESS     '+TRIM(!(sc:addr))+', ';
+TRIM(!(sc:city))+', '+!(sc:zip)
   @ 7,7 SAY 'HOME PHONE          '+sc:hphone
   @ 8,7 SAY 'CHURCH OR RELIGION  '+!(sc:church)
   @ 9,7 SAY 'OFFICE              '+!(sc:office)
   @ 10,7 SAY 'PATROLNAME          '+!(patrolname)
   @ 11,7 SAY 'STATUS              '+!(status)
   DO CASE
      CASE !(uniform) = 'Y'
         @ 12,7 SAY 'UNIFORM             COMPLETED'
      OTHERWISE
         @ 12,7 SAY 'UNIFORM             NOT COMPLETED'
   ENDCASE
   DO CASE
      CASE !(boy:life) = 'Y'
         @ 13,7 SAY "BOY'S LIFE MAGAZINE SUBSCRIBED"
      OTHERWISE
         @ 13,7 SAY "BOY'S LIFE MAGAZINE NOT SUBCRIBED"
   ENDCASE
   @ 14,7 SAY 'RANK                '+!(rank)
   @ 15,7 SAY "GUARDIAN'S NAME    "+!(g:name)
   @ 16,7 SAY "RELATION TO SCOUT  "+!(g:relate)
   STORE ' ' TO decide
   @ 18,10 SAY'Is this the record you like to update? (Y ;
or N)' GET decide
   READ
   IF !(decide) = 'N'
      LOOP
   ENDIF
   STORE 'F' TO a
ENDDO
DO WHILE T
   ERASE
   STORE ' ' TO select
   @ 1,22 SAY sctitle
   @ 3,10 SAY"0 - STOP AND RETURN TO SCOUT'S MENU"
   @ 4,10 SAY"1 - UPDATE SCOUT'S PERSONAL INFORMATION"
   @ 5,10 SAY"2 - UPDATE GUARDIAN'S RECORD"
   @ 6,10 SAY"3 - UPDATE MEMBERSHIP RECORD"
   @ 7,10 SAY"4 - UPDATE LEADERSHIP RECORD"
   @ 8,10 SAY'5 - UPDATE AWARD (MERIT BADGES OR SKILL ;
AWARD)'
   @ 9,10 SAY"6 - UPDATE SCOUT'S RANK"
   @ 10,10 SAY"7 - UPDATE SCOUT'S ACCOUNT"
   @ 12,15 SAY'ENTER YOUR SELECTION' GET select
   READ
   DO CASE
      CASE select = '0'
         RELEASE ALL
         RETURN
      CASE select = '1'
         DO updsc
      CASE select = '2'
         DO updgud
      CASE select = '3'
         DO updmem
      CASE select = '4'
         DO updled
      CASE select = '5'
         DO updawd
      CASE select = '6'
         DO updrk
      CASE select = '7'
         DO updacc
      OTHERWISE
         @ 23,10 SAY'WRONG CODE - TRY AGAIN'
         DO delay
         @ 23,10 SAY'                       '
   ENDCASE
ENDDO
```

```
UPDSC.CMD
*
*
*
ERASE
@ 1,22 SAY"UPDATE SCOUT'S PERSONAL INFORMATION"
USE B:screcord
LOCATE  FOR  !(sc:last)  =  !(l:name)  .AND.  !(sc:first)  =
!(f:name)
STORE sc:last TO t:last
STORE sc:first TO t:first
STORE sc:birth TO t:birth
STORE sc:birth TO s:birth
STORE sc:addr TO t:addr
STORE sc:city TO t:city
STORE sc:zip TO t:zip
STORE sc:hphone TO t:hphone
STORE sc:church TO t:church
STORE sc:office TO t:office
STORE patrolname TO t:pat
STORE uniform TO t:unif
STORE status TO t:sta
STORE boy:life TO t:boy
STORE rank TO o:rank
STORE rank TO t:rank
STORE g:name TO tg:name
STORE g:name TO tgname
STORE g:relate TO tg:rel
@ 3,1 SAY'LAST NAME           '+!(sc:last)
@ 3,43 SAY'CHANGE TO ' GET t:last
@ 4,1 SAY'FIRST NAME          '+!(sc:first)
@ 4,43 SAY'CHANGE TO ' GET t:first
READ
IF !(sc:last)  !(t:last) .OR. !(sc:first)  !(t:first)
   COUNT TO no:rec FOR !(sc:last) = !(t:last) .AND. ;
!(sc:first) = !(t:first)
   IF no:rec  0
      @ 23,10 SAY'THIS RECORD HAS ALREADY EXISTED'
      DO delay
      @ 23,10 SAY'                                '
      LOOP
   ENDIF
ENDIF
STORE 'T' TO a
DO WHILE a = 'T'
   STORE s:birth TO t:birth
   @ 5,1 SAY'BIRTH DATE          '+s:birth
   @ 5,43 SAY'CHANGE TO ' GET t:birth PICTURE '99/99/99'
   READ
   STORE t:birth TO cdate
   DO ckdate
   IF state = '1'
      @ 23,10 SAY'INVALID DATE - TRY AGAIN'
      DO delay
      @ 23,10 SAY'                           '
      LOOP
   ENDIF
   STORE 'F' TO a
ENDDO
@ 6,1 SAY'MAILING ADDRESS     '+!(sc:addr)
@ 6,43 SAY'CHANGE TO ' GET t:addr
@ 7,1 SAY'CITY                '+!(sc:city)
@ 7,43 SAY'CHANGE TO ' GET t:city
@ 8,1 SAY'STATE AND ZIP       '+!(sc:zip)
@ 8,43 SAY'CHANGE TO ' GET t:zip
@ 9,1 SAY'HOME PHONE          '+sc:hphone
@ 9,43 SAY'CHANGE TO ' GET t:hphone PICTURE '(999)999-9999'
@ 10,1 SAY'OFFICE              '+!(sc:office)
@ 10,43 SAY'CHANGE TO ' GET t:office
@ 11,1 SAY'CHURCH              '+!(sc:church)
@ 11,43 SAY'CHANGE TO ' GET t:church
@ 12,1 SAY'PATROLNAME          '+!(patrolname)
@ 12,43 SAY'CHANGE TO ' GET t:pat
STORE 'T' TO a
DO WHILE a = 'T'
   @ 13,1 SAY'STATUS              '+!(status)
   @ 13,43 SAY'CHANGE TO ' GET t:sta
   READ
   DO CASE
      CASE !(t:sta) = 'N'
         STORE 'F' TO a
      CASE !(t:sta) = 'B'
         STORE 'F' TO a
      CASE !(t:sta) = 'R'
         STORE 'F' TO a
      CASE !(t:sta) = 'S'
         STORE 'F' TO a
      CASE !(t:sta) = 'T'
         STORE 'F' TO a
      OTHERWISE
         @ 23,10 SAY'INVALID CODE - TRY AGAIN'
         DO delay
         @ 23,10 SAY'                          '
   ENDCASE
ENDDO
@ 14,1 SAY'UNIFORM (Y OR N)    '+!(uniform)
@ 14,43 SAY'CHANGE TO ' GET t:unif
@ 15,1 SAY"BOY'S LIFE (Y OR N) "+!(boy:life)
@ 15,43 SAY'CHANGE TO ' GET t:boy
READ
STORE 'T' TO a
DO WHILE a = "T"
   STORE o:rank TO t:rank
   @ 16,1 SAY'RANK                '+!(o:rank)
   @ 16,43 SAY'CHANGE TO ' GET t:rank
   READ
   DO CASE
      CASE !(t:rank) = 'N'
         STORE 'F' TO a
      CASE !(t:rank) = 'T'
         STORE 'F' TO a
      CASE !(t:rank) = '2'
         STORE 'F' TO a
      CASE !(t:rank) = '1'
         STORE 'F' TO a
      CASE !(t:rank) = 'S'
         STORE 'F' TO a
      CASE !(t:rank) = 'L'
         STORE 'F' TO a
      CASE !(t:rank) = 'E'
         STORE 'F' TO a
      OTHERWISE
         @ 23,10 SAY'INVALID CODE - TRY AGAIN'
         DO delay
         @ 23,10 SAY'                         '
   ENDCASE
   IF !(o:rank)  !(t:rank)
      DO CASE
         CASE !(o:rank) = 'N'
            STORE 'T' TO n:rank
         CASE !(o:rank) = 'T'
            STORE '2' TO n:rank
         CASE !(o:rank) = '2'
            STORE '1' TO n:rank
         CASE !(o:rank) = '1'
            STORE 'S' TO n:rank
         CASE !(o:rank) = 'S'
            STORE 'L' TO n:rank
         CASE !(o:rank) = 'L'
            STORE 'E' TO n:rank
      ENDCASE
      USE B:advance
      COUNT TO no:rec FOR !(o:rank) = !(t:rank) .AND. ;
!(sc:last) = !(l:name) .AND. !(sc:first) = !(f:name)
      IF no:rec  0
         @ 23,10 SAY'THE SCOUT HAS ALREADY PASSED THIS ;
RANK'
         DO delay
         @ 23,10 SAY'                                     ;
  '
         LOOP
      ENDIF
      IF !(n:rank)  !(t:rank)
         STORE ' ' TO change
         @ 22,10 SAY'THE NEW RANK IS SUPPOSED TO BE'+;
!(n:rank)
         @ 23,10 SAY"DO YOU STILL WANT TO CHANGE TO RANK-";
+!(t:rank)+" ? ( Y or N)" GET change
         READ
         IF change = 'N'
            DO delay
            @ 22,10 SAY'                                  ;
                         '
            @                   23,10                   SAY'
'
            LOOP
         ENDIF
      ENDIF
      STORE '             ' TO t:rkdate
      @ 17,1 SAY'DATE PASSED THIS RANK' GET t:rkdate ;
PICTURE '99/99/99'
      READ
      STORE t:rkdate TO cdate
      DO ckdate
      IF state = '1'
         @ 23,10 SAY'INVALID DATE - TRY AGAIN'
         DO delay
         @ 23,10 SAY'                        '
         LOOP
      ENDIF
   ENDIF
ENDDO
@ 18,1 SAY"GUARDIAN'S NAME    "+!(tg:name)
@ 18,43 SAY'CHANGE TO ' GET tg:name
@ 19,1 SAY"RELATION TO SCOUT  "+!(tg:rel)
@ 19,43 SAY'CHANGE TO ' GET tg:rel
READ
STORE ' ' TO correct
@ 20,10 SAY 'Are those changes corrected? (Y or N)' GET ;
correct
READ
IF !(correct) ='N'
   RETURN
ENDIF
USE B:screcord
REPLACE sc:last WITH t:last,sc:first WITH t:first,sc:birth;
WITH t:birth,sc:addr WITH t:addr,sc:city WITH t:city,;
sc:zip WITH t:zip,sc:hphone WITH t:hphone,sc:church WITH;
```

```
t:church,sc:office WITH t:office,patrolname WITH t:pat,;
status WITH t:sta,uniform WITH t:unif,boy:life WITH t:boy,;
rank WITH t:rank,g:name WITH tg:name,g:relate WITH tg:rel ;
FOR !(sc:last) = !(l:name) .AND. !(sc:first) = !(f:name)
*
IF !(tg:name)  !(tgname)
   USE B:guardian
   REPLACE g:name WITH tg:name FOR !(g:name) = !(tgname)
ENDIF
*
USE B:member
REPLACE sc:last WITH t:last,sc:first WITH t:first,status ;
WITH t:sta FOR !(sc:last) = !(l:name) .AND. !(sc:first) ;
=!(f:name)
*
USE B:ADVANCE
IF !(o:rank)  !(t:rank)
   APPEND BLANK
   REPLACE sc:last WITH t:last,sc:first WITH t:first,rank ;
WITH t:rank, rank:date WITH t:rkdate
ENDIF
REPLACE sc:last WITH t:last,sc:first WITH t:first FOR ;
!(sc:last) = !(l:name) .AND. !(sc:first) = !(f:name)
*
USE B:leader
REPLACE sc:last WITH t:last,sc:first WITH t:first FOR ;
!(sc:last) = !(l:name) .AND. !(sc:first) = !(f:name)
*
USE B:award
REPLACE sc:last WITH t:last,sc:first WITH t:first FOR ;
!(sc:last) = !(l:name) .AND. !(sc:first) = !(f:name)
*
USE B:account
REPLACE sc:last WITH t:last,sc:first WITH t:first FOR ;
!(sc:last) = !(l:name) .AND. !(sc:first) = !(f:name)
STORE t:last TO l:name
STORE t:first TO f:name
RETURN
```

```
UPDGUD.CMD
*
*
*
ERASE
USE B:screcord
LOCATE FOR !(sc:last) = !(l:name) .AND. !(sc:first) = ;
!(f:name)
STORE g:name TO tgname
USE B:guardian
LOCATE FOR !(g:name) = !(tgname)
STORE g:occup TO tgoccup
STORE g:bphone TO tgbphone
@ 1,22 SAY"UPDATE GUARDIAN'S RECORD"
@ 3,3 SAY"GUARDIAN'S NAME "+!(g:name)
@ 4,3 SAY'OCCUPATION      '+!(g:occup)
@ 4,46 SAY'CHANGE TO ' GET tgoccup
@ 5,3 SAY'BUSINESS PHONE  '+g:bphone
@ 5,46 SAY'CHANGE TO ' GET tgbphone PICTURE '(999)999-9999'
READ
STORE ' ' TO change
@ 7,10 SAY'Are these changes corrected? (Y or N)' GET ;
change
READ
IF !(change) = 'N'
   RETURN
ENDIF
REPLACE g:occup WITH tgoccup,g:bphone WITH tgbphone FOR ;
!(g:name) = !(tgname)
@ 10,10 SAY'The record has been updated'
@ 11,10 SAY'Press RETURN to continue'
WAIT
RETURN
```

```
UPDMEM.CMD
*
*
*
DO WHILE T
   ERASE
   @ 1,22 SAY'UPDATE MEMBERSHIP RECORD'
   @ 3,10 SAY'Enter blank to stop and return to menu'
   STORE '                    ' TO u:member
   @ 5,3 SAY'MEMBERSHIP ACTIVITY' GET u:member
   READ
   IF u:member = '                    '
      RETURN
   ENDIF
   USE B:screcord
   LOCATE FOR !(sc:last) = !(l:name) .AND. !(sc:first) = ;
!(f:name)
   STORE status TO t:sta
   USE B:member
   COUNT TO no:rec FOR !(sc:last) = !(l:name) .AND. ;
!(sc:first) = !(f:name) .AND. !(member:act) = !(u:member) ;
.AND. !(status) = !(t:sta)
   IF no:rec  0
      LOCATE FOR !(sc:last) = !(l:name) .AND. !(sc:first) ;
= !(f:name) .AND. !(u:member) = !(member:act) .AND. ;
!(status) = !(t:sta)
      STORE member:act to t:member
      STORE date:act TO u:memdate
      STORE date:act TO t:memdate
      @ 5,46 SAY'CHANGE TO ' GET u:member
      @ 6,3 SAY'DATE                              '+t:memdate
      @ 6,46 SAY'CHANGE TO ' GET u:memdate PICTURE  ;
'99/99/99'
      READ
      IF !(t:member)!(u:member)
         COUNT TO numrec FOR !(sc:last)=!(l:name) .AND. ;
!(sc:first) = !(f:name) .AND. !(member:act)=!(u:member)
         IF numrec  0
            @ 22,10 SAY'THE CHANGE CANNOT BE MADE '
            @ 23,10 SAY'BECAUSE THAT MEMBERSHIP ;
ACTIVITY HAS ALREADY EXISTED'
            DO delay
            LOOP
         ENDIF
      ENDIF
      STORE ' ' TO change
      @ 8,10 SAY'Are these changes corrected? (Y or N)' ;
GET change
```

```
      READ
      IF !(change) = 'N'
         LOOP
      ENDIF
      REPLACE member:act WITH u:member,date:act WITH ;
u:memdate FOR !(sc:last) = !(l:name) .AND. !(sc:first) ;
= !(f:name) .AND. !(member:act) = !(t:member) .AND.;
date:act = t:memdate
   ENDIF
   IF no:rec = 0
      STORE ' ' TO add
      @ 7,10 SAY'THE RECORD DOES NOT EXIST '
      @ 9,10 SAY'DO YOU WANT TO ADD THIS RECORD? (Y OR N)';
GET add
      READ
      IF !(add) = 'N'
         LOOP
      ENDIF
      STORE 'T' TO a
      DO WHILE a = 'T'
         STORE '             ' TO u:memdate
         @ 10,10 SAY'WHEN DOES THIS ACTIVITY OCCUR?' GET ;
u:memdate PICTURE '99/99/99'
         READ
         STORE u:memdate TO cdate
         DO ckdate
         IF state = '1'
            @ 23,10 SAY'INVALID DATE ERROR - TRY AGAIN'
            DO delay
            @ 23,10 SAY'                                '
            LOOP
         ENDIF
         STORE 'F' TO a
      ENDDO
      APPEND BLANK
      REPLACE sc:last WITH l:name,sc:first WITH f:name,;
status WITH t:sta,member:act WITH u:member,date:act WITH ;
u:memdate
   ENDIF
ENDDO
```

```
UPDLED.CMD
*
*
*
DO WHILE T
   ERASE
   @ 1,22 SAY'UPDATE LEADERSHIP RECORD'
   @ 3,10 SAY'Enter blank to return to MENU'
   STORE '                    ' TO u:lead
   @ 5,3 SAY'LEADERSHIP NAME ' GET u:lead
   READ
   IF u:lead = '                    '
      RETURN
   ENDIF
   USE B:leader
   COUNT TO no:rec FOR !(sc:last) = !(l:name) .AND. ;
!(sc:first) =!(f:name) .AND. !(u:lead) = !(leader)
   IF no:rec  0
      LOCATE FOR !(sc:last) = !(l:name) .AND. !(sc:first) ;
= !(f:name) .AND. !(u:lead) = !(leader)
         STORE leader TO t:lead
         STORE le:date TO t:ledate
         STORE le:date TO u:ledate
         @ 5,46 SAY'CHANGE TO ' GET u:lead
         @ 6,3 SAY'DATE             '+!(le:date)
         @ 6,46 SAY'CHANGE TO ' GET u:ledate PICTURE ;
'99/99/99'
         READ
         IF !(t:lead)  !(u:lead)
            COUNT TO numrec FOR !(sc:last) = !(l:name);
.AND. !(sc:first) = !(f:name) .AND. !(leader) = !(u:lead) ;
.AND. le:date = u:ledate
            IF numrec  0
               @ 22,10 SAY'THE CHANGE CANNOT BE MADE'
               @ 23,10 SAY'BECAUSE THAT LEADERSHIP TYPE ;
HAS ALREADY EXISTED'
               DO delay
               LOOP
            ENDIF
         ENDIF
         STORE ' ' TO change
         @ 8,10 SAY'Are these changes corrected? (Y or N)';
GET change
         READ
         IF !(change) = 'N'
            LOOP
         ENDIF
         USE B:leader
         REPLACE leader WITH u:lead,le:date WITH u:ledate ;
FOR !(sc:last) = !(l:name) .AND. !(sc:first) = !(f:name) ;
.AND. !(leader) = !(t:lead) .AND. le:date = t:ledate
      ENDIF
      IF no:rec = 0
         STORE ' ' TO add
         @ 7,10 SAY'THE RECORD  DOES NOT EXIST '
         @ 8,10 SAY'DO YOU WANT TO ADD THIS RECORD? (Y OR ;
N)' GET add
         READ
         IF !(add) = 'N'
            LOOP
         ENDIF
         STORE 'T' TO a
         DO WHILE a = 'T'
            STORE '        ' TO u:ledate
            @ 10,10 SAY'WHAT DATE' GET u:ledate PICTURE ;
'99/99/99'
            READ
            STORE u:ledate TO cdate
            DO ckdate
            IF state = '1'
               @ 23,10 SAY'INVALID DATE ERROR - TRY AGAIN'
               DO delay
               @ 23,10 SAY'                              '
               LOOP
            ENDIF
            STORE 'F' TO a
         ENDDO (* a = 'T' *)
         USE B:leader
         APPEND BLANK
         REPLACE sc:last WITH l:name,sc:first WITH f:name,;
leader WITH u:lead,le:date WITH u:ledate
      ENDIF
   ENDDO
```

```
UPDAWD.CMD
*
*
*
DO WHILE T
   ERASE
   @ 1,22 SAY'UPDATE AWARD RECORD'
   STORE ' ' TO u:type
   @ 4,5 SAY'AWARD TYPE (B-BADGE, S-SKILL AWARD, OR ;
BLANK-TO STOP) ' GET u:type
   READ
   IF u:type = ' '
      RETURN
   ENDIF
   STORE '                    ' TO u:name
   @ 6,5 SAY'AWARD NAME                              ' ;
GET u:name
   READ
   USE B:award
   COUNT TO no:rec FOR !(sc:last) = !(l:name) .AND. ;
!(sc:first) =!(f:name) .AND. !(u:type) = !(type) .AND. ;
!(awd:name) = !(u:name)
   IF no:rec  0
      LOCATE FOR !(sc:last) = !(l:name) .AND. !(sc:first) ;
= !(f:name) .AND. !(u:type) = !(type) .AND. !(u:name) = ;
!(awd:name)
      STORE type TO t:type
      STORE awd:name TO t:awname
      STORE awd:date TO u:date
      @ 4,5 SAY'                                         ;
                     '
      @ 6,5 SAY'                                         ;
                     '
      @ 4,3 SAY'AWARD TYPE '+!(u:type)
      @ 4,43 SAY'CHANGE TO ' GET u:type
      @ 5,3 SAY'AWARD NAME '+!(u:name)
      @ 5,43 SAY'CHANGE TO ' GET u:name
      @ 6,3 SAY'AWARD DATE '+!(awd:date)
      @ 6,43 SAY'CHANGE TO ' GET u:date PICTURE '99/99/99'
      READ
      IF !(t:type)  !(u:type) .OR. !(t:awname)  ;
!(u:name)
         COUNT TO numrec FOR !(sc:last) = !(l:name) .AND. ;
!(sc:first) = !(f:name) .AND. !(type) = !(u:type) .AND. ;
!(awd:name) = !(u:name)
         IF numrec  0
            @ 23,10 SAY'THAT RECORD HAS ALREADY EXISTED'
            DO delay
            LOOP
         ENDIF
      ENDIF
      STORE ' ' TO change
      @ 8,10 SAY'Are these changes corrected? (Y or N)' ;
GET change
      READ
      IF !(change) = 'N'
         LOOP
      ENDIF
      REPLACE type WITH u:type,awd:name WITH u:name,;
awd:date WITH u:date FOR !(sc:last) = !(l:name) .AND. ;
!(sc:first) = !(f:name) .AND. !(type) = !(t:type) .AND. ;
!(awd:name) = !(t:awname)
   ENDIF
   IF no:rec = 0
      STORE ' ' TO add
      @ 6,5 SAY'                                        ;
                      '
      @ 6,10 SAY'THE RECORD DOES NOT EXIST'
      @ 7,10 SAY'DO YOU WANT TO ADD THIS RECORD? (Y OR N) ;
' GET add
      READ
      IF !(add) = 'N'
         LOOP
      ENDIF
      STORE 'T' TO a
      DO WHILE a = 'T'
         STORE '        ' TO u:date
         @ 9,10 SAY'DATE  ' GET u:date PICTURE '99/99/99'
         READ
         STORE u:date TO cdate
         DO ckdate
         IF state = '1'
            @ 23,10 SAY'INVALID DATE ERROR - TRY AGAIN'
            DO delay
            @ 23,10 SAY'                                 '
            LOOP
         ENDIF
         STORE 'F' TO a
      ENDDO
      APPEND BLANK
      REPLACE sc:last WITH l:name,sc:first WITH f:name,;
type WITH u:type,awd:name WITH u:name,awd:date WITH u:date
   ENDIF
ENDDO
```

```
UPDRK.CMD
*
*
*
DO WHILE T
   ERASE
   @ 1,22 SAY"UPDATE SCOUT'S RANK"
   STORE ' ' TO decide
   @ 3,10 SAY'IS THIS A NEW RANK? (Y OR N)' GET decide
   @ 4,10 SAY'ENTER BLANK TO RETURN TO MENU'
   READ
   IF decide = ' '
      RETURN
   ENDIF
   USE B:screcord
   LOCATE FOR !(sc:last ) = !(l:name) .AND. !(sc:first) = ;
!(f:name)
   DO CASE
      CASE !(rank) = 'N'
         STORE 'T' TO n:rank
      CASE !(rank) = 'T'
         STORE '2' TO n:rank
      CASE !(rank) = '2'
         STORE '1' TO n:rank
      CASE !(rank) = '1'
         STORE 'S' TO n:rank
      CASE !(rank) = 'S'
         STORE 'L' TO n:rank
      CASE !(rank) = 'L'
         STORE 'E' TO n:rank
   ENDCASE
   IF !(decide) = 'Y'
      STORE '        ' TO u:rkdate
      @ 6,10 SAY'NEW RANK    ' +!(n:rank)
      @ 7,10 SAY'DATE PASSED' GET u:rkdate PICTURE ;
'99/99/99'
      READ
      IF u:rkdate = '  /  /  '
         RETURN
      ENDIF
      USE B:advance
      APPEND BLANK
      REPLACE sc:last WITH l:name,sc:first WITH f:name,;
rank WITH n:rank,rank:date WITH u:rkdate
      USE B:screcord
      REPLACE rank WITH n:rank FOR !(sc:last ) = ;
!(l:name) .AND. !(sc:first) = !(f:name)
   ENDIF
   IF !(decide) = 'N'
      STORE 'T' TO a
      DO WHILE a = 'T'
         STORE ' ' TO u:rank
         @ 6,10 SAY'ENTER RANK NAME OR BLANK TO STOP' ;
GET u:rank
         READ
         IF u:rank = ' '
            RETURN
         ENDIF
         USE B:advance
         COUNT TO no:rec FOR !(sc:last) = !(l:name) ;
.AND. !(sc:first) = !(f:name) .AND. !(rank) = !(u:rank)
         IF no:rec = 0
            @ 23,10 SAY'NOT FOUND - TRY AGAIN'
            DO delay
            LOOP
         ENDIF
         STORE 'T' TO b
         DO WHILE b = 'T'
            LOCATE FOR !(sc:last) = !(l:name) .AND. ;
!(sc:first) = !(f:name) .AND. !(rank) = !(u:rank)
            STORE rank to t:rank
            STORE rank:date TO t:rkdate
            STORE rank:date TO u:rkdate
            @ 7,10 SAY'DATE PASSED '+rank:date
            @ 7,40 SAY'CHANGE TO ' GET u:rkdate PICTURE ;
'99/99/99'
            READ
            STORE u:rkdate TO cdate
            DO ckdate
            IF state = '1'
               @ 23,10 SAY'INVALID DATE ERROR - TRY AGAIN'
               DO delay
               @ 23,10 SAY'                                  '
               @                  7,10                     SAY'
'

               LOOP
            ENDIF
            STORE 'F' TO b
         ENDDO
         STORE ' ' TO change
         @ 10,10 SAY'Is this change corrected? (Y or N)' GET
change
         READ
         IF !(change) = 'N'
            @ 7,10 SAY'                                    ;
                    '

            @ 10,10 SAY'                                   ;
                     '
            LOOP
         ENDIF
         STORE 'F' TO a
      ENDDO
      USE B:advance
      REPLACE rank:date WITH u:rkdate FOR !(sc:last) = ;
!(l:name) .AND. !(sc:first) = !(f:name) .AND. !(rank) = ;
!(u:rank)
   ENDIF
ENDDO
```

```
UPDACC.CMD
*
*
*
USE B:account
COPY TO B:temp FOR !(sc:last) = !(l:name) .AND. ;
!(sc:first) = !(f:name)
* b:temp contains only the record of a particular scout
SELECT PRIMARY
USE B:trprog
SELECT SECONDARY
USE B:temp
JOIN TO B:temp1 FOR prog:date  S.prog:date FIELD ;
sc:last,sc:first,prog:date,event,att:code,amt:due,;
amt:paid,balance
USE B:temp
INDEX ON $(prog:date,7,2)+$(prog:date,1,2)+;
$(prog:date,4,2) TO B:tmindx
GOTO BOTTOM
STORE balance TO bal
STORE prog:date TO last:date
* find previous balance
USE B:temp1
COPY TO B:temp2 FOR $(prog:date,7,2) = $(last:date,7,2) ;
.AND. $(prog:date,1,2) = $(last:date,1,2) .AND. ;
$(prog:date,4,2)  $(last:date,4,2)
USE B:temp2
INDEX ON $(prog:date,7,2)+$(prog:date,1,2)+;
$(prog:date,4,2) TO B:t2indx
REPLACE att:code WITH 'N',amt:paid WITH 0,balance WITH bal
USE B:account
APPEND FROM B:temp2
DO WHILE T
   ERASE
   @ 1,22 SAY"UPDATE THE SCOUT'S ACCOUNT"
   @ 3,5 SAY'ENTER BLANK TO STOP'
   STORE '        ' TO u:date
   @ 5,5 SAY'ENTER PROGRAM DATE ' GET u:date PICTURE  ;
'99/99/99'
   READ
   IF u:date = '  /  /  '
      DELETE FILE B:temp2
      DELETE FILE B:temp1
      RETURN
   ENDIF
   COUNT TO no:rec FOR prog:date = u:date .AND. !(sc:last);
= !(l:name) .AND. !(sc:first) = !(f:name)
   IF no:rec = 0
```

```
* the case - there are no program on that day
      @ 23,10 SAY'NOT FOUND - TRY AGAIN'
      DO delay
      LOOP
   ENDIF
   IF no:rec  1
* the case - there are more than one program on that day, ;
a particular
* activity is used to help identify scout's record
      STORE '                         ' TO u:act
      @ 6,5 SAY'      ACTIVITY                    ' GET ;
u:act
      READ
      IF u:act = '                         '
         LOOP
      ENDIF
* check if there is such activity on that day
      COUNT TO numrec FOR prog:date = u:date .AND. ;
!(sc:last) = !(l:name) .AND. !(sc:first) = !(f:name) ;
.AND. !(event) = !(u:act)
      IF numrec = 0
* the case there is no such activity
         @ 23,10 SAY'NOT FOUND TRY AGAIN'
         DO delay
         LOOP
      ENDIF
* find the amount due on that day
      USE B:trprog
      LOCATE FOR prog:date = u:date .AND. !(event) = ;
!(u:act)
      STORE amt:due TO t:due
      USE B:account
      LOCATE FOR prog:date = u:date .AND. !(sc:last) = ;
!(l:name) .AND. !(sc:first) = !(f:name) .AND. !(event) ;
= !(u:act)
   ENDIF
   IF no:rec = 1
* the case there is only 1 program on that day
      USE B:trprog
      LOCATE FOR prog:date = u:date
      STORE amt:due TO t:due
      USE B:account
      LOCATE FOR prog:date = u:date .AND. !(sc:last) = ;
!(l:name) .AND. !(sc:first) = !(f:name)
   ENDIF
   STORE 0 TO adj:paid
   STORE prog:date TO t:date
   STORE att:code TO t:code
   STORE amt:paid TO t:paid
   STORE att:code  TO u:code
   STORE amt:paid TO u:paid
   @ 7,5 SAY '      ATTEND CODE (S,A,T,OR N) '+!(t:code)
   @ 7,45 SAY'CHANGE TO ' GET u:code
   READ
   IF !(t:code)  !(u:code)
* the case - there is a change on the attend code
      IF !(u:code)  'N'
*  if scout participated the activity, however chack if that
record existed
         COUNT TO norec FOR !(sc:last) = !(l:name) .AND. ;
!(sc:first) = !(f:name) .AND. prog:date = t:date .AND. ;
att:code = u:code
         IF norec  0
* the change cannot be made because that record existed
            @ 23,10 SAY'THE SCOUT HAS ALREADY ;
PARTICIPATED ANOTHER ACTIVITY'
            DO delay
            LOOP
         ENDIF
      ENDIF
      IF !(u:code) = 'N'
* the case - the scout did not participate the activity
         STORE 0 TO u:paid
         STORE t:due TO adj:paid
      ENDIF
   ENDIF
   @ 8,5 SAY'      AMOUNT PAID                          ;
'+STR(t:paid,7,2)
   @ 8,45 SAY'CHANGE TO ' GET u:paid PICTURE '9999.99'
   READ
   IF amt:paid = u:paid .AND. u:paid = 0 .AND. !(u:code) ;
 'N'
* the case - the scout participated, however, did not pay
* the due
      STORE adj:paid-t:due TO adj:paid
   ENDIF
   IF amt:paid  u:paid .AND. !(u:code)  'N'
* the case - the scout participated, and there is some
* changes in the
* amount he pay
      IF $(u:date,7,2) = $(last:date,7,2) .AND. ;
$(u:date,1,2) = $(last:date,1,2) .AND. $(u:date,4,2) ;
= $(last:date,4,2)
         STORE (u:paid-amt:paid) TO adj:paid
      ENDIF
      IF $(u:date,7,2) = $(last:date,7,2) .AND. ;
$(u:date,1,2) = $(last:date,1,2) .AND. $(u:date,4,2) ;
 $(last:date,4,2)
         STORE (u:paid-amt:paid)-t:due TO adj:paid
      ENDIF
   ENDIF
   STORE ' ' TO change
   @ 11,10 SAY'Are those changes corrected? (Y or N)' GET ;
change
   READ
   IF !(change) = 'N'
      LOOP
   ENDIF
   IF no:rec  1
      REPLACE att:code WITH u:code,amt:paid WITH u:paid,;
balance WITH (balance- adj:paid) FOR !(sc:last) = ;
!(l:name) .AND. !(sc:first) = !(f:name) .AND. prog:date ;
= t:date .AND. !(att:code) = !(t:code) .AND. !(event) ;
= !(u:act)
   ENDIF
   IF no:rec = 1
      REPLACE att:code WITH u:code,amt:paid WITH u:paid,;
balance WITH (balance-adj:paid) FOR !(sc:last) = ;
!(l:name) .AND. !(sc:first) = !(f:name) .AND. prog:date ;
= t:date .AND. !(att:code) = !(t:code)
   ENDIF
   REPLACE balance WITH (balance - adj:paid) FOR ;
!(sc:last) = !(l:name) .AND. !(sc:first) = !(f:name) ;
.AND.    $(prog:date,7,2)     =     $(t:date,7,2)     .AND.
$(prog:date,1,2) = $(t:date,1,2) .AND. $(prog:date,4,2) ;
 $(t:date,4,2)
ENDDO
```

SCLIST.CMD

```
*
*
*
DO setup
RESTORE FROM B:memfile
DO WHILE T
   ERASE
   STORE ' ' TO sc:sel
   @ 1,15  SAY " LIST SCOUT'S INFORMATION"
   @ 1,60  SAY 'DATE'
   @ 1,65  SAY DATE()
   @ 3,15  SAY " 0 - RETURN TO SCOUT'S MENU"
   @ 4,15  SAY ' 1 - LIST TROOP ROSTER'
   @ 5,15  SAY ' 2 - LIST FAMILY DATA'
   @ 6,15  SAY ' 3 - LIST MEMBERSHIP DATA'
   @ 7,15  SAY ' 4 - LIST LEADERSHIP DATA'
   @ 8,15  SAY ' 5 - LIST MERIT BADGE  SKILL AWARDS'
   @ 9,15  SAY ' 6 - LIST ADVANCEMENT'
   @ 10,15 SAY ' 7 - LIST ATTENDANCE AND DUES'
   @ 11,15 SAY ' 8 - LIST OUR CAMPING LOG '
   @ 14,15 SAY ' ENTER YOUR SELECTION' GET sc:sel
   READ
   DO CASE
   CASE sc:sel = '0'
      RELEASE ALL
      RETURN
   CASE sc:sel = '1'
      STORE "TROOP ROSTER RECORD" TO stitle
      STORE 'screcord' TO scfile
      STORE 'slone' TO onelist
      STORE 'slall' TO alllist
      DO slistl
   CASE sc:sel = '2'
      STORE "SCOUT'S FAMILY DATA" TO stitle
      STORE 'screcord' TO scfile
      STORE 'guardian' TO gfile
      STORE 's2one' TO onelist
      STORE 's2all' TO alllist
      DO slistl
   CASE sc:sel = '3'
      STORE "SCOUT'S MEMBERSHIP RECORD" TO stitle
      STORE 'screcord' TO scfile
      STORE 'member' TO mbfile
      STORE 's3one' TO onelist
      STORE 's3all' TO alllist
      DO slistl
   CASE sc:sel = '4'
```

```
      STORE "SCOUT'S LEADERSHIP RECORD" TO stitle
      STORE 'leader' TO lefile
      STORE 'screcord' TO scfile
      STORE 's4one' TO onelist
      STORE 's4all' TO alllist
      DO slistl
   CASE sc:sel = '5'
      STORE "MERIT BADGES  SKILL AWARDS" TO stitle
      STORE 'screcord' TO scfile
      STORE 'award' TO awfile
      STORE 's5one' TO onelist
      STORE 's5all' TO alllist
      DO slistl
   CASE sc:sel = '6'
      STORE "ADVANCEMENT" TO stitle
      STORE 'member' TO mbfile
      STORE 'advance' TO advfile
      STORE 'screcord' TO scfile
      STORE 's6one' TO onelist
      STORE 's6all' TO alllist
      DO slistl
   CASE sc:sel = '7'
      STORE "ATTENDANCE  DUES" TO stitle
      STORE 'trprog' TO trfile
      STORE 'account' TO accfile
      STORE 's7one' TO onelist
      STORE 's7all' TO alllist
      DO slistl
   CASE sc:sel = '8'
      STORE "OUR CAMPING AND ACTIVITIES LOG" TO stitle
      STORE 'account' TO accfile
      STORE 'trprog' TO trfile
      STORE 's8one' TO onelist
      STORE 's8all' TO alllist
      DO slistl
   OTHERWISE
      @ 23,10 SAY 'WRONG CODE - TRY AGAIN'
      DO delay
      LOOP
   ENDCASE
ENDDO
RETURN
```

```
SLIST1.CMD
*
*
*
DO setup
STORE 23 TO line
STORE 'ERASE' TO npage
ERASE
@ 1,22 SAY stitle
@ 1,60 SAY DATE()
@ 3,13 SAY 'DO YOU WANT TO LIST - '
@ 4,16 SAY"1. ONE PERSON'S RECORD - OR -"
@ 5,16 SAY"2. EVERYONE'S RECORD"
@ 6,18 SAY'(RETURN TO EXIT)'
WAIT TO want
IF want = '1'
   STORE 'T' TO a
   DO WHILE a = 'T'
      STORE '               ' TO 1:name
      STORE '               ' TO f:name
      ERASE
      @ 5,20 SAY'TO LIST A PARTICULAR RECORD '
      @ 7,10 SAY'ENTER LAST NAME ' GET 1:name
      READ
      IF !(1:name) = '               '
         RETURN
      ENDIF
      @ 8,16 SAY'FIRST NAME' GET f:name
      READ
      USE B:screcord
      COUNT TO no:rec FOR !(sc:last) = !(1:name) .AND ;
!(sc:first) = !(f:name)
      IF no:rec = 0
         @ 23,10 SAY'There is no such record.'
         DO delay
         LOOP
      ENDIF
      STORE 'F' TO a
   ENDDO
   ERASE
   DO onelist
   IF flag = '1'
      RETURN
   ENDIF
   DO delay
   STORE ' ' TO rpt
   @ row,15 SAY'DO YOU WANT A REPORT OF THAT LIST? (Y/N) ';
GET rpt
```

```
   READ
   IF !(rpt) = 'N'
      RETURN
   ENDIF
   ERASE
   @ 5,15 SAY'PREPARE THE PRINTER FOR PRINT OUT'
   @ 7,15 SAY'PRESS ANY KEY WHEN READY'
   WAIT
   STORE 55 TO line
   SET FORMAT TO PRINT
   SET CONSOLE OFF
   SET PRINT ON
   STORE 'EJECT' TO npage
   DO onelist
   SET FORMAT TO SCREEN
   SET PRINT OFF
   SET CONSOLE ON
   @ 10,15 SAY'THE REPORT HAS ALREADY FINISHED'
   @ 11,15 SAY'PRESS ANY KEY TO RETURN TO MENU'
   WAIT
ENDIF
IF want = '2'
   ERASE
   @ 2,22 SAY stitle
   @ 4,15 SAY'TO HALT THE LISTING - PRESS CTRL  S TOGETHER'
   @ 5,15 SAY'TO CONTINUE - PRESS ANY KEY'
   @ 7,15 SAY'IF YOU ARE READY PRESS RETURN'
   WAIT
   ERASE
   DO alllist
   DO delay
   ERASE
   @ 5,10 SAY'THIS IS THE END OF FILE.'
   STORE ' ' TO rpt
   @ 7,10 SAY'DO YOU WANT A REPORT ON THAT LIST? (Y/N) ' GET
rpt
   READ
   IF !(rpt) = 'N'
      RETURN
   ENDIF
   ERASE
   @ 5,15 SAY' PREPARE THE PRINTER FOR PRINT OUT'
   @ 7,15 SAY'PRESS ANY KEY WHEN READY'
   WAIT
   STORE 55 TO line
   SET FORMAT TO PRINT
   SET CONSOLE OFF
   SET PRINT ON
   STORE 'EJECT' TO npage
   DO alllist
   SET FORMAT TO SCREEN
   SET PRINT OFF
   SET CONSOLE ON
   @ 10,15 SAY'THE REPORT HAS ALREADY FINISHED'
   @ 11,15 SAY'PRESS ANY KEY TO RETURN TO MENU'
   WAIT
ENDIF
RETURN
```

S1ALL.CMD

```
*
*
*
STORE VAL($(pre:date,7,2)) TO p:yr
STORE VAL($(pre:date,1,2)) TO p:mo
SELECT PRIMARY
USE B:scfile
INDEX ON !(sc:last)+!(sc:first) TO B:scindx
GOTO TOP
DO WHILE .NOT. EOF
   @ 3,20 SAY stitle
   @ 3,60 SAY 'DATE'
   @ 3,65 SAY date()
   @ 5,7  SAY 'NAME'
   @ 5,25 SAY 'ADDRESS'
   @ 5,44 SAY 'OFFICE'
   @ 5,57 SAY 'AGE STA PRO UNI BOY'
   STORE 7 TO row
   DO WHILE (.NOT. EOF) .AND. row  line
      STORE '  ' TO sc:yr
      STORE '  ' TO sc:mo
      STORE 0 TO age
      STORE VAL($(sc:birth,7,2)) TO sc:yr
      STORE VAL($(sc:birth,1,2)) TO sc:mo
      STORE p:yr-sc:yr TO age
      IF sc:mo  p:mo
         STORE age-1 TO age
      ENDIF
      @ row,2  SAY TRIM(!(sc:last))+', '
      @ row,20 SAY TRIM(!(sc:addr))+' '+TRIM(!(sc:office))
      @ row,58 SAY STR(age,2,0)+'  '+!(status)
      @ row,66 SAY !(rank)+'   '+!(uniform)+'    '+;
!(boy:life)
      STORE row+1 TO row
      @ row,2 SAY TRIM(!(sc:first))
      @ row,20 SAY TRIM(!(sc:city))+', '+!(sc:zip)
      STORE row+2 TO row
      DO delay
      SKIP
   ENDDO
   npage
ENDDO
DO delay
RETURN
```

```
S1ONE.CMD
*
*
*
SELECT PRIMARY
USE B:screcord
LOCATE FOR !(sc:last)=!(l:name) .AND. !(sc:first)=!(f:name)
STORE VAL($(PRE:DATE,7,2)) TO p:yr
STORE VAL($(PRE:DATE,1,2)) TO p:mo
STORE VAL($(SC:BIRTH,7,2)) TO sc:yr
STORE VAL($(SC:BIRTH,1,2)) TO sc:mo
STORE p:yr-sc:yr TO age
IF sc:mo  p:mo
      STORE age-1 TO age
ENDIF
@ 3,20 SAY stitle
@ 3,60 SAY 'DATE'
@ 3,65 SAY date()
@ 5,7  SAY 'NAME'
@ 5,25 SAY 'ADDRESS'
@ 5,44 SAY 'OFFICE'
@ 5,57 SAY 'AGE. STA. PRO UNI. BOY '
@ 7,2  SAY TRIM(!(sc:last))+','
@ 7,20 SAY TRIM(!(sc:addr))+' '+TRIM(!(sc:office))
@ 7,58 SAY STR(age,2,0)+'   '+!(status)
@ 7,66 SAY !(rank)+'   '+!(uniform)+'    '+!(boy:life)
@ 8,2  SAY TRIM(!(sc:first))
@ 8,20 SAY TRIM(!(sc:city))+', '+!(sc:zip)
RETURN
```

```
S2ALL.CMD
*
*
*
SELECT PRIMARY
USE B:scfile
SELECT SECONDARY
USE B:gfile
SELECT PRIMARY
JOIN TO B:temp FOR !(g:name)=!(s.g:name) FIELD sc:last,;
sc:first,sc:hphone, sc:church,g:name,g:relate,g:occup,;
g:bphone
USE B:temp
INDEX ON !(sc:last)+!(sc:first) TO b:tmindx
GOTO TOP
DO WHILE .NOT. EOF
   @ 2,25 SAY stitle
   @ 2,60 SAY 'DATE'
   @ 2,65 SAY DATE()
   @ 4,6 SAY 'NAME'
   @ 4,21 SAY 'HOME PHONE'
   @ 4,33 SAY 'CHURCH'
   @ 4,55 SAY "GUARDIAN'S"
   STORE 6 TO row
   DO WHILE (.NOT. EOF) .AND. row  line
      @ row,2 SAY TRIM(!(sc:last))+', '
      @ row,19 SAY sc:hphone
      @ row,34 SAY !(sc:church)
      @ row,46 SAY 'NAME'
      @ row,57 SAY !(g:name)
      STORE row+1 TO row
      @ row,2  SAY !(sc:first)
      @ row,46 SAY 'OCCUPATION '+!(g:occup)
      STORE row+1 TO row
      @ row,46 SAY 'BUS.PHONE  '+g:bphone
      STORE row+1 TO row
      @ row,46 SAY 'RELATION   ' +!(g:relate)
      STORE row+2 TO row
      DO delay
      SKIP
      IF (row + 5)  line
         STORE line + 1 TO row
      ENDif
   ENDDO
   npage
ENDDO
DO delay
USE B:scfile
```

```
DELETE FILE B:temp
RETURN
```

S2ONE.CMD

```
*
*
*
SELECT PRIMARY
USE B:screcord
SELECT SECONDARY
USE B:guardian
SELECT PRIMARY
JOIN TO B:temp FOR !(g:name)=!(s.g:name) FIELD sc:last,;
sc:first,sc:hphone,sc:church,g:name,g:relate,g:occup,;
g:bphone
USE B:temp
LOCATE FOR !(sc:last)=!(l:name) .AND. !(sc:first)=!(f:name)
@ 2,20 SAY stitle
@ 2,60 SAY'DATE'
@ 2,65 SAY DATE()
@ 4,6 SAY 'NAME'
@ 4,21 SAY 'HOME PHONE'
@ 4,35 SAY 'CHURCH'
@ 4,55 SAY "GUARDIAN'S"
@ 6,2 SAY TRIM(!(sc:last))+', '
@ 6,19 SAY sc:hphone
@ 6,34 SAY !(sc:church)
@ 6,46 SAY 'NAME'
@ 6,57 SAY !(g:name)
@ 7,2  SAY !(sc:first)
@ 7,46 SAY 'OCCUPATION '+!(g:occup)
@ 8,46 SAY 'BUSI.PHONE '+g:bphone
@ 9,46 SAY 'RELATION   '+!(g:relate)
USE B:screcord
DELETE FILE B:temp
RETURN
```

S3ALL.CMD

```
*
*
*
SELECT PRIMARY
USE B:mbfile
SELECT SECONDARY
USE B:scfile
JOIN TO B:temp FOR !(sc:last)=!(s.sc:last) .AND. ;
!(sc:first) = !(s.sc:first) FIELD sc:last,sc:first,status,;
member:act,date:act,sc:birth,patrolname
USE B:temp
INDEX ON !(sc:last)+!(sc:first)+!(patrolname);
+!(member:act)+$(date:act,7,2)+$(date:act,1,2)+;
$(date:act,4,2) TO B:tmindx
GOTO TOP
STORE sc:last TO t:last
STORE sc:first TO t:first
STORE sc:birth TO t:birth
STORE patrolname TO t:patname
DO WHILE .NOT. EOF
   @ 1,25 SAY stitle
   @ 1,60 SAY 'DATE'
   @ 1,65 SAY date()
   @ 2,5 SAY 'NAME'
   @ 2,21 SAY 'BIRTH DATE  PATROL NAME'
   @ 2,47 SAY 'MEMBERSHIP ACTIVITY'
   @ 2,69 SAY 'ACT. DATE'
   STORE 4 TO row
   DO WHILE .NOT. EOF .AND. row  line
      @ row,2 SAY TRIM(!(t:last))+', '
      @ row,21 SAY !(t:birth)
      @ row,33 SAY !(t:patname)
      @ row,46 SAY !(member:act)
      @ row,69 SAY date:act
      STORE row+1 TO row
      @ row,2 SAY TRIM(!(t:first))
      SKIP
      DO WHILE .NOT EOF .AND. (!(sc:Last)=!(t:last) .AND. ;
!(sc:first) = !(t:first)) .AND. row  line
         @ row,46 SAY !(member:act)
         @ row,69 SAY date:act
         STORE row+1 TO row
         DO delay
         SKIP
         IF (row+5)  line
            STORE line + 2 TO row
         ENDIF
      ENDDO
      STORE row + 1 TO row
      IF .NOT. EOF
         STORE sc:last TO t:Last
         STORE sc:first TO t:first
         STORE sc:birth TO t:birth
         STORE patrolname TO t:patname
      ENDIF
   ENDDO
   DO delay
   npage
ENDDO
DO delay
USE B:scfile
DELETE FILE B:temp
RETURN
```

```
S3ONE.CMD
*
*
*
SELECT PRIMARY
USE B:screcord
LOCATE FOR !(sc:last)=!(l:name) .AND. !(sc:first)=!(f:name)
STORE sc:birth TO t:birth
STORE patrolname TO t:patname
SELECT SECONDARY
USE B:member
COPY TO B:temp FOR !(sc:last)=!(l:name) .AND. !(sc:first)=;
!(f:name)
USE B:temp
INDEX ON !(member:act)+$(date:act,7,2)+$(date:act,1,2)+;
$(date:act,4,2) TO b:tmindx
GOTO TOP
DO WHILE .NOT. EOF
   @ 1,25 SAY stitle
   @ 1,60 SAY 'DATE'
   @ 1,65 SAY date()
   @ 3,6 SAY 'NAME'
   @ 3,21 SAY 'BIRTH DATE'
   @ 3,33 SAY 'PATROL NAME'
   @ 3,47 SAY 'MEMBERSHIP ACTIVITY'
   @ 3,69 SAY 'ACT. DATE'
   @ 5,2 SAY TRIM(!(sc:last))+', '
   @ 5,21 SAY t:birth
   @ 5,33 SAY !(t:patname)
   @ 5,46 SAY !(member:act)
   @ 5,69 SAY date:act
   @ 6,2 SAY TRIM(!(sc:first))
   DO delay
   SKIP
   STORE 6 TO row
   DO WHILE (.NOT. EOF) .AND. row  line
      @ row,46 SAY !(member:act)
      @ row,69 SAY date:act
      STORE row+1 TO row
      SKIP
      DO delay
   ENDDO
   IF .NOT. EOF
      npage
   ENDIF
ENDDO
DO delay
USE B:screcord
DELETE FILE B:temp
```

```
RETURN
```

S4ALL.CMD

```
*
*
*
SELECT PRIMARY
USE B:lefile
INDEX ON !(sc:last)+!(sc:first)+!(leader)+$(le:date,7,2)+;
$(le:date,1,2)+$(le:date,4,2) TO B:leindx
GOTO TOP
STORE sc:last TO t:last
STORE sc:first TO t:first
SELECT SECONDARY
USE B:scfile
LOCATE FOR !(sc:last)=!(t:last) .AND. !(sc:first)=;
!(t:first)
STORE sc:office TO t:office
SELECT PRIMARY
DO WHILE .NOT. EOF
   @ 1,24 SAY stitle
   @ 1,60 SAY 'DATE'
   @ 1,65 SAY date()
   @ 3,9 SAY 'NAME'
   @ 3,36 SAY 'OFFICE'
   @ 3,49 SAY 'LEADERSHIP ACTIVITY'
   @ 3,72 SAY 'DATE'
   STORE 5 TO row
   DO WHILE .NOT. EOF .AND. row  line
      @ row,2 SAY TRIM(!(sc:last))+', '+TRIM(!(sc:first))
      @ row,35 SAY TRIM(!(sc:office))
      DO WHILE (.NOT. EOF) .AND. (!(sc:last)=!(t:last) ;
.AND. !(sc:first)=!(t:first)) .AND. row  line
         @ row,50 SAY !(leader)
         @ row,70 SAY le:date
         STORE row+1 TO row
         DO delay
         SKIP
         IF (row+5)  line
            STORE line + 2 TO row
         ENDIF
      ENDDO
      STORE row+1 TO row
      IF .NOT. EOF
         STORE sc:last TO t:last
         STORE sc:first TO t:first
         SELECT SECONDARY
         LOCATE FOR !(sc:last)=!(t:last) .AND. !(sc:first);
=!(t:first)
         STORE sc:office TO t:office
```

```
          SELECT PRIMARY
       ENDIF
    ENDDO
    npage
ENDDO
DO delay
RETURN
```

```
S4ONE.CMD
*
*
*
USE B:screcord
LOCATE FOR !(sc:last)=!(l:name) .AND. !(sc:first)=!(f:name)
STORE sc:office TO t:office
USE B:leader
COUNT TO no:rec FOR !(sc:last)=!(l:name) .AND. !(sc:first);
=!(f:name)
IF no:rec = 0
   STORE '1' TO flag
   @ 23,10 SAY'NO RECORDS OF LEADERSHIP'
   DO delay
   RETURN
ENDIF
COPY TO B:temp FOR !(sc:last)=!(l:name) .AND. !(sc:first);
=!(f:name)
USE B:temp
INDEX ON !(sc:last)+!(sc:first)+!(leader)+$(le:date,7,2)+;
$(le:date,1,2)+$(le:date,4,2) TO b:tmindx
GOTO TOP
DO WHILE .NOT. EOF
   @ 1,24 SAY stitle
   @ 1,60 SAY 'DATE'
   @ 1,65 SAY date()
   @ 3,9 SAY 'NAME'
   @ 3,36 SAY 'OFFICE'
   @ 3,52 SAY 'LEADERSHIP ACTIVITY'
   @ 3,72 SAY 'DATE'
   @ 5,2 SAY TRIM(!(sc:last))+', '+TRIM(!(sc:first))
   @ 5,35 SAY !(t:office)
   STORE 5 TO row
   DO WHILE (.NOT. EOF) .AND. row  line
      @ row,50 SAY !(leader)
      @ row,70 SAY le:date
      STORE row+1 TO row
      DO delay
      SKIP
   ENDDO
   IF .NOT. EOF
      npage
   ENDIF
ENDDO
USE B:leader
DELETE FILE B:temp
RETURN
```

```
S5ALL.CMD
*
*
*
SELECT PRIMARY
USE B:awfile
INDEX ON !(sc:last)+!(sc:first)+!(type)+!(awd:name) TO ;
B:awindx
GOTO TOP
STORE sc:last TO t:last
STORE sc:first TO t:first
SELECT SECONDARY
USE B:scfile
LOCATE FOR !(sc:last)=!(t:last) .AND. !(sc:first)=;
!(t:first)
STORE sc:office TO t:office
DO WHILE .NOT. EOF
   @ 1,29 SAY stitle
   @ 1,60 SAY 'DATE'
   @ 1,65 SAY date()
   @ 3,10 SAY 'NAME'
   @ 3,35 SAY 'OFFICE'
   @ 3,48 SAY 'TYPE'
   @ 3,54 SAY 'NAME'
   @ 3,73 SAY 'DATE'
   STORE 5 TO row
   DO WHILE .NOT. EOF .AND. row  line
      @ row,2 SAY TRIM(!(t:last))+', '+TRIM(!(t:first))
      @ row,35 SAY TRIM(!(t:office))
      DO WHILE (.NOT. EOF) .AND. !(sc:last)=!(t:last) ;
.AND. !(sc:first)= !(t:first) .AND. rowline
         @ row,50 SAY !(type)
         @ row,52 SAY !(awd:name)
         @ row,72 say awd:date
         STORE row+1 TO row
         DO delay
         SELECT PRIMARY
         SKIP
         IF row+5  line
            STORE line + 2 TO row
         ENDIF
      ENDDO
      STORE row+1 TO row
      IF .NOT. EOF
         STORE sc:last TO t:last
         STORE sc:first TO t:first
         SELECT SECONDARY
         LOCATE FOR !(sc:last)=!(t:last) .AND. !(sc:first);
```

```
=!(t:first)
         STORE sc:office TO t:office
         SELECT PRIMARY
      ENDIF
   ENDDO
   npage
ENDDO
DO delay
RETURN
```

S5ONE.CMD

```
*
*
*
USE B:screcord
LOCATE FOR !(sc:last)=!(l:name) .AND. !(sc:first)=!(f:name)
STORE sc:office TO t:office
USE B:award
COUNT TO no:rec FOR !(sc:last)=!(l:name) .AND. !(sc:first);
= !(f:name)
IF no:rec = 0
   STORE '1' TO flag
   @ 23,10 SAY'NO RECORD OF AWARD RECEIVED'
   DO delay
   RETURN
ENDIF
COPY TO B:temp FOR !(sc:last)=!(l:name) .AND. !(sc:first)=;
!(f:name)
USE B:temp
INDEX ON !(sc:last)+!(sc:first)+!(type)+!(awd:name) TO ;
b:tmindx
GOTO TOP
DO WHILE .NOT. EOF
   @ 1,29 SAY stitle
   @ 1,60 SAY 'DATE'
   @ 1,65 SAY date()
   @ 3,10 SAY 'NAME'
   @ 3,35 SAY 'OFFICE'
   @ 3,48 SAY 'TYPE'
   @ 3,54 SAY'NAME'
   @ 3,73 SAY 'DATE '
   @ 5,3 SAY TRIM(!(sc:last))+', '+TRIM(!(sc:first))
   @ 5,34 SAY TRIM(!(t:office))
   STORE 5 TO row
   DO WHILE (.NOT. EOF) .AND. row  line
      @ row,50 SAY !(type)
      @ row,52 SAY !(awd:name)
      @ row,72 SAY awd:date
      STORE row+1 TO row
      DO delay
      SKIP
   ENDDO
   IF .NOT. EOF
      npage
   ENDIF
ENDDO
USE B:award
DELETE FILE B:temp
```

```
RETURN
```

```
S6ALL.CMD
*
*
*
USE B:mbfile
COPY TO B:temp FOR !(member:act) = 'BECAME SCOUT'
SELECT PRIMARY
USE B:temp
SELECT SECONDARY
USE B:scfile
JOIN TO B:temp1 FOR !(sc:last)=!(s.sc:last) .AND. ;
!(sc:first)=!(s.sc:first) FIELD sc:last,sc:first,;
sc:office,date:act
SELECT PRIMARY
USE B:temp1
SELECT SECONDARY
USE B:advfile
JOIN TO B:temp2 FOR !(sc:last)=!(s.sc:last) .AND. ;
!(sc:first)=!(s.sc:first) FIELD sc:last,sc:first,;
sc:office,date:act,rank,rank:date
USE B:temp2
INDEX ON !(sc:last)+!(sc:first)+$(rank:date,7,2)+;
$(rank:date,1,2)+$(rank:date,4,2) TO B:tmindx
GOTO TOP
STORE sc:last TO t:last
STORE sc:first TO t:first
STORE sc:office TO t:office
STORE date:act TO t:mact
DO WHILE .NOT. EOF
   @ 1,27 SAY stitle
   @ 1,60 SAY 'DATE'
   @ 1,65 SAY date()
   @ 3,6 SAY 'NAME'
   @ 3,30 SAY 'OFFICE'
   @ 3,50 SAY 'BE SCOUT RANK    DATE'
   STORE 5 TO row
   DO WHILE .NOT. EOF .AND. row  line
      @ row,2 SAY TRIM(!(sc:last))+', '+TRIM(!(sc:first))
      @ row,35 SAY !(t:office)
      @ row,50 SAY t:mact
      DO WHILE (.NOT. EOF) .AND. (!(sc:last)=!(t:last) ;
.AND. !(sc:first)=!(t:first)) .AND. row  line
         @ row,60 SAY !(rank)
         @ row,65 SAY rank:date
         STORE row+1 TO row
         DO delay
         IF (row+6)  line
            STORE line+2 TO row
         ENDIF
         SKIP
      ENDDO
      STORE row+1 TO row
      IF .NOT. EOF
         STORE sc:last TO t:last
         STORE sc:first TO t:first
         STORE sc:office TO t:office
         STORE date:act TO t:mact
      ENDIF
   ENDDO
   npage
ENDDO
DO delay
USE B:scfile
DELETE FILE B:temp
DELETE FILE B:temp1
DELETE FILE B:temp2
RETURN
```

```
S6ONE.CMD
*
*
*
SELECT PRIMARY
USE B:screcord
IF !(rank) = 'N'
   STORE '1' TO flag
   @ 23,10 SAY'NO RECORD OF ADVANCEMENT'
   DO delay
   RETURN
ENDIF
LOCATE FOR !(sc:last)=!(l:name) .AND. !(sc:first)=!(f:name)
STORE sc:office TO t:office
USE B:member
LOCATE FOR !(sc:last)=!(l:name) .AND. !(sc:first)=;
!(f:name) .AND. member:act='BECAME SCOUT'
STORE date:act TO t:mact
USE B:advance
COPY TO B:temp FOR !(sc:last)=!(l:name) .AND. !(sc:first);
=!(f:name)
USE B:temp
INDEX ON $(rank:date,7,2)+$(rank:date,1,2)+;
$(rank:date,4,2) TO B:tmindx
GOTO TOP
DO WHILE .NOT. EOF
   @ 1,27 SAY stitle
   @ 1,60 SAY 'DATE'
   @ 1,65 SAY date()
   @ 3,6 SAY 'NAME'
   @ 3,30 SAY 'OFFICE'
   @ 3,50 SAY 'BE SCOUT RANK    DATE'
   @ 5,2 SAY TRIM(!(sc:last))+', '+TRIM(!(sc:first))
   @ 5,35 SAY !(t:office)
   @ 5,50 SAY t:mact
   STORE 5 TO row
   DO WHILE (.NOT. EOF) .AND. row  line
      @ row,60 SAY !(rank)
      @ row,65 SAY rank:date
      STORE row+1 TO row
      DO delay
      SKIP
   ENDDO
   IF .NOT. EOF
      npage
   ENDIF
ENDDO
DO delay
```

```
USE B:screcord
DELETE FILE B:temp
RETURN
```

```
S7ALL.CMD
*
*
*
STORE 0 TO bal
USE B:trfile
COPY FIELD prog:date,event,amt:due TO B:temp
SELECT PRIMARY
USE B:accfile
SELECT SECONDARY
USE B:temp
JOIN TO B:templ FOR prog:date=S.prog:date .AND. !(event) ;
= !(S.event) FIELD sc:last,sc:first,prog:date,att:code,;
amt:due,amt:paid,balance
USE B:templ
INDEX ON !(sc:last)+!(sc:first)+$(prog:date,7,2)+;
$(prog:date,1,2)+$(prog:date,4,2) TO B:tmindx
GOTO TOP
STORE balance TO bal
STORE $(prog:date,7,2) TO yr
STORE $(prog:date,1,2) TO mont
STORE sc:last TO t:last
STORE sc:first TO t:first
STORE 0 TO totdue
STORE 0 TO totpaid
DO WHILE .NOT. EOF
   @ 1,20 SAY stitle
   @ 1,60 SAY'DATE'
   @ 1,65 SAY DATE()
   STORE 2 TO row
   DO WHILE .NOT. EOF .AND. rowline
      DO moname
      @ row,20 SAY'MONTH '+!(mon)+"' "+yr
      @ row,47 SAY'PREVIOUS BALANCE '+STR(bal,9,2)
      STORE row+1 TO row
      @ row,7 SAY"NAME"
      @ row,33 SAY'DATE ATT. CODE AMT.DUE'
      @ row,57 SAY'AMT.PAID   BALANCE'
      STORE row+2 TO row
      IF row  line
         DO delay
         npage
         STORE 3 TO row
      ENDIF
      @ row,1 SAY TRIM(!(sc:last))+', '+TRIM(!(sc:first))
      DO WHILE .NOT. EOF .AND. rowline .AND. !(sc:last) ;
= !(t:last) .AND. !(sc:first) = !(t:first) .AND. ;
$(prog:date,7,2) = yr .AND. $(prog:date,1,2) = mont
```

```
         STORE yr TO yrs
         STORE mont TO monts
         STORE sc:last TO 1:name
         STORE sc:first TO f:name
         STORE totdue+amt:due TO totdue
         STORE totpaid+amt:paid TO totpaid
         STORE balance TO bal
         @ row,34 SAY $(prog:date,4,2)
         @ row,40 SAY !(att:code)
         @ row,47 SAY amt:due
         @ row,57 SAY amt:paid
         @ row,66 SAY balance
         STORE row+1 TO row
         IF row  line
            DO delay
            npage
            STORE 3 TO row
         ENDIF
         SKIP
         DO delay
      ENDDO
      STORE $(prog:date,7,2) TO yr
      STORE $(prog:date,1,2) TO mont
      STORE sc:last TO t:last
      STORE sc:first TO t:first
      STORE row + 1 to row
      IF row  line
         DO delay
         npage
         STORE 3 TO row
      ENDIF
      @ row,20 SAY "TOTAL"
      @ row,44 SAY totdue
      @ row,54 SAY totpaid
      @ row,66 SAY bal
      STORE balance TO bal
      DO delay
      STORE row+2 TO row
      IF !(1:name)  !(t:last) .OR. !(f:name)  ;
!(t:first) .OR. yr  yrs .OR. mont  monts
         STORE 0 TO totdue
         STORE 0 TO totpaid
      ENDIF
      LOOP
   ENDDO
   DO delay
   npage
ENDDO
DO delay
IF row  line
```

```
   STORE 3 TO row
ENDIF
USE B:trfile
DELETE FILE B:templ
RETURN
```

```
S7ONE.CMD
*
*
*
STORE 0 TO bal
USE B:account
COPY TO B:temp FOR !(sc:last)=!(l:name) .AND. !(sc:first) ;
= !(f:name)
USE B:trprog
COPY FIELD prog:date,event,amt:due TO B:templ
SELECT PRIMARY
USE B:temp
SELECT SECONDARY
USE B:templ
JOIN TO B:temp2 FOR prog:date = S.prog:date .AND. !(event);
= !(S.event) FIELD sc:last,sc:first,prog:date,;
att:code,amt:due,amt:paid,balance
USE B:temp2
INDEX ON $(prog:date,7,2)+$(prog:date,1,2)+;
$(prog:date,4,2) TO B:tmindx
GOTO TOP
STORE $(prog:date,1,2) TO mont
STORE $(prog:date,7,2) TO yr
STORE balance TO bal
DO WHILE .NOT. EOF
   @ 1,20 SAY stitle
   @ 1,60 SAY 'DATE'
   @ 1,65 SAY DATE()
   STORE 2 TO row
   DO WHILE .NOT. EOF .AND. rowline
      STORE 0 TO totdue
      STORE 0 TO totpaid
      DO moname
      @ row,20 SAY 'MONTH '+!(mon)+', '+yr
      @ row,47 SAY 'PREVIOUS BALANCE '+STR(bal,9,2)
      STORE row+1 TO row
      @ row,7 SAY 'NAME'
      @ row,33 SAY 'DATE ATT. CODE AMT. DUE'
      @ row,57 SAY 'AMT. PAID   BALANCE'
      STORE row+2 TO row
      @ row,1 SAY TRIM(!(sc:last))+', '+TRIM(!(sc:first))
      DO    WHILE    .NOT.    EOF    .AND.    rowline    .AND.
$(prog:date,7,2) = yr .AND. $(prog:date,1,2)=mont
         STORE totdue+amt:due TO totdue
         STORE totpaid+amt:paid TO totpaid
         STORE balance TO bal
         @ row,34 SAY $(prog:date,4,2)
         @ row,40 SAY !(att:code)
         @ row,47 SAY amt:due
         @ row,57 SAY amt:paid
         @ row,66 SAY balance
         STORE row+1 TO row
         DO delay
         SKIP
      ENDDO
      STORE row+1 TO row
      IF row  line
         npage
         STORE 3 TO row
      ENDIF
      STORE $(prog:date,7,2) TO yr
      STORE $(prog:date,1,2) TO mont
      @ row,20 SAY'TOTAL'
      @ row,44 SAY totdue
      @ row,54 SAY totpaid
      @ row,65 SAY bal
      STORE row+2 TO row
      DO delay
      LOOP
   ENDDO
   DO delay
ENDDO
DO delay
USE B:trprog
DELETE FILE B:temp2
RETURN
```

```
S8ALL.CMD
*
*
*
USE B:accfile
COPY FIELD sc:last,sc:first,prog:date,event TO B:temp FOR ;
!(att:code)'N'
SELECT PRIMARY
USE B:temp
SELECT SECONDARY
USE B:trfile
JOIN TO B:templ FOR prog:date = S.prog:date .AND. !(event);
= !(S.event) FIELD sc:last,sc:first,prog:date,event,no:day
USE B:templ
INDEX ON $(prog:date,7,2)+$(prog:date,1,2)+;
$(prog:date,4,2)+!(sc:last)+!(sc:first)+!(event) TO ;
B:tmindx
GOTO TOP
STORE sc:last TO t:last
STORE sc:first TO t:first
STORE $(prog:date,7,2) TO yr
STORE $(prog:date,1,2) TO mont
STORE 0 TO totday
STORE event TO act
DO WHILE .NOT. EOF
   @ 1,20 SAY stitle
   @ 1,60 SAY'DATE'
   @ 1,65 SAY DATE()
   STORE 3 TO row
   DO WHILE .NOT. EOF .AND. row  line
      STORE 0 TO totday
      DO moname
      @ row,20 SAY'MONTH '+!(mon)+"' "+yr
      STORE row+1 TO row
      @ ROW,11 SAY'NAME'
      @ row,30 SAY'DATE'
      @ row,42 SAY'ACTIVITIES'
      @ row,69 SAY'NO.DAYS'
      STORE row + 1 TO row
      DO WHILE .NOT. EOF .AND. row  line .AND. ;
$(prog:date,7,2) = yr .AND.$(prog:date,1,2) = mont
         @ row,3 SAY TRIM(!(sc:last))+', '+;
TRIM(!(sc:first))
         DO WHILE .NOT. EOF .AND. row  line .AND. ;
$(prog:date,7,2) = yr .AND. $(prog:date,1,2) = mont .AND. ;
!(sc:last) = !(t:last) .AND. !(sc:first) = !(t:first)
            DO WHILE .NOT. EOF .AND. row  line .AND. ;
$(prog:date,7,2) = yr .AND. $(prog:date,1,2) = mont .AND. ;
```

```
!(sc:last)  = !(t:last) .AND. !(sc:first) = !(t:first) .AND.
!(event) = !(act)
               STORE sc:last TO l:name
               STORE sc:first TO f:name
               STORE mont TO monts
               STORE yr TO yrs
               STORE event TO t:act
               STORE totday + no:day TO totday
               @ row,30 SAY $(prog:date,4,2)
               @ row,42 SAY TRIM(!(event))
               @ row,72 SAY no:day
               STORE row+1 TO row
               IF rowline
                  DO delay
                  npage
                  STORE 3 TO row
               ENDIF
               DO delay
               SKIP
            ENDDO
            STORE row + 1 TO row
            IF row  line
               npage
               STORE 3 TO row
            ENDIF
            @ row,61 SAY'TOTAL    '+STR(totday,4,0)
            IF !(t:last)  !(l:name) .OR. !(t:first)  ;
!(f:name) .OR.!(event)  !(t:act) .OR. yr  yrs .OR. ;
mont  monts
               STORE 0 TO totday
            ENDIF
            STORE event TO act
            STORE row+1 TO row
            DO delay
            IF row line
               DO delay
               npage
               STORE 3 TO row
            ENDIF
         ENDDO
         STORE sc:last TO t:last
         STORE sc:first TO t:first
      ENDDO
      STORE $(prog:date,7,2) TO yr
      STORE $(prog:date,1,2) TO mont
   ENDDO
   DO delay
   npage
ENDDO
DO delay
```

```
USE B:trfile
DELETE FILE B:templ
RETURN
```

```
S8ONE.CMD
*
*
*
USE B:account
COPY FIELD sc:last,sc:first,prog:date,event TO B:temp FOR ;
!(sc:last) = !(l:name) .AND. !(sc:first) = !(f:name) .AND.;
!(att:code)  'N'
SELECT PRIMARY
USE B:temp
SELECT SECONDARY
USE B:trprog
JOIN TO B:templ FOR prog:date = S.prog:date .AND. !(event);
= !(S.event) FIELD sc:last,sc:first,prog:date,event,no:day
USE B:templ
INDEX ON $(prog:date,7,2)+$(prog:date,1,2)+;
$(prog:date,4,2)+!(event) TO B:tmindx
GOTO TOP
STORE $(prog:date,7,2) TO yr
STORE $(prog:date,1,2) TO mont
STORE event TO act
DO WHILE .NOT. EOF
   @ 1,20 SAY stitle
   @ 1,60 SAY'DATE'
   @ 1,65 SAY DATE()
   STORE 3 TO row
   DO WHILE .NOT. EOF .AND. row  line
      DO moname
      @ row,20 SAY'MONTH '+!(mon)+"' "+yr
      STORE row+1 TO row
      @ row,11 SAY'NAME'
      @ row,34 SAY'DATE'
      @ row,42 SAY'ACTIVITIES'
      @ row,69 SAY'NO.DAYS'
      STORE row + 1 TO row
      DO WHILE .NOT. EOF .AND. row  line .AND. ;
$(prog:date,7,2) = yr .AND. $(prog:date,1,2) = mont
         STORE 0 TO totday
         @ row,1 SAY TRIM(!(sc:last))+', '+;
TRIM(!(sc:first))
         DO WHILE .NOT. EOF .AND. rowline .AND. ;
$(prog:date,7,2) = yr .AND. $(prog:date,1,2) = mont ;
.AND. !(event) = !(act)
            STORE totday + no:day TO totday
            @ row,35 SAY $(prog:date,4,2)
            @ row,42 SAY TRIM(!(event))
            @ row,72 SAY no:day
            STORE row + 1 TO row
            DO delay
            SKIP
         ENDDO
         STORE row+1 TO row
         @ row,61 SAY'TOTAL    '+STR(totday,4,0)
         STORE row+2 TO row
         STORE $(prog:date,7,2) TO yr
         STORE $(prog:date,1,2) TO mont
         STORE event TO act
         DO delay
      ENDDO
   ENDDO
   DO delay
ENDDO
USE B:trprog
DELETE FILE B:templ
RETURN
```

```
MISC.CMD
*
*
*
DO setup
DO while t
   ERASE
   STORE ' ' TO mis:sel
   @ 3,22 SAY'MISCELLANEOUS'
   @ 5,15 SAY'0 - EXIT'
   @ 6,15 SAY'1 - RECOVER THE LOST FILES'
   @ 7,15 SAY'2 - DELETE OLD RECORDS OF TROOP PROGRAM ;
SCHEDULE'
   @ 9,10 SAY'ENTER YOUR SELECTION' GET mis:sel
   READ
   DO CASE
      CASE mis:sel = '0'
         RELEASE ALL
         RETURN
      CASE mis:sel = '1'
         DO recover
      CASE mis:sel = '2'
         DO tdell
      OTHERWISE
         @ 23,10 SAY'WRONG CODE - TRY AGAIN'
         DO delay
         LOOP
   ENDCASE
ENDDO
```

```
RECOVER.CMD
*
*
*
DO setup
?
?
?'     The program is copying the backup files.'
?
?'           Press RETURN when ready'
WAIT
USE B:btrprog
COPY TO B:trprog
USE B:badlead
COPY TO B:adlead
USE B:bscrecord
COPY TO B:screcord
USE B:bmember
COPY TO B:member
USE B:bleader
COPY TO B:leader
USE B:baward
COPY TO B:award
USE B:baccount
COPY TO B:account
USE B:badvance
COPY TO B:advance
USE B:bguardian
COPY TO B:guardian
?
?'     The files have been copied from the backup files'
?
DO delay
DO delay
RETURN
```

```
TDEL1.CMD
*
*
*
DO setup
RESTORE FROM B:memfile
STORE 'T' TO first
STORE 'T' TO aa
DO WHILE aa = 'T'
   ERASE
   STORE 0 TO yr
   STORE '        ' TO t:date
   STORE '                         ' TO t:event
   STORE '               ' TO t:loc
   STORE 0 TO tamtdue
   STORE 0 TO tnoday
   STORE '          ' TO mon
   STORE '  ' TO mont
   STORE '  ' TO yr:pro
   STORE '  ' TO mo:pro
   STORE ' ' TO period
   STORE ' ' TO confirm
   USE B:trprog
   COUNT TO no:rec
   IF no:rec = 0
      @ 23,10 SAY'I am sorry this file is empty now'
      DO delay
      RELEASE ALL
      RETURN
   ENDIF
   @ 1,10 SAY'DELETE ALL PROGRAMS IN A PERIOD OF TIME'
   @ 1,60 SAY'DATE '
   @ 1,65 SAY DATE()
   @ 6,10 SAY'Which periods of time do you want to delete '
   @ 6,54 SAY'the troop program?'
   @ 7,10 SAY'(Month - M or Year - Y or RETURN - To  ;
stop)' GET period
   READ
   IF !(period) = ' '
      IF first = 'T'
         RELEASE ALL
         RETURN
      ENDIF
      STORE 'F' TO aa
      LOOP
   ENDIF
   IF !(period) = 'M'
      @ 9,10 SAY'What month do you want? (01 to 12)' GET ;
```

```
mo:pro
      @ 10,10 SAY'And in what year?' GET yr:pro
      READ
      STORE 1900 + VAL(yr:pro) TO yr
      STORE mo:pro TO mont
      DO moname
      COUNT TO no:rec FOR $(prog:date,1,2) = mo:pro .AND. ;
$(prog:date,7,2) = yr:pro
      IF no:rec = 0
         @ 23,10 SAY'There are no programs in that month'
         DO delay
         LOOP
      ENDIF
      @ 12,10 SAY'Please confirm that you wish to delete ;
all the '
      @ 12,57 SAY'troop program'
      @ 13,13 SAY'in the month of '+TRIM(!(mon))+','+;
STR(yr,4,0)+'? (Y or N)' GET confirm
      READ
      IF !(confirm) = 'N'
         LOOP
      ENDIF
      COPY TO B:temp FOR $(prog:date,1,2) = mo:pro .AND. ;
$(prog:date,7,2) = yr:pro
      USE B:trbakup
      APPEND FROM B:temp
      IF mo:pro = $(pre:date,1,2) .AND. yr:pro = ;
$(pre:date,7,2)
         USE B:account
         COPY TO B:templ FOR $(prog:date,1,2) = mo:pro ;
.AND. $(prog:date,7,2) = yr:pro
         USE B:accbak
         APPEND FROM B:templ
         USE B:account
         DELETE FILE B:templ
         DELETE ALL FOR $(prog:date,7,2) = yr:pro .AND. ;
$(prog:date,1,2) = mo:pro
         PACK
         STORE '1' TO flag
      ENDIF
      USE B:trprog
      DELETE FILE B:temp
      DELETE ALL FOR $(prog:date,1,2) = mo:pro .AND. ;
$(prog:date,7,2) = yr:pro
      PACK
      STORE 'F' TO first
   ENDIF (* PERIOD = 'M' *)
   IF !(period) = 'Y'
      @ 9,10 SAY'What year do you want to delete ' GET ;
yr:pro
```

```
      READ
      STORE 1900 + VAL(yr:pro) TO yr
      COUNT TO no:rec FOR $(prog:date,7,2) = yr:pro
      IF no:rec = 0
         @ 23,10 SAY'There are no programs in that year'
         DO delay
         LOOP
      ENDIF
      @ 11,10 SAY'Please confirm that you wish to delete ;
all '
      @ 11,54 SAY'the troop programs'
      @ 12,13 SAY' of the year '+STR(yr,4,0)+'? (Y OR N)';
GET confirm
      READ
      IF !(confirm) = 'N'
         LOOP
      ENDIF
      COPY TO B:temp FOR $(prog:date,7,2) = yr:pro
      USE B:trbakup
      APPEND FROM B:temp
      IF yr:pro = $(pre:date,7,2)
         USE B:account
         COPY TO B:templ FOR $(prog:date,7,2) = yr:pro
         USE B:accbak
         APPEND FROM B:templ
         USE B:account
         DELETE FILE B:templ
         DELETE ALL FOR $(prog:date,7,2) = yr:pro
         PACK
         STORE '1' TO flag
      ENDIF
      USE B:trprog
      DELETE FILE B:temp
      DELETE ALL FOR $(prog:date,7,2) = yr:pro
      PACK
      STORE 'F' TO first
   ENDIF (* PERIOD = 'Y' *)
ENDDO
ERASE
@ 5,10 SAY'*****************************************;
*********'
@ 6,10 SAY'*                                        ;
        *'
@ 7,10 SAY'*      PREPARE THE PRINTER READY TO PRINT;
        *'
@ 8,10 SAY'*      THE RECORD OF CANCELLED SCHEDULE  ;
        *'
@ 9,10 SAY'*                                        ;
        *'
@ 10,10 SAY'*       WHEN YOU ARE READY PRESS RETURN;
```

```
        *'
@ 11,10 SAY'*                                        ;
        *'
@ 12,10 SAY'*****************************************;
*********'
WAIT
SET FORMAT TO PRINT
SET PRINT ON
SET CONSOLE OFF
STORE 55 TO line
STORE 'EJECT' TO npage
ERASE
STORE 'trbakup' TO trfile
DO tllistl
IF flag = '1'
   STORE 'accbak' TO accfile
   STORE 'ATTENDANCE AND DUES' TO stitle
   DO s7all
   STORE 'OUR CAMPING LOG' TO stitle
   DO s8all
ENDIF
SET FORMAT TO SCREEN
SET PRINT OFF
SET CONSOLE ON
ERASE
@ 10,15 SAY'THE REPORT ON DELETING TROOP SCHEDULE HAS ;
FINISHED'
DO delay
USE B:trbakup
DELETE ALL
PACK
USE B:accbak
DELETE ALL
PACK
RELEASE ALL
RETURN
```

# LIST OF FOOTNOTES

[1] Carl Townsend, Using dBASE II (Berkeley, California: Osborne/McGraw-hill, 1984), p.11 ... 4

[2] Manual of Boy Scouts of America (North Brunswick, New Jersey, 1978) .................... 18

# BIBLIOGRAPHIES

Martin, James. Computer Data-Base Organization 2nd Edition. Englewood Cliffs, New Jersey: Prentice-Hall, Inc., 1977

Martin, James. Managing The Data-Base Environment. Englewood Cliffs, New Jersey: Prentice-Hall, Inc., 1983

Atres, S. Data-Base: Structured Techniques For Design, Performance, And Management. A Wiley-Interscience Publication, John Wiley Sons, 1980

Townsend, Carl. Using dBASE II. Berkeley, California: Osborne/Mcgraw-Hill, 1984

Wayne, Ratliff. dBASE II: User-Manual. Culver city, California, 1981