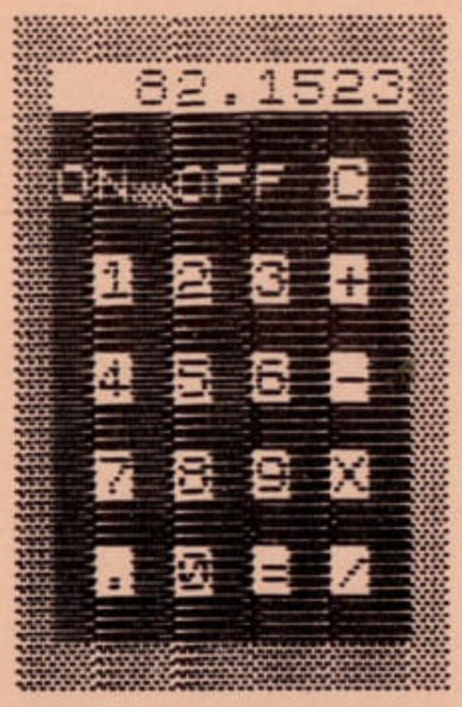
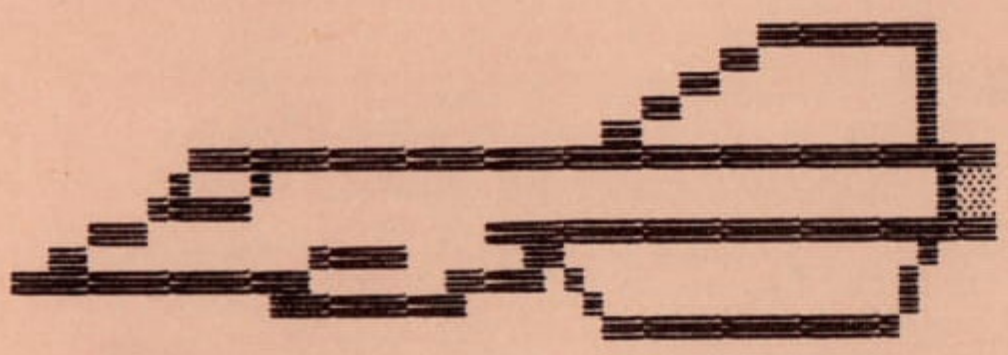
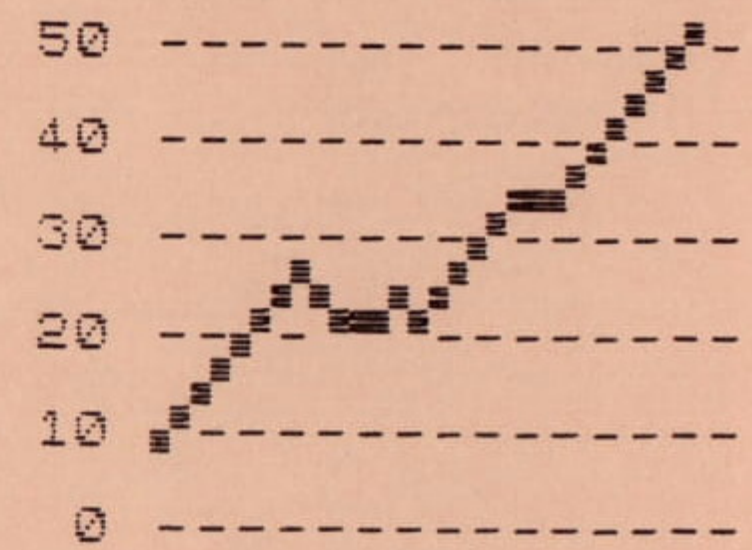
PUTTING YOUR TIMEX SINCLAIR 1000™ TO WORK FOR YOU

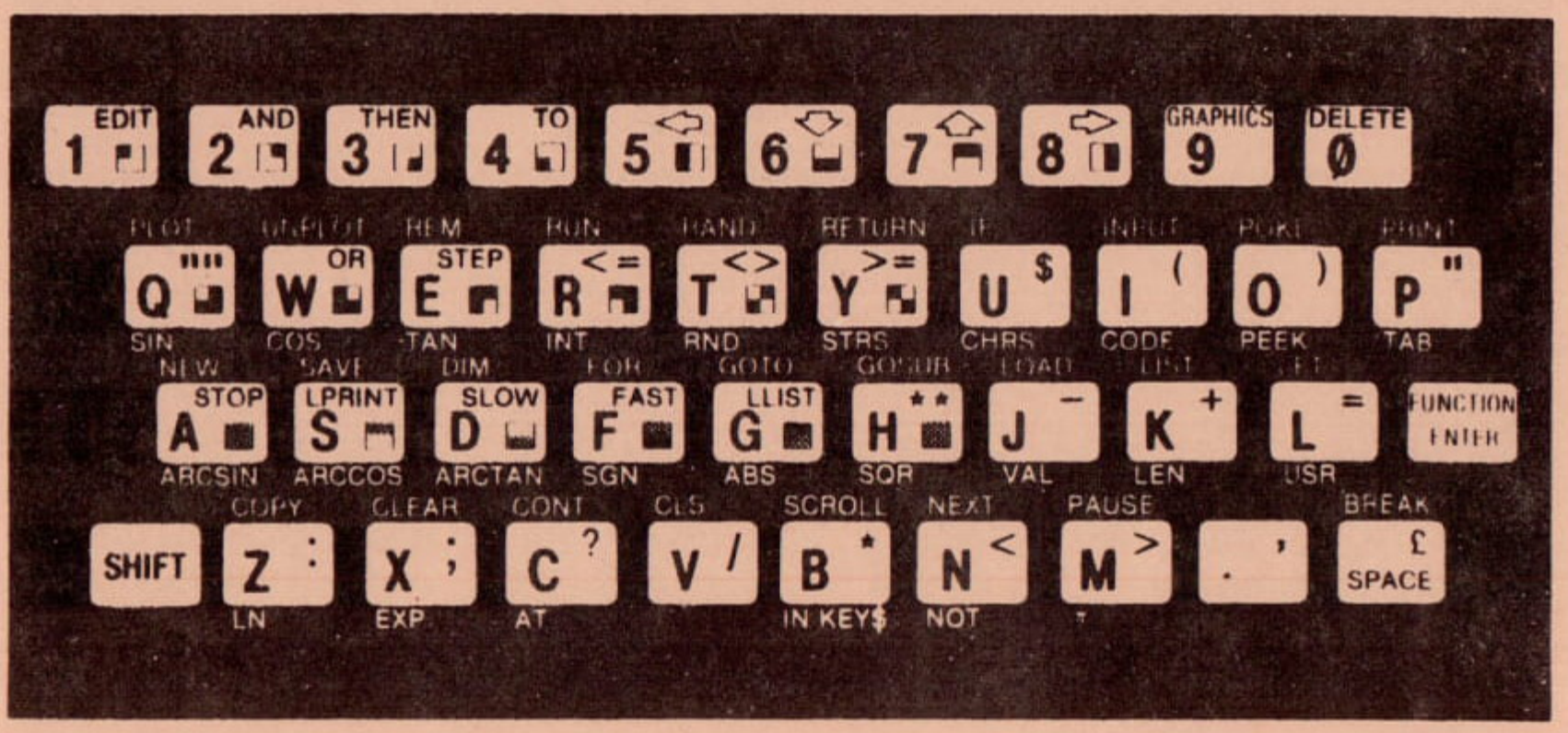


BY BOB JOHNSON



```
PHOENIX              9:30
DENVER               8:45
ATLANTA              1:10
LOS ANGELES          4:20
PORTLAND            11:15
```





PUTTING YOUR TIMEX SINCLAIR 1000™ TO WORK FOR YOU

by Bob Johnson

Printed in Salem, Oregon

TABLE OF CONTENTS

INTRODUTION

My first experience with BASIC programming was on a Hewlett-Packard 2000 timesharing system. I punched all of my programs on paper tape and made hard copy listings, which I kept stored away for five years. Then in March of 1983, after careful comparison with other computers, I plunked down my change for the Timex/Sinclair 1000. I was impressed. While it cannot match all of the capabilities of a large mainframe computer, for the price and size the T/S 1000 is an amazing machine. I have been able to convert all of my HP 2000 programs to run on my Timex. In fact, a few of them are in this collection.

If you have bought a few cassette programs for your T/S 1000 and have decided to learn how a program is written or if you have been writing your own programs and want to add some more to your library, then this book is just for you! The first section explores some of the mathematical capabilities of the computer. One program turns the computer into a calculator, complete with a graphic screen display. Other programs convert Fahrenheit and Celsius temperatures, English and metric measurements. There is even a program that similates a travel computer like those found in some automobiles. The next section covers data handling--both numerical and text. There are programs that help plan your budget, keep names and addresses, and display data in a graph. The last section is where the games are. The last one uses the power of a machine code routine to produce a fast action arcade type game.

Assuming that you have the basic 2K system with a users manual, then these programs are ready to use just as they are printed. I have included some suggestions for possible variations and definitely encourage you to modify a program if it would be more suitable for you to do so. A program can always be improved on.

You might want to make more use of REM statements or perhaps you found a faster, memory efficient way to do something. If you have more than 2K of memory available you may want to add more features in a program. In any case these programs should help you get better aquainted with your T/S 1000 and have some fun while doing it.

Math & Conversions

THE CALCULATOR IN YOUR COMPUTER

Hidden inside the sleek black case of your T/S 1000 is a full featured floating point calculator. This creature is not too hard to find, though. In the command mode type in PRINT 4+5 ENTER This gives you a nine at the top of the screen. That was not particularly taxing on the logic circuits so lets try PRINT ((4+5)*3+36/2)/5 This also gives you a nine. The ROM crunches numbers in a certain order of priority. Calculations are made from left to right, with operations in parenthesis done first. Then the functions are evaluated. Next, the * and / operators are recognized. Now, to get your calculator up and running, type in this:

```
 10 REM T/S 1000 CALCULATOR
 20 REM (C)1984 BOB JOHNSON
 30 GOSUB 190
 40 LET B$=""
 50 INPUT A$
 60 IF A$="=" THEN GOTO 110
 70 IF LEN A$>9 THEN LET A$=A$(
TO 9)
 80 LET B$=B$+A$
 90 IF CODE A$(1)>25 OR CODE A$
(1)<20 THEN PRINT AT 6,11;"
    ";AT 6,20-LEN A$;A$
100 GOTO 50
110 LET C=VAL B$
120 LET C$=STR$ C
130 IF LEN C$>9 THEN LET C$=C$(
TO 9)
140 IF C>999999999 THEN LET C$(
1)=">"
150 PRINT AT 6,11;"          ";A
T 6,20-LEN C$;C$
160 FOR I=1 TO 40
170 NEXT I
180 GOTO 40
190 PRINT AT 3,6;"T/S 1000 CALC
ULATOR";TAB 15;TAB 10;"▒▒▒▒▒▒▒▒▒▒
▒▒▒▒";TAB 10;"▒▒          ▒▒";TAB 10;
"▒▒ ▒▒▒▒▒▒▒▒▒▒ ▒▒";TAB 10;"▒▒ ON OFF C▒▒
▒▒";TAB 10;"▒▒ ▒▒▒▒▒▒▒▒▒▒ ▒▒";TAB 10;"
▒▒ 1 2 3 + ▒▒";TAB 10;"▒▒▒▒▒▒▒▒▒▒▒▒
";TAB 10;"▒▒ 4 5 6 - ▒▒";TAB 10;"▒▒
▒▒▒▒▒▒▒▒▒▒▒▒";TAB 10;"▒▒ 7 8 9 X ▒▒"
;TAB 10;"▒▒ ▒▒▒▒▒▒▒▒▒▒ ▒▒";TAB 10;"▒▒
. 0 = / ▒▒";TAB 10;"▒▒▒▒▒▒▒▒▒▒▒▒";
TAB 10;"▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒"
200 RETURN
```

This program utilizes the calculating abilities of the T/S 1000. It begins with a subroutine call in line 30 to perform the graphics routine. Line 40 initializes B$ as an empty string. A$ is then input, rounded off to nine digits and then added to B$. If A$ is not an operator it is printed in the calculator display. (Line 90) This continues until a "=" is entered and detected in line 60. The program then branches to line 110 where the value of B$ is stored as the variable C. C is checked for overflow and converted to a string variable--C$. It is rounded to nine digits, if necessary, and the result is printed in the display. After a delay loop in lines 160-170 the program is repeated.

When this program is run, numbers are input alternately with operators. Ex: 4 [ENT] + [ENT] 6 [ENT] = [ENT] which gives you a 10. Functions may be used, but I have found that the ROM "sees" a function as a string length of one. As a result, if A$ contains a function, it is not printed far enough to the left. Also keep in mind that this calculator does not keep a running total, as many others do, but evaluates the entire equation using the order of priority described before.

TEMPERATURE CONVERSION

This program accepts a number input by the user to represent a temperature. The number is followed by an "F" or a "C" to represent either Fahrenheit or Celsius. The number is then converted to a new number that represents the other temperature scale.

```
  10 REM TEMP CONVERSION
  20 CLS
  30 PRINT AT 5,0;"ENTER TEMPERA
TURE?";AT 5,17;
  40 INPUT T$
  50 PRINT ": ";T$( TO LEN T$-1)
;AT 7,0;
  60 LET T=VAL T$( TO LEN T$-1)
  70 IF T$(LEN T$)="F" THEN GOTO
 100
  80 IF T$(LEN T$)="C" THEN GOTO
 140
  90 GOTO 20
```

```
100 LET C=(T-32)*5/9
110 PRINT "CELSIUS TEMPERATURE:
";C
120 PAUSE 100
130 GOTO 20
140 LET F=T*9/5+32
150 PRINT "FAHRENHEIT TEMPERATU
RE: ";F
160 PAUSE 100
170 GOTO 20
```

To test this one, run it, then type in: 212F. What Celsius temperature does it return? On the next run type in that number, followed by a "C". Did you get your original number back? If not, check your program listing for accuracy. As you can see, the temperature is represented by T$. The number in T$ is stored in T, while the "C" or "F" is detected to determine which conversion to perform. After each conversion, there is a PAUSE. Then the program jumps back to the beginning and runs through again. A final note--the PAUSE in lines 120 and 160 may be adjusted to any suitable delay time.

METRIC CONVERSION

Do you get confused by kilograms, centimetres and litres? Or maybe you wonder how many centimetres are in an inch. If so, then this program is just what you need.

```
10 REM METRIC CONVERSION
20 REM (C)1984 BOB JOHNSON
30 LET H$="HOW MANY "
40 PRINT AT 0,2;"**** METRIC C
ONVERSION ****";AT 5,10;"SELECT
ONE";AT 7,9;"1----GALLONS";TAB 9
;"2-----POUNDS";TAB 9;"3-----INC
HES";TAB 9;"4------MILES";AT 15,
0;
50 LET S$=INKEY$
60 IF INKEY$="" THEN GOTO 50
70 GOSUB (5*(CODE S$-28)+6)*10
80 PAUSE 200
90 CLS
100 GOTO 40
110 PRINT H$;"GALLONS?"
120 INPUT G
130 LET L=G*3.785
140 PRINT AT 17,0;G;" GALLONS *
*** ";L;" LITRES"
150 RETURN
```

```
 160 PRINT H$;"POUNDS?"
 170 INPUT P
 180 LET K=P*.453
 190 PRINT AT 17,0;P;" POUNDS **
** ";K;" KILOGRAMS"
 200 RETURN
 210 PRINT H$;"INCHES?"
 220 INPUT I
 230 LET C=I*2.54
 240 PRINT AT 17,0;I;" INCHES **
** ";C;" CENTIMETRES"
 250 RETURN
 260 PRINT H$;"MILES?"
 270 INPUT M
 280 LET K=M*1.609
 290 PRINT AT 17,0;M;" MILES ***
* ";K;" KILOMETRES"
 300 RETURN
```

The first part of the program prints out a menu. That is a sort of multiple choice question that determines where the program will branch to next. The GOSUB destination in line 70 is based on the menu selection. After the proper conversion is made there is a PAUSE that can be modified to any suitable delay. This program is easy to adjust to convert from metric back to English. Rewrite lines 110, 160, 210 and 260 to ask for how many litres, kilograms, centimetres, and kilometres are to be converted. Then use the follwing conversion

converted. Then use the following formulas:

1 litre=.264 gallons 1 kilogram=2.204 pounds
1 centimetre=.393 inches 1 kilometre=.621 miles

Be sure to change the menu and output statements to show that you are now coverting metric back to English. If you have the 16K RAM pack then you can have a program that converts both ways.

TRAVEL COMPUTER

One of the automotive gadgets available these days is a travel computer. This neat little box displays information such as miles per gallon, and remaining fuel at the touch of a button. Here is a program that similates some of those functions found on a travel computer. It works a lot like the last program in that it uses a menu and a subroutine for each selection. Let's take a quick look at what each selection does.

A--MPG: input miles traveled and gallons of fuel used, output is miles per gallon

B--RANGE: input MPG and gallons of fuel available, output is travelling range

C--MPH: input miles and hours traveled, output is average miles per hour

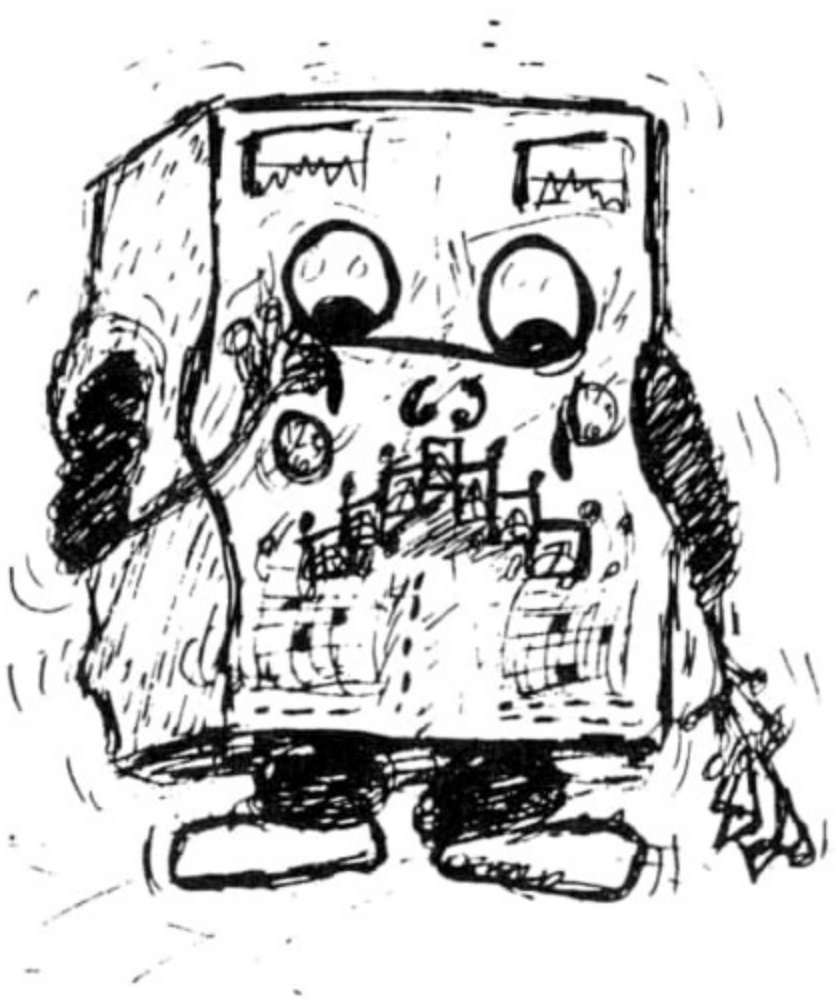
D--DIST: input hours and average miles per hour, output is distance traveled

E--TIME: input average speed and miles traveled, output is travel time

```
  10 PRINT TAB 12;"TRAVEL";TAB 1
1;"COMPUTER";AT 5,4;"A---MPG";TA
B 19;"B---RANGE";AT 9,12;"C---MP
H";AT 13,4;"D---DIST";TAB 19;"E-
--TIME";AT 17,10;"SELECT ONE"
  20 LET C$=INKEY$
  30 IF INKEY$="" THEN GOTO 20
  40 LET C=CODE C$-37
  50 CLS
  60 GOSUB C*100
  70 PAUSE 300
  80 CLS
  90 GOTO 10
 100 PRINT AT 2,0;"MILES: ?","GA
LLONS: ?"
 110 INPUT M
 120 PRINT AT 2,7;M;
 130 INPUT G
 140 PRINT TAB 25;G;AT 4,0;
 150 LET MPG=M/G
 160 PRINT "MILES PER GALLON: ";
MPG
 170 RETURN
 200 PRINT AT 2,0;"MPG: ?","GALL
ONS: ?"
 210 INPUT MPG
 220 PRINT AT 2,5;MPG;
 230 INPUT G
 240 PRINT TAB 25;G;AT 4,0;
 250 LET R=MPG*G
 260 PRINT "RANGE: ";R;" MILES"
 270 RETURN
 300 PRINT AT 2,0;"MILES: ?","HO
URS: ?"
 310 INPUT M
 320 PRINT AT 2,7;M;
 330 INPUT H
 340 PRINT TAB 23;H;AT 4,0;
 350 LET S=M/H
 360 PRINT "AVERAGE SPEED: ";S;"
 MPH"
 370 RETURN
```

```
 400 PRINT AT 2,0;"AVG SPEED:  ?"
,"HOURS:  ?"
 410 INPUT S
 420 PRINT AT 2,11;S;
 430 INPUT H
 440 PRINT TAB 23;H;AT 4,0;
 450 LET D=S*H
 460 PRINT "TRAVEL DISTANCE:  ";D
;"  MILES"
 470 RETURN
 500 PRINT AT 2,0;"AVG SPEED:  ?"
,"MILES:  ?"
 510 INPUT S
 520 PRINT AT 2,11;S;
 530 INPUT M
 540 PRINT TAB 23;M;AT 4,0;
 550 LET T=M/S
 560 PRINT "TRAVEL TIME:  ";T;"  H
OURS"
 570 RETURN
```

Try running this program next time you are planning a car trip and you will get an idea of how much time will be spent on the road, average speed, fuel consumed and other travel data.



...and then they said I was being replaced by a Timex/Sinclair 1000...

Data Handling

BUDGET MINDER

It seems that a lot of people I know go through the month paying their bills and then wondering where all of their money went to. At times like this it is nice to have a budget planned out. Here is a short, yet powerful program that shows where each dollar of a particular income is spent. It features a special user prompt to show which data is expected. The income is entered first. Then the budgeted amount for each category is entered when the cursor shows it is expected. You may enter multiple figures for any category. Ex: 12+32+29 The figure printed for that category would be $78. The program also keeps a running total of the dollars remaining after each expense. Here is the listing:

```
  10 REM BUDGET MINDER
  20 REM (C)1984 BOB JOHNSON
  30 GOSUB 180
  40 PRINT AT 3,10;"▒"
  50 INPUT M
  60 PRINT AT 3,10;"$";M
  70 FOR I=1 TO 6
  80 PRINT AT X,Y;"▒"
  90 INPUT N
 100 GOSUB 130
 110 NEXT I
 120 STOP
 130 LET M=M-N
 140 PRINT AT X,Y;"$";N;AT 21,3;
"$";M;" REMAINING          "
 150 LET X=X+2
 160 IF M<=0 THEN STOP
 170 RETURN
 180 LET X=7
 190 LET Y=22
 200 PRINT AT 1,8;"BUDGET MINDER
";AT 3,3;"INCOME:";AT 5,3;"▒▀▒▀▒▀
▀▒▀▒▀▒▀▒▀▒▀▒▀▒▀▒▀▒▀▒";AT 7,3;"RE
NT/MORT";AT 9,3;"UTILITES";AT 11
,3;"TRANSPORTATION";AT 13,3;"GRO
CERIES";AT 15,3;"INSURANCE";AT 1
7,3;"OTHER";AT 19,3;"▀▄▀▄▀▄▀▄▀▄▀▄
▄▀▄▀▄▀▄▀▄▀▄▀▄▀"
 210 RETURN
```

Numbers are entered without the dollar sign and whole dollars may be rounded off. $14.62 would be entered as 14.62. $29.00 would be 29. After the program is run this is what it would look like. I know it can't be mine because there is still some money remaining!!

BUDGET MINDER

INCOME: $1200

RENT/MORT	$375
UTILITES	$178.62
TRANSPORTATION	$143
GROCERIES	$112
INSURANCE	$85.27
OTHER	$154

$152.11 REMAINING

DIRECTORY

Would you like to type a name into your computer and have it display that name with the proper address and phone number? This program can do that. You can search for one name or print out the entire list. There is even a provision for saving the latest updated version of the list on tape.

```
  10 REM DIRECTORY
  20 DIM A$(16,30)
  30 DIM B$(30)
  40 LET A=1
  50 PRINT AT 2,13;"THE";TAB 11;
"DIRECTORY";AT 7,5;"W-WRITE";TAB
 16;"T-SAVE ON TAPE";AT 11,5;"S-
SEARCH";TAB 16;"P-PRINTOUT";AT 1
5,5;
  60 LET S$=INKEY$
  70 IF INKEY$="" THEN GOTO 60
  80 IF S$="W" THEN GOTO 120
  90 IF S$="T" THEN GOTO 430
 100 IF S$="S" THEN GOTO 280
 110 GOTO 200
 120 PRINT "ENTER NAME-ADDRESS-P
HONE"
 130 INPUT A$(A)
 140 INPUT A$(A+1)
 150 INPUT A$(A+2)
 160 INPUT A$(A+3)
 170 LET A=A+4
 180 CLS
 190 GOTO 50
 200 CLS
```

```
 210 FOR I=1 TO 13 STEP 4
 220 PRINT A$(I),A$(I+1),A$(I+2)
 A$(I+3)
 230 PRINT
 240 NEXT I
 250 PAUSE 500
 260 CLS
 270 GOTO 50
 280 CLS
 290 PRINT "  SEARCH MODE  ";AT 2,
0;"ENTER NAME: ";
 300 INPUT B$
 310 PRINT B$;AT 4,0;"<SEARCHING
>";AT 5,0;
 320 FOR I=1 TO 13 STEP 4
 330 IF A$(I)=B$ THEN GOTO 390
 340 NEXT I
 350 PRINT B$," NOT FOUND"
 360 PAUSE 200
 370 CLS
 380 GOTO 50
 390 PRINT A$(I),A$(I+1),A$(I+2)
,A$(I+3);AT 12,0;"PRESS ENTER FO
R MENU"
 400 PAUSE 4E4
 410 CLS
 420 GOTO 50
 430 SAVE "DIRECTORY"
 440 GOTO 50
```

In this version of the program there is memory space reserved for only four names, but it is easy to expand. It will fit into 2K of memory, but to be really practical a 16K RAM pack must be used. With the expanded memory 100 or more names can be easily stored.

Before you modify the storage capacity you must decide how many names you want to store. As an example, we can make room for 25 names. Each name needs four lines so 25 times 4 is 100 lines. If each line has 30 characters then in line 20 you would write DIM A$(100,30). The printout routine loops in steps of four and prints the entire list of names. Line 210 controls the number of times the routine loops. The 13 is the number of the first line of the fourth name in A$. Since there is now 100 names the 13 must be changed to the first line of the 25th name--97. Line 320, in the search routine, is changed in the same manner. If you have space for 25 names and have stored fewer than that, then you need a way to break out of the print routine when you get to the last name.

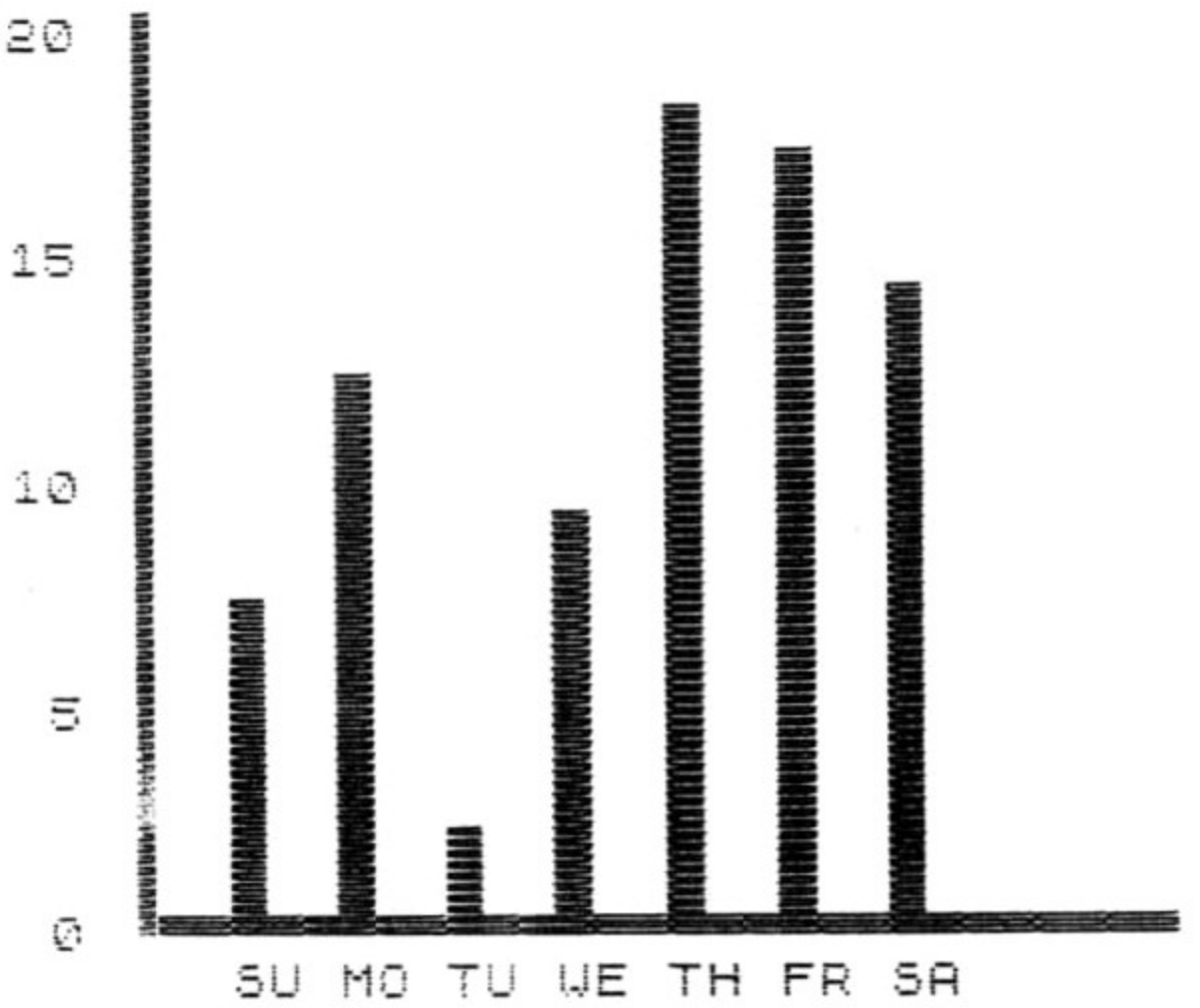
Type in: 215 IF A$(I,1) 1 THEN GOTO 250
With only four names it was possible to print them all onto the screen. But with 25 names you would get a full screen and the program would stop with a report code 5. To handle this problem, simply insert four SCROLL statements between lines 215 and 220. For a lineprint output if you wish to have a hardcopy:

```
210 FOR I=1 TO 13 STEP 4
220 LPRINT A$(I),A$(I+1),A$(I+2
),A$(I+3)
230 LPRINT
240 NEXT I
```

IMPORTANT: The first time this program is run, names may be written into it. It is not necessary to write all the names on the first run. When you finish each session of entering names, be sure to select "T" from the menu and save the program on tape. New names may be added each time, but only if the space for names is not full! The program will start running after it is loaded. If you break out or somehow stop then start again with GOTO 50, since RUN will erase all the names. Once the list is full it only needs to be saved one final time and that version is the completed directory.

GRAPHICS

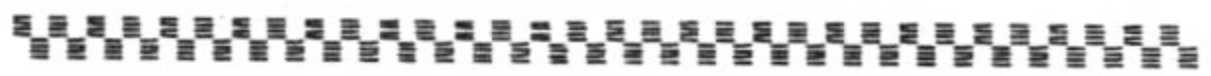
Data that is output in numerical form is suitable for most applications. However, there are times when a graphic display of information is more meaningful. Bar charts, pie charts and graphs are common ways to represent data. Here is an example. Suppose that last week at Shaun's Car Wash from Sunday through Saturday they washed 7,12,2,9,18,17 and 14 cars. Did you get all of that? Okay, here is a bar chart with the same information.



Using the bar chart, you can quickly see which were Shaun's busy days and which were slow. To write a graph routine requires that you plan a layout of the graph before you start. I use regular graph paper. Then write down exactly what you want to print and where you want to print it. The next step is to write a program sequence that will set up the graph by printing any borders, labels or values. After that is working, you can write the routine to print the bars or whatever you are using to represent data. The data used may be stored in the program or input by the user. This is the program that produced the chart above.

```
 10 REM BAR CHART
 20 FOR I=0 TO 19
 30 PRINT AT I,3;" ■"
 40 NEXT I
 50 PRINT AT 20,3;" ■"
 60 FOR I=4 TO 31
 70 PRINT AT 20,I;"■"
 80 NEXT I
 90 PRINT AT 0,0;20;AT 5,0;15;A
T 10,0;10;AT 15,1;5;AT 20,1;0;AT
 21,6;"SU MO TU WE TH FR SA"
100 FOR I=6 TO 24 STEP 3
110 INPUT N
120 IF N=0 THEN GOTO 160
130 FOR J=19 TO 20-N STEP -1
140 PRINT AT J,I;"■■"
150 NEXT J
160 NEXT I
```

Lines 20-50 print the vertical border, while 60-80 prints the horizontal part of the border. Line 90 print the scale of values and the labels for each weekday. The loop in 100-160 sets the print position for each bar. The loop in 130-150 prints each bar in steps that are determined by the input variable, N. If N=0 then the printing loop is bypassed by line 120. This short routine is an example of setting up and printing a graph for one particular application. This is only a sample. Other graphics characters can also be used. Plot statements will give you greater resolution or range of increments. And finally, different formats, such as horizontal bar, curve plotting or plotting points on a graph will give you the flexibility you need to give a pleasing visual effect to your display of data.



To find out program length:

```
PRINT PEEK 16396+256*PEEK 16397
-16509
```

To POKE a character into the display without changing the print position:

```
POKE (PEEK 16396+256*PEEK 16397+
33*X+Y+1)
```

X=Row Y=Column The number POKEed is the CODE of the character you want

To PRINT to all 24 lines:

```
POKE 16418,0
```

Note: Do not use INPUT or SCROLL while in this mode.

Games

My collection of software includes conversion programs, graph plotting, financial planning and games. When a friend, neighbor or relative drops by and wants to see the computer in action I load in a program and let it run. Most people are fairly impressed to see the computer balance my checkbook and draw a chart, but what really catches their attention is my collection of games. People really do enjoy trying their hand at games of strategy, dexterity or luck. These following programs should provide all of those.

FLIPPO

When a program requires an element of chance the T/S 1000 uses its random number generator. This amazing feature produces a number between zero and one. EX: .0011291504. That can be used to make a new number of any value. To produce a random number from 1 to 10 use this BASIC expression:

INT(RND * 10+1)

This first game uses the random number generator to similate the flipping of a coin. When the computer flips its coin then you must guess the result. The computer's response is tailored to the accuracy of your guess.

```
I AM FLIPPING MY COIN
CALL IT, HEADS OR TAILS?
WRONG GUESS, YOU BONEHEAD.
WANT TO PLAY AGAIN?
OKAY THEN, BYE
```

```
  10 PRINT TAB 6;"*** FLIPPO ***
"
  20 PRINT AT 4,0;"THE GREAT COI
N FLIP GAME"
  30 PRINT
  40 PRINT "DO YOU WANT INSTRUCT
IONS?"
  50 INPUT I$
  60 IF I$(1)="Y" THEN GOSUB 310
  70 PRINT
  80 CLS
  90 PRINT "I AM FLIPPING MY COI
N"
 100 LET F=INT (RND*2)
 110 IF F=0 THEN LET C$="HEADS"
 120 IF F=1 THEN LET C$="TAILS"
 130 PRINT
 140 PRINT "CALL IT, HEADS OR TA
ILS?"
 150 INPUT G$
 160 IF G$=C$ THEN GOSUB 250
 170 IF G$<>C$ THEN GOSUB 280
 180 PRINT
 190 PRINT "WANT TO PLAY AGAIN?"
 200 INPUT P$
 210 IF P$(1)="Y" THEN GOTO 80
 220 PRINT
 230 PRINT "OKAY THEN, BYE"
 240 STOP
 250 PRINT
 260 PRINT G$;", YOU ARE CORRECT
"
 270 RETURN
 280 PRINT
 290 PRINT "WRONG GUESS, YOU BON
EHEAD."
 300 RETURN
 310 PRINT
 320 PRINT "I WILL FLIP MY C.IN
AND YOU"
 330 PRINT "GUESS IF IT IS HEAD
 OR TAILS."
 340 PAUSE 300
 350 RETURN
```

The game starts out printing the title and asks if instructions are desired. If the first letter of the answer is a "Y" then instructions are given. In line 100 the random number is generated. The coin result is based on this. Lines 160 and 170 input the guess and compare it to the coin. At this point the program branches to a response that tells if your guess is correct or not. The last section asks if you want to play again. If the answer is yes-"Y"-then the program jumps to line 80 and runs again, otherwise it terminates.

STARSEEKER

In esoteric circles a Starseeker is someone who seeks knowledge of lifes mysteries. One of the Starseekers abilities is that of clairvoyance. That is the ability to detect events by means other than the physical senses. This program gives the Starseeker a random number from 1 to 200 to guess at, a wide enough range to test for ESP or plain luck.

```
  5 PRINT TAB 10;"****"
 10 PRINT TAB 5;"* STARSEEKER *
 15 PRINT TAB 10;"****"
 20 PRINT AT 5,0;"DO YOU WANT I
NSTRUCTIONS?"
 25 INPUT I$
 30 IF I$="YES" THEN GOSUB 200
 35 CLS
 40 PRINT "I AM THINKING OF A N
UMBER."
 45 PAUSE 180
 50 LET I=1
 55 LET N=INT (RND*200+1)
 60 SCROLL
 65 PRINT "WHAT IS YOUR GUESS?"
 70 INPUT G
 75 SCROLL
 80 IF G>N THEN PRINT G;"--TOO
HIGH"
 85 IF G<N THEN PRINT G;"--TOO
LOW"
 90 IF G=N THEN GOSUB 300
 95 LET I=I+1
100 GOTO 60
200 PRINT AT 5,0;"I WILL THINK
OF A NUMBER "
210 PRINT "BETWEEN ONE AND TWO
HUNDRED."
220 PRINT "YOU, STARSEEKER MUST
 GUESS"
230 PRINT "THAT NUMBER."
240 PAUSE 400
250 RETURN
300 PRINT
305 SCROLL
310 PRINT N;"--CORRECT. YOU GUE
SSED MY"
315 SCROLL
320 PRINT "NUMBER IN ";I;" GUES
SES"
325 SCROLL
330 PRINT "DO YOU WANT TO PLAY
AGAIN?"
340 INPUT P$
350 IF P$="YES" THEN GOTO 35
355 SCROLL
360 PRINT
365 SCROLL
370 PRINT "BYE"
380 STOP
```

Like in FLIPPO, this game asks if instructions are desired, and then provides them if wanted. In line 50, the variable "I" is set at one and keeps track of the number of guesses made. The target number is produced in line 55. Lines 60-100 ask for the guess, reports if high or low, increments I and if the guess matches the number it jumps to line 300. Then the Starseeker is told the number and the number of guesses it took to find it. The last portion of the program asks if a new game is wanted then runs again or terminates.

Anywhere from 8 to 12 guesses to find the number is average. 4 to 7 guesses is pretty good and 3 or fewer would indicate luck...or perhaps something more.

TREASURE SEARCH

This game makes use of a playing board and graphics characters. The object is to use the arrow (⇦ ⇧ ⇩ ⇨) keys to move your search cursor around to find the treasure that the computer has hidden on the screen. You are guided by the meter at the right, which indicates if the last move brought you closer or farther from the treasure. The program just squeezes into 2K when it is run. If only 1K is available then deleting the instructions and using only half of the screen for the board should trim it down to size.

```
 10 RAND
 20 LET X=INT (RND*19+1)
 30 LET Y=INT (RND*29+1)
 40 PRINT "<><>TREASURE SEARCH<
><>"
 50 PRINT AT 5,0;"HI, WHAT IS Y
OUR NAME?"
 60 INPUT N$
 70 CLS
 80 PRINT "I WILL HIDE TREASURE
 SOMEWHERE"
 90 PRINT "ON THE SCREEN BORDER
ED BY STARS."
 100 PRINT "USE THE ARROW KEYS T
O MOVE"
 110 PRINT "YOUR FINDER. THE MET
ER AT THE"
 120 PRINT "RIGHT WILL SHOW YOUR
 PROXIMITY"
 130 PRINT "TO THE TREASURE:"
 140 PRINT TAB 12;"C=CLOSER F=FA
RTHER"
```

```
150 PRINT
160 PRINT "PRESS ENTER TO BEGIN
"
170 IF INKEY$="" THEN GOTO 170
180 CLS
190 FOR I=0 TO 30
200 PRINT AT 0,I;"*"
210 PRINT AT 20,I;"*"
220 NEXT I
230 FOR I=1 TO 19
240 PRINT AT I,0;"*"
250 PRINT AT I,30;"*"
260 NEXT I
270 LET A=11
280 LET B=14
290 LET N=11
300 LET P1=ABS (A-X)+ABS (B-Y)
310 PRINT AT 9,31;"C"
320 PRINT AT 13,31;"F"
330 PRINT AT A,B;"▓"
340 PRINT AT N,31;"▓"
350 PRINT AT 21,0;"HAPPY HUNTIN
G"
360 LET R$=INKEY$
370 IF R$="" THEN GOTO 360
380 PRINT AT A,B;" "
390 IF CODE R$=35 AND A>1 THEN
LET A=A-1
400 IF CODE R$=34 AND A<19 THEN
 LET A=A+1
410 IF CODE R$=33 AND B>1 THEN
LET B=B-1
420 IF CODE R$=36 AND B<29 THEN
 LET B=B+1
430 IF A=X AND B=Y THEN GOTO 52
0
440 PRINT AT N,31;" "
450 LET P2=ABS (A-X)+ABS (B-Y)
460 IF P1>P2 THEN LET N=10
470 IF P1<P2 THEN LET N=12
480 LET P1=P2
490 PRINT AT A,B;"▓"
500 PRINT AT N,31;"▓"
510 GOTO 360
520 PRINT AT 21,0;"YOU FOUND IT
";N$
530 PRINT AT A,B;"X"
540 FOR I=1 TO 10
550 NEXT I
560 PRINT AT A,B;"▓"
570 FOR I=1 TO 10
580 NEXT I
590 IF INKEY$="" THEN GOTO 530
600 CLS
610 GOTO 10
```

There are five main sections of this program:

10-30 Hides the "Treasure"

40-170 Introduction and Instructions

180-350 Sets Variables and Draws the Board

360-510 Playing the Game
(move finder, check for treasure, adjust the meter)

520-590 Final Routine for Found Treasure

The variable P1 stores the difference in position between the finder and the treasure. P2 stores the previous value of P1. The computer can determine if a move is toward or away from the treasure by comparing P1 and P2. Then the meter can show the result. When you have found the treasure, the program executes a routine that is a closed loop. To break out, just touch any key and line 590 will release you. The game will restart automatically. Type in this program and grab your pick and shovel. Happy Hunting!

STAR DODGER

Here is an arcade type game that will demonstrate the use of machine code. Using the ⇧ and ⇩ keys, you maneuver your spaceship among a cluster of stars, trying to avoid a fatal crash. If you make it through the cluster without hitting a star then the game ends. To start writing this program we are going to write the machine code into a REM statement in the first line. First, type in this:

```
5 REM 12345678901234567890
```

Then, starting at address 16514, POKE in these numbers: 1,118,25,42,16,64,22,0,43,126,185,40, 4,114,87,24,247,16,243,201

The REM statement will now look like this:

```
    5 REM ■
6442 - F/ RUN  ( NEXT TAN
```

Now, type in the rest of the program.

```
  10 LET X=10
  20 FOR I=1 TO 120
  30 PRINT AT X,0;
  40 IF PEEK (PEEK 16398+256*PEE
K 16399)=23 THEN GOTO 120
  50 PRINT ">";
  60 LET X=X+(INKEY$="6" AND X<1
5)-(INKEY$="7" AND X>1)
  70 IF I<100 THEN PRINT AT RND*
16,20;"*"
  80 LET Z=USR 16514
  90 NEXT I
 100 PRINT AT 10,9;"YOU MADE IT"
;AT X,0;">"
 110 STOP
 120 PRINT "▒ BOOM";
```

The machine code stored in the REM statement scrolls the screen contents from right to left. The USR function in line 80 causes this routine to execute starting at 16514. The variable in line 10 sets the vertical position of the ship on the screen. Lines 20-90 make up a loop that is the main part of the program. During each loop, the program sets the print position of the ship and checks if a star is there. If so, it jumps to a crash message in line 120. Otherwise, the ship is printed. In the next line the print position is reset if the or is pressed. If the program has looped fewer than 100 times a new star is generated in the right of the screen. Then the machine code scroll is called by the USR function. This completes the loop. The last 21 passes through the loop make no new stars but scrolls the remaining ones off screen. If you succeed in dodging all of the stars, then the loop ends and you get the message in line 100.

You can see how powerful machine language is. This example is only a subroutine of a program written in BASIC. It is possible to write entire programs in machine language. For further reading on this fascinating subject I suggest the book MASTERING MACHINE CODE ON YOUR ZX-81 by Toni Baker.

NOTES