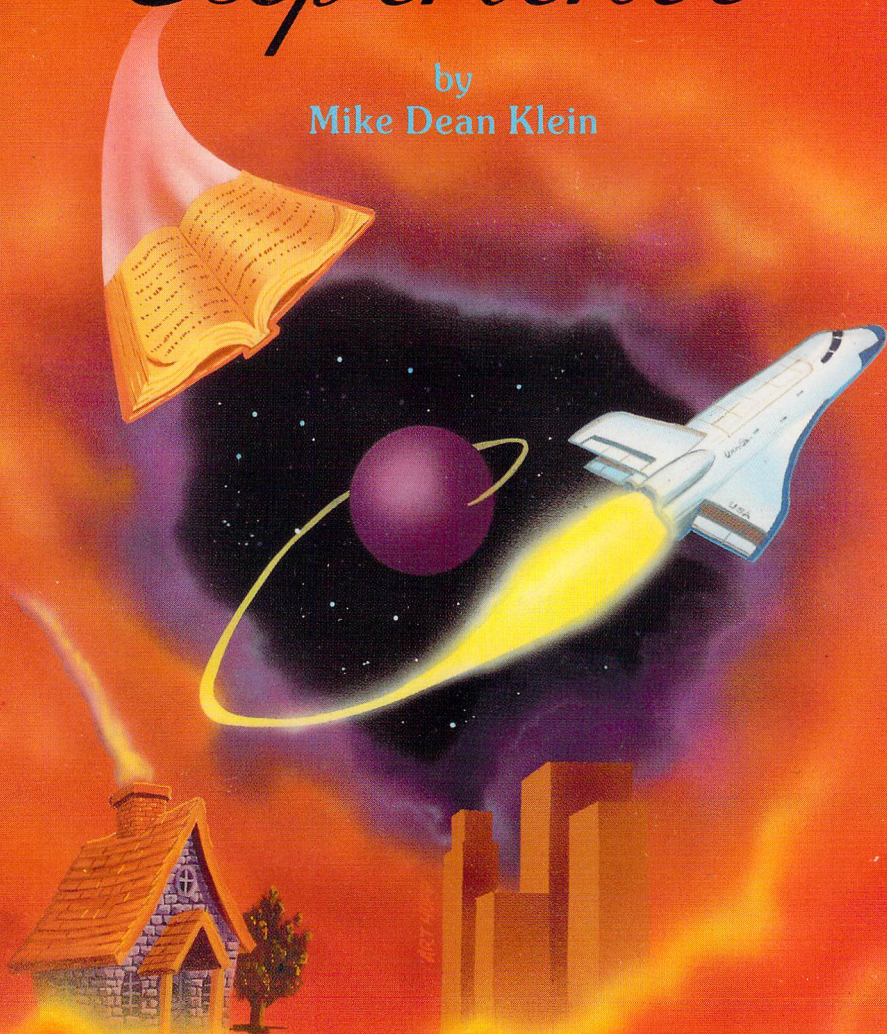




\$14.95

The
VIC-20
Experience

by
Mike Dean Klein



**Experience the many worlds of Commodore BASIC.
These easy-to-follow programs cover home, educational,
entertainment, business, and utility applications.**

The
VIC-20
Experience

By
Mike Dean Klein

Edited by
William B. Sanders, Ph.D.

Illustrations by
Tes Bonnarith



20660 Nordhoff Street, Chatsworth, CA 91311-6152
(818) 709-1202



ISBN 0-88190-296-9

**Copyright © 1984 By DATAMOST, Inc.
All Rights Reserved**

This manual is published and copyrighted by DATAMOST, Inc. Copying, duplicating, selling or otherwise distributing this product is hereby expressly forbidden except by prior written consent of DATAMOST, Inc.

The word VIC and the VIC-20 logo are registered trademarks of COMMODORE BUSINESS MACHINES, INC.

COMMODORE BUSINESS MACHINES, INC. was not in any way involved in the writing or other preparation of this manual, nor were the facts presented here reviewed for accuracy by that company. Use of the term VIC-20 should not be construed to represent any endorsement, official or otherwise, by COMMODORE BUSINESS MACHINES, INC.

Printed in U.S.A.

ACKNOWLEDGEMENTS

This book is dedicated to all of those people who decide for themselves what they want to do with the various aspects of their lives, and don't let others or family tradition sway them.

The following is a list of people who helped either directly or indirectly with the production of this book:

Joel Johnstone
Mitch Evans
Kevin Kelly
Bill Parker
Eric Goetz
Guy Grotke
Don Klein
June Klein
David Klein
Dean Garrett
Mr. Jose Weill
Dr. William B. Sanders

TABLE OF CONTENTS

Editor's Introduction 7

Programs for the Home 17

Recipe Book	19
Shopping List	25
Phone Book	29
Kitchen Metrics	37
Budget Helper	41

Educational Programs 47

Learning the States	49
Learning the Capitals	55
Math Quizzer	61
Divisors	65
Trig Helper	67
Primes	69
Spelling Practice	71
Foreign Language Study	75
Graph Plotter	81

Entertainment Programs 87

Death Hunt	89
Guided Missile Commander	93
Music Machine	99
Tank Command	103
Memory Master	109
Pick a Number	113
Doodle	117

Business Programs 123

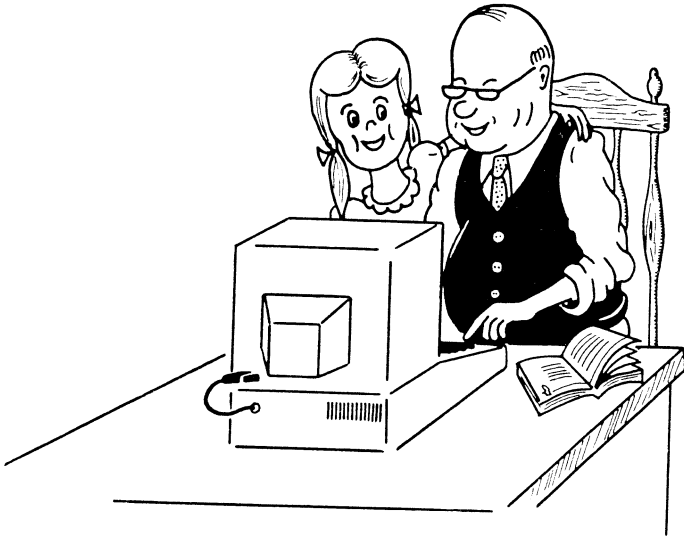
Appointment Scheduler	125
Bar Graph	131
Cash Flow	135
Interest Rate	139
Checkbook	143
Inventory	147

Utilities and Programming Aids ... 155

Character Creator	159
Sequential File Reader	163
Menu Ideas	165
Alarm Clock	169
File Restorer	173

Programming Notes 177

Coupon for Disk 183





EDITOR'S INTRODUCTION

William B. Sanders, Ph. D.

How many times have you heard someone say, "But what can you *do* with your computer?" This is an important question for both new owners of computers and those who are considering the purchase of their first home computer. For those who have a VIC-20, that question is not limited by the number of things one can do, but rather by the exhaustive list of applications!

There are two ways to make your computer operate. One way is to buy programs and run them on your computer. Just let them do their magic, sit back, and watch. The other way is to learn how to write and program your own software. The first method can be somewhat expensive, but you have access to programs that do everything from word processing to playing games. The second method can be a lot of fun, but it takes time for you to learn to program your computer to do exactly what you want it to.

This book represents a third alternative. It helps you learn how to write programs, and at the same time, provides you with programs you can use right away. If you are not interested in learning how to program, just key in the programs as they appear in this book, and then SAVE them to tape or disk. When you want to use them. Just LOAD and RUN them, and they will happily perform. On the other hand, if you want to learn how to program at the same time you are copying application programs, this book will show you many programmer's tricks. The best way to learn something is to be shown a useful application, and the programs in this book do just that.

The VIC-20 Experience was designed to provide a wide range of applications which can be of use to almost everyone. It is arranged into five distinct sections:

- I HOME
- II EDUCATION
- III ENTERTAINMENT
- IV BUSINESS
- V UTILITIES and PROGRAMMING AIDS

The first part is a collection of programs that have a wide variety of applications in the HOME. Domestic chores, such as cooking, are made simpler with a Recipe Book program that will give you immediate access to your favorite recipe. (It won't fix dinner . . . yet.) For EDUCATIONAL applications, you will find several programs to help children learn everything from their Math to the Capitals of the various states. Third, the ENTERTAINMENT section has a number of games you can play with your computer, as well as a drawing program to create designs and pictures in high resolution (also referred to as hires) graphics. For the businessperson, there are several programs to aid in making decisions and planning finances. If you would like to see stock market graphs, the Business Graph program will make annual graphs for you. Finally, for the programmer, there are several UTILITY and PROGRAMMING AID programs. "Utilities" are programs that help you program. You want to write a graphics game, but don't like the standard character set? The Character Creator will help you design your own graphic character set. Several other utilities will help the programmer both in writing programs and in understanding how to program.

As you can see, *The VIC-20 Experience* has a wide variety of computer applications. What's more, you can modify the programs to suit your own particular needs and tastes. All of the programs are clearly presented and explained, and while they will work perfectly well as they are presented, there is no mystery if you want to change them.

The author of *The VIC-20 Experience*, Mike Klein, is a seasoned programmer not only on the VIC-20, but on other microcomputers as well. With an eye on both clarity and practicality,

Mike created this collection of programs so that they could be used and understood by both novice and experienced programmers.



Section 1

Programs for the Home

- Recipe Book
- Shopping List
- Phone Book
- Kitchen Metrics
- Budget Helper





PROGRAMS FOR THE HOME

There are many ways to use a computer at home. Obviously, it is possible to use several of the programs under EDUCATION, ENTERTAINMENT, and BUSINESS in the home, but in designing programs for the home, I was somewhat stumped. After all, what exactly is “the home”? Does it mean entertainment for the kids or help for a housewife? Does it signify business programs for fathers and mothers? Well, the first and last are already covered, so I chose to help the person in between. The programs I thought would be the most useful around the house are a Phone Book (also a handy program for your business), a Recipe Book (to keep track of recipe data), and a Shopping List program (prints out a weekly shopping list when items are entered into it). Finally, Kitchen Metrics is provided for figuring out metric notations, and Budget Helper will assist your children in keeping tabs on their allowances.





Recipe Book

I'm not sure why, but one of the programs I enjoyed writing most was the Recipe Book program. Maybe it's because it does a lot and was easy to write. After all, one does gain a certain amount of satisfaction from a program that performs a particular task and works correctly "to boot." With Recipe Book, you have the ability to write a recipe, save it to disk, and recall it any time you need to review its ingredients. There are four basic commands in the menu:

- 1> EDIT INGREDIENTS
- 2> WRITE RECIPE
- 3> READ RECIPE
- 4> QUIT



Now, if you want to enter your recipe, type a 1; a second menu will appear:

- <1> LIST INGREDIENTS
- <2> DEL. INGREDIENTS (Del. stands for Delete.)
- <3> ADD INGREDIENTS
- <4> RETURN TO MAIN MENU

Now enter a 1 to give a list of all the ingredients that you have entered so far. If you wish to delete a particular ingredient, make a note of its number on the list, and type a 2. Then enter the number of the item you wish to delete. Upon listing the ingredients again, you will find that it is gone from your list. Adding ingredients is done by entering a 3. Ingredients are entered one at a time, each one separated by a carriage return. When you have finished, simply enter a carriage return with nothing behind it. Entering a 4 will bring you back to the original menu.

RECIPE BOOK

```
100 PRINT CHR$(147) CHR$(144)
110 POKE 36879,25
120 DIM INGR$(30)
200 REM *****
210 REM * PRINT SCREEN *
220 REM *****
230 PRINT CHR$(147) TAB(5) "RECIPE BOOK"
240 PRINT
245 PRINT
250 PRINT "1> EDIT INGREDIENTS"
260 PRINT "2> WRITE RECIPE"
270 PRINT "3> READ RECIPE"
280 PRINT "4> QUIT"
290 PRINT
295 F$=""
300 INPUT "CHOOSE ONE: ";F$
```



```
310 F=VAL(F#)
320 IF F<1 OR F>4 THEN 230
330 ON F GOTO 430,1230,1430
340 PRINT CHR$(147) "BYE BYE..."
350 END
400 REM *****
410 REM * ENTER DATA *
420 REM *****
430 PRINT CHR$(147)
440 PRINT
445 PRINT
450 PRINT "MENU:"
455 PRINT
460 PRINT "<1> LIST INGREDIENTS"
470 PRINT "<2> DEL. INGREDIENTS"
480 PRINT "<3> ADD INGREDIENTS"
490 PRINT "<4> RETURN TO MENU"
500 PRINT
505 PRINT
510 F#=""
520 INPUT "WHICH: ";F#
530 F=VAL(F#)
540 IF F<1 OR F>4 THEN 430
550 ON F GOTO 630,830,1030,230
600 REM *****
610 REM * LIST INGREDIENTS *
620 REM *****
630 PRINT CHR$(147)
640 PRINT
645 PRINT
650 FOR T=1 TO INDR
660 PRINT "#";T;" ";INGR$(T)
670 IF T=19 THEN 720
680 NEXT
690 PRINT "THAT'S IT"
700 GOSUB 1630
710 GOTO 430
```

```

720 PRINT "PRESS A KEY"
730 GOSUB 1630
740 PRINT CHR$(19)
750 PRINT
760 PRINT
770 GOTO 680
800 REM *****
810 REM * DELETE INGREDIENTS *
820 REM *****
830 PRINT CHR$(147)
840 PRINT
845 PRINT
850 F$=""
860 INPUT "WHICH # TO DELETE:";F$
870 IF F$="" THEN 430
880 NUM=VAL(F$)
890 IF NUM<1 OR NUM>INGR THEN 430
900 FOR T=NUM TO INGR-1
910 INGR$(T)=INGR$(T+1)
920 NEXT
930 INGR=INGR-1
940 GOTO 430
1000 REM *****
1010 REM * ADD INGREDIENTS *
1020 REM *****
1030 PRINT CHR$(147)
1040 PRINT
1045 PRINT
1050 PRINT "INGR.#";INGR+1 " : ";
1060 INGR$=""
1070 INPUT INGR$
1080 IF INGR$="" THEN 430
1090 INGR=INGR+1
1100 INGR$(INGR)=INGR$
1110 GOTO 1050
1200 REM *****
1210 REM * WRITE RECIPE *
1220 REM *****

```



```
1230 PRINT CHR$(147)
1240 FILE$=""
1250 INPUT "RECIPE NAME:";FILE$
1260 IF FILE$="" THEN 230
1265 PRINT
1270 PRINT "WRITING OUT RECIPE"
1280 OPEN 15,8,15
1290 PRINT#15,"90:";FILE$
1300 CLOSE 15
1310 OPEN 4,8,4,"0:";FILE$;S,W"
1320 FOR T=1 TO INGR
1330 PRINT#4,INGR$(T)
1340 NEXT
1350 CLOSE 4
1360 GOTO 230
1400 REM *****
1410 REM * READ IN RECIPE *
1420 REM *****
1430 PRINT CHR$(147)
1440 FILE$=""
1450 INPUT "RECIPE NAME:";FILE$
1460 IF FILE$="" THEN 230
1470 PRINT
1480 PRINT "READING IN RECIPE"
1490 OPEN 4,8,4,"0:";FILE$;S,R"
1500 INGR=0
1510 INGR=INGR+1
1520 INPUT#4,INGR$(INGR)
1530 IF ST=0 THEN 1510
1540 CLOSE 4
1550 GOTO 230
1600 REM *****
1610 REM * GET KEY *
1620 REM *****
1630 GET R$ : IF R$="" THEN 1630
1640 RETURN
```





Shopping List

How many shopping lists look more like shredded beef than paper? All those nasty erasures and scribbles! With Shopping List, you can proofread your grocery items before they are printed out to your printer or to the screen. You can add and delete items from your list at will; enter a 1 to add items and enter a 2 to delete items. When you are adding items, simply enter a RETURN when you are through. When you delete items, remember the items' numbers in the list, and use that as a reference when the computer asks you the number of the item(s) to delete. The number of items the list will hold is up to you.



All you have to do is change the DIM statement at the beginning of the program to reflect the number of items you wish the program to contain. The entire list may be dumped to either the screen or the printer by selecting option 3 from the main menu. If you press 3, the program will ask you whether you want it on the <S>CREEN or the <P>RINTER. Enter an S or a P, depending on which you prefer. It's as easy as that! With your shopping list, you will be assured that everything included on your list is readable, and you will save an extra trip to the market!

SHOPPING LIST

```
100 DIM ITEMS$(50)
110 ITEM=0
120 PRINT CHR$(147) CHR$(144)
130 POKE 36879,25
200 REM *****
210 REM * PRINT SCREEN *
220 REM *****
230 PRINT CHR$(147);
235 PRINT TAB(6) "GROCERIES"
240 PRINT
245 PRINT
250 PRINT "MENU:"
260 PRINT
270 PRINT "<1> ADD TO LIST"
280 PRINT "<2> DELETE FROM LIST"
290 PRINT "<3> DUMP LIST"
300 PRINT "<4> QUIT"
310 GET A$ : IF A$="" THEN 310
320 A=VAL(A$)
330 IF A<1 OR A>4 THEN 310
340 ON A GOTO 430,530,730
350 PRINT CHR$(147) "BYE BYE...."
360 PRINT
370 PRINT
380 PRINT
```



```
390 END
400 REM *****
410 REM * ADD TO LIST *
420 REM *****
430 PRINT CHR$(147)
440 PRINT "ITEM#";ITEM+1;": ";
450 ITEM$(ITEM)=" "
460 INPUT ITEM$(ITEM)
470 IF ITEM$(ITEM)=" " THEN 230
480 ITEM=ITEM+1
490 GOTO 440
500 REM *****
510 REM * DELETE FROM LIST *
520 REM *****
530 PRINT CHR$(147)
540 A$=""
550 INPUT "DELETE ITEM#:";A$
560 IF A$="" THEN 230
570 A=VAL(A$)
580 IF A<0 OR A>ITEM THEN 230
590 FOR T=A-1 TO ITEM 1
600 ITEM$(T)=ITEM$(T+1)
610 NEXT
620 ITEM=ITEM 1
630 GOTO 230
700 REM *****
710 REM * TYPE OF DUMP *
720 REM *****
730 PRINT CHR$(147)
740 A$=""
750 PRINT "<P>PRINTER/<S>SCREEN:"
760 GET A$ : IF A$="" THEN 760
770 IF A$<>"P" AND A$<>"S" THEN 230
780 IF A$="S" THEN 230
800 REM *****
810 REM * PRINTER DUMP *
820 REM *****
830 OPEN 1,5
```

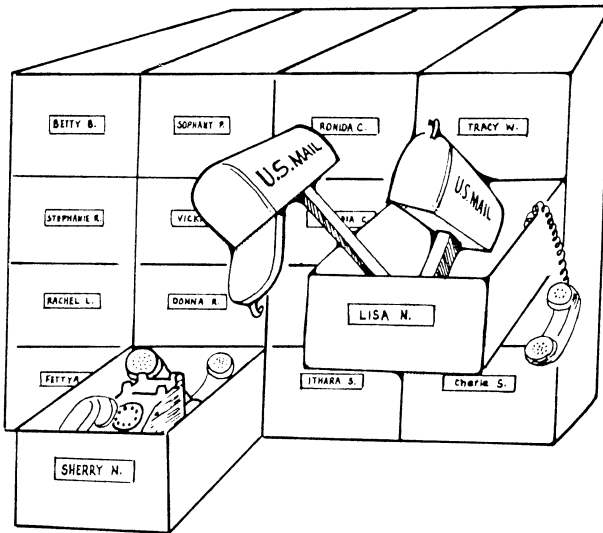
```
040 PRINT#1,"GROCERY LIST"
050 PRINT#1,CHR$(13),CHR$(13)
060 FOR T=0 TO ITEM-1
070 PRINT#1,ITEM$(T)
080 NEXT
085 CLOSE 1
090 GOTO 230
900 REM *****
910 REM * SCREEN DUMP *
920 REM *****
930 PRINT CHR$(147)
940 IF ITEM=0 THEN 1130
950 FOR T=0 TO ITEM-1
960 PRINT "ITEM#";T+1;";";ITEM$(T)
970 NEXT
980 PRINT
990 PRINT "HIT RETURN TO RESUME"
1000 GET R$ : IF R$(<>CHR$(13)) THEN 1000
1010 GOTO 230
1100 REM *****
1110 REM * NO ITEMS ENTERED *
1120 REM *****
1130 PRINT
1140 PRINT "NO ITEMS HAVE BEEN ENTERED!"
1150 GET R$ : IF R$="" THEN 1150
1160 GOTO 230
```



Phone Book

The Phone Book program is my personal favorite because I use it more than any other program. With it, you can enter names and numbers, delete them, add to them, and get printouts of the entire proceedings. Here is the menu you should see upon completion of the program:

- 1 > ADD ENTRY
- 2 > DELETE ENTRY
- 3 > DUMP ALL ENTRIES
- 4 > INIT. PHONE FILE
- 5 > QUIT



The last option is probably self-explanatory, but the others need some explaining. When the program is first run, you should initialize a Phone Directory File by typing a 4. It will then ask you if you are sure. Enter a Y if you are, or an N if you aren't. You should initialize a file only once, as any subsequent attempt to do so will erase all the old information that is in that file. You do not want that to happen. Anyway, once you have initialized the file, you may start entering your information via number 1. Typing in a 1 will display the following questions:

LAST NAME:
FIRST NAME:
AREA CODE:
PHONE#:

(In case you are wondering why there are no question marks with my input statements, it's because I opened a file to the screen, and that eliminates the little buggers!) When all the questions have been answered, you will be asked:

CORRECT? {Y/N/X}:X

Respond with a Y for yes, an N for no, and an X for exit. Again, hitting a RETURN immediately will put you back into the main menu. If you answer with a Y for yes, the drive will whirr for a bit (be patient, it's serial), and then, TAH DAH!, an entry has been made.

If you make a mistake after an entry has been made, no need to fear. Entering a 2 from the main menu will let you delete your entries. If you do decide to delete an entry, after pressing 2 the computer will ask you the phone number of the entry you wish to delete. After the phone number to be deleted is entered, the drive will go into action and your entry will no longer be present on future dumpings. When a printout of your entire file is required, enter a 3 from the beginning menu. Now the computer will respond with:

<P>RINTER/<S>CREEN?



Enter a P or an S, depending on whether you would like to direct output to the printer or to the screen. If either of these choices is selected, the contents of the phone directory file should start scrolling onto the screen or printer. VIC-owners with expanded memory in their units might wish to add a search and recall function to the program, which is a very handy routine to have.

PHONE BOOK

```
100 REM *****
110 REM * SET SCREEN *
120 REM *****
130 PRINT CHR$(147) CHR$(144)
140 PRINT CHR$(142) CHR$(8)
150 OPEN 1,0
160 C#=CHR$(13)
170 BK#=CHR$(157)
180 POKE 36879,25
200 REM *****
210 REM * PRINT SCREEN *
220 REM *****
230 PRINT CHR$(147) TAB(5) "PHONE BOOK"
235 PRINT
240 PRINT
245 PRINT
250 PRINT "MENU:"
255 PRINT
260 PRINT "1) ADD ENTRY"
270 PRINT "2) DELETE ENTRY"
280 PRINT "3) DUMP ALL ENTRIES"
290 PRINT "4) INIT. PHONE FILE"
300 PRINT "5) QUIT"
310 PRINT CHR$(19)
315 PRINT
320 PRINT
325 PRINT
330 PRINT TAB(5);
```

```

335 POKE 190,0
340 INPUT#1,A#
350 A=VAL(A#)
360 IF A<1 OR A>5 THEN 230
370 ON A GOTO 430,1530,930,1330
375 PRINT CHR$(147) "BYE BYE"
380 PRINT
385 PRINT
390 CLOSE 1
395 END
400 REM *****
410 REM * ENTER NAME & # *
420 REM *****
430 PRINT CHR$(147)
440 PRINT "LAST NAME:";
450 INPUT#1,LN#
460 PRINT
470 PRINT "FIRST NAME:";
480 INPUT#1,FI#
490 PRINT
500 PRINT "AREA CODE:";
510 INPUT#1,AC#
520 PRINT
530 PRINT "PHONE#:";
540 INPUT#1,PN#
550 PRINT
560 PRINT
565 PRINT
570 PRINT "CORRECT? (Y/N/X):X" DK#;
580 INPUT#1,A#
590 IF A#="Y" THEN 730
600 IF A#="N" THEN 430
610 GOTO 230
700 REM *****
710 REM * TRANSFER DATA *
720 REM *****
730 OPEN 4,9,4,"0:PHONE FILE.DAT,5,R"
740 OPEN 5,9,5,"0:COPY,5,W"

```




```
750 INPUT#4,W#,X#,Y#,Z#
760 IF W#="EOF" THEN PRINT#5,LN#C#FI#C#AC#C#
PN#
765 IF W#="EOF" THEN 790
770 PRINT#5,W#C#X#C#Y#C#Z#
780 GOTO 750
790 PRINT#5,W#C#X#C#Y#C#Z#
800 CLOSE 4
805 CLOSE 5
810 OPEN 15,8,15
820 PRINT#15,"50:PHONE FILE.DATA"
830 PRINT#15,"80:PHONE FILE.DATA=COPY"
840 CLOSE 15
850 GOTO 230
900 REM *****
910 REM * DUMP FILE *
920 REM *****
930 PRINT CHR$(147)
940 GOSUB 2030
950 IF 0#="0" THEN 1020
960 OPEN 2,4
970 PRINT#2,C#
980 FOR I=1 TO 7
990 PRINT#2," ";
1000 NEXT
1010 PRINT#2,"PHONE BOOK LISTING" C# C#
1020 ENTRIES=0
1030 OPEN 5,8,5,"0:PHONE FILE.DATA,S,R"
1040 INPUT#5,W#,X#,Y#,Z#
1050 IF W#="EOF" THEN 1140
1060 IF 0#="S" THEN 1090
1070 PRINT#2,"NAME : " W# "," X#
1080 PRINT#2,"PHONE#: (" Y# ")" Z# C#
1090 PRINT W# "," X#
1100 PRINT "(" Y# ")" Z#
1110 ENTRIES=ENTRIES+1
1120 PRINT
1130 GOTO 1040
```

```

1140 CLOSE 5
1145 CLOSE 2
1150 D$="END OF FILE DATA"
1160 IF ENTRIES=0 THEN D$="FILE IS EMPTY"
1170 PRINT
1175 PRINT D$
1180 PRINT
1185 PRINT ENTRIES "ENTRIES"
1190 IF PEEK(197)=64 THEN 1190
1200 GOTO 230
1300 REM *****
1310 REM * INITIALIZE FILE *
1320 REM *****
1330 PRINT CHR$(147);
1335 PRINT "ARE YOU SURE (Y/N):N" DK$;
1340 INPUT#1,A$
1350 IF A$<>"Y" THEN 230
1360 OPEN 15,0,15
1370 PRINT#15,"SO:PHONE FILE.DATA"
1380 CLOSE 15
1390 OPEN 5,0,5,"O:PHONE FILE.DATA,S,W"
1400 PRINT#5,"EOF" C$ "Z" C$ "Z" C$ "Z"
1410 CLOSE 5
1420 GOTO 230
1500 REM *****
1510 REM * DELETE ENTRY *
1520 REM *****
1530 PRINT CHR$(147)
1540 PRINT "# TO DEL:";
1550 INPUT#1,PN$
1560 PRINT
1565 PRINT
1570 PRINT "ARE YOU SURE(Y/N)";
1580 INPUT#1,A$
1590 IF A$="Y" THEN 1610
1600 GOTO 230
1610 PRINT CHR$(147)
1620 C=0

```



```
1630 OPEN 4,8,4,"0:PHONE FILE.DATA,S,R"
1640 OPEN 5,8,5,"0:COPY,S,W"
1650 INPUT#4,W$,X$,Y$,Z$
1660 IF Z$=PN$ THEN 1700
1670 PRINT#5,W$C$X$C$Y$C$Z$C$
1680 IF W$="EOF" THEN 1830
1690 GOTO 1650
1700 C=1
1710 PRINT W$ "," X$
1720 PRINT "(" Y$ ")" Z$
1730 PRINT CHR$(18) "DELETED"
1740 PRINT
1750 GOTO 1650
1800 REM *****
1810 REM * END OF FILE *
1820 REM *****
1830 CLOSE 4
1835 CLOSE 5
1840 OPEN 15,8,15
1850 PRINT#15,"S0:PHONE FILE.DATA"
1860 PRINT#15,"R0:PHONE FILE.DATA=COPY"
1870 CLOSE 15
1880 M$="END OF FILE"+C$+"ITEM NOT FOUND"
1890 IF C=1 THEN M$="FINISHED"
1900 PRINT
1905 PRINT
1910 PRINT M$
1920 GET A$ : IF A$="" THEN 1920
1930 GOTO 230
2000 REM *****
2010 REM * PRINTER OR SCREEN *
2020 REM *****
2030 PRINT CHR$(147) "<P>RINTER/<S>CREEN:S"
BK$;
2040 INPUT#1,0$
2050 IF 0$<>"P" AND 0$<>"S" THEN 2030
2060 PRINT CHR$(147)
2070 RETURN
```





Kitchen Metrics

Metrics!? Who knows, or even understands metrics? Sure, everyone knows that there are 100 centigrams in a gram, but who knows exactly how heavy 30 centigrams are? Around the kitchen, some means of converting metrics is needed. Voila! Kitchen Metrics is born. Now you too can convert metrics in the privacy of your own home.



The menu should look like this:

WHICH:

- 1 > QUARTS-LITERS
- 2 > LITERS-QUARTS
- 3 > OUNCES-GRAMS
- 4 > GRAMS-OUNCES
- 5 > FAHRENHEIT-CELSIUS
- 6 > CELSIUS-FAHRENHEIT

Simply type in the number of the conversion you would like, followed by a number. Suppose you would like to find out how many liters are in two quarts. Type a 1 to convert from quarts to liters. When it asks for units, enter a 2, for two quarts. Presto! Your two quarts have been converted into a metrics unit.

KITCHEN METRICS

```
100 PRINT CHR$(147) CHR$(144)
110 POKE 36979,25
200 REM *****
210 REM * PRINT SCREEN *
220 REM *****
230 PRINT CHR$(147) TAB(3) "KITCHEN METRICS"
240 PRINT
245 PRINT
250 PRINT "WHICH:"
260 PRINT
270 PRINT "1) QUARTS-LITERS"
280 PRINT "2) LITERS-QUARTS"
290 PRINT "3) OUNCES-GRAMS"
300 PRINT "4) GRAMS-OUNCES"
310 PRINT "5) FAHRENHEIT-CELSIUS"
320 PRINT "6) CELSIUS-FAHRENHEIT"
330 GET R$: IF R$="" THEN 330
340 R=VAL(R$)
350 IF R<1 OR R>6 THEN 330
400 REM *****
410 REM * GET UNIT & CONVERT *
420 REM *****
430 PRINT
435 PRINT
440 INPUT "UNIT: ";UNIT
450 ON R GOSUB 630,730,830,930,1030,1130
455 PRINT
460 PRINT "CONVERTED IT'S:" AM
```



```
470 PRINT
480 PRINT "ANY KEY TO CONTINUE"
490 POKE 197,64
500 IF PEEK(197)=64 THEN 500
510 GOTO 230
600 REM *****
610 REM * QUARTS-LITERS *
620 REM *****
630 AM=.95*UNIT
640 RETURN
700 REM *****
710 REM * LITERS-QUARTS *
720 REM *****
730 AM=1.06*UNIT
740 RETURN
800 REM *****
810 REM * OUNCES-GRAMS *
820 REM *****
830 AM=28.35*UNIT
840 RETURN
900 REM *****
910 REM * GRAMS-OUNCES *
920 REM *****
930 AM=.035*UNIT
940 RETURN
1000 REM *****
1010 REM * FAHRENHEIT-CELSIUS *
1020 REM *****
1030 AM=(UNIT-32)*5/9
1040 RETURN
1100 REM *****
1110 REM * CELSIUS-FAHRENHEIT *
1120 REM *****
1130 AM=(UNIT*9/5)+32
1140 RETURN
```





Budget Helper

Some people are not good at sports. I am not good at money. I've always been that way, unfortunately, but now there's something I can do about it, and by golly, there's something you can do about it too! Budget Helper is designed primarily for younger children to help them budget their allowances. Key in a 1 to enter your allowance. Then enter a number from 2-12. You will now be prompted for a string and an amount. Simply enter in your expense, such as GAS FOR CAR, then enter its cost, like 35. Now your current amount of allowance left to spend is shown on the screen, and you are ready to enter more expenses.



BUDGET HELPER

```
100 REM *****
110 REM * SET SCREEN *
120 REM *****
130 PRINT CHR$(147) CHR$(144)
140 POKE 36879,25
150 DIM I$(12),A(12)
200 REM *****
210 REM * PRINT TITLE *
220 REM *****
230 PRINT CHR$(147) TAB(4) "BUDGET HELPER"
240 PRINT
250 PRINT "1> START AMT:" IA
255 CA=0
260 FOR T=2 TO 12
270 CA=CA+A(T)
275 NEXT
280 CA=IA-CA
290 PRINT " AMT. LEFT:" CA
300 PRINT TAB(4) CHR$(19) "ITEMS AMOUNTS"
310 PRINT
400 REM *****
410 REM * PRINT EXPENSES *
420 REM *****
430 FOR T=2 TO 12
440 PRINT TAB(0) RIGHT$( " "+STR$(T),2) " ";
450 PRINT I$(T) ":" A(T)
460 NEXT
500 REM *****
510 REM * GET INPUT *
520 REM *****
530 PRINT
535 PRINT "WHICH(1-12):";
540 INPUT W$
550 W=VAL(W$)
560 IF W<1 OR W>12 THEN 230
570 IF W=1 THEN 730
```



```
600 REM *****
610 REM * ENTER EXPENSE DATA *
620 REM *****
630 PRINT
640 I$(W)=" "
645 A(W)=0
650 INPUT "ITEM NAME: "; I$(W)
660 INPUT "EXPENSE: "; A(W)
670 GOTO 230
700 REM *****
710 REM * ENTER INIT AMOUNT *
720 REM *****
730 PRINT
740 INPUT "INITIAL AMOUNT: "; IA
750 GOTO 230
```

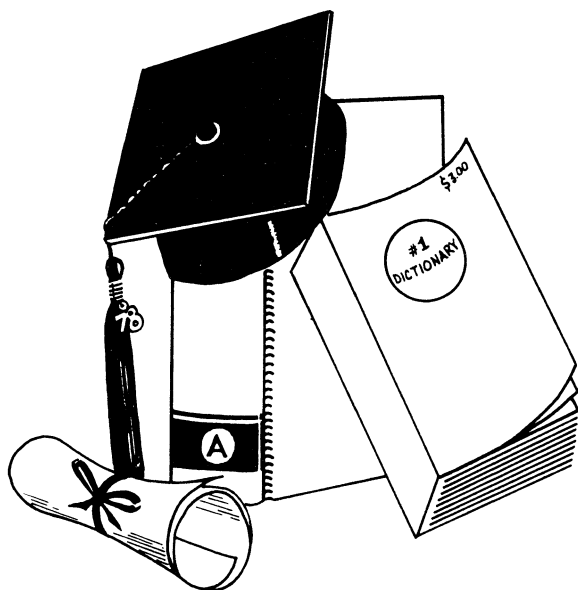


Section 2

Educational Programs

- Learning the States
- Learning the Capitals
- Math Quizzer
- Divisors
- Trig Helper
- Primes
- Spelling Practice
- Foreign Language Study
- Graph Plotter



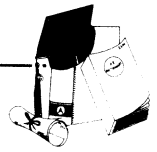


EDUCATIONAL PROGRAMS

Educational software is finally coming into its own. Today, a wide range of software is available for education. These programs are far more “learning oriented” than their predecessors. Included in the Education section of this book are five extremely useful programs for learning at different levels. The first is Learning the States, a program for memorizing the 50 states of the United States. The second is Learning the Capitals, a program for learning all 50 capitals. Using the Learning the Capitals program, I learned the capitals of all 50 states, something I could never seem to be able to do before. Math Quizzer will test a student on his addition, subtraction, multiplication and division. For more advanced math, there are programs for finding divisors, trigonometric functions and prime numbers. How many times have you wanted a list of prime

numbers? Well, now you can do it with Primes, a program that prints out all the prime numbers between two numbers input by the user. Spelling Practice helps students learn to spell lists of words for their spelling quizzes. Any list of words can be entered; then they will be flashed back momentarily for preparation. Finally, for students learning a foreign language, Foreign Language Study will help you remember all those new words. If you're planning a trip where only Spanish is spoken, or if you're trying to pass a vocabulary test, this program will be a patient tutor.

This section is good for learning programs by modification. As you will see, the States and Capitals programs make extensive use of DATA statements. By changing the contents of the DATA statements, students can learn other material as well. Of course, other aspects of the program will also have to be changed, but only slightly. Use your creativity and see what you can do.

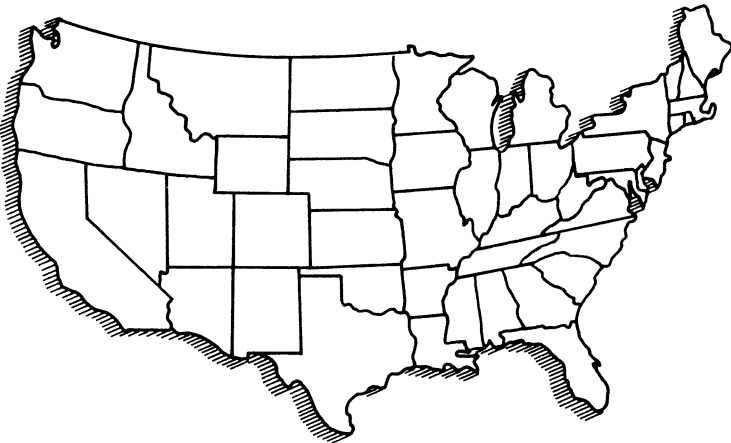


Learning the States

Almost everyone can recognize the individual states of the United States. However, when it comes to reciting all 50 of them from memory, that's where most people draw the line. Using this program, children can successfully memorize all 50 states, and have fun doing it. The program is quite simple. Here's how to use the menu:

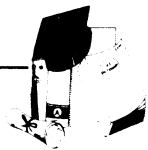
You can list all 50 states by typing a 1.
Type in a 2 to start the program.
Type in a 3 to end it.

When putting in the 50 states, you have the option of listing the ones you have already entered. This is done by entering LIST. You can end the program and get back to the menu by typing END. If you enter all 50 successfully, you win the game!



STATES

```
100 PRINT CHR$(147) CHR$(144)
110 POKE 36879,25
120 DIM STATE$(50)
125 DIM ASKED$(50)
130 FOR T=1 TO 50
140 READ STATE$(T)
150 NEXT
200 REM *****
210 REM * PRINT SCREEN *
220 REM *****
230 PRINT CHR$(147) TAB(2);
240 PRINT "OUR UNITED STATES"
250 PRINT
260 PRINT "<1> LIST STATES"
270 PRINT "<2> START"
280 PRINT "<3> QUIT"
285 PRINT
290 PRINT
295 PRINT
300 PRINT "END=END"
310 PRINT "LIST=LIST GUESSES"
315 PRINT CHR$(19)
320 PRINT
325 PRINT
330 PRINT
335 PRINT
340 PRINT
345 PRINT
350 INPUT "WHICH :";R$
355 R=VAL(R$)
360 IF R<1 OR R>3 THEN 320
370 IF R=3 THEN PRINT CHR$(147) : END
380 IF R<>1 THEN 640
400 REM *****
410 REM * PRINT ALL STATES *
420 REM *****
```

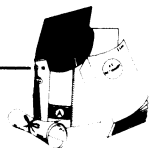


```
430 PRINT CHR$(147)
440 FOR T=1 TO 50
450 PRINT STATE$(T)
460 GET R$ : IF R$="" THEN 460
470 NEXT T
475 PRINT
480 PRINT "THAT'S ALL"
490 GET R$ : IF R$="" THEN 490
500 GOTO 230
600 REM *****
610 REM * ENTER & TELL *
620 REM * IF CORRECT *
630 REM *****
640 PRINT CHR$(147)
650 GUESS=0
660 FOR T=1 TO 50
665 PRINT
670 PRINT "STATE#";T;
675 GUESS$=""
680 INPUT " ";GUESS$
690 IF GUESS$="LIST" THEN 1030
700 IF GUESS$="END" THEN RUN
710 IF GUESS$="" THEN 670
720 FOR X=1 TO 50
730 IF GUESS$=STATE$(X) THEN 730
740 NEXT
750 PRINT
760 PRINT "THERE'S NO SUCH STATE!"
770 GOTO 670
780 FOR X=1 TO GUESS
790 IF GUESS$=ASKED$(X) THEN 930
800 NEXT
810 GUESS=GUESS+1
820 ASKED$(GUESS)=GUESS$
830 NEXT
840 FOR T=0 TO 255
850 POKE 53280,T : NEXT
860 PRINT : PRINT
```

```

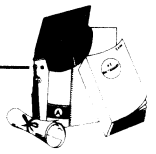
870 PRINT "YOU'VE GUESSED THEM!"
880 GET R$: IF R#="" THEN 880
890 RUN
900 REM *****
910 REM * ALREADY ENTERED *
920 REM *****
930 X=100
940 PRINT
950 PRINT "ALREADY BEEN USED!"
960 GOTO 670
1000 REM *****
1010 REM * STATES USED *
1020 REM *****
1030 IF GUESS=0 THEN 1120
1040 PRINT
1050 FOR X=1 TO GUESS
1060 PRINT "STATE#";X;": ";ASKED$(X)
1070 GET G$: IF G#="" THEN 1070
1080 NEXT
1090 PRINT
1090 PRINT "THERE'S NO MORE!"
1100 GET R$: IF R#="" THEN 1100
1110 GOTO 670
1120 PRINT
1130 PRINT "NONE ARE ENTERED YET!"
1140 GOTO 670
1200 REM *****
1210 REM * STATES DATA *
1220 REM *****
1230 DATA ALABAMA,ALASKA,ARIZONA
1240 DATA ARKANSAS,CALIFORNIA,COLORADO
1250 DATA CONNECTICUT,DELAWARE
1260 DATA FLORIDA,GEORGIA,HAWAII,IDAHO
1270 DATA ILLINOIS,INDIANA,IOWA,KANSAS
1280 DATA KENTUCKY
1290 DATA LOUISIANA,MAINE,MARYLAND
1300 DATA MASSACHUSETTS,MICHIGAN
1310 DATA MINNESOTA,MISSISSIPPI

```



1320 DATA MISSOURI, MONTANA, NEBRASKA
1330 DATA NEVADA, NEW HAMPSHIRE
1340 DATA NEW JERSEY, NEW MEXICO
1350 DATA NEW YORK, NORTH CAROLINA
1360 DATA NORTH DAKOTA, OHIO, OKLAHOMA
1370 DATA OREGON, PENNSYLVANIA
1380 DATA RHODE ISLAND, SOUTH CAROLINA
1390 DATA SOUTH DAKOTA, TENNESSEE, TEXAS
1400 DATA UTAH, VERMONT
1410 DATA VIRGINIA, WASHINGTON
1420 DATA WEST VIRGINIA, WISCONSIN
1430 DATA WYOMING

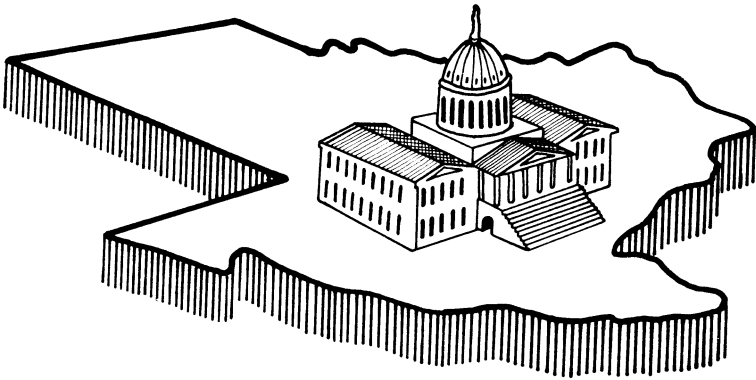




Learning the Capitals

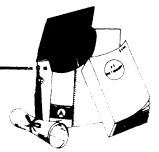
Sure, everyone knows a few capitals. If you think that the capital of Texas is Dallas, the capital of Florida is Miami, and the capital of California is San Diego, are you in for a surprise!

In addition to knowing all 50 states, learning their respective capitals is equally important. This program is basically the same as the States program, except that it cannot list the states whose capitals you have already entered. It will tell you if your entry is incorrect by printing the state's correct capital when you enter a wrong answer. You can exit to the menu by typing END in response to a capital entry.



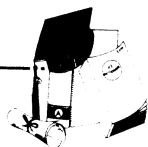
CAPITALS

```
100 PRINT CHR$(147) CHR$(144)
110 POKE 36879,25
115 DIM ST$(50)
120 DIM CAP$(50)
125 DIM GUESSED(50)
130 FOR T=1 TO 50
140 READ ST$(T)
145 READ CAP$(T)
150 NEXT
200 REM *****
210 REM * PRINT SCREEN *
220 REM *****
230 PRINT CHR$(147) TAB(2);
235 PRINT "STATES & CAPITALS"
240 PRINT
245 PRINT
250 PRINT
255 PRINT
260 PRINT "<1> LIST CAPITALS"
265 PRINT "<2> START"
270 PRINT "<3> QUIT"
275 PRINT
280 PRINT "END=END"
285 PRINT CHR$(19)
290 PRINT
295 PRINT
300 INPUT "WHICH :";R$
310 IF VAL(R$)<1 OR VAL(R$)>3 THEN 290
320 IF R$="3" THEN PRINT CHR$(147) : END
330 IF R$("<1") THEN 630
400 REM *****
410 REM * PRINT ALL *
420 REM * STATES & CAPS *
430 REM *****
440 PRINT CHR$(147)
450 FOR T=1 TO 50
470 PRINT ST$(T)
```

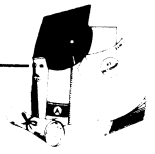
```
475 PRINT CHR$(18) CAP$(T)
480 PRINT
485 GOSUB 1530
490 NEXT T
500 PRINT : PRINT "THAT'S ALL"
510 GOSUB 1530
520 GOTO 230
600 REM *****
610 REM * ENTER CAPITAL *
620 REM *****
630 PRINT CHR$(147)
640 NUM=INT(RND(1)*50+1)
650 IF GUESSED(NUM)=1 THEN 640
660 PRINT
670 PRINT "CAP. OF " ST$(NUM)
680 GUESS$=""
685 PRINT "NAME:";
690 INPUT GUESS$
695 IF GUESS$="END" THEN 230
700 IF GUESS$=CAP$(NUM) THEN 740
710 PRINT
720 PRINT "WRONG, IT'S:"
725 PRINT CHR$(18) CAP$(NUM)
730 GOTO 640
740 PRINT
745 PRINT "YOU'RE CORRECT!!"
750 GUESSED(NUM)=1
760 FOR T=1 TO 50
770 IF GUESSED(T)=0 THEN T=60
780 NEXT
790 IF T>=60 THEN 640
800 PRINT
810 PRINT "YOU'VE DONE THEM ALL!!"
820 GOSUB 1530
830 RUN
900 REM *****
910 REM * STATES & CAPITALS *
920 REM *****
```

930 DATA ALABAMA, MONTGOMERY, ALASKA
940 DATA JUNEAU, ARIZONA, PHOENIX
950 DATA ARKANSAS, LITTLE ROCK
960 DATA CALIFORNIA, SACRAMENTO, COLORADO
970 DATA DENVER, CONNECTICUT, HARTFORD
980 DATA DELAWARE, DOVER
990 DATA FLORIDA, TALLAHASSEE, GEORGIA
1000 DATA ATLANTA, HAWAII, HONOLULU, IDAHO
1010 DATA BOISE
1020 DATA ILLINOIS, SPRINGFIELD, INDIANA
1030 DATA INDIANAPOLIS, IOWA, DES MOINES
1040 DATA KANSAS, TOPEKA
1050 DATA KENTUCKY, FRANKFORT, LOUISIANA
1060 DATA BATON ROUGE, MAINE, AUGUSTA
1070 DATA MARYLAND, ANNAPOLIS
1080 DATA MASSACHUSETTS, BOSTON, MICHIGAN
1090 DATA LANSING
1100 DATA MINNESOTA, ST. PAUL, MISSISSIPPI
1110 DATA JACKSON, MISSOURI
1120 DATA JEFFERSON CITY
1130 DATA MONTANA, HELENA, NEBRASKA
1140 DATA LINCOLN, NEVADA, CARSON CITY
1150 DATA NEW JERSEY, TRENTON
1160 DATA NEW HAMPSHIRE, CONCORD
1170 DATA NEW MEXICO, SANTA FE, NEW YORK
1180 DATA ALBANY
1190 DATA NORTH DAKOTA, BISMARCK
1200 DATA NORTH CAROLINA, RALEIGH, OHIO
1300 DATA COLUMBUS
1310 DATA OKLAHOMA, OKLAHOMA CITY, OREGON
1320 DATA SALEM, PENNSYLVANIA, HARRISBURG
1330 DATA RHODE ISLAND, PROVIDENCE
1340 DATA SOUTH CAROLINA, COLUMBIA
1350 DATA SOUTH DAKOTA, PIERRE
1360 DATA TEXAS, AUSTIN, TENNESSEE
1370 DATA NASHVILLE
1380 DATA UTAH, SALT LAKE CITY, VERMONT
1390 DATA MONTPELIER, VIRGINIA, RICHMOND



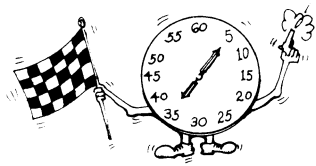
```
1400 DATA WASHINGTON,OLYMPIA
1410 DATA WEST VIRGINIA,CHARLESTON
1420 DATA WISCONSIN,MILWAUKEE
1430 DATA WYOMING,CHEYENNE
1500 REM *****
1510 REM * GET PRESS *
1520 REM *****
1530 GET R$: IF R$="" THEN 1530
1540 RETURN
```





Math Quizzer

The Math Quizzer program was designed to help students who need repetitive review to learn mathematical concepts. With this program, the student can be quizzed in addition, subtraction, multiplication or division, by entering a number from 1–4. The lower and upper limits from which the computer will pick random numbers may also be set. Entering values of 10 and 30 will instruct the computer to pick numbers from 10 to 30 when constructing sample problems. The division option picks out problems with no remainder, as simple remainders would be quite difficult to do on a computer. If the problem is answered incorrectly, the correct answer will be shown, and a new problem will be picked out.



$$5 + 1 =$$

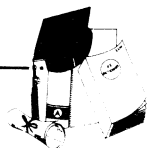


$$2 - 2 =$$



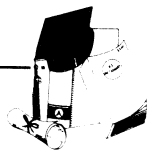
MATH QUIZZER

```
100 PRINT CHR$(147) CHR$(144)
110 POKE 36879,25
120 LL=1
130 UL=100
140 NUM=100
200 REM *****
210 REM * PRINT SCREEN *
220 REM *****
230 PRINT CHR$(147) TAB(5);
235 PRINT "MATH SKILLS"
240 PRINT
245 PRINT
250 PRINT "WHICH:"
255 PRINT
260 PRINT "<1> ADDITION"
270 PRINT "<2> SUBTRACTION"
280 PRINT "<3> MULTIPLICATION"
290 PRINT "<4> DIVISION"
300 PRINT "<5> SET LIMITS"
310 GET R# : IF R#="" THEN 310
320 R=VAL(R#)
330 IF R<1 OR R>5 THEN 310
340 IF R=5 THEN 430
350 IF R=2 THEN S#=" - "
360 IF R=1 THEN S#="+ "
370 IF R=3 THEN S#="X "
380 IF R=4 THEN S#="/ "
390 GOTO 530
400 REM *****
410 REM * SET LIMITS *
420 REM *****
430 PRINT
435 INPUT "LOWER LIMIT: ";LL
440 PRINT
445 INPUT "UPPER LIMIT: ";UL
450 NUM=UL-LL+1
```



```
460 GOTO 230
500 REM *****
510 REM * TEST ON +,-,/,X *
520 REM *****
530 PRINT CHR$(147)
540 N1=INT(RND(1)*NUM+1)+LL 1
550 IF N1=NM THEN 540
560 N2=INT(RND(1)*NUM+1)+LL-1
570 IF N2=NN THEN 560
580 IF S$(">")/ " THEN 620
590 IF N2=0 THEN 560
600 FIG=N1/N2
610 IF FIG<>INT(FIG) THEN 540
620 IF S$="- " THEN FIG=N1-N2
630 IF S$="X " THEN FIG=N1*N2
640 IF S$="+ " THEN FIG=N1+N2
650 PRINT
655 PRINT "WHAT IS " N1;S$;N2 " ?"
660 ANS$=""
665 PRINT
670 INPUT "THE ANSWER IS:";ANS$
680 IF ANS$="" THEN 230
690 NM=N1
695 NN=N2
700 ANS=VAL(ANS$)
710 PRINT
720 IF ANS<>FIG THEN 930
800 REM *****
810 REM * YOU'RE RIGHT *
820 REM *****
830 PRINT "RIGHT YOU ARE!"
840 GET R$ : IF R$="" THEN 840
850 GOTO 540
900 REM *****
910 REM * YOU'RE WRONG *
920 REM *****
930 PRINT "WRONG! IT'S " FIG
940 GOTO 540
```





Divisors

Divisors is a program that shows you all of the possible factors of a number. This program is one of the easier ones in this section, so even beginners should have no problem understanding it. Let's say that you have entered the number 8. The computer would print the following:

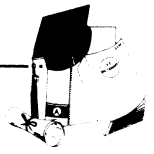
```
1
2
4
8
```

The program displays the divisors from one to half of the number you have entered. Remember that the divisor of a number cannot be more than half of that number. To be a divisor, the number must divide into your original number without a remainder.

```
DIVISORS

100 PRINT CHR$(147) CHR$(144)
110 POKE 36879,25
120 NUM=0
130 INPUT "ENTER NUMBER: ";NUM
140 IF NUM=0 THEN 210
150 FOR T=1 TO NUM/2
160 IF INT(NUM/T)=NUM/T THEN PRINT T
170 NEXT
180 PRINT NUM
190 PRINT
200 GOTO 120
210 PRINT
220 END
```





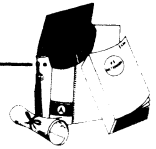
Trig Helper

Having taken quite a few trigonometric courses, I know the importance of having trigonometric functions at your disposal (especially since the precision in books only goes to about four digits). With the Trigonometric Helper program, you have six basic functions to choose from. Sine, cosine, tangent, cosecant, secant, and cotangent. Simply type the number of the function you want (1–6), and then the computer will prompt you for an angle. Enter the angle (0–360) and the computer will print out the value for whatever function you choose. See, it's as easy as 3.14159!

```
TRIG HELPER
```

```
100 PRINT CHR$(147) CHR$(144)
110 POKE 36879,25
200 REM *****
210 REM * PRINT SCREEN *
220 REM *****
230 PRINT CHR$(147) "MATHEMATICAL FUNCTIONS"
:
240 PRINT "-----"
250 PRINT
260 PRINT "<1> SINE"
270 PRINT "<2> COSINE"
280 PRINT "<3> TANGENT"
290 PRINT "<4> COSECANT"
300 PRINT "<5> SECANT"
310 PRINT "<6> COTANGENT"
320 PRINT : PRINT "WHICH:";
330 INPUT W#
340 W=VAL(W#)
350 IF W<1 OR W>6 THEN 230
400 REM *****
410 REM * GET ANGLE *
420 REM *****
```

```
430 PRINT CHR$(147) "ANGLE(0-360):";
440 INPUT ANGLE#
450 ANGLE=VAL(ANGLE#)
460 IF ANGLE<0 OR ANGLE>360 THEN 230
470 ANGLE=ANGLE*3.14159/180
480 ON W COSUB 630,650,670,690,710,730
500 REM *****
510 REM * PRINT ANSWER *
520 REM *****
530 PRINT
535 PRINT
540 PRINT "ANSWER=" NUM
550 PRINT
555 PRINT
560 PRINT "ANY KEY TO CONTINUE"
570 POKE 197,64
580 IF PEEK(197)=64 THEN 500
590 RUN
600 REM *****
610 REM * FUNCTIONS *
620 REM *****
630 NUM=SIN(ANGLE)
640 RETURN
650 NUM=COS(ANGLE)
660 RETURN
670 NUM=TAN(ANGLE)
680 RETURN
690 NUM=1/SIN(ANGLE)
700 RETURN
710 NUM=1/COS(ANGLE)
720 RETURN
730 NUM=1/TAN(ANGLE)
740 RETURN
```



Primes

Math can be made easy! Do you think I'm loony? Actually, math can be if you have a computer. Computers make mathematical calculations millions of times easier (in some cases that's literally true!). Math Quizzer, Trigonometric Helper, and Graph Plotter are all cases in point. If you have ever taken a math course, chances are you've had to work with prime numbers somewhere along the line. To refresh your memory, a prime number is one that is only divisible by one and itself. A few examples are:

2
3
5
7
11
13
17
19
23

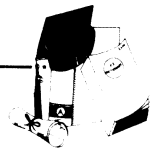
You get the idea. Finding prime numbers on a computer is easy and quick, as the program below demonstrates:

```
PRIMES
100 PRINT CHR$(147) CHR$(144)
110 POKE 36879,25
200 REM *****
210 REM * GET NUMBERS *
220 REM *****
230 PRINT CHR$(147) "ENTER TWO NUMBERS THAT"
;
240 PRINT "YOU WOULD LIKE TO KNOW";
250 PRINT "THE PRIMES BETWEEN."
260 PRINT : PRINT
270 PRINT "(EX. 4,40)";
```

```

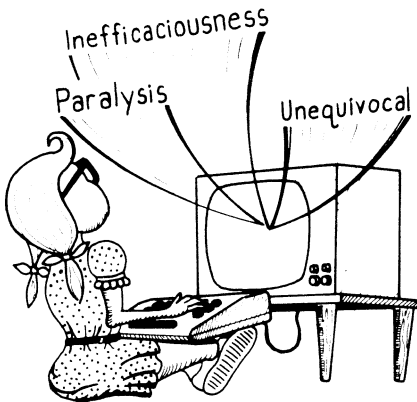
280 INPUT Q,X
290 IF Q=X THEN 520
300 IF Q>X THEN 550
400 REM *****
410 REM * FIND PRIMES *
420 REM *****
430 PRINT CHR$(147)
440 DIM A(X/2)
450 B=0
460 PRINT "THE PRIME NUMBERS BETWEEN"
470 PRINT Q "AND" X "ARE:"
480 PRINT : PRINT
490 FOR Y=2 TO X
500 IF Y=2 THEN 550
510 FOR M=0 TO B-1
520 IF A(M)>Y/2 THEN 550
530 IF Y/A(M)=INT(Y/A(M)) THEN 590
540 NEXT M
550 IF Y<Q THEN 570
560 PRINT Y,
570 A(B)=Y
580 B=B+1
590 NEXT Y
600 END
700 REM *****
710 REM * SAME NUMBER *
720 REM *****
730 PRINT
740 PRINT "THERE'S NOTHING BETWEEN" Q "AND"
X "Q"
750 END
800 REM *****
810 REM * BAD RANGE *
820 REM *****
830 PRINT : PRINT
840 PRINT "INPUT THE SMALLER NUMBER ";
850 PRINT "BEFORE THE"
860 PRINT "LARGER ONE!"

```



Spelling Practice

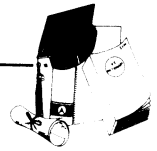
This is a fun and useful way to prepare for a spelling quiz. When you RUN this program, you will be asked to enter your list of spelling words. After you have entered them all, the words will be given back to you, one at a time, to spell. They are flashed on the screen for just a moment; long enough to recognize the word, but not long enough to completely look at its spelling. Believe me, it works! You will then be asked to spell the word correctly. If you get the word right, the computer will tell you so. If not, then you will be given more chances until you do get it right.



Spelling Practice can be run by the student himself, or with his or her parent's help. If parents do help, they should enter the spelling words when the program first asks for the list, and the student can spell them when they flash on the screen. The words are presented in reverse order from the way they were entered; that way, they are not memorized in a listed order. This program has been tested, and students using it improved markedly on their spelling quizzes. See if you can improve your spelling as well!

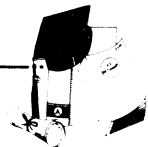
SPELLING PRACTICE

```
100 PRINT CHR$(147) CHR$(144)
110 POKE 36879,25
200 REM *****
210 REM * ENTER SPELLING WORDS *
220 REM *****
230 CLR
240 RESTORE
250 PRINT
260 PRINT
270 INPUT "HOW MANY WORDS: ";N
280 DIM SP$(N)
290 PRINT
300 PRINT
310 FOR X=1 TO N
320 PRINT "WORD#" X " : ";
330 INPUT SP$(X)
340 NEXT
400 REM *****
410 REM * SPELLING QUIZ *
420 REM *****
430 FOR X=1 TO N
440 PRINT CHR$(147)
450 FOR T=1 TO 10
460 PRINT
470 NEXT
480 L=LEN(SP$(X))
490 PL=11-L/2
500 PRINT TAB(PL) SP$(X)
510 FOR P=1 TO 200
520 NEXT
530 PRINT CHR$(147)
540 PRINT "ENTER CORRECT SPELLING"
550 PRINT
560 PRINT
570 INPUT " : ";CS$
580 IF CS$=SP$(X) THEN G30
```

```
590 PRINT
600 PRINT
610 PRINT "INCORRECT! TRY AGAIN."
620 FOR F=1 TO 400
630 NEXT
640 NEXT
650 PRINT
660 PRINT
670 POKE 198,0
680 PRINT "ANOTHER RUN(Y/N)"
690 GET A$: IF A$="" THEN 690
700 IF A$="Y" THEN 430
710 PRINT CHR$(147)
720 PRINT "GOOD LUCK ON THE TEST!"
730 END
800 REM *****
810 REM * CORRECT ANSWER *
820 REM *****
830 PRINT CHR$(147)
840 FOR T=1 TO 10
850 PRINT
860 NEXT
870 PRINT TAB(3) "THAT'S CORRECT!"
880 FOR T=1 TO 400
890 NEXT
900 GOTO 640
```

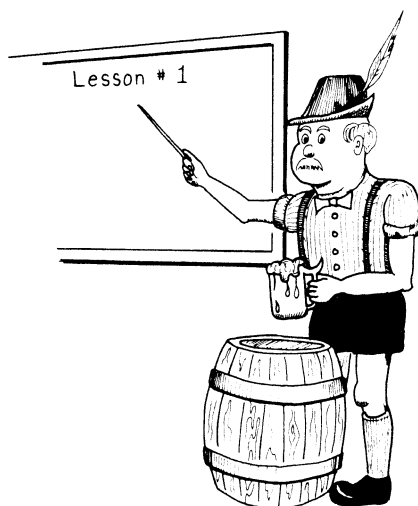




Foreign Language Study

I remember taking German at Grossmont High School, with a teacher named Mr. Jose Weill. He was a great teacher, but there are some things that a teacher just does not have time for. One of these things is repetitive review. A computer is just perfect for this sort of thing. German grammar is easy. The difficulty is in memorizing definite articles, verbs and vocabulary. Foreign Language Study eliminates this problem by quizzing you on whatever vocabulary you have entered into its file. It will do this for whatever language you happen to be learning. Foreign Language Study is moderately easy to understand. There are three commands in the main menu:

- <1> EDIT VOCABULARY
- <2> QUIZ TIME
- <3> QUIT



Now let's start entering data. Do this by pressing 1. Now the computer will prompt you with:

```
<1> LIST VOCABULARY
<2> ADD VOCAB.
<3> RETURN TO MENU
```

Entering a 3 will return you to the main menu as shown above. To list all of the entries you have made so far, enter a 1. It will ask you if you would like to direct output to the <P>RINTER or to the <S>CREEN. Enter a P or an S. When all of your entries have been listed, the message THAT'S ALL FOLKS! will be printed, and it will wait for a key to be pressed. Pressing any key will put you into the second menu.

Adding entries is easy. Simply enter a 2. Now it will ask you for the English word. Enter for example, THE DOG. Next it will ask you for the appropriate foreign word. Enter DER HUND, which is German for "the dog." You may have noticed that an X is always printed out. This is done so that if you are finished entering items, you can just press RETURN and you will be put back into the second menu. Feel free to change the string "foreign" to whatever language you are currently studying. When you are ready to be quizzed on your vocabulary, enter a 2 in the main menu. You will now see this displayed:

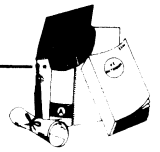
```
<1> ENGLISH TO FOREIGN
<2> FOREIGN TO ENGLISH
<3> RETURN TO MENU
```

If you pick number 1, the computer might say:

```
GIVE ME THE FOREIGN WORD FOR: THE DOG
```

If you pick number 2, the computer might say:

```
GIVE ME THE ENGLISH WORD FOR: DER HUND
```



If you picked number 1, then you would enter DER HUND. The computer will respond with an appropriate answer, telling you whether or not your response was correct. The computer will ask only vocabulary that you have not yet correctly translated. I have found this program to be tremendously helpful as far as memorizing nouns, verbs and the like, and I hope you do too.

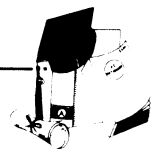
FOREIGN LANGUAGE STUDY

```
100 PRINT CHR$(147) CHR$(144)
110 POKE 36879,25
120 OPEN 1,0 : C#=CHR$(13)
130 DIM WRD$(1,20),ANS(20)
140 WRDS=0 : BK#=CHR$(157)
150 RVS#=CHR$(18)
160 PRINT CHR$(147) TAB(7) "TONGUES"
170 PRINT : PRINT : PRINT
180 PRINT "MENU:" : PRINT
190 PRINT "<1> EDIT VOCABULARY"
200 PRINT "<2> QUIZ TIME"
210 PRINT "<3> QUIT"
220 PRINT CHR$(19) : PRINT : PRINT
230 PRINT : PRINT TAB(5);
240 INPUT#1,A#
250 A=VAL(A#)
260 IF A<1 OR A>3 THEN 160
270 ON A GOTO 310,1030
280 PRINT CHR$(147) "BYE BYE"
290 END
300 REM *****
310 REM * EDIT *
320 REM *****
330 PRINT CHR$(147) TAB(3) "EDIT VOCABULARY"
340 PRINT : PRINT
350 PRINT "WHICH:" : PRINT
360 PRINT "<1> LIST VOCABULARY"
```

```

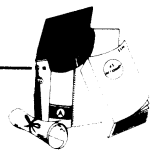
370 PRINT "<2> ADD VOCAB."
380 PRINT "<3> RETURN TO MENU"
390 GET A$: IF A$="" THEN 390
400 A=VAL(A$)
410 IF A<1 OR A>3 THEN 390
420 ON A GOTO 530,830,160
500 REM *****
510 REM * LIST *
520 REM *****
530 IF WRDS=0 THEN 330
540 PRINT CHR$(147) "<P>PRINTER/<S>CREEN"
550 GET H$: IF H$="" THEN 550
560 IF H$<>"P" AND H$<>"S" THEN 330
570 IF H$="S" THEN 610
580 OPEN 2,4
590 PRINT#2,C$C$
600 PRINT#2,TAB(10) "VOCABULARY FOR " FILE$
C$
610 PRINT
620 FOR T=0 TO WRDS-1
630 T$=RIGHT$(" "+STR$(T+1),3)
640 IF H$="S" THEN 660
650 PRINT#2,T$ ":" WRD$(0,T),,WRD$(1,T)
660 PRINT T$ ":" WRD$(0,T)
670 PRINT RVS$ TAB(4) WRD$(1,T)
680 IF PEEK(197)<>60 THEN 720
690 X=PEEK(197)
700 IF X=64 THEN 690
710 IF X<>60 THEN T=500
720 NEXT
730 PRINT : PRINT "THAT'S ALL FOLKS!"
740 GET R$: IF R$="" THEN 740
750 CLOSE 2 : GOTO 330
800 REM *****
810 REM * ADD *
820 REM *****
830 IF WRDS=20 THEN 160
840 PRINT CHR$(147) TAB(4) "ADD VOCABULARY"

```



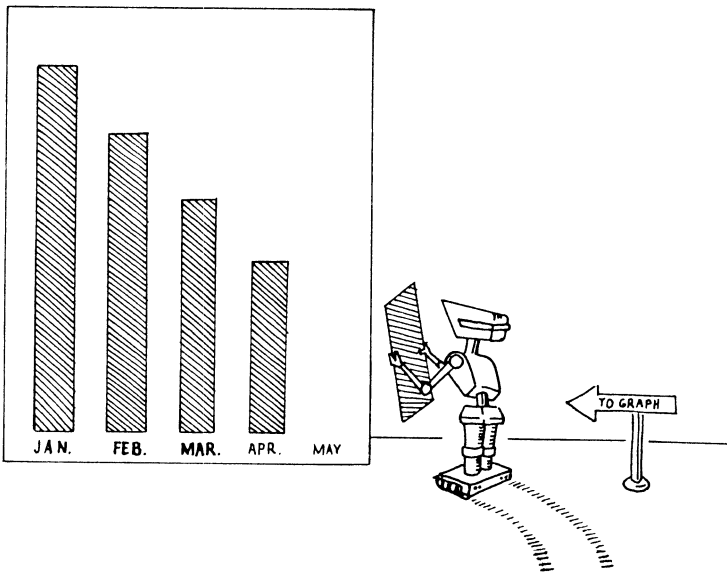
```
850 PRINT : PRINT
860 PRINT : PRINT
870 W$=RIGHT$(" "+STR$(WRDS+1),3)
880 PRINT "ENGLISH#" W$ ":X" BK$;
890 INPUT#1,EW$
900 IF EW$="X" THEN 330
910 PRINT : PRINT "FOREIGN#" W$ ":X" BK$;
920 INPUT#1,FW$
930 IF FW$="X" THEN 330
940 PRINT
950 INPUT "CORRECT (Y/N)";C$
960 IF C$<>"Y" THEN 860
970 WRD$(0,WRDS)=EW$ : WRD$(1,WRDS)=FW$
980 WRDS=WRDS+1 : GOTO 860
1000 REM *****
1010 REM * QUIZ *
1020 REM *****
1030 IF WRDS=0 THEN 160
1040 PRINT CHR$(147) TAB(6) "QUIZ TIME"
1050 PRINT : PRINT : PRINT
1060 PRINT "<1> ENGLISH TO FOREIGN";
1070 PRINT "<2> FOREIGN TO ENGLISH";
1080 PRINT "<3> RETURN TO MENU"
1090 PRINT
1100 PRINT "WHICH:";
1110 INPUT#1,A$
1120 A=VAL(A$)
1130 IF A<1 OR A>3 THEN 1030
1140 IF A=3 THEN 160
1150 PRINT CHR$(147)
1160 FOR T=0 TO WRDS-1
1170 ANS(T)=0 : NEXT
1180 A$="ENGLISH" : ANS=0 : WO=1
1190 IF A=1 THEN A$="FOREIGN" : ANS=1 : WO=0
1200 NUM=INT(RND(1)*WRDS)
1210 IF ANS(NUM)=1 THEN 1200
1220 PRINT "GIVE ME THE " A$
1230 PRINT "WORD FOR " WRD$(WO,NUM)
```

```
1240 PRINT "ANSWER:";
1250 INPUT#1,ANS#
1260 IF ANS#="X" THEN 160
1270 PRINT
1280 IF ANS#(<)WRD$(ANS,NUM) THEN 1350
1290 PRINT "CORRECT!":PRINT
1300 ANS(NUM)=1
1310 FOR S=0 TO WRDS-1
1320 IF ANS(S)=1 THEN NEXT:GOTO 1370
1330 S=500: NEXT
1340 GOTO 1200
1350 PRINT "WRONG, IT'S " WRD$(ANS,NUM)
1360 PRINT:GOTO 1200
1370 PRINT:PRINT
1380 PRINT "YOU'VE ANSWERED ALL OF THEM!"
1390 GET R# : IF R#="" THEN 1390
1400 GOTO 160
```

Graph Plotter

In order to use the graphics that the VIC-20 is capable of producing, it is necessary to use POKEs (in BASIC), machine language programs, or assemblers. Graphics are easy in machine language, but not particularly easy if you're working in BASIC.



To me, one of the most exciting things I can watch a computer do is produce graphs. Bar graphs can be done with graphic characters, but for the precision required for mathematical functions, hi-res graphics is the only solution.

Once the graph has been drawn, the computer waits for you to press a key. The program will then put you back into the text mode. To change the height of the graph, you must change the variable AMPLITUDE. Changing it changes the height of the graph. You can use negative numbers if you wish. An effect that really looks nice can be achieved by using a FOR/NEXT loop that varies the amplitude from -1 to 1, step .1. It gives a very good 3-D effect. To change the period of the graph, divide the variable X by some number. My graphing program uses a domain from 0 to 2Pi. If you would like to do a COS plot, then merely change the SIN to COS. It's as easy as that! For other ideas for equations, consult page 275 in the *VIC-20 Programmer's Reference Guide*.

GRAPH PLOTTER

```

100 PRINT CHR$(147) CHR$(144)
110 POKE 36877,10
200 REM *****
210 REM * CHAR SET POINTER *
220 REM *****
230 POKE 36867,255
300 REM *****
310 REM * CLR HI RES SCREEN *
320 REM *****
330 FOR I=7160 TO 7679
340 POKE I,0
350 NEXT
400 REM *****
410 REM * BLANK OUT SCREEN *
420 REM *****
430 FOR I=7680 TO 8195
440 POKE I,160
450 NEXT
500 REM *****
510 REM * PUT PRG CHRS ON SCREEN *
520 REM *****

```



```
530 FOR L=0 TO 7
540 FOR M=0 TO 7
550 POKE 7680+M*22+L,L*8+M
560 NEXT
570 NEXT
600 REM *****
610 REM * EQUATION OF GRAPH *
620 REM *****
625 AMPLITUDE=1
630 FOR DOMAIN=0 TO 2*3.14 STEP .1
640 Y=30-AMPLITUDE*SIN(DOMAIN)*15
650 X=DOMAIN*10
660 IF X<0 OR X>63 THEN 600
670 IF Y<0 OR Y>63 THEN 600
700 REM *****
710 REM * EQUATION FOR DOT *
720 REM *****
730 CHAR=INT(X/8)*8+INT(Y/8)
740 ROW=(Y/8-INT(Y/8))*8
750 BYTE=7168+CHAR*8+ROW
760 BIT=7-(X AND 7)
770 POKE BYTE,PEEK(BYTE) OR (2^BIT)
780 NEXT
800 REM *****
810 REM * GRAPH IS FINISHED *
820 REM *****
830 GET A$
840 IF A$="" THEN 830
850 POKE 36879,27
860 POKE 36869,240
870 PRINT CHR$(147) CHR$(31)
```

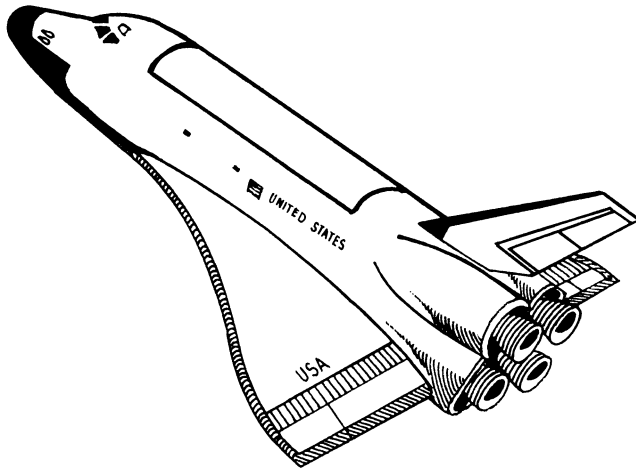


Section 3

Entertainment Programs

- Death Hunt
- Guided Missile Commander
- Music Machine
- Tank Command
- Memory Master
- Pick a Number
- Doodle



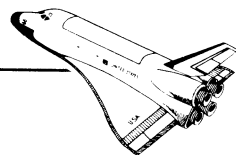


ENTERTAINMENT PROGRAMS

For the Entertainment section of this book, I wanted to give an example of the two ways of programming games on the VIC-20. Included are one game using text characters, and one using programmable characters. The game Death Hunt uses only text and graphic characters, and Guided Missile Commander uses programmable characters, that is characters which are redefined from the Commodore character set. Another little ditty is Music Machine, a program which lets you compose melodies, annoying or pleasant, depending upon your musical ability.

Tank Command is a great little arcade-type program that simulates a tank going through mine infested territory. For those of you wanting to expand your horizons, consciousness and all-around brain ability, there's Memory Master, a program which tests your color code recognition skills. Two other programs in this section are Pick a Number, where the computer generates a random number that the user has to guess, and Doodle, a program that allows the user to create and edit high resolution pictures.

Look to see how the programs produce animation and how graphic characters are generated. There are hints in all of these programs on how arcade games are created. Who knows? Someday you may make your own!



Death Hunt

The VIC-20 supports quite an impressive character set, including upper and lower case and even graphic characters! These graphic symbols add flair to any program whether they're used for a border or for a game character. Death Hunt is an example of a text character game. In it, I used graphic characters instead of programmable characters to represent the game pieces. The object of the game is to maneuver your circle (◻) around the screen and capture the black dot (●) (the enemy), while being extremely careful not to hit any of the mines (Xs) or tombstones (clovers) left by the deaths of your foes. After the game is loaded, you will be asked to enter your Skill Level, from 1-9. Enter the level you want. The screen will clear and the playing field will be drawn. You always begin in the top left corner under the SCORE. Maneuver your circle using the following keys: I (up), K (down), J (left) and L (right); any other key, including the space bar, will stop the circle. Try to collect as many points as you can before you CRASH.

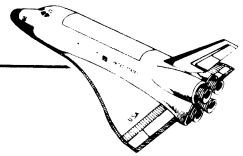
DEATH HUNT

```
100 REM *****
110 REM * SET SCREEN *
120 REM *****
130 PRINT CHR$(147) CHR$(144)
140 POKE 36879,25
150 INPUT "SKILL LEVEL(1-9):";SL
160 IF SL<1 OR SL>9 THEN 130
170 NM=SL*30
180 B3=0185
190 SF=32
200 B1=7702
210 B2=38422
220 CH=87
230 EN=81
240 CR=88
```

```

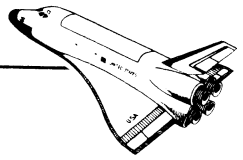
250 MI=86
260 AM=30720
270 O=1
280 F=22
285 PRINT CHR$(147) "SCORE:" SC
290 FOR T=1 TO NM
300 PL=INT(RND(1)*484)
310 POKE B1+F+PL,MI
320 POKE B2+F+PL,O
330 NEXT
340 PF=D1
350 EP=7702+INT(RND(1)*483)
360 P$=""
400 REM *****
410 REM * START GAME *
420 REM *****
430 POKE PP,CH
440 POKE PP+AM,0
450 POKE EP,EN
460 POKE EP+AM,0
470 GET R$
480 IF R$="" AND P$="" THEN 620
490 IF R$="" THEN R$=P$
500 P$=R$
510 M=0
520 IF R$="J" THEN M=-O
530 IF R$="L" THEN M=O
540 IF R$="K" THEN M=F
550 IF R$="I" THEN M=-F
560 IF M=0 THEN P$="" : GOTO 620
570 IF PP+M<B1 OR PP+M>B3 THEN 620
580 L=PEEK(PP+M)
590 IF L<>32 AND L<>0 THEN 830
600 POKE PP,SP
610 PP=PP+M
620 X=INT(RND(1)*4)
630 IF X=0 THEN M=O
640 IF X=1 THEN M=-O

```



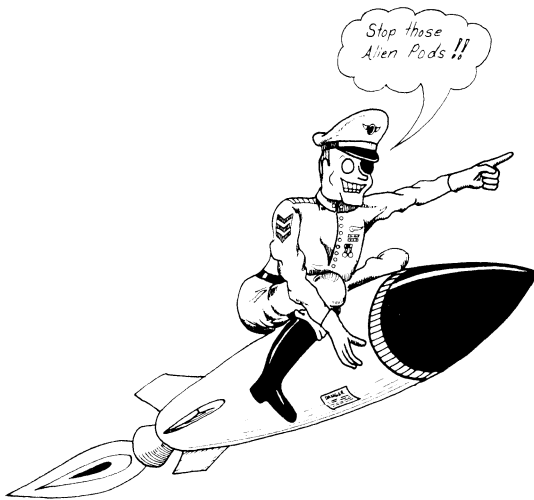
```
650 IF X=2 THEN M=F
660 IF X=3 THEN M=-F
670 E=PEEK(EP+M)
680 IF E<>SP AND E<>CH THEN 620
690 POKE EP,SF
700 EP=EP+M
710 IF EP>B3 THEN EP=B3
720 IF EP<B1 THEN EP=B1
730 POKE EP,EN
740 POKE EP+AM,0
750 IF PP=EP THEN 840
760 GOTO 430
800 REM *****
810 REM * COLLISION *
820 REM *****
830 IF L<>EN THEN 1030
840 FOR T=1 TO 3
850 POKE PP+M,CR+128
860 FOR S=1 TO 100
870 NEXT
880 POKE PP+M,CR
890 FOR S=1 TO 100
900 NEXT
910 NEXT
920 SC=SC+1
930 PRINT CHR$(19) TAB(7) SC
940 P$=""
950 EP=7702+INT(RND(1)*483)
960 IF PEEK(EP)<>32 THEN 950
970 GOTO 430
1000 REM *****
1010 REM * MUNCH IT *
1020 REM *****
1030 FOR T=0 TO 255
1040 POKE 36879,T
1050 NEXT
1060 POKE 36879,25
1070 PRINT CHR$(147) "YOU MUNCHED IT!!"
```

```
1080 PRINT
1090 PRINT
1100 PRINT
1110 PRINT "PLAY AGAIN (Y/N)"
1120 POKE 198,0
1130 GET R$: IF R$="" THEN 1130
1140 IF R$="Y" THEN RUN
1150 PRINT : PRINT : PRINT "SO LONG...."
```



Guided Missile Commander

Programmable characters are an extremely powerful feature of the VIC-20 and Commodore 64 computers. They can be used for designing foreign character sets, or simply for making odd shapes to use in games. Character Create, a program included in this book, was employed for making programmable characters. It saved a great deal of programming time.

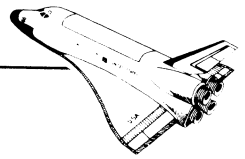


The main thrust behind Guided Missile Commander is to kill as many invading alien pods as possible in under thirty seconds. A timer in the upper right portion of the screen constantly reminds you of the time remaining. In the upper left corner of the screen is your current score. Aliens will come shooting across the screen in a horizontal motion, and at random levels, subject to your missile fire all the while. The two cursor keys control your gun's horizontal movement, and the space bar fires your missile. Once a missile is fired, the cursor keys control the horizontal position of the missile, not that of your gun turret.

To end a missile's flight before it goes off the screen, simply press the space bar, and a new one will be deployed, taking the old one's place. If you wish to end a missile's flight without launching another one, simply press Q. At the end of the game, the computer will tell you your score. Good luck in shooting the invaders from outer space!

GUIDED MISSILE COMMANDER

```
100 REM *****
110 REM * SET SCREEN *
120 REM *****
130 PRINT CHR$(147) CHR$(144)
140 POKE 36879,25
150 POKE 52,28
160 POKE 56,28
170 CLR
180 FOR I=7168 TO 7679
190 POKE I,PEEK(I+25600)
200 NEXT
210 POKE 36869,255
220 BA=7702
230 AM=30720
240 PB=8164
250 PC=8185
300 REM *****
310 REM * GET SHIP DATA *
320 REM *****
330 FOR T=0 TO 7
340 READ C
350 POKE 7168+T,C
360 NEXT
370 DATA 195,60,24,24,24,24,255,255
400 REM *****
410 REM * GET BULLET DATA *
420 REM *****
430 FOR T=0 TO 7
440 READ C
```

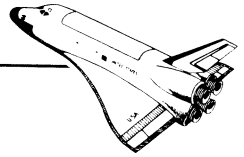


```
450 POKE 7176+T,C
460 NEXT
470 DATA 16,16,16,40,40,40,108,130
500 REM *****
510 REM * GET ENEMY DATA *
520 REM *****
530 FOR T=0 TO 7
540 READ C
550 POKE 7184+T,C
560 NEXT
570 DATA 0,0,0,60,66,255,66,60
600 REM *****
610 REM * START GAME *
620 REM *****
630 PRINT CHR$(147) "SCORE:";
640 PRINT TAB(15) "TIME:"
650 PP=8164
660 TI$="000000"
670 AN=0
680 SC=0
690 MP=PP
700 MS=0
710 Y=INT(RND(1)*12+1)
720 AN=AN+1
730 EP=BA+22*Y
740 ES=EP
750 POKE EP,2
760 POKE EP+AM,0
770 POKE PP,0
780 POKE PP+AM,0
790 K=PEEK(197)
800 T$=RIGHT$(TI$,2)
810 T$=RIGHT$("0"+STR$(30-VAL(T$)),2)
820 IF VAL(T$)<0 THEN 1430
830 PRINT CHR$(19) TAB(20) T$
840 IF K<>31 AND K<>23 THEN 870
850 IF MS=1 THEN 870
```

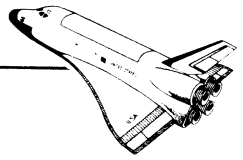
```

860 POKE PP,32
870 IF MS=1 THEN POKE MP,32 : GOTO 900
880 IF K=31 AND PP>PB THEN PP=PP-1
890 IF K=23 AND PP<PC THEN PP=PP+1
900 IF K=31 AND PP>PB THEN MP=MP-1
910 IF K=23 AND PP<PC THEN MP=MP+1
920 IF K=32 THEN MS=1
930 IF K=32 OR K=36 THEN MP=PP
940 IF K=48 THEN MS=0 : GOTO 1050
950 IF K<>31 AND K<>23 THEN 990
960 POKE PP-1,32
970 POKE PP,0
980 POKE PP+AM,0
990 IF MS=0 THEN 1050
1000 MP=MP-22
1010 IF MP<BA THEN MP=PP : MS=0 : GOTO 1050
1020 IF PEEK(MP)=2 THEN 1230
1030 POKE MP,1
1040 POKE MP+AM,0
1050 POKE EP,32
1060 EP=EP+1
1070 IF EP>ES+21 THEN 710
1080 POKE EP,2
1090 POKE EP+AM,0
1100 GOTO 790
1200 REM *****
1210 REM * ENEMY HIT *
1220 REM *****
1230 FOR T=1 TO 10
1240 POKE MP,1
1250 FOR S=1 TO 30
1260 NEXT
1270 POKE MP,2
1280 FOR S=1 TO 30
1290 NEXT
1300 NEXT
1310 POKE MP,32
1320 POKE 197,64

```

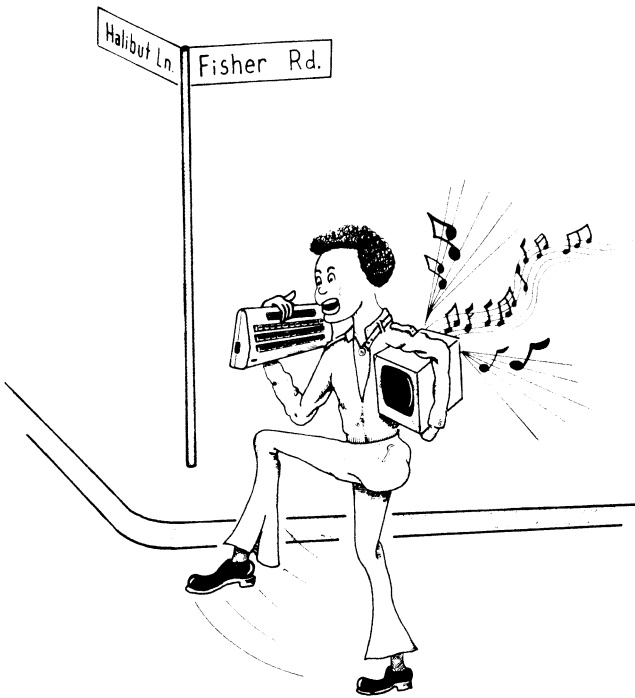



```
1330 POKE PP,32
1340 SC=SC+1
1350 PRINT CHR$(19) TAB(6) SC
1360 GOTO 690
1400 REM *****
1410 REM * OUT OF TIME *
1420 REM *****
1430 FOR T=1 TO 20
1440 POKE PP,32
1450 FOR S=1 TO 50
1460 NEXT
1470 POKE PP,0
1480 FOR S=1 TO 50
1490 NEXT
1500 NEXT
1510 PRINT CHR$(147)
1520 POKE 36869,240
1530 PRINT "YOU'RE OUT OF TIME!"
1540 PRINT
1550 PRINT "YOU KILLED" SC "OUT OF" AN;
1560 PRINT "POSSIBLE!"
1570 PRINT
1580 PRINT "PLAY AGAIN (Y/N)"
1590 POKE 198,0
1600 POKE 197,64
1610 GET R$: IF R$="" THEN 1610
1620 IF R$="N" THEN END
1630 IF R$(">")"Y" THEN 1610
1640 PRINT CHR$(147)
1650 POKE 36869,255
1660 GOTO 630
```

Music Machine

The VIC's potential as a music-making machine is absolutely fantastic. Its sound producing capabilities are far better than many other computers out today, in both quality and ease of use. With four different voices covering a multiple range of octaves, it is a very nice machine to program. The potential is great, especially if you're programming in machine language, and even more is possible if you have added memory.

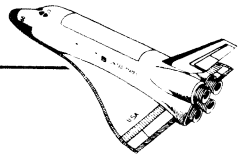


Music Machine demonstrates the VIC's various sound registers and volume control. It also includes a routine to adjust the release value of a note. Your notes can be quick and sharp, or they can be long and drawn out. Understanding and using Music Machine is quite simple. If you would like to adjust the volume, enter a + or a -, depending on whether you would like to raise or lower the volume accordingly. Entering a 1 through a 4 will change the voice register. Using it, you can give your notes a low, medium or high pitch. Entering a 4 will activate the white noise register, which will produce a raspy, grinding sound, an effect useful for crashes, skidding, and various other unpleasantries.

Lastly, to adjust the length of a note, press the cursor down key or hold in the shift key and press cursor up. This will, as previously stated, control how long a particular note is played. Many enjoyable and not-so-enjoyable effects can be made with this program. Have fun!

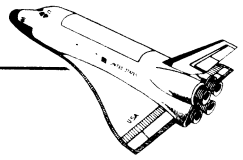
MUSIC MACHINE

```
100 PRINT CHR$(147) CHR$(144)
110 POKE 36879,25
120 VL=36878
130 V(1)=36874
140 V(2)=36875
150 V(3)=36876
160 V(4)=36877
170 DELAY=50
180 VOICE=V(3)
190 CV=5
200 POKE VL,CV
300 REM *****
310 REM * PRINT SCREEN *
320 REM *****
330 PRINT CHR$(19) TAB(4) "MUSIC MACHINE"
340 PRINT
```



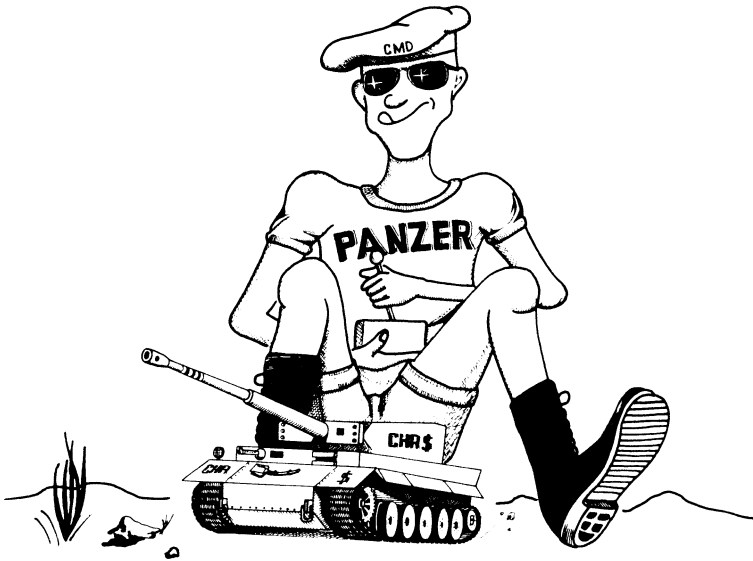
```
350 PRINT
360 PRINT "ENTER (1-4) FOR VOICE"
370 PRINT "+ OR - FOR VOLUME"
380 PRINT "CRSR UP OR DOWN FOR"
390 PRINT "NOTE LENGTH"
400 PRINT
410 PRINT
420 PRINT "NOTES ARE BASED ON"
430 PRINT "ASCII FROM KEYS"
440 PRINT "PRESSED."
450 REM *****
460 REM * GET PRESS *
470 REM *****
480 GET KEY$: IF KEY$="" THEN 530
490 KEY=ASC(KEY$)+63
500 IF KEY$="+" OR KEY$="-" THEN 830
510 IF VAL(KEY$)>0 AND VAL(KEY$)<15 THEN 730
520 IF KEY$=CHR$(17) OR KEY$=CHR$(145) THEN
530
540 IF KEY>255 THEN 530
550 POKE VOICE,KEY
560 FOR T=1 TO DELAY
570 NEXT
580 POKE VOICE,0
590 GOTO 530
600 REM *****
610 REM * SET VOICE *
620 REM *****
630 VOICE=V(VAL(KEY$))
640 GOTO 530
650 REM *****
660 REM * SET VOLUME *
670 REM *****
680 IF KEY$="+" AND CV<15 THEN CV=CV+1
690 IF KEY$="-" AND CV>0 THEN CV=CV-1
700 POKE VL,CV
710 GOTO 530
```

```
900 REM *****
910 REM * SET DELAY *
920 REM *****
930 IF KEY$=CHR$(145) THEN DELAY=DELAY+50
940 IF KEY$=CHR$(17) AND DELAY>0 THEN DELAY=
DELAY 50
950 GOTO 530
```



Tank Command

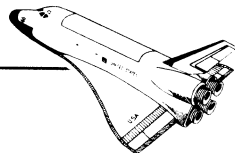
Remember the very first video games like PONG? When they hit the market, you saw them only in bowling alleys. I remember a particular one with a tank I liked very much, because I've always envisioned myself as a famous tank commander. I decided I wanted to make something like that for the VIC. Unfortunately, mine is only a 5K machine, and worse yet, it only has one joystick port! As you can guess by now, I had to settle for something less elaborate. But not to fear! Tank Command is a very neat game. It gives a perfect example of controlling a programmable character via the joystick. Your television or monitor screen will fill up with beautiful green pine trees and horrible, nasty steel mines. Your tank appears at the lower left portion of the screen, ready for your commands.



Your mission, should you decide to accept it, is to clear the field of all its mines so your company can march through safely. Controlling the tank is easy. Just move the joystick in the direction you wish to travel, and press the fire button when you wish to shoot a ground-to-ground missile. Touching a mine will destroy your tank and end the game. I tried to give a time limit on the game, but it seems that when I tried to use the joystick and the clock at the same time, the clock kept resetting itself. Perhaps those of you with more machine memory can change it so two people can battle each other. Feel free to modify it to suit your needs.

TANK COMMAND

```
100 PRINT CHR$(147) CHR$(144)
110 POKE 52,28
120 POKE 56,28
130 CLR
140 POKE 36879,25
150 POKE 36878,15
160 PA=37137
170 PB=PA+15
180 POKE PA+2,0
190 DD=PB+2
200 NO=36877
210 POKE 36869,255
220 FOR I=7168 TO 7679
230 POKE I,PEEK(I+25600)
240 NEXT
250 IN=33
260 IS=34
270 IE=35
280 IW=36
290 IR=37
300 MI=38
310 OT=IN
320 FOR I=7432 TO 7497
330 READ INFO
```

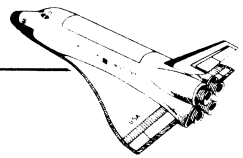



```
340 POKE I,INFO
350 NEXT
360 SC=7680
370 CS=38400
380 C1=0 : C2=0
390 TX=0 : TY=22
400 FOR T=1 TO 100
410 TP=INT(RND(1)*505)
420 POKE SC+TP,TR
430 POKE CS+TP,5
440 NEXT
450 FOR T=1 TO 30
460 MP=INT(RND(1)*505)
470 POKE SC+MP,MI
480 POKE CS+MP,6
490 NEXT
500 POKE SC+TX+22*TY,CT
510 POKE CS+TX+22*TY,0
600 REM *****
610 REM * START *
620 REM *****
630 GOSUB 1530
640 IF JV=198 THEN 630
650 IF JV=156 THEN 930
660 Y=TY : X=TX
670 IF JV=184 THEN CT=TN : IF TY>0 THEN Y=TY
  1
680 IF JV=180 THEN CT=TS : IF TY<22 THEN Y=T
  Y+1
690 IF JV=172 THEN CT=TW : IF TX>0 THEN X=TX
  -1
700 IF JV=60 THEN CT=TE : IF TX<21 THEN X=TX
  +1
710 IF CT<>0 THEN OT=CT : POKE SC+TX+22*TY,
  CT : GOTO 500
720 OT=C1
730 OB=PEEK(SC+X+22*Y)
740 IF OB=TR THEN 630
```

```

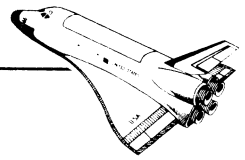
750 IF OB=MI THEN 1330
760 POKE SC+TX+22*TY,32
770 TX=X : TY=Y
780 GOTO 500
800 REM *****
810 REM * FIRE!!! *
820 REM *****
830 POKE 36877,240
840 POKE 36874,240
850 FOR T=1 TO 20 : NEXT
860 POKE 36877,0
870 POKE 36874,0
880 IF CT=TN AND TY=0 THEN 630
890 IF CT=TS AND TY=22 THEN 630
900 IF CT=TW AND TX=0 THEN 630
910 IF CT=TE AND TX=21 THEN 630
920 IF CT=TW THEN LI=0 : SP=-1 : GOTO 1090
930 IF CT=TE THEN LI=21 : SP=1 : GOTO 1080
940 IF CT=TN THEN LI=0 : SP=-1
950 IF CT=TS THEN LI=22 : SP=1
960 FOR MY=TY+SP TO LI STEP SP
970 POKE 36876,240
980 POKE 36876,0
990 PP=SC+TX+22*MY
1000 UM=PEEK(PP)
1010 POKE PP,39
1020 POKE PP+30720,2
1030 IF UM=MI THEN C1=C1+1 : GOSUB 1200
1040 IF UM=TR THEN C2=C2+1
1050 POKE PP,32
1060 IF UM=TR OR UM=MI THEN MY=SP*23 : NEXT
: GOTO 630
1070 NEXT : GOTO 630
1080 FOR MX=TX+SP TO LI STEP SP
1090 PP=SC+MX+22*TY
1100 UM=PEEK(PP)
1110 POKE PP,45
1120 POKE PP+30720,2

```



```
1130 IF UM=MI THEN CC=CC+1 : GOSUB 1200
1140 POKE PP,32
1150 IF UM=TR OR UM=M1 THEN MX=SP*23 : NEXT
: GOTO 630
1160 NEXT : GOTO 630
1200 REM *****
1210 REM * EXPLOSION *
1220 REM *****
1230 POKE 36877,128
1240 FOR T=15 TO 0 STEP 1
1250 FOR S=1 TO 400 : NEXT
1260 POKE 36878,T
1270 POKE 36877,0
1280 POKE 36878,15
1290 RETURN
1300 REM *****
1310 REM * INTO MINE *
1320 REM *****
1330 BP=SC+TX+22*TY
1340 FOR T=1 TO 5
1350 FOR S=36 TO 33 STEP -1
1360 POKE BP,S
1370 N=INT(RND(1)*128+128)
1380 POKE 36876,N
1390 FOR W=1 TO 50
1400 NEXT
1410 NEXT
1420 NEXT
1430 POKE 36876,0
1440 POKE 36869,240
1450 PRINT CHR$(147) "ANOTHER GAME (Y/N)"
1460 POKE 198,0
1470 GET Y$: IF Y#="" THEN 1470
1480 IF Y#="Y" THEN RUN
1490 END
1500 REM *****
1510 REM * JOY READ *
1520 REM *****
```

```
1530 POKE DD,127
1540 JV=PEEK(PA) AND 60
1550 JV=JV OR (PEEK(PB) AND 120)
1560 POKE DD,255
1570 POKE NO,200
1580 POKE NO,0
1590 RETURN
1600 REM *****
1610 REM * PRDG CHARS *
1620 REM *****
1630 DATA 8,107,107,127,127,127,127,99
1640 DATA 99,127,127,127,127,107,107,8
1650 DATA 0,254,254,120,127,120,254,254
1660 DATA 0,127,127,30,254,30,127,127
1670 DATA 16,56,56,124,124,254,16,16
1680 DATA 62,99,85,73,95,99,62,0
1690 DATA 0,16,16,16,16,16,16,0
```



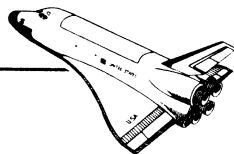
Memory Master

Having a good memory is of benefit to anyone. Whether you're into Pre-Law, Pre-Med, Biology or dumping the trash, it can help you. Programmers with a good memory can easily remember PEEK and POKE locations, and they hardly ever need to use the reference manual. Believe me, this can save a lot of time. There are various ways to improve your memory. Some people listen to cassettes played while they are sleeping, and others read manuals on the subject. There is also a third way, and that's just plain practice. Memory Master is a program to do just that. Not only is it fun, but it also helps improve your recording skills. Based on color code recognition, it helps you recognize specific patterns on cue. There are many different forms of memory, and this program helps you with just one of them. (Spelling Practice helps you with another!)

Memory Master is easy to get going. When the program is RUN, you will notice that the screen is divided up into four color-coded sections. Next, Memory Master will play a note for you and flash the matching portion of the screen. You, in return, must memorize and then replay that note, using the keys Q, W, A, and S, each of which represents one section of the screen. Notice how these four keys make a square, with each one representing a corner of the box. The note that Memory Master played for you is stored in memory, and another is added on. Now both notes are played, and you are expected to repeat them. Seem easy? Not for long it isn't! This process goes on until you blow it and enter a wrong note. When this is done, you will hear a sour note being played, and the screen will flash in a variety of colors. This game is like the commercial game, Simon. As a cheating device, I have included a replay function. That is, if you want to hear the series of notes played over, press F1, and the entire collection will be played again so that you may enter it correctly. When you lose (you will!), the total number of notes that you memorized will be displayed. Improving your score should not be difficult.

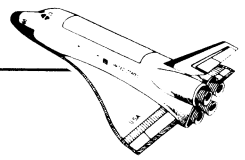
MEMORY MASTER

```
100 PRINT CHR$(147) CHR$(144)
110 POKE 36879,25
120 POKE 36878,15
130 V1=36876
140 DIM SQ(99)
150 FOR T=0 TO 3
160 NO(T)=130+T*20
170 NEXT
180 FOR LOC=7680 TO 8163
190 POKE LOC,160
200 NEXT
210 FOR LOC=7580 TO 7580+98
220 READ BYTE
230 POKE LOC,BYTE
240 NEXT
250 FOR T=0 TO 99
260 SQ(T)=-1
270 NEXT
280 FOR T=0 TO 3
290 POKE 253,T
300 SYS 7580
310 NEXT
320 FOR T=1 TO 500
330 NEXT
340 SQ=0
400 REM *****
410 REM * PLAY SEQ *
420 REM *****
430 FOR TM=0 TO SQ
440 SF=SQ(TM)
450 IF SF<0 THEN SF=INT(RND(1)*4) : SQ(SQ)=S
F
460 GOSUB 830
470 NEXT TM
500 REM *****
510 REM * GET RESP *
520 REM *****
```



```
530 FOR TM=0 TO SQ
540 R=PEEK(197) : IF R=64 THEN 540
550 IF R=39 THEN TM=SQ+1 : NEXT : GOTO 430
560 IF R<>48 AND R<>9 AND R<>17 AND R<>41 TH
EN 540
570 IF R=48 THEN SF=0
580 IF R=9 THEN SF=1
590 IF R=17 THEN SF=2
600 IF R=41 THEN SF=3
610 IF SF<>SQ(TM) THEN V1=36877
620 GOSUB 830
630 IF SF<>SQ(TM) THEN 930
640 NEXT TM
700 REM *****
710 REM * ANOTHER *
720 REM *****
730 SQ=SQ+1
740 FOR T=1 TO 500
750 NEXT
760 GOTO 430
770 END
800 REM *****
810 REM * FLASH BOX *
820 REM *****
830 POKE 253,SF
840 SYS 7580
850 POKE V1,NO(SF)
860 FOR T=1 TO 100
870 NEXT
880 POKE V1,0
890 RETURN
900 REM *****
910 REM * LOST *
920 REM *****
930 FOR T=0 TO 255
940 POKE 36879,T
950 NEXT
960 POKE 36879,25
```

```
970 PRINT CHR$(147)
980 PRINT "YOU MEMORIZED TO " G0
990 PRINT "SEQUENCES."
1000 POKE 198,0
1010 END
1100 REM *****
1110 REM * FLASH ROUTINE *
1120 REM *****
1130 DATA 165,253,10,170,189,243,29,133
1140 DATA 251,141,60,3,232,189,243,29
1150 DATA 133,252,141,61,3,169,1,133
1160 DATA 254,32,215,29,160,150,162,0
1170 DATA 202,200,253,136,208,248,166,253
1180 DATA 189,251,29,133,254,173,60,3
1190 DATA 133,251,173,61,3,133,252,32
1200 DATA 215,29,96,162,10,160,10,165
1210 DATA 254,145,251,136,16,251,24,165
1220 DATA 251,105,22,133,251,165,252,105
1230 DATA 0,133,252,202,16,231,96,0
1240 DATA 150,11,150,242,150,253,150,6
1250 DATA 5,2,7
```

Pick A Number

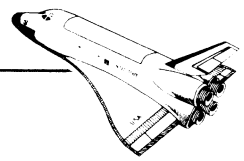
Just about every child enjoys guessing games. For those of you new to programming, it is a good place to start understanding how games are made using random numbers. You may want to enhance it, but in the meantime, your children should enjoy it immensely as it is.



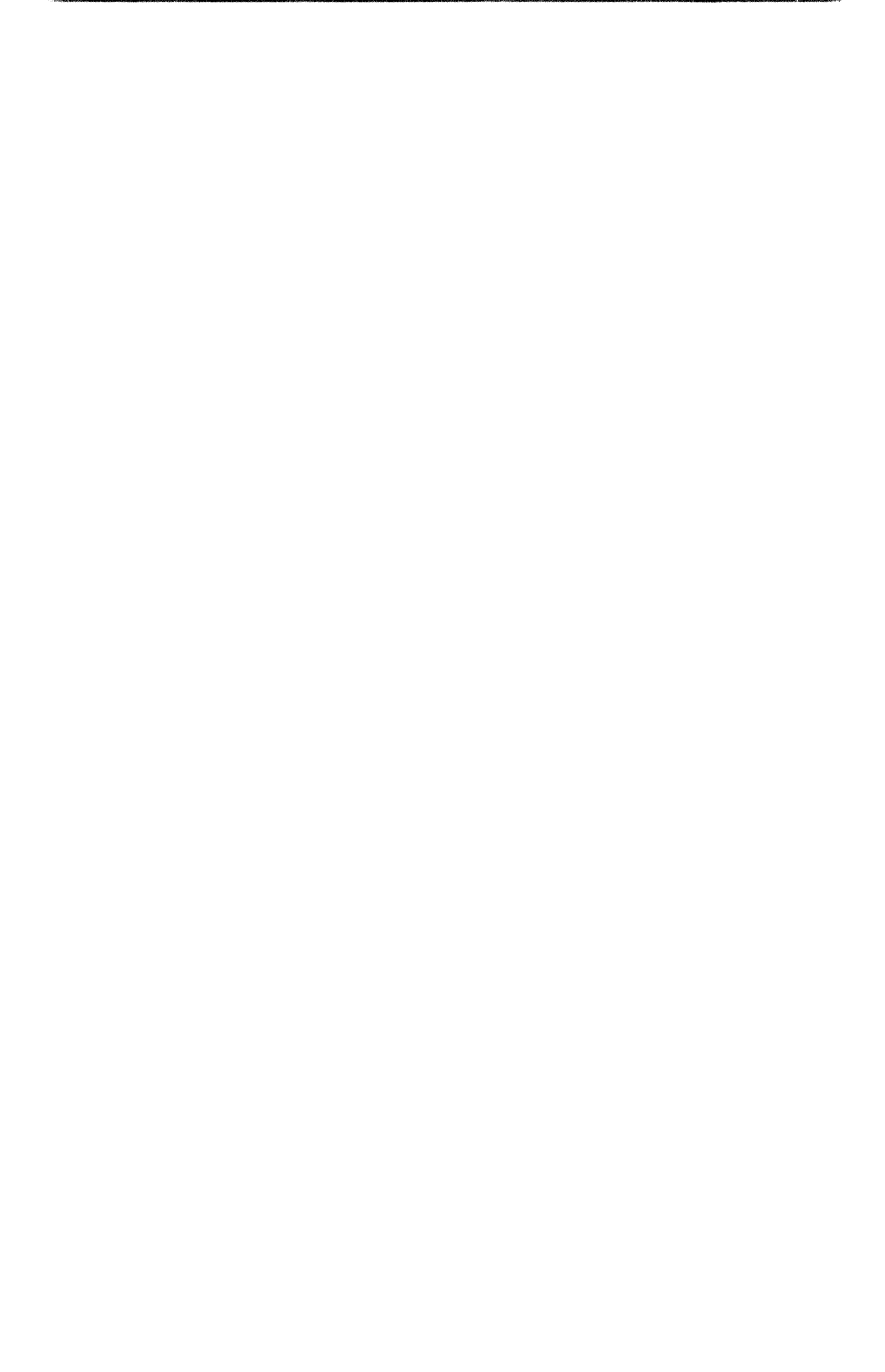
Pick a Number randomly selects a number from 1–100 and asks you to find the number it has picked. The computer tells you if your guess is too low or too high. Should you win, the computer flashes the screen in a variety of colors. After each wrong guess, you must press a key to continue. The number of guesses it took for you to determine the number is posted at the end of the game. See if you can determine the most efficient strategy for deducing the computer's number. If you play the numbers, do not use this program!

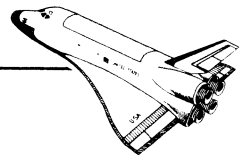
PICK A NUMBER

```
100 PRINT CHR$(147) CHR$(144)
110 POKE 36879,25
200 REM *****
210 REM * GET GUESS *
220 REM *****
230 ANS=INT(RND(1)*100+1) : G=0
240 GOSUB 630
250 PRINT CHR$(19) : PRINT : PRINT
260 PRINT "GUESS IT:";
270 GUESS$=""
280 INPUT GUESS$
290 IF GUESS$="" THEN 240
300 G=G+1
310 GUESS=VAL(GUESS$)
320 IF GUESS<1 OR GUESS>100 THEN 930
330 IF GUESS<ANS THEN 530
340 IF GUESS=ANS THEN 1030
400 REM *****
410 REM * GUESS IS > ANSWER *
420 REM *****
430 PRINT : PRINT "TOO HIGH!"
440 GOSUB 730
450 GOSUB 830
460 GOTO 240
500 REM *****
510 REM * GUESS IS < ANSWER *
520 REM *****
530 PRINT : PRINT "TOO LOW!"
540 GOSUB 730
550 GOSUB 830
560 GOTO 240
600 REM *****
610 REM * PRINT TOP BORDER *
620 REM *****
630 PRINT CHR$(147) TAB(5);
640 PRINT "NUMBER GUESS"
```



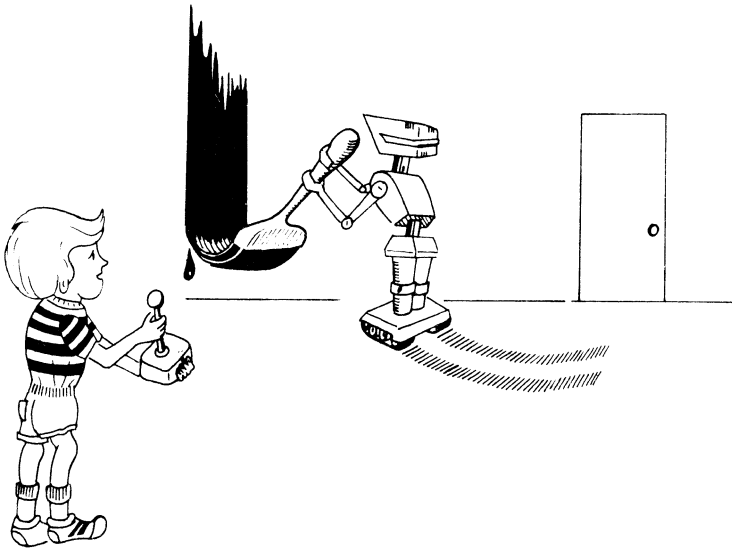
```
650 PRINT TAB(5) "*****"
660 RETURN
700 REM *****
710 REM * GET PRESS *
720 REM *****
730 GET R$: IF R#="" THEN 730
740 RETURN
800 REM *****
810 REM * POSITION CURSOR *
820 REM *****
830 PRINT CHR$(19) : PRINT : PRINT
840 PRINT : PRINT
850 RETURN
900 REM *****
910 REM * OUT OF BOUNDS *
920 REM *****
930 PRINT
940 PRINT "OUT OF BOUNDS 1-100!"
950 GOSUB 730
960 GOSUB 830
970 GOTO 240
1000 REM *****
1010 REM * GUESSED IT *
1020 REM *****
1030 PRINT
1040 PRINT "DONE IN" G "GUESSES!"
1050 FOR T=0 TO 255
1060 POKE 36879,T : NEXT
1070 POKE 36879,25
1080 GOSUB 730
1090 RUN
```





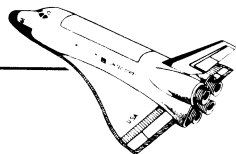
Doodle

High resolution graphics is one of my favorite aspects of a computer. There is nothing I like better than a scrolling graphics display, or a well drawn picture. Colors also add to any display. The unexpanded VIC is somewhat limited in its high resolution capabilities. The VIC is capable of displaying two colors in an 8×8 block. The resolution is also limited to 64 vertical dots \times 64 horizontal dots. Doodle is a great program for doing exactly as its name implies, doodling. To control the dots' north, south, east and west direction, type I, K, J or L, respectively. To draw in black, press 1, and to draw in white, press 2. If you are not too pleased with the picture you are drawing, press CLR to erase the hi-res screen. Here is the listing:

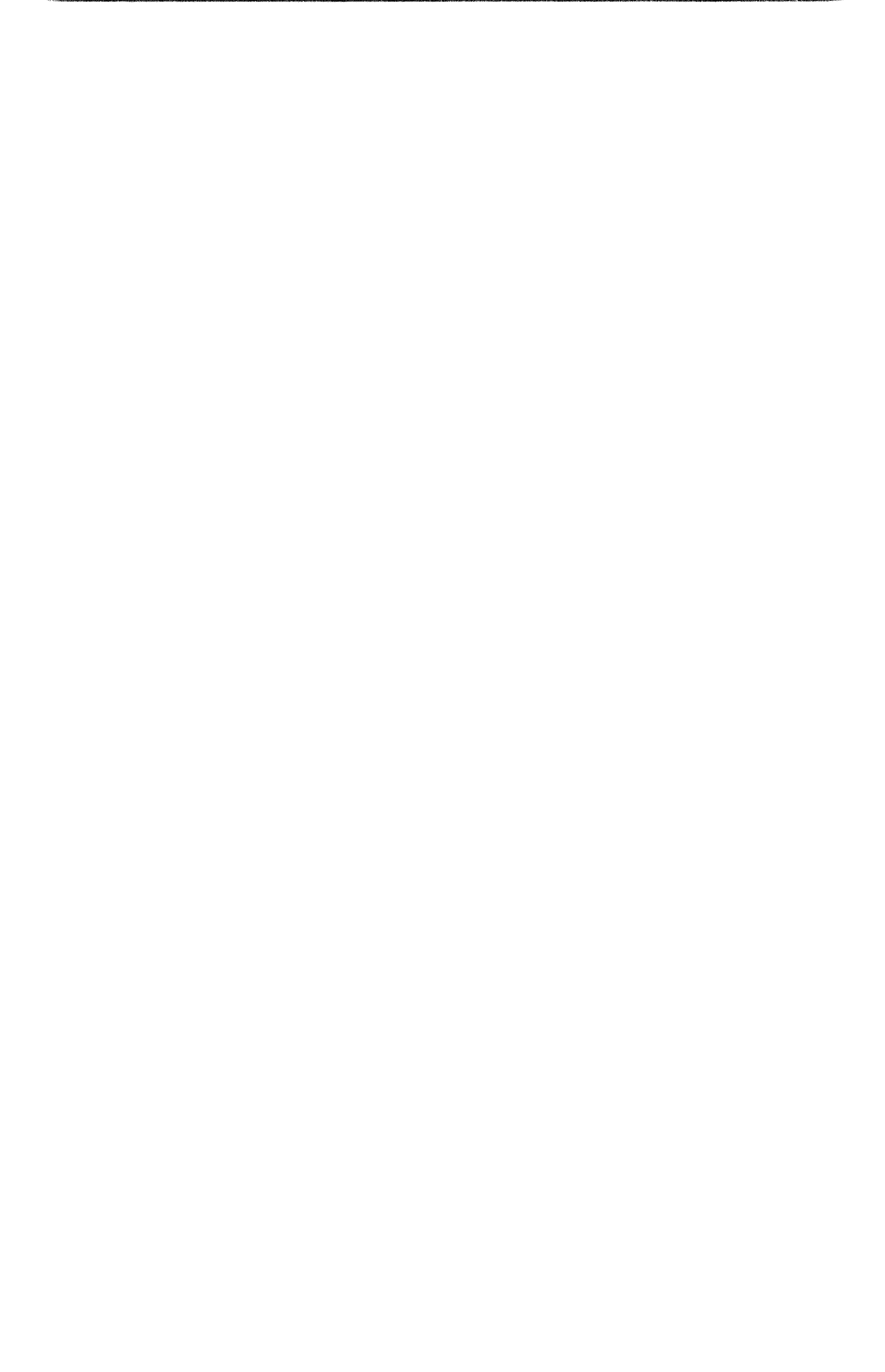


DOODLE

```
100 PRINT CHR$(147); CHR$(144)
110 POKE 36879,10
120 CO=1
200 REM *****
210 REM * CHAR SET POINTER *
220 REM *****
230 POKE 36869,255
300 REM *****
310 REM * CLR HI RES SCREEN *
320 REM *****
330 FOR I=7160 TO 7679
340 POKE I,0
350 NEXT
400 REM *****
410 REM * BLANK OUT SCREEN *
420 REM *****
430 FOR I=7680 TO 8195
440 POKE I,160
450 NEXT
500 REM *****
510 REM * PUT PRG CHRS ON SCREEN *
520 REM *****
530 FOR L=0 TO 7
540 FOR M=0 TO 7
550 POKE 7680+M*22+L,L*8+M
560 NEXT
570 NEXT
580 X=0
590 Y=0
600 REM *****
610 REM * START *
620 REM *****
630 K=PEEK(197)
640 IF K=62 THEN 330
650 IF K=0 THEN POKE 36879,8 : CO=0
660 IF K=56 THEN POKE 36879,10 : CO=1
```



```
630 IF K=20 AND X>0 THEN X=X-1
640 IF K=21 AND X<63 THEN X=X+1
650 IF K=12 AND Y>0 THEN Y=Y-1
660 IF K=44 AND Y<63 THEN Y=Y+1
700 REM *****
710 REM * EQUATION FOR DOT *
720 REM *****
730 CHAR=INT(X/8)*8+INT(Y/8)
740 ROW=(Y/8-INT(Y/8))*8
750 BYTE=7168+CHAR*8+ROW
760 BIT=7-(X AND 7)
800 REM *****
810 REM * PUT DOT ON SCREEN *
820 REM *****
830 IF CO=0 THEN POKE BYTE,PEEK(BYTE) AND (2
55-2^BIT) : GOTO 630
840 POKE BYTE,PEEK(BYTE) OR (2^BIT)
850 GOTO 630
```

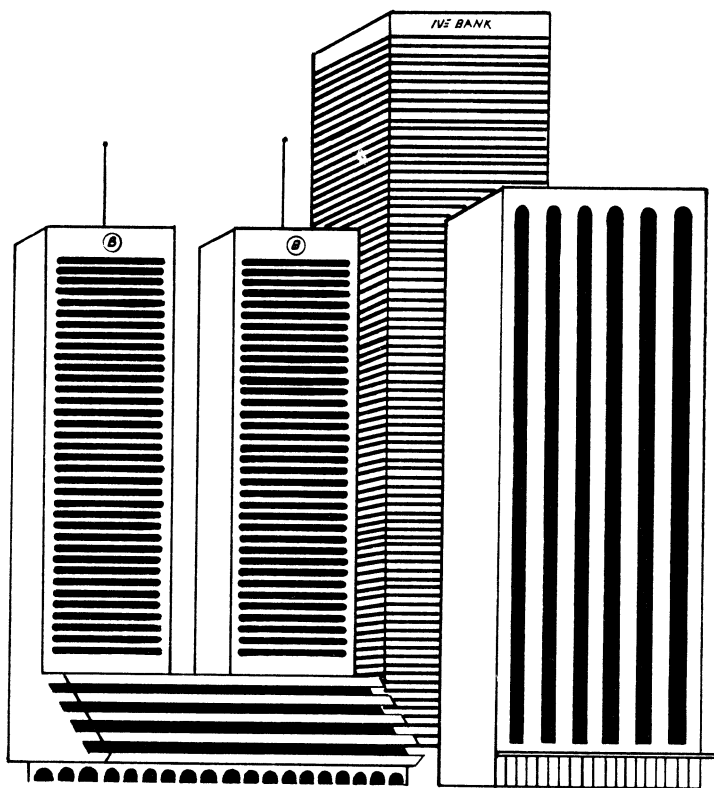


Section 4

Business Programs

- Appointment Scheduler
- Bar Graph
- Cash Flow
- Interest Rate
- Checkbook
- Inventory



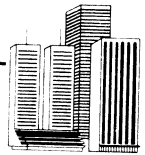


BUSINESS PROGRAMS

Designing a business program for a general audience is somewhat difficult since many such programs are custom-tailored to a specific company's or person's specifications. What type of program in this category can apply to a general group of business applications? The programs included in this section can be modified for a specific business need, but most of them are written for any business with general business applications.

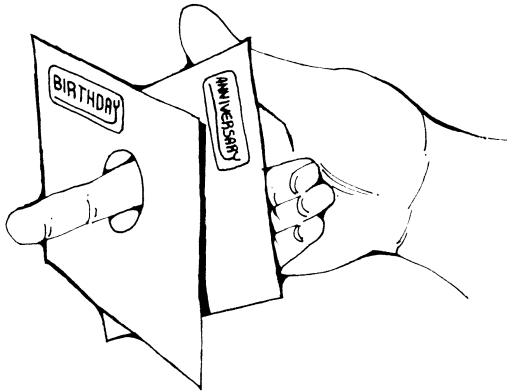
The five business programs cover a wide range of practical applications. The Appointment Scheduler provides the busy business executive with a clear record of his various appointments and other dates on a monthly basis. Bar Graph gives a graphic representation of various business trends. This allows the user to see, at a glance, how a business, a stock or any other value changes over a calendar year. Cash Flow shows the real value of an investment with a known cash flow over a given number of years. Interest Rate quickly determines how much money can be made with a particular investment over the years, depending on whether it is compounded yearly, quarterly or monthly. The Checkbook program is an accurate and simple way to keep track of what you earn and spend in business, and the Inventory program will record files of what stock you have and what stock you have sold. It is a handy database program for the small business person.

The programs can, and should, be changed to meet your own specific needs. In making such changes, not only can you learn something about programming, but you can also create new ways to evaluate business decisions.



Appointment Scheduler

Business schedules can become hectic if they aren't properly organized, and if an appointment is missed, it can mean the difference between an important contract being won or lost.



Using Appointment Scheduler, it is easy to keep track of everything from a business meeting to your anniversary. Everything is menu driven. To use it for the first time, you must initialize a file for that month. You do this only once, at the first of the month (reinitializing a file will destroy all the data you have entered for that month). To initialize a file, type a 5 on the master menu. Next, it will ask you the month for which you would like to create a file. Simply enter a three-digit code for the month (e.g., JAN, FEB, MAR). Now you're on your way. The next thing to do is to read in your appointments from the disk. If you haven't entered anything yet, that's okay; it won't do anything. Now you can edit and add dates to the file. Simply enter a 4, and you will be prompted with the following:

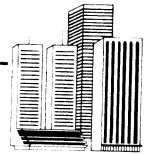
```
<A>DD/<D>ELETE/  
<L>IST/<Q>UIT:
```

Enter an A for add. You will be prompted for a DATE and a WHAT. Enter the date of your appointment and hit RETURN, then enter the significance of the date. After doing this, you will see all your appointments listed on the screen, and the same questions will be asked again. If, at any time, you wish to delete an entry, type a D. The program will ask you which entry you wish to delete. Enter the number to the left of the date of your function, and hit RETURN. Your dates will be relisted and your function will no longer be listed. Type a Q when you are finished entering dates. At this point, you can either write the dates to disk or print them out. First, to be safe, let's write them to disk. Enter a 3 and the month (3 characters), and the computer will do its job. Now that you are back into the main menu again, you are ready to get a printout of all your appointments. Enter a 1. You will be asked if you wish to direct the output to the <P>RINTER or the <S>CREEN. Enter a P or an S accordingly. When it's all through listing your appointments, the program will ask you to press RETURN. Upon doing so, you will be returned to the master menu again. Now you're on your own.

Remember to create a month only once, and that before you ever do anything to a month, you must read it in first.

APPOINTMENT SCHEDULER

```
100 PRINT CHR$(147) CHR$(144)
110 POKE 36879,25
120 C$=CHR$(13)
130 DIM DAT$(31),WHAT$(31)
140 OPEN 3,0
200 REM *****
210 REM * PRINT SCREEN *
220 REM *****
230 PRINT CHR$(147) TAB(6) "DATE BOOK"
240 PRINT : PRINT : PRINT
250 PRINT "MENU:" : PRINT
260 PRINT "<1> PRINT OUT DATES"
```

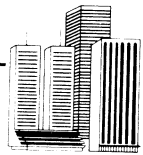


```
270 PRINT "<2> READ IN DATES"
280 PRINT "<3> WRITE OUT DATES"
290 PRINT "<4> EDIT DATES"
300 PRINT "<5> CREATE A MONTH"
310 PRINT "<6> QUIT"
320 PRINT : PRINT "CHOICE:";
330 INPUT#3,A#
340 A=VAL(A#)
350 IF A<1 OR A>6 THEN 330
360 ON A GOTO 430,730,930,1130,1730
370 PRINT : PRINT : END
400 REM *****
410 REM * PRINT DATES *
420 REM *****
430 PRINT CHR$(147) "<P>RINTER<S>CREEN:";
440 INPUT#3,W#
450 IF W#<>"P" AND W#<>"S" THEN 230
460 IF W#="S" THEN 510
470 OPEN 1,4
480 PRINT#1,C# C#
490 PRINT#1,"APPOINTMENTS FOR THE MONTH OF "
MO#
500 PRINT#1,C#
510 PRINT : PRINT
520 FOR T=0 TO CNTR-1
530 IF CNTR=0 THEN PRINT "NONE ENTERED" : GO
TO 560
540 IF W#="P" THEN PRINT#1,DATE$(T) C# WHAT$(
T) C# C#
550 PRINT DATE$(T) : PRINT WHAT$(T) : PRINT
560 NEXT
570 POKE 197,64
580 IF PEEK(197)=64 THEN 570
590 IF W#="P" THEN CLOSE 1
600 GOTO 230
700 REM *****
710 REM * READ IN DATES *
720 REM *****
```

```

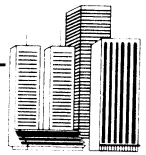
730 PRINT CHR$(147) "FOR THE MONTH OF:";
740 INPUT#3,MO$
750 IF LEN(MO$)<3 THEN 230
760 OPEN 2,8,2,"0:DATES-"+MO$+".S,R"
770 CNTR=0
780 INPUT#2,DAT$(CNTR),WHAT$(CNTR)
790 IF DAT$(CNTR)="" OR DAT$(CNTR)="EOF" THE
N 810
800 CNTR=CNTR+1 : GOTO 780
810 CLOSE 2 : GOTO 230
900 REM *****
910 REM * WRITE OUT DATES *
920 REM *****
930 PRINT CHR$(147) "FOR THE MONTH OF:";
940 INPUT#3,MO$
950 IF LEN(MO$)<3 THEN 230
960 OPEN 15,8,15
970 PRINT#15,"S0:DATES-"+MO$
980 CLOSE 15
990 OPEN 2,8,2,"0:DATES-"+MO$+".S,W"
1000 FOR T=0 TO CNTR
1010 PRINT#2,DAT$(T) C$ WHAT$(T)
1020 NEXT
1030 CLOSE 2 : GOTO 230
1100 REM *****
1110 REM * EDIT DATES *
1120 REM *****
1130 PRINT CHR$(147)
1140 FOR T=0 TO CNTR-1
1150 IF CNTR=0 THEN PRINT "NONE ENTERED" : G
OTO 1190
1160 PRINT T+1 " DATE:";DAT$(T)
1170 PRINT TAB(4) "WHAT:";WHAT$(T)
1180 PRINT
1190 NEXT
1200 PRINT : PRINT "(A)DD/(D)DELETE"
1210 PRINT "(L)IST/(O)UIT:";
1220 INPUT#3,A$

```

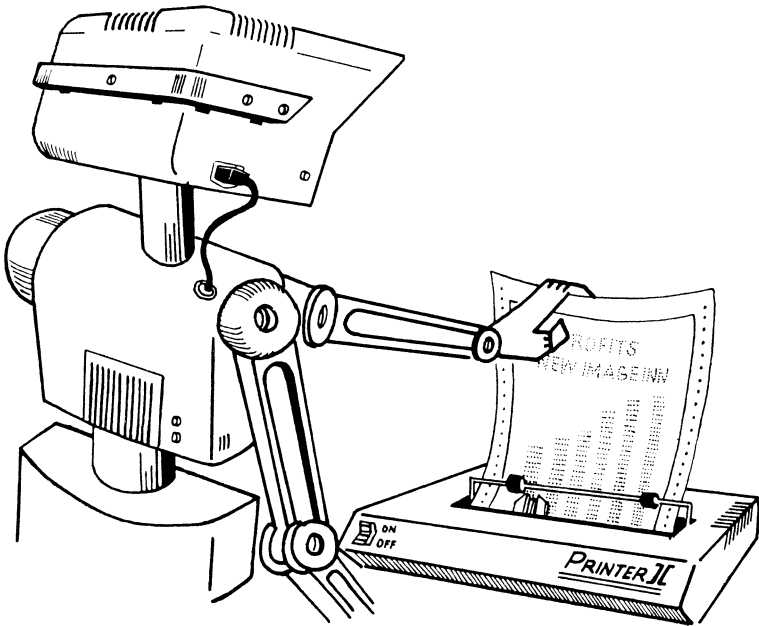
```
1230 PRINT : PRINT
1240 IF A$="Q" THEN 230
1250 IF A$="L" THEN 1130
1260 IF A$="D" THEN 1530
1300 REM *****
1310 REM * ADD DATE *
1320 REM *****
1330 IF CNTR>=31 THEN 1130
1340 PRINT "DATE:";
1350 INPUT#3,D$
1360 IF D$="Q" THEN 230
1370 PRINT
1380 PRINT "WHAT:";
1390 INPUT#3,W$
1400 IF W$="Q" THEN 230
1410 DAT$(CNTR)=D$
1420 WHAT$(CNTR)=W$
1430 CNTR=CNTR+1
1440 DAT$(CNTR)="EOF"
1450 WHAT$(CNTR)="EOF"
1460 GOTO 1130
1500 REM *****
1510 REM * DELETE DATE *
1520 REM *****
1530 PRINT "ITEM# TO DELETE:";
1540 INPUT#3,NUM
1550 IF NUM<1 OR NUM>CNTR THEN 1130
1560 FOR T=NUM-1 TO CNTR-1
1570 DAT$(T)=DAT$(T+1)
1580 WHAT$(T)=WHAT$(T+1)
1590 NEXT
1600 CNTR=CNTR-1
1610 GOTO 1130
1700 REM *****
1710 REM * CREATE A MONTH *
1720 REM *****
1730 PRINT CHR$(147) "MONTH TO CREATE:";
1740 INPUT#3,MO$
```

```
1750 IF LEN(MO#)<>3 THEN 230
1760 OPEN 15,8,15
1770 PRINT#15,"S0:DATES "+MO#
1780 CLOSE 15
1790 OPEN 2,8,2,"0:DATES-"+MO#+".S.W"
1800 PRINT#2,"EOF" C# "EOF"
1810 CLOSE 2
1820 GOTO 230
```



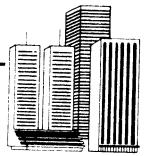
Bar Graph

It's always nice to see profits or losses, buying trends, stock prices, and similar business changes in chart or bar graph form. It gives the viewer a much clearer and more useful understanding of the data. That's why the Bar Graph program for business was constructed. It is extremely simple in design and you can enhance it for your particular business applications. All it asks you for is twelve monthly amounts to graph. Once they have been entered, press a key to view the graph. Each section of the bars is represented by a VIC-20 graphics character. Once the graph is shown, you may press any key to start the program over again (*Remember: Buy LOW, Sell HIGH*).



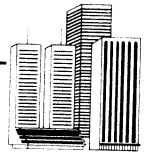
BAR GRAPH

```
100 REM *****
110 REM * SET VARIABLES *
120 REM *****
130 M=30720
140 AM=22
150 F=1.2
160 D=99
170 DIM MNTH$(11),MNTH(11)
180 FOR T=0 TO 11
190 READ MNTH$(T)
200 NEXT
210 DATA "JAN","FEB","MAR","APR","MAY"
220 DATA "JUN","JUL","AUG","SEP","OCT"
230 DATA "NOV","DEC"
300 REM *****
310 REM * SET SCREEN *
320 REM *****
330 PRINT CHR$(147) CHR$(144);
340 POKE 36879,25
350 PRINT TAB(6) "BAR GRAPH"
360 PRINT
370 PRINT
380 PRINT "ENTER 12 MONTHLY AMTS"
390 PRINT
400 FOR T=0 TO 11
410 PRINT MNTH$(T) " : ";
420 DAT$=""
430 INPUT DAT$
440 MNTH(T)=VAL(DAT$)
450 NEXT
460 PRINT
470 PRINT "PRESS A KEY TO VIEW"
480 GET R$ : IF R$="" THEN 480
500 REM *****
510 REM * SHOW GRAPH *
520 REM *****
```



```
530 PRINT CHR$(147);
540 BAS=7683
550 FOR T=0 TO 11
560 IF MNTH(T)<19 THEN NEXT : GOTO 630
570 T=12
580 NEXT
590 FOR T=0 TO 11
600 MNTH(T)=MNTH(T)/F
610 NEXT
620 GOTO 550
630 FOR T=0 TO 11
640 PRINT MNTH$(T)
650 IF MNTH(T)=0 THEN 690
660 FOR S=BAS TO BAS+MNTH(T)-1
670 POKE S,D
680 POKE S+M,0
690 NEXT
700 BAS=BAS+AM
710 NEXT
720 PRINT
730 PRINT "PRESS ANY KEY TO RUN"
740 GET R$: IF R#="" THEN 740
750 RUN
900 OPEN 15,0,15
910 INPUT#15,A$,B$
```





Cash Flow

This type of program is used by many different business people to gain knowledge when deciding on an investment. With a given investment and a known cash flow, the program determines what the rate of inflation would have to be to equal the initial investment. This percentage is evaluated against known rates of inflation to see whether there would be any real gain in such an investment. For example, let's say that a company was thinking of investing \$100,000 in a project that would have the following cash flow:

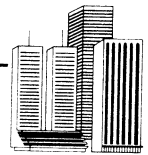
Year 1 \$50,000
Year 2 \$40,000
Year 3 \$30,000
Year 4 \$20,000
Year 5 \$10,000

If the inflation rate were just under thirty-three percent for each of the five years, the investor would break even. However, since inflation, in the United States at least, is not that high, the real return on the initial investment would be pretty good. It does not matter whether the cash flow increases or decreases over the years of the investment, the program will determine what the percent on the investment would be for the period of return.

There is some error built into the program to save execution time. If you wish to increase or decrease what would be an acceptable error, you may do so by changing both the percentage increment and the level of error. It also provides a user entered "starting point" so that you can guess what the rate of inflation would have to be to break even. The program shows you how close you are and then works up or down to find the correct rate. Again, this can be changed to a fixed beginning point if you wish.

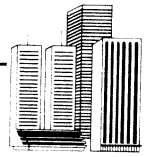
CASH FLOW

```
100 REM *****
110 REM * INFORMATION *
120 REM *****
130 PRINT CHR$(147); CHR$(144)
140 POKE 36879.25
150 PRINT
160 PRINT
170 INPUT "INIT. INVESTMENT:";PRINCIPAL
180 PRINT
190 PRINT
200 INPUT "YEARS OF CASH FLOW:";YEARS
210 PRINT
220 PRINT
230 DIM CF(YEARS),CR(YEARS)
240 FOR X=1 TO YEARS
250 PRINT "FLOW FOR YEAR" X;
260 INPUT CF(X)
270 NEXT
280 PRINT
290 INPUT "EXPECTED RATE:";RATE
300 RATE=RATE/100
400 REM *****
410 REM * CALCULATE RATE *
420 REM *****
430 FOR X=1 TO YEARS
440 CR(X)=(CF(X)-(CF(X)*RATE))
450 SUM=SUM+CR(X)
460 NEXT
500 REM *****
510 REM * ERROR LEVEL SET TO $300 *
520 REM *****
530 IF SUM<(PRINCIPAL*300) THEN 800
540 IF SUM>(PRINCIPAL*300) THEN 900
600 REM *****
610 REM * THE ACTUAL RATE *
620 REM *****
```

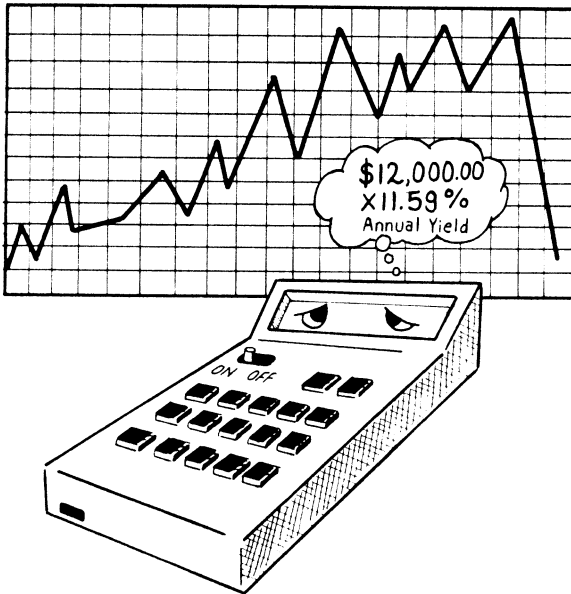
```
630 PRINT "ACTUAL RATE IS:";
640 R=RATE*100
650 PRINT R "%"
660 PRINT "ABS. ERROR IS $" SUM-PRINCIPAL
670 PRINT
680 PRINT
690 PRINT "ANOTHER CALC.(Y/N)"
700 GET A$: IF A$="" THEN 700
710 IF A$="Y" THEN RUN
720 IF A$<>"N" THEN 700
730 END
800 REM *****
810 REM * TOO LOW *
820 REM *****
830 PRINT RATE*100 "%","$" SUM
840 IF (PRINCIPAL-SUM)>1000 THEN RATE=RATE-.
01 : GOTO 860
850 RATE=RATE-.001
860 SUM=0
870 GOTO 430
900 REM *****
910 REM * TOO HIGH *
920 REM *****
930 PRINT RATE*100 "%","$" SUM
940 IF (SUM-PRINCIPAL)>1000 THEN RATE=RATE+.
01 : GOTO 960
950 RATE=RATE+.001
960 SUM=0
970 GOTO 430
```





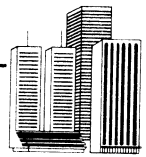
Interest Rate

When you put money into interest earning programs such as the money market, T-Bills, or even a bank account, it's nice to watch your money grow. When interest is compounded with the principal, it can grow at a geometric rate. That's really nice to watch! This little program will show you the growth rate when the initial investment, the interest rate, and the period of compounding are known. While entering the figures, you will quickly be able to see the difference between interest rates and compounding periods. The program is quite simple and you can easily add modifications for your own particular application. (If you are involved in sales, customers like seeing their investments grow too!)

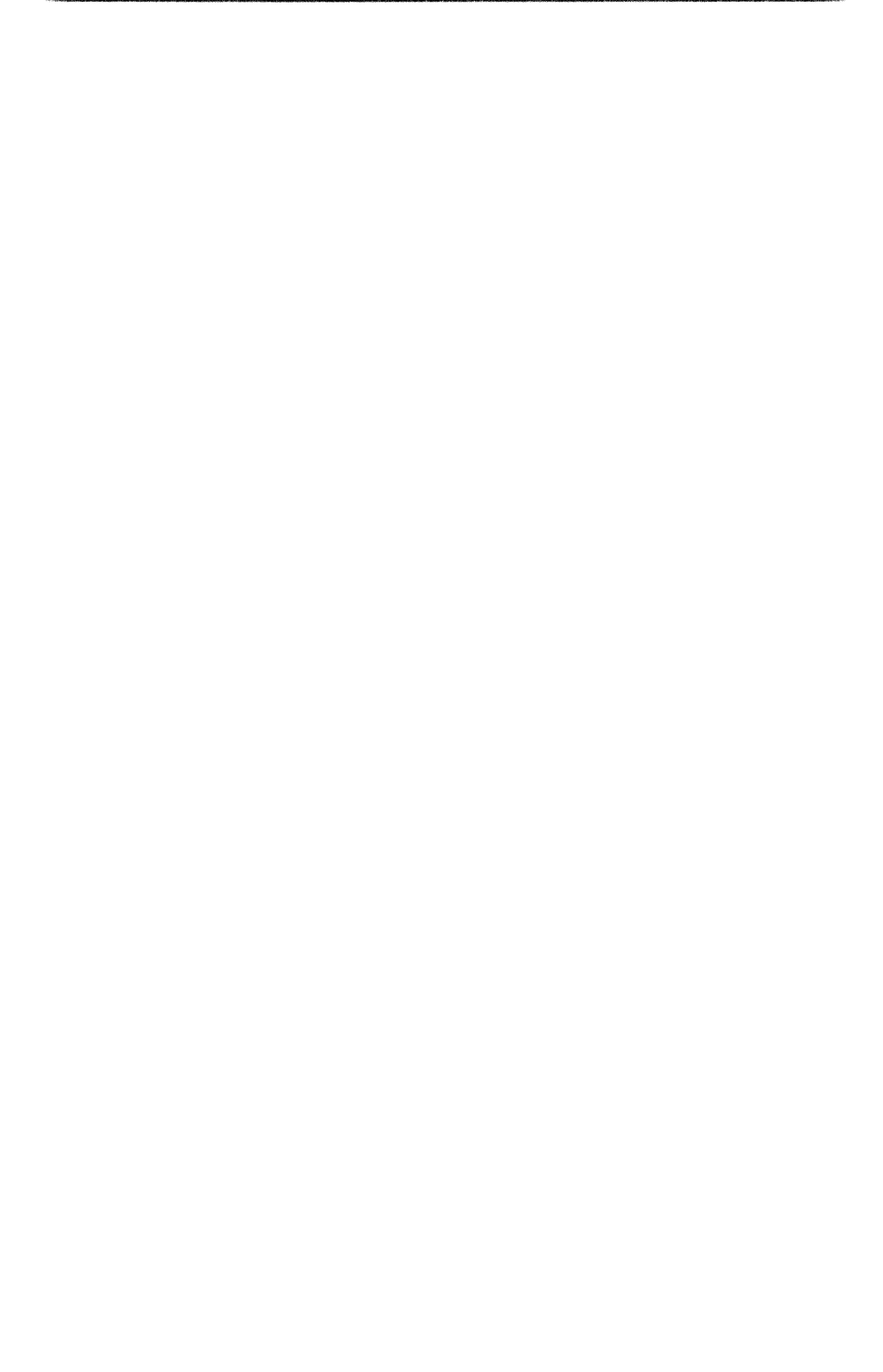


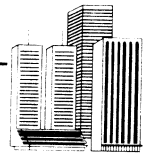
INTEREST RATE

```
100 PRINT CHR$(147) CHR$(144)
110 POKE 36879,25
200 REM *****
210 REM * ENTER PRINCIPAL & INTEREST *
220 REM *****
230 PRINT
240 PRINT
250 INPUT "INIT. INVESTMENT:";PRINCIPAL
260 PRINT
270 PRINT
280 INPUT "INTEREST RATE:";RATE
290 RATE=RATE/100
300 PRINT
310 INPUT "HOW MANY YEARS:";YEARS
320 PRINT
330 PRINT "WHICH INTERVAL"
340 PRINT
350 PRINT "<1> YEARLY"
360 PRINT "<2> QUARTERLY"
370 PRINT "<3> MONTHLY"
380 PRINT
390 PRINT "CHOOSE BY NUMBER"
400 GET A$: IF A$="" THEN 400
410 A=VAL(A$)
420 ON A GOSUB 500,600,700
430 PRINT
440 PRINT "AFTER" YEARS "YEARS."
450 PRINT "YOU WILL HAVE $" PRINCIPAL
460 PRINT
470 INPUT "ANOTHER(Y/N)";AN$
480 IF AN$="Y" THEN RUN
490 END
500 REM *****
510 REM *** YEARLY COMPOUND ***
520 REM *****
530 FOR X=1 TO YEARS
```



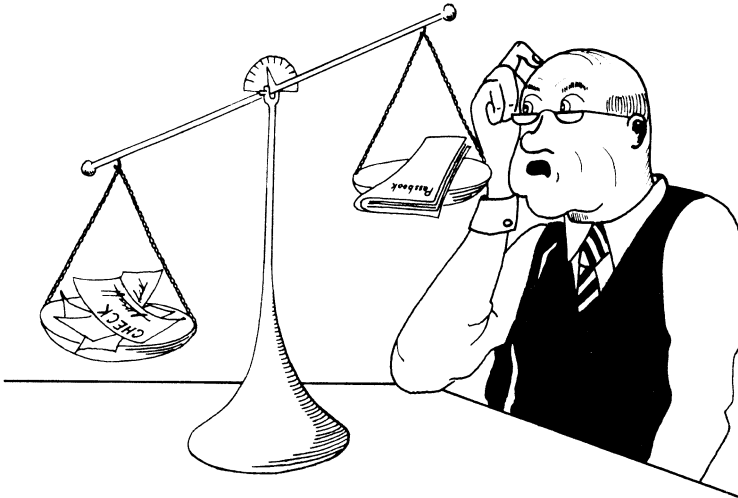
```
540 SUM=PRINCIPAL*RATE
550 PRINCIPAL=PRINCIPAL+SUM
560 NEXT
570 RETURN
600 REM *****
610 REM *** QUARTERLY COMPOUND ***
620 REM *****
630 RATE=RATE/4
640 FOR X=1 TO YEARS*4
650 SUM=PRINCIPAL*RATE
660 PRINCIPAL=PRINCIPAL+SUM
670 NEXT
680 RETURN
700 REM *****
710 REM *** MONTHLY COMPOUND ***
720 REM *****
730 RATE=RATE/12
740 FOR X=1 TO YEARS*12
750 SUM=PRINCIPAL*RATE
760 PRINCIPAL=PRINCIPAL+SUM
770 NEXT
780 RETURN
```





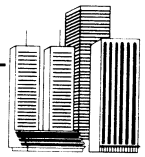
Checkbook

Balancing a checkbook can be a tedious task, and there is always an element of human error. With a computer, that task becomes more accurate and less time-consuming. Enter a 1 to input the starting balance. Then, using menu options 2-4, enter in all of your checks, deposits and adjustments. Press RETURN if you have no more to enter. Then enter a 5 when you would like your ending balance calculated. Once this is done, enter a RETURN to restart the program, or any other key to continue where you left off.



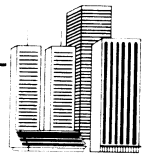
CHECKBOOK

```
100 PRINT CHR$(147) CHR$(144)
110 POKE 36879,25
200 REM *****
210 REM * PRINT SCREEN *
220 REM *****
230 PRINT CHR$(147) TAB(6) "CHECK BOOK"
240 PRINT : PRINT
250 PRINT "WHICH:"
260 PRINT
270 PRINT "1> START BAL.:" BALANCE
280 PRINT "2> ENTER CHECKS"
290 PRINT "3> ENTER DEPOSITS"
300 PRINT "4> ENTER ADJUSTMENTS"
310 PRINT "5> CALC. END BAL."
320 GET R$ : IF R$="" THEN 320
330 R=VAL(R$)
340 IF R<1 OR R>5 THEN 320
350 ON R GOTO 430,530,630,730,830
400 REM *****
410 REM * ENTER BALANCE *
420 REM *****
430 PRINT CHR$(147) "ENTER BALANCE:";
440 INPUT BALANCE
450 GOTO 230
500 REM *****
510 REM * ENTER CHECKS *
520 REM *****
530 PRINT CHR$(147)
540 PRINT
550 INPUT "CHECK AMT.:";CHECK
560 IF CHECK=0 THEN 230
570 EBAL=EBAL-CHECK
580 CHECK=0
590 GOTO 540
600 REM *****
610 REM * ENTER DEPOSITS *
620 REM *****
```

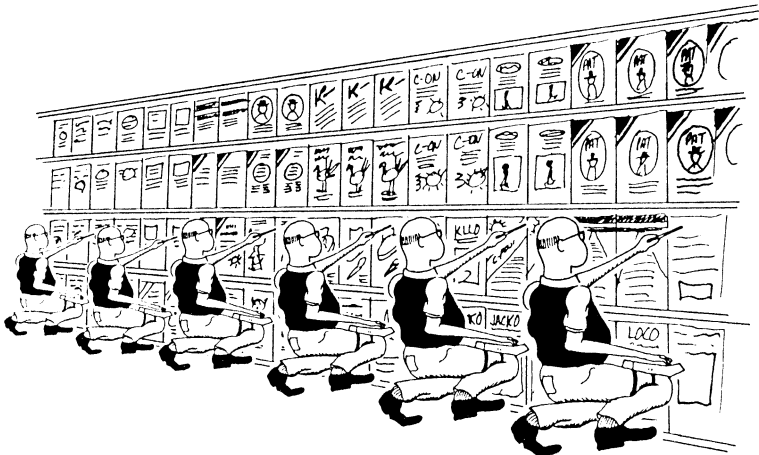
```
630 PRINT CHR$(147)
640 PRINT
650 INPUT "DEPOSIT AMT.:";DEP
660 IF DEP=0 THEN 230
670 EBAL=EBAL+DEP
680 DEP=0
690 GOTO 640
700 REM *****
710 REM * ENTER ADJUSTMENT *
720 REM *****
730 PRINT CHR$(147)
740 PRINT
750 INPUT "ADJ. AMT.:";ADJ
760 IF ADJ=0 THEN 230
770 EBAL=EBAL+ADJ
780 ADJ=0
790 GOTO 740
800 REM *****
810 REM * CALCULATE BALANCE *
820 REM *****
830 PRINT CHR$(147) "ENDING BAL.=";
840 EC=BALANCE+EBAL
850 PRINT EC
860 PRINT ; PRINT
870 PRINT "HIT RETURN TO START OVER, ANY OTH
ER TO CONTINUE"
880 GET R$ ; IF R$="" THEN 880
890 IF R$=CHR$(13) THEN RUN
900 GOTO 230
```





Inventory

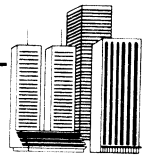
What is a business section without an inventory program? That's exactly what I thought to myself, so here is the program. This inventory program will store an almost unlimited number of items in a sequential disk file, and the quantities of each in stock. To start the program, enter a 5 to initialize the inventory file. Enter a Y for Yes (to initialize the file), and when the computer is done, you are ready to start entering items. Back in the main menu, type a 1 to enter items. The computer will prompt you with ITEM:. For example, enter the item, CARBURETORS. Now the computer will ask you for a quantity; enter the number of carburetors you have in stock. After the computer is through storing the information to disk, it asks you for another ITEM. If that's all you want to do, then press RETURN. When an X is shown and you hit RETURN, you will be put back into the main menu. Now you are ready to view your entries. Enter a 4, and the computer will ask you if you would like to direct output to the <P>RINTER or to the <S>CREEN.



Enter an S, and the computer will display all the entries you have made so far. After a while, your supplies might get depleted or altogether eliminated. If you wish to update a quantity, enter a 3. You will then be asked which item you wish to update. Enter CARBURETORS. Now you will be asked for a new quantity. Enter whatever number you wish. When you hit RETURN, the computer will go and update your entry for you, showing you the file's new status. If you would like to delete an entry altogether, enter a 2. Now enter CARBURETORS. After this is done, your entry will be erased from the file forever. To exit Inventory, enter a 6 at the main menu.

INVENTORY

```
100 PRINT CHR$(147) CHR$(144)
110 POKE 36879,25
120 OPEN 1,0 : BK#=CHR$(157)
130 C#=CHR$(13)
200 PRINT CHR$(147) TAB(6) "INVENTORY"
210 PRINT : PRINT
220 PRINT "MENU:" : PRINT
230 PRINT "<1> ENTER ITEM"
240 PRINT "<2> DELETE ITEM"
250 PRINT "<3> UPDATE QUANTITY"
260 PRINT "<4> DUMP INVENTORY"
270 PRINT "<5> INITIALIZE FILE"
280 PRINT "<6> QUIT"
290 PRINT CHR$(19) : PRINT : PRINT
300 PRINT TAB(5);
310 INPUT#1,A#
320 A=VAL(A#)
330 IF A<1 OR A>6 THEN RUN
340 ON A GOTO 530,830,1130,1630,1930
400 REM *****
410 REM * QUIT *
420 REM *****
430 PRINT CHR$(147) "BYE BYE"
440 PRINT : PRINT : END
```

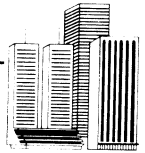


```
500 REM *****
510 REM * ENTER ITEM *
520 REM *****
530 PRINT CHR$(147) TAB(4) "ENTER ITEMS"
540 PRINT : PRINT
550 PRINT "ITEM:X" BK$;
560 INPUT#1,ITEM$
570 IF ITEM$="X" THEN RUN
580 PRINT
590 PRINT "QUANTITY:X" BK$;
600 INPUT#1,QUANT$
610 IF QUANT$="X" THEN RUN
620 OPEN 4,8,4,"0:INVENTORY FILE,S,R"
630 OPEN 5,8,5,"0:COPY,S,W"
640 INPUT#4,I$,Q$
650 IF I$="EOF" THEN 710
660 PRINT#5,I$C$Q$C$
670 GOTO 640
680 REM *****
690 REM * END OF FILE *
700 REM *****
710 PRINT#5,ITEM$C$QUANT$C$
720 GOSUB 1530
730 GOTO 540
800 REM *****
810 REM * DELETE ITEM *
820 REM *****
830 PRINT CHR$(147) TAB(4) "DELETE ITEM"
840 PRINT : PRINT : F=0
850 PRINT "ITEM TO DELETE:X" BK$;
860 INPUT#1,ITEM$
870 IF ITEM$="X" THEN RUN
880 OPEN 4,8,4,"0:INVENTORY FILE,S,R"
890 OPEN 5,8,5,"0:COPY,S,W"
900 INPUT#4,I$,Q$
910 IF I$="EOF" THEN 980
920 IF I$=ITEM$ THEN QUANT$=Q$ : F=1 : GOTO
900
```

```

930 PRINT#5,I#C#Q#
940 GOTO 900
950 REM *****
960 REM * END OF FILE *
970 REM *****
980 GOSUB 1530
990 PRINT : PRINT
1000 IF F=0 THEN PRINT "ENTRY NOT FOUND!" :
GOTO 1030
1010 PRINT "ITEM:" ITEM$,"QUANTITY:" QUANT#
1020 PRINT CHR$(18) "DELETED"
1030 IF PEEK(197)=64 THEN 1030
1040 RUN
1100 REM *****
1110 REM * UPDATE QTY. *
1120 REM *****
1130 PRINT CHR$(147) TAB(2) "UPDATE QUANTITY
"
1140 PRINT : PRINT
1150 PRINT "ITEM TO UPDATE:X" BK#;
1160 INPUT#1,ITEM#
1170 IF ITEM#="X" THEN 200
1180 PRINT : PRINT "NEW QUANTITY:X" BK#;
1190 INPUT#1,QUANT#
1200 IF QUANT#="X" THEN 200
1210 F=0
1220 OPEN 4,8,4,"0:INVENTORY FILE,S,R"
1230 OPEN 5,8,5,"0:COPY,S,W"
1240 INPUT#4,I#,Q#
1250 IF I#="EOF" THEN 1400
1260 IF I#=ITEM# THEN 1320
1270 PRINT#5,I#C#Q#
1280 GOTO 1240
1290 REM *****
1300 REM * ENTRY FOUND *
1310 REM *****
1320 PRINT : PRINT : F=1
1330 PRINT "OLD QUANTITY :" Q#

```



```
1340 PRINT : Q#=QUANT#
1350 PRINT "NEW QUANTITY : " Q#
1360 GOTO 1270
1370 REM *****
1380 REM * END OF FILE *
1390 REM *****
1400 PRINT : PRINT
1410 IF F=0 THEN PRINT "ITEM NOT FOUND!"
1420 GOSUB 1530
1430 RUN
1500 REM *****
1510 REM * CLOSE FILES *
1520 REM *****
1530 PRINT#5,"EOF" C# "EOF"
1540 CLOSE 4 : CLOSE 5
1550 OPEN 15,8,15
1560 PRINT#15,"S0:INVENTORY FILE"
1570 PRINT#15,"R0:INVENTORY FILE=COPY"
1580 CLOSE 15
1590 RETURN
1600 REM *****
1610 REM * DUMP FILE *
1620 REM *****
1630 PRINT CHR$(147) TAB(5) "DUMP FILE"
1640 PRINT : PRINT
1650 PRINT "<P>RINTER/<S>CREEN:S" BK#;
1660 INPUT#1,Q#
1670 IF Q#="X" THEN 200
1680 IF Q#="S" THEN 1710
1690 OPEN 2,4
1700 PRINT#2,C#,TAB(10) "INVENTORY FILE" C#
1710 OPEN 4,8,4,"0:INVENTORY FILE,S,R"
1720 PRINT CHR$(147)
1730 INPUT#4,I#,Q#
1740 IF I#="EOF" THEN 1840
1750 IF Q#="S" THEN 1780
1760 PRINT#2,"ITEM:" I#
1770 PRINT#2,"QUANTITY:" Q# C#
```

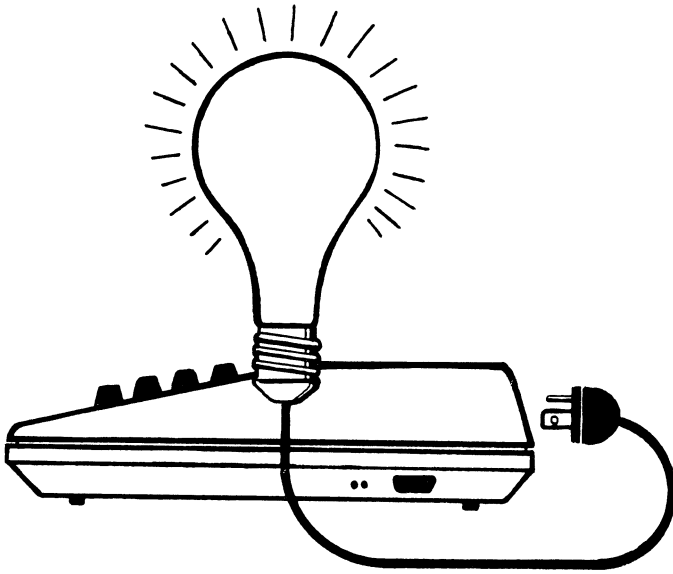
```
1780 PRINT "ITEM:" I$
1790 PRINT "QUANTITY:" Q$ C$
1800 GOTO 1730
1810 REM *****
1820 REM * END OF FILE *
1830 REM *****
1840 CLOSE 4 : CLOSE 2
1850 PRINT : PRINT "END OF FILE"
1860 IF PEEK(197)=64 THEN 1860
1870 RUN
1900 REM *****
1910 REM * INITIALIZE FILE *
1920 REM *****
1930 PRINT CHR$(147) TAB(2) "INITIALIZE FILE
"
1940 PRINT : PRINT
1950 PRINT "ARE YOU SURE(Y/N):N" BK$;
1960 INPUT#1,A$
1970 IF A$("<"Y" THEN 200
1980 OPEN 15,0,15
1990 PRINT#15,"S0:INVENTORY FILE"
2000 CLOSE 15
2010 OPEN 5,0,5,"0:INVENTORY FILE,S,W"
2020 PRINT#5,"EOF" C$ "EOF"
2030 CLOSE 5
2040 RUN
```


Section 5

Utilities and Programming Aids

- Character Creator
- Sequential File Reader
- Menu Ideas
- Alarm Clock
- File Restorer





UTILITIES AND PROGRAMMING AIDS

Since this is the Utility and Programming Aids section of the book, I thought it would be a good idea to include some machine language subroutines. I chose machine language for several reasons. First, it is compact. Second, machine language programs are less likely to “bomb” (malfunction) than are BASIC ones. Another reason, and perhaps the most important, is that machine language programs are far faster in accomplishing a given task than are BASIC programs.

The programs in this section are shown in two formats. The first is the assembler format (this is the easy-to-read version). These types of listings are done with what is called an assembler, a very useful machine language tool. For the programs presented here, I used the MERLIN assembler by Southwestern Data Systems. The other format is that of the DATA statements, which can be read and POKEd into memory by BASIC. All of these routines can be put into memory anywhere without modification, except for the LOAD and SAVE routines which must be altered. The MERGE routine is particularly useful with the hi-res draw program. Some possible locations of where to put these routines are:

HEX	DEC
\$8000-\$9FFF	32768-40959
\$C000-\$CFFF	49152-53247

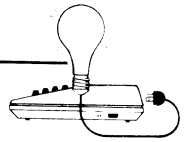
These are the hexadecimal ranges and their decimal equivalents. Remember, you use decimal from BASIC!

Let's see an example of how to use one of these subroutines. Here is the BASIC data statement for a fictitious subroutine:

```
DATA 96,60,53,255,32,44,96
```

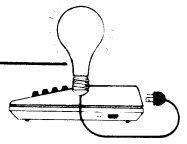
Let's say we want to start it at location 49152 in memory. Here is how you would use it from a program:

```
10 FOR T=49152 TO 49152+L
20 REM L IS LENGTH OF SUBROUTINE-1
30 REM IN OUR CASE IT'S 7
40 READ P
50 POKE T,P
60 NEXT
70 SYS 49152
80 DATA 96,60,53,255,32,44,96
```



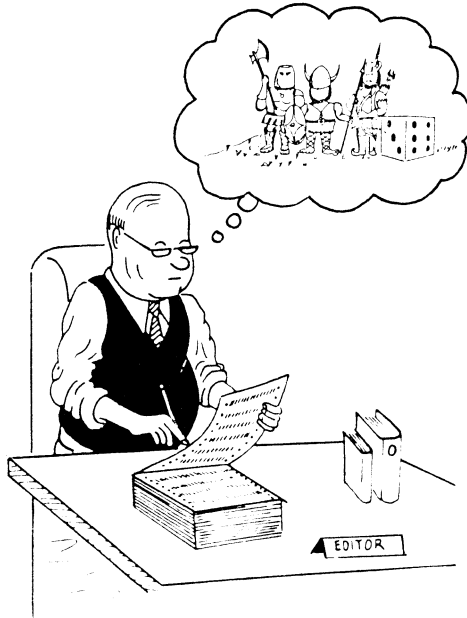
That's all there is to it. The SYS command will execute the machine language program at the specified address (in our example, location 49152). The programs and subroutines in this section enhance any BASIC or machine language program, and they will make certain programming problems a lot simpler. Also, they represent the first step into more advanced applications.





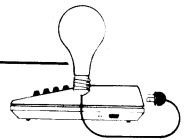
Character Creator

Although designing programmable characters is easier to do than designing sprites, it is still time-consuming. I found it ten times easier to have a character editor. This one is extremely easy to use. Your work space is an 8×8 area. You move a crosshair around on the screen, and use the space bar to set the various bits of the character. To control the crosshair's movement going up, down, left and right, type I, K, J and L respectively, with the space bar plotting dots. The function keys also have particular purposes: pressing F1 once will display your newly designed programmable character; pressing F1 again will display the character's DATA statements; and pressing F3 will clear the screen so you can start again.



CHARACTER CREATER

```
100 PRINT CHR$(147) CHR$(144)
110 POKE 36979,25
120 POKE 52,28
130 POKE 56,28
140 CLR
200 REM *****
210 REM * PRINT SCREEN *
220 REM *****
230 PRINT CHR$(19) TAB(2);
240 PRINT "CHARACTER CREATE"
250 PRINT CHR$(32);
260 FOR T=1 TO 8
270 PRINT CHR$(164);
280 NEXT : PRINT
290 FOR T=1 TO 8
300 PRINT CHR$(167) TAB(9) CHR$(165)
310 NEXT
320 PRINT CHR$(32);
330 FOR T=1 TO 8
340 PRINT CHR$(163);
350 NEXT
360 PRINT
370 PRINT CHR$(19)
380 PRINT
390 RESTORE
400 FOR T=0 TO 6
410 READ T#
420 PRINT TAB(10) T#
430 NEXT
500 REM *****
510 REM * INITIALIZE POSITION *
520 REM *****
530 LOC=7725
540 X=0
550 Y=0
560 GOSUB 1030
```

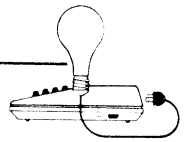



```
600 REM *****
610 REM * GET KEYPRESS *
620 REM *****
630 POKE 197,64 : R=PEEK(197)
640 IF R=64 THEN 630
650 IF SS=1 THEN 670
660 POKE LOC,32
670 IF R=20 THEN LOC=LOC-1 : X=X-1
680 IF X= 1 THEN X=7 : LOC=LOC+8
690 IF R=21 THEN LOC=LOC+1 : X=X+1
700 IF X=8 THEN X=0 : LOC=LOC-8
710 IF R=44 THEN LOC=LOC+22 : Y=Y+1
720 IF LOC>7886 THEN LOC=7725+X : Y=0
730 IF R=12 THEN LOC=LOC-22 : Y=Y-1
740 IF LOC<7725 THEN LOC=7879+X : Y=20
750 IF R=47 THEN RUN
760 IF R=32 THEN 930
770 IF R=39 THEN 1130
780 SS=0
790 GOSUB 1030
800 POKE 197,64
910 POKE 198,0
820 GOTO 630
900 REM *****
910 REM * PUT A DOT ON THE SCREEN *
920 REM *****
930 POKE LOC,160
940 POKE LOC+30720,6
950 POKE 198,0
960 SS=1
970 GOTO 630
1000 REM *****
1010 REM * CROSS ON SCREEN *
1020 REM *****
1030 POKE LOC,91
1040 POKE LOC+30720,0
1050 RETURN
```

```

1100 REM *****
1110 REM * SHOW CHARACTER *
1120 REM *****
1130 FOR Y=0 TO 7
1140 BYTE(Y)=0
1150 FOR X=0 TO 7
1160 IF PEEK(7725+22*X+Y)<>160 THEN 1180
1170 BYTE(Y)=BYTE(Y)+2*(ABS(7-X))
1180 NEXT
1190 NEXT
1200 FOR Y=0 TO 7
1210 POKE 7169+Y,BYTE(Y)
1220 NEXT
1230 PRINT CHR$(147)
1240 POKE 36869,255
1250 PRINT CHR$(147) "@"
1260 POKE 178,0
1270 GET R$: IF R$="" THEN 1270
1280 POKE 36869,240
1290 PRINT CHR$(147) "CHARACTER DATA"
1300 PRINT
1310 FOR I=0 TO 7
1320 PRINT BYTE(I)
1330 NEXT
1340 GET A$: IF A$="" THEN 1340
1350 RUN
1400 REM *****
1410 REM * COMMAND STRINGS *
1420 REM *****
1430 DATA I=UP,R=DOWN,J=LEFT,L=RIGHT
1440 DATA SPC=GET PT,F1=SHW CHR
1450 DATA F3=CLR SCRN

```



Sequential File Reader

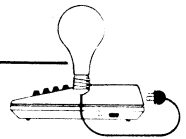
Every once in a while, it seems that I have to check out what's in a SEQ file, whether it's to verify its contents or just out of plain curiosity. That's where this program started. Given a file name, File Reader will go through and dump the contents of the file to the screen or to a printer, depending on which you specify at the beginning of the program.

As you become more involved with communications on the VIC-20, you will find that the contents of SEQential files are very important. This is because most computer-to-computer transmissions use sequential files rather than program files.

```
FILE READER
```

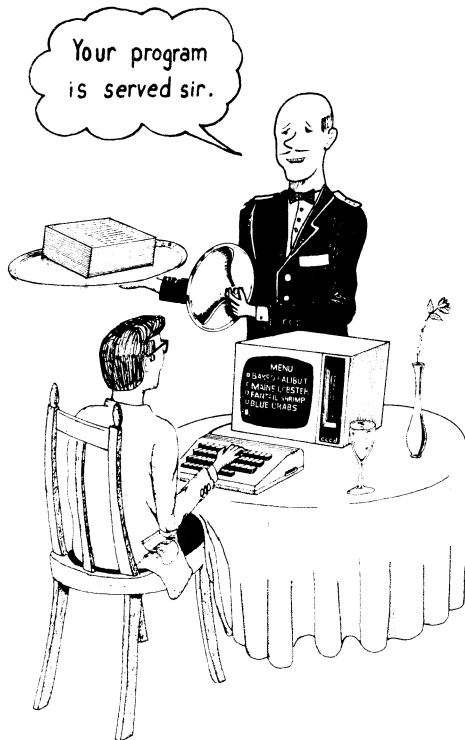
```
100 PRINT CHR$(147) CHR$(144)
110 POKE 36879,25
120 OPEN 2,0
200 REM *****
210 REM * GET FILENAME *
220 REM *****
230 PRINT CHR$(147) "FILE TO READ:";
240 INPUT#2,FILE#
250 PRINT
260 PRINT "<P>RINTER/<S>CREEN:";
270 INPUT#2,0#
280 IF 0#<>"P" AND 0#<>"S" THEN END
290 IF 0#="S" THEN 430
300 OPEN 1,4
310 CMD 1
400 REM *****
410 REM * GET DATA *
420 REM *****
430 OPEN 4,0,4,"0:"+FILE#+",S,R"
440 PRINT
450 PRINT
```

```
460 GET#4,A#  
470 PRINT A#;  
480 IF ST=0 THEN 460  
490 IF O#="P" THEN PRINT#1 : CLOSE 1  
500 CLOSE 4  
510 PRINT  
520 PRINT "END OF FILE"
```



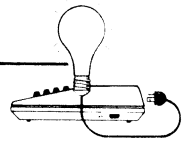
Menu Ideas

Menus are an extremely important part of any program. An attractive menu could mean the difference between whether a program is successful or not. An easy-to-read, easy-to-understand menu is the best way to start off any program. Menu Ideas gives a few pointers on the proper way to make a menu. Keys A and Z move the marker up and down next to your list of choices. Pressing the space bar will choose the item that you are currently next to.



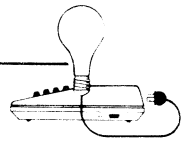
MENU IDEAS

```
100 PRINT CHR$(147)
110 CC=0
120 SC=25
130 POKE 36879,SC
140 POKE 646,CC
200 REM *****
210 REM * PRINT MENU *
220 REM *****
230 PRINT CHR$(147) TAB(5) "MENU SELECT"
240 PRINT
250 PRINT
260 PRINT
270 PRINT CHR$(10);
280 PRINT "SPC=SELECT/A=UP/Z=DOWN"
290 PRINT
300 PRINT " SCREEN COLOR"
310 PRINT " CHARACTER COLOR"
320 PRINT " QUIT"
330 POKE 7680+1+22*(C+7),62
340 POKE 38400+1+22*(C+7),6
350 GET R$: IF R$="" THEN 350
360 POKE 7680+1+22*(C+7),32
370 IF R$="Z" AND C<2 THEN C=C+1
380 IF R$="A" AND C>0 THEN C=C-1
390 IF R$=" " THEN 410
400 GOTO 330
410 ON C+1 GOTO 530,630
420 PRINT
430 PRINT
440 PRINT
450 END
500 REM *****
510 REM * SCREEN COLOR *
520 REM *****
530 SC=SC+1
540 IF SC>255 THEN SC=0
```



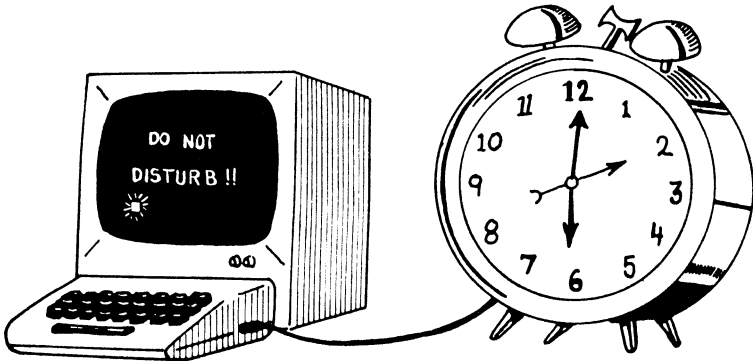
```
550 POKE 36879,5C
560 GOTO 330
600 REM *****
610 REM * CHARACTER COLOR *
620 REM *****
630 CC=CC+1
640 IF CC>7 THEN CC=0
650 POKE 646,CC
660 GOTO 230
```





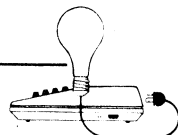
Alarm Clock

One of the many nice things about the Commodore 64 and VIC-20 computers is that they both have built-in clocks. Both are very accurate. These come in handy for games, telecommunications, and just about any other application where time-keeping is needed. I thought it would be neat to have a functional alarm clock: a program where you enter both the current time, and the time that you want the alarm to go off. This can be handy if you don't have an alarm watch or clock with such a feature. It's also just plain fun. When the current time is equal to when you set the alarm to go off, color and sounds abound! To turn the alarm status on and off, enter an O and an F accordingly. Once the alarm has gone off, pressing F1 will turn it off. One warning though; keep the volume down so you won't disturb the neighbors!



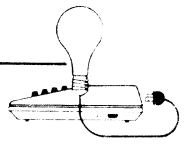
ALARM CLOCK

```
100 PRINT CHR$(147) CHR$(144)
110 POKE 36879,25
200 REM *****
210 REM * GET CLOCK TIME *
220 REM *****
230 INPUT "THE TIME IS: ";TM$
240 IF LEN(TM$)<>6 THEN 230
250 TI$=TM$
260 PRINT
270 PRINT
300 REM *****
310 REM * GET ALARM TIME *
320 REM *****
330 INPUT "ALARM TIME: ";AT$
340 IF LEN(AT$)<>4 THEN 330
400 REM *****
410 REM * PRINT CLOCK *
420 REM *****
430 PRINT CHR$(147) "ALARM OFF"
440 FOR T=1 TO 7
450 PRINT
460 NEXT
470 PRINT TAB(5) CHR$(117);
480 FOR T=1 TO 10
490 PRINT CHR$(96);
500 NEXT
510 PRINT CHR$(105)
520 PRINT TAB(5) CHR$(125) CHR$(176);
530 FOR T=1 TO 8
540 PRINT CHR$(96);
550 NEXT
560 PRINT CHR$(174) CHR$(125)
570 PRINT TAB(5) CHR$(125) CHR$(125) SPC(8)
CHR$(125) CHR$(125)
580 PRINT TAB(5) CHR$(125) CHR$(173);
590 FOR T=1 TO 9
```



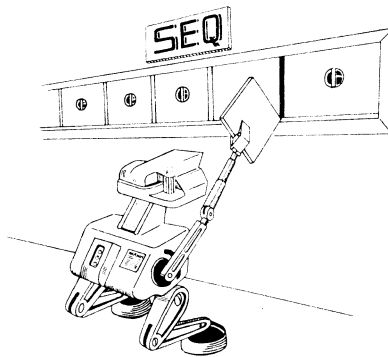
```
600 PRINT CHR$(96);
610 NEXT
620 PRINT CHR$(109) CHR$(125)
630 PRINT TAB(5) CHR$(173);
640 FOR T=1 TO 10
650 PRINT CHR$(96);
660 NEXT
670 PRINT CHR$(109)
700 REM *****
710 REM * MAIN ROUTINE *
720 REM *****
730 GET R$
740 IF R$="A" THEN PRINT CHR$(147) : GOTO 330
750 IF R$="F" THEN PRINT CHR$(19) "ALARM OFF"
   : AS=0
760 IF R$="D" THEN PRINT CHR$(19) "ALARM ON"
   : AS=1
770 GOSUB 1130
780 TT#=LEFT$(TI#.4)
790 IF VAL(TT#)=VAL(AT#) AND AS=1 THEN 930
800 GOTO 730
900 REM *****
910 REM * ALARM ON *
920 REM *****
930 POKE 36878,15
940 POKE 36876,245
950 FOR T=1 TO 200
960 NEXT
970 POKE 36876,0
980 GOSUB 1130
990 GET A# : IF A#(<>)CHR$(133) THEN 940
1000 PRINT CHR$(19) "ALARM OFF"
1010 AS=0
1020 GOTO 730
1100 REM *****
1110 REM * UPDATE TIME *
1120 REM *****
```

```
1130 PRINT CHR$(19)
1140 FOR T=1 TO 9
1150 PRINT
1160 NEXT
1170 HO$=LEFT$(TI$,2)
1180 MI$=MID$(TI$,3,2)
1190 SE$=RIGHT$(TI$,2)
1200 PRINT TAB(7) HO$ ":" MI$ ":" SE$
1210 RETURN
```



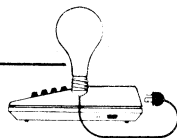
File Restorer

Here it is, the VIC-20 program of the century! How many times have you deleted a file from disk and then said, "It's okay. I know what I'm doing." How many times have you been wrong? I thought so. Don't worry, it happens to the best of us. I've made that mistake plenty of times, accidentally erasing files from my diskette. Don't ask me how, either. Needless to say, I didn't make a back-up. This is why File Restorer was created. Now I'm so paranoid about disk crashes that I make three copies of anything I treasure. Using File Restorer is easy. When the program is run, you will be asked for the name of the file you wish to restore. Next, it will ask you for the type of file to be restored (PRG, SEQ, etc.). The last two questions pertain to the drive and device number of the unit being used. When all the requested data has been entered, the drive will turn on and search the directory for your entry. If it is not found, a FILE NOT FOUND message will be printed. Otherwise, you will see FILE RESTORED up on the screen. You now can load and re-use that little gem of a program. Once you've scratched a file that you wanted to keep, don't save anything else on that diskette, as it is likely to write over precious program data. Instead, run this program immediately. Then, once you've restored the file and have it in memory, scratch it and then save it again. I hope that this is one of your least used programs.



FILE RESTORER

```
100 PRINT CHR$(147) CHR$(144)
110 POKE 36979,25
120 BK#=CHR$(157)
130 NAME$=""
200 REM *****
210 REM * QUESTIONS *
220 REM *****
230 PRINT CHR$(147)
240 NAME$=""
250 INPUT "RESTORE FILE: ";NAME$
260 IF NAME$="" THEN END
270 IF LEN(NAME$)>16 THEN 230
280 FOR I=1 TO 16 LEN(NAME$)
290 NAME$=NAME$+CHR$(160)
300 NEXT
310 NAME$=LEFT$(NAME$,16)
320 PRINT
330 PRINT
340 PRINT "(1) SEQ"
350 PRINT "(2) PROB"
360 PRINT "(3) USER"
370 PRINT "(4) REL"
380 PRINT
390 PRINT "WHICH TYPE: 2";
400 PRINT BK# BK# BK#;
410 INPUT TYP#
420 TYP=VAL(TYP#)
430 IF TYP>0 AND TYP<5 THEN 460
440 PRINT
450 END
460 TYP=TYP+129
470 PRINT
480 PRINT "DISK DEVICE#: 0";
490 PRINT BK# BK# BK#;
500 INPUT DEV#
510 DEV=VAL(DEV#)
520 IF DEV>0 AND DEV<9 THEN 320
530 PRINT
```



```
540 PRINT "DRIVE#: 0";
550 PRINT BK$ BK$ BK$;
560 INPUT DRV$
570 DRV=VAL(DRV$)
580 IF DRV<>0 AND DRV<>1 THEN 530
590 REM *****
610 REM X CHANGE DIRECTORY *
620 REM *****
630 PRINT
640 PRINT "READING THE DIRECTORY"
650 PRINT
660 OPEN 15,DEV,15
670 OPEN 5,DEV,5,"#"
680 SEC=1
690 PRINT#15,"U1:";5;DRV;18;SEC
700 X=1
710 FOR T=5 TO 229 STEP 32
720 PRINT#15,"B-P:";5;T
730 FILE$(X)=" "
740 FOR S=1 TO 16
750 GET#5,A$
760 FILE$(X)=FILE$(X)+A$
770 NEXT
780 IF FILE$(X)=NAME$ THEN 930
790 X=X+1
800 NEXT
810 PRINT#15,"B-P:";5;1
820 GET#5,SEC$
830 SEC=ASC(SEC$)
840 IF SEC<19 THEN 690
850 PRINT
860 PRINT "FILE NOT FOUND!"
870 PRINT
880 END
900 REM *****
910 REM *** NAME FOUND ***
920 REM *****
```

```
930 BYTE=T-3
940 T=250
950 NEXT
960 PRINT#15,"D-P: ";5;BYTE
970 PRINT#5,CHR$(TYP);
980 PRINT#15,"U2: ";5;DRV:10;SEC
990 CLOSE 15
1000 CLOSE 5
1010 PRINT
1020 PRINT "IT'S RECDVERED!!"
1030 END
```


From the author of

The VIC-20 *Experience*

The VIC-20 Experience on disk!

For only \$15.95 with this coupon, you can have all the programs contained within this book on one diskette! Save yourself hours of typing and de-bugging.

All the programs contained on the diskette are listable, copyable and contain REM statements which identify each part of the program. You can list and save sections of the programs for use in your own programs.

The VIC-20 Experience diskette helps you get the most out of both your VIC-20 Home Computer and The VIC-20 Experience.

Send \$15.95 plus \$2.00 shipping and handling charge to:



20660 Nordhoff Street, Chatsworth, CA 91311-6152

Name _____

Address _____

City _____ State _____ Zip _____

MASTERCARD

Check

VISA

Money Order

Credit Card # _____ Expires _____

Signature (for credit card) _____

California Residents add 6½% sales tax.

*VIC-20 is a trademark of Commodore Business Machines, Inc.



The
VIC-20
Experience

THE NEXT TIME SOMEONE ASKS, "WHAT CAN YOU REALLY DO WITH A VIC-20 HOME COMPUTER?" SHOW THEM THIS BOOK!

It's an important question. One that deserves an answer. After all, you have one of the most powerful tools in the universe at your service. But . . . where to begin?

THE VIC-20 EXPERIENCE will help you learn about your Commodore VIC-20. Take that important first step and learn how to program BASIC by keying in completed programs. What if you could care less about programming? Simply key in the useful programs as they appear and save them to disk or tape. Either way, you'll soon have a library of software that can be used with everything from daily chores to complex business applications!

Written in Commodore BASIC, each program follows a standardized format and is clearly documented. These easy-to-follow programs cover home, educational, entertainment, business, and utility applications. In addition, the programs can be modified to meet your particular needs.

What can you really do with a Commodore VIC-20? The answer lies within these pages.

ISBN 0-88190-296-9



 **DATAMOST**

20660 Nordhoff Street, Chatsworth, CA 91311-6152
(818) 709-1202