PROGRAMMER'S NOTEBOOK

By Jim Keogh

Keeping Score

THERE are many times when you want your computer to keep track of one or more events. The most common occasion is probably when you want to keep score during a computer game. In this month's column, we will discuss how you can use your microcomputer to keep score during a game or count events.

For a computer to keep score, the programmer must include, in the program, one or more registers. These are similar in function to the cash registers used in retail stores. In the store, every transaction is recorded. For instance, the clerk reads the price tag on the merchandise and records the amount on the cash register.

The register then adds the amount of the transaction to the previous transactions. Many times there are two or more registers (counters) within the equipment. These include registers for taxes and departments. At the end of the day, the totals for each register are printed out on the register tape. The equipment is then reset to zero and is ready to record the next day's transactions.

How a score appears on screen.

PROGRAMS FOR KEEPING SCORE OR COUNTING EVENTS

TRS-80

```
5 REM CLEAR SCREEN
10 CLS
15 REM SET REGISTERS TO ZERO
20 C = 0
30 X = 0
35 REM RANDOM NUMBER
   GENERATOR
40 A = RND (5)
45 REM REGISTER STATEMENT
50 X = X + 1
55 REM GAME INSTRUCTIONS
60 PRINT @ 400, "GUESS WHAT
   NUMBER I PICKED (1 TO 5)?"
70 REM INPUT STATEMENT
80 INPUT "                    "; F
85 REM DISPLAY OF REGISTER
   COUNTER
90 PRINT @ 832, "TIMES
   GUESSED";X
95 REM COMPUTER DECIDES IF AN-
   SWER IS CORRECT
100 IF F = A THEN GOTO 160 ELSE
    110
105 REM COMPUTER TELLS IF AN-
    SWER IS WRONG
110 PRINT @ 600, "WRONG ANSWER"
115 REM TIMING LOOP
120 FOR EE = 1 TO 75
130 DD = EE + 1
140 NEXT EE
145 REM LOOP TO BEGINNING OF AN-
    OTHER CHANCE
150 GOTO 40
155 REM COMPUTER TELLS IF AN-
    SWER IS RIGHT
160 PRINT @ 600, "RIGHT"
165 REM REGISTER STATEMENT
170 C = C + 1
175 REM COMPUTER TELLS
    OPERATOR HIS OR HER SCORE
180 PRINT @ 870, "YOUR SCORE IS";C
185 REM TIMING LOOP
190 FOR EE = 1 TO 75
200 SS = EE + 1
210 NEXT EE
215 REM LOOP TO BEGINNING OF AN-
    OTHER CHANCE
220 GOTO 40
```

APPLE II

```
5 REM CLEAR SCREEN
10 CALL − 936
15 REM SET REGISTERS TO ZERO
20 C = 0
30 X = 0
35 REM RANDOM NUMBER
   GENERATOR
40 A = RND (5)
45 REM REGISTER STATEMENT
50 X = X + 1
55 REM GAME INSTRUCTIONS
60 HTAB (15)
61 VTAB (20)
63 PRINT "GUESS WHAT NUMBER I
   PICKED (1 TO 5)?"
70 REM INPUT STATEMENT
80 INPUT "                    ";F
85 REM DISPLAY OF REGISTER
   COUNTER
90 HTAB (5)
91 VTAB (35)
92 PRINT "TIMES GUESSED";X
95 REM COMPUTER DECIDES IF AN-
   SWER IS CORRECT
100 IF F = A THEN GOTO 160 ELSE
    110
105 REM COMPUTER TELLS IF AN-
    SWER IS WRONG
110 HTAB (15)
111 VTAB (25)
112 PRINT "WRONG ANSWER"
115 REM TIMING LOOP
120 FOR EE = 1 TO 150
130 DD = EE + 1
140 NEXT EE
145 REM LOOP TO BEGINNING OF AN-
    OTHER CHANCE
150 GOTO 40
155 REM COMPUTER TELLS IF AN-
    SWER IS RIGHT
160 HTAB (15)
161 VTAB (25)
162 PRINT "RIGHT"
165 REM REGISTER STATEMENT
170 C = C + 1
175 REM COMPUTER TELLS
    OPERATOR HIS OR HER SCORE
180 HTAB (20)
181 VTAB (35)
182 PRINT "YOUR SCORE IS";C
185 REM TIMING LOOP
190 FOR EE = 1 TO 150
200 SS = EE + 1
210 NEXT EE
215 REM LOOP TO BEGINNING OF AN-
    OTHER CHANCE
220 GOTO 40
```


The register in our computer program operates in a similar manner. First, the registers are set to zero. This must be done at the beginning of the program. For example, we will call the register A. At the start of the program, we use the statement "A = 0." The program will then be executed. Before the program is concluded or looped to its beginning again, the register (counter) statement must be inserted into the program. This is "A = A + 1." (Read as "A equals A, incremented by one.")

The program can then continue and be looped to the line that immediately follows the statement "A = 0." When the program once again reaches the line with the statement "A = A + 1," the computer will add 1 to the value of A. Since this is the second time the program was executed, the value of A is now 1. The new value of "A = A + 1" is 2. This statement has just recorded that the program was executed twice.

To find the final tally, we must insert the statement "PRINT A." This is usually the last statement in the program loop. Briefly, the computer executes the program and adds the number of times the program is executed to the previous total in the register. At the end of the program, the computer prints the final tally of each register.

The following set of commands illustrates a program register:

```
10 A = 0
20 REM BODY OF PROGRAM IS
   EXECUTED
30 A = A + 1
40 PRINT A
50 GOTO 20
```

To illustrate the application of this subroutine, the following guessing game was developed. It is rather simple, using two registers. The first register counts the number of right guesses while the second counts the total number of guesses, whether right or wrong.

The program also includes a random number generator. The computer will generate a random number from one to five without displaying the number. It is up to the operator to guess which number the computer picked. If the guess is correct, the computer will print the word "RIGHT" and score the answer as being correct. Both registers will increase by one. If the guess is wrong, the computer will print "WRONG ANSWER" and increase the second register count by one.

By reviewing the details of the program, you should be able to develop a thorough understanding of how computer registers within a program operate. You can have as many registers as you require by just using a few command statements and let the computer do all the counting for you. ◇