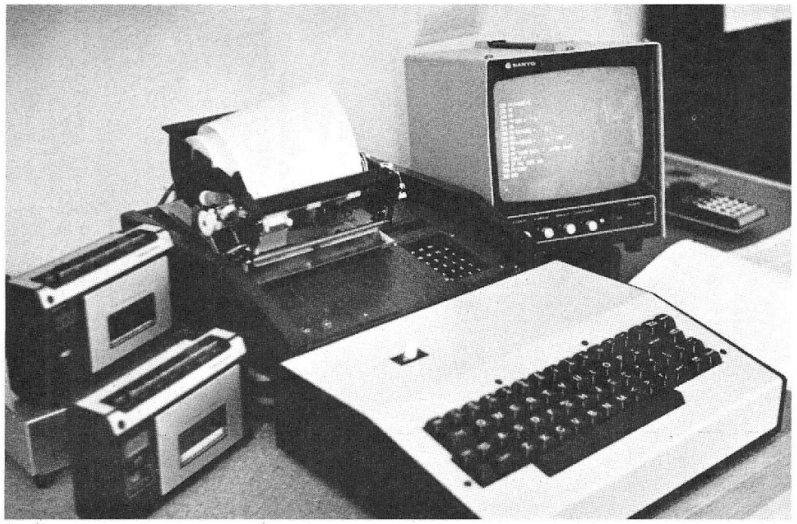
Charles R. Carpenter
2228 Montclair Place
Carrollton TX 75006

Not So Tiny

Tiny BASIC is the favorite higher-level language of many microcomputer users. Here is one technique for overcoming Tiny's lack of FOR/NEXT loops.



KIM-1 and KIM-2 in redwood enclosure, ACT-1 TVT, Telpar Printer, Computerist power supply, Radio Shack recorders.

```
:LIST

10 REM   ORIGINAL VERSION
11 REM
100 FOR Y=1 TO 10
110 LET C=0
120 FOR X=1  TO 50
130 LET F=INT(2*RND(1))
140 IF F=1 THEN 180
150 PRINT "T";
160 GOTO 200
170 REM   C COUNTS NO OF HEADS
180 LET C=C+1
190 PRINT "H";
200 NEXT X
210 PRINT
220 PRINT "HEADS    ";C;"    OUT OF 50 FLIPS"
230 NEXT Y
240 END
```

Listing 1.

Programs written in Tiny BASIC and other small interpreters can be useful and fun. First, some changes in programming techniques and philosophy are needed, though, because there are fewer statements and commands in small interpreters.

One basic and very useful programming tool is the loop. Several articles have been written about the power and use of loops properly written and executed in a program. Usually in larger BASICs, these loops are written with FOR-NEXT statements. In Tiny BASIC, the equivalent statements are LET, IF . . . THEN GOTO.

To illustrate the conversion of FOR-NEXT statements to LET, IF . . . THEN GOTO statements, I have used the program in Listing 1. This is a coin-flipping routine with one counting loop inside another. The outside loop resides between lines 100 and 230; the inside loop is between lines 120 and 230. Lines 10 and 11 are my comment and are not part of the original program. It is not possible to run this program on my system because the Tiny BASIC interpreter would not recognize line 100 and would stop.

Listing 2 is my version rewritten in Tiny BASIC. I have added a couple of features, such as the INPUT N line, which lets you select N sets of 50 flips. Also, I like to see DONE (or something) at the end of a program. This way I know the program didn't quit in the middle (if the algorithm was right, anyway). Otherwise, Tiny BASIC used two more program lines than the larger BASIC version.

In my program, the two main loops comparable to the sample program are started with a LET statement. The outside loop is between lines 110 and 250 and controls the number of passes of 50 flips set in line 100. The inside loop is between lines 130 and 210 and controls the number of flips set in line 210. As I stated there are two additional lines—the counters for the two loops. The loop counter in line 200 increments by one on each pass through the program until it reaches the values in line 210. Incrementing the I loop (in line 240) by one occurs until the value in line 250 is reached. In this case, I is compared to N, the value input in line 100. The value of N lets the user select how many sets of 50 flips are to be run by the program before it ends.

Coin flipping, counting and printing are handled in lines 140 to 190. Line 140 randomizes the number 2 (1 is added so there are no zeros). If the random number is 1, it becomes a "head" and passes to the head counter in line 180. The head counter increments by one and prints an H, then increments the X loop by one. If X is less

than the limiting value (50), the program returns to the flip routine at line 140 and starts through again.

If F does not equal 1 in line 150, the value becomes a "tail," a T is printed, X is incremented (by jumping to line 200) and compared to the limiting value. This time, if 50 flips have occurred, the program falls through to the print statement in line 230. Heads (C) counted in line 180 are printed out and the program tests the relationships in lines 240 and 250. When I > N, the program prints DONE and ends.

Tiny BASIC, even though small in size, has power enough to produce significant programs. Applications are limited only by your imagination and user space in your computer's memory. In addition to some tricks using implied statements and commands to save memory, I have written programs to plot a graph, do simple graphics, do some limited data processing and simulate assembly processes in a small manufacturing company.

I plan to try several potential capabilities that include use of the USR function to save and load from a cassette tape. I would like to share my ideas with anyone interested, and I believe *Kilobaud* would be happy to publish programs for the development of a Tiny BASIC software library.■

```
:LIST
10 REM   TINY BASIC FOR KIM-1
11 REM   6502 V.IK BY T. PITTMAN.
12 REM
13 REM   PROGRAMED BY:
14 REM   C. R. (CHUCK) CARPENTER W5USJ
15 REM   2228 MONTCLAIR PL.
16 REM   CARROLLTON, TX. 75006
17 REM
18 REM   FLIPS A COIN 'N' TIMES 50 AS SELECTED
19 REM   IN LINE 100, THEN PRINTS THE NUMBER OF
20 REM   HEADS IN EACH 50 FLIPS.
21 PR
22 PR
100 INPUT N
110 LET I=1
120 LET C=0
130 LET X=1
140 LET F=(RND(2)+1)
150 IF F=1 GOTO 180
160 PRINT "T";
170 GOTO 200
180 LET C=C+1
190 PRINT "H";
200 LET X=X+1
210 IF X<=50 GOTO 140
220 PRINT
230 PRINT "HEADS    ";C;"    OUT OF 50 FLIPS"
240 LET I=I+1
250 IF I<=N GOTO 120
260 PRINT
270 PRINT "DONE"
280 END

:RUN
? 5
HTTHTTTHHTHHTTTTHHHHHHTHHHTHTHHTTTHHTTHHTTTHHTHTTH
HEADS  26  OUT  OF  50  FLIPS
HHTHHHTHHHHHTTHTTHHTTHHTHHTTTTHHTHHHTHTHTHTTTTTHHH
HEADS  28  OUT  OF  50  FLIPS
TTHHTTTHHHHTTTHHTHHHHHTHTTHTHHTHHTHHHTHHTHTTTTTHHH
HEADS  28  OUT  OF  50  FLIPS
THTHHHHTTTTHTTTTTHTTTTHHHTHTHTHHHHHHTTTTHTHHHTHTHH
HEADS  25  OUT  OF  50  FLIPS
TTHTTHHTTTTTTTTHTTHTHTTTTHTTTHTTHHHTTHTHHTHTHTHTHT
HEADS  18  OUT  OF  50  FLIPS

DONE
```

Listing 2.