WORKING WITH GWBASIC

When you have successfully loaded GWBASIC, the OK prompt will appear on the VDU. This prompt informs you that GWBASIC is ready to accept its own commands. You can now use GWBASIC in the immediate or the program mode. In the immediate mode, the computer will immediately perform any BASIC instruction you type in after you press <***ENTER***>.

In this and the following sections, you are introduce to the BASIC language by way of hands on exercises that involve the execution of BASIC instruction in the immediate mode and then in the program mode. You will be working with GWBASIC without much emphasis on this particular dialect of the BASIC language. Note that the immediate mode is also referred to as the direct mode. In the program mode, you will design program logic using flowcharts. You will then use the flowchart to write the BASIC program.

LOGICAL OPERATORS

Consider the statement: "THIS MANUAL IS WRITTEN IN FRENCH". This expression is called an assertion. We assign the truth-value TRUE or FALSE to assertions. On a digital computer, we use the bits ("1" and "0") to represent these truth-values. They are also called Logical values. The assertion at the start of this paragraph is assigned the truth-value FALSE. This truth-value can be represented on the computer system by a 0. Suppose we now negate the expression by saying: "THIS BOOK IS NOT WRITTEN IN FRENCH". We have change the truth-value of the original expression by inserting a NOT in the original statement. The NOT logical operator changes the truth-value of a logical variable. Now consider the following arithmetic assertions:
9 > 54, -85 < 42, 2+5 = 5 NOT 4<>9 and NOT 9> 54
Evaluate the truth-value for each of these expressions and then use the PRINT instruction to verify your evaluation.

Note that on some computer systems a 0 is used to represent the truth-value FALSE and a -1 is used to represent the truth-value TRUE. Also note that no computer system will dispute the truth-value assigned to an arithmetic expression. Use the PRINT instruction to evaluate the set of logical expression listed at the end of the previous paragraph. Try to predict the outcome before the instruction is actually executed.

We consider the expression: "I AM NOW READING CHAPTER ONE". We can assign the truth-value TRUE (-1 or 1) to this statement and we can combine this statement with the previous statement to get the compound statement: "THIS BOOK IS WRITTEN IN FRENCH AND I AM NOW READING CHAPTER ONE".

We now have a single statement, made up of two statements combine by the word AND. The truth-value of this compound expression is 0. This is because one of the original expressions has a truth-value of 0. Generally, we can say that when two statements are combine with the logical AND operator, the

compound statement that is formed will have a truth value of 0 if at least one of the original statement have a truth value of 0.

Use PRINT instructions to evaluate the following expressions. Try to predict the logical value before each of them is actually executed by the computer system

```
PRINT 9>54 AND 2+5 = 5
```

```
PRINT NOT (9 > 54) AND (-85 < 42)
```

```
PRINT (3+4 <> 7) AND (3+4 = 7)
```

```
PRINT 9 > 54 OR -85 < 42
```

Now consider the following expression: "THIS BOOK IS WRITTEN IN FRENCH" OR "I AM NOW READING CHAPTER ONE". This single expression has a truth value TRUE. In general, when two statements or expressions are combine with the logical OR operator, the result has a truth value of TRUE if at least one of the statement or expression has the truth value TRUE.

Type in the following expression and try to predict their results before the system actually execute each of them.

```
(a) PRINT NOT (0 OR -1)
(b) PRINT 9 > 54 OR 5 = 2+5
(c) PRINT NOT (9 > 54) OR -85 < 42
(d) PRINT -1 OR (3+4 <> -7)
(e) PRINT (-90<20) AND (3+4-7) OR (3+4 <> 7)
```

Now verify what we have been saying about the two statements:
(1) THIS BOOK IS WRITTEN IN FRENCH
(2) I AM NOW READING A PART OF CHAPTER ONE.
By using a variable to represent each statement, first assign a truth value to each statement with the LET instruction and then combine them with the logical AND and the logical OR operators as in the expressions above.

WRITING A BASIC PROGRAM

In the following exercises, a < ***ENTER***> written at the end of a line mean that you should press the ENTER (or RETURN) key when you have finish typing the line. Type in the following code of instructions:

```
10 REM    MY FIRST PROGRAM  < ENTER>
20 PRINT  "THIS IS MY FIRST PROGRAM"  < ENTER>
30 LET N = 0
40 IF N > 8 THEN 80 < ENTER>
50    LET N = N + 1    < ENTER>
60    PRINT N  < ENTER>
70 GOTO 40  < ENTER>
80 END < ENTER>
```

You will notice that noting happens when you type a line of instruction and then press < ***ENTER***>. The BASIC interpreter will not execute a command that is preceded by a line number when you press the RETURN key. Instruction written with a preceding line number is written in the program mode. When you type a line number before a command statement, the BASIC interpreter assumes that you are writing a sequence of instructions that constitute a program. A program is a sequence of commands designed to solve a specific problem on a computer system. The BASIC interpreter will not execute the instructions in a program until it receives the RUN command. Now type the following:

```
RUN < ENTER>
```

You will also notice that each proceeding line number in the program is incremented by 10. This allows easy program maintenance. This is further explained in the next chapter. The instructions in a program usually follow a logical sequence. The logical sequence in the above program is given by the following flowchart.

THE PROGRAM LOGIC

The logic to be used in the solution of this programming problem is the same as that used for the sample program in this chapter. If you did receive a tutorial copy of Logic Coder with this exercise manual, then a sample copy of the flowchart used by the sample program is on your installation disk. Run Logic Coder and then load the sample flowchart with file name "**Tele.flw**". You will edit the source code text of this flowchart to generate the required source program as given in the problem specification. You do not need to edit the algorithmic view of the flowchart, as the step by step description remains the same as with the sample program. Set the start line number for the program to be generated by selecting the Settings option on the main menu bar. Once you have edited the algorithmic text of the flowchart, you should save it with a new file name. You should ensure that the content of the original file is preserved.

Use LogicCoder to generate the required program. Open the file that contains the generated source program and edit it so that it list the data values in the table above. You should notice that LogicCoder does not create the listed data file for you whenever you use it to generate a source program. LogicCoder only uses the logic of the flowchart it is presented with along with the algorithmic text content to generate a source program. If you are not certain or do not know how to associate a description in the program flowchart with a BASIC instruction, then you should do the following exercises before you attempt to write the algorithmic text view as instructed above.

You will be using the flowchart to code the program problem solution by way of imitation.

CODING THE PROGRAM SOLUTION

Look back carefully on the example program logic design and the program code. Make careful observation of the line number in the program code where each point on the program flowchart is implemented. Choose three variable names to read the data values from each record in the input data file and then complete the table below.

VARIABLE NAME CHOSEN	FIELD FROM WHICH DATA WILL BE READ	VARIABLE TYPE
_____	_____	_____
_____	_____	_____
_____	_____	_____

Write the variables in the sequence in which they should be written (a) after the READ statement and (b) after the PRINT statement in the program code:

(a) 1._____, 2.._____, 3._____

(b) 1._____, 2.._____, 3._____

Write the program code starting with the program documentation in a manner similar to that of the sample program coding listed on page 17. Use a clean sheet of coding paper or a clean sheet of paper if coding sheets are not available. Write as neat as possible without committing any syntax error.

When you have finish writing the program code, use letters of the alphabet to label each point on your program flowchart where an intended instruction is to be implemented. Write the same letter beside each line number in the program code where the intended instruction is implemented. If you are experiencing doubt in implementing the program code, then look back on the example program in this chapter and imitate the coding as close as possible without committing any syntax or logical error. Hand in your program design and the associated code to the lab supervisor or your tutor when you are finish.

ADDITIONAL PROGRAMMING EXERCISES

1. A list of names, street address, cities, and postcode of customers is to be prepared. The input test data file is listed below. The printed report should have the name on the first line, the street address on the second line, and the city and postcode on the next line. Two blank lines should separate the output data from each record. The message "END OF ADDRESS LIST", is to be printed at the end of processing all records,

Table 2x-3 Test data to be used by the program

NAME	STREET ADDRESS	CITY	TELE. NUMBER	POSTCODE
ACE COMPANY	111 PINE WAY	LAMOIN	953 6693	LA29 6ZN
DAN BROTHERS	985 ARROW DRIVE	SEAVIEW	312 7998	SV5 21Q
YANK PC GAMES	773 SKY CLOSE	PARKVIEW	925 1119	PA35 10QD
GOGO FOODS	881 OAK ROAD	MONTHCLARE	063 5540	MC6 15A
MICRO ELECS	95 BAKERS ROAD	SILICON	323 8989	SI21 5CU

THE PROGRAM DESIGN

7. _____

The logic to be used in the program solution is illustrated in the flowchart below.

THE PROGRAM LOGIC DESIGN

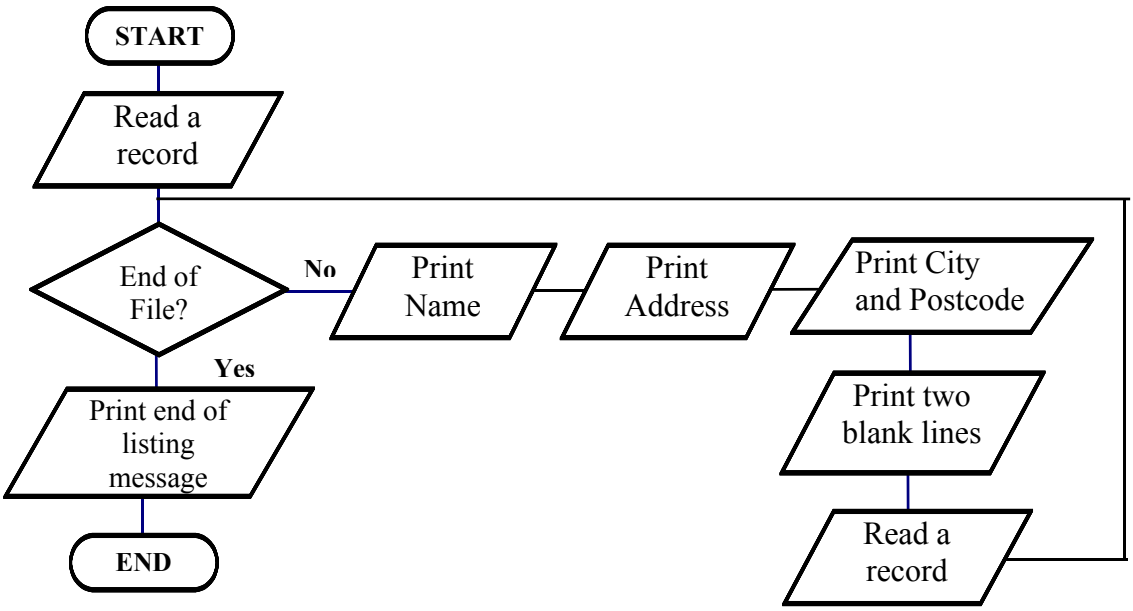


Figure 2x-7
Unlike previous programs, the processes performed in the main loop of the program logic does several print operation for each record that is processed.

Use the logic illustrated in the flowchart to code the program that will produce the required output.

Write down the name of the fields in each record of the input data file that are (a) string and that are (b) Numeric.

(a)	(b)
_____	_____
_____	_____
_____	_____
_____	_____

Write down the variable name that you intend to use in the READ instruction to read each record in the input data file. Write these variable names in the sequence in which they will be written in the READ instruction.

PROGRAMMING ASSIGNMENT #3

A traffic citation report is to be prepared. You are required to design and code a program that will produce the report.

INPUT DATA FILE

The input test data file consists of records. Each record contain the name of the person receiving the traffic citation, a code that indicate whether the citation is for moving (code M) or for non-moving (code N) violation, and the number of citation for that person over the pass three years. The input test data is given in the table below.

INPUT TEST DATA

NAME	TYPE OF VIOLATION	NUMBER OF VIOLATION
JUNIOR HAINES	N	2
TOM JULION	N	6
JUNE RHODEWS	M	4
DALE SMEARS	N	1
EVERET MILLS	N	5

OUTPUT

Each line on the output report list the name of the person receiving the traffic citation, a fine of 30.00 for moving violation or 10.00 for non-moving violation, a penalty of 20.00 if the number of violation is more than 3 and no penalty if the number of violation is less than or equal to 3. Each line on the report should also list the total fine due (fine + penalty). The total number of tickets, the total moving violation, the total non-moving violations, the total fines, the total penalties, and the total amount due are to be printed at the end of processing all records. The format for the output report is illustrated below.

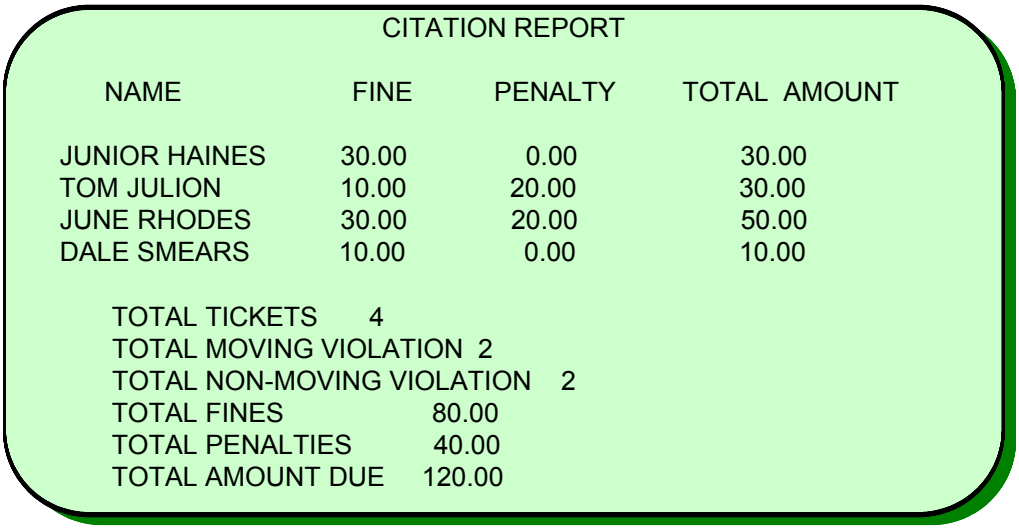
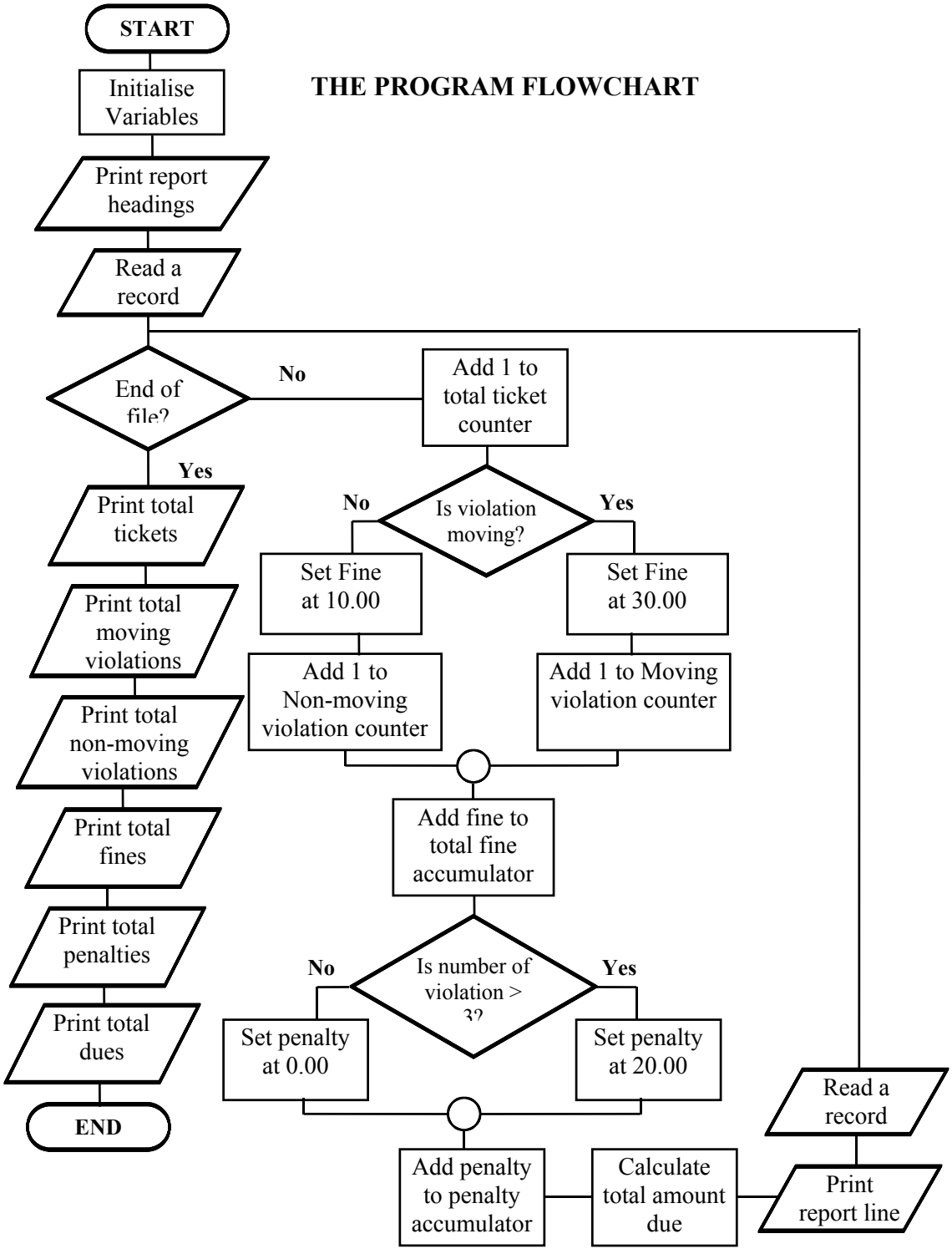


Figure 4x-9
Screen format for the program output.

PROGRAM LOGIC DESIGN

The logic to be used in the program that implements the program specifications given above is illustrated below. Use the logic in this flowchart to code the program that will produce the specified output report.

THE PROGRAM FLOWCHART



EXERCISES:

(1) Write down the variables that you use to read each record of the input data file. State the field from which each variable reads data.

_____

_____

_____

(2) It is now decided to set the penalty by multiplying the excess violation by 15.00. Write down the expression that will determine the penalty. Use the variable name that reads the number of violations in your formula.

_____

(3) Explain the change that is to be done to the second case structure in the loop of the program logic illustrated above in order to implement this new requirement in the program.

_____

_____

_____

_____

_____

(4) What line number(s) should now be removed from your previous program code? Show it to your class tutor.

_____

(5) Write down the instructions that are to replace the instructions at these line number(s).

_____

_____

_____

_____

_____

_____

ADDITIONAL PROGRAMMING EXERCISE

INSTRUCTIONS

A program is to be designed and then coded in BASIC that will prepare country club dues report. The Input and Output specifications are given below.

INPUT DATA

The input data file is made out of records. Each record contain the country club member's name, the type of membership, and the number of years the individual have been a member of the club. If the membership type field in the record contain an **F**, then the membership is a family type. If the membership type field contain an **I**, then the membership is an individual type. The table below lists the input test data.

TEST DATA

NAME	MEMBERSHIP TYPE	YEARS OF MEMBERSHIP
HARVEY HANLEY	F	9
WILMA LITT	F	7
EUGENE MITTER	F	2
WALLY PITT	I	6
EUNICE PONNIR	I	8

OUTPUT

The program should produce an output listing of the country club members, giving their name, the type of membership (Family or Individual), the number of years of membership to the club, and the country club dues. The dues are calculated as follows. If the member is a family member and have been a member for more than six years, the dues is 1,200.00. If the member is a family member and has been a member for six years or less, the due is 1,600.00. If a member is an Individual member and has been a member for more than six years, the club due is 800.00. If the member is an individual member and has been a member for six years or less, the dues is 1,100.00. Total for the number of members, the number of family members, the number of individual members, and the dues are to be printed at the end of the listing. The format for the output is illustrated below.

Figure 5X-4
Output screen to be created by the program.

THE PROGRAM DESIGN

We begin the program design by specifying the tasks to be accomplished by the program.

PROGRAM TASKS

1. **Read input records**
2. **Determine club membership dues.**
3. **Accumulate final totals.**
4. **Print output report line**
5. **Print final totals.**

The logic to be used in the solution of the programming problem is illustrated by the flowchart on the following page. Carefully examine the logic illustrated in this flowchart and then compare it to the logic in your modified flowchart of the previous program editing exercise.

THE PROGRAM LOGIC DESIGN

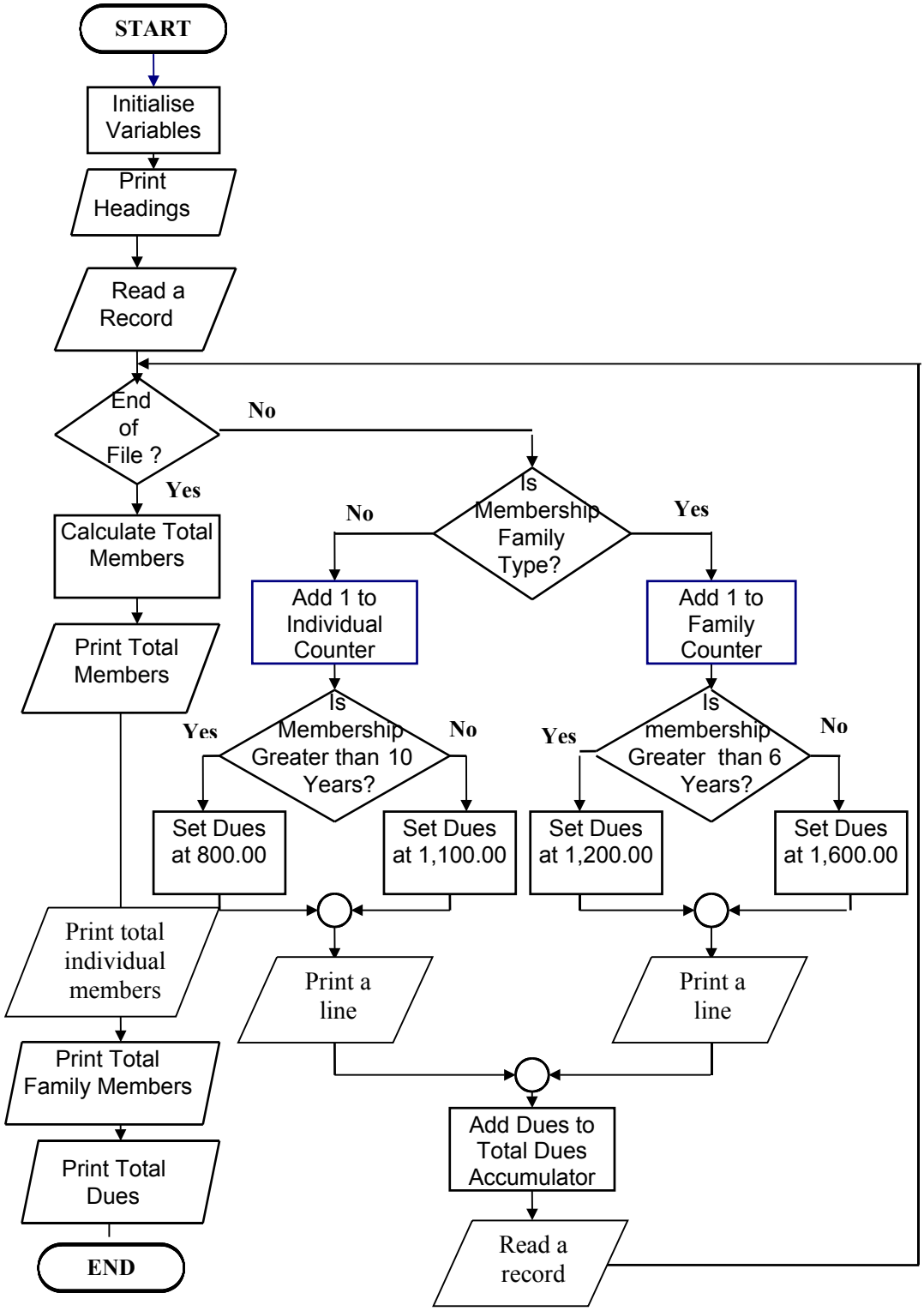


Figure 5x-5
Nested case structures are used in the body of the main processing loop to determine the due calculation for each record read from the data file.

THE PROGRAM CODE (Page 1 of 2)

```
100 REM  COUNTRY CLUB          MARCH , 97                    C.  EVELYN
110                                                                  REM
120 REM THIS PROGRAM PRODUCE A COUNTRY CLUB MEMBERSHIP DUES
130 REM REPORT. MEMBERSHIP DUES ARE BASE UPON THE MEMBERSHIP TYPE
140 REM AND THE YEARS OF MEMBERSHIP. FINAL TOTALS OF EACH
150 REM TYPE AND THE TOTAL FEES ARE PRINTED ON THE REPORT.
160                                                                  REM
170 REM   VARIABLE NAMES :
180 REM    N$ . . . . . MEMBER NAME
190 REM    M$ . . . .  MEMBERS TYPE CODE
200 REM    Y . . . . .  YEARS OF MEMBERSHIP
210 REM    Y1 . . . .  SIX YEARS CONSTANT
220 REM    D . . .  . . CLUB DUES
230 REM    M1. . . .  DUES FOR FAMILY MEMBER SIX YEARS OR MORE
240 REM    M2 . . . .  DUES FOR FAMILY MEMBERS LESS THAN SIX YEARS
250 REM    M3 . . . .  .USE FOR INDIVIDUALS SIX OR MORE YEARS
260 REM    M4 . . .  DUES FOR INDIVIDUALS  LESS THAN SIX YEARS
270 REM    F$ . . . . . FAMILY MEMBER CONSTANT
280 REM    I$ . . . . INDIVIDUAL MEMBER CONSTANT
290 REM    T1. . . .  . . TOTAL INDIVIDUAL MEMBERS
300 REM    T2 . . . . TOTAL FAMILY MEMBERS
310 REM    T3 . . . . TOTAL MEMBERSHIP DUES
320 REM    F1$ . . . PRINT USING FORMAT FOR DETAIL LINE
330 REM    F2$ . . . PRINT USING FORMAT FOR T1
340 REM    F3$ . . . PRINT USING FORMAT FOR T2
350 REM    F4$ . . . PRINT USING FORMAT FOR T - TOTAL NUMBER OF MEMBERS
360 REM    F5$ . . . PRINT USING FORMAT FOR T3
370                                                                  REM
380 REM       *********      INITIALISATION OF VARIABLES      *********
390                                                                   REM
400 LET Y1 = 6
410 LET M1 = 800 .00
420 LET  M2 = 1100.00
430 LET M3 = 1200.00
440 LET M4 = 1600.00
450 LET F$  = "FAMILY"
460 LET I$  = "INDIVIDUAL"
470 LET T1 = 0
480 LET T2 = 0
490 LET T3 = 0
 500 LET F1$  = "\                        \      \                  \        ##             ££,###.##"
510 LET F2$  =  "NUMBER OF MEMBERS       ###"
520 LET F3$  = "NUMBER OF INDIVIDUALS     ##"
530 LET F4$  = "NUMBER OF FAMILY      ###"
540 LET F5$  = "TOTAL DUES:   ##,###.## "
550                                                                  REM
560 REM  ***********    PROCESSING     *************
570                                                                   REM
580 PRINT TAB(12) "COUNTRY CLUB DUES REPORT"
590 PINT "  "
600 PRINT "          NAME          TYPE          YEARS          DUES "
```

```
610 PRINT "    "
620                                                              REM
630 READ N$,  M$,  Y
640                                                              REM
650  IF N$ = "END OF FILE"  THEN 920
660      IF N$ = "F"  THEN
670           LET T1 = T1 + 1
680           IF Y > Y1  THEN 720
690               LET D  =  M2
700               GOTO 750
710                                                              REM
720                LET D = M1
730                GOTO 750
740                                                             REM
750           PRINT USING F1$; N$, I$, Y, D
760           GOTO
770                                                             REM
780           LET T2 = T2 + 1
790           IF Y >  Y1  THEN
800             LET D = M3
810             GOTO 860
820                                                             REM
830             LET D = M4
840             GOTO 860
850                                                             REM
860           PRINT USING F1$; N$,  F$,  Y,  D
870           GOTO 890
880                                                             REM
890      LET T3 = T3 + D
900       READ N$, M$, Y
910 GOTO
920                                                             REM
930 LET T = T1 + T2
940 PRINT "   "
950 PRINT USING F2$; T
960 PRINT USING F3$; T1
970 PRINT USING F4$; T2
980 PRINT USING F5$; T3
990 END
```

Figure 5x-6
The BASIC program that implements the logic illustrated by the flowchart in figure 5x-5.