**Radio Shack**
**TRS-80**
**HANDS ON**
**BCL PROGRAM**
BASIC COMPUTER LITERACY

# My TRS-80 Likes Me*

*When I teach kids how to use it!

by Bob Albrecht

...a resource guide for the elementary teacher.

## a foreword by Bob Albrecht

The Radio Shack TRS-80 is a great machine for teaching kids how to use, program and enjoy computers. I've been helping kids learn how to enjoy BASIC for 16 years. For this purpose (helping kids enjoy BASIC), the TRS-80 is the best computer I have used, especially if the learners are elementary school children.

> IMPORTANT NOTICE! I am not saying that the TRS-80 is the best computer for *all* purposes. I am not saying the TRS-80 is the best overall educational computer. I am saying that I think the TRS-80 is the best computer that I have used (so far) to teach elementary school children, grades 4, 5 and 6, how to understand and enjoy BASIC.

So, if you have a TRS-80, grab a bunch of kids and help them learn how to understand simple programs in TRS-80 BASIC. Here are ideas on how to go about it, including numerous short programs that many kids have enjoyed. Most of these programs will work in either Level I or Level II BASIC. A few are written in Level II; you can easily change them to Level I.

Think of the following as an outline of things to do. You explain what is happening, answer questions, invent variations on my ideas and (of course!), use your own ideas. But don't do the typing. Let the kids do all the hands-on stuff. Be patient—let them make mistakes, correct their own mistakes, and above all, encourage them to EXPERIMENT!

EXPERIMENT!
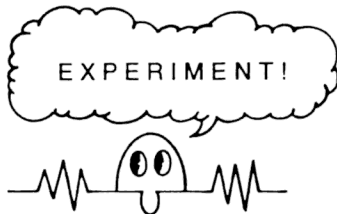
Oh yes, one more thing. I don't teach people how to program in BASIC. First, I help them learn how to read and understand simple BASIC programs. In order to create something in a language, you must first understand the language. (That does sound reasonable, doesn't it?) Then, once you understand the language (or some of it), perhaps you can express your own original ideas in the language.

So, I like to help kids (and some adults) learn to read and understand BASIC. Presto! Many of them then teach themselves how to use the language creatively.

magic!

But even those people who don't write world-shaking original programs might learn something else—computers are sure stupid, and not very mysterious. Or, people might learn to modify someone else's programs to better serve a personal purpose. Or team up with friends who love to write programs in order to create a synergistic program that would not otherwise exist. BASIC, in the universe of home/school/personal computers, is an easy-to-understand starting point for learning about computers.

## BEGIN . . . .

- Show your eager young learners how to hook up the TRS-80 and turn it on.

- Show them how to clear the screen. Start each "episode" with a clear screen.

- Tell them about the prompt (>) and the cursor (—).

When you see this it is your turn to type.  ➡ /  > —

- Have someone clear the screen, type her name, then press the ENTER key. The TRS-80 will probably type an error message. Explain that "The TRS-80 doesn't understand you." Never mind—tell them not to worry about occasional misunderstandings; they will soon learn how to make the TRS-80 understand!

- Kids are fascinated by their names. In this article, we will do lots of things with names. Start with direct statements such as the following. You will have to remind them several times to press ENTER after typing a statement.

PRINT "LUCY"

PRINT "CHARLIE"

PRINT "SNOOPY"

PRINT "LUKE SKYWALKER"

*and press ENTER*

and so on.

Let every kid do this.

- Now the fun begins. Have each kid enter and run a short two line program which will fill the screen with her name. (Boys are also encouraged to do this.)

```
10 CLS
20 PRINT "LUCY";
30 GOTO 20
```

When the above program is RUN, the screen quickly fills up with the name 'LUCY'. Then the names seem to move to the right (an optical illusion). Experiment with names and with other characters. Take your time; keep it slow and easy. Let the kids set the pace. Don't try to turn them into computer phreaks. Just let them have fun learning at their own speeds.

- Of course, you will show them how to interrupt a program by pressing the BREAK key. Also show them how to LIST a program and how to change only Line 20 in order to put a different name on the screen.

> BEWARE! In Level II BASIC, a syntax error causes the TRS-80 to go into Edit Mode. Press Q to get out—don't get into heavy editing this early in the game.

- Make it easier to change the name. Tell them about "boxes" such as A$, B$, etc., which can hold names or other strings of characters.

```
10 CLS
20 LET A$ = "LUCY"
30 PRINT A$;
40 GOTO 30
```

Try other "boxes" such as B$, C$, N$, etc. Use single letter variables only; it will save you lots of time. If you are a "computer scientist" or "computer hobbyist" please remember: most of us aren't.

- Still easier. Use the INPUT statement.

```
10 CLS
20 INPUT A$
30 PRINT A$;
40 GOTO 30
```

Let everyone use this program.

- Slow down the action. Use an "empty" FOR-NEXT loop as a time delay.

```
10 CLS
20 INPUT A$
30 PRINT A$;
40 FOR Z = 1 TO 100
50 NEXT Z
60 GOTO 30
```
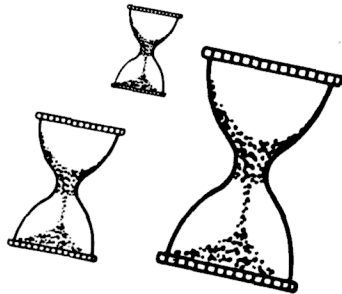
The FOR-NEXT loop (Lines 40 and 50) causes the TRS-80 to count from 1 to 100. Kids are usually surprised at how fast the computer can count to 100. Try other numbers (instead of 100) in the FOR statement.

Slower: Use numbers greater than 100.
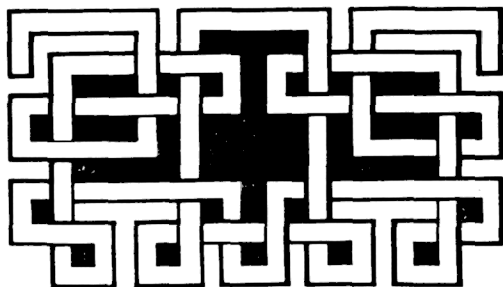
Faster: Use numbers less than 100.

- Hmmm . . . how about a variable time delay?

```
10 CLS
20 INPUT A$
30 PRINT A$;
40 LET T = 100
50 FOR Z = 1 to T
60 NEXT Z
70 GOTO 30
```

Now change T by replacing line 40 with:

```
40 LET T = 200
```

- We can blink a name on and off.

```
10 CLS
20 INPUT A$

30 CLS
40 PRINT A$;
50  LET T = 100        ⎫
60 FOR Z = 1 TO T      ⎬  Name ON
70 NEXT Z              ⎭

80 CLS                 ⎫
90 LET T = 100         ⎬  Name OFF
100 FOR Z = 1 TO T
110 NEXT Z             ⎭

120 GOTO 30
```
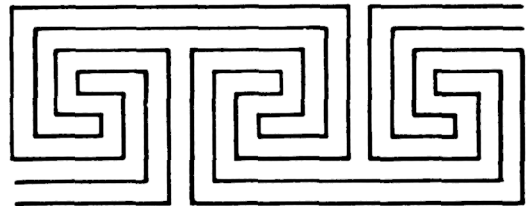
To slow down the action, make T>100. To speed it up, make T<100.

- Our program has two identical time delays. Great opportunity to introduce SUBROUTINES!

```
10 CLS
20 INPUT A$                    ◀ Save room for Line 30 (below)

40 CLS
50 PRINT A$;         ⎫
60 GOSUB 100         ⎬  Name ON

70 CLS               ⎫
80 GOSUB 100         ⎬  Name OFF

90 GOTO 40

100 LET T = 100
110 FOR Z = 1 TO T
120 NEXT Z
130 RETURN
```

Take lots of time to explain this one!

- Well, let's not wear out the upper left hand corner of the screen. Tell them about PRINT @ (or PRINT AT in Level I BASIC). In the above program, replace Line 50 with one of these. In fact, do all of them, one by one.

```
50 PRINT @0,A$;
50 PRINT @32,A$;
50 PRINT @64,A$;
50 PRINT @500,A$;
```

and so on.

Of course, tell them that there are 1024 print positions numbered from 0 to 1023. Make a map of the screen showing where the print positions are located. Below is a handy program for putting your name where you want it on the screen. Explain the use of strings within INPUT statements.

```
10 CLS
20 INPUT "WHAT IS YOUR NAME";A$
30 INPUT "WHERE SHOULD I BLINK IT";N

40 CLS
50 PRINT @N,A$;          Name ON
60 GOSUB 100             at position N

70 CLS                   Name OFF
80 GOSUB 100             (everywhere)

90 GOTO 30

100 LET T = 100
110 FOR Z = 1 TO T
120 NEXT Z
130 RETURN
```

As usual, take your time and make sure everyone understands this program. Remember, they are learning how to read and understand a language. The first step in learning how to create original stuff in a language is to first understand the language!

• As Tom Lehrer might say, "If you are getting bored by this stuff, you have only yourself to blame; you should never have let me begin." And so, I will reluctantly end with just one more program, which causes a name to scamper across the screen from upper left to lower right, then again and again and . . .

```
10 CLS
20 INPUT "WHAT IS YOUR NAME";A$

30 FOR K = 0 TO 1023
40 CLS
50 PRINT @K,A$;
60 NEXT K

70 GOTO 30
```

Too fast? Probably—so, you put in a time delay to slow down the action.

## MEANDERING . . . .

So far, we've done "deterministic" things with words on the screen. Now, we will meander—do some random stuff—cause words to wander about the screen, or appear helter-skelter, in unpredictable non-patterns.

One of the nicest things about the TRS-80 is the way the RND function works. It gives integer random numbers in a way that is easy to understand and use by people who are not math wizards. Most other BASICs give random numbers between 0 and 1. I really can't understand why—most applications (especially games!) require integer random numbers. In most BASICs, much mind boggling math is required to get the desired integer random numbers.

Desired random numbers: 1 or 2

TRS-80: $RND(2)$

Most others: $INT(2*RND(1)) + 1$ ←————

Desired random numbers: 1, 2 or 3

TRS-80: $RND(3)$

Most others: $INT(3*RND(1)) + 1$
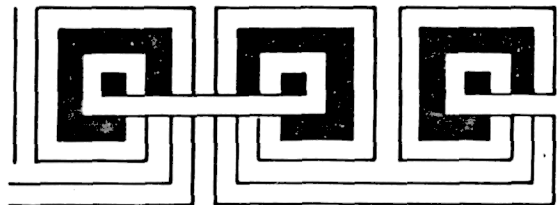
Desired random numbers: 1, 2, 3, 4, 5, 6

TRS-80: $RND(6)$

Most others: $INT(6*RND(1)) + 1$

Desired random numbers: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

TRS-80: $RND(10) - 1$

Most others: $INT(10*RND(1))$

Got the idea? It goes like this. If n is a positive integer, then $RND(n)$ will give random positive integers in the range, 1 to n. Simple and neat!

Bonus! $RND(0)$ will give random numbers in the range, 0 to 1, just like all the other computers. The TRS-80 gives you the best of both worlds.

IMPORTANT NOTICE! Remember, the series is about teaching BASIC to kids. So, the RND function is one of the most important elements of BASIC. It is one of the things that makes BASIC fun for kids! For ease in teaching, use the integer RND function.
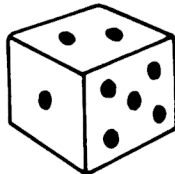
# Experiment!

Do some experiments.

```
10 CLS
20 PRINT RND(2);
30 GOTO 20
```

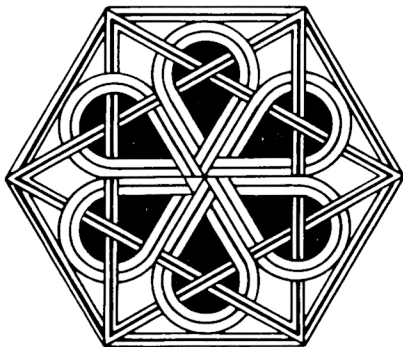This fills the screen with 1's and 2's.
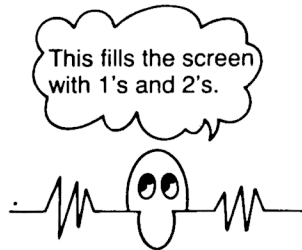
```
10 CLS
20 PRINT RND(3);
30 GOTO 20
```

```
10 CLS
20 PRINT RND(6);
30 GOTO 20
```

```
10 CLS
20 PRINT RND(10) – 1;
30 GOTO 20
```

OK, so RND gives random numbers. So what? Well, let's use them to put a name here, there and everywhere on the screen. Here for a moment, there for a moment, somewhere else for a moment.

```
10 CLS
20 X = RND(1023)
30 PRINT @X,"LUCY";
40 T = 100
50 FOR Z = 1 TO T        } Time delay
60 NEXT Z
70 GOTO 10
```

Explain lines 20 and 30 carefully. Line 20 gives a random integer from 1 to 1023, inclusive, and puts this number in box X. Line 30 causes the name "LUCY" to be printed at position X on the screen. Lines 50 and 60 are a time delay. Change Line 40 to decrease or increase the delay.

Faster: Make T<100

Slower: Make T>100

Let each kid change Line 30 to his or her name. Why let Lucy get all the glory? To make it easier, try the following program.

```
10 CLS
20 INPUT "WHAT IS YOUR NAME";N$
30 CLS
40 X = RND(1023)
50 PRINT @X,N$;
60 T = 100
70 FOR Z = 1 TO T
80 NEXT Z
90 GOTO 30
```

As usual, slowly and patiently explain what is happening, to your eager young learners. If they say, "What if . . . ?" then you say, "Try it and find out!"

## GRAPHICS . . . .

This material is intended as an outline for teachers on how to help kids learn to use, program and enjoy computers. These ideas are best used when a kid asks, "How does the computer do that?," or "How can I make the computer do what I want it to do?," or "Can the computer tell me (whatever)?" or . . .

So, you have a TRS-80 and the kids have played Hurkle, Hangman, Guess My Number, Taxman, Invasion Force, and countless other games. Now, they want to know more; they want to write game-playing programs; they want to put interesting visual patterns on the screen; THEY WANT TO CONTROL THE COMPUTER.

Why not? They control the future; so, let them control the computer, the tool of the future; give your kids this tool; let them shape it in ways unknown to us. Then stand back and enjoy!!

# Patterns

Patterns! . . . That's what we will do this time. Show you how to put patterns on the screen, using tiny rectangles of light. And, we will do some "computer art," using the familiar RND function and a new BASIC statement called SET.

- SET turns on a tiny rectangle of light somewhere on the screen.

Somewhere on the screen? Well, of course, we must tell the TRS-80 where to turn on the tiny rectangle of light.

SET (64,24)

This far OVER     This far DOWN



24 DOWN
64 OVER
Tiny rectangle of light

Got it?
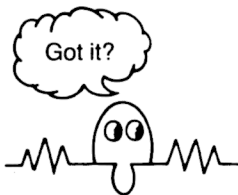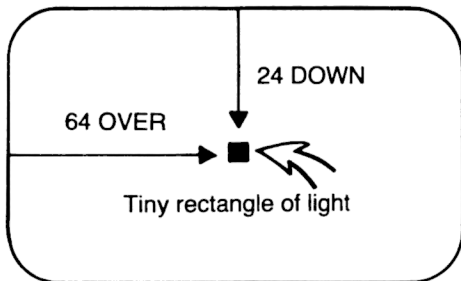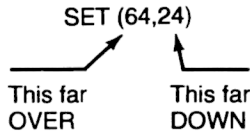
Think of it like this: SET (OVER, DOWN)

- OVER can be a whole number, 0 to 127.

- DOWN can be a whole number, 0 to 47.

OVER
0 to 127

D
O
W     0 to 47
N
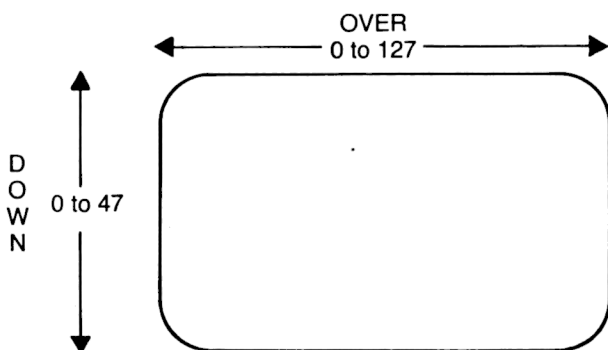
# Experiment!

Try these to get the hang of things.

- Press CLEAR, then press ENTER

- Type SET (0,0) and press ENTER
  Where is the tiny rectangle?

- Type SET (127,0) and press ENTER
  Where is the tiny rectangle?

- Type SET (0,47) and press ENTER
  Where is the tiny rectangle?

- Type SET (127,47) and press ENTER
  Where is the tiny rectangle?

Now, for each of the following, guess where the rectangle will appear before you press ENTER. Then do it . . . press ENTER . . . were you close?

- Type SET (0,24) and press ENTER

- Type SET (127,24) and press ENTER

- Type SET (64,0) and press ENTER

- Type SET (64,47) and press ENTER

- Type SET (64,24) and press ENTER

- Type SET (32,12) and press ENTER

# Starfall

Now, pretend that the screen is the sky. Also, pretend that each tiny rectangle of light is a star appearing in the night sky. Make it happen by storing and running this program.
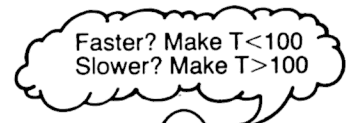
```
NEW
100 CLS
110 OVER = RND(127)
120 DOWN = RND(47)
130 SET (OVER,DOWN)
140 GOTO 110
```

or try,
110 OVER = RND(128) − 1
120 DOWN = RND(48) − 1
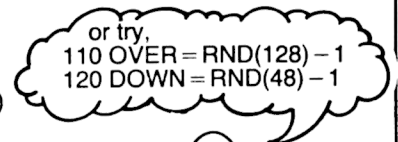
When you RUN this program, the sky will begin filling with stars. Try to find . . . constellations, shapes, patterns . . . as stars turn on.

Too fast? Add a time delay.

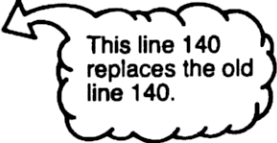Faster? Make T<100
Slower? Make T>100

```
133 T = 100
135 FOR Z = 1 TO T
137 NEXT Z
```

Or, here is a trick you can use to stop the action momentarily while you stargaze, then continue letting more stars come out.

Add these lines to the program.

```
140 IF INKEY$ = " "THEN 110
150 IF INKEY$ = " "THEN 150
160 GOTO 110
```

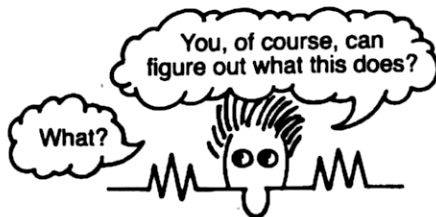> This line 140 replaces the old line 140.

Have you added the above lines (140, 150, 160) to the original program? Yes . . . good! Now, RUN the modified program. To stop it, press any key; then, to continue, press any key. Now you can stop time! Or restart it! Happy stargazing.

# Walkabout

Suppose you start near the center of the screen (OVER = 64, DOWN = 24) and walk (stagger?) at random. Up? Down? Left? Right? Try this program.

```
NEW
100 CLS

110 OVER = 64          Start near the center

120 DOWN = 24          of the screen

130 SET (OVER,DOWN)    Turn on a light

140 A = RND(3) − 2

150 B = RND(3) − 2
```

> You, of course, can figure out what this does?
>
> What?

```
160 OVER = OVER + A    Well, obviously (?), this
                       will change OVER and
170 DOWN = DOWN + B    DOWN.

180 GOTO 130
```

Dismay not! Try this program. It will start near the center of the screen and . . . meander. Eventually, it will (probably) stop with an error message.

### ?FC ERROR IN 130

That happens because we tried to wander off the screen. Off the screen? Well, that means: OVER became less than 0 or more than 127 or DOWN became less than 0 or more than 47. More about that in a minute . . .

In the meantime, try this program.

```
NEW
100 CLS

110 X = RND(128) − 1      X will mean 'over.'

120 Y = RND(48) − 1       Y will mean 'down.'

130 SET (X,Y)

140 SET (X,47 − Y)

150 SET (127 − X,Y)       Remember: the first
                          number means OVER;
160 SET (127 − X,47 − Y)  the second number
                          means DOWN.
170 GOTO 110
```

Run it for awhile, then press BREAK, RUN it again, etc. . . .

# More Walkabout

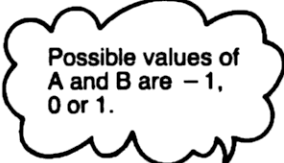Here again is our "walkabout" program.

```
100 CLS

110 OVER = 64
120 DOWN = 24

130 SET (OVER,DOWN)

140 A = RND(3) − 2
150 B = RND(3) − 2

160 OVER = OVER + A
170 DOWN = DOWN + B

180 GOTO 130
```

> Possible values of A and B are − 1, 0 or 1.

Remember, it eventually tries to wander off the screen. This happens if OVER<0 or OVER>127 or DOWN<0 or DOWN>47. When this happens, the TRS-80 stops with the following error message:

?FC ERROR IN 130

Aha! Let's fix that little bug in our program. Here is our bug tranquilizer. Add it to the program.

```
163 IF OVER<0 THEN OVER = OVER + 1
167 IF OVER>127 THEN OVER = OVER − 1
173 IF DOWN>0 THEN DOWN = DOWN + 1
177 IF DOWN>47 THEN DOWN = DOWN − 1
```

These additions to the program will sometimes seem to make the screen image "bounce along" against the edge of the screen. Try it. As the pattern develops, what shapes do you see? A castle? A dragon? A flower? An angular cloud? Buildings on a hillside?

Wouldn't it be nice to be able to stop the computer and look at the pattern on the screen? Or photograph it? Easy to do. Just add the following statements.

```
180 IF INKEY$ = " "THEN 130

190 IF INKEY$ = " "THEN 190 ELSE 130
```

IF THEN . . . ELSE, is very handy!
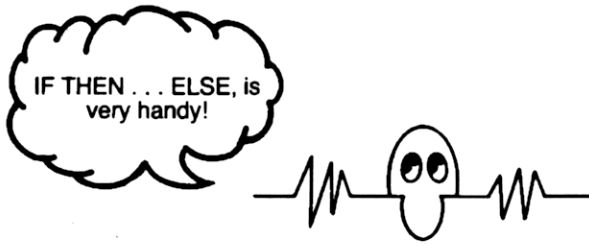
Now, having added the above lines 180 and 190, RUN the program. To stop it, press any key (except BREAK!). Gaze upon the pattern on the screen. Then, press any key (except BREAK) and the pattern will continue growing.

INTERLUDE—A GAME

Once upon a time, computers were veerrryyyy expensive. Many of them "talked" to people by means of a rather large, and definitely noisy, device called a . . . (suspense) . . . Teletype®!

The Teletype (or TTY, as it became affectionately known) typed information on paper. From these almost prehistoric times, we bring you a simple computer game enjoyed by multitudes of children—a game called NUMBER or, if you prefer, GUESS MY NUMBER. Our version of NUMBER, shown below, is modified slightly (but only slightly) for TRS-80.

```
100 REM***NUMBER, A NUMBER GUESSING GAME
       FOR TRS-80

200 REM***TELL PLAYER ABOUT THE GAME
210 CLS
220 PRINT "I AM THINKING OF A NUMBER FROM
       1 TO 100."
230 PRINT "GUESS MY NUMBER!!!"

300 REM***COMPUTER 'THINKS' OF A NUMBER, X
310 RANDOM
320 X = RND(100)

400 REM***GET GUESS, G, COMPARE WITH X
410 INPUT "YOUR GUESS"; G
420 IF G = X THEN 610
430 IF G<X THEN PRINT "TRY BIGGER"
440 IF G>X THEN PRINT "TRY SMALLER"
450 GOTO 410

600 REM***WINNER!
610 PRINT "THAT'S IT!!! MY NUMBER IS" X
620 FOR T = 1 TO 2000:NEXT
630 GOTO 210
```

Well, try the above program. It uses the screen abysmally, doesn't it? This program is fine for printed output, but terrible for TV. The printer is a one-dimensional medium; the TV screen is a two-dimensional medium.

So, here is a new version of NUMBER which illustrates a different way to use the TV screen.

```
100 REM***NUMBER, A NUMBER GUESSING GAME FOR TRS-80
110 REM***COPYRIGHT 1979 BY BOB ALBRECHT

200 REM***TELL PLAYER ABOUT THE GAME
210 CLS
220 PRINT @268, "I AM THINKING OF A NUMBER FROM 1 TO 100."
230 PRINT @396, "GUESS MY NUMBER! WHEN I ASK FOR YOUR"
240 PRINT @524, "GUESS, TYPE A NUMBER AND THEN PRESS THE"
250 PRINT @652, "BIG 'ENTER' KEY."
260 PRINT @780, "PRESS THE 'ENTER' KEY AND I WILL BEGIN."
270 IF INKEY$ = " " THEN 270 ELSE 310

300 REM***COMPUTE SECRET, A RANDOM NUMBER FROM 1 TO 100
310 RANDOM
320 SECRET = RND(100)

400 REM***GET GUESS, COMPARE WITH SECRET (UP TO 13 GUESSES)
410 CLS
420 FOR N = 0 TO 12
430    PRINT @ 64*N," " ;
440    INPUT "YOUR GUESS"; GUESS
450    IF GUESS = SECRET THEN 610
460    IF GUESS<SECRET THEN PRINT @ 64*N + 20, "TRY BIGGER"
470    IF GUESS>SECRET THEN PRINT @ 64*N + 20, "TRY SMALLER"
480 NEXT N

500 REM***OOPS! RAN OUT OF GUESSES — LIMIT IS 13
510 PRINT "OOPS! THAT'S 13 GUESSES, MY NUMBER IS" SECRET
520 PRINT "TO PLAY AGAIN, PRESS THE 'ENTER' KEY."
530 IF INKEY$ = " " THEN 530 ELSE 210

600 REM***WINNER! WATCH THE STARS COME OUT!
610 CLS
620 FOR STARS = 1 TO 100
630    PRINT @RND(1023),"*";
640 NEXT STARS
650 PRINT @406, "MY NUMBER IS" SECRET
660 PRINT @529, "YOU GOT IT IN" N + 1 "GUESSES!!!"
670 PRINT @656, "TO PLAY AGAIN: PRESS 'ENTER' "
680 IF INKEY$ = " " THEN 680 ELSE 210

999 END
```
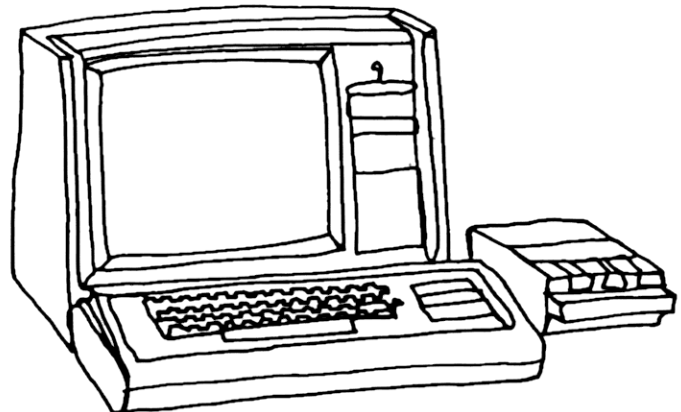
The Tandy-Radio Shack TRS-80