program

TRY THIS ON YOUR HOME COMPUTER

BY CARMINE PRESTIA

Balancing the family checkbook, keeping track of the monthly budget, easy fare for the home computer.

"What are you going to do with it?" was the first question my wife asked when I said, "I'm going to get a home computer." After my new Heathkit H-8 computer system arrived, I found out it might not be such a bad question.

Fortunately, answers are readily available if you just take a few minutes to think about it and use a little imagination. Playing games with a computer is one of the most popular and common uses around. However, to most people that really isn't a practical use, and I was going to have a lot of convincing to do!

One of the first ideas to come to mind was the family checkbook. In our case there are monthly statements to reconcile that require tallying the outstanding checks and deposits, computing a final balance and comparing it with the balance in my checkbook.

To make the chore easier and to put the new Heath H-8 to work, I designed a *program* that allows me to enter the data and then has the H-8 do all the computations and comparisons.

A program is a sequential set of instructions that tell the computer what to do. The checkbook program is written in "Extended Benton Harbor BASIC," a Heathkit version of the original Dartmouth BASIC computer language. Although very easy to use, BASIC (Beginner's All-Purpose Symbolic Instruction Code) is also extremely powerful, making program writing much simpler than with the more complex computer languages.

The program, called Checkbook, is stored on magnetic cassette tape and takes less than a minute to *load* into the H-8 from a regular cassette tape recorder/player, also purchased from Heathkit.

To start the program I enter the BASIC *command* RUN. The program then asks me to enter the checkbook balance and the bank's closing balance. It then goes to or *calls* two *subroutines;* one to enter and tally the outstanding checks and the other to do the same with the outstanding deposits. Then it

program

```
100 REM CHECKBOOK, VER. 1, 10/24/77, CWP
101 REM CHECKBOOK BALANCING PROGRAM  THE MACHINE WILL
102 REM CALL FOR DATA FOR YOU TO ENTER  WHENEVER YOU
103 REM REACH THE LAST CHECK OR DEPOSIT ENTER THE FLAG
104 REM NUMBER 00.00  THE MACHINE WILL THEN COMPUTE
105 REM THE BALANCE AND ADVISE YOU IF THE CHECKBOOK BALANCES
106 INPUT "THE BANK BALANCE IS     ";B1:PRINT
110 INPUT "THE CHECKBOOK BALANCE IS     ";B2
115 REM THE PROGRAM NOW CALLS A SUBROUTINE TO
116 REM ENTER THE OUTSTANDING CHECKS
120 GOSUB 200
125 REM THE MACHINE NOW CALLS ANOTHER SUBROUTINE
126 REM TO ENTER THE OUTSTANDING DEPOSITS
130 GOSUB 300
135 REM THE MACHINE NOW COMPUTES THE FINAL BALANCE
140 F1 = B1 + D1 - C1
145 PRINT:PRINT:PRINT:PRINT
146 PRINT "THE FINAL BALANCE IS",,F1
150 PRINT "THE CHECKBOOK BALANCE IS",,B2
151 PRINT
152 REM THE MACHINE NOW COMPARES THE BALANCES
155 IF F1     B2 THEN 170
160 PRINT "THE CHECKBOOK BALANCES!"
165 END
170 PRINT "THE CHECKBOOK DOES NOT BALANCE,"
175 PRINT "THERE IS SOMETHING WRONG !!!!!!"
180 END

200 INPUT "OUTSTANDING CHECK     ";C2
205 IF C2 = 00.00 GOTO 220
210 C1 = C1 + C2
215 GOTO 200
220 RETURN

300 INPUT "OUTSTANDING DEPOSIT     ";D2
305 IF D2 = 00.00 GOTO 320
310 D1 = D1 + D2
315 GOTO 300
320 RETURN
```

returns to the *main* program. Subroutines are smaller programs that perform a task that is needed repeatedly throughout the main program.

The main program figures out the final balance, then prints both the final balance and the checkbook balance. It makes a comparison between the two balances; and if they are the same, the computer prints, "The Checkbook Balances!" If the figures do not agree, the machine prints, "There is something wrong, the checkbook does not balance!" Then I have to go back and find out where *I* subtracted wrong in the book.

A *listing* of my program is included for those who would like to try it. Of course, there may be other ways to program this problem (in addition some modifications might be necessary under different versions of BASIC). Part of the challenge of program design is finding better ways to make the computer do the things you want it to do, the way you want them done.

This program is just that first step in the thousand mile journey. Coming soon will be a program to compute the heat loss factors of your home so that you can plan the most efficient and cost effective changes in your heating system or insulating properties.

Two for the price of one: here's another easy-to-understand program written in Basic language so you can try it out on your home computer.

BY PETE STARK

One of the many uses for a small, personal computer is in Computer Aided Instruction (CAI). A very popular concept about a dozen years ago, CAI never became widely used because of the high costs involved with the traditional computer approach. But now, with the aid of inexpensive small computers, computer aided teaching and drill may finally come into its own.

Teaching new concepts with the help of a computer is still an involved and difficult job. A trained educator with a talent for computer programming usually has to prepare a set of lessons which takes a student through the material to be learned in a slow and precise way. The time required to prepare these lessons is so large, that it's unlikely that a casual computer user or owner will even tackle it. The best we can hope for is that sometime in the near future prepared computer programs for teaching specific subjects will be available.

On the other hand, preparing the program for computer review and drill is fairly easy. This program is a short Basic program to take a youngster through some multiplication drill. The program makes up 10 different multiplication problems and asks the youngster to type in the answers. For each answer, the computer prints out a short message indicating whether the answer is right or wrong. After 10 tries, the computer prints out the youngster's score, along with a short message telling him how well he is doing.

Most of the program is easy to follow. Line 1 prints out the words MULTIPLICATION DRILL. Line 2 makes a number S equal to zero; S will be used to keep score by indicating the number of correct answers. It is initially set to zero because at the very beginning the youngster has not answered any questions correctly.

Lines 4 and 5 select two random numbers called A and B. Unless you are familiar with Basic, these two lines are a bit difficult to follow, so let us just assume that they work. Both A and B will be numbers between 1 and 12, different for each problem. Line 6 then multiplies them to get the number P, which is the correct answer. But note that the computer doesn't yet print out the answer.

program

```
0001 PRINT "MULTIPLICATION DRILL."
0002 LET S=0
0003 FOR I = 1 TO 10
0004 LET A = INT (12*RND(0)+1)
0005 LET B = INT (12*RND(0)+1)
0006 LET P = A*B
0007 PRINT "WHAT IS "; A; " TIMES "; B;
0008 INPUT G
0009 IF G=P THEN GO TO 12
0010 PRINT "WRONG.  THE ANSWER IS "; P
0011 GOTO 14
0012 PRINT "GOOD!"
0013 LET S=S+1
0014 NEXT I
0015 PRINT "YOU GOT "; S; " RIGHT.  ";
0016 IF S<5 THEN PRINT "THAT'S TERRIBLE!"
0017 IF S<10 THEN PRINT "YOU NEED MORE PRACTICE"
0018 IF S=10 THEN PRINT "FANTASTIC.  YOU'RE AN EXPERT."
0019 PRINT "WANT TO TRY AGAIN?  YES OR NO?"
0020 INPUT A$
0021 IF A$=" YES" THEN GO TO 2
0022 PRINT "OK, SEE YOU LATER."
0023 END
```

In line 7, the computer prints out the words WHAT IS, then the value of A followed by the word TIMES, and finally the value of B. This is followed by line 8, when the computer waits for the youngster to type in (or *input*) his guess G. If the guess is right, then line 9 tells the computer to go to line 12, which prints the word GOOD! On the other hand, if the guess is wrong, the computer continues from line 9 to line 10 and prints out WRONG. THE ANSWER IS followed by the real, correct product P.

Lines 3 and 14 work together to repeat this entire series of steps exactly 10 times. Each time the youngster answers correctly, line 13 adds 1 to the value of S, so that after 10 tries S is equal to the number of correct answers. Then line 15 prints out YOU GOT, the value of S, and the word RIGHT.

Lines 16 through 18 now examine the number of correct answers. If this number is less than five, line 16 instructs the computer to print the words THAT'S TERRIBLE! If the number is less than 10, then it prints YOU NEED MORE PRACTICE. Only if the number S is exactly equal to 10 does line 18 print FANTASTIC. YOU'RE AN EXPERT.

Finally, line 19 tells the computer to ask whether the youngster would like to try again with another 10 problems, and line 20 waits for an answer which the computer calls A$. If the answer is YES, then line 21 tells the computer to go back to step 2 and run the program again. Otherwise, the computer is told to print OK, SEE YOU LATER, and the program ends at line 23.

The program could easily be changed to provide drill in addition or subtraction instead of multiplication just by changing line 1, and also by changing the * in line 6 (which means *times*) to either + or −.