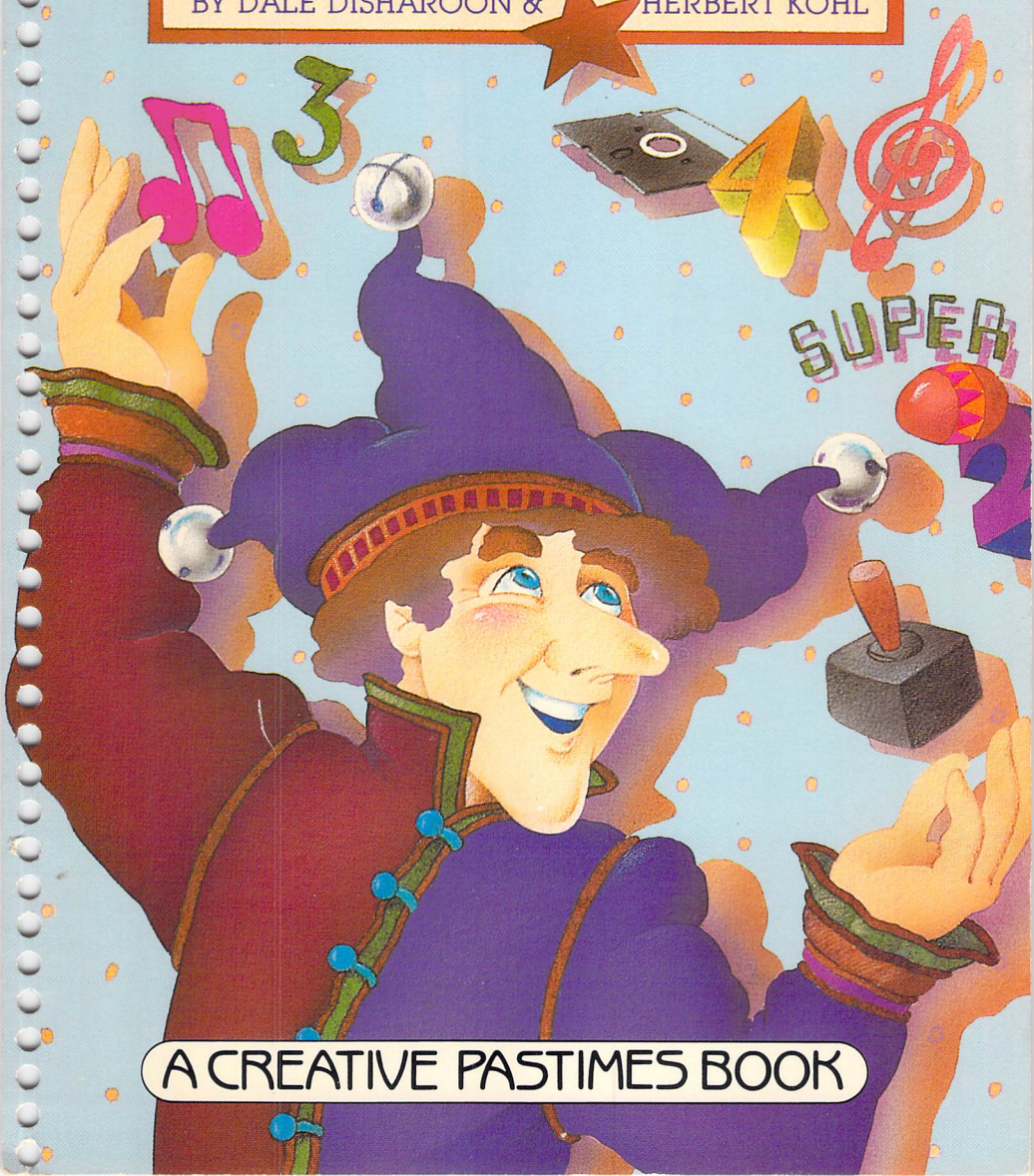
41½ FUN PROJECTS FOR YOUR COMMODORE®

BY DALE DISHAROON & HERBERT KOHL



A CREATIVE PASTIMES BOOK

This book belongs to

41½ Fun Projects for the Commodore 64®

DALE DISHAROON
and
HERBERT KOHL

Herbert Kohl, Series Adviser

A Creative Pastimes Book
Reston Publishing Company
A Prentice-Hall Company
Reston, Virginia

Library of Congress Cataloging in Publication Data
Disharoon, Dale.
41 1/2 fun projects for the Commodore 64.

"A Creative Pastimes book."
1. Computer games. 2. Commodore 64 (Computer)—Programming. 3. Basic (Computer program language) I. Kohl, Herbert R. II. Title.
GV1469.2.D58 1984 794.8'2 84-3260
ISBN 0-8359-2102-3

Interior design and production by Laura Cleveland.

10 9 8 7 6 5 4 3 2 1

Printed in the United States of America.

Contents

Introduction

Almost anything new and different can be exciting for a time. And almost all excitement wears off after a while. For many people, owning their own microcomputer is such an experience. At first there seem to be thousands of things to do to make life easier and more interesting. For many people, however, the magic fades and the computer becomes simply a checkbook balancer and game player if it doesn't actually end up in the closet. The intent of this book is to provide a new infusion of excitement about what you can do with your home computer.

With one exception, the programs in this book are quite short, easy to copy, and easy to modify. The text will suggest ways you can tailor the programs for your own use—for party games, data analysis, school work, or learning to program. The 41 1/2 things to do range all the way from a dating program that will match you with a potentially congenial date to a mini-word processor, a Dungeons and Dragons dice rolling program, an excuse generator, and strategy and music games. It includes practical programs such as a guitar tuner and chord teacher, a metric converter, and a comparison shopping program. It also contains party games and other activities that you can do with your friends. Underlying all of these programs are the ideas that you can modify a program and change it to suit your purposes and that programming can be fun and lead to fun as well.

The central part of the book provides you with a number of simple computer utilities. The most common utilities in our lives are gas, electricity, and the telephone. They are utilities in that they can be used in many different ways. They provide the energy or the instrument and you determine the particular use. The same thing is true of computer utilities. They provide a form that you can use in many different ways with a simple input of your own. Thus, the balloting program in this book can be used for club elections, picking the top ten records, doing personal preference polls, etc. The utility gives you the structure and you provide the content. The same is true of the graph generating program. It can be used to measure sales and inventories, to compare school grades, to illustrate the results of

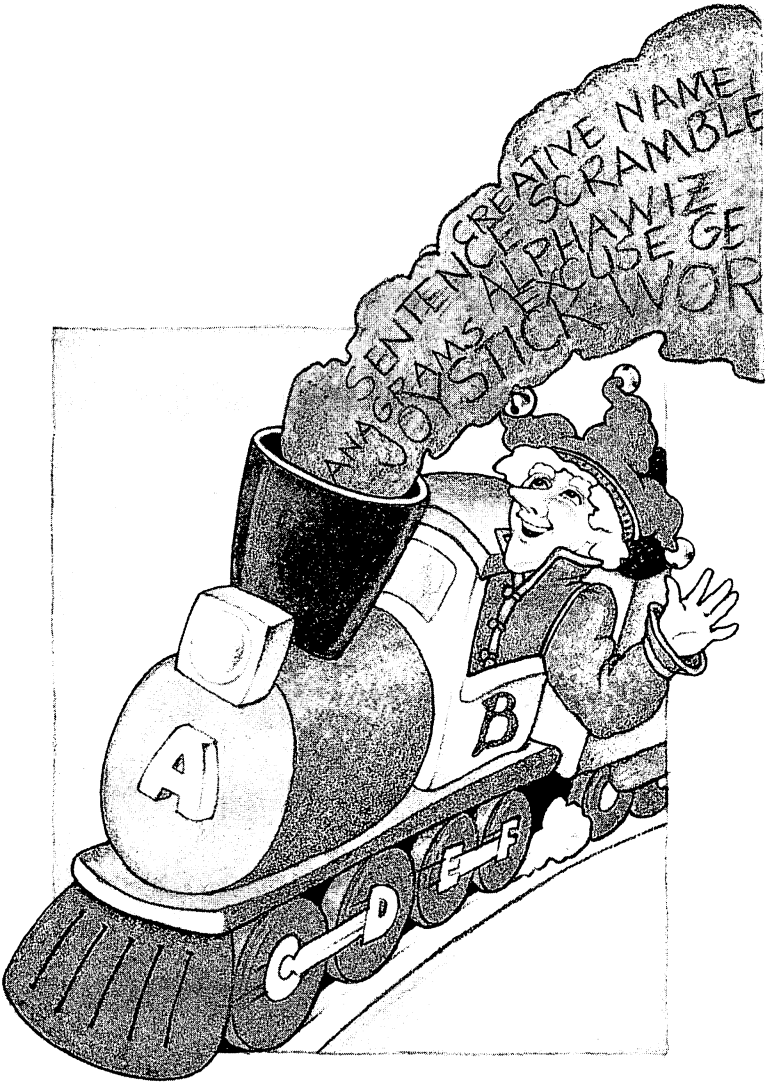
science experiments, or for opinion surveys. The text will suggest some applications for the utilities and show you how to list and change the programs in the book. You should feel free to go beyond the text and create your own modifications, and, maybe, build your own utilities.

All of the programs in this book fit on one disk. With the exception of the program for the Dozo game, they are easy to copy and take up less than two pages of text. It would make sense for you to copy and save on disk those programs you want to use or modify. Copying the same program more than once is a bore and you can always save your original copy as well as your modifications. After you copy a program, however, run it several times to make sure that some simple copying error didn't slip in. Try all the options the programs offer. Play to win, to deliberately lose, and try any way you can to trick the program into bombing. We've done this ourselves and are confident that the programs work. But, human error always has a way of creeping in. We have been careful to eliminate ours and believe it makes sense for you to take the extra time to check your work.

Finally, for the 1/2 of a thing you can do with your computer. Actually it is not one-half of a thing but many halves—many ideas that you can turn into programs as you become comfortable with BASIC. These final ideas are just starters, teasers, suggestions of ways in which you can focus the use of your programming skills on creating programs others will enjoy using.

PART I

Word Play



The programs in this section all fool around with words. They range from an anagram generator to an excuse generator and include word hunt puzzles, a sentence scrambler, and even an arcade-type game that will teach you how to use the keyboard. Many of the programs are easily modified and we encourage you to copy them and change them in any way you like.

1

Joystick Word Hunt

You probably have seen many word hunt puzzles. They consist of a square or rectangle of letters in which a number of words are hidden either vertically, horizontally, or diagonally, and backwards and forwards. Here is a very simple word hunt grid and the list of words hidden in it:

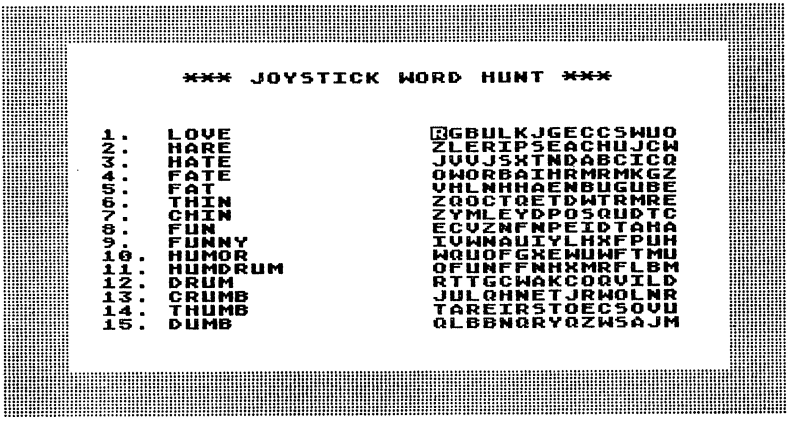
hidden words: last, cat, rat
curl, pot, coat

hidden word grid:

c o r l
o a a r
a s t u
t o p c

The object of the word hunt is to find the words on the list hidden in the grid.

The program for Joystick Word Hunt allows you to choose any 15 words of up to 15 letters long. Then, the computer automatically generates a word grid for you or your friend to solve. Here is a sample list and the grid the computer generated. You'll find that even if you chose the words yourself it is not easy to find them all in the grid.



You need a joystick in port 1 to play this game. Move the flashing cursor on the game board to the first letter of any word you find. Press the joystick button. A message will appear: WORD NUMBER. Type only the number that corresponds to the word you've found. Press **RETURN**. If you've found the correct match, the word will appear in inverse characters on the grid. Keep playing until you find all 15 original words.

If you want to play another game, press the [F1] function key on the right of your Commodore keyboard.

```
10 REM  *** JOYSTICK WORD HUNT ***
100 SO=54272:CR=49152:CY=CR+10:CX=CY+1:G
OSUB1000:POKE53280,0:POKE53281,0
110 POKEDSO+5,0:POKESO+6,240:PRINTCHR$(1
47)CHR$(5):GOTO700
200 GOSUB800:FORM=0TO14:L=LEN(W$(M)):XT=
2:YT=6+M:GOSUB900:PRINTM+1;CHR$(157);"."
210 XD=INT(RND(1)*3-1):YD=INT(RND(1)*3-1
):IF XD=0ANDYD=0THEN210
220 X=INT(RND(1)*(15-(ABS(XD)*(L-1)))):I
FXD<0THENX=14-X
230 Y=INT(RND(1)*(15-(ABS(YD)*(L-1)))):I
FYD<0THENY=14-Y
240 XP=X:YP=Y:FORT=1TOL:IFXP<00RXP>140RY
P<00RYP>14THEN210
250 XT=22+XP:YT=6+YP:GOSUB900:IFK<>32AND
K+64<>ASC(MID$(W$(M),T,1))THEN210
260 XP=XP+XD:YP=YP+YD:NEXT:X(M)=X:Y(M)=Y
:XD(M)=XD:YD(M)=YD:FORT=1TOL
270 XT=22+X:YT=6+Y:GOSUB900:PRINTMID$(W$
(M),T,1):X=X+XD:Y=Y+YD:NEXT:XT=7:YT=6+M
280 GOSUB900:PRINTW$(M):NEXT
290 FORX=0TO14:FORY=0TO14:XT=22+X:YT=6+Y
:GOSUB900
300 IFK=32THENPRINTCHR$(INT(RND(1)*26)+6
5)
310 NEXT:NEXT:X=0:Y=0:WW=0
320 Z=PEEK(56321)AND15:IF(PEEK(56321)AND
16)=0THEN500
330 IFZ>8ANDZ<12THENX=X-1:IFX<0THENX=14
340 IFZ>4ANDZ<8THENX=X+1:IFX>14THENX=0
350 IFZ=10ORZ=14ORZ=6THENY=Y-1:IFY<0THEN
Y=14
360 IFZ=9ORZ=13ORZ=5THENY=Y+1:IFY>14THEN
Y=0
370 IFZ<>15THENPOKESO+4,33:POKESO+4,0
380 XT=X+22:YT=6+Y:GOSUB900:KO=K:IFKO<12
8THENPRINTCHR$(18);CHR$(KO+64)
390 IFKO>127THENPRINTCHR$(KO-64)
```

```
400 GOSUB900:IFKO>127THENPRINTCHR$(18);
410 PRINTCHR$((KOANDI27)+64):GOTO320
500 XT=4:YT=4:GOSUB900:POKE198,0:INPUT"W
ORD NUMBER";G$:G=VAL(G$)-1:IFG=-1THEN500
510 GOSUB900:PRINT"                    ":
IFG>14THEN320
520 IFX(G)<>XORY(G)<>YTHENGOTO320
530 XP=X:YP=Y:FORT=1TOLEN(W$(G)):XT=22+X
P:YT=6+YP:GOSUB900
540 PRINTCHR$(18);MID$(W$(G),T,1);:XT=6+
T:YT=6+G:GOSUB900:PRINTMID$(W$(G),T,1)
550 XP=XP+XD(G):YP=YP+YD(G):NEXT:XT=4:YT
=4:X(G)=-1:WW=WW+1:IFWW<15THEN320
560 XT=4:YT=4:GOSUB900:PRINT"YOU DID IT!
   PRESS <F1>"
570 GETA$:IFA$<>CHR$(133)THEN570
580 RUN
700 DIMW$(14),X(14),Y(14),XD(14),YD(14):
GOSUB800:XT=2:YT=5:GOSUB900
710 PRINT"ENTER FIFTEEN WORDS...":FORT=0
TO14:XT=2:YT=T+7:GOSUB900
720 PRINTT+1;CHR$(157)".";TAB(7);:INPUTW
$(T):IFW$(T)=""THENT=T-1:NEXT
730 W$(T)=LEFT$(W$(T),15):NEXT:XT=2:YT=5
:GOSUB900:PRINT"ARE YOUR WORDS CORRECT?"
740 GETA$:IFA$=""THEN740
750 IFA$="Y"THEN200
760 RUN
800 PRINTCHR$(147):PRINT:PRINT"        **
* JOYSTICK WORD HUNT ***":RETURN
900 POKECX,XT:POKECY,YT:SYSCR:AD=YT*40+X
T+1024:K=PEEK(AD):POKE198,0:RETURN
1000 FORBYTE=CRTOCR+9:READA:POKEBYTE,A:N
EXT
1010 POKESO+24,15:POKESO+1,100:RETURN
1020 DATA24,174,10,192,172,11,192,76,240
,255
```

2

Anagrams

An anagram is a scramble of letters. For example, the word *and* has the following anagrams:

nad
nda
adn
dna
dan

Sometimes the anagram of a word gives rise to other words. For example, consider these two sets of anagrams:

word:	top
anagrams:	pot
	pta
	tpa
	opt
	otp

In this case, two anagrams of "top" are also words—"opt" and "pot."

word:	meat	
anagrams:	team	eatm
	tema	eamt
	tmea	emat
	tmae	emta
	tame	etma
	taem	etam
	aemt	meta
	aetm	mate
	amet	maet
	amte	mtae
	atem	mtea
	atme	

Of the 24 variations, only three and at most four (if you accept *meta* as a word) make up words. An interesting and, so far as we know, unsolved word problem is whether there is a four letter word all of whose anagrams make up other words. You could easily write a computer program that could generate all the anagrams of any given word using string variables. However, the program here is more of a games form. You type in any word and the computer scrambles it. Then ask

a friend to try to unscramble it. The computer will keep score and tell your friend when he or she has the correct unscrambling.

```
10 REM       *** ANAGRAM ***
100 PRINTCHR$(147):PRINT:PRINT:PRINT"
          *** ANAGRAMS ***"
110 GOSUB300:PRINT:PRINT:PRINT:PRINT"ENT
ER YOUR WORD...":PRINT:INPUTA$
120 PRINTCHR$(145);CHR$(145);CHR$(145);:
IFA$=""THENRUN
130 PRINT"TRY TO SOLVE THIS ANAGRAM...":
PRINT
140 L=LEN(A$):DIMW(L):FORT=1TOL
150 R=INT(RND(1)*L)+1:IFW(R)=-1THEN150
160 W(R)=-1:B$=B$+MID$(A$,R,1):NEXT
170 PRINTB$;"   ":GOSUB300:PRINTTAB(20-L/
2);
180 FORT=1TOL:PRINTCHR$(164);:NEXT:GOSUB
350
190 B$="":INPUTB$:IFB$=""THENPRINTCHR$(1
45);CHR$(145):GOTO190
200 FORT=1TOL
210 IFMID$(A$,T,1)=MID$(B$,T,1)THENGOSUB
300:PRINTTAB(19-L/2+T);MID$(A$,T,1)
220 NEXT:IFA$=B$THEN250
230 GOSUB300:GOSUB350:PRINT"
                              ";CHR$(145)
240 GOTO190
250 GOSUB300:GOSUB350:PRINT:PRINT:PRINT"
                THAT'S IT!"
260 GETA$:IFA$=""THEN260
270 RUN
300 PRINTCHR$(19):PRINT:PRINT:PRINT:PRIN
T:PRINT:PRINT:PRINT:RETURN
350 PRINT:PRINT:PRINT:PRINT:PRINT:PRINT:
PRINT:PRINT:RETURN
```

3

Sentence Scrambler

This program will take a sentence and scramble all of the words in it. For short sentences, the unscrambling is quite easy. For example:

walk. took a I

quite easily unscrambles into:

I took a walk.

Try this one, however:

at time the You stars. tell looking by the can

When you type in the program be very careful about capitals, punctuation, and spacing. The computer responds to all of these different aspects of writing. Spaces between words are particularly crucial in mixing up sentences because sensing spaces is the only way the computer can tell where a word ends. The computer cannot read, at least yet. A blank space is just another symbol to it, one which, in this program, indicates where a unit to be scrambled ends.

The main functions used in this program are LEN(A$) and MID$(A$,X,Y). LEN(A$) tells you the length of any word you store in the computer's memory. For example, in Commodore BASIC, if A$ is the word "FINE" then

```
LEN(A$)=4
```

The function MID$(A$,X,Y) lets you print out any letter or sequence of letters in A$. Thus, using the same example:

```
        MID$(A$,1,1)=F
and     MID$(A$,2,2)=IN
```

Here is a simple educational program illustrating the use of LEN and A$:

```
10 DIM A$(26),B$(1)
20 A$="ABCDEFGHIJKLMNOPQRSTUVWXYZ"
30 PRINT "HOW MANY LETTERS ARE IN THE ALPHABET?
40 INPUT X
50 IF X= LEN (A$) THEN PRINT "GOOD. NOW TRY THIS:"
60 LET Y=INT(RND(1)*26)+1
70 PRINT "WHAT LETTER COMES ";Y;"th in the alphabet?"
```

```
80 INPUT B$
90 IF B$=(A$,Y,1) THEN PRINT "RIGHT."
100 IF B$ < > MID$(A$,Y,1) THEN PRINT "I'M SORRY YOU
MISSED."
```

This program checks the length of A$ and matches it with the number you put in for X. Then it picks a random position out of A$ using the Y variable and asks you which letter corresponds to that position. This is just a simple use of LEN and MID$. Play with these functions and look at your BASIC Reference Manual for more information on how to use them.

Now here is the scrambled sentence program:

```
10 REM *** SENTENCE SCRAMBLER ***
100 CR=49152:CY=CR+10:CX=CY+1:GOSUB500
110 POKE53280,0:POKE53281,0:PRINTCHR$(14
7);CHR$(5)
115 DIM A(50),M(50),A$(50),B$(50):POKE54
296,15:POKE52277,0:POKE54278,240
120 PRINT:PRINT:PRINT"          *** SENTENC
E SCRAMBLER ***"
130 X=2:Y=7:GOSUB400:PRINT"ENTER YOUR SE
NTENCE...":PRINT:INPUTS$:PRINT:PRINT
140 L=LEN(S$):C=0:P=1:EE=0:FORT=1TOL:B$(
L)=" ":NEXTT
150 FORT=1TOL:IFMID$(S$,T,1)=" "THEN200
160 Z$=MID$(S$,T,1):IFZ$="."ORZ$="!"ORZ$
="?"THENEE=1:GOTO200
170 NEXTT:T=T-1:EE=1
200 L(C)=T-P+1:A$(C)=MID$(S$,P,T-P+1):IF
EE=0THENC=C+1:P=T+1:NEXTT
250 L(C)=L(C)+1:FORT=0TOC
260 R=INT(RND(1)*(C+1))+1:IFB$(R)="*"THE
N260
270 B$(R)="*":M(R-1)=T:NEXTT
280 X=2:Y=7:GOSUB400:PRINT"TRY TO UNSCRA
MBLE THIS....":PRINT:A$(C)=A$(C)+" "
290 Y=9:GOSUB600:X=2:GOSUB400:FORT=0TOC:
PRINTA$(M(T));:NEXT
300 X=2:Y=16:GOSUB400:INPUTB$
305 IFB$<>S$THENX=15:Y=21:GOSUB400:PRINT
"TRY AGAIN":Y=16:GOSUB600:GOTO300
310 X=14:Y=21:GOSUB400:PRINT"YOU DID IT"
:POKE54276,33:FORT=50TO80STEP5
320 POKE54273,T:FORJ=1TO200STEP50:POKE54
272,J:NEXTJ:NEXTT
330 POKE54276,0
340 GETK$:IFK$=""THEN340
```

```
350 RUN
400 POKECX,X:POKECY,Y:SYSCR:RETURN
500 FORBYTE=CRTOCR+9:READA:POKEBYTE,A:NE
XT:RETURN
510 DATA24,174,10,192,172,11,192,76,240,
255
600 X=1:GOSUB400:PRINT"
                    ":RETURN
```

4

Alphawiz

This is the basic shoot 'em up arcade game in its simplest form. The game displays the alphabet at the bottom of the screen. A target appears above one of the letters at the top of the screen. You have to press the correct letter to hit the target. The letters have a small amount of fire power and you will see your hits and misses on the screen. You get five points per hit and lose five per miss.

If you want to play a little code game, take a look at the PRINT statement at line 120. It tells the computer to print out the alphabet in order at the bottom of the screen. Now, if you changed that line using a simple shift code as follows:

ABCDEFGHIJKLMNOPQRSTUVWXYZ
ZABCDEFGHIJKLMNOPQRSTUVWXY

then, if you pressed A the Z would shoot; if you pressed B the A would shoot, etc. Your shooting would then be in code. After a while you could probably figure out the structure of the code.

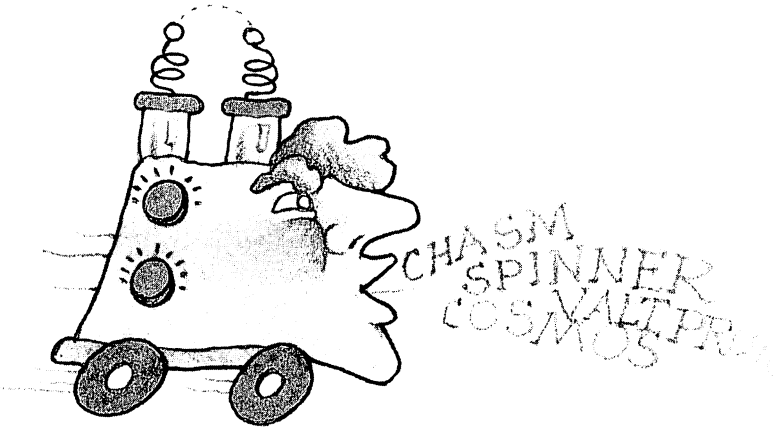
```
10 REM      *** ALPHAWIZ ***
100 SO=54272:CR=49152:CY=CR+10:CX=CY+1:H
S=-9999:GOSUB500
110 POKE53280,11:POKE53281,0:PRINTCHR$(5
);CHR$(147):TI$="000000":S=0
120 PRINTCHR$(5):XT=7:YT=21:GOSUB400:PRI
NT"ABCDEFGHIJKLMNOPQRSTUVWXYZ"
```

```
130 TX=INT(RND(1)*26)+7:FORT=0TO6:TX(T)=
RND(1)*6-3+TX:TY(T)=RND(1)*6+1:NEXT
140 XT=TX:YT=4:GOSUB400:PRINTCHR$(166):X
T=3:YT=23:GOSUB400:GOSUB450
150 IFTI>3600THEN300
160 GETA$:IFA$=""THEN150
170 XT=ASC(A$):IFXT<65ORXT>90THEN150
180 XT=XT-58:POKE50+4,129:FORYT=20TO0STE
P-1:POKE50+1,YT*2+5:GOSUB400
190 PRINTCHR$(28);CHR$(122);CHR$(5);:IFY
T<20THENPRINTCHR$(157);CHR$(17);" "
200 IFYT<>4ORXT<>TXTHENGOSUB400:NEXT:PRI
NT" ":POKE50+4,0:POKE198,0:S=S-5:GOTO140
210 POKE50+4,128:POKE50+1,3:POKE50+4,129
:PRINTCHR$(129):FORT=0TO6:XT=TX(T)
220 YT=TY(T):GOSUB400:PRINTCHR$(113):NEX
T:POKE50+4,128:FORT=0TO6:XT=TX(T)
230 YT=TY(T):GOSUB400:PRINT" ":NEXT:S=S+
5:XT=TX:YT=4:GOSUB400:PRINT" ":GOTO120
300 IFS>HSTHENHS=S
310 XT=17:YT=23:GOSUB400:PRINT"*** GAME
OVER ***":PRINT"    *HIGH:";HS;TAB(17);
320 PRINT"     PRESS <F1>";
330 GETA$:IFA$<>CHR$(133)THEN330
340 GOTO110
400 POKECX,XT:POKECY,YT:SYSCR:RETURN
450 PRINT"SCORE:";S;CHR$(157)" ":RETURN
500 FORBYTE=CRTOCR+9:READA:POKEBYTE,A:NE
XT
510 FORT=50TO50+24:POKET,0:NEXT:POKE50+2
4,15:POKE50+5,10:POKE50+6,9:RETURN
530 DATA24,174,10,192,172,11,192,76,240,
255
```

5

Creative Name Machine

Did you ever wonder how people come up with names for computer games or businesses or products? Most of the process is free association and fiddling around with words. The computer can often do this



as well as people, especially if you make your original lists of words interesting enough. This program generates some names you might like to use one day. Some of them are COSMOS WEAVER, CHASM SPINNER, and VAULT PRODUCER. If you look at lines 550 and 590 of the program you'll see the DATA statements that provide the words for the computer to play with. You can change these with any lists you have so long as you remember to separate the words by a comma. Line 110 reads a word from the list at 550 and line 120 reads a word from 590. The combination is what you see printed out. The READ ... DATA .... statement in BASIC allows you to play around with lists of data.

Now try our names and add some twists of your own.

```
1 REM  *** CREATIVE NAME MACHINE ***
2 REM
100 CR=49152:CY=CR+10:CX=CY+1:GOSUB500:P
OKE53281,11:PRINTCHR$(5):DIMA$(1,100)
110 READA$(0,G1):G1=G1+1:IFA$(0,G1-1)<>"
END OF LIST"THEN110
120 READA$(1,G2):G2=G2+1:IFA$(1,G2-1)<>"
END OF LIST"THEN120
130 G1=G1-1:G2=G2-1
140 POKE53280,INT(RND(1)*16):PRINTCHR$(1
47)
150 A$=A$(0,INT(RND(1)*G1))+" "+A$(1,INT
(RND(1)*G2))
```

```
160 POKECX,20-LEN(A$)/2:POKECY,10:SYSCR:
PRINTA$
170 GETA$:IFA$=""THEN170
180 GOTO140
500 FORBYTE=CRTOCR+9:READA:POKEBYTE,A:NE
XT
510 RETURN
530 DATA24,174,10,192,172,11,192,76,240,
255
540 REM      *** WORD GROUP #1 ***
550 DATADUNGEON,VAULT,CRYPT,CAVERN,ADVEN
TURE,WORLD,ENCOUNTER,SPACE,CHASM
560 DATAUNIVERSE,MICROCOSM,COSMOS
570 DATA END OF LIST
580 REM      *** WORD GROUP #2 ***
590 DATAMACHINE,WEAVER,SPINNER,VENTURE
600 DATABUILDER,SYNTHESIZER,INVENTOR,CRE
ATOR,COMPOSER,PRODUCER
610 DATA END OF LIST
```

6

Excuse Generator

Did you ever wish you had an automatic excuse generator that would cover you when you were late for work, forgot to do homework, or wanted to get out of a party or meeting? Well, here's an attempt at one. Basically, this program consists of an empty letter form in which you fill in the particulars. If you don't like the way we've done it, you change all of the PRINT statements in the program or add some new ones of your own. The program is a phoney personal letter generator. You can use it to create personalized business letters, invitations to parties, etc. Our excuse for including it in the book was to show what a letter or response form is like and how easily it can be changed on a computer.

```
DEAR PALLID,
I'M VERY SORRY THAT I DON'T HAVE
THE TIME.  YOU SEE, MY
BEST FRIEND LOST IT
WHILE I PICKED FLEAS FROM HIS COAT.
GEE, I SURE AM SORRY.  I WON'T LET
IT HAPPEN AGAIN.  I PROMISE.

SINCERELY,
LIVID
```

```
10 REM  *** EXCUSE GENERATOR ***
100 PRINTCHR$(147):PRINT:PRINT"
   EXCUSE GENERATOR"
110 DIME$(2,7):FORT=0TO2:FORU=0TO6:READE
$(T,U):NEXT:NEXT:DEFFNR(X)=INT(RND(1)*X)
120 PRINT:PRINT:PRINT:INPUT"YOUR NAME";A
$:IFA$=""THENA$="ME"
130 PRINT:PRINT:INPUT"PERSON TO GET EXCU
SE";B$:IFB$=""THENB$="TEACHER"
140 PRINT:PRINT:INPUT"ITEM TO BE EXCUSED
";C$:IFC$=""THENC$="HOMEWORK"
150 PRINTCHR$(147):PRINT:PRINT"DEAR ";B$
;",":PRINT
160 PRINT"I'M VERY SORRY THAT I DON'T HA
VE":PRINT:PRINT"THE ";C$;". YOU SEE, MY"
170 PRINT:PRINTE$(0,FNR(7));" ";E$(1,FNR
(7)):PRINT:PRINT"WHILE I ";E$(2,FNR(7))
180 PRINT:PRINT"GEE, I SURE AM SORRY. I
WON'T LET":PRINT
190 PRINT"IT HAPPEN AGAIN. I PROMISE.":P
RINT:PRINT:PRINT"SINCERELY,":PRINT
200 PRINTA$
210 GETA$:IFA$=""THEN210
220 RUN
500 DATADOG,DAD,BABY BROTHER,OLDER BROTH
ER,CAT,BEST FRIEND,BOSS
510 DATAATE IT,USED IT FOR KINDLING,SLOB
BERED ALL OVER IT,TORE IT TO PIECES
520 DATASHINED HIS SHOES WITH IT,STOLE I
T,LOST IT,PICKED FLEAS FROM HIS COAT.
530 DATAWAS ON A TRIP.,CHANGED HIS DIAPE
RS.,WASN'T WATCHING.,WAS ASLEEP.
540 DATAWAS FEEDING HIM CAT FOOD.,STOOD
BY HELPLESSLY.
```

PART II

Number Play

This section contains several number games you may not have seen before as well as some conversion and sorting programs you can use for many different purposes. However, the first program in the section is a challenge that comes from Marlyn Burns, the author of the *I Hate Math Book* and *Math for Smarty Pants*. We translated her challenge into computer form. It can be fun to do similar things by taking books on mathematical games and recreations and giving those challenges a computer twist. Often, the computer can make the game more powerful because it can check challenges quickly, provide hints, tally scores, and set up timing devices.

7

The $1.00 Word Challenge

Here is the challenge in Marlyn Burns' own words:

"Excellent" is a $1.00 word. Mathematically speaking, that is. So are "friendlier" and "grumpy" and "elephants" And so is "discipline," which is what it takes to find more of these $1.00 wonders. To find out how much a word is worth, you give each letter a cent value: a=$.01, b=$.02, c=$.03, and so on up to z=$.26. Then you add the value of each letter in a word.

Check for yourself to make sure the words given above really are worth exactly $1.00. Then you're on your own. How about starting with your first name? Is it worth $1.00? What about your friends' names? Or the town or city you live in?

The $1.00 word fever has been spreading throughout the United States. To date over 500 $1.00 words have been found by addition-loving kids across the country. Some have been writing $1.00 sentences—sentences in which every word is worth $1.00. Some examples: "Prevent inflation." "Whenever Henrietta whistled, thirty trembling costumed elephants merrily performed."

Of course, dedicated $1.00 word zealots have brought their computers into the search, programming them to calculate the value of whatever word they input. Try it. It takes all the drudge out of the search.

Some hints may be useful for getting you started. There is one Halloween word that is worth $1.00 (It's not Halloween; that's only worth $.95.) There's one Thanksgiving $1.00 word, as well as one astrological sign, and one United States coin. There are at least two United States cities—one of them in Wisconsin, and the other in both Oregon and Maine. To date, one five-letter $1.00 word has been found. No one, however, has submitted a word with four letters. (Is it possible to have a four-letter word worth $1.00?

There's a beverage that is illegal for minors to drink that's worth $1.00, though it sometimes costs more than that to buy in restaurants and bars. There are several zoo animals that are $1.00 words, and at

least one underwater creature. Something that rarely strikes twice in the same place, so they say, is also a $1.00 word. And there is a number that is less than 100 that is worth $1.00 when written as a word.

This program will check to see if your words are $1.00 words. If your word doesn't quite make a dollar, the program will also tell you how much it is worth using the coding system where A is a penny, B two cents, etc.

```
10 REM   *** ONE-DOLLAR WORDS ***
100 PRINTCHR$(147)
110 PRINTTAB(7);"THE ONE-DOLLAR WORD MAC
HINE"
120 PRINT:PRINT:INPUT"PLEASE ENTER YOUR
WORD";A$:IFA$=""THEN120
130 V=0:FORT=1TOLEN(A$):IFMID$(A$,T,1)<"
A"ORMID$(A$,T,1)>"Z"THEN120
140 V=V+ASC(MID$(A$,T,1))-64:NEXT
150 IFV=100THENPRINTA$;" IS A ONE-DOLLAR
 WORD!":GOTO120
160 PRINTA$;" IS WORTH";V;"CENTS.":GOTO1
20
```

If you want to make a $2.00 word machine or a half-dollar word machine, all you have to do is change the IF-THEN and PRINT commands on line 150. Thus, the following changes in line 150 will give a Fifty Cents Word Machine:

```
150 IF V=50 THEN PRINT A$; "IS A FIFTY CENT
WORD!":GOTO 120
```

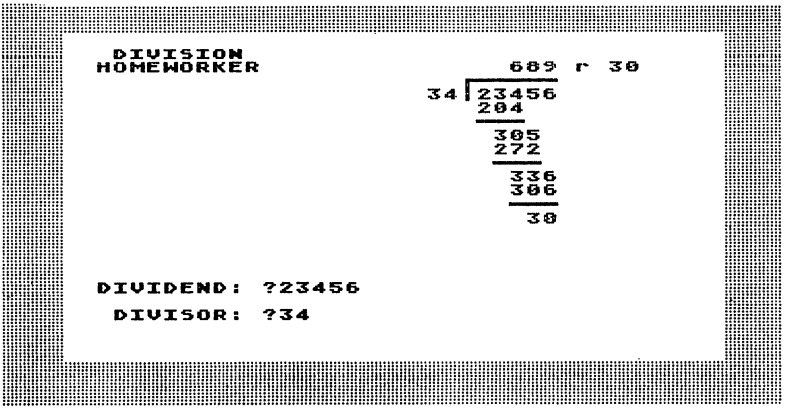
8

Division Homeworker

Did you ever get frustrated when you were asked to "show the work" when you were doing math homework? After all, a calculator or com-

puter could do the work. Why bother with the tediousness of long division anymore? There is even a rumor that long division is a device created by teachers to torture their students with boredom. We've come up with a solution to the long division nightmare. Here is a program that not merely gives you the answer to a long division problem, remainder and all, but also "shows the work." We hope it will save you hours of boredom that could be used in more productive and pleasurable activity.

The program has another value. It will take you step by step through long division problems and can be a useful way to learn the process.




```
1 REM     *** DIVISION HOMEWORKER ***
2 REM
100 CR=49152:CY=CR+10:CX=CY+1:GOSUB600:X
=24:Y=4
110 POKE53280,9:POKE53281,0:PRINTCHR$(14
7);CHR$(5):POKECX,2:POKECY,18
120 PRINT"    DIVISION":PRINT"  HOMEWORKE
R":SYSCR:INPUT "DIVIDEND";N$
130 N=VAL(N$):IFN=OORN>999999THENRUN
140 PRINT:INPUT"    DIVISOR";D$:D=VAL(D$)
:IFD=OORD>NTHENRUN
150 POKECX,X:POKECY,Y:SYSCR:PRINTCHR$(98
);CHR$(157);CHR$(145);CHR$(176);
160 FORT=1TOLEN(N$):PRINTCHR$(99);:NEXT:
POKECX,X-LEN(D$):SYSCR:PRINTD$
170 POKECX,X+1:SYSCR:PRINTN$:GOSUB400:NS
=1:QP=0:BP=0:FF=0:A$=LEFT$(N$,1)
180 NT=VAL(LEFT$(N$,NS)):IFNT=>DTHEN220
```

```
190 QP=QP+1:IFFF=1THENPOKECX,X+QP:POKECY
,Y-2:SYSCR:PRINT"O":GOSUB400
200 NS=NS+1:IFNS>LEN(N$)THENR=NT:GOTO500
210 A$=LEFT$(N$,NS):POKECX,2+X+QP-LEN(A$
):POKECY,Y+BP:SYSCR:PRINTA$:GOTO180
220 FF=1:QT=INT(NT/D):R=NT-D*QT:QP=QP+1:
POKECX,X+QP:POKECY,Y-2:SYSCR
230 PRINTRIGHT$(STR$(QT),1):GOSUB400
240 T=1+X+QP-LEN(STR$(D*QT)):POKECX,T:PO
KECY,Y+BP+1:SYSCR:PRINTQT*D:GOSUB400
250 POKECY,Y+BP+2:SYSCR:PRINT" ";:FORT=1
TOLEN(STR$(QT*D))-1:PRINTCHR$(99);:NEXT
260 POKECX,1+X+QP-LEN(STR$(R)):POKECY,Y+
BP+3:SYSCR:PRINTR:GOSUB400
270 IFNS=LEN(N$)THEN500
280 T$=RIGHT$(STR$(R),LEN(STR$(R))-1):T$
=T$+MID$(N$,NS+1,1)
290 POKECX,2+X+QP-LEN(STR$(R)):POKECY,Y+
BP+3:SYSCR:PRINTT$:A$=T$:GOSUB400
300 T$=RIGHT$(STR$(R),LEN(STR$(R))-1):T$
=T$+MID$(N$,NS+1,6):N$=T$
310 NS=LEN(A$):BP=BP+3:GOTO180
400 GETZ$:IFZ$=""THEN400
410 RETURN
500 POKECX,X+QP+2:POKECY,Y-2:SYSCR:PRINT
"R";R:GOSUB400:RUN
600 FORBYTE=CRTOCR+9:READA:POKEBYTE,A:NE
XT
610 RETURN
630 DATA24,174,10,192,172,11,192,76,240,
255
```

9

Kabala Name Game

According to the Kabala, a mystical Jewish treatise about the relationship between numbers, letters, and things spiritual, every word has a numerical value which determines its sacredness and quality. This is especially true for names. We've turned the name/number equivalence into a game. Each letter of the alphabet is given a value from 1 to 26, A being 1 and Z 26. The names displayed in the game

have a total value which consists of the sum of the values of their letters. Thus, *Abe* has a value of 1+2+5=8. (Can you think of a name with a lower value than Abe?)

The game works in the following way: A name is displayed on the top of the screen and below it four other names are displayed. You have to figure out which one of the four has the same value as the first name displayed.

There is a strategy you might use to help you eliminate some names immediately. Look at the original name and see how many of its letters are skewed toward the front of the alphabet, how many are in the middle, and how many at the end. That should give you an idea of whether the value of the whole will be low, medium, or high. For example, it is clear that *Abe* will have a low value and *Val* a much higher one. With this first take, you can scan all the names and eliminate those whose values clearly don't match up.

```
10 REM   *** KABALA NAME GAME ***
100 SO=54272:CR=49152:CY=CR+10:CX=CY+1:D
IMK$(19,3):GOSUB900
110 FORT=0TO19:FORJ=0TO3:READK$(T,J):NEX
T:NEXT
120 POKE53280,0:POKE53281,0:PRINTCHR$(14
7);CHR$(5):X=10:Y=2:GOSUB800
130 PRINT"THE KABALA NAME GAME":FORT=0TO
1:FORJ=1TO13:X=T*6+4:Y=5+J:GOSUB800
140 PRINTCHR$(T*13+J+64);"=";RIGHT$(STR$
(T*13+J),LEN(STR$(T*13+J))-1):NEXT:NEXT
150 P(3)=INT(RND(1)*20):R1=INT(RND(1)*4)
:R2=INT(RND(1)*4):IFR2=R1THEN150
160 X=25:Y=6:GOSUB800:PRINTK$(P(3),R1):F
ORT=0TO2:P(T)=-1:NEXT:FORT=0TO2
170 R=INT(RND(1)*20):FORJ=0TO3:IFP(J)=RT
HEN170
180 NEXT:P(T)=R:NEXT:FORT=0TO3:M(T)=-1:N
EXT:FORT=0TO3
190 R=INT(RND(1)*4):IFM(R)>-1THEN190
200 M(R)=T:NEXT:FORT=0TO3:X=21:Y=10+T*2:
GOSUB800:PRINTT+1;
210 IFM(T)=3THENPRINTK$(P(3),R2):R=T:GOT
O230
220 PRINTK$(P(M(T)),INT(RND(1)*4))
230 NEXT
240 GETA$:IFA$<"1"ORA$>"4"THEN240
250 A=VAL(A$)-1:X=20:Y=10+A*2:GOSUB800:P
RINT"*":X=24:Y=10+R*2:GOSUB800
260 PRINTCHR$(18);:FORT=1TOLEN(K$(P(3),R
2)):PRINTMID$(K$(P(3),R2),T,1);:NEXT
```

```
270 PRINT:POKE50+1,100:IFR=ATHENGOTO290
280 POKE50+4,17:FORT=100TO50STEP-1:POKES
0+1,T:NEXT:POKE50+4,0:GOTO300
290 POKE50+4,17:FORT=100TO150:POKE50+1,T
:NEXT:POKE50+4,0:S=S+1
300 TL=TL+1:X=7:Y=20:GOSUB800:PRINT"YOU'
VE GOTTEN";S;"OUT OF";TL;CHR$(157);"!"
310 IFTL<10THENFORT=1TO3000:NEXT:GOTO120
320 X=8:Y=22:GOSUB800:PRINT"PRESS <F1> T
O PLAY AGAIN"
330 GETA$:IFA$<>CHR$(133)THEN330
340 RUN
800 POKECX,X:POKECY,Y:SYSCR:RETURN
900 FORBYTE=CRTOCR+9:READA:POKEBYTE,A:NE
XT:POKE50+5,0:POKE50+6,240:POKE50+24,15
910 RETURN
920 DATA24,174,10,192,172,11,192,76,240,
255
1000 DATAEDDIE,ADELE,JAKE,DICK
1010 DATADIANA,TED,ANN,GAIL
1020 DATADEBRA,JANE,JOE,ALICE
1030 DATAJIM,DAVE,NEAL,LEO
1040 DATADIANE,SAM,MABEL,FRED
1050 DATABILL,BETH,HEIDI,ERIC
1060 DATAKATE,PAT,NICK,GREG
1070 DATADAVID,JOAN,NEIL,LINDA
1080 DATAMARIA,JANICE,TIM,ALEX
1090 DATAANDY,RAY,HELEN,TINA
1100 DATADANIEL,SUE,PHIL,CLARK
1110 DATABECKY,PETE,BOBBY,KATIE
1120 DATARON,SARAH,JOHN,DOUG
1130 DATAIRENE,PAULA,GARY,FREDDIE
1140 DATACATHY,GEORGE,MARY,NANCY
1150 DATAALBERT,DANNY,PEDRO,JESSE
1160 DATACHARLES,RUBY,MANUEL,MICKEY
1170 DATAVICKY,HARRY,HENRY,SIMON
1180 DATATONY,SUSAN,PENNY,JOSHUA
1190 DATAJEREMY,THOMAS,JERRY,HERBERT
```

Here are the data statements for the game. We've made lists of names of equivalent number value. You add or subtract names so long as all the names on a given line have the same value. You can also replace a line of your own with animal names or plant names and change the nature of the game that way. Just be sure to change the PRINT commands and check the numerical values of your words.

```
1000 DATAEDDIE,ADELE,JAKE,DICK
1010 DATADIANA,TED,ANN,GAIL
1020 DATADEBRA,JANE,JOE,ALICE
```

```
1030 DATAJIM,DAVE,NEAL,LEO
1040 DATADIANE,SAM,MABEL,FRED
1050 DATABILL,BETH,HEIDI,ERIC
1060 DATAKATE,PAT,NICK,GREG
1070 DATADAVID,JOAN,NEIL,LINDA
1080 DATAMARIA,JANICE,TIM,ALEX
1090 DATAANDY,RAY,HELEN,TINA
1100 DATADANIEL,SUE,PHIL,CLARK
1110 DATABECKY,PETE,BOBBY,KATIE
1120 DATARON,SARAH,JOHN,DOUG
1130 DATAIRENE,PAULA,GARY,FREDDIE
1140 DATACATHY,GEORGE,MARY,NANCY
1150 DATAALBERT,DANNY,PEDRO,JESSE
1160 DATACHARLES,RUBY,MANUEL,MICKEY
1170 DATAVICKY,HARRY,HENRY,SIMON
1180 DATATONY,SUSAN,PENNY,JOSHUA
1190 DATAJEREMY,THOMAS,JERRY,HERBERT
```

10

Pitch and Toss

This game is thoroughly determined by chance.

Games of chance exist everywhere in the world. Often they are associated with predicting the future or making important decisions when there are no reasons to choose one way of acting over another. They also are good gambling games since every player is supposed to have an equal chance of winning. In our society, the most familiar chance games involve dice, playing cards, and spinners like the roulette wheel. In other parts of the world, different devices are used. For example, throughout the African continent cowrie shells, nuts, and elaborately carved wood tokens are commonly used in chance games. The characteristic of these devices is that they have two sides like our coins. A number of them are cast and scores are made according to which faces show up.

For example, there is a game played by the Igbo of Nigeria which consists of casting four cowrie shells which can fall back up or open side up.

The best scores are four backs up or four backs down.

There are 16 possibilities when tossing the four cowries: (U = back up, D = back down)

UUUU	UUDD	DDUU
UUUD	UDUD	UDDD
UUDU	UDDU	DUDD
UDUU	DUUD	DDUD
DUUU	DUDU	DDDU
		DDDD

Since the cowries can be considered to look alike, these possibilities can be simplified in the following way:

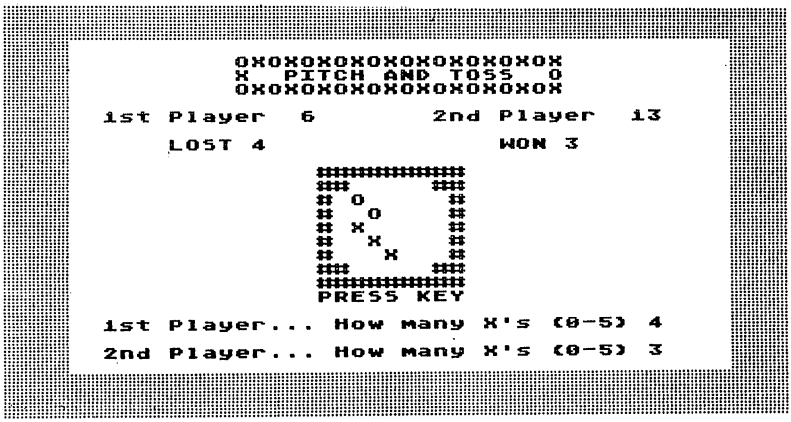
4 up 0 down — 1 way
3 up 1 down — 4 ways
2 up 2 down — 6 ways
1 up 3 down — 4 ways
0 up 4 down — 1 way

Because of this, the following scoring system could be used for the game, giving fewer points to the arrangement most likely to happen, and more points to the rarer combinations.

4 up or 4 down	4 points
1 up 3 down	2 points
3 up 1 down	2 points
2 up 2 down	1 point

Try the game with cowries, walnuts, or pennies. It is as interesting as playing dice and provides more awareness of arrays of figures and scoring combinations.

This game is a computer version of the cowrie game. The computer will throw five cowries onto the playing board. One side of the cowrie will be represented by X and the other side as O. You have to guess how many X's will come up on your toss. The game is for two people and both players guess on each throw. The scoring is as follows: If you guess correctly you get points equal to the number of X's that you predicted would come up. If your partner guesses correctly, you lose points equal to the number of your guess. If neither of you guess correctly there is no score and you go on to the next round.




```
10 REM     *** PITCH AND TOSS ***
100 CR=49152:CY=CR+10:CX=CY+1:GOSUB900
110 POKE53280,0:POKE53281,0:PRINTCHR$(14
7);CHR$(5):S$(0)="O":S$(1)="X"
115 PRINTCHR$(153);:S1=10:S2=10
120 PRINT"           OXOXOXOXOXOXOXOXOX"
125 PRINT "           X   ";CHR$(156);"PITC
H AND TOSS";CHR$(153);"   X"
130 PRINT"           OXOXOXOXOXOXOXOXOX"
:PRINTCHR$(5):GOSUB600:GOSUB700
200 X=2:Y=20:GOSUB800:PRINT"1ST PLAYER..
.HOW MANY X'S (0-5) ";:GOSUB400
210 PRINTK:G1=K:X=2:Y=22:GOSUB800:PRINT"
2ND PLAYER...HOW MANY X'S (0-5) ";:
220 GOSUB400:IFK=G1THEN220
230 PRINTK:G2=K
240 C=0:FORT=0TO4:X=INT(RND(1)*5):P=INT(
RND(1)*2):X=17+X:Y=11+T:GOSUB800
250 PRINTS$(P):IFP=1THENC=C+1
260 NEXT:IFC=G1THENS1=S1+G1:S2=S2-G2:GOS
UB300
270 IFC=G2THENS1=S1-G1:S2=S2+G2:GOSUB350
280 GOSUB600:IFS1<10RS2<1THENX=5:Y=13:GO
SUB800:GOTO650
290 GOTO810
300 X=6:Y=7:GOSUB800:PRINT"WON";G1:X=26:
GOSUB800:PRINT"LOST";G2:RETURN
350 X=6:Y=7:GOSUB800:PRINT"LOST";G1:X=26
:GOSUB800:PRINT"WON";G2:RETURN
400 GETK$:IFK$=""THEN400
410 K=ASC(K$)-48:IFK<0ORK>5THEN400
420 RETURN
600 X=2:Y=5:GOSUB800:PRINTCHR$(159);"1ST
 PLAYER ";S1;CHR$(157);"   ":X=22
```

```
610 GOSUB800:PRINT"2ND PLAYER ";S2;CHR$(
157);"   ":PRINTCHR$(5):RETURN
650 PRINTCHR$(158);"GAME OVER":X=25:Y=13
:GOSUB800:PRINTCHR$(158);"GAME OVER"
660 GOTO820
700 X=15:Y=9:GOSUB800:PRINTCHR$(31);"###
######":Y=10:GOSUB800
705 PRINT"##      ##":FORT=1TO5:Y=10+T:GO
SUB800
710 PRINT"#        #":NEXT:Y=16:GOSUB800:
PRINT"##      ##"
720 Y=17:GOSUB800:PRINT"##########":PRINT
CHR$(5):RETURN
800 POKECX,X:POKECY,Y:SYSCR:RETURN
810 X=15:Y=18:GOSUB800:PRINT"PRESS KEY"
820 GETK$:IFK$=""THEN820
830 X=15:Y=18:GOSUB800:PRINT"           "
:X=6:Y=7:GOSUB800:PRINT"       ":X=26
840 GOSUB800:PRINT"       "
850 FORT=20TO23:X=2:Y=T:GOSUB800:PRINT"
                                    ";
860 NEXT:GOSUB700:IFS1<10RS2<1THENRUN
870 GOTO200
900 FORBYTE=CRTOCR+9:READA:POKEBYTE,A:NE
XT:RETURN
910 DATA24,174,10,192,172,11,192,76,240,
255
```

11

Date to Weekday

This is a conversion program. You can enter any date—past, present, or future—and this program will tell you which day of the week it happened to be. Leap years are also indicated. Using this program, you can find out the day you and your friends and family were born. You can find out the days Napoleon, Emma Goldman, and Julius Caesar were born. You can find out what day Christmas will fall on in 2100 or what day it fell on in 1166.

```
10 REM    *** DATE TO WEEKDAY ***
100 CR=49152:CY=CR+10:CX=CY+1:GOSUB900:P
RINTCHR$(5);CHR$(147):POKE53281,0
110 POKE53280,0:X=9:Y=3:GOSUB800:PRINT"D
ATE TO DAY CONVERSION":X=2:Y=6:GOSUB800
120 DIMM(27),Y1(12),W$(6),Y2$(12):FORT=0
TO27:READM(T):NEXT:FORT=1TO12:READY1(T)
130 NEXT:FORT=0TO6:READW$(T):NEXT:FORT=1
TO12:READY2$(T):NEXT
140 INPUT"MONTH (1-12)";M:PRINT:PRINT:IN
PUT"  DAY (1-31)";D:PRINT:PRINT
150 INPUT"  YEAR (0-  )";Y:PRINT:PRINT
160 YH=INT(Y/28):YK=M(Y-YH*28):IFINT(Y/4
)=Y/4THENLP=1
170 FORT=0TOM-1:N=N+Y1(T):NEXT:N=N+D-1+Y
K:IFM>2THENN=N+LP
180 N=N-7*INT(N/7):D$=W$(N):M$=Y2$(M)
190 PRINTM$;D;CHR$(157);",";Y;"WAS A ";D
$;"."
200 IFLP=1THENPRINT:PRINTY;"WAS A LEAP Y
EAR"
210 X=15:Y=20:GOSUB800:PRINT"PRESS <F1>"
220 GETA$:IFA$<>CHR$(133)THEN220
230 RUN
800 POKECX,X:POKECY,Y:SYSCR:RETURN
900 FORBYTE=CRTOCR+9:READA:POKEBYTE,A:NE
XT:POKE50+5,0:POKE50+6,240:RETURN
910 DATA24,174,10,192,172,11,192,76,240,
255
920 DATA4,6,0,1,2,4,5,6,0,2,3,4,5,0,1,2,
3,5,6,0,1,3,4,5,6,1,2,3
930 DATA31,28,31,30,31,30,31,31,30,31,30
,31
940 DATAMONDAY,TUESDAY,WEDNESDAY,THURSDA
Y,FRIDAY,SATURDAY,SUNDAY
950 DATAJANUARY,FEBRUARY,MARCH,APRIL,MAY
,JUNE,JULY,AUGUST,SEPTEMBER,OCTOBER
960 DATANOVEMBER,DECEMBER
```

12

President Precedence

For multiple choice fans, here's a challenge to test your knowledge of the Presidential terms of office. A date appears at the top of the screen. The names of three Presidents appear by the numbers one, two, and three. Press a number on the keyboard to match the correct President with his term.

```
10 REM *** PRESIDENT PRECEDENCE ***
100 CR=49152:CY=CR+10:CX=CY+1:GOSUB800:P
RINTCHR$(5);CHR$(147):POKE53280,0
110 POKE53281,0:DIMA$(39,1):FORT=0TO39:R
EADA$(T,0):READA$(T,1):NEXT
120 P(0)=INT(RND(1)*40):P(1)=INT(RND(1)*
40):P(2)=INT(RND(1)*40)
130 IFP(1)=P(2)ORP(2)=P(0)ORP(0)=P(1)THE
N120
140 PRINTCHR$(147):PRINT:PRINTTAB(10)"PR
ESIDENT PRECEDENCE":X=15:Y=6:GOSUB700
150 PRINTA$(P(0),1):FORT=0TO2:M(T)=-1:NE
XT:Y=10:X=0:GOSUB700:FORT=0TO2
160 R=INT(RND(1)*3):IFM(R)>-1THEN160
170 M(R)=T:NEXT:FORT=0TO2:PRINTT+1;A$(P(
M(T)),0):NEXT
180 GETA$:IFA$<"1"ORA$>"3"THEN180
190 A=VAL(A$)-1:IFM(A)<>OTHEN180
200 X=16:Y=18:GOSUB700:PRINT"CORRECT":FO
RT=1TO1000:NEXT:GOTO120
700 POKECX,X:POKECY,Y:SYSCR:RETURN
800 FORBYTE=CRTOCR+9:READA:POKEBYTE,A:NE
XT:POKESO+5,0:POKESO+6,240:RETURN
810 DATA24,174,10,192,172,11,192,76,240,
255
850 DATAGEORGE WASHINGTON,1789-1797,JOHN
 ADAMS,1797-1801
855 DATATHOMAS JEFFERSON,1801-1809,JAMES
 MADISON,1809-1817
860 DATAJAMES MONROE,1817-1825,JOHN QUIN
CY ADAMS,1825-1829
865 DATAANDREW JACKSON,1829-1837,MARTIN
VAN BUREN,1837-1841
870 DATAWILLIAM HENRY HARRISON,1841-1841
,JOHN TYLER,1841-1845
```

```
875 DATAJAMES K. POLK,1845-1849,ZACHARY
TAYLOR,1849-1850
880 DATAMILLARD FILLMORE,1850-1853,FRANK
LIN PIERCE,1853-1857
885 DATAJAMES BUCHANAN,1857-1861,ABRAHAM
 LINCOLN,1861-1865
890 DATAANDREW JOHNSON,1865-1869,ULYSSES
 S. GRANT,1869-1877
895 DATARUTHERFORD B. HAYES,1877-1881,JA
MES A. GARFIELD,1881-1881
900 DATACHESTER A. ARTHUR,1881-1885,GROV
ER CLEVELAND (1),1885-1889
905 DATABENJAMIN HARRISON,1889-1893,GROV
ER CLEVELAND (2),1893-1897
910 DATAWILLIAM MCKINLEY,1897-1901,THEOD
ORE ROOSEVELT,1901-1909
915 DATAWILLIAM H. TAFT,1909-1913,WOODRO
W WILSON,1913-1921
920 DATAWARREN G. HARDING,1921-1923,CALV
IN COOLIDGE,1923-1929
925 DATAHERBERT C. HOOVER,1929-1933,FRAN
KLIN D. ROOSEVELT,1933-1945
930 DATAHARRY S. TRUMAN,1945-1953,DWIGHT
 D. EISENHOWER,1953-1961
935 DATAJOHN F. KENNEDY,1961-1963,LYNDON
 B. JOHNSON,1963-1969
940 DATARICHARD M. NIXON,1969-1974,GERAL
D R. FORD,1974-1977
945 DATAJIMMY CARTER,1977-1981,RONALD RE
AGAN,1981-????
```

This program can be modified to match events to dates. For example, in the data lines from 850 to 945 you can put the name of a battle with its dates or the name of a great invention with the date it was invented. As long as you do not change the structure of the data and keep to associating a name with a number you will be able to modify this program and make up your own multiple choice quizzes. There is one thing to be careful about, however. On line 110, the program is set up to deal with 40 data inputs (FOR T=0 to 39) and on line 120 to deal with the length of your DATA statement. If you keep to the same number of items and same length of items in the program you'll have no trouble. If not, you should change the parameters on lines 110 and 120.

PART III

IQ Building

Intelligence Quotient (IQ) implies that there is some objective measure of how smart people are. An IQ of 100 is supposed to be normal, 140 brilliant, and 75 slow. However, no one knows how accurate even the most sophisticated tests are. Your IQ has to do with how you perform on the tests you take and not with how intelligently you function with others and at your work.

Since the measure of IQ is so test-bound, your score can be changed if you practice the skills that the test designers have chosen as epitomizing the essence of "intelligence." Usually, these skills have to do with memorizing lists, figuring out sequences, estimating answers quickly, and knowing how to manipulate strings of symbols. This Part provides a number of games and exercises that hone these skills. Of course, there is no guarantee that your IQ will be raised by playing them but there is a chance that, if you master these modes of thinking, you will find it easier to deal with the challenges of IQ tests.

13

Gestalt

The word *Gestalt* refers to a perceptual whole. When we look at a chair or table we do not see its individual parts and then conclude that they go together to make a chair or table. We see the chair or table as an integrated whole or Gestalt. A Gestalt problem that English philosophers debated about 30 years ago led to this computer program. The problem was usually phrased this way:

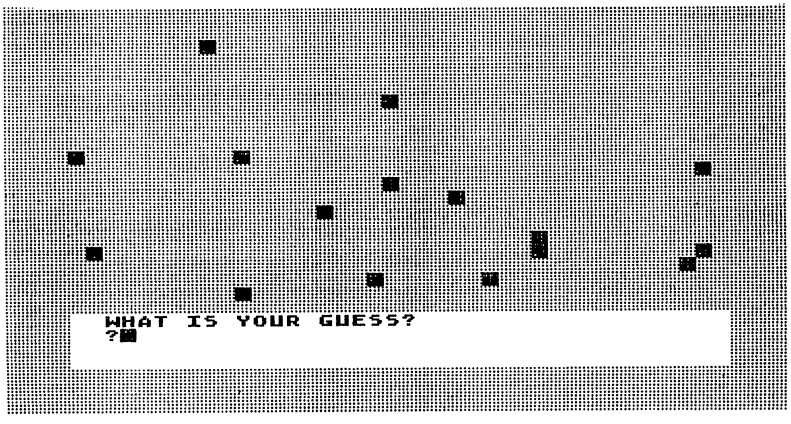
You are looking at a field of sheep. Exactly how many sheep do you see? If a sheep were added or subtracted would you see any more or fewer sheep?

If you do not stop to count the sheep, it would be accurate to say that you saw a whole bunch of sheep, not a particular number of sheep. You could, however, estimate the number of sheep and, if you had a great deal of experience with sheep, could get more and more accurate. Estimation even within the context of an initially disordered Gestalt can be refined and improved. This program gives you an opportunity to guess at the number of dots on your computer screen without counting them. The computer will tell you how close your guess is and then give you another series of dots to guess about. This kind of estimation is related to IQ tests in that the ability to estimate the probability of an answer makes it possible to eliminate the least likely choices of a multiple choice question and, therefore, gives you more time to work on problems.

The program also helps you concentrate on responding to your perceptions and refining them. It is practice in looking at and interpreting what you see.

```
10 REM        *** GESTALT ***
100 CR=49152:CY=CR+10:CX=CY+1:GOSUB900
110 POKE53280,0:POKE53281,0:PRINTCHR$(14
7);CHR$(5)
120 PRINT:PRINT"THIS IS AN ESTIMATION GA
ME.":PRINT:PRINT"YOU WILL SEE DOTS ON TH
E SCREEN."
```

```
130 PRINT:PRINT"DO NOT TRY TO COUNT THE
DOTS.":PRINT:PRINT"JUST ESTIMATE HOW MAN
Y.":PRINT
140 PRINT"AFTER A WHILE YOU SHOULD GET B
ETTER"
150 PRINT:PRINT:PRINT"PRESS A KEY TO BEG
IN"
160 GETK$:IFK$=""THEN160
170 PRINTCHR$(147);:Q=INT(RND(1)*20)+5
180 FORZ=1TOQ:X=INT(RND(1)*39):Y=INT(RND
(1)*21):GOSUB800:PRINTCHR$(119):NEXT
190 X=2:Y=21:GOSUB800:INPUT"YOUR ESTIMAT
E ";S
200 IFS=QTHENPRINT"   YOU GOT IT!!":GOTO
220
210 PRINT"   YOU ARE";ABS(S-Q);"OFF"
220 FORT=1TO1500:NEXTT:RUN
800 POKECX,X:POKECY,Y:SYSCR:RETURN
900 FORBYTE=CRTOCR+9:READA:POKEBYTE,A:NE
XT:RETURN
910 DATA24,174,10,192,172,11,192,76,240,
255
```



This program may put too many dots on the screen at the same time or run too quickly for you. If you want fewer dots on the screen to begin with, you can change the value of Q on line 170:

```
Q=INT(RND(1)*20)+5
```

For example, if you change the 5 at the end of the line to 0, you'll get a maximum of 20 dots and if you change it to 5 you'll get a maximum of 10.

If you want to slow the program down, you can put a pause loop such as

```
FOR X=1 TO 500:NEXT X
```

into the program after you plot the points (line 220).

14

Flash Line

Flash Line is a length estimating program. It flashes two lines on the screen and asks you which one is longer. If you get the correct answer, it gives you another set of lines. If you get it wrong, it gives you another chance. Most of the challenges of this program are easy to answer once you get the rhythm of looking and inputting your answer. However, the game can be speeded up by changing the pause loops at lines 170 and 230. You can also change the length of the lines by reducing the random value in line 120. When you reduce the size of the line the game gets more complex because the sizes of the lines don't vary as much.

Notice that there is a machine language routine at lines 900 and 910 that you will find in many of the programs in this book. This routine is in effect a positioning statement that sets up the screen placement of the game or utility.

```
10 REM       *** FLASHLINE ***
100 CR=49152:CY=CR+10:CX=CY+1:GOSUB900
110 POKE53280,0:POKE53281,0:PRINTCHR$(14
7);CHR$(5)
120 X(1)=INT(RND(1)*15)+10:X(2)=INT(RND(
1)*15)+10:IFX(1)=X(2)THEN120
130 PRINTCHR$(147):PRINT:PRINT" WHICH ON
E IS LONGER,THE 1ST OR 2ND?"
140 FORT=1TO2:X=1:Y=10:GOSUB800
150 PRINT"
        ":GOSUB800
160 FORJ=1TOX(T):PRINTCHR$(18);CHR$(28);
" ";:NEXTJ:PRINTCHR$(146);CHR$(5)
170 FORJ=1TO400:NEXTJ:NEXTT
180 X=1:Y=19:GOSUB800:PRINT"TYPE 1 OR 2"
190 GETA$:IFA$<>"1"ANDA$<>"2"THEN190
200 A=VAL(A$):B=2:IFX(1)>X(2)THENB=1
210 IFA=BTHENPRINT:PRINT"RIGHT. HERE'S A
NOTHER ONE.":C=1
220 IFA<>BTHENPRINT:PRINT"WRONG. TAKE AN
OTHER LOOK.":C=2
230 FORT=1TO1000:NEXT:ONCGOTO120,130
799 STOP
800 POKECX,X:POKECY,Y:SYSCR:RETURN
900 FORBYTE=CRTOCR+9:READA:POKEBYTE,A:NE
XT:RETURN
910 DATA24,174,10,192,172,11,192,76,240,
255
```

15

Did You See It?

Here is a third estimating game. This program moves circles around the screen randomly. A challenging perceptual experiment would be

to see whether substituting other symbols changes the difficulty of the task. For example, you could substitute symbols such as "?" or "m" or "#" and see whether some symbols are easier to decode than others in a game such as this.

The substitution should be made at line 160 for CHR$(119). If you look at the back of your *Commodore Reference Manual* you will see a list of CHR$ and ASCII code numbers. ASCII codes give numbers to each symbol you can use on your Commodore 64. The CHR$ function allows you to print out or use a character. For example, the ASCII code for @ is 64. If you typed PRINT CHR$(64) you would get @. If, on the other hand, you typed ASCII("@") you would get the number 64.

Here is a simple program that will print out ASCII code numbers and their character equivalents:

```
10 FOR X=32 TO 191
20 PRINT "ASCII= ";X;" ";"CHR$= ";CHR$(X)
30 NEXT
```

```
10 REM      *** DO YOU SEE IT ***
100 CR=49152:CY=CR+10:CX=CY+1:GOSUB900
110 POKE53280,0:POKE53281,0:PRINTCHR$(14
7);CHR$(5)
120 X=2:Y=20:GOSUB800:PRINT"------------
-------------------------":X=5:Y=22
130 GOSUB800:PRINT"HOW MANY DO YOU SEE?"
:S=INT(RND(1)*10)+5
140 FORM=1TO500:NEXTM
150 FORW=1TOS
160 X=INT(RND(1)*40):Y=INT(RND(1)*20):GO
SUB800:PRINTCHR$(119)
170 FORR=1TO275:NEXTR:GOSUB800:PRINT" ":
NEXTW
180 X=5:Y=23:GOSUB800:INPUTG
190 IFG=STHENPRINT"              YOU GOT IT
!";
200 IFG<>STHENPRINT"              TRY AGAIN
";:FORT=1TO500:NEXTT:GOTO150
210 FORT=1TO500:NEXT:RUN
800 POKECX,X:POKECY,Y:SYSCR:RETURN
900 FORBYTE=CRTOCR+9:READA:POKEBYTE,A:NE
XT:RETURN
910 DATA24,174,10,192,172,11,192,76,240,
255
```

16

Memory Muscle Builder

This challenge begins by showing you two digits to memorize. If you can type them in correctly you will then be shown three digits and so on until your mind is overcome with numbers. When you finally miss you will be assigned one of the 15 ranks of memory power we have built into the program. Repeating number sequences occur on a number of individually administered IQ tests and, with some training, you can become a memory master. The memorization of sequences is also a good exercise in the development of concentration.

```
10 REM *** MEMORY MUSCLE BUILDER ***
100 CR=49152:CY=CR+10:CX=CY+1:GOSUB900
110 POKE53280,0:POKE53281,0:PRINTCHR$(14
7);CHR$(5):DIM A(15),N$(14):N=1
115 FORT=0TO14:READN$(T):NEXT
120 PRINTCHR$(147):X=7:Y=2:GOSUB800:PRIN
T"* MEMORY MUSCLE BUILDER *"
130 X=2:Y=6:GOSUB800:PRINT"WATCH CAREFUL
LY...":FORT=1TO150:NEXT
140 X=1:Y=8:GOSUB800:FORT=0TON:A(T)=INT(
RND(1)*10):PRINTA(T);:NEXT
150 FORT=1TO200+N*600:NEXT:X=1:Y=8:GOSUB
800:FORT=0TON:PRINT" ? ";:NEXT
160 X=2:Y=10:GOSUB800:PRINT"NOW, ENTER T
HE NUMBERS YOU SAW":X=2:Y=12:GOSUB800
170 POKE198,0:FORT=0TON
180 GETK$:IFK$=""THEN180
190 K=ASC(K$)-48:IFK<0ORK>9THEN180
200 PRINTK;:IFK<>A(T)THEN300
210 NEXT:X=2:Y=14:GOSUB800:PRINT"WOW, WH
AT A MEMORY!":N=N+1:IFN=16THEN400
220 FORT=1TO750:NEXT:GOTO120
300 X=2:Y=14:GOSUB800:PRINT"OOPS! THE NU
MBERS WERE..."
310 Y=16:GOSUB800:FORT=0TON:PRINTA(T);:N
EXT
320 Y=18:GOSUB800:PRINT"YOUR RANK:  ";N$
(N)
330 FORT=1TO2000:NEXT:RUN
800 POKECX,X:POKECY,Y:SYSCR:RETURN
```

```
900 FORBYTE=CRTOCR+9:READA:POKEBYTE,A:NE
XT:RETURN
910 DATA24,174,10,192,172,11,192,76,240,
255
920 DATA NOVICE CLASS 2,NOVICE CLASS 1,I
NITIATE CLASS 2,INITIATE CLASS 1
930 DATA AVERAGE CLASS 2,AVERAGE CLASS 1
,MEMORY WHIZ CLASS 2
940 DATA MEMORY WHIZ CLASS 1,MASTER CLAS
S 2,MASTER CLASS 1
950 DATA MEMORY MASTER CLASS 2,MEMORY MA
STER CLASS 1,GRANDMASTER
960 DATA PRODIGY,GENIUS
```

If you look carefully at this program you will see that the variable N indicates how many numbers you guessed correctly and, at line 320, is used to assign you a rank. The assignment of ranks depends upon a READ...DATA... command in which each rank becomes a single data statement as listed below:

```
920 DATA NOVICE CLASS 2,NOVICE CLASS 1,I
NITIATE CLASS 2,INITIATE CLASS 1
930 DATA AVERAGE CLASS 2,AVERAGE CLASS 1
,MEMORY WHIZ CLASS 2
940 DATA MEMORY WHIZ CLASS 1,MASTER CLAS
S 2,MASTER CLASS 1
950 DATA MEMORY MASTER CLASS 2,MEMORY MA
STER CLASS 1,GRANDMASTER
960 DATA PRODIGY,GENIUS
```

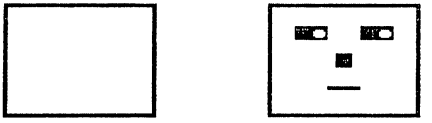
If you want to create your own ranks, all you have to do is list the program and change the data lines to any titles you choose. If you want to create a subroutine in another program that uses a ranking system, all you have to do is adapt lines 320 and 920 to 960 for your own purposes.

17

Mug Shot

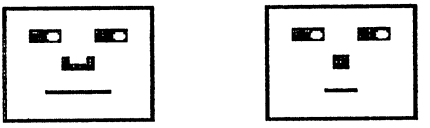
Mug Shot is a different kind of memory game. Mug shots are face photos that victims of crime are given in order to identify criminals. In this program, the computer creates a face for you to look at for a while and then clears the screen and challenges you to recreate the face from a choice of eyes, noses, and mouths. After you think you have successfully reconstructed the face, the computer shows you the original face next to your reconstruction. This program provides a deceptively complex challenge and gives a sense of how hard it is to reconstruct a face from memory. You can study this program and add a number of other features to the face. For example, you can put on eyebrows or hair, or even make a whole series of different shaped faces (oval, round, squarish, etc.).

MUGSHOT MEMORY TEST



REMEMBER THIS MAN'S FACE....

MUGSHOT MEMORY TEST



SORRY, THAT'S NOT THE MAN.

MUGSHOT MEMORY TEST

OKAY, DO YOU RECOGNIZE THE NOSE?

1.

2.

3.

```
10 REM  *** MUGSHOT MEMORY TEST ***
100 CR=49152:CY=CR+10:CX=CY+1:GOSUB800
110 O$(0)="EYES":O$(1)="NOSE":O$(2)="MOU
TH"
120 POKE53280,0:POKE53281,0:PRINTCHR$(14
7);CHR$(5)
130 FORT=0TO2:FORU=0TO2:FORV=0TO5:READA:
F$(T,U)=F$(T,U)+CHR$(A):NEXT:NEXT:NEXT
140 PRINTTAB(11)"MUGSHOT MEMORY TEST":GO
SUB600:GOSUB610:FORT=0TO2
150 A(T)=INT(RND(1)*3):NEXT:GOSUB550:X=6
:Y=16:GOSUB700
160 PRINT"REMEMBER THIS MAN'S FACE....":
FORT=1TO1500:NEXT:GOSUB610:GOSUB500
170 FORT=0TO2:X=2:Y=16:GOSUB700:PRINT"OK
AY, DO YOU RECOGNIZE THE ";O$(M(T));"? "
180 FORJ=0TO2:Y=18+J*2:GOSUB700:PRINTJ+1
;"   ";F$(M(T),J):NEXT
190 GETA$:IFA$=""THEN190
200 IFVAL(A$)<1ORVAL(A$)>3THEN190
210 B(M(T))=VAL(A$)-1:X=9:Y=8+2*M(T):GOS
UB700:PRINTF$(M(T),B(M(T))):NEXT
220 X=2:Y=16:GOSUB700:FORT=1TO7:PRINT"
                                    ":NEXT
230 GOSUB550:FORT=0TO2:IFA(T)=B(T)THENNE
XT:GOTO250
240 X=7:Y=16:GOSUB700:PRINT"SORRY, THAT'
S NOT THE MAN.":GOTO260
250 X=13:Y=16:GOSUB700:PRINT"THAT'S THE
MAN!"
260 GETA$:IFA$=""THEN260
270 RUN
499 END
500 FORT=0TO2:M(T)=-1:NEXT:FORT=0TO2
510 R=INT(RND(1)*3):IFM(R)>-1THEN510
520 M(R)=T:NEXT:RETURN
550 FORT=0TO2:X=25:Y=8+2*T:GOSUB700:PRIN
TF$(T,A(T)):NEXT:RETURN
600 X=7:GOSUB650:RETURN
610 X=23:GOSUB650:RETURN
650 Y=6:GOSUB700:PRINTCHR$(117);:FORT=1T
08:PRINTCHR$(99);:NEXT:PRINTCHR$(105)
660 FORT=1TO7:Y=Y+1:GOSUB700:PRINTCHR$(1
25);"          ";CHR$(125):NEXT:Y=14
670 GOSUB700:PRINTCHR$(106);:FORT=1TO8:P
RINTCHR$(99);:NEXT:PRINTCHR$(107):RETURN
700 POKECX,X:POKECY,Y:SYSCR:RETURN
800 FORBYTE=CRTOCR+9:READA:POKEBYTE,A:NE
XT:RETURN
810 DATA24,174,10,192,172,11,192,76,240,
255
```

```
900 DATA40,119,41,40,119,41,32,113,32,32
,113,32,117,105,32,32,117,105
910 DATA32,32,109,110,32,32,32,32,106,10
7,32,32,32,32,172,187,32,32,32,99,99,99
920 DATA99,32,32,32,99,99,32,32,106,99,9
9,99,99,107
```

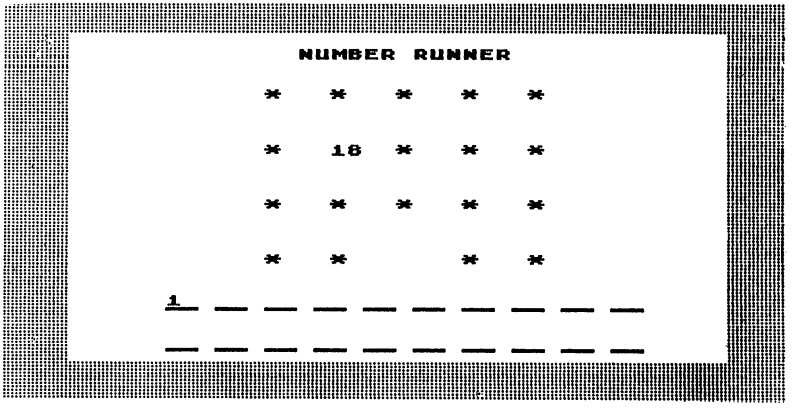
This program uses **SHIFT** and **COMMODORE KEY** graphics characters. Your Commodore 64 actually has four keyboards. Capitals and lower case letters make up one. The other two use the graphics symbols printed on your keyboard. You get the symbols on the left by pressing the C= key and those on the right by pressing the **SHIFT** key. Thus, C= S gives you a right angle shape and **SHIFT S** gives you a small heart.

In Mugshot we have used the ASCII numbers to tell the computer what characters to print. There is an explanation of the use of ASCII and CHR$ in Project 15, "Did You See It?"

18

Numbers Runner

In this game you are presented with a grid consisting of 20 stars as well as 20 blanks at the bottom of the screen. The numbers from 1 to 20 have been hidden under the stars randomly. The game is played with a joystick in port 1. Using the joystick, you can move around the grid. When you press the button on the joystick a number will appear under the star you are currently on. You goal is to place the numbers on the blank spaces at the bottom of the screen in order from 1 to 20. When you find 1, press the button and a 1 will appear in the first blank on the upper left. When you find 2 press the button and a 2 will appear in the blank to the right of 1. You have to remember what is under the stars that you have tested out in order to get all 20 numbers before your time runs out. There is a timer built into this program as



well as a score keeper. The timer sets the length of a turn to one minute. The scorer will record your highest score.

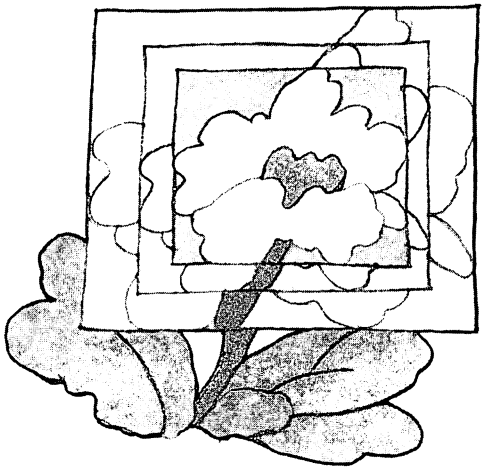
This is a memory development game—one that involves the kinds of skills called upon when you are asked about details of stories you have read or pictures you have seen. It is possible, after a while, to register all of the numbers that you land on and solve the puzzle in half the time you are allowed.

The timer in the program is controlled by line 290. The variable TI$ refers to a second timing clock built into the Commodore 64. Line 290 jumps to 500 once it has counted approximately 60 seconds. Line 500 prints out your score. Notice that line 170 also has the variable TI$ and that it sets it to "000000," at the beginning of each game. Experiment with these lines in other programs and soon you'll be able to set up timing devices in your own programs.

```
1 REM        *** NUMBER RUNNER ***
2 REM
100 SO=54272:CR=49152:CY=CR+10:CX=CY+1:G
OSUB1000:DIMX(4,3),Y(4,3),N(4,3),M(20)
110 POKE53280,0:POKE53281,0:PRINTCHR$(14
7)TAB(14)CHR$(5)"NUMBER RUNNER"
120 PRINTCHR$(154):FORT=1TO20:M(T)=0:NEX
T:FORX=0TO4:FORY=0TO3
130 N=INT(RND(1)*20)+1:IFM(N)=1THEN130
140 M(N)=1:N(X,Y)=N:X(X,Y)=X*4+12:Y(X,Y)
=Y*4+2:GOSUB700
150 PRINTCHR$(18)" "CHR$(146):NEXT:NEXT
160 FORY=20TO23STEP3:FORX=0TO27STEP3:POK
ECX,X+6:POKECY,Y:SYSCR
```

```
170 PRINTCHR$(101)CHR$(101)" ":NEXT:NEXT
:X=2:Y=1:RX=2:RY=19:R=1:TI$="000000"
180 IF((PEEK(56321)AND16)=0)ANDN(X,Y)>0
THEN400
190 T=PEEK(56321)AND15:OX=X:OY=Y
200 IFT>8ANDT<12THENX=X-1:IFX<0THENX=4
210 IFT>4ANDT<8THENX=X+1:IFX>4THENX=0
220 IFT=10ORT=14ORT=6THENY=Y-1:IFY<0THEN
Y=3
230 IFT=9ORT=13ORT=5THENY=Y+1:IFY>3THENY
=0
240 IF(OX=X)AND(OY=Y)THEN260
250 POKES0+1,40:POKES0+4,33:FORT=60TO0ST
EP-10:POKES0+1,100-T:NEXT:POKES0+4,0
260 GOSUB700:PRINT" ":IFN(X,Y)=0THENGOSU
B700:PRINTCHR$(5)"0"CHR$(154)
270 FORT=1TO50:NEXT:IFN(X,Y)>0THENGOSUB7
00:PRINTCHR$(18)" "CHR$(146):GOTO290
280 GOSUB700:PRINT" "
290 IFVAL(TI$)>100THEN500
300 GOTO180
400 GOSUB700:PRINTCHR$(5)CHR$(157)N(X,Y)
CHR$(154):GOSUB600
410 IF(PEEK(56321)AND16)=16THEN410
420 IFN(X,Y)<>RTHENGOSUB700:PRINTCHR$(18
)" "CHR$(146)" ":GOSUB600:GOTO180
430 RX=RX+3:IFRX>34THENRX=5:RY=22
440 POKECX,RX:POKECY,RY:SYSCR:PRINTCHR$(
5)RCHR$(154):R=R+1:GOSUB700:PRINT"   "
450 POKES0+1,0:POKES0+4,33:FORT=0TO255ST
EP5:POKES0+1,T:NEXT:POKES0+4,0
460 N(X,Y)=0:IFR<21THENGOSUB600:GOTO180
470 PRINTCHR$(5)CHR$(19)TAB(14)"     YAY!
    "
500 POKECX,14:POKECY,16:SYSCR:PRINTCHR$(
5)"YOUR SCORE:"R-1:IFR-1>HSTHENHS=R-1
510 PRINTTAB(14)"HIGH SCORE:"HS:GOSUB600
:FORT=1TO1000:NEXT
520 IF(PEEK(56321)AND16)=0THEN110
530 GOTO520
600 IF(PEEK(56321)AND16)=0THEN600
610 RETURN
700 POKECX,X(X,Y):POKECY,Y(X,Y):SYSCR:RE
TURN
1000 FORBYTE=CRTOCR+9:READA:POKEBYTE,A:N
EXT
1010 FORT=S0TOS0+24:POKET,0:NEXT:POKES0+
24,6:POKES0+5,15:POKES0+6,15:RETURN
1020 DATA24,174,10,192,172,11,192,76,240
,255
```

A Pause for Meditation



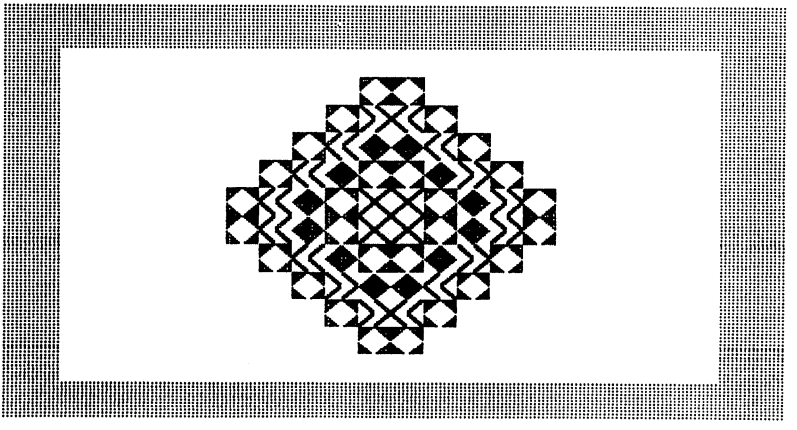
19

Meditative Images

Here is a program you can relax with. It generates different colors and images that show off the **SHIFT** and **COMMODORE KEY** graphics as well as the color capabilities of your Commodore 64. Hopefully it will provide a relaxing, slowly changing visual experience to savor, a respite after IQ madness.

```
10 REM  *** MEDITATIVE IMAGES ***
100 CR=49152:CY=CR+10:CX=CY+1:GOSUB900:F
ORT=0TO4:FORU=0TO9:READK
110 M$(T)=M$(T)+CHR$(K):NEXT:NEXT:POKE53
280,9:POKE53281,2:PRINTCHR$(147);CHR$(5)
120 FORY=0TO8STEP2
130 LI=INT(RND(1)*5)
140 IFLI=I(Y)THEN130
150 I(Y)=LI:RI=LI:IFLI>2THENRI=7-LI
```

```
160 FORX=0TOYSTEP2
170 XT=18-X:YT=10-Y+X:GOSUB800:PRINTM$(L
I)
180 XT=18-X:YT=12+Y-X:GOSUB800:PRINTM$(L
I)
190 XT=20+X:YT=10-Y+X:GOSUB800:PRINTM$(R
I)
200 XT=20+X:YT=12+Y-X:GOSUB800:PRINTM$(R
I)
210 NEXT:FORQ=1TO600:NEXT:NEXT
220 POKE53280,INT(RND(1)*15):GOTO120
800 POKECX,XT:POKECY,YT:SYSCR:RETURN
900 FORBYTE=CRTOCR+9:READA:POKEBYTE,A:NE
XT:RETURN
910 DATA24,174,10,192,172,11,192,76,240,
255
920 DATA31,18,169,127,146,157,157,17,127
,169
930 DATA156,169,127,157,157,17,18,127,16
9,0
940 DATA158,109,110,157,157,17,110,109,0
,0
950 DATA5,109,109,157,157,17,110,110,0,0
960 DATA5,110,110,157,157,17,109,109,0,0
```

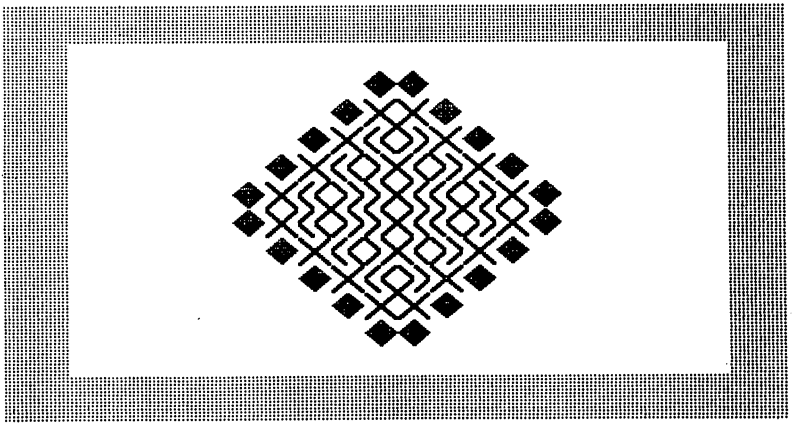


In order to change the colors of the characters printed to the screen on the Commodore 64 you must print the character code corresponding to the color desired. That color will be the color of all subsequent prints until it is changed by printing another code.

The first character in each of the shapes defined in the program is a color code. For instance, the first number in the data on lines 950 and 960 is a 5 which is the code for white.

The background and border colors are set by POKEing a number from 0 to 15 into locations 53281 and 53280, respectively. On line 110, POKE 53280,9 turns the border to brown and POKE 53281,2 turns the background red.

The codes for changing text colors are ASCII codes and may be found in the ASCII charts in the User's Guide or Reference Manual. But the POKEs to change the background and border colors are COLOR NUMBERS 0 to 15. A chart of these can also be found in the Manual or the Guide.



PART IV

Strategy Puzzles

The games and puzzles in this chapter require some thought. Their outcome is not decided by a spinner, a roll of dice, or any other chance device. If you work at them you should be able to figure out a way to solve their challenges (or, in one case, prove that a solution is impossible) and even beat the computer which has been programmed to play against you. If you read through the commentary on the games you'll also find out how to add a timer to your own games as well as some other information about game challenges and programming strategies.

20

The Pattern Game

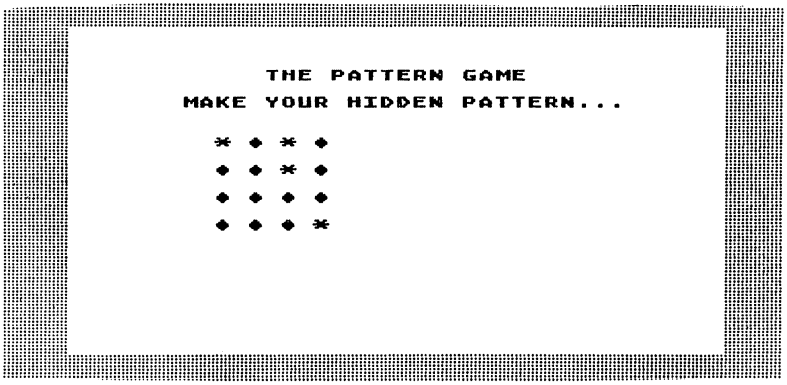
This game requires cooperation in an indirect way. You are given a 4 x 4 grid and asked to make a pattern on it. Using a joystick, you can create a star pattern by moving the cursor to any spot on the grid where you would like to place a star. If you press the button, a star will appear on your grid. Press **F1** when you are finished and your pattern will be covered over. Have a friend try to guess your pattern by moving the joystick and pressing the button to place a star where he or she thinks it might be. Here's where the cooperation comes in. If you don't create a regular pattern on the grid, your friend will certainly have a hard time reconstructing your grid. In fact, if you simply place stars at random on the grid, the game will be no fun at all. You have to create a pattern that is challenging and decipherable. By the way, if your friend is getting frustrated he or she can press the **F7** key to get a clue. One of the stars will be revealed.

Here are a few challenging patterns:

```
* * * *
*  "  "  *
*  "  "  *
* * * *
```

```
* * " "
" " * *
* * " "
" " * *
```

```
"  *  "  *
*  "  *  "
"  *  "  *
*  "  *  "
```



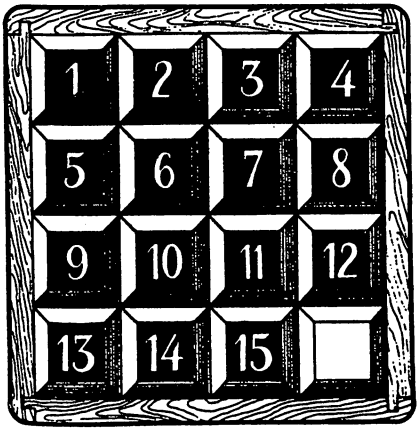

```
10 REM   *** THE PATTERN GAME ***
100 DIMP1(3,3),P2(3,3):SO=54272:CR=49152
:CY=CR+10:CX=CY+1:GOSUB900:C(1)=122
110 C(2)=42:C(3)=88:POKE53280,0:POKE5328
1,0:PRINTCHR$(147);CHR$(5)
120 XT=12:YT=3:GOSUB800:PRINT"THE PATTER
N GAME":XT=7:YT=5:GOSUB800
130 PRINT"MAKE YOUR HIDDEN PATTERN...":X
T=9:YT=8:GOSUB800:K=1:GOSUB700
140 XT=4:YT=19:GOSUB800:PRINT"PRESS <F1>
 WHEN YOU'VE FINISHED.":X=0:Y=0:CL=0
150 P=PEEK(56321)AND16:IFP=0THENP1(X,Y)=
P1(X,Y)+1:IFP1(X,Y)>3THENP1(X,Y)=1
160 IFP=0THENXT=9+X*2:YT=8+Y*2:GOSUB800:
PRINTCHR$(C(P1(X,Y))):GOSUB550
170 GOSUB600:XT=9+X*2:YT=8+Y*2:GOSUB800:
PRINT" ":IFX<>ZXORY<>ZYTHENGOSUB650
180 GOSUB500:XT=9+X*2:YT=8+Y*2:GOSUB800:
PRINTCHR$(C(P1(X,Y))):GOSUB500
190 GETA$:IFA$<>CHR$(133)THENPOKE198,0:G
OTO150
200 XT=4:YT=5:GOSUB800:PRINT"NOW GUESS T
HE HIDDEN PATTERN...":XT=9:YT=8:GOSUB800
210 K=3:GOSUB700:XT=24:YT=8:GOSUB800:K=2
:GOSUB700:XT=4:YT=21:GOSUB800:X=0:Y=0
220 PRINT"PRESS <F7> TO GET A CLUE."
400 P=PEEK(56321)AND16:IFP=0THENP2(X,Y)=
P2(X,Y)+1:IFP2(X,Y)>3THENP2(X,Y)=1
410 IFP=0THENXT=24+X*2:YT=8+Y*2:GOSUB800
:PRINTCHR$(C(P2(X,Y))):GOSUB550
420 GOSUB600:XT=24+X*2:YT=8+Y*2:GOSUB800
:PRINT" ":IFX<>ZXORY<>ZYTHENGOSUB650
430 GOSUB500:XT=24+X*2:YT=8+Y*2:GOSUB800
:PRINTCHR$(C(P2(X,Y))):GOSUB500
440 GETA$:IFA$=CHR$(136)THEN750
```

```
450 IFA$<>CHR$(133)THENPOKE198,0:GOTO400
460 POKE198,0:FORJ=0TO3:FORT=0TO3:IFABS(
P1(T,J))<>P2(T,J)THENGOSUB670:GOTO400
470 NEXT:NEXT:XT=1:YT=5:GOSUB800:PRINT"Y
OU DID IT WITH";CL;
480 PRINTCHR$(157);" CLUES!  PRESS <F1>"
:FORT=1TO9:GOSUB650:NEXT
490 GETA$:IFA$<>CHR$(133)THEN490
495 RUN
500 FORT=1TO5:IF(PEEK(56321)AND16)=0OR(P
EEK(56321)AND15)<>15THENT=5
510 NEXT:RETURN
550 IF(PEEK(56321)AND16)=0THEN550
560 RETURN
600 Z=(PEEK(56321)AND15):ZX=X:ZY=Y:IFZ>8
ANDZ<12THENX=X-1:IFX<0THENX=0
610 IFZ>4ANDZ<8THENX=X+1:IFX>3THENX=3
620 IFZ=10ORZ=14ORZ=6THENY=Y-1:IFY<0THEN
Y=0
630 IFZ=9ORZ=13ORZ=5THENY=Y+1:IFY>3THENY
=3
640 RETURN
650 POKE50+1,100:POKE50+4,33:FORQ=100TO2
50STEP10:POKE50+1,Q:NEXT:POKE50+4,0
660 RETURN
670 POKE50+1,100:POKE50+4,33:FORQ=100TO0
STEP-1:POKE50+1,Q:NEXT:POKE50+4,0
680 RETURN
700 FORJ=0TO3:FORT=0TO3:IFK=2THENPRINTCH
R$(C(P2(T,J)));" ";
710 IFK=1THENP1(T,J)=1:P2(T,J)=1:PRINTCH
R$(122);" ";
720 IFK=3THENPRINT". ";
730 NEXT:YT=YT+2:GOSUB800:NEXT:RETURN
750 POKE198,0:IFP1(X,Y)<0THEN400
760 XT=9+X*2:YT=8+Y*2:GOSUB800:PRINTCHR$
(C(P1(X,Y))):CL=CL+1:P2(X,Y)=P1(X,Y)
770 P1(X,Y)=-P1(X,Y):GOTO400
800 POKECX,XT:POKECY,YT:SYSCR:RETURN
900 FORBYTE=CRTOCR+9:READA:POKEBYTE,A:NE
XT:POKE50+5,0:POKE50+6,240:RETURN
910 DATA24,174,10,192,172,11,192,76,240,
255
```

21, 22

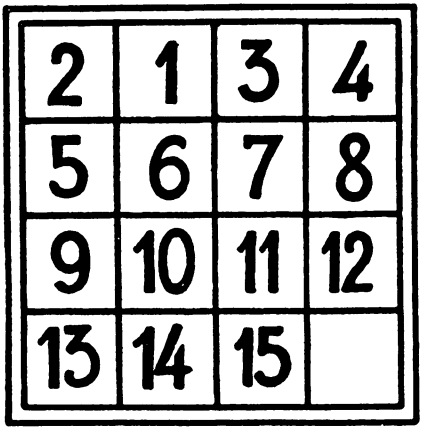
3 x 3 and 4 x 4 Number Puzzlers

The 3x3 and 4x4 Number Puzzlers are computer versions of puzzles that are over 100 years old. The earliest versions of the puzzles consisted of wooden trays with numbered wood blocks laid into them as in this old drawing:

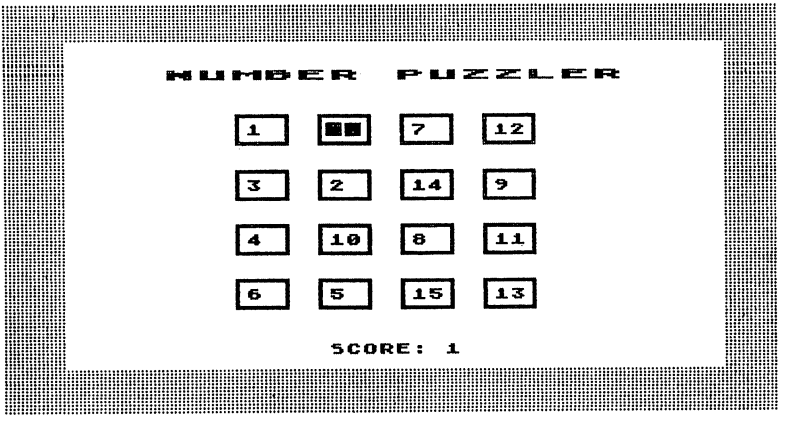


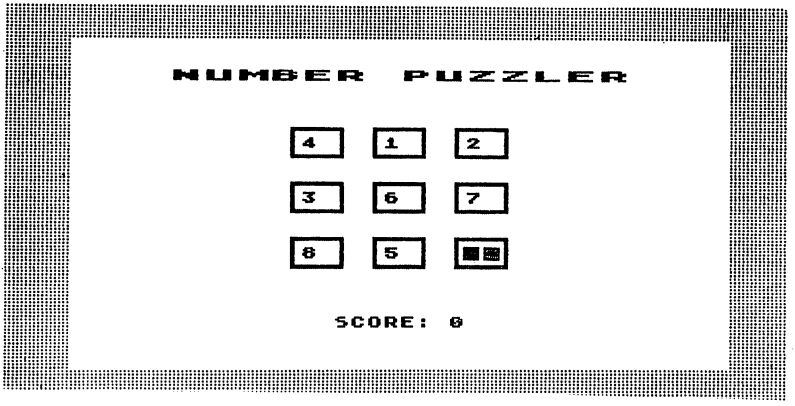
The 4x4 puzzle consists of a 4x4 square grid with the numbers 1 through 15 printed on the blocks. There is one blank square on the grid. The blocks can be moved onto the empty square and, little by little, shuffled around the entire grid. The puzzle consists of shifting the numbers around the grid and then reassembling them in consecutive order. Here is one possible challenge on the 4x4 board:

Put this square in order so that 1 is at the top left and all the other numbers proceed in order from left to right and then top to bottom as in the first drawing:



The computer versions of the program first use a 3x3 board and the numbers 1 to 8, and then a 4x4 board and numbers 1 to 15. The computer shuffles the numbers for you and gives you a board to solve. The empty square is marked by double blocks (■■). Using a joystick, you can move the empty square vertically or horizontally onto any adjoining square. The number in that square will then move to the previously empty square. Thus, in the computer version you are moving the empty square and the numbers are switched for you. The net result, however, is identical to moving the blocks into the empty square. There is a counter built into the program that will let you know how many numbers you have in the right place.






```
1 REM       *** 3 BY 3 PUZZLER ***
2 REM
100 SO=54272:DIMP(2,2),A(9):CR=49152:CY=
CR+10:CX=CY+1:GOSUB500
110 POKE53280,0:POKE53281,0:PRINTCHR$(14
7);CHR$(5);CHR$(17)
120 FORT=1TO26:READK:B$=B$+CHR$(K):NEXT
130 PRINT:PRINTTAB(8)"N U M B E R   P U
Z Z L E";CHR$(158)
140 FORY=0TO2:FORX=0TO2
150 T=INT(RND(1)*9):IFA(T+1)=1THENGOTO15
0
160 POKECX,14+5*X:POKECY,7+4*Y:SYSCR
170 PRINTB$;:IFT<>0THENPRINTRIGHT$(STR$(
T),LEN(STR$(T))-1):GOTO190
180 PRINTCHR$(18)"  "CHR$(146):PX=X:PY=Y
190 A(T+1)=1:P(X,Y)=T:NEXT:NEXT:GOTO310
200 S=PEEK(56321)AND15:OX=PX:OY=PY
220 IFS=14THENPY=PY-1:IFPY<0THENPY=0:GOT
O200
230 IFS=13THENPY=PY+1:IFPY>2THENPY=2:GOT
O200
240 IFS=11THENPX=PX-1:IFPX<0THENPX=0:GOT
O200
250 IFS=7THENPX=PX+1:IFPX>2THENPX=2:GOTO
200
260 IFPX=OXANDPY=OYTHEN200
270 POKECX,14+PX*5:POKECY,7+PY*4:SYSCR:P
RINTB$;CHR$(18)"  "CHR$(146)
280 POKECX,14+OX*5:POKECY,7+OY*4:SYSCR:P
RINTB$;
290 PRINTRIGHT$(STR$(P(PX,PY)),LEN(STR$(
P(PX,PY)))-1)
300 P(OX,OY)=P(PX,PY):P(PX,PY)=0
```

```
310 POKESO+4,33:FORT=1TO30:NEXT:POKESO+4
,0:C=1:FORY=0TO2:FORX=0TO2
320 IFP(X,Y)=0THEN350
330 IFP(X,Y)<>CTHEN360
340 C=C+1
350 NEXT:NEXT
360 POKECX,17:POKECY,23
370 SYSCR:PRINTCHR$(5)"SCORE:";C-1;CHR$(
158);:POKESO+4,0:GOTO200
500 FORBYTE=CRTOCR+9:READA:POKEBYTE,A:NE
XT
510 FORT=SOTOSO+24:POKET,0:NEXT:POKESO+2
4,6:POKESO+5,15:POKESO+6,15
520 POKESO+1,100:RETURN
530 DATA24,174,10,192,172,11,192,76,240,
255
540 DATA176,195,195,174,17,157,157,157,1
57,221,32,32,221,17,157,157,157,157
550 DATA173,195,195,189,145,157,157,157
```

```
1 REM       *** 4 BY 4 PUZZLER ***
2 REM
100 SO=54272:DIMP(3,3),A(16):CR=49152:CY
=CR+10:CX=CY+1:GOSUB500
110 POKE53280,0:POKE53281,0:PRINTCHR$(14
7);CHR$(5);CHR$(17)
120 FORT=1TO26:READK:B$=B$+CHR$(K):NEXT
130 PRINT:PRINTTAB(8)"N U M B E R   P U
Z Z L E";CHR$(158)
140 FORY=0TO3:FORX=0TO3
150 T=INT(RND(1)*16):IFA(T+1)=1THENGOTO1
50
160 POKECX,11+5*X:POKECY,7+4*Y:SYSCR
170 PRINTB$;:IFT<>0THENPRINTRIGHT$(STR$(
T),LEN(STR$(T))-1):GOTO190
180 PRINTCHR$(18)"  "CHR$(146):PX=X:PY=Y
190 A(T+1)=1:P(X,Y)=T:NEXT:NEXT:GOTO310
200 S=PEEK(56321)AND15:OX=PX:OY=PY
220 IFS=14THENPY=PY-1:IFPY<0THENPY=0:GOT
0200
230 IFS=13THENPY=PY+1:IFPY>3THENPY=3:GOT
0200
240 IFS=11THENPX=PX-1:IFPX<0THENPX=0:GOT
0200
250 IFS=7THENPX=PX+1:IFPX>3THENPX=3:GOTO
200
260 IFPX=OXANDPY=OYTHEN200
270 POKECX,11+PX*5:POKECY,7+PY*4:SYSCR:P
RINTB$;CHR$(18)"  "CHR$(146)
```

```
280 POKECX,11+OX*5:POKECY,7+OY*4:SYSCR:P
RINTB$;
290 PRINTRIGHT$(STR$(P(PX,PY)),LEN(STR$(
P(PX,PY)))-1)
300 P(OX,OY)=P(PX,PY):P(PX,PY)=0
310 POKE50+4,33:FORT=1TO30:NEXT:POKE50+4
,0:C=1:FORY=0TO3:FORX=0TO3
320 IFP(X,Y)=0THEN350
330 IFP(X,Y)<>CTHEN360
340 C=C+1
350 NEXT:NEXT
360 POKECX,17:POKECY,23
370 SYSCR:PRINTCHR$(5)"SCORE:";C-1;CHR$(
157);CHR$(158)" ":POKE50+4,0:GOTO200
500 FORBYTE=CRTOCR+9:READA:POKEBYTE,A:NE
XT
510 FORT=50TO50+24:POKET,0:NEXT:POKE50+2
4,6:POKE50+5,15:POKE50+6,15
520 POKE50+1,100:RETURN
530 DATA24,174,10,192,172,11,192,76,240,
255
540 DATA176,195,195,174,17,157,157,157,1
57,221,32,32,221,17,157,157,157,157
550 DATA173,195,195,189,145,157,157,157
```

There is a slight problem with this puzzle. Not all mixes of numbers can be unscrambled. There are some positions that are impossible to move into consecutive order. Sam Lloyd, one of the great puzzle inventors of all time, discovered several such positions in the 1870's and in one of his books describes how he turned this impossibility into a new challenge:

Older inhabitants of Puzzleland will remember how in the seventies I drove the entire world crazy over a little box of movable blocks which became known as the "14-15 Puzzle." The fifteen blocks were arranged in the square box in regular order, but with the 14 and 15 reversed as shown in the above illustration. The puzzle consisted of moving the blocks about, one at a time, to bring them back to the present position in every respect except that the error in the 14 and 15 was corrected.

A prize of $1,000, offered for the first correct solution to the problem, has never been claimed, although there are thousands of persons who say they performed the required feat.

People became infatuated with the puzzle and ludicrous tales are told of shopkeepers who neglected to open their stores; of a distinguished clergyman who stood under a street lamp all through a wintry night trying to recall the way he had performed the feat. The mysterious feature of the puzzle is that none seem to be able to remember the sequence of moves whereby they feel sure they succeeded in solving the puzzle. Pilots are said to have wrecked their ships, and engineers rush their trains past stations. A famous Baltimore editor tells how he went for his noon lunch and was discovered by his frantic staff long past midnight pushing little pieces of pie around on a plate! Farmers are known to have deserted their plows, and I have taken one such instance as an illustration for the sketch.

Several new problems which developed from the original puzzle are worth giving:

Second Problem—Start again with the blocks as shown in the large illustration and move them so as to get the numbers in regular order, but with the vacant square at upper left-hand corner instead of lower right-hand corner (see Fig. 1).

Third Problem—Start with the blocks as before, turn the box a quarter-way round and move the blocks until they rest as in Fig. 2.

Fourth Problem—Start as before, then shift the pieces until they form a "magic square," the numbers adding to thirty along all vertical and horizontal rows, and the two diagonals.

	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

Fig. 1

4	8	12	
3	7	11	15
2	6	10	14
1	5	9	13

Fig. 2

Following are the answers to Lloyd's other challenges, taken from Dover Books' *Mathematical Puzzles of Sam Lloyd*, edited by Martin Gardner whose commentary provides the answers to Lloyd's challenges.

[The original puzzle is impossible to solve except by such skullduggery as turning the 6 and 9 blocks upside down. One of the

puzzle's peculiarities is that any such interchange, involving two blocks, immediately converts the puzzle to a solvable one. In fact, any odd number of interchanges has the same effect, whereas an even number leaves the puzzle unsolvable as before. Readers interested in learning something about the interesting mathematical structure underlying this puzzle are referred to the classic analysis by W. W. Johnson and W. E. Story in their article, "Notes on the 15-Puzzle," *American Journal of Mathematics*, Vol. 2, 1879, p. 397f, and to briefer discussions of the puzzle in standard references on recreational mathematics.—M.G.]

The other three problems are solved as follows:

Fig. 1 can be reached in 44 moves: 14, 11, 12, 8,7,6, 10, 12, 8, 7, 4, 3, 6, 4, 7, 14, 11, 15, 13, 9, 12, 8, 4, 10, 8, 4, 14, 11, 15, 13, 9, 12, 4, 8, 5, 4, 8, 9, 13, 14, 10, 6, 2, 1.

Fig. 2 can be reached in 39 moves: 14, 15, 10, 6, 7, 11, 15, 10, 13, 9, 5, 1, 2, 3, 4, 8, 12, 15, 10, 13, 9, 5, 1, 2, 3, 4, 8, 12, 15, 14, 13, 9, 5, 1, 2, 3, 4, 8, 12.

The magic square can be produced in 50 moves: 12, 8, 4, 3, 2, 6, 10, 9, 13, 15, 14, 12, 8, 4, 7, 10, 9, 14, 12, 8, 4, 7, 10, 9, 6, 2, 3, 10, 9, 6, 5, 1, 2, 3, 6, 5, 3, 2, 1, 13, 14, 3, 2, 1, 13, 14, 3, 12, 15, 3.

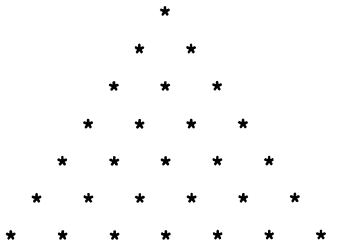
Try to figure out what characterizes an impossible challenge and certainly don't quit on the puzzles too soon. Something that may initially look impossible may merely be difficult.

23

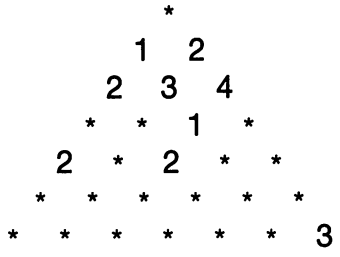
Dozo

Dozo is a game that Japanese GO masters use to warm up. It is deceptively simple and involves a different sort of strategy than usually adopted in the more aggressive games of chess and checkers. Dozo requires that you play to not lose rather than play to win. This will become clear from a description of the board, pieces, and goals of the game.

Dozo is played on a board that is an equilateral triangle. In our version of the game, each side is seven units long. Here is a picture of the board:



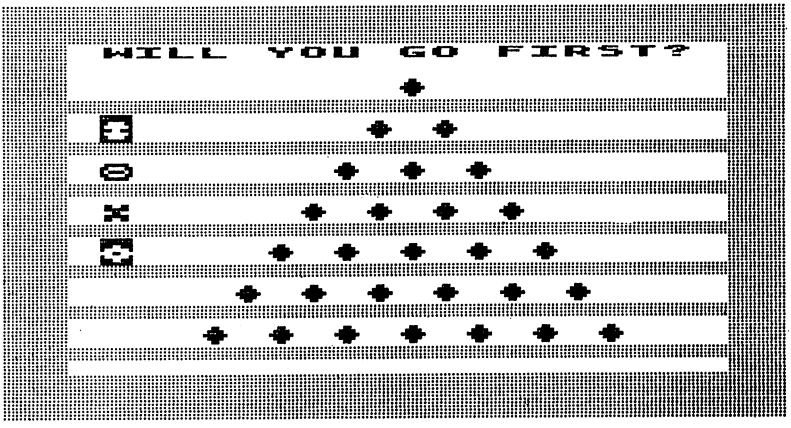
There are four different pieces that you can use in the game. As opposed to most Western games, the pieces do not belong to any one of the players. On each move of the game, you can choose any one of the pieces and place it on any empty point on the board. The goal of the game is to be the first player to make an equilateral triangle of any size with the three vertices of the triangle having the same piece. Suppose the pieces were called 1, 2, 3, and 4. Here is a sample game winning position:

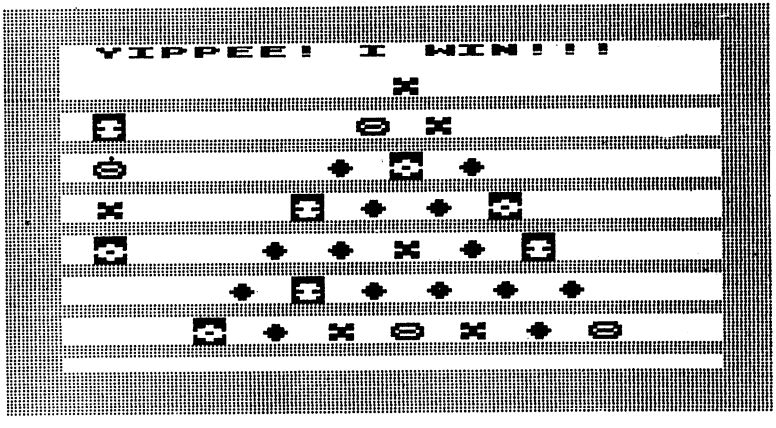


Notice that the 2's in lines 3 and 5 make up an equilateral triangle. The player who placed the last 2 on the board is the winner. The player who placed the second 2 on the board trapped him- or herself into a losing position. The complexity of this game emerges from the fact that you always have to think about not making a move that will give your opponent an equilateral triangle. This involves analyzing positions that won't lead to a loss as well as looking out for winning

positions. Simple and direct aggression will almost always lead to a loss in this game.

The Dozo program is the longest in this book and the most tedious to copy because it turns the computer into your opponent. Strategy for playing a good game of Dozo is built into the program and the computer is pretty hard to beat. However, the computer is not guaranteed a win and no one has yet analyzed Dozo fully enough to know what the best possible move is in any situation. Try your hand at crossing up the computer. It will take time and concentration. You will need a joystick to play this game.





```
10 REM        *** DOZO ***
100 CR=49152:CY=CR+10:CX=CY+1:GOSUB9000:
POKE53280,2:POKE53281,0
110 PRINTCHR$(147);CHR$(5):GOTO1000
500 POKECX,H:POKECY,V:SYSCR:RETURN
1000 GOSUB8000:IFC<>17076THEN10000
2000 POKE198,0:GOSUB6000:GOSUB6200
2100 GOSUB6300:IFFP=1THEN3000
2110 H=1:V=2:GOSUB500:PRINT"MY TURN..."
2120 IFCC(P)<=10RCC(P)=7THEN2400
2130 FORT=0TOCC-1:M1=M(P,CC):M2=M(P,T):F
1=M1:F2=M2:IFM1>M2THENF1=M2:F2=M1
2140 B=B(F1)+(F2-(F1+1))*2:P1=A(B):P2=A(
B+1)
2150 IFP1<=27THENIFS(P1)=45THENW1=P1:GOT
02300
2160 IFP2<=27THENIFS(P2)=45THENW1=P2:GOT
02300
2170 NEXT:GOTO2400
2300 GOSUB6300:V=2:H=12:GOSUB500:PRINT"Y
IPPEE! I WIN!!!":X=X(SX(W1),SY(W1))
2310 Y=Y(SX(W1),SY(W1)):FORT=1TO2:H=X:V=
Y:GOSUB500:PRINTT$:FORJ=1TO120:NEXT
2320 GOSUB500:PRINTJ$:FORJ=1TO80:NEXT:NE
XT:GOSUB500:PRINTT$:X1=X(SX(F1),SY(F1))
2330 Y1=Y(SX(F1),SY(F1)):X2=X(SX(F2),SY(
F2)):Y2=Y(SX(F2),SY(F2)):FORT=1TO5
2340 H=X:V=Y:GOSUB500:PRINTJ$:H=X1:V=Y1:
GOSUB500:PRINTJ$:H=X2:V=Y2:GOSUB500
2350 GOSUB500:PRINTJ$:FORJ=0TO80:NEXT:H=
X:V=Y:GOSUB500:PRINTT$:H=X1:V=Y1
2360 GOSUB500:PRINTT$:H=X2:V=Y2:GOSUB500
:PRINTT$:FORJ=1TO120:NEXT:NEXT
2370 FORT=1TO7500:NEXT:GOTO2000
2400 P=0:IFCC(1)<CC(P)THENP=1
2410 IFCC(2)<CC(P)THENP=2
2420 IFCC(3)<CC(P)THENP=3
2430 EP=P:PC=-1
2440 TT=0:TC=-1
2500 IFTT<10THEN2520:REM POP WAS HERE
2505 TC=TC+1:T=TC:IFT>27THEN6400
2510 IFS(T)<>45THEN2505
2515 GOTO2535
2520 T=INT(RND(1)*28):IFS(T)<>45THEN2520
2525 IFCC(P)=0THEN2600
2530 TT=TT+1
2535 FORJ=0TOCC(P)-1:M1=M(P,J):M2=T:F1=M
1:F2=M2:IFM1>M2THENF1=M2:F2=M1
2540 B=B(F1)+(F2-(F1+1))*2:P1=A(B):P2=A(
B+1)
2545 IFP1<=27THENIFS(P1)=45THEN2500
```

```
2550 IFP2<=27THENIFS(P2)=45THEN2500
2555 NEXT
2600 T$=P$(P)
2610 X=X(SX(T),SY(T)):Y=Y(SX(T),SY(T))
2620 FORK=1TO5:H=X:V=Y:GOSUB500:PRINTT$:
FORQ=1TO50:NEXT:GOSUB500:PRINTJ$
2630 FORQ=1TO50:NEXT:NEXT:GOSUB500:PRINT
T$
2640 S(T)=ASC(T$):PX(T)=X(SX(T),SY(T)):P
Y(T)=Y(SX(T),SY(T)):M(P,CC(P))=T
2650 CC(P)=CC(P)+1
2700 FP=1:GOTO2100
3000 H=1:V=2:GOSUB500:PRINT"YOUR TURN...
"
3100 IF(PEEK(56321)AND16)=0ANDCC(P)<7THE
NX=0:Y=0:GOSUB6100:GOTO3500
3110 Z=PEEK(56321)AND15:IFZ=14THENP=P-1:
IFP<OTHENP=3
3120 IFZ=13THENP=P+1:IFP>3THENP=0
3130 H=3:V=P*3+6:GOSUB500:PRINT" "
3140 T$=P$(P)
3150 FORT=0TO10:IF(PEEK(56321)AND16=0)TH
ENGOSUB500:PRINTT$:GOTO3100
3160 NEXT:GOSUB500:PRINTT$:FORT=0TO10:IF
(PEEK(56321)AND16)=0THEN3100
3170 NEXT:GOTO3100
3500 IF(PEEK(56321)AND16)=0THENGOSUB6100
:GOTO3800
3510 Z=(PEEK(56321)AND15):OX=X:OY=Y
3520 IFZ=14THENX=X-1:Y=Y-2
3530 IFZ=13THENX=X+1:Y=Y+2
3540 IFZ=7THENX=X+1
3550 IFZ=11THENX=X-1
3560 IFZ=6THENY=Y-1
3570 IFZ=9THENY=Y+1
3580 IFZ=5THENX=X+1:Y=Y+1
3590 IFZ=10THENX=X-1:Y=Y-1
3600 IFX<00RY>60RX>YTHENX=OX:Y=OY
3610 SS=P(X,Y):J$=CHR$(S(SS)):XS=X(X,Y):
YS=Y(X,Y)
3620 H=XS:V=YS:GOSUB500:PRINT" ":FORT=0T
O10:NEXT:GOSUB500:PRINTT$
3630 FORT=0TO10:IF(PEEK(56321)AND16)=0TH
ENGOSUB500:PRINTJ$:GOTO3500
3640 NEXT:GOSUB500:PRINTJ$:FORT=0TO10:IF
(PEEK(56321)AND16)=0THEN3500
3650 NEXT:GOTO3500
3800 IFS(SS)<>45THEN3510
3810 S(SS)=ASC(T$):H=XS:V=YS:GOSUB500:PR
INTT$:PX(SS)=XS:PY(SS)=YS
3820 M(P,CC(P))=SS:CC=CC(P):CC(P)=CC(P)+
1:IFCC(P)<3THENFP=0:GOTO2100
```

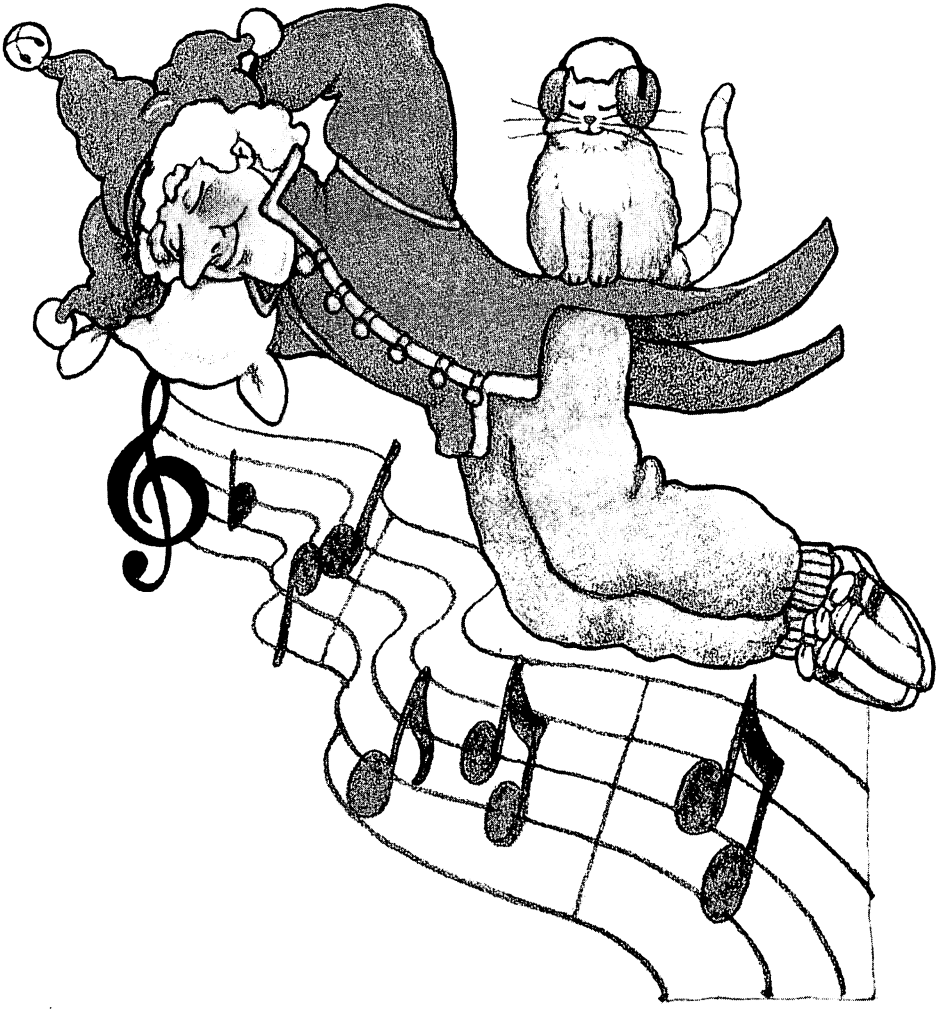
```
3830 FORT=0TOCC-2:M1=M(P,CC):M2=M(P,T):F
1=M1:F2=M2:IFM1>M2THENF1=M2:F2=M1
3840 B=B(F1)+(F2-(F1+1))*2:P1=A(B):P2=A(
B+1)
3850 FORQ=T+1TOCC-1:IFP1=M(P,Q)ORP2=M(P,
Q)THEN4000
3860 NEXT:NEXT:FP=0:GOTO2100
4000 GOSUB6300
4010 V=2:H=11:GOSUB500:PRINT"YOU WON THE
 GAME!"
4020 FORT=1TO7500:NEXT:GOTO2000
6000 PRINTCHR$(147):FORX=0TO6:FORY=0TO6:
IFY<XTHENNEXT
6010 H=X(X,Y):V=Y(X,Y):GOSUB500:PRINT"-"
:NEXT:NEXT:FORT=0TO3:H=3:V=T*3+6
6020 GOSUB500:PRINTP$(T):NEXT:FORT=0TO27
:S(T)=45:NEXT:FORT=0TO3:CC(T)=0:NEXT
6030 RETURN
6100 IF(PEEK(56321)AND16)=0THEN6100
6110 RETURN
6200 V=2:H=11:GOSUB500:PRINT"WILL YOU GO
 FIRST?"
6210 GETA$:IFA$=""THEN6210
6220 IFA$="N"THENFP=0:RETURN
6230 IFA$="Y"THENFP=1:RETURN
6240 GOTO6210
6300 H=0:V=2:GOSUB500:PRINT"
                          ":RETURN
6400 PC=PC+1:P=PC:IFP<4THEN2440
6410 T=INT(RND(1)*28):IFS(T)<>45THEN6410
6420 P=EP:GOTO2600
8000 DIMA(756),S(27),PX(27),PY(27),B(26)
,P(6,6),X(6,6),Y(6,6),M(3,6)
8010 J$="-":PRINTCHR$(145):DIMSX(27),SY(
27):FORX=0TO6:FORY=0TO6:IFY<XTHENNEXT
8020 X(X,Y)=20-Y*2+4*X:Y(X,Y)=Y*3+4:NEXT
:NEXT:FORT=0TO3:READK:P$(T)=CHR$(K)
8030 NEXT:T=0:FORY=0TO6:FORX=0TO6:IFX>YT
HENNEXTY
8040 P(X,Y)=T:SX(T)=X:SY(T)=Y:T=T+1:NEXT
:NEXT:V=10:H=14:GOSUB500:FORT=0TO26
8050 READB(T):NEXT:PRINT"PLEASE WAIT":C=
0:FORT=1TO756:READA(T):C=C+A(T):NEXT
8060 RETURN
9000 FORBYTE=CRTOCR+9:READA:POKEBYTE,A:N
EXT:RETURN
9100 DATA24,174,10,192,172,11,192,76,240
,255
9300 DATA119,123,42,118
9400 DATA1,55,107,157,205,251,295,337,37
7,415,451,485,517,547,575,601,625,647
```

```
9410 DATA667,685,701,715,727,737,745,751
,755
9500 DATA2,32,1,32,5,32,32,32,3,32,9,32,
32,32,32,32,6,32,14,32
9510 DATA32,32,32,32,32,32,10,32,20,32,3
2,32,32,32,32,32,32,32,15,32
9520 DATA27,32,32,32,32,32,32,32,32,32,3
2,32,21,32,0,4,4,32,2,3
9530 DATA7,32,8,32,5,32,6,32,11,32,13,32
,9,32,32,32,10,32,16,32
9540 DATA19,32,14,32,32,32,32,32,15,32,2
2,32,26,32,20,32,32,32,32,32
9550 DATA32,32,21,32,32,32,8,32,1,5,4,32
,13,32,9,32,3,32,7,32
9560 DATA19,32,14,32,32,32,6,32,11,32,26
,32,20,32,32,32,32,32,10,32
9570 DATA16,32,32,32,27,32,32,32,32,32,3
2,32,15,32,22,32,1,7,0,12
9580 DATA7,32,4,6,2,11,17,32,12,32,8,32,
5,10,16,32,23,32,18,32
9590 DATA13,32,9,32,15,32,22,32,32,32,25
,32,19,32,14,32,32,32,21,32
9600 DATA32,32,32,32,2,8,12,32,3,8,5,7,1
2,32,18,32,13,32,6,9
9610 DATA11,32,17,32,25,32,19,32,14,32,1
0,32,16,32,23,32,32,32,26,32
9620 DATA20,32,32,32,15,32,22,32,32,32,1
8,32,1,13,4,9,8,32,25,32
9630 DATA19,32,3,14,7,32,12,32,32,32,26,
32,20,32,6,32,11,32,17,32
9640 DATA32,32,32,32,27,32,32,32,10,32,1
6,32,23,32,3,11,1,17,0,24
9650 DATA11,32,7,10,4,16,2,23,32,32,17,3
2,12,32,8,15,5,22,32,32
9660 DATA32,32,24,32,18,32,13,32,9,21,32
,32,32,32,32,32,4,12,2,18
9670 DATA17,32,6,12,8,11,5,17,24,32,24,3
2,18,32,10,13,9,16,23,32
9680 DATA32,32,32,32,25,32,19,32,14,15,2
2,32,32,32,32,32,5,13,24,32
9690 DATA3,18,7,13,9,12,18,32,32,32,25,3
2,6,19,11,14,17,32,24,32
9700 DATA32,32,32,32,26,32,10,20,16,32,2
3,32,32,32,32,32,1,25,4,19
9710 DATA8,14,13,32,32,32,2,32,3,26,7,20
,12,32,18,32,32,32,32,32
9720 DATA32,32,6,27,11,32,17,32,24,32,6,
16,3,23,1,32,0,32,16,32
9730 DATA11,15,7,22,4,32,32,32,32,32,23,
32,15,16,12,21,8,32,5,32
9740 DATA32,32,32,32,7,17,4,24,2,32,23,3
2,10,17,12,16,8,23,5,32
```

```
9750 DATA32,32,32,32,24,32,15,18,13,22,9
,32,32,32,32,32,8,18,5,25
9760 DATA32,32,6,24,11,18,13,17,9,24,32,
32,32,32,32,32,10,25,16,19
9770 DATA14,23,32,32,32,32,9,19,32,32,3,
32,7,25,12,19,14,18,25,32
9780 DATA32,32,32,32,6,32,11,26,17,20,24
,32,32,32,32,32,1,32,4,32
9790 DATA8,26,13,20,19,32,32,32,32,32,3,
32,7,32,12,27,18,32,25,32
9800 DATA10,22,6,32,3,32,1,32,0,32,22,32
,16,21,11,32,7,32,4,32
9810 DATA2,32,32,32,11,23,7,32,4,32,2,32
,32,32,15,23,17,22,12,32
9820 DATA8,32,5,32,32,32,12,24,8,32,5,32
,32,32,10,32,16,24,18,23
9830 DATA13,32,9,32,32,32,13,25,9,32,32,
32,6,32,11,32,17,25,19,24
9840 DATA14,32,32,32,14,26,32,32,3,32,7,
32,12,32,18,26,20,25,32,32
9850 DATA32,32,1,32,4,32,8,32,13,32,19,2
7,26,32,15,32,10,32,6,32
9860 DATA3,32,1,32,0,32,16,32,11,32,7,32
,4,32,2,32,17,32,12,32
9870 DATA8,32,5,32,18,32,13,32,9,32,19,3
2,14,32,20,32
10000 PRINT"   INCORRECT DATA IN LINES 9
500-9870":END
```

PART V

Music and Noise

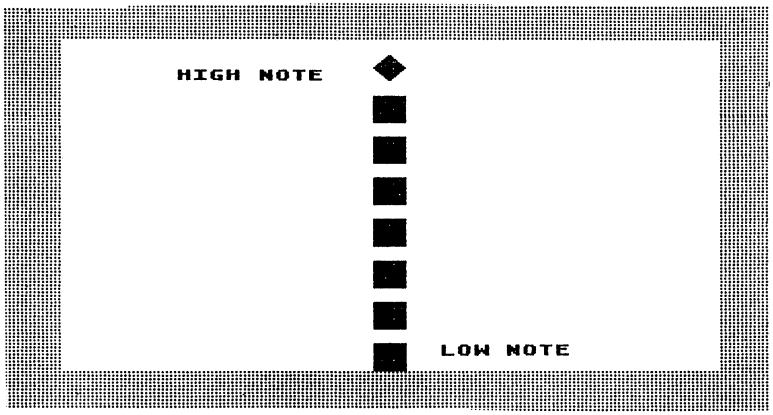


Your Commodore 64 has a powerful music function with a nine octave range. This section contains several music programs that take advantage of this power.

24

High Note Low Note

In this game you use a joystick to rearrange a series of notes from high to low. Eight notes are scrambled and hidden under squares on the screen.



By pushing the joystick forward or backward, you move up and down the musical scale. Press the button to pick up a note in position and carry it to another position. Press the button again to drop the note. Put the lowest note at the bottom of the screen and the highest at the top. The computer will let you know when you have succeeded in ordering the notes.

```
10 REM  *** HIGH NOTE LOW NOTE ***
100 SO=54272:POKESO+24,15:CR=49152:CY=CR
+10:CX=CY+1:GOSUB900
110 POKE53280,8:POKE53281,2:PRINTCHR$(14
7);CHR$(5)
120 FORT=1TO8:READK:B$=B$+CHR$(K):NEXT:F
ORT=1TO9:READK:D$=D$+CHR$(K):NEXT
130 FORT=1TO7:READK:C$=C$+CHR$(K):NEXT:F
ORT=0TO7:P(T)=-1:NEXT:FORT=0TO7
140 R=INT(RND(1)*8):IFP(R)>-1THEN140
```

```
150 P(R)=T:NEXT:O=INT(RND(1)*4)*8
160 X=7:Y=2:GOSUB800:PRINT"HIGH NOTE":FO
RT=0TO7:X=19:Y=1+T*3:GOSUB800:PRINTB$
170 S=M(P(T)+O):GOSUB500:NEXT:X=23:Y=22:
GOSUB800:PRINT"LOW NOTE":GOSUB600
180 IF(PEEK(56321)AND16)=0THEN400
185 GETA$:IFA$=CHR$(133)ANDW=1THENRUN
190 POKE198,0:IF(PEEK(56321)AND15)=15THE
N180
200 KY=YY:Z=(PEEK(56321)AND15):IFZ=14THE
NYY=YY-1:IFYY<0THENYY=0
210 IFZ=13THENYY=YY+1:IFYY>7THENYY=7
220 IFYY=KYTHEN180
230 X=19:Y=1+KY*3:GOSUB800:PRINTB$:GOSUB
600:GOTO180
400 NN=1-NN:KY=YY:GOSUB600
410 IF(PEEK(56321)AND16)=0THEN410
420 GOTO180
500 POKESO+1,S/256:POKESO,S-INT(S/256)*2
56:POKESO+4,17:FORQ=1TO100:NEXT
510 POKESO+4,0:RETURN
600 X=19:Y=1+YY*3:GOSUB800:IFNN=1THENPRI
NTC$:T=P(YY):P(YY)=P(KY):P(KY)=T
610 IFNN=0THENPRINTD$
620 S=M(P(YY)+O):GOSUB500:C=0:FORT=0TO7:
IFP(T)=CTHENC=C+1:NEXT:GOTO700
630 GOTO730
700 IFNN=1THENRETURN
710 X=4:Y=10:GOSUB800:PRINT"YOU DID IT!"
:X=25:GOSUB800:PRINT"PRESS <F1>":W=1
720 RETURN
730 IFNN=1THENRETURN
740 X=4:Y=10:GOSUB800:PRINT"                "
:X=25:GOSUB800:PRINT"          ":W=0
750 RETURN
800 POKECX,X:POKECY,Y:SYSCR:RETURN
900 FORBYTE=CRTOCR+9:READA:POKEBYTE,A:NE
XT:POKESO+5,0:POKESO+6,240:DIMM(31)
910 FORT=0TO31:READM(T):NEXT:RETURN
920 DATA24,174,10,192,172,11,192,76,240,
255
930 DATA8583,8101,7217,6430,5728,5407,48
17,4291
940 DATA7217,6430,5728,5407,4817,4291,40
50,3608
950 DATA6430,6069,5407,4817,4291,4050,36
08,3215
960 DATA5407,4817,4291,4050,3608,3215,30
34,2703
970 DATA18,32,32,157,157,17,32,32
980 DATA18,169,127,157,157,17,146,127,16
9
```

```
990 DATA110,109,157,157,17,109,110
1000 DATA18,169,127,157,157,17,146,127,1
69
1010 DATA110,109,157,157,17,109,110
```

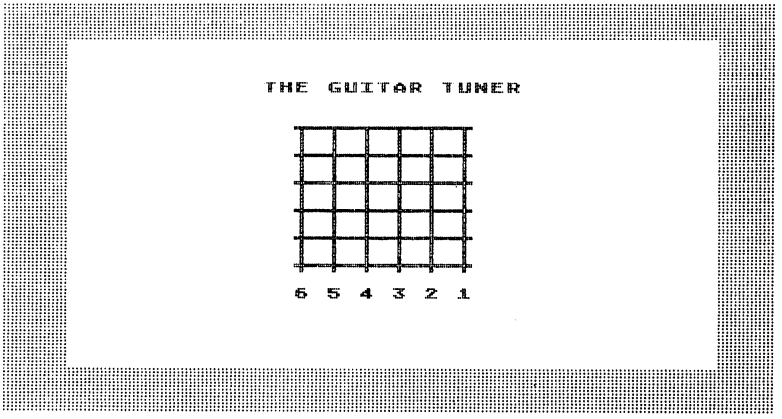
Here are the DATA statements used in the program. Notice that each line has eight numbers representing the eight notes used in the game and also that these numbers are ordered from low to high. You can change these statements and put your own scales or intervals in their place. Be sure, however, to keep eight notes on a line and order them from lowest to highest. Your reference manual will give you note/number equivalents.

```
920 DATA24,174,10,192,172,11,192,76,240,
255
930 DATA8583,8101,7217,6430,5728,5407,48
17,4291
940 DATA7217,6430,5728,5407,4817,4291,40
50,3608
950 DATA6430,6069,5407,4817,4291,4050,36
08,3215
960 DATA5407,4817,4291,4050,3608,3215,30
34,2703
970 DATA18,32,32,157,157,17,32,32
980 DATA18,169,127,157,157,17,146,127,16
9
990 DATA110,109,157,157,17,109,110
1000 DATA18,169,127,157,157,17,146,127,1
69
1010 DATA110,109,157,157,17,109,110
```

25

The Guitar Tuner

You can use this program to tune your guitar. The screen presents you with a picture of a grid indicating the six strings on a guitar. Just press the number corresponding to any string to turn on that tone. When you want to turn off the tone, press any key.




```
10 REM   *** GUITAR TUNER ***
100 PRINTCHR$(147):PRINT:SO=54272:POKESO
+24,15:POKESO+5,0:POKESO+6,240:DIMS(1,5)
110 FORU=0TO5:FORT=0TO1:READS(T,U):NEXT:
NEXT:PRINTTAB(12)"THE GUITAR TUNER"
120 FORT=1TO11:READG1,G2,G3:G1$=G1$+CHR$
(G1):G2$=G2$+CHR$(G2):G3$=G3$+CHR$(G3)
130 NEXT:PRINT:PRINT:PRINTTAB(14)G1$:FOR
T=1TO5:PRINTTAB(14)G2$:PRINTTAB(14)G3$
140 NEXT:PRINT:PRINTTAB(14)"6 5 4 3 2 1"
:POKESO+4,17:GOSUB250
150 GETA$:IFA$=""THEN150
160 A=7-VAL(A$):GOSUB250:IFA<1ORA>6THEN1
50
170 POKESO,S(0,A-1):POKESO+1,S(1,A-1):PR
INTCHR$(19):PRINT:PRINT:PRINT
180 PRINTTAB(12+A*2)CHR$(113):GOTO150
```

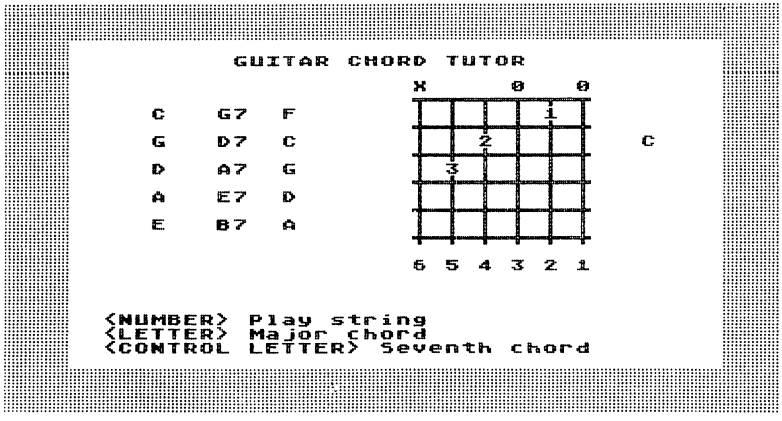
```
250 PRINTCHR$(19):PRINT:PRINT:PRINT:PRIN
TTAB(14)"                    "
260 POKE50,0:POKE50+1,0:RETURN
500 DATA143,10,32,14,205,18,30,25,165,31
,62,42
510 DATA178,194,123,99,32,99
520 DATA178,194,123,99,32,99
530 DATA178,194,123,99,32,99
540 DATA178,194,123,99,32,99
550 DATA178,194,123,99,32,99
560 DATA178,194,123
```

26

Guitar Chord Tutor

Once you've got your guitar tuned up, you might want a little help with mastering guitar chords. This program will give you the finger-

ing, name, and sound of the chords most commonly used on the guitar.



The Guitar Tutor program displays the top frets of a guitar finger board and the most used chords in five keys. Type the letter corresponding to the chord you want to learn. Press **CONTROL** plus a letter to select a major seventh chord. Press the numbers one through six to hear each note of the chord. The numbers on the frets tell you which fingers to use in forming the chord.

```
10 REM  *** GUITAR CHORDS TUTOR ***
100 DIMS(5,3),A(1,6):SO=54272:CR=49152:C
Y=CR+10:CX=CY+1:GOSUB900:POKE53280,0
110 DIMT$(10,1,5):POKE53281,0:PRINTCHR$(
147);CHR$(5):FORT=0TO10:FORU=0TO5
120 FORV=0TO1:READT$(T,V,U):NEXT:NEXT:NE
XT
130 FORT=0TO5:FORJ=0TO3:READS(T,J):NEXT:
NEXT:FORT=1TO11:READK,L,M:A$=A$+CHR$(K)
140 B$=B$+CHR$(L):C$=C$+CHR$(M):NEXT:XX=
21:YY=4:FORJ=0TO1:FORT=0TO6:READA(J,T)
150 NEXT:NEXT:X=10:Y=1:GOSUB800:PRINT"GU
ITAR CHORD TUTOR":FORT=0TO4:FORJ=0TO2
160 X=5+J*4:Y=5+T*2:GOSUB800:READT$:PRIN
TT$:NEXT:NEXT:X=2:Y=20:GOSUB800
170 PRINT"<NUMBER> PLAY STRING":PRINT"
<LETTER> MAJOR CHORD"
180 PRINT"  <CONTROL LETTER> SEVENTH CHO
RD":A=67:GOTO300
```

```
200 GOSUB700:FORT=0TO5:P=VAL(T$(FF*6+A(F
F,A),0,T)):T$=T$(FF*6+A(FF,A),1,T)
210 C(T)=P:P=-(P<>-1)*P:X=XX+10-T*2:Y=YY
-1+P*2:GOSUB800:PRINTT$:NEXT
220 GETQ$:IFQ$=""THEN220
230 A=ASC(Q$):IFA<49ORA>54THEN300
240 A=A-49:GOSUB750:X=XX+10-A*2:Y=YY+14:
GOSUB800:PRINTCHR$(113)
250 IFC(A)>-1THENGOSUB600
260 GOTO220
300 IFA>0ANDA<8THEN350
310 IFA<65ORA>71THEN220
320 A=A-65:IFA=1THEN220
330 X=35:Y=7:GOSUB800:PRINTCHR$(65+A);"
":FF=0:GOTO200
350 A=A-1:IFA=2ORA=5THEN220
360 X=35:Y=7:GOSUB800:PRINTCHR$(65+A);"7
":FF=1:GOTO200
600 S=S(A,C(A)):POKESO+1,S/256:POKESO,S-
INT(S/256)*256
610 POKESO+4,17:FORT=1TO250:NEXT:POKESO+
4,0:RETURN
700 X=XX:Y=YY-1:GOSUB800:PRINT"
  ":Y=YY:GOSUB800:PRINTA$
710 FORT=1TO10STEP2:Y=YY+T:GOSUB800:PRIN
TB$:Y=Y+1:GOSUB800:PRINTC$:NEXT:Y=Y+2
720 GOSUB800:PRINT"6 5 4 3 2 1"
750 X=XX:Y=YY+14:GOSUB800:PRINT"
   ":RETURN
799 STOP
800 POKECX,X:POKECY,Y:SYSCR:RETURN
900 FORBYTE=CRTOCR+9:READA:POKEBYTE,A:NE
XT:POKESO+5,9:POKESO+6,0:POKESO+24,15
910 RETURN
920 DATA24,174,10,192,172,11,192,76,240,
255
930 DATA0,0,2,4,2,3,2,2,0,0,0,0,0,0,1,1,
0,0,2,2,3,3,-1,X
940 DATA2,2,3,3,2,1,0,0,0,0,-1,X,0,0,0,0
,1,1,2,3,2,2,0,0
950 DATA1,1,1,1,2,2,3,4,3,3,1,1,3,3,0,0,
0,0,0,0,2,1,3,2
960 DATA0,0,2,3,0,0,2,2,0,0,0,0,2,4,0,0,
2,3,1,1,2,2,-1,X
970 DATA2,3,1,1,2,2,0,0,0,0,-1,X,0,0,3,4
,1,1,2,3,2,2,0,0
980 DATA1,1,0,0,0,0,0,0,2,2,3,3
990 DATA10814,11457,12139,12860,8101,858
3,9094,9634,6430,6812,7217,7647,4817
1000 DATA5103,5407,5728,3608,3823,4050,4
291,2703,2864,3034,3215
1010 DATA178,125,123,99,32,99,178,125,12
```

```
3,99,32,99,178,125,123,99,32,99,178,125
1020 DATA123,99,32,99,178,125,123,99,32,
99,178,125,123
1030 DATA0,0,1,2,3,4,5,0,1,0,2,3,0,4
1040 DATAC,G7,F,G,D7,C,D,A7,G,A,E7,D,E,B
7,A
```

27, 28

Music Match and Noise Match

Here are two matching games involving sound. In the Music Match game you are presented with a grid with the letters from A to P printed in order in the squares. When you press a letter, you will hear a tune. Each tune is repeated twice on the grid. The goal of this two-player game is to press two letters on your turn that play the same tune. The game provides good ear training. You have to remember tunes and match them up. Each player presses two letters per turn. Whenever someone makes a match, however, they get another chance. Here's a picture of the grid followed by the game program:

```
10 REM  *** MUSIC CONCENTRATION ***
100 SO=54272:CR=49152:CY=CR+10:CX=CY+1:D
IMJ(16),A(16),X(15),Y(15):PL=1:GOSUB900
110 POKE53280,6:POKE53281,6:PRINTCHR$(14
7);CHR$(5):GOTO400
200 ONR+1GOSUB250,260,270,280,290,300,31
0,320
210 POKESO+1,0:POKESO+4,17
220 POKESO+1,N(MU)/256:POKESO,N(MU)-INT(
N(MU)/256)*256:FORU=1TON(MU+1)*50:NEXT
230 POKESO+4,0:MU=MU+2:IFN(MU)>0THEN210
240 RETURN
250 MU=00:RETURN
260 MU=39:RETURN
270 MU=70:RETURN
280 MU=91:RETURN
290 MU=112:RETURN
300 MU=153:RETURN
310 MU=184:RETURN
320 MU=215:RETURN
400 FORT=0TO35:READK:A$=A$+CHR$(K):NEXT:
A1$=LEFT$(A$,9):A2$=MID$(A$,10,9)
410 A3$=MID$(A$,19,9):L=0:FORY=0TO3:FORX
=0TO3:X(L)=16+X*2:Y(L)=7+Y*2:L=L+1:NEXT
420 NEXT:X=6:Y=2:GOSUB800:PRINT"*** MUSI
C CONCENTRATION ***":X=15:Y=6
430 GOSUB800:PRINTA1$:FORT=1TO3:PRINTTAB
(15);A2$:PRINTTAB(15);A3$:NEXT
440 PRINTTAB(15);A2$:PRINTTAB(15);RIGHT$
(A$,9):FORT=0TO15:X=X(T):Y=Y(T):GOSUB800
450 PRINTCHR$(T+65):NEXT:FORT=0TO15:A(T)
=-1:NEXT:FORT=0TO7:FORJ=0TO1
460 R=INT(RND(1)*16):IFA(R)>-1THEN460
470 A(R)=T:NEXT:NEXT:GOSUB790:X=4:Y=12:G
OSUB800:PRINT"          "
480 PL=1-PL:GOSUB790:X=4:Y=12:GOSUB800:P
RINT"          "
490 IFPL=0THENGOSUB800:PRINT"PLAYER 1"
500 X=27:GOSUB800:PRINT"          ":IFPL=1
THENGOSUB800:PRINT"PLAYER 2"
510 FORC=0TO1
520 GETA$:IFA$<"A"ORA$>"P"THEN520
530 A=ASC(A$)-65:IFC=1THENIFA=K(0)THEN52
0
540 IFJ(A)=1THEN520
550 X=X(A):Y=Y(A):GOSUB800:PRINT"*":R=A(
A):GOSUB200:K(C)=A:R(C)=R:NEXT
560 IFR(0)=R(1)THEN600
570 FORC=0TO1:X=X(K(C)):Y=Y(K(C)):GOSUB8
00:PRINTCHR$(65+K(C)):NEXT:GOTO480
600 FORC=0TO1:X=X(K(C)):Y=Y(K(C)):GOSUB8
00:PRINTCHR$(18);" ":J(K(C))=1:NEXT
```

```
610 PRINT:S(PL)=S(PL)+10:GOSUB790:MM=MM+
1:IFMM<8THENPL=1-PL:GOTO480
620 X=2:Y=17:GOSUB800:IFS(0)=S(1)THENPRI
NT"TIE GAME!":GOTO650
630 T=1:IF S(1)>S(0)THEN T=2
640 PRINT"YOU DID IT PLAYER";T
650 X=12:Y=21:GOSUB800:PRINT"PRESS ANY K
EY..."
660 GETA$:IFA$=""THEN660
670 RUN
789 GOTO789
790 X=4:Y=8:GOSUB800:PRINT"SCORE:";S(0):
X=27:GOSUB800:PRINT"SCORE:";S(1):RETURN
800 POKECX,X:POKECY,Y:SYSCR:RETURN
900 FORBYTE=CRTOCR+9:READA:POKEBYTE,A:NE
XT:POKESO+24,15:POKESO+5,0:POKESO+6,240
910 DIMN(257):FORT=0TO257:READN(T):NEXT:
RETURN
920 DATA24,174,10,192,172,11,192,76,240,
255
1000 DATA6430,2,8583,2,6430,2,7217,2,643
0,1,6430,1,8583,2,6430,2,7217,2,6430,1
1010 DATA6430,1,8583,2,6430,2,7217,2,643
0,1,6430,1,5407,2,4817,2,4291,3,0
1020 DATA4817,2,6430,2,6430,2,6430,2,481
7,2,5407,2,5407,2,4817,2,4817,2,6430,2
1030 DATA6430,2,7217,2,7217,2,8101,4,643
0,4,0
1040 DATA6430,1,6430,1,6430,2,5728,2,540
7,2,5407,2,4817,2,5407,2,5728,4,4817,4,0
1050 DATA4817,4,4817,4,4817,3,5407,1,606
9,4,6069,3,5407,1,6069,3,6430,1
1060 DATA7217,8,0
1070 DATA4817,2,8101,1,8101,1,8101,1,810
1,1,8583,2,8101,2,7217,2,7217,4,4817,1
1080 DATA4817,1,7217,1,7217,1,7217,1,721
7,1,8101,1,8101,1,7217,2,6430,2,6430,4,0
1090 DATA8101,2,6430,4,8101,2,6430,3,643
0,1,5407,1,4817,3,6430,1,6430,1,6430,1
1100 DATA6430,1,8101,1,9634,1,9634,2,963
4,4,0
1110 DATA6430,2,8101,2,8583,2,9634,4,810
1,4,8101,2,6430,2,8101,2,8583,2,9634,4
1120 DATA8101,2,10814,2,9634,4,8101,4,72
17,4,0
1130 DATA6430,4,6430,3,7217,1,8101,4,810
1,4,7217,3,6430,1,7217,3,8101,1,6430,4
1140 DATA4817,4,6430,4,6430,3,7217,1,810
1,4,8101,4,7217,3,6430,1,7217,3,8101,1
1150 DATA6430,4,0
1500 DATA176,99,178,99,178,99,178,99,174
1510 DATA98,32,98,32,98,32,98,32,98
```

```
1520 DATA171,99,123,99,123,99,123,99,179
1530 DATA173,99,177,99,177,99,177,99,189
```

The Noisy Match or, as we've called it, Noisy Concentration uses the same grid as Music Match and is played in the same way. However, it uses the special effects and game sounds like zooms and crashes instead of tunes. It is an easier game to play than Music Match and you might want to begin with it rather than with Music Match.

If you study both of these programs carefully you'll find that the main difference between them is in their DATA statements in line 900 and on. You can copy one of the programs and then, after saving it, modify what you copied to get the other one. That should save some time.

You can also change the tunes and noises used in the matching game by changing the DATA statements. In fact, the READ...DATA command in BASIC can be used to modify many programs without having to copy them from scratch.

```
10 REM  *** NOISY CONCENTRATION ***
100 SO=54272:CR=49152:CY=CR+10:CX=CY+1:D
IMJ(16),A(16),X(15),Y(15):PL=1:GOSUB900
110 POKE53280,0:POKE53281,0:PRINTCHR$(14
7);CHR$(5):GOTO400
150 POKESO+5,0:POKESO+6,240:POKESO+1,0:P
OKESO+4,33:POKES2+1,0:POKES2+5,0
160 POKES2+6,240:POKES2+4,17
200 ONR+1GOSUB220,230,240,250,260,270,28
0,290
210 POKESO+4,0:POKES2+4,0:RETURN
220 FORU=0TO255STEP5:POKESO+1,U:POKES2+1
,255-U:NEXT
225 RETURN
230 FORU=1TO10:Q=INT(RND(1)*150):POKESO+
1,Q+25:POKES2+1,175-Q:FORV=1TO40:NEXT
235 NEXT:RETURN
240 FORU=75TO10STEP-5:POKESO+1,U:POKES2+
1,100-U:FORV=1TOU:NEXT:NEXT
245 RETURN
250 POKES2+4,33:POKESO+1,5:POKES2+1,5:FO
RU=0TO255STEP2
255 POKESO,U:POKES2,255-U:NEXT:RETURN
260 FORU=10TO50STEP10:FORV=UTO75STEP5:PO
KESO+1,V:POKES2+1,U:NEXT:NEXT
265 RETURN
```

```
270 FORU=200TO240:POKES0+1,U:POKES2+1,U-
1:NEXT:FORU=240TO100STEP-4
275 POKES0+1,U:POKES2+1,U+10:NEXT:RETURN
280 FORU=0TO12:FORV=20+UTO25+U:POKES0+1,
V:POKES2+1,V:NEXT:NEXT
285 RETURN
290 FORU=0TO240STEP16:POKES2+1,U:POKES0+
1,U+3:FORV=1TO20:NEXT:NEXT
305 RETURN
400 FORT=0TO35:READK:A$=A$+CHR$(K):NEXT:
A1$=LEFT$(A$,9):A2$=MID$(A$,10,9)
410 A3$=MID$(A$,19,9):L=0:FORY=0TO3:FORX
=0TO3:X(L)=16+X*2:Y(L)=7+Y*2:L=L+1:NEXT
420 NEXT:X=6:Y=2:GOSUB800:PRINT"*** NOIS
Y CONCENTRATION ***":X=15:Y=6
430 GOSUB800:PRINTA1$:FORT=1TO3:PRINTTAB
(15);A2$:PRINTTAB(15);A3$:NEXT
440 PRINTTAB(15);A2$:PRINTTAB(15);RIGHT$
(A$,9):FORT=0TO15:X=X(T):Y=Y(T):GOSUB800
450 PRINTCHR$(T+65):NEXT:FORT=0TO15:A(T)
=-1:NEXT:FORT=0TO7:FORJ=0TO1
460 R=INT(RND(1)*16):IFA(R)>-1THEN460
470 A(R)=T:NEXT:NEXT:GOSUB790:X=4:Y=12:G
OSUB800:PRINT"          "
480 PL=1-PL:GOSUB790:X=4:Y=12:GOSUB800:P
RINT"          "
490 IFPL=0THENGOSUB800:PRINT"PLAYER 1"
500 X=27:GOSUB800:PRINT"          ":IFPL=1
THENGOSUB800:PRINT"PLAYER 2"
510 FORC=0TO1
520 GETA$:IFA$<"A"ORA$>"P"THEN520
530 A=ASC(A$)-65:IFC=1THENIFA=K(0)THEN52
0
540 IFJ(A)=1THEN520
550 X=X(A):Y=Y(A):GOSUB800:PRINT"*":R=A(
A):GOSUB150:K(C)=A:R(C)=R:NEXT
560 IFR(0)=R(1)THEN600
570 FORC=0TO1:X=X(K(C)):Y=Y(K(C)):GOSUB8
00:PRINTCHR$(65+K(C)):NEXT:GOTO480
600 FORC=0TO1:X=X(K(C)):Y=Y(K(C)):GOSUB8
00:PRINTCHR$(18);" ":J(K(C))=1:NEXT
610 PRINT:S(PL)=S(PL)+10:GOSUB790:MM=MM+
1:IFMM<8THENPL=1-PL:GOTO480
620 X=2:Y=17:GOSUB800:IFS(0)=S(1)THENPRI
NT"TIE GAME!":GOTO650
630 T=1:IF S(1)>S(0)THEN T=2
640 PRINT"YOU DID IT PLAYER";T
650 X=12:Y=21:GOSUB800:PRINT"PRESS ANY K
EY..."
660 GETA$:IFA$=""THEN660
670 RUN
789 GOTO789
```

```
790 X=4:Y=8:GOSUB800:PRINT"SCORE:";S(0):
X=27:GOSUB800:PRINT"SCORE:";S(1):RETURN
800 POKECX,X:POKECY,Y:SYSCR:RETURN
900 FORBYTE=CRTOCR+9:READA:POKEBYTE,A:NE
XT:POKES0+24,15:POKES0+5,0:POKES0+6,240
910 S2=54279:POKES2+5,0:POKES2+6,240:RET
URN
920 DATA24,174,10,192,172,11,192,76,240,
255
930 DATA176,99,178,99,178,99,178,99,174
940 DATA98,32,98,32,98,32,98,32,98
950 DATA171,99,123,99,123,99,123,99,179
960 DATA173,99,177,99,177,99,177,99,189
```

PART VI

Computer Utilities

A *computer utility* is a program that can be used in many different contexts. A good example of a utility is a word processing program. Such program can be used to write business or personal letters, take notes, or write a poem, novel, or science fiction book. It can also be used to lay out a newsletter or advertising brochure. You determine the use given the constraints of the program. Without your use, however, it does nothing. This is quite different than, say, a preprogrammed game that runs one way and can only be used as the programmer intended.

In this section we'll provide you with a number of simple utilities. For example, one may be the world's cheapest and shortest word processor. Others will allow you to tailor graphs for yourself, to set up a ballot and ballot counter, and turn your computer into a calculator. Some of the utilities will have wide applications and others will be more specialized. Once you begin to feel comfortable as a programmer you should think of creating a library of your own utilities and subroutines that is stored on disk. That way, you will be able to add them to other programs you build or use them without having to go through the tedious process of typing them in. It's a good idea to number your subroutines and utilities beginning somewhere around 2000. That way, you can add them onto other programs just by loading them on top of programs that have lower numbers.

29

Scratch Pad: A Mini-Word Processor

This program provides a quick and easy 40-column word processor. When you run this program, you'll see nothing but a blank screen and the cursor. That's how it should be. There are no frills here, just a blank page for you to write on. Type anything you want. Use upper and lower case letters. All of the normal editing functions of the Commodore are available to you, including clear screen, insert space, delete space, and backspace. When you near the end of a line, press **RETURN** just as you would on a typewriter. After you are done and want to print out what you have written, turn on your printer and press the **F1** key. Whatever is on the screen will print out. To stop the printing before it's finished, press **F1** again.

Here's a little shape poem created with the sketch pad and printed out on an EPSON MX-80 printer:

```
          OV         OV
         LOVE       LOVE
       LOVELOV    LOVELOVE
     LOVELOVELOVELOVELOVEL
   LOVELOVELOVELOVELOVELOV
    LOVELOVELOVELOVELOVEL
     LOVELOVELOVELOVELOV
      LOVELOVELOVELOVEL
       LOVELOVELOVELOV
        LOVELOVELOVEL
         LOVELOVELOV
          LOVELOVEL
           LOVELOV
            LOVEL
             LOV
              L
```

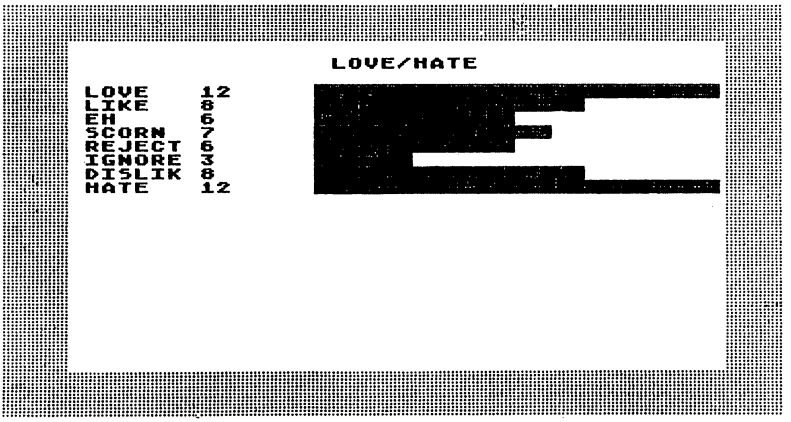
```
10 REM        *** SCRATCHPAD ***
100 PRINTCHR$(147);CHR$(5);CHR$(14);:GOT
O180
110 POKE199,0:POKE212,0:POKE216,0:GETA$:
IFA$=""THEN110
120 IFA$=CHR$(148)THENOY=Y:OX=X:GOSUB210
130 PRINTA$;:OX=X:OY=Y:X=PEEK(211)+40*(P
EEK(211)>39):IFA$=CHR$(133)THEN500
140 IFA$=CHR$(13)ORA$=CHR$(17)OR(OX=39AN
DX=0)THENY=Y+1
```

```
150 IFA$=CHR$(147)ORA$=CHR$(19)THENY=0
160 IF(OX=0ANDX=39)ORA$=CHR$(145)THENY=Y
-1
165 IF(OX=0ANDX=39)THENIFA$=CHR$(157)ORA
$=CHR$(20)THENPRINTA$;:X=38
170 IFX=39THENPRINTCHR$(13);:Y=Y+1:X=0
180 IFY<0THENY=0
190 IFY>24THENY=24:OY=OY-1
200 GOSUB210:GOSUB220:GOTO110
210 AD=1024+OY*40+OX:POKEAD,PEEK(AD)AND1
27:RETURN
220 AD=1024+Y*40+X:POKEAD,PEEK(AD)OR128:
POKEAD+54272,1:RETURN
500 OPEN4,4
510 FORA=0TO24:FORB=0TO39:C=PEEK(A*40+B+
1024)
520 IFC<32THENC=C+96
530 PRINT#4,CHR$(C);:NEXT:PRINT#4,CHR$(1
3)
540 GETA$:IFA$=CHR$(133)THENCLOSE4:GOTO1
10
550 NEXT:PRINT#4:CLOSE4:GOTO110
```

30

Graph Generator

This short program will produce a bar graph with up to 20 different bars. You can enter as many as 20 different labels and values. You can use up to seven-place numbers. The program will automatically scale the values you place in it so that what you will see is a graph of the relative sizes of the values of your labels. The values will also be printed out next to the labels as in this graph of the LOVE/HATE continuum:



In each graph, the largest number will fill up the entire space allotted to the graph. Here is the program, followed by some suggestions of the kinds of graphs you might want to experiment with:

```
10 REM       *** BARGRAPH ***
100 CR=49152:CY=CR+10:CX=CY+1:GOSUB500:D
IMV(20),L$(19):POKE53280,0:POKE53281,0
110 N=-1:DIMB$(3,1):FORT=0TO3:FORU=0TO1:
READK:B$(T,U)=CHR$(K):NEXT:NEXT
120 PRINTCHR$(147);CHR$(5);"              B
AR GRAPH GENERATOR":PRINT
130 INPUT"GRAPH TITLE";T$:PRINTCHR$(147)
;"          BAR GRAPH GENERATOR":PRINT
140 FORT=0TO19:A$="":PRINTT+1;TAB(24);"V
ALUE";CHR$(145):PRINTTAB(4);
150 INPUT"LABEL";L$(T):PRINTCHR$(145);TA
B(24);:INPUT"VALUE";A$
160 IFA$=""THENN=T-1:T=20
170 V(T)=VAL(A$+" "):IFV(T)>VTHENV=V(T)
180 NEXT:IFN=-1THENN=19
190 PRINTCHR$(147);TAB(20-LEN(T$)/2)T$:P
RINT:V=100/V:FORT=0TON
200 X=0:Y=T+2:GOSUB400:PRINTLEFT$(L$(T),
6):X=6:GOSUB400:PRINTV(T):X=14:GOSUB400
210 K=INT(V(T)*V):HV=INT(K/4):PRINTCHR$(
18);:IFHV>0THENFORJ=1TOHV:PRINT" ";:NEXT
220 PRINTB$(K-HV*4,0);B$(K-HV*4,1):NEXT
230 X=15:Y=23:GOSUB400:PRINT"PRESS <F1>"
240 GETA$:IFA$<>CHR$(133)THEN240
250 RUN
400 POKECX,X:POKECY,Y:SYSCR:RETURN
```

```
500 FORBYTE=CRTOCR+9:READA:POKEBYTE,A:NE
XT:RETURN
510 DATA24,174,10,192,172,11,192,76,240,
255
520 DATA146,0,146,180,146,161,0,234
700 FORM=0TOP:GOSUB850:PRINTA$(M):FORT=0
TOC(M)
710 X=2:Y=10+T*2:GOSUB800:PRINTB$(M,T);"
......";V(M,T):NEXT
720 X=14:Y=22:GOSUB800:PRINT"PRESS RETUR
N"
730 GETK$:IFK$<>CHR$(13)THEN730
740 NEXT:GOTO200
800 POKECX,X:POKECY,Y:SYSCR:RETURN
850 PRINTCHR$(147);" ":PRINT:PRINT"
   THE OFFICIAL BALLOT BOX":PRINT:PRINT:
PRINT:RETURN
900 FORBYTE=CRTOCR+9:READA:POKEBYTE,A:NE
XT:RETURN
910 DATA24,174,10,192,172,11,192,76,240,
255
```

Some graphing ideas:

- Graph school grades.
- Graph your monthly expenses and see the relative proportions of your income that you spend on food, rent, entertainment, etc.
- Graph income distribution, age distribution, voting patterns, and other demographics of your community.
- Graph the won-lost percentages of your favorite teams and see which teams have done the best over a period of time.
- Do a personal preference test or a survey of people's favorite records or movies and see the relative positions of the first 20 printed out in graph form.

31

The Ballot Box

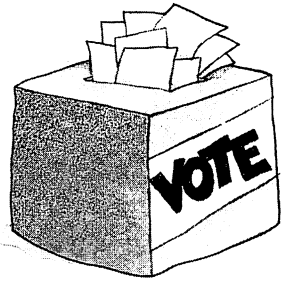
This is a voting utility. It allows you to set up anywhere from 1 to 5 different offices or positions to vote for. In each position, you can have up to five candidates. After the ballot is set up you can use your computer as a voting machine or you can tally individual ballots with it yourself. You can save your ballot on disk, load it any time you want and have tallys of the results of the voting listed out for you as in this screen dump for the election of a HIGH MUCK MUCK:

```
        THE OFFICIAL BALLOT BOX

HIGH MUCK MUCK

CRAZY CAT                          3
FELIX                              7
```

This utility can be used for school or club elections, popularity polls, voter preference surveys, etc. . . . anything that requires a vote can be recorded and tallied using it.



```
10 REM     *** THE BALLOT BOX ***
100 CR=49152:CY=CR+10:CX=CY+1:GOSUB900
110 POKE53280,0:POKE53281,0:PRINTCHR$(14
7);CHR$(5):CLOSE5
120 DIM V(4,4),B$(4,4)
200 GOSUB850:PRINT"1. SET UP THE BALLOT"
:PRINT:PRINT"2. SAVE BALLOT TO DISK"
210 PRINT:PRINT"3. LOAD BALLOT TO DISK":
PRINT:PRINT"4. VOTE"
220 GETK$:IFK$=""THEN220
230 K=ASC(K$)-48:IFK<1ORK>4THEN220
240 ONKGOTO300,500,550,600
300 GOSUB850:PRINT"HOW MANY OFFICES? (1-
5)"
310 GETK$:IFK$=""THEN310
320 K=ASC(K$)-49:IFK<0ORK>4THEN310
330 P=K:FORM=0TOP:GOSUB850:PRINT"ENTER N
AME OF OFFICE #";M+1:PRINT:INPUTT$
340 IFT$=""THENM=M-1:NEXT
350 A$(M)=T$:GOSUB850:PRINTT$:PRINT:PRIN
T"HOW MANY CANDIDATES? (1-5)"
360 GETK$:IFK$=""THEN360
370 C=ASC(K$)-49:IFC<0ORC>4THEN360
380 C(M)=C:FORT=0TOC:X=2:Y=10+T*2:GOSUB8
00:PRINTE$:GOSUB800
390 PRINT"CANDIDATE #";T+1;" ";:INPUTT$:
IFT$=""THENT=T-1:NEXTT
400 B$(M,T)=T$:NEXT:NEXT:GOTO200
500 OPEN5,8,5,"@0:CDATA,S,W":PRINT#5,P:F
ORT=0TOP:PRINT#5,C(T):NEXT
510 FORM=0TOP:PRINT#5,A$(M):FORT=0TOC(M)
:PRINT#5,B$(M,T):NEXT:NEXT
520 CLOSE5:GOTO200
550 OPEN5,8,5,"0:CDATA,S,READ":INPUT#5,P
:FORT=0TOP:INPUT#5,C(T):NEXT
560 FORM=0TOP:INPUT#5,A$(M):FORT=0TOC(M)
:INPUT#5,B$(M,T):NEXT:NEXT
570 CLOSE5:GOTO200
600 FORT=0TO4:FORJ=0TO4:V(T,J)=0:NEXT:NE
XT
610 GOSUB850:PRINT"PRESS SPACEBAR TO CAS
T YOUR BALLOT":PRINT
620 PRINT"PRESS RETURN TO END VOTING"
630 GETK$:IFK$=CHR$(13)THEN700
640 IFK$<>" "THEN630
650 FORM=0TOP:GOSUB850:PRINTA$(M):FORT=0
TOC(M):X=2:Y=10+T*2:GOSUB800
660 PRINTT+1;CHR$(157);". ";B$(M,T):NEXT
670 GETK$:IFK$=""THEN670
680 V=ASC(K$)-49:IFV<0ORV>C(M)THEN670
690 V(M,V)=V(M,V)+1:NEXT:GOTO610
```

32

Address Card File

The Address Card File can be used to store filing cards with any six different pieces of information. The form we are providing asks a person's name, address, city/state, home phone, and work phone; it also provides an extra line for remarks as in this entry:

As you can see, the bottom of the screen shows you the options available to you with the program. You can flip forward and backward to see other cards, begin at any letter (the program automatically alphabetizes your entries), add a new entry, change an old entry, and save and retrieve your cards on disk. Here is the program:

```
90 REM  *** ADDRESS CARD FILE ***
100 CR=49152:CY=CR+10:CX=CY+1:GOSUB900
110 POKE53280,0:POKE53281,0:PRINTCHR$(14
7);CHR$(5)
120 DIMN$(150,5):AZ=-1
130 C=0:P=C:PRINTCHR$(147);" ":PRINT:PRI
NT"      *** ADDRESS CARD FILE ***"
135 PRINT:PRINT:PRINT
```

```
140 FORT=1TO6:PRINTCHR$(18);M$(T):NEXT:F
ORT=1TO8:X=10:Y=14+T:GOSUB800
150 PRINTT;CHR$(157);". ";F$(T):NEXT:GOS
UB850
200 GETK$:IFK$<>""THENK=ASC(K$)-48:IFK<1
ORK>8THEN200
201 IFK$=""THEN200
210 ONKGOTO350,300,400,450,500,550,600,6
50
300 P=P-1:IFP<0THENP=AZ:IFP=-1THENP=0
310 GOSUB850:GOTO200
350 P=P+1:IFP>AZTHENP=0
360 GOSUB850:GOTO200
400 X=2:Y=4:GOSUB800:PRINT"WHICH LETTER?
"
405 GETK$:IFK$=""THEN405
410 K=ASC(K$):IFK<65ORK>90THEN405
415 IFAZ<0THEN440
420 FORJ=AZTOCSTEP-1:IFN$(J,0)>K$THENI=J
422 NEXTJ:IFI>=0THENJ=I:GOTO430
425 GOTO440
430 P=J:GOSUB850
440 X=2:Y=4:GOSUB800:PRINT"
 ":GOTO200
450 J=0:GOSUB810
455 FORV=0TO5:X=12:Y=6+V:GOSUB800:INPUTT
$:IFV>0THEN470
458 IFT$<N$(0,0)THENJ=0:GOTO465
460 FORJ=CTOAZ:IFT$>N$(J,0)THENNEXT:AZ=A
Z+1:J=AZ:GOTO470
465 AZ=AZ+1:FORW=AZTOJ+1STEP-1:FORWW=0TO
5:N$(W,WW)=N$(W-1,WW):NEXTWW:NEXTW
470 N$(J,V)=T$:T$=" ":NEXTV:P=J:GOSUB850
:GOTO200
500 IFAZ<0THENGOTO200
510 FORJ=PTOAZ:FORT=0TO5:N$(J,T)=N$(J+1,
T):NEXT:NEXT
520 AZ=AZ-1:P=AZ:IFAZ>-1THENGOSUB850:GOT
O200
530 GOSUB810:GOTO200
550 IFAZ=-1THEN200
560 FORJ=PTOAZ:FORT=0TO5:N$(J,T)=N$(J+1,
T):NEXT:NEXT
570 AZ=AZ-1:P=AZ:GOTO455
600 X=3:Y=4:GOSUB800:PRINT"REPLACE FILE
IN MEMORY? Y/N"
610 GETK$:IFK$="Y"THEN630
620 IFK$<>"N"THEN610
625 X=3:Y=4:GOSUB800:PRINT"
                    ":CLOSE5:GOTO200
630 CLOSE5:OPEN5,8,5,"0:ADDFILE,S,R":INP
UT#5,AZ:FORJ=0TOAZ:FORT=0TO5
```

```
640 INPUT#5,N$(J,T):NEXT:NEXT:P=0:GOSUB8
50:GOTO625
650 X=3:Y=4:GOSUB800:PRINT"REPLACE FILE
ON DISK? Y/N"
660 GETK$:IFK$="Y"THEN690
670 IFK$<>"N"THEN660
680 GOTO625
690 CLOSE5:OPEN5,8,5,"@0:ADDFILE,S,W":PR
INT#5,AZ:FORJ=0TOAZ:FORT=0TO5
700 PRINT#5,N$(J,T):NEXT:NEXT:P=0:GOSUB8
50:GOTO625
800 POKECX,X:POKECY,Y:SYSCR:RETURN
810 FORT=0TO5:X=12:Y=6+T:GOSUB800:PRINT"
                                ":NEXT
820 RETURN
850 GOSUB810:FORT=0TO5:X=13:Y=6+T:GOSUB8
00:PRINT" ";N$(P,T):NEXT:RETURN
900 FORBYTE=CRTOCR+9:READA:POKEBYTE,A:NE
XT:FORT=1TO7:READM$(T):NEXT
905 FORT=1TO8:READF$(T):NEXT:RETURN
910 DATA24,174,10,192,172,11,192,76,240,
255
920 DATA NAME       ,ADDRESS    ,CITY/STAT
E,HOME PHONE,WORK PHONE,REMARKS    ,XX
930 DATA FLIP FORWARD,FLIP BACKWARD,FLIP
 TO LETTER,ADD A NEW CARD
940 DATA DELETE CURRENT CARD,MODIFY CURR
ENT CARD
950 DATA LOAD FILE FROM DISK,SAVE FILE T
O DISK
```

Although the program is set up for addresses, it can easily be changed so that you can put recipes on it or organize a collection. For example, if you collect baseball cards, you can set up the program so that it stores:

PLAYERS NAME:
TEAM:
POSITION:
BATTING AVERAGE:
FIELDING PERCENT:
COMMENTS:

To do this, you have to change the data line 920. If you also want to organize your baseball cards alphabetically by team, put the TEAM category first and all of the players on the same team will be grouped together.

33

A Calculator Utility

For all of the sophistication of home computers, it can become frustrating when you can't use them as simple calculators. In the future, calculator functions will probably be built into the machines. However, if you don't feel like waiting, here is a simple program you can copy and save to disk that will allow you to use your computer as a calculator whenever you care to. You can also add this program to other programs and build in a small calculator to games or other programs where fast calculation is useful.

To use this program, you enter values and operators (+-*/=) as you would with any calculator. Press "C" for a complete clear. Press **ESC** to clear only the current entry. Press "S" to change the sign value of the value currently in the window.

```
10 REM    *** THE CALCULATOR ***
100 CR=49152:CY=CR+10:CX=CY+1:GOSUB900
110 POKE53280,0:POKE53281,0:PRINTCHR$(14
7);CHR$(5)
120 X=13:Y=6:GOSUB800:PRINT"THE CALCULAT
OR":X=10:Y=8:GOSUB800:PRINTCHR$(176);
130 FORT=1TO18:PRINTCHR$(99);:NEXT:PRINT
CHR$(174):Y=9:GOSUB800:PRINTCHR$(125);
140 PRINTTAB(29);CHR$(125):Y=10:GOSUB800
:PRINTCHR$(173);:FORT=1TO18
150 PRINTCHR$(99);:NEXT:PRINTCHR$(189)
160 T$=""
170 D$="0":GETK$:IFK$=""THEN170
180 IFK$="+"THENP=1:X=34:Y=9:GOSUB800:PR
INTK$:GOTO500
190 IFK$="-"THENP=2:X=34:Y=9:GOSUB800:PR
INTK$:GOTO500
200 IFK$="*"THENP=3:X=34:Y=9:GOSUB800:PR
INTK$:GOTO500
210 IFK$="/"THENP=4:X=34:Y=9:GOSUB800:PR
INTK$:GOTO500
220 IFK$="="THENGOSUB700:FF=1:AA=0:GOTO1
60
230 IFK$="C"THENGOSUB400:AA=0:KN=0:N=0:P
=0:X=34:Y=9:GOSUB800:PRINT"  ":GOTO160
```

```
240 IFK$=CHR$(95)THENGOSUB400:N=0:GOTO16
0
250 IFK$="."THEN300
260 IFK$="S"ANDN<>0THENN=-N:T$=STR$(N):G
OSUB400:GOTO340
270 IFK$<"0"ORK$>"9"THEN170
300 IFLEN(T$)=16THEN170
310 T$=T$+K$
320 IFT$<>"."THENN=VAL(T$)
330 IFFF=1THENGOSUB400:FF=0
340 X=28-LEN(T$):Y=9:GOSUB800:PRINTT$:GO
TO170
400 X=12:Y=9:GOSUB800:PRINT"
    ":RETURN
500 FF=1:IFAA=0THENAA=1:KN=N:KP=P:GOTO16
0
510 IFKP<>PTHENT=P:P=KP:KP=T:GOSUB700:P=
KP:GOTO160
520 GOSUB700:GOTO160
700 ONPGOTO710,720,730,740
710 N=N+KN:GOTO750
720 N=KN-N:GOTO750
730 N=N*KN:GOTO750
740 IFN<>0THENN=KN/N
750 T$=STR$(N):GOSUB400:X=28-LEN(T$):Y=9
:GOSUB800:PRINTT$:KN=N:RETURN
799 STOP
800 POKECX,X:POKECY,Y:SYSCR:RETURN
900 FORBYTE=CRTOCR+9:READA:POKEBYTE,A:NE
XT:RETURN
910 DATA24,174,10,192,172,11,192,76,240,
255
```

34

Dungeon Dice

This utility allows you to choose the number of dice you want to roll and the number of sides per die. You can choose up to 8 dice and up to 20 sides per die. In addition, it calculates the percentage of the possible maximum score that your roll produces. For people who have played Dungeons and Dragons, the convenience of this program will

probably be apparent. For anyone who wants to design a computer game with a complex element of choice built in, this can be a useful game subroutine.

Here is an example of what the program can do. Suppose you choose to roll five 10-sided dice. The computer will print out the results on each die and then tell you the percentage of the maximum score that your roll achieved. (In this case the maximum is 5*10=50.) Here is a possible roll:

```
DIE         1    2    3    4    5
SCORE       5    7    3    9    1
PERCENTAGE=(TOTAL/50)*100=25/50*100=50%
```

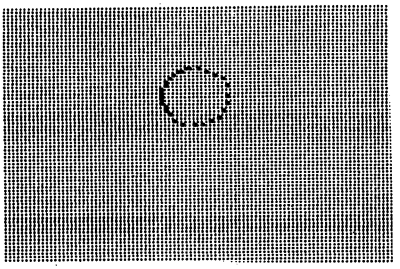
```
10 REM          *** DICE ***
100 CR=49152:CY=CR+10:CX=CY+1:GOSUB900
110 POKE53280,0:POKE53281,0:PRINTCHR$(14
7);CHR$(5)
120 DIMD$(26):FORT=1TO26:READK:D$=D$+CHR
$(K):NEXT
130 POKE54296,15:POKE54277,0:POKE54278,2
40
200 GOSUB500:X=2:Y=6:GOSUB800:INPUT"HOW
MANY DICE (1-8) ";N:IFN<1ORN>8THEN200
210 X=2:Y=8:GOSUB800:INPUT"HOW MANY SIDE
S (4-20) ";S:IFS<4ORS>20THEN210
220 P=S*N:N=N-1:GOSUB500
230 X=8:Y=19:GOSUB800:PRINT"PRESS SPACEB
AR TO ROLL":X=6:Y=21:GOSUB800
240 PRINT"PRESS RETURN TO CHANGE DICE":Z
=20-((N+1)*2)
300 GETK$:IFK$=CHR$(13)THEN200
310 IFK$<>" "THEN300
320 FORJ=0TOINT(RND(1)*3)+2:C=0:FORT=0TO
N:X=Z+T*4:Y=7:GOSUB800
330 R=INT(RND(1)*S)+1:R$=STR$(R):R$=RIGH
T$(R$,LEN(R$)-1):PRINTD$;R$
340 POKE54276,33:POKE54273,T*10:C=C+R:FO
R0=1TO10:NEXT
345 POKE54276,0:NEXT:NEXT:POKE54276,0
350 X=15:Y=13:GOSUB800:PRINT"TOTAL: ";C;
CHR$(157);" ":X=12:Y=14:GOSUB800
360 PRINT"POSSIBLE: ";P;"  ":X=10:Y=15:G
OSUB800
370 PRINT"PERCENTAGE: ";INT(C/P*100);CHR
$(157);"% ":GOTO 300
500 PRINTCHR$(147);:X=10:Y=2:GOSUB800:PR
INT"DUNGEON DICE ROLLER":RETURN
```

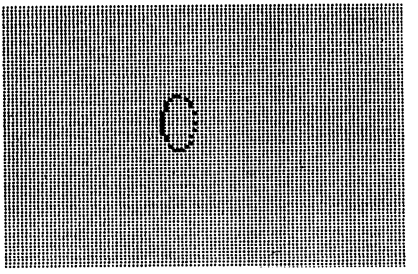
```
800 POKECX,X:POKECY,Y:SYSCR:RETURN
900 FORBYTE=CRTOCR+9:READA:POKEBYTE,A:NE
XT:RETURN
910 DATA24,174,10,192,172,11,192,76,240,
255
920 DATA 176,195,195,238,157,157,157,157
,17,194,32,32,194,157,157,157,157,17
930 DATA 173,195,195,189,157,157,157,145
```

35

Circle and Ellipse

Drawing a circle in BASIC is difficult and time-consuming. Drawing an ellipse is almost impossible. Here is a program that draws two circles, one big and one small, as well as two ellipses that also differ in size. These programs use the trigonometric functions SINE and COSINE and make it possible to be much more accurate in drawing curves on a TV or monitor screen which cannot actually draw continuous lines but depends on discrete points called *pixels* for its images. A way to experiment with these programs is to separate them and change all of the parameters in each miniprogram separately. If you do animation or try to create interesting visual formats for your programs, you might find these subroutines very useful.





```
10 REM  *** CIRCLES & ELLIPSES ***
100 CR=49152:CY=CR+10:CX=CY+1:GOSUB900
110 POKE53280,0:POKE53281,0:PRINTCHR$(14
7);CHR$(5)
120 REM *** CIRCLE ***
130 PRINTCHR$(147);CHR$(18);
150 A=A+.28: X= INT(SIN(A)*5+20): Y=INT(
COS(A)*4+12): GOSUB 800:T=INT(RND(1)*8)
160 PRINTCHR$(C(T));" ";:GETK$:IFK$=""TH
EN150
200 REM *** BIG CIRCLE ***
210 A=A+.28:X=INT(SIN(A)*9+20):Y=INT(COS
(A)*8+12):GOSUB800:T=INT(RND(1)*8)
220 PRINTCHR$(C(T));" ";:GETK$:IFK$=""TH
EN210
300 REM *** ELLIPSE ***
305 PRINTCHR$(147);CHR$(18);
310 A=A+.28:X=INT(SIN(A)*6+20):Y=INT(COS
(A)*3+12):GOSUB800:T=INT(RND(1)*8)
320 PRINTCHR$(C(T));" ";:GETK$:IFK$=""TH
EN310
400 REM *** BIG ELLIPSE ***
410 A=A+.28:X=INT(SIN(A)*13+20):Y=INT(CO
S(A)*8+12):GOSUB800:T=INT(RND(1)*8)
420 PRINTCHR$(C(T));" ";:GETK$:IFK$=""TH
EN410
430 GOTO120
800 POKECX,X:POKECY,Y:SYSCR:RETURN
900 FORBYTE=CRTOCR+9:READA:POKEBYTE,A:NE
XT
905 FORT=0TO7:READC(T):NEXT:RETURN
910 DATA24,174,10,192,172,11,192,76,240,
255
920 DATA 5,28,159,156,30,31,158,129
```

36

The Simple Averager

This program allows you to average numbers. As you enter a list of numbers, it gives you a running average and a count of how many numbers you have entered. It is convenient for averaging school grades, income, or in the climate we come from, daily rainfall.

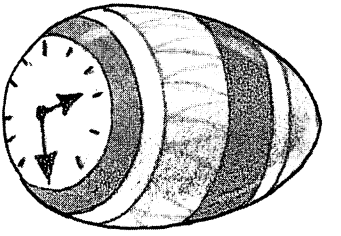
```
10 REM       *** AVERAGER ***
100 PRINTCHR$(147):PRINT:PRINT"
 SIMPLE AVERAGER":PRINT:PRINT
110 PRINT"ENTER VALUES - PRESS RETURN TO
 CLEAR":PRINT:PRINT:PRINT"AVERAGE:";
120 IFC>0THENPRINTINT(T/C*100)/100;
130 PRINTTAB(24);"VALUES:";C:PRINT:PRINT
:PRINT:A$="":INPUTA$:IFA$=""THENRUN
160 N=VAL(A$):T=T+N:C=C+1:GOTO100
```

37

The Egg Timer

It's not very likely that you have your computer in the kitchen or that you'll use it to time your eggs. However, a timing program is a very convenient insert into other programs or to use to remind you to make a phone call or wake up from a nap. Actually, there are even a few kitchen computer freaks who do use their computers to time their eggs.

In this program you enter the number of minutes or seconds you want to be timed. The computer will display and count down the time, emitting a sound (hopefully pleasant) when it reaches 0. If you want, you can change that sound to a Bach fugue or a Mozart sonata. Just set up a subroutine at line 300 and set to work programming the music of your choice.



```
10 REM       *** EGG TIMER ***
100 SO=54272:CR=49152:CY=CR+10:CX=CY+1:G
OSUB900:GOSUB400:Y=14:Y=6:GOSUB800
110 POKE53280,0:POKE53281,0:PRINTCHR$(5)
:POKESO+24,15
120 PRINT"SET TIMER...":X=2:Y=10:GOSUB80
0:INPUT"MINUTES";M:Y=12
130 GOSUB800:INPUT"SECONDS";S:IFS<00RS>5
9THEN130
140 GOSUB400:X=19-LEN(STR$(M)):Y=7:GOSUB
800:PRINTM;CHR$(157);":";
150 IFS<10THENPRINT"0";
160 PRINTRIGHT$(STR$(S),LEN(STR$(S))-1):
X=15:Y=10:GOSUB800:PRINT"PRESS <F1>"
170 GETA$:IFA$<>CHR$(133)THEN170
180 POKE162,0:GOSUB800:PRINT"              "
190 X=19-LEN(STR$(M)):Y=7:GOSUB800:PRINT
M;CHR$(157);":";:IFS<10THENPRINT"0";
200 PRINTRIGHT$(STR$(S),LEN(STR$(S))-1)
210 IFPEEK(162)<60THEN210
220 POKE162,0:S=S-1:IFS<0THENS=59:M=M-1:
IFM<0THENX=14:Y=10:GOTO300
230 GOTO190
300 POKESO+4,33:FORT=0TO255STEP5:POKESO+
1,T:NEXT:POKESO+4,0:GOSUB800:Q=1-Q
310 POKE53280,(1-Q)*11:IFQ=0THENGOSUB800
:PRINT"                "
320 IFQ=1THENGOSUB800:PRINT"PRESS ANY KE
Y"
330 GETA$:IFA$=""THEN300
340 RUN
400 PRINTCHR$(147);CHR$(5):X=9:Y=3:GOSUB
800:PRINT"*** THE EGG TIMER ***":RETURN
799 STOP
800 POKECX,X:POKECY,Y:SYSCR:RETURN
900 FORBYTE=CRTOCR+9:READA:POKEBYTE,A:NE
XT:POKESO+5,0:POKESO+6,240:RETURN
910 DATA24,174,10,192,172,11,192,76,240,
255
```

38

The Wise Shopper

This program converts the price of a product to its price per ounce or per gram which is useful for comparison shopping. Sometimes, big boxes contain a lot of empty space and buying by unit weight can save you money. To use the program, enter the product's total price and then choose ounces or grams before you enter the product weight.

```
10 REM     *** THE WISE SHOPPER ***
100 CR=49152:CY=CR+10:CX=CY+1:GOSUB900
110 POKE53280,0:POKE53281,0:PRINTCHR$(14
7);CHR$(5)
120 PRINTCHR$(147);:PRINT:PRINT"         *
** THE WISE SHOPPER ***"
130 PRINT:PRINT:PRINT:INPUT" TOTAL PRODU
CT PRICE: ";P
140 GOSUB850:PRINT:PRINT:PRINT"WEIGHT IN
 (1) OUNCES OR (2) GRAMS."
150 GETK$:IFK$<>"1"ANDK$<>"2"THEN150
160 K=VAL(K$):W$="OUNCES":IFK=2THENW$="G
RAMS"
170 PRINT:PRINT"ENTER THE WEIGHT IN ";W$
;:INPUTW
180 IFK=1THENG=INT(28.35*W*100)/100
190 IFK=2THENG=W:W=INT(G/28.35*100)/100
200 PRINT:PRINT"OUNCES.....";W:PR
INT"PRICE PER OUNCE...";INT(P/W*10000)/1
00;"CENTS."
210 PRINT:PRINT"GRAMS,.....";G:PR
INT"PRICE PER GRAM....";
220 T=INT(P/G*10000)/100:IFT=0THENPRINT"
--"
230 IFT<>0THENPRINTT;"CENTS."
240 PRINT:PRINT:PRINT"              PRESS A
NY KEY"
250 GETK$:IFK$=""THEN250
260 RUN
800 POKECX,X:POKECY,Y:SYSCR:RETURN
850 PRINTCHR$(147);:PRINT:PRINT"         *
** THE WISE SHOPPER ***"
860 PRINT:PRINT:PRINT:PRINT"TOTAL PRODUC
T PRICE: ";P:RETURN
```

```
900 FORBYTE=CRTOCR+9:READA:POKEBYTE,A:NE
XT:RETURN
910 DATA24,174,10,192,172,11,192,76,240,
255
```

39

Metric Converts

This is a simple conversion utility. It allows you to convert the following pairs into each other's units of measurement: centimeters/inches; meters/yards; and kilometers/miles.

```
10 REM        *** METRIC ***
100 PRINTCHR$(147):PRINT:PRINTTAB(11);"M
ETRIC CONVERSIONS"
110 PRINT:PRINT:PRINT:PRINT"1. CENTIMETE
R TO INCH":PRINT"2. METER TO YARD"
120 PRINT"3. KILOMETER TO MILE":PRINT"4.
 INCH TO CENTIMETER"
130 PRINT"5. YARD TO METER":PRINT"6. MIL
E TO KILOMETER"
140 GETA$:A=VAL(A$):IFA<1ORA>6THEN140
150 PRINT:PRINT:ONAGOTO160,170,180,190,2
00,210
160 INPUT"CENTIMETERS";A:A=A*.394:PRINT:
PRINT"INCHES:"A:GOTO300
170 INPUT"METERS";A:A=A*1.094:PRINT:PRIN
T"YARDS:";A:GOTO300
180 INPUT"KILOMETERS";A:A=A*.621:PRINT:P
RINT"MILES:";A:GOTO300
190 INPUT"INCHES";A:A=A*2.538:PRINT:PRIN
T"CENTIMETERS:";A:GOTO300
200 INPUT"YARDS";A:A=A*.914:PRINT:PRINT"
METERS:";A:GOTO300
210 INPUT"MILES";A:A=A*1.609:PRINT:PRINT
"KILOMETERS:";A
300 PRINT:PRINT:PRINTTAB(8)"PRESS
ANY KEY TO CONTINUE"
310 GETA$:IFA$=""THEN310
320 RUN
```

Here are the lines where the conversion formulas are stored. You can change them if you want to provide other conversions such as pounds

into ounces (using the formula OUNCE=POUND*16), Fahrenheit into Centigrade, tons into pounds, yards into feet, etc.

```
110 PRINT:PRINT:PRINT:PRINT"1. CENTIMETE
R TO INCH":PRINT"2. METER TO YARD"
120 PRINT"3. KILOMETER TO MILE":PRINT"4.
 INCH TO CENTIMETER"
130 PRINT"5. YARD TO METER":PRINT"6. MIL
E TO KILOMETER"
140 GETA$:A=VAL(A$):IFA<1ORA>6THEN140
150 PRINT:PRINT:ONAGOTO160,170,180,190,2
00,210
160 INPUT"CENTIMETERS";A:A=A*.394:PRINT:
PRINT"INCHES:"A:GOTO300
170 INPUT"METERS";A:A=A*1.094:PRINT:PRIN
T"YARDS:";A:GOTO300
180 INPUT"KILOMETERS";A:A=A*.621:PRINT:P
RINT"MILES:";A:GOTO300
190 INPUT"INCHES";A:A=A*2.538:PRINT:PRIN
T"CENTIMETERS:";A:GOTO300
200 INPUT"YARDS";A:A=A*.914:PRINT:PRINT"
METERS:";A:GOTO300
210 INPUT"MILES";A:A=A*1.609:PRINT:PRINT
```

40

Decimal/Hex/Binary Conversion

This program is very useful for people who work with computers. It converts numbers from the binary, decimal, and hexadecimal (base 16) number systems into equivalent form in the other systems. You can start with any of the three systems and get the equivalent in the other two systems. In case you are not familiar with some of these systems of numeration, here is a summary of how they work:

Base 10: Symbols 0 1 2 3 4 5 6 7 8 9

The base 10 system is the one we all learn first. It has ten symbols and uses the powers of ten to change place value. Thus, counting in base 10 goes as follows:

1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17

Note that once all the symbols are used up you move over one place (from 9 to 10). The powers of ten that are given in the following chart determine the values of the symbols:

10^3	10^2	10^1	10^0
1000	100	10	1

Thus, the number 1234 actually means:

$1*10^3 +2*10^2 +3*10^1 +4+10^0 =1234$

You might remember learning this in fifth or sixth grade though most people usually forget it quickly since they don't need to know it in order to calculate. However, when you move from one base system into another the question of place value is crucial.

Base 2: Symbols 0 1

There are only two symbols in base 2: 0 and 1, which is why it is so important for computer programming. Every computer is a complex and well-organized collection of ON and OFF switches that can be represented by 1(ON) and 0(OFF). The language your computer understands is base 2 machine language and everything you type in has to be translated into a base 2 system before the computer can process it. Base 2 functions structurally like base ten. It uses place values only they represent powers of 2 instead of powers of 10. Here are the values of the first four places in a base 2 number system:

	2^3	2^2	2^1	2^0
value	8	4	2	1

Here are the numbers from 1 to 10 in base 2:

1, 10, 11, 100, 101, 110, 111, 1000, 1001, 1010

To understand this, let's take a closer look at 10 which is represented in base 2 as 1010 and translate it to base 10:

$1010=1*2^3+0*2^2+1*2^1+0*2^0=8+0+2+0=10$

Base 2 is a natural language for programming computers. However, programming in this machine language is tedious and leads to many errors. For example, imagine trying to copy this machine language program without making an error or going to sleep:

1011100010101010
1100000100011001
0110010101010101
1000001001100010
1111100100010101

In order to facilitate programming on a level that is close to machine language and that runs more quickly than BASIC (which has to be translated into machine language internally before it can be RUN by the computer) an intermediate code using base 16 (called the *hexadecimal system*) is often used. The reason for this is that four places in the binary system (i.e., 2 to the 4th power of 2*2*2*2) equal 16 and, so, one digit in the hexadecimal system can represent four in the binary system. Here is the hex symbolism. If you want to learn more about this you should consult an introductory book that has a chapter on assembly language programming.

Base 16 Number System (Hexadecimal): Symbols 0 1 2 3 4 5 6 7 8 9 A B C D E F

The letters A, B, C, D, E, F represent the base 10 numbers 10, 11, 12, 13, 14 and 15.

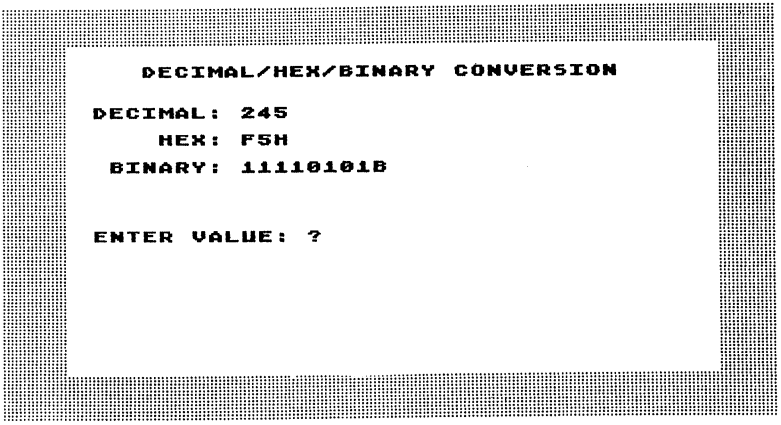
Counting from 1 to 10 in HEX looks like this: 1, 2, 3, 4, 5, 6, 7, 8, 9, A. Here are the HEX place values:

	16^3	16^2	16^1	16^0
values	4096	256	16	1

Thus 100A in Hex would be:

$1*16^3+0*16^2+0*16^1+10*16^0=4096+0+0+10=5006$ in base 10.

Enough of the details. Here is a conversion program which might come in handy if you do machine or assembly language programming:




```
10 REM  *** DECIMAL/HEX/BINARY ***
100 CR=49152:CY=CR+10:CX=CY+1:GOSUB900
110 POKE53280,0:POKE53281,0:PRINTCHR$(14
7);CHR$(5)
120 DIMB(19):HH$="0123456789ABCDEF"
130 D$="0":H$=D$:B$=D$:GOSUB840
140 FORT=0TO4:READK:H(T)=K:B(T*4)=K:FORJ
=1TO3:READB(T*4+J)
150 NEXT:NEXT:E$="
                  "
200 X=2:Y=14:GOSUB800:PRINTE$:X=2:Y=14:I
NPUT"ENTER VALUE: ";T$
210 L=LEN(T$):IFRIGHT$(T$,1)="B"THEN400
220 IFRIGHT$(T$,1)="H"THEN500
230 D=VAL(T$):D$=T$:IFD>1048575THEN200
300 R=1:IFD>255THENR=3:IFD>65535THENR=4
310 H$="":K=D:FORT=RTO0STEP-1:J=INT(K/H(
T)):K=K-J*H(T)
320 H$=H$+MID$(HH$,J+1,1):NEXT
350 R=7:IFD>255THENR=15:IFD>65535THENR=1
9
360 B$="":K=D:FORT=RTO0STEP-1:J=INT(K/B(
T)):K=K-J*B(T):J$=STR$(J)
370 J$=RIGHT$(J$,1):B$=B$+J$:NEXT:GOSUB8
40:GOTO200
400 D=0:C=0:FORT=L-1TO1STEP-1:IFMID$(T$,
T,1)="1"THEND=D+B(C):GOTO420
410 IFMID$(T$,T,1)<>"0"THEN200
420 C=C+1:NEXT:D$=STR$(D):GOTO300
```

```
500 D=0:C=0:FORT=L-1TO1STEP-1:A=ASC(MID$
(T$,T,1)):IFA>64ANDA<71THENA=A-55:B=5
510 IFA>47ANDA<58THENA=A-48:B=5
520 IFB<>5THEN200
530 B=0:D=D+A*H(C):C=C+1:NEXT
540 D$=STR$(D):GOTO 300
800 POKECX,X:POKECY,Y:SYSCR:RETURN
840 PRINTCHR$(147);:PRINT:PRINT" *** HEX
/DECIMAL/BINARY CONVERSIONS ***"
850 X=1:Y=5:GOSUB800
860 PRINT" DECIMAL: ";D$:PRINT:PRINT"
   HEX: ";H$;"H"
870 PRINT:PRINT"    BINARY: ";B$;"B":RETU
RN
900 FORBYTE=CRTOCR+9:READA:POKEBYTE,A:NE
XT:RETURN
910 DATA24,174,10,192,172,11,192,76,240,
255
920 DATA 1,2,4,8,16,32,64,128,256,512,10
24,2048,4096,8192,16384,32768,65536
930 DATA 131072,262144,524288
```

Finale

41

A Computer Dating Service

And now, to end the book (almost) on an up note, here is a very liberal dating program. The program is set up to record the answers to 20 questions after taking your name and phone number and assigning you a code number. When you have completed your profile, you can store the program on disk. One person a dating program does not make. Therefore, you have to gather data from a lot of people before it can begin to match their responses and come up with what, from its

point of view (really our point of view since a computer does not have convictions or sets up criteria for matching), is your ideal mate from the available pool.

```
80 REM * COMPUTER DATING SERVICE *
90 DIMNA$(99),PA$(99),Q(99,19),A1(19),A2
(19),A3(19),Q1$(20),Q2$(20)
100 CR=49152:CY=CR+10:CX=CY+1:GOSUB900
110 POKE53280,0:POKE53281,0:PRINTCHR$(14
7);CHR$(5):OPEN15,8,15
130 GOSUB810:PRINT:PRINT:PRINT"1. ADD A
PERSON TO DATING POOL"
140 PRINT:PRINT"2. SELECT PERSON TO DATI
NG POOL":PRINT:PRINT"3. SAVE DATA BANK T
O DISK"
150 PRINT:PRINT"4. LOAD DATA BANK TO DIS
K"
160 GETK$:IFK$=""THEN160
170 K=ASC(K$)-48:IFK<1ORK>4THEN160
180 ONKGOTO300,200,500,600
200 GOSUB810:INPUT"ENTER YOUR I.D. NUMBE
R: ";N:IFN<0ORN>=CTHEN130
210 PRINT:PRINT"FINDING A DATE FOR:":PRI
NTNA$(N):PRINTPA$(N)
```

```
220 DD=0:DT=0:FORT=0TOC-1:TT=0
230 IFT=NTHEN250
235 FORJ=0TO19
240 IFQ(N,A1(J))=Q(T,A2(J))THENTT=TT+A3(
J)
245 NEXT
250 IFTT>DTTHENDT=TT:DD=T
260 NEXT:PRINT:PRINT
270 PRINT"YOUR IDEAL DATE IS:":PRINTNA$(
DD):PRINTPA$(DD):PRINT:PRINT"..PRESS A K
EY"
280 GETK$:IFK$=""THEN280
290 GOTO130
300 GOSUB810:INPUT"ENTER YOUR NAME ";N$:
NA$(C)=N$
310 GOSUB810:INPUT"ENTER PHONE NUMBER ";
N$:PA$(C)=N$
320 FORT=1TO20:GOSUB810:PRINTQ1$(T):PRIN
TQ2$(T)
330 GETK$:IFK$=""THEN330
340 K=ASC(K$)-48:IFK<1ORK>3THEN330
350 Q(C,T-1)=K:NEXT
360 PRINT:PRINTNA$(C):PRINT"YOUR I.D. NU
MBER IS ";C:FORT=1TO2000:NEXT:C=C+1
370 GOTO130
500 OPEN5,8,5,"@0:DATEDATA,S,W":PRINT#5,
C:FORT=0TOC-1:PRINT#5,NA$(T)
510 PRINT#5,PA$(T):FORJ=0TO19:PRINT#5,Q(
T,J):NEXT:NEXT:CLOSE5:GOTO130
600 OPEN5,8,5,"0:DATEDATA,S,R":INPUT#5,C
:FORT=0TOC-1:INPUT#5,NA$(T)
610 INPUT#5,PA$(T):FORJ=0TO19:INPUT#5,Q(
T,J):NEXT:NEXT:CLOSE5:GOTO130
800 POKECX,X:POKECY,Y:SYSCR:RETURN
810 PRINTCHR$(147)," ":PRINT:PRINT"
   COMPUTER DATING SERVICE":PRINT:PRINT:
RETURN
900 FORBYTE=CRTOCR+9:READA:POKEBYTE,A:NE
XT
920 FORT=1TO20:READQ$,T$:Q1$(T)=Q$:Q2$(T
)=T$:NEXT:FORT=0TO19
930 READA1(T),A2(T),A3(T):NEXT:RETURN
950 DATA24,174,10,192,172,11,192,76,240,
255
960 DATA ARE YOU (1)MALE (2)FEMALE (3)BO
TH,
965 DATA DO YOU WISH TO DATE A (1)MALE,(
2)FEMALE (3)BOTH
970 DATA ARE YOU (1)YOUNG (2)MIDDLE-AGED
,(3)OLD
975 DATA IS YOUR IDEAL DATE (1)YOUNG,(2)
MIDDLE-AGED (3)OLD
```

```
980 DATA HOW ATTRACTIVE ARE YOU (1)EXTRE
MELY,(2)MODERATELY (3)NOT VERY
985 DATAHOW ATTRACTIVE IS YOUR IDEAL DAT
E,(1)EXTREMELY (2)MODERATELY (3)NOT VERY
990 DATA WHERE WOULD YOU LIKE TO GO ON A
 FIRST,DATE (1)MOVIE (2)DANCING (3)MOTEL
995 DATA ARE YOU AN (1)OUTDOOR PERSON,(2
)INDOOR PERSON (3)BOTH
1000 DATA HOW IMPORTANT IS SEX (1)VERY,(
2)FAIRLY (3)NOT VERY
1010 DATA ARE YOU ARTISTIC (1)YES (2)NO,
(3)SORT OF
1015 DATA WOULD YOU LIKE YOUR DATE TO BE
,ARTISTIC (1)YES (2)NO (3)SORT OF
1020 DATA HOW DO YOU FEEL ABOUT KIDS,(1)
LOVE THEM (2)HATE THEM (3)BOTH
1025 DATA DO YOU ENJOY ALCOHOL OR DRUGS,
(1)YES (2)NO (3)SOMETIMES
1030 DATA WHICH TYPE OF MUSIC DO YOU LIK
E MOST,(1)CLASSICAL (2)ROCK (3)COUNTRY
1035 DATA HOW MUCH TELEVISION DO YOU WAT
CH,(1)A LOT (2)SOME (3)VERY LITTLE
1040 DATA ARE YOU (1)WEALTHY (2)DOING OK
,(3)GETTING BY
1045 DATA IS YOUR IDEAL DATE (1)WEALTHY,
(2)DOING OKAY (3)GETTING BY
1050 DATA ARE YOUR HABITS (1)UNHEALTHY,(
2)HEALTHY (3)NO HABITS
1055 DATA ARE YOU USUALLY (1)HAPPY (2)UN
HAPPY,(3)BOTH
1060 DATA IS YOUR IDEAL DATE USUALLY (1)
HAPPY,(2)UNHAPPY (3)BOTH
1065 DATA 1,0,50,0,1,50,3,2,25,2,3,25,5,
4,25,4,5,25,6,6,50,7,7,50,8,8,50,10,9
1070 DATA 25,9,10,25,11,11,25,12,12,50,1
3,13,25,14,14,25,16,15,25,15,16,25,17
1075 DATA 17,25,19,18,25,18,19,25
```

Notice that the questions are all listed in the following data statements:

```
960 DATA ARE YOU (1)MALE (2)FEMALE (3)BO
TH,
965 DATA DO YOU WISH TO DATE A (1)MALE,(
2)FEMALE (3)BOTH
970 DATA ARE YOU (1)YOUNG (2)MIDDLE-AGED
,(3)OLD
975 DATA IS YOUR IDEAL DATE (1)YOUNG,(2)
MIDDLE-AGED (3)OLD
980 DATA HOW ATTRACTIVE ARE YOU (1)EXTRE
MELY,(2)MODERATELY (3)NOT VERY
```

```
985 DATAHOW ATTRACTIVE IS YOUR IDEAL DAT
E,(1)EXTREMELY (2)MODERATELY (3)NOT VERY
990 DATA WHERE WOULD YOU LIKE TO GO ON A
 FIRST,DATE (1)MOVIE (2)DANCING (3)MOTEL
995 DATA ARE YOU AN (1)OUTDOOR PERSON,(2
)INDOOR PERSON (3)BOTH
1000 DATA HOW IMPORTANT IS SEX (1)VERY,(
2)FAIRLY (3)NOT VERY
1010 DATA ARE YOU ARTISTIC (1)YES (2)NO,
(3)SORT OF
1015 DATA WOULD YOU LIKE YOUR DATE TO BE
,ARTISTIC (1)YES (2)NO (3)SORT OF
1020 DATA HOW DO YOU FEEL ABOUT KIDS,(1)
LOVE THEM (2)HATE THEM (3)BOTH
1025 DATA DO YOU ENJOY ALCOHOL OR DRUGS,
(1)YES (2)NO (3)SOMETIMES
1030 DATA WHICH TYPE OF MUSIC DO YOU LIK
E MOST,(1)CLASSICAL (2)ROCK (3)COUNTRY
1035 DATA HOW MUCH TELEVISION DO YOU WAT
CH,(1)A LOT (2)SOME (3)VERY LITTLE
1040 DATA ARE YOU (1)WEALTHY (2)DOING OK
,(3)GETTING BY
1045 DATA IS YOUR IDEAL DATE (1)WEALTHY,
(2)DOING OKAY (3)GETTING BY
1050 DATA ARE YOUR HABITS (1)UNHEALTHY,(
2)HEALTHY (3)NO HABITS
1055 DATA ARE YOU USUALLY (1)HAPPY (2)UN
HAPPY,(3)BOTH
1060 DATA IS YOUR IDEAL DATE USUALLY (1)
HAPPY,(2)UNHAPPY (3)BOTH
```

The liberality of the program shows in the very first question. It allows for you to be male, female, or both. The second question allows you to date the sexual preferences of your choice. Of course, you might feel this program goes too far. All you have to do is list the data statements and change them to suit your own ways of looking at the problem of finding a best mate or date. Just be sure that each change you make goes on one data line. The weighting factors built into the program can also be changed if you have some programming experience. You'll find them in lines 1065 to 1075.

1/2 of Many Things You Can Do With Your Computer

You can use the programs as they are listed in this book. You can also consider them program forms that provide a framework for developing new programs on your own. Here are a number of ideas that take off from the 41 programs we have worked out. They provide you 1/2 of what you need to develop applications for your computers. The other half, the new programs themselves, are for you to work out.

1 An All-Purpose Instrument Tuner

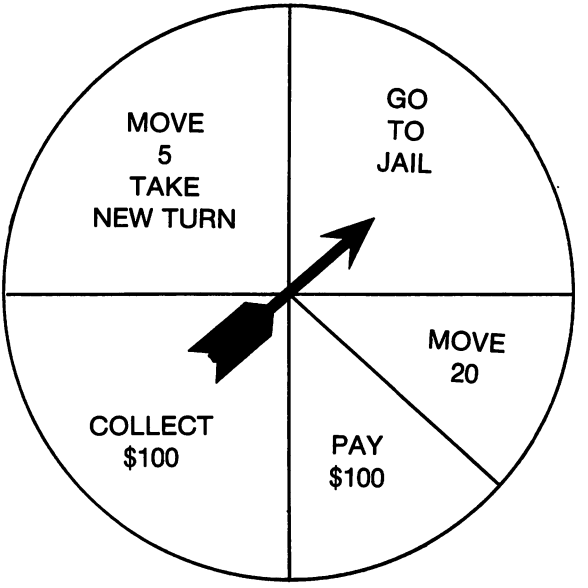
One of the programs in this book provides a guitar tuner. Using the sound capabilities of your computer, you can also create a saxophone, clarinet, violin, fiddle, trumpet, or all-purpose instrument tuner. Instead of string diagrams, you would have to suit the graphics to the nature of the different instruments and, therefore, have many different graphics screens in your program.

2 A Music Tutor

This program would extend the notion of a guitar chord tutor. It could show the fingering of woodwinds, the valve positions on brass, etc. You could even develop an extensive music teaching program.

3 A Game Spinner

You can use some of the same ideas used in the Dungeons and Dragons dice maker to create a game spinner. The spinner differs from a usual die in that it can have the same values on different sections, or it can have verbal instructions or game moves on it. Different sections can also be weighted so that there is not the same probability of getting each possible result. Here is an example of what a spinner might look like:



On the sample spinner there is a greater chance of landing in jail than collecting money. Notice that the instructions on the spinner refer to game moves and to paying or collecting money. These different modes and weights can easily be incorporated into a customized game spinner program.

4 A Split Screen Note Pad

Try to create a note pad that splits the screen in half so that you can write on either half, or write on one half and make sketches and notes on the other half. Try to build in a way to press a letter or number which allows you to control which part of the screen you use.

5 A Calculator with Memory

Add a memory storage capability to the calculator program so that the results of any calculation can be stored and then used at another time.

6 Animated Guessing Games

Modify the Gestalt guessing game so that you see small figures running all around the screen and have to guess how many there are. Remember that animation can easily be done using Control graphics and clearing the screen after each positioning of your figures. When you draw a figure, clear the screen and then draw the figure in a slightly different position or posture you create the illusion of animation.

7 Mandala Meditation

Create your own meditation programs playing with the color function of your computer. Add sound to the program and see if you can create a multimedia work of art.

8 Pie Chart and Line Graph Generator

The bar graph generator presented in the book is only one of many graph forms that can be programmed. Try to make a line graph. (Remember that the bar graph can easily be turned into a line graph by turning it on its side and drawing lines from the top points.) Also

create a pie graph generator. A pie graph generating program can also be used in a game spinner generating program.

9 Multiple Sorting Address Card File

Modify the address file program so that you can sort by state, city, phone number, and comments as well as by name. What this requires is that you build routines similar to the one used in the program in this book to do name sorting for each line in the address book entry.

10 Profile Mug Shots

Using character graphics, change the Mug Shot program from one that uses full faces to one that uses profiles.

11 Multilingual Dictionary

Create a multilingual dictionary. For example, create a dictionary in five languages. Set up the program so that you have a few entries to begin with (for example all the pronouns in five different languages) and then new entries can be added and saved on disk. This way you will not merely have a multilingual dictionary but a dictionary form. As you learn more vocabulary, you build your dictionary. Many of the routines needed to create such a program can be taken from programs listed in this book (the disk storage routine, for example).

12 Prime Number Generator

Create a program that starts at 1 and lists the first n prime numbers for any integer n. Add another routine to the program in which you input any integer and the computer tells you whether it is prime or not.

13 *Least Common Denominator*

Here is another homework helper program. Set up the computer so that it will list all the common denominators of any two or three numbers and also list the least common denominator.

14 *A Book Title Generator*

We included an excuse generator in our 41 programs. See if you can create a book title generator using some of the common book title forms such as:

A ..... (1) ..... in the ..... (2) .....

List (1)	List (2)
flash	pan
day	life
scream	wilderness

giving such variations as

A Flash in the Wilderness and
A Day in the Pan.

Some other title forms to play with are: The ..... (1) of the ..... (2) ..... and To ..... (1) ..... a ..... (2) ..... .

15 *A Code Maker*

Create a program that will scramble the whole alphabet (instead of just a word as in our Word Scrambler) and then turn that scrambling into a code. The program should let you put in a word or sentence and give you the coded version. It should also let you put in a message in that code and translate it back into English.

$6.95

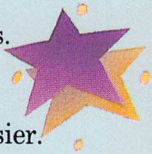
41½ FUN PROJECTS FOR THE COMMODORE 64®

Dale Disharoon & Herbert Kohl

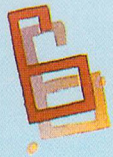
Do you remember the day you brought your shiny new Commodore 64 home? What an exciting time! At first it seemed as if you could think of thousands of ways it would make your life easier, better, and more fun. However, if you're like most people, as the months rolled by and the novelty of owning a computer subsided, routine chores slowly replaced that initial anticipation.

41½ FUN PROJECTS FOR THE COMMODORE® gives you back that wonderful excitement. One of the first Creative Pastimes books, from the Reston Computer Group, this fantastic book shows you 41½ good reasons to fall in love with your Commodore all over again.

In a burst of creative energy, Dale Disharoon and Herbert Kohl deliver some fresh ideas that you'll find easy to key in and modify. The 41½ projects range all the way from a dating program to a mini word processor. You'll be absolutely obsessed with a Dungeons and Dragons dice rolling program, and you'll probably want to take full advantage of the innovative strategy and music games. Practically speaking, programs such as a guitar tuner and chord teacher, a metric converter, and a comparison shopping program truly make your life easier.



A Creative Pastimes Book
RESTON PUBLISHING COMPANY, INC.
A Prentice-Hall Company
Reston, Virginia



Commodore 64 is a registered trademark of Commodore Business Machines, Inc.

0-8359-2102-6