ohio scientific's SMALL SYSTEMS JOURNAL

Between these covers you will find a wide variety of software ranging from business applications to games and hardware utilities and much more. These widely varied programs highlight many different aspects of OSI's computer systems. The individual program descriptions were limited in order to allow the maximum amount of software for you the user.

HOBBIT QUIZ ----

This program will run on all machines. It is a short quiz about Tolkien's book "THE HOBBIT" .

```
 3 NR=4:NQ=4
 5 FORI=1TO30:PRINT:NEXT
 10 FORI=1TO9:READQ$:PRINTQ$:NEXT
 11 FORI=1TO14:PRINT:NEXT
 12 GOSUB1000
 15 FORI=1TONQ
 17 FORIO=1TO30:PRINT:NEXT
 20 READQ$,RA,A$(1),A$(2),A$(3),A$(4),D$(1),D$(2),D$(3),D$(4)
 30 PRINTQ$:PRINT:PRINT:FORJ=1TONR:PRINTJ")",A$(J):NEXT
 35 PRINT:INPUT"WHAT IS YOUR ANSWER";A
 40 IFA>NRORA<>INT(A)ORA<1THENPRINT"WHAT?":GOTO35
 45 IFA=RATHEN100
 47 FORL=1TOWC:IFW(L)=ATHENPRINT"NO, YOU TRIED "A$(A):PRINT:GOTO30
 48 NEXTL
 50 PRINTD$(A):PRINT"TRY AGAIN"
 55 WC=WC+1:IFWC=NRTHENWC=0
 60 W(WC)=A
 65 GOTO35
 100 PRINT:PRINT"RIGHT! "A$(A)" IS CORRECT.":PRINTD$(A)
 110 GOSUB1000
 120 FORL=1TONR:W(L)=0:NEXT
 200 NEXTI
 210 PRINT:PRINT"YOU WON!"
 220 PRINT"YOU HAVE ANSWERED ALL"NQ"QUESTIONS CORRECTLY.
 230 INPUT"DO YOU WANT TO PLAY AGAIN";Q$
 240 IFLEFT$(Q$,1)="Y"ORLEFT$(Q$,1)="Y"THENCLEAR:GOTO3
 250 PRINT:PRINT"THANK YOU FOR PLAYING WITH ME. GOOD-BYE.
 400 END
 500 DATA"THIS IS A QUESTION AND ANSWER GAME ABOUT THE HOBBIT, BY
 510 DATAJ. R. R. TOLKIEN.,""
 520 DATAI WILL ASK YOU QUESTIONS ABOUT THE BOOK. YOU WIN WHEN YOU
 530 DATAHAVE GOTTEN ALL THE QUESTIONS RIGHT. IF YOU NEED HELP ASK
 540 DATAYOUR TEACHER. REMEMBER THAT A COMPUTER IS NOT VERY SMART.
 550 DATAAFTER ALL I AM ONLY A MACHINE!,""
 560 DATAWHEN YOU ARE READY TO BEGIN TYPE 'R' AND THEN PUSH 'RETURN'.
 600 DATAWHICH ONE DO YOU THINK IS THE MAIN CHARACTER?
 605 DATA2,FRODO,BILBO,GANDALF,BARD
 610 DATA"NO, FRODO IS ONLY IN THE LORD OF THE RINGS BY TOLKIEN."
 615 DATABILBO IS THE ONLY ONE WHO IS IN EVERY CHAPTER.
 620 DATA"GOOD GUESS, BUT GANDALF IS NOT IN THE STORY AS MUCH AS BILBO"
 630 DATA"BARD SHOOTS SMAUG, BUT HE DOES NOT DO MUCH ELSE."
 650 DATAWHAT KIND OF CREATURE WAS BILBO?
 660 DATA3,DWARF,DRAGON,HOBBIT,ELF
 670 DATA"NICE TRY, BUT BILBO WAS EVEN SMALLER THAN A DWARF."
 680 DATA"SMAUG IS THE NAME OF THE DRAGON. BILBO IS NOT A DRAGON!"
 690 DATATHE BOOK IS CALLED THE HOBBIT BECAUSE OF BILBO BAGGINS
 695 DATA"NO, ELVES ARE TALL AND ARE NOT USEUALLY THIEVES"
 700 DATAWHICH ONE CAN TURN INTO A BEAR?
 705 DATA1,BEORN,ELROND,BILBO,THORIN
 710 DATABEORN COMES TO THE BATTLE OF FIVE ARMIES IN THE BEAR-SHAPE
 715 DATA"NO, ELROND IS THE HALF-ELF WHO READS THE SWORDS
 720 DATA"NO, BILBO IS A HOBBIT!"
 725 DATA"NOT QUITE. THORIN IS THE KING OF THE DWARVES."
 750 DATAWHO STEALS THE BIG GEM CALLED THE ARKENSTONE?
 755 DATA3,GANDALF,GOLLUM,BILBO,BALIN
 760 DATA"NO, NOT GANDALF THE WIZARD.
 765 DATA"NOT REALLY. GOLLUM IS THE MONSTER WHO PLAYS THE RIDDLE GAME"
 770 DATA"BILBO GIVES THE ARKENSTONE TO THE ELF-KING."
 775 DATA"SORRY. BALIN IS THE DWARVES' OLD WATCHMAN."
 1000 PRINT:PRINT
 1002 INPUT"TYPE 'R' TO GO ON";R$
 1005 IFR$<>"R"THENPRINT"WHAT?":GOTO1000
 1010 RETURN
```

DECIMAL TO HEXADECIMAL CONVERTER ----

This program will run on all machines. It converts decimal numbers to hexadecimal numbers.

```
5 PRINT"DECIMAL TO HEXDECIMAL CONVERTER"
10 INPUT X
12 PRINT"=";
15 B=4096
20 A=X/4096:GOSUB 30
21 B=256
22 A=X/B:GOSUB 30
23 B=16
24 A=X/B:GOSUB 30
25 B=1
26 A=X/B:GOSUB 30
28 GOTO 150
30 IF A>0 THEN A=INT(A)
40 X=X-(B*A)
50 IF A<10 THEN PRINTA;
60 IF A=10 THEN PRINT"A";
70 IF A=11 THEN PRINT"B";
80 IF A=12 THEN PRINT"C";
90 IF A=13 THEN PRINT"D";
100 IF A=14 THEN PRINT "E";
110 IF A=15 THEN PRINT"F";
120 IF A>15 THEN PRINT"TOO LARGE"
130 RETURN
150 PRINT:GOTO10
```

HEXADECIMAL TO DECIMAL CONVERTER ----

This program will run on all machines. It converts hexadecimal numbers to decimal numbers.

```
3 FORI=1TO20:PRINT:NEXT
5 PRINT"HEXDECIMAL TO DECIMAL CONVERTER"
7 FORI=1TO5:PRINT:NEXT
10 FORI=1TO4
20 PRINT"DIGIT #";I;:INPUT A$(I)
30 IFA$(I)="A"THENA(I)=10:GOTO100
40 IFA$(I)="B"THENA(I)=11:GOTO100
50 IFA$(I)="C"THENA(I)=12:GOTO100
60 IFA$(I)="D"THENA(I)=13:GOTO100
70 IFA$(I)="E"THENA(I)=14:GOTO100
80 IFA$(I)="F"THENA(I)=15:GOTO100
90 A(I)=VAL(A$(I))
100 IFA(I)<0ORA(I)>15ORA(I)<>INT(A(I))THEN20
110 NEXT
120 PRINT:PRINT
200 PRINTA(1)*4096+A(2)*256+A(3)*16+A(4)
210 PRINT:PRINT:PRINT:PRINT:GOTO10
300 END
```

AMAZIN ----

This program will run on all machines (Note: you must use smaller mazes if programed on a C1P.). It will generate a maze to the dimension specified. The generated mazes will always have at least one correct path.

```
 50 PRINT"AMAZIN"
 60 PRINT
 70 PRINT"THIS PROGRAM TAKES A LONG TIME ON LARGE MAZES
 80 PRINT"MAXIMUM SIZE IS 23 BY 23
100 LET A=2
110 DIM W(25,25),V(25,25)
111 FOR I=0 TO 25
112 FOR J=0 TO 25
113 W(I,J)=0:V(I,J)=0
114 NEXT J
115 NEXT I
120 PRINT "YOUR WIDTH";:INPUT H:PRINT "AND LENGTH";:INPUT V
122 PRINT
130 IF H<2 GOTO 132:IF H>23 GOTO 132:IF V<2 GOTO 132:IF V>23 GOTO 132
131 GOTO 150
132 PRINT "MEANINGLESS DIMENSIONS.  TRY AGAIN."
140 PRINT
141 GOTO 120
150 PRINT
151 PRINT
160 LET Q=0
161 LET Z=0
162 LET X=INT (RND(A)*H+1)
163 FOR I=1 TO H
170 IF I=X THEN 173
171 PRINT ":--";
172 GOTO 180
173 PRINT ":  ";
180 NEXT I
190 PRINT ":"
191 LET C=1
192 LET W(X,1)=C
193 LET C=C+1
200 LET R=X
201 LET S=1
202 GOTO 260
210 IF R<>H THEN 240
211 IF S<>V THEN 230
220 LET R=1
221 LET S=1
222 GOTO 250
230 LET R=1
231 LET S=S+1
232 GOTO 250
240 LET R=R+1
250 IF W(R,S)=0 THEN 210
260 IF R-1=0 THEN 530
261 IF W(R-1,S)<>0 THEN 530
270 IF S-1=0 THEN 390
280 IF W(R,S-1)<>0 THEN 390
290 IF R=H THEN 330
300 IF W(R+1,S)<>0 THEN 330
310 LET X =INT(RND(A)*3+1)
320 IF X=1 THEN 790
321 IF X=2 THEN 820
323 IF X=3 THEN 860
330 IF S<>V THEN 340
331 IF Z=1 THEN 370
332 LET Q=1
333 GOTO 350
340 IF W(R,S+1)<>0 THEN 370
350 LET X=INT(RND(A)*3+1)
360 IF X=1 THEN 790
361 IF X=2 THEN 820
362 IF X=3 THEN 910
370 LET X=INT(RND(A)*2+1)
```

```
380 IF X=1 THEN 790
381 IF X=2 THEN 820
390 IF R=H THEN 470
400 IF W(R+1,S)<>0 THEN 470
401 IF S<>V THEN 420
410 IF Z=1 THEN 450
411 LET Q=1
412 GOTO 430
420 IF W(R,S+1)<>0 THEN 450
430 LET X=INT (RND(A)*3+1)
440 IF X=1 THEN 790
441 IF X=2 THEN 860
442 IF X=3 THEN 910
450 LET X=INT(RND(A)*2+1)
460 IF X=1 THEN 790
461 IF X=2 THEN 960
470 IF S<>V THEN 490
480 IF Z=1 THEN 520
481 LET Q=1
482 GOTO 500
490 IF W(R,S+1)<>0 THEN 520
500 LET X=INT(RND(A)*2+1)
510 IF X=1 THEN 790
511 IF X=2 THEN 910
520 GOTO 790
530 IF S-1=0 THEN 670
540 IF W(R,S-1)<>0 THEN 670
541 IF R=H THEN 610
542 IF W(R+1,S)<>0 THEN 610
550 IF S<>V THEN 560
551 IF Z=1 THEN 590
552 LET Q=1
553 GOTO 570
560 IF W(R,S+1)<>0 THEN 590
570 LET X=INT(RND(A)*3+1)
580 IF X=1 THEN 820
581 IF X=2 THEN 860
582 IF X=3 THEN 910
590 LET X=INT(RND(A)*2+1)
600 IF X=1 THEN 820
601 IF X=2 THEN 860
610 IF S<>V THEN 630
620 IF Z=1 THEN 660
621 LET Q=1
622 GOTO 640
630 IF W(R,S+1)<>0 THEN 660
640 LET X=INT(RND(A)*+1)
650 IF X=1 THEN 820
651 IF X=2 THEN 910
660 GOTO 820
670 IF R=H THEN 740
680 IF W(R+1,S)<>0 THEN 740
681 IF S<>V THEN 700
690 IF Z=1 THEN 730
691 LET Q=1
692 GOTO 830
700 IF W(R,S+1)<>0 THEN 730
710 LET X=INT(RND(A)*2+1)
720 IF X=1 THEN 860
721 IF X=2 THEN 910
730 GOTO 860
740 IF S<>V THEN 760
750 IF Z=1 THEN 780
751 LET Q=1
752 GOTO 770
760 IF W(R,S+1)<>0 THEN 780
770 GOTO 910
780 GOTO 1000
790 LET W(R-1,S)=C
800 LET C=C+1
801 LET V(R-1,S)=2
802 LET R=R-1
810 IF C=H*V+1 THEN 1010
811 LET Q=0
812 GOTO 260
820 LET W(R,S-1)=C
830 LET C=C+1
840 LET V(R,S-1)=1
841 LET S=S-1
842 IF C=H*V+1 THEN 1010
850 LET Q=0
851 GOTO 260
860 LET W(R+1,S)=C
870 LET C=C+1
871 IF V(R,S)=0 THEN 880
872 LET V(R,S)=3
873 GOTO 890
880 LET V(R,S)=2
890 LET R=R+1
900 IF C=H*V+1 THEN 1010
902 GOTO 530
910 IF Q=1 THEN 960
920 LET W(R,S+1)=C
921 LET C=C+1
922 IF V(R,S)=0 THEN 940
930 LET V(R,S)=3
931 GOTO 950
940 LET V(R,S)=1
950 LET S=S+1
951 IF C=H*V+1 THEN 1010
952 GOTO 260
960 LET Z=1
970 IF V(R,S)=0 THEN 980
971 LET V(R,S)=3
972 LET Q=0
973 GOTO 1000
980 LET V(R,S)=1
981 LET Q=0
982 LET R=1
990 LET S=1
991 GOTO 250
1000 GOTO 210
1010 FOR J=1 TO V
1011 PRINT "I";
1012 FOR I=1 TO H
1013 IF V(I,J)<2 THEN 1030
1020 PRINT "   ";
1021 GOTO 1040
1030 PRINT "  I";
1040 NEXT I
1041 PRINT
1043 FOR I=1 TO H
1045 IF V(I,J)=0 THEN 1060
1050 IF V(I,J)=2 THEN 1060
1051 PRINT ":  ";
1052 GOTO 1070
1060 PRINT ":--";
1070 NEXT I
1071 PRINT ":"
1072 NEXT J
1074 END
```

A DECISION MAKER ----

This program will run on all machines. It calculates the best choice from a group of items you are considering, taking into consideration factors specified by you.

```
100 REM  A DECISION MAKER
110 REM  WRITTEN BY PHIL FELDMAN AND TOM RUGG - APRIL 1977
120 DIM F$(10),L$(10),F(10),M(10,10),V(10),Z(10):PRINT
130 Y$="Y":PRINT"    I CAN HELP YOU MAKE DECISIONS BY CHOOSING THE"
140 PRINT"BEST POSSIBILITY FOR YOU OUT OF SEVERAL ALTERNATIVES."
150 PRINT"ALL I NEED TO DO IS ORGANIZE INFORMATION YOU ALREADY HAVE."
160 PRINT:PRINT"  WHICH OF THESE BEST DESCRIBES THE TYPE OF"
170 PRINT"DECISION YOU HAVE TO MAKE?":PRINT
180 PRINT"1 - CHOOSE AN ITEM FROM SEVERAL ALTERNATIVES"
190 PRINT"2 - CHOOSE A COURSE OF ACTION FROM SEVERAL ALTERNATIVES"
200 PRINT:INPUT"WHICH TYPE (1 OR 2) DO YOU HAVE TO MAKE?";C
210 IF C<1 OR C>2 THEN 200
220 PRINT:IF C=1 THEN PRINT"    WHAT TYPE OF ITEM IS IT THAT YOU
230 IF C=1 THEN PRINT"NEED TO DECIDE UPON":INPUT S$
240 IF C=2 THEN S$="COURSE OF ACTION
250 PRINT:PRINT"    I WILL NOW NEED A LIST OF EACH ";S$
260 PRINT"THAT YOU ARE CONSIDERING.  PLEASE INPUT THEM ONE AT A
270 PRINT"TIME.  THE ORDER IS NO PARTICULAR IMPORTANCE.
280 PRINT:INPUT"   FIRST, HOW MANY ARE THERE ALTOGETHER";L0
290 IF L0=2 AND L0<=10 THEN 310
300 GOSUB 890:GOTO 280
310 PRINT:FOR I=1 TO L0:PRINT:PRINT"NUMBER ";I;" PLEASE
320 INPUT L$(I):NEXT I:PRINT
330 PRINT"O.K.  THIS IS THE LIST UNDER CONSIDERATION":PRINT
340 FOR I=1 TO L0:PRINTI;TAB(5);L$(I):NEXT I:PRINT:GOSUB 900
350 IF B$<>Y$ THEN 250
360 PRINT:PRINT"   NOW, HOW MANY DIFFERENT FACTORS ARE IMPORTANT
370 PRINT "TO YOU IN CHOOSING A ";S$:INPUT F0
380 IF F0>=1  AND F0<=10 THEN 400
390 GOSUB 890:GOTO 360
400 PRINT:PRINT"   I NEED A LIST OF EACH OF THESE FACTORS
410 FOR I=1 TO F0:PRINT:PRINT" FACTOR NUMBER ";I:INPUT F$(I):NEXT
420 PRINT:PRINT"   NOW LOOK AT THE FOLLOWING LIST OF FACTORS YOU
430 PRINT"HAVE PROVIDED AND DECIDE WHICH IS THE MOST IMPORTANT
440 PRINT:FOR I=1 TO F0:PRINTI;TAB(5);F$(I):NEXT
450 PRINT:PRINT"  WHICH FACTOR (BY NUMBER) IS MOST IMPORTANT?
460 PRINT"(INPUT 0 IF YOU WISH TO CHANGE THE LIST":INPUT F2
470 IF F2=0 THEN 360
480 IF F2<1 OR F2>F0 THEN 450
490 PRINT:PRINT"   O.K.  SUPPOSE WE HAVE A SCALE OF IMPORTANCE THAT
500 PRINT"RANGES FROM 0 TO 10.  LET'S SAY ";F$(F2)
510 PRINT"HAS A VALUE OF 10 SINCE IT IS AT THE TOP OF THE SCALE.  ON
520 PRINT"THIS SCALE, WHAT VALUE WOULD EACH OF THE OTHER FACTORS
530 PRINT"HAVE (DECIMAL NUMBERS ARE O.K.)?":PRINT:FOR I=1 TO F0
540 IF I=F2 THEN 570
550 PRINTF$(I):INPUT F(I):IF F(I)>=0 AND F(I)<=10 THEN 570
560 PRINT"  YOUR INPUT IS NO GOOD.  TRY AGAIN.":GOTO 550
570 NEXT:F(F2)=10:C=0:FOR I=1 TO F0:C=C+F(I):NEXT:FOR I=1 TO F0
580 F(I)=F(I)/C:NEXT:PRINT:PRINT"   NOW CONSIDER HOW EACH ";S$
590 PRINT"RATES WITH RESPECT TO EACH OF THE FACTORS.  WE WILL
600 PRINT"CONSIDER EACH FACTOR SEPARATELY AND THEN RATE EACH
610 PRINT S$;" IN TERMS OF THAT FACTOR ONLY.":PRINT
620 PRINT"LET'S CONSIDER "L$(1);" TO HAVE A
630 PRINT"VALUE OF 10 ON EACH SCALE.  THEN EVERY OTHER ";S$
640 PRINT"WILL BE GIVEN A NUMBER HIGHER OR LOWER THAN 10
650 PRINT"ACCORDING TO HOW MUCH BETTER OR WORSE THAN
660 PRINTL$(1);" YOU THINK IT IS.":FOR I=1 TO F0:PRINT
670 PRINT" CONSIDERING ";F$(I);" ONLY, AND
680 PRINT"ASSUMING ";L$(1);" HAS A VALUE OF 10,
690 PRINT"WHAT VALUE WOULD YOU GIVE TO EACH OF THE FOLLOWING:"
700 FOR J=2 TO L0
710 PRINTL$(J):INPUT M(J,I):IF M(J,I)>=0 THEN 730
720 PRINT"   C'MON.  NO NEGATIVE NUMBERS.  TRY AGAIN.":GOTO 710
730 NEXT J:PRINT:M(1,I)=10:NEXT I:FOR I=1 TO F0:C=0:FOR J=1 TO L0
740 C=C+M(J,I):NEXT J:FOR J=1 TO L0:M(J,I)=M(J,I)/C:NEXT J:NEXT I
750 FOR J=1 TO L0:V(J)=0:FOR I=1 TO F0:V(J)=V(J)+M(J,I)*F(I)
760 NEXT I:NEXT J:FOR I=1 TO 10:Z(I)=I:NEXT:C=L0-1:FOR J=1 TO L0
770 FOR I=1 TO C:N1=Z(I):N2=Z(I+1):IF V(N1)>V(N2) THEN 790
780 Z(I+1)=N1:Z(I)=N2
790 NEXT I:NEXT J:C1=Z(1):C2=Z(2):C=100/V(C1):FORJ=1 TO L0
800 V(J)=C*V(J):NEXT:PRINT:PRINT:D=V(C1)-V(C2)
810 PRINT"  WELL, YOUR BEST CHOICE SEEMS TO BE ";L$(C1)
820 IF D<10 THEN PRINT" BUT IT'S PRETTY CLOSE.
830 IF D<20 AND D>=10 THEN PRINT"BY A FAIR MARGIN.
840 IF D>=20 THENPRINT"BY A GOOD-SIZED MARGIN.
850 PRINT:PRINT"   HERE'S THE FINAL LIST WITH ";L$(C1)
860 PRINT"GIVEN A VALUE OF 100 AND THE OTHERS SET ACCORDINGLY
870 PRINT:PRINT:PRINT"RATING",S$:FOR J=1 TO L0:C=Z(J)
880 PRINT V(C),L$(C):NEXT:END
890 PRINT"   THE NUMBER MUST BE BETWEEN 2 AND 10":RETURN
900 INPUT"   IS THIS CORRECT (YES OR NO)";A$
910 B$=LEFT$(A$,1):RETURN
```

LEAST SQUARE ----

This program will run on any machine. It calculates the "best fit" straight line through a set of given data points (there must be at least three data points) . Then it will give you the equation of the line with the line's deviations.

```
10 REM*** PAUL JOVIAK ***
15 REM**LEAST SQUARE***
20 CLEAR
40 DIMZ(100)
50 DIMX(100):DIMY(100):DIML(100)
60 FORI=1TO100:X(I)=0:Y(I)=0:NEXTI
90 FORI=1TO32:PRINT:NEXTI
100 PRINT"THIS PROGRAM WILL":PRINT"CALCULATE THE EQUATION"
105 PRINT"OF A LINE IN THE FORM":PRINT"OF 'Y=MX+B' FROM DATA"
110 PRINT"POINTS. ":PRINT
115 PRINT"YOU NEED AT LEAST 3":PRINT"DATA POINTS"
120 PRINT:PRINT:PRINT"HOW MANY POINTS ARE"
123 INPUT"BEING CONSIDERED";A
125 IFA<3ORA>100THEN115
130 PRINT:PRINT:PRINT:FORI=1TOA
140 PRINT"POINT #";I;:INPUT"'X,Y'";X(I),Y(I)
150 PRINT:PRINT:NEXTI
155 FORI=1TO30:PRINT:NEXT
170 PRINT:PRINT:FORI=1TOA
180 PRINT"POINT #";I
183 PRINT
185 PRINT"X VALUE";X(I)
190 PRINT"Y VALUE";Y(I)
191 PRINT
193 GOSUB200:NEXTI
195 INPUT"LIST AGAIN";Y$:IFLEFT$(Y$,1)="Y"THEN170
198 GOTO300
200 INPUT"CORRECT";Y$
205 PRINT:PRINT
210 IFLEFT$(Y$,1)="Y"THENRETURN
230 PRINTX(I);",";Y(I)
235 PRINT:PRINT
240 INPUT"CHANGE TO 'X,Y'";X(I),Y(I)
245 PRINT:PRINT
250 RETURN
300 FORI=1TOA
310 S1=S1+X(I):S2=S2+Y(I)
330 S3=S3+X(I)^2:S5=S5+X(I)*Y(I)
335 S9=S9+Y(I)^2
350 NEXTI
355 S6=S1^2:G=A*S3-S6
356 IFG=0THENPRINT"ERROR":GOTO1000
360 M=(A*S5-S1*S2)/G
370 B=(S2-M*S1)/A
400 FORI=1TO32:PRINT:NEXT
500 PRINT"THE SLOPE IS":PRINTM
505 PRINT
510 PRINT"THE INTERCEPT IS":PRINTB
520 PRINT:PRINT
550 IFM=0THEN1000
555 INPUT"READY";Y$
600 FORI=1TOA
610 Z(I)=M*X+B
620 S=S+(Y(I)-Z(I))^2
630 NEXTI
640 Y=SQR(ABS(S/A-1))
650 FORJ=1TOA
660 L(J)=M*X(J)+B
670 S4=S4+(Y(J)-L(J))^2
690 NEXTJ
710 G1=(A-2)*G
720 M1=SQR(ABS(A*S4/G1))
730 B1=SQR(ABS(S4*S3/G1))
735 FORI=1TO20:PRINT:NEXT
740 PRINT"DEVIATIONS:"
745 PRINT
750 PRINT"SLOPE:":PRINTM1
755 PRINT
760 PRINT"INTERCEPT:":PRINTB1
765 PRINT
770 PRINT"Y:";Y
780 R=S5-(S1*S2)/A
790 R1=S3-S1^2/A:R2=S9-S2^2/A
800 R3=R1*R2
810 C=R/(R3^.5)
840 PRINT:PRINT:PRINT"CORRELATION COEFFICIENT:"C
900 PRINT:PRINT
1000 INPUT"AGAIN";A$
1010 IFLEFT$(A$,1)="Y"THEN10
49999 END
```

HEXADECIMAL-DECIMAL-OCTAL CONVERTER ----

This program can be run on all machines. It converts hexadecimal numbers to decimal numbers, decimal numbers to hexadecimal numbers, octal numbers to decimal numbers, and decimal numbers to octal numbers.

```
1 REM HEX TO DEC
5 PRINT:PRINT "THIS IS NUMCON":PRINT
10 PRINT:INPUT"HEX ";H$:IFH$="0"THEN100
20 PRINT:D=0
30 X=ASC(LEFT$(H$,1)):X=X-ASC("0"):IFX>9THENX=X-7
40 D=D*16+X:H$=MID$(H$,2):IFH$<>""THEN30
50 PRINT:PRINTD:PRINT:GOTO10
100 REM DEC TO HEX
110 PRINT:HX$="0123456789ABCDEF"
120 INPUT"DEC ";D:IFD=0THEN200
130 H$="":FORI=1TO4:X=INT(D/16):H$=MID$(HX$,D-X*16+1,1)+H$:D=X:
131 NEXT I
140 PRINT:PRINT "$"+H$:PRINT:GOTO120
200 REM OCTAL TO DEC
210 PRINT:INPUT"OCTAL ";OC$:IFOC$="0"THEN300
220 D=0
230 X=ASC(LEFT$(OC$,1)):X=X-ASC("0"):IFX>9THENX=X-7
240 D=D*8+X:OC$=MID$(OC$,2):IFOC$<>""THEN230
250 PRINT:PRINTD:PRINT:GOTO210
300 REM DEC TO OCTAL
310 PRINT:OCT$="01234567"
320 INPUT"DEC ";D:IFD=0THEN5
330 O$="":FORI=1TO6:X=INT(D/8):O$=MID$(OCT$,D-X*8+1,1)+O$
340 D=X:NEXT I
350 PRINT:PRINTCHR$(64)+O$:PRINT:GOTO320
```

DECIMAL TO BINARY TUTOR'S AID ----

This program can be run on any machine. It converts decimal numbers to binary numbers.

```
10 REM DECIMAL TO BINARY TUTOR'S AID
20 POWER$=" 15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00"
30 HALFBY$=" 08 04 02 01 08 04 02 01 08 04 02 01 08 04 02 01"
35 BINARY$="01":BASE=2
38 FORX=1TO33:PRINT:NEXT
40 INPUT"STARTING VALUE ";Y:Y=Y-1
45 Y=Y+1:DECIMAL=Y
50 IFDEC<0 OR DEC>65535 OR DEC<>INT(DEC)THENPRINT:PRINT "WHAT":GOTO 40
80 FORI=1TO16:TEMP=INT(DECIMAL/BASE)
100 ANSWER$="  "+MID$(BINARY$,DECIMAL-TEMP*BASE+1,1)+ANSWER$
110 DECIMAL=TEMP:NEXT
150 FORX=15TO0STEP-1
160 PRINTPOWERS$
170 PRINTHALFBY$
180 FORX=1TO48:PRINT TAB(X); "-";:NEXT:PRINT
190 PRINTANSWER$; "  DECIMAL VALUE = ";Y
200 FORX=1TO48:PRINT TAB(X); "-";:NEXT:PRINT
210 PRINT:PRINT:ANSWER$="":GOTO45
```

RESISTOR ---

This program converts to and from the color codes on resistors. The program will run on all machines.

```
10 INPUT"DO YOU WANT DIRECTIONS";G$
12 IF ASC(G$)=78 THEN 29
14 PRINT:PRINT:PRINT"      A RESISTOR HAS 4 TO 7 COLOR BANDS ON"
15 PRINT:PRINT"IT.  THE FIRST 3 ONLY ARE THE RESISTANCE IN OHMS"
16 PRINT:PRINT"THE LAST GIVE INFORMATION ABOUT WATTAGE AND "
17 PRINT:PRINT"VARIANCE.":PRINT"   THE CORRECT END TO START IS THE "
18 PRINT:PRINT"OPPOSITE OF THE GOLD AND SILVER BANDS"
19 PRINT:PRINT"AND CAN BE BLACK, BROWN, RED, ORANGE,"
20 PRINT:PRINT"YELLOW, GREEN, BLUE, GREY, OR WHITE."
29 PRINT:PRINT:PRINT
30 PRINT"THE FIRST 2 BANDS ARE THE BIGGEST AND THE"
32 PRINT
35 PRINT"SECOND BIGGEST DIGITS, AND THE THIRD BAND"
37 PRINT
40 PRINT"TELLS THE NUMBER OF ZERO'S AFTER THEM"
42 PRINT:PRINT:PRINT
45 INPUT"COLORS FROM NUMBERS";G$
47 IF ASC(G$)=89 THEN250
50 PRINT:PRINT:PRINT:PRINT"COLORS (3 ONLY)"
53 INPUT A$,B$,C$
55 FOR N=1 TO 3
60 IF N<>1 THEN 80
65 G$=A$:GOSUB 2000
70 A$=G$
80 IF N<>2 THEN 100
95 G$=B$:GOSUB 2000
90 B$=G$
100 IF N<>3 THEN 130
105 G$=C$:GOSUB 2000
110 C$=G$
130 NEXT N
140 IF C$="0" THEN C$=" "
145 IF C$="1" THEN C$="0"
150 IF C$="2" THEN C$="00"
155 IF C$="3" THEN C$="000"
160 IF C$="4" THEN C$="0000"
165 IF C$="5" THEN C$="00000"
170 IF C$="6" THEN C$="000000"
175 IF C$="7" THEN C$="0000000"
180 IF C$="8" THEN C$="00000000"
185 IF C$="9" THEN C$="000000000"
186 IF C$="SILVER" THEN C$="*10↑-2"
187 IF C$="GOLD" THEN C$="*10↑-1"
189 PRINT:PRINT:PRINT
190 PRINTA$;B$;C$;"  OHMS"
200 INPUT"ANY MORE";G$
210 IF ASC(G$)=89 THEN 45
215 END
250 INPUT"RESISTANCE IN OHMS";R
252 IF R<10 THEN 400
255 R0=INT(LOG(R)/LOG(10))
257 S=.03
260 R1=INT(R/10↑R0+S)
270 R2=INT((R/10↑R0+S-R1)*10)
280 FOR N=1 TO 3
285 IF N<>1 THEN 295
290 G=R0-1:GOSUB 3000
```

```
292 R$=G$
295 IFN<>2 THEN 310
300 G=R1:GOSUB 3000
305 R1$=G$
310 IF N<>3 THEN 325
315 G=R2:GOSUB 3000
320 R2$=G$
325 NEXT N
330 PRINT:PRINT:PRINT"IN THIS ORDER, THE COLORS ARE :"
335 PRINT:PRINTR1$,R2$,R$
340 PRINT:PRINT:PRINT
345 GOTO 200
400 IF R<1 THEN 500
420 R1$="BLACK":G=R:R$="BLACK"
425 GOSUB 3000:R2$=G$
430 GOTO 330
500 RL=ABS(INT(LOG(R)/LOG(10)))
505 RS=R*10↑RL
506 R$="?????"
507 IF RL=1 THEN R$="GOLD"
508 IF RL=2 THEN R$="SILVER"
510 RV=INT(RS)
515 IF INT(RS*10-RV*10)=0 THEN 570
520 RR=INT(RS*10-RV*10)
525 G=RS:GOSUB 3000
530 R1$=G$:G=RR
535 GOSUB 3000
540 R2$=G$
550 GOTO 330
570 R1$="BLACK":G=RS:GOSUB 3000
575 R2$=G$
580 GOTO 330
2000 REM COLOR FINDER
2005 IF G$="BLACK" AND N=1 THEN G$=" "
2010 IF G$="BLACK"  THEN G$="0"
2015 IF G$="BROWN" THEN G$="1"
2020 IF G$="RED" THEN G$="2"
2025 IF G$="ORANGE" THEN G$="3"
2030 IF G$="YELLOW" THEN G$="4"
2035 IF G$="GREEN" THEN G$="5"
2040 IF G$="BLUE" THEN G$="6"
2045 IF G$="VIOLET" THEN G$="7"
2050 IF G$="GREY" THEN G$="8"
2055 IF G$="WHITE" THEN G$="9"
2060 RETURN
3000 REM COLORS FR. #'S
3002 RESTORE
3005 FOR I=0 TO G:READ G$:NEXT I
3020 RETURN
4000 DATA "BLACK","BROWN","RED","ORANGE","YELLOW"
4005 DATA "GREEN","BLUE","VIOLET","GREY","WHITE"
```

NIGHT OF THE LIVING VOLKSWAGENS ----

THIS PROGRAM WILL RUN ON THE CIP AND SUPERBOARD II ONLY. YOU ARE DRIVING THE CAR MOVING UP THE SCREEN, BUT WATCH OUT THE CARS COMING TOWARD YOU ARE FIRING AT YOU. YOU MOVE RIGHT BY DEPRESSING "2", LEFT BY DEPRESSING "1", AND STRAIGHT BY DEPRESSING BOTH. YOU CAN FIRE BACK BY DEPRESSING THE "3" THE CLOSING NUMBER IS HOW MANY CARS YOU SHOT.

```
1 REM *** ROBERT L.
2 REM *** COPPEDGE
3 KE=57088:POKE530,1
4 POKEKE,64
5 INPUTZ1:FORX=1TO35:PRINT:NEXT
6 POKE54117,32
10 FORX=53355TO54219STEP32
20 POKEX,147:POKEX+5,146
30 NEXT
35 DP=53996:POKEDP,0
50 RP=INT(RND(Z1)*4+1)+53355
51 FORY=1TO30:POKERP-RC,32
52 RP=RP+INT(RND(Z1)*2+1)*32:POKERP,1
53 FORX=1TOZ:NEXT:Z=Z*.8788
54 IFRP=DPTHEN60
58 RC=0:GOTO80
60 FORX=1TO150:POKERP,X:NEXT
61 POKERP,0:Z=75:POKERP-32,32
62 FORX=1TO500:NEXT
65 GOTO50
80 POKEKE,64:G=PEEK(KE)
81 IFG=126THEND=-1
82 IFG=190THEND=1
83 IFG=62THEND=-32
84 IFG=254THEND=32
85 IFG=246THEN10000
92 IFPEEK(DP+D)=32THENDP=DP+D:POKEDP-D,32
93 IFDP<53355THENDP=DP+27*32
94 IFDP>54224THENDP=DP-27*32
95 POKEDP,0:D=0
100 FI=RND(Z1)*5
110 IFFI>4THEN150
120 GOTO490
150 FORX=1TO24:POKERP+32*X,139
151 IFRP+32*X=DPTHEN190
160 NEXTX:FORX=1TO24:POKERP+32*X,32:NEXT
170 GOTO490
190 FORY=1TOX:POKERP+Y*32,32:NEXT
200 FORX=1TO150:POKEDP,X:NEXT:POKEDP,32
205 PRINTKI:KI=0
210 CLEAR:RUN
490 IFI>0THEN700
500 H=INT(RND(Z1)*23)+53363
550 H(1)=INT(RND(Z1)*3+13)
600 I=1
700 POKEH,32:H=H+64
710 POKEH,H(1)
720 IFH>54233THENPOKEH,32:I=0
800 Q=INT(RND(Z1)*6-6)
801 IFQ=5THENRC=1
802 IFQ=-5THENRC=-1
820 RP=RP+RC
850 IFPEEK(RP)<>32THENRP=RP-RC:RC=0
860 IFRC<>0THENPOKERP-RC,32
900 NEXT
1000 POKERP,32:GOTO50
10000 FORX=1TO30:POKEDP-32*X,140
10010 IFDP-32*X=RPTHENFORY=1TOX:POKEDP-32*Y,32:NEXT:FORY=1TO150:
11020 IFDP-32*X=RPTHENPOKERP,Y:NEXT:POKERP,32:KI=KI+1:GOTO50
11030 NEXTX:FORX=1TO30:POKEDP-32*X,32:NEXTX
11040 GOTO93
```

SANTA AND EIGHT REINDEER ----

THIS PROGRAM WILL RUN ON THE C1P AND SUPERBOARD II ONLY. IT IS EIGHT TINY REINDEER FOLLOWED BY SANTA AND HIS SLEIGH, WISHING YOU SEASON'S GREETINGS.

```
1 REMK. STEPHENS&P. JOVIAK:  SANTA:  IP
2 DIMY(32):DIMZ(32):DIMW(32):DIMV(32):DIME(27):DIMF(27)
3 I=1:H=4:DIMP(22):DIMO(22)
4 FORX=ITOH:READZ(X):NEXT:FORX=ITOH:READV(X):NEXT
5 FORX=ITOH:READY(X):NEXT:FORX=ITOH:READW(X):NEXT
7 IFH=32THEN11
8 I=I+4:H=H+4:RESTORE:GOTO4
11 GOTO1450
91 FORX=1TO5:READA(X):NEXT:FORX=1TO5:READB(X):NEXT
92 FORX=1TO9:READC(X):NEXT:FORX=1TO5:READD(X):NEXT
93 FORX=1TO27:READE(X):NEXT:FORX=1TO27:READF(X):NEXT
94 FORX=1TO22:READP(X):NEXT:FORX=1TO22:READO(X):NEXT
95 FORK=-1TO-12STEP-1:L=K+53943:H=53596+K
102 FORX=1TO27:POKEH+E(X),F(X):NEXT
105 FORX=1TO5:POKEL-A(X),D(X):NEXT
106 FORX=1TO5:POKEL-A(X)-1,D(X):NEXT
110 FORX=1TO5:POKEL-A(X)+1,32:NEXT
115 FORX=1TO22:POKEP(X)+H,O(X):NEXT
120 FORX=1TO12:POKEL+X,162:POKEL+X+32,188:POKEL+X-96,164:NEXTX
129 POKEL+13,32:POKEL+45,32:POKEL-83,32:L=L-61
150 FORX=1TO5:POKEL+X,B(X):NEXT:POKEL+6,32:L=L+30
182 FORX=1TO9:POKEL+X,C(X):NEXT:POKEL+10,32:IFK=-12THEN1300
1150 FORX=1TO22:POKEP(X)+H,32:NEXT
1200 FORX=1TO27:POKEH+E(X),32:NEXT
1300 NEXTK:RETURN
1450 PRINT:PRINT" O. S. I.  COMPUTERS SAY. . ":FORX=1TO20:PRINT:NEXT
1500 FORI=1TO50
1510 Q=(54100-53432)*RND(X)+53432
1515 IFPEEK(Q)<>32THEN1510
1520 POKE54117,32:POKEQ,46:NEXT
1999 FORC=1TO9
2000 FORX=4TO32STEP4
3000 L=53856
3001 L=L-(X+6)
3007 FORI=9TO12:POKEL+I,128:NEXT
3008 POKEL-128,240:POKEL-125,240
3009 POKEL-95,240:POKEL-94,240
3010 POKEL-63,253
3011 POKEL-33,227
3020 POKEL-32,226
3100 FORI=1TO8:POKEL+I,42:NEXT:FORI=2TO8:POKEL+I+32,42:NEXT
3111 POKEL-34,173
4150 R=-(X+6)
4160 X=X/4
4200 POKEY(X)+R,Z(X):POKEY(X)+V(X)+R,W(X)
4201 POKEY(X)+6+R,Z(X):POKEY(X)+V(X)+6+R,W(X)
4202 FORI=1TO101:NEXT
4210 POKEY(X)+R,32:POKEY(X)+V(X)+R,32:POKEY(X)+V(X)+6+R,32
4211 POKEY(X)+6+R,32
4300 X=X*4
4510 POKEL-34,32
4550 POKEL-128,32:POKEL-125,32:POKEL-95,32:POKEL-94,32
4560 POKEL-63,32:POKEL-33,32:POKEL-32,32
4570 FORI=1TO8:POKEL+I,32:POKEL+I+32,32:NEXT
4580 IFC=9ANDX=24THEN4600
4581 NEXTX,C
4600 FORI=1TO9:POKEL+I,128:NEXT
5000 GOSUB91
7000 DATA 189,143,143,190,31,32,33,33
7001 DATA 53921,53921,53922,53923
7002 DATA 189,143,190,190
8000 DATA 64,128,97,65,32,77,69,82,82,89,67,72,82,73,83,84,77,65,83
8001 DATA 32,64,64,64,64
9000 DATA 0,-31,-30,1,32,33,2,3,64,65,34,66,98,63,95,31,128,160
9001 DATA 96,130,162,226,194,127,158,190,222
9002 DATA 189,189,190,143,155,155,135,226,168,41,136,136,136,189,135
9003 DATA 143,161,175,205,200,139,139,139,196,201,140,140
9005 DATA 67,68,69,70,104,71,136,168,200,231,230
9006 DATA 100,101,131,133,134,163,164,166,196,198,228
9010 DATA 196,195,135,197,190,198,143,143,189,195,196
9011 DATA 5,6,0,9,10,181,182,17,248,14,1
10000 FORI=1TO5000:NEXT
10001 CLEAR:GOTO1
```

12

RAINDROPS AND WILLY THE WIGGLE ----

This program will run on all graphics machines except the C1P. It is a shower of raindrops and a wiggly worm using graphic techniques.

```
1 REM RAINDROPS & WILLY THE WIGGLE ------8/21/78
10 FORI=1TO27:PRINT:NEXTI
20 A$="O H I O  S C I E N T I F I C  C O M P U T E R S"
30 B$=" "
40 C$="P R O U D L Y  P R E S E N T S . . ."
50 B=54021
60 FORI=1TOLEN(A$)
70 POKEB+I,ASC(MID$(A$,I,1))
80 NEXTI
90 B=54278
100 FORI=1TOLEN(B$)
110 POKEB+I,ASC(MID$(B$,I,1))
120 NEXTI
125 FORI=1TO1000:NEXTI
130 B=54996
140 FORI=1TOLEN(C$)
150 POKEB+I,ASC(MID$(C$,I,1))
160 NEXTI
170 FORI=1TO1000:NEXTI
180 A$="RAINDROPS "
190 FORI=2TOLEN(A$)STEP2
200 FORJ=1TO63STEP8
210 PRINTTAB(J);MID$(A$,I-1,1)
215 FORL=1TO150:NEXTL
220 NEXTJ
230 FORJ=63TO1STEP-8
240 PRINTTAB(J);MID$(A$,I,1)
250 FORL=1TO149:NEXTL
260 NEXTJ
270 NEXTI
280 FORI=1TO10:PRINT:NEXTI
290 PRINTTAB(18);"R  A  I  N  D  R  O  P  S"
300 FORI=1TO14:PRINT:NEXTI
400 C=500
410 FORI=1TOC
420 J=26*RND(1):K=63*RND(1)
430 POKE53440+64*INT(J)+K,ASC("*")
440 NEXTI
450 FORI=53440TO55200:POKEI,32:NEXTI
500 A$=" WILLY THE WIGGLE ":B=53440
510 FORI=1TOLEN(A$)
520 POKEB+I,ASC(MID$(A$,I,1))
530 NEXTI
540 B=B+I:IFB<55103THEN510
550 FORI=1TO29:PRINT:NEXTI
560 B=54303:C=5000:L(1)=53440:L(2)=53440
570 L(3)=53440:L(4)=53440:L(5)=53440
580 I=0
590 A=INT(RND(1)*8)+1
600 ONAGOTO610,620,630,640,650,660,670,680
610 A=B-65:IFA<53440THEN590
615 B=A:POKEB,79:POKEL(5),32:L(5)=L(4):L(4)=L(3)
618 L(3)=L(2):L(2)=L(1):L(1)=B:I=I+1:IFI>500THEN700
619 GOTO590
620 A=B-64:IFA<53440THEN590
625 GOTO615
630 A=B-63:IFA<53440THEN590
635 GOTO615
640 GOTO650
645 GOTO615
650 A=B+1:IFA>55167THEN590
655 GOTO615
660 A=B+63:IFA>55167THEN590
665 GOTO615
690 GOTO615
700 FORI=1TO15:PRINT:NEXTI
705 PRINTTAB(10);"I S N ' T  H E  T A L E N T E D ?
710 FORI=1TO15:PRINT:NEXTI
1000 FORI=1TO10000:NEXTI:GOTO10
```

FUNCTION GRAPHER ----

This program will run on all machines that have a polled keyboard and graphics except the C1P. It will graph any function entered into the program. NOTE: X and Y limits determines what the function will look like on the screen.

```
1 REM **** BY PAUL A. JOVIAK AND ROBERT L. COPPEDGE ****
2 REM *** FUNCTION GRAPHER ***
3 FORX=1TO15:PRINT:NEXT:PRINT"INSTRUCTIONS:":PRINT
4 GOSUB 1000:GOSUB 300
5 PRINT"AFTER 'OK' TYPE '50 DEF FND(X)=' AND THEN"
10 PRINT"THE FUNCTION IN 'X' FOLLOWING THE EQUAL SIGN."
15 PRINT"NOW PUSH RETURN AND TYPE 'RUN 50' FOLLOWED"
20 PRINT"BY PUSHING RETURN.  EXAMPLE: "
25 PRINT:PRINT"50 DEF FND(X)=SIN(X)":PRINT
27 A$=" ":T=0
30 INPUT"DO YOU WISH AN EXAMPLE RUN";A$
35 IFLEFT$(A$,1)="Y"THEN42
40 STOP
42 PRINT
43 PRINT"USE  .1  FOR SCALES FOR THE X AXIS AND Y AXIS
50 DEF FND(X)=SIN(X)
80 INPUT"SCALE FOR X-AXIS";W
85 INPUT"SCALE FOR Y-AXIS";Z
90 A=W*28:B=-A
97 FORI=1TO30:PRINT:NEXTI
100 FORI=53532TO55132STEP64:POKEI,49:NEXTI
105 FORI=54336TO54399:POKEI,45:NEXTI:POKE54364,43
120 X=(A-B)*RND(P)+B
123 KEY=57088:POKE530,1:POKE KEY,128
124 IFPEEK(KEY)=128ORPEEK(KEY)=64THEN163
130 Y=FND(X)
140 Y=INT((Y/Z)+.5):X=INT(X/W)
150 P=54364+X-Y*64
155 IFP>55167ORP<53440THEN120
160 POKEP,46
163 POKE KEY,2
165 F=PEEK(KEY)
170 IFF=16THEN500
173 POKE KEY,128:F=PEEK(KEY)
175 IFF=128THEN80
180 IFF=64THEN5
185 IFLEFT$(A$,1)="Y"THENT=T+1
190 IFLEFT$(A$,1)="Y"ANDT=300THEN1
200 GOTO120
300 REM ***INSTRUCTIONS
305 PRINT:PRINT
310 PRINT"WHEN GRAPH IS ON SCREEN :"
320 PRINT"     PRESS '1' FOR GRAPHING SAME FUNCTION"
330 PRINT"     PRESS '2' FOR GRAPHING A NEW FUNCTION"
340 PRINT"     PRESS ' ' FOR ENDING PROGRAM"
345 PRINT:PRINT
350 RETURN
500 GOTO 1500
1000 PRINT"DEFINE THE X AND Y SCALES BY TYPING THE VALUE PER SPACE"
1001 PRINT"OF X AND Y.  THERE ARE 30 SPACES EITHER SIDE OF THE"
1002 PRINT"Y-AXIS AND 12 SPACES ABOVE AND BELOW THE X-AXIS."
1003 PRINT"IF THE SCALES FOR X & Y DO NOT COMPLY THE GRAPH MAY"
1004 PRINT"NOT APPIER ON THE SCREEN.  IF THIS HAPPENS, PRESS '1'
1005 PRINT"AND ENTER NEW SCALES."
1010 RETURN
1500 END
```

BOUNCER ----

This program will run on all graphics machines except the C1P. It simulates a ball bouncing around the screen at various speeds.

```
1 REM--- BOUNCER
2 REM---B. BENNETT
3 PRINT"SPEED- FROM 1 TO 200"
5 INPUT H1
6 H=INT(200/H1)
7 PRINT"INPUT RANDOM SEED #"
8 INPUT K
9 A=INT(7*RND(K)+65)
11 L1=53282
15 FOR J=53300 TO 55425:POKE J,32:NEXT J
20 C=53440
29 Q2=53497
30 FOR W=Q2 TO 55300 STEP 64:POKE C,79
35 IF C>55168 THEN 170
37 L1=L1+64
40 POKE C-A,32
42 F=0:IF A>64 THEN F=1
45 P=120:IF C>W THEN P=L
47 IF C<L1 AND A<64 THEN 100
50 FOR S=1 TO H :NEXT S
60 IF F=1 AND C>(53440+64*P) THEN A=64-ABS(A-64)
65 IF F=0 AND C>(53440+64*P) THEN A=64+ABS(A-64)
80 C=C+A:NEXT W
90 GOTO 170
100 F=0:P=L:GOTO65
170 POKE C-A,32:G=55173:M=55226
200 A=INT(14*RND(C)+57)
205 F=0:IF A>64 THEN F=1
207 IF C<53440 THEN 280
210 POKE C,79:POKE (C+A),32
211 N1=C
212 IF C>M AND F=0, THEN 300
214 IF C<G AND F=1 THEN 320
215 C=C-A
255 FOR R1=0 TO H:NEXT R1
260 M=M-64:G=G-64
270 GOTO 205
280 L1=53318:Q2=53432
283 POKE N1,32
285 IF A<64 THEN 330
286 IF A>64 THEN A=64-(A-64)
290 GOTO 30
300 A=64+(64-A):GOTO 215
320 A=64-(A-64):GOTO 215
330 A=A+2*(64-A):GOTO 290
```

RANDOM SQUARE ----

This is a short graphics demonstration program. The program is compatable with all graphics machines except the C1P. Note, this program is also listed in the section "Special Hardware", with color conversions.

```
12 REM ** RANDOM SQUARE  **
15 POKE56832,6:PRINT:PRINT:PRINT:PRINT
17 PRINT"       OHIO SCIENTIFIC:"
30 FORI=1TO24:PRINT:NEXT
50 D=4096:B(1)=1:B(2)=-64:B(3)=-1:B(4)=64
80 FORPS=1TO5
90 FORL=1TO2:H=58318
100 FORI=1TO11STEP2
140 H=H+63:FORK=1TO4:Z=H
144 IFK=2THENZ=H+I
145 IFK=3THENZ=H+I+I*-64
146 IFK=4THENZ=H+I*-64
150 FORJ=0TOI-1
160 TS=INT(16*RND(Z))
170 RS=INT(255*RND(TS))
200 POKEZ+J*B(K),TS
220 F=PEEK(57088):IFF<>1THEN800
250 IFL=2THENPOKEZ+J*B(K)-D,32:GOTO500
300 POKEZ+J*B(K)-D,RS
500 NEXTJ,K,I,L
650 NEXTPS
```

OTHELLO ----

This program will run on all machines except the C1P. The game follows conventional rules for OTHELLO. This is a two player game. Enter column, comma, row (example: 3,5) for your moves.

```
1 REM ***ROBERT L.
2 REM ***COPPEDGE
4 FOR R=1 TO 32:PRINT:NEXT
10 FOR W=1 TO 8:FOR V=1 TO 8
11 N(V,W)=45:NEXT V:NEXT W
12 N(4,4)=79:N(5,5)=79:N(4,5)=66:N(5,4)=66:GOTO 115
110 PRINT"PLAYER ";A:PRINT"TYPE YOUR MOVE";:INPUT X,Y
111 IF X>8 OR X<1 OR Y>8 OR Y<1 THEN 110
112 IF N(X,Y)<>45 THEN 1500
113 H=0:GOSUB 600
115 FOR W=1 TO 8
116 FOR V=1 TO 7
117 PRINTTAB(25)CHR$(N(V,W));"
118 NEXT V
120 PRINTCHR$(N(8,W)):PRINT
135 NEXT W
136 A=A+1:IF A=3 THEN A=1
137 GOTO 110
140 GOTO 1000
600 N(X,Y)=66:IF A=1 THEN N(X,Y)=79
601 B=N(X,Y)
602 W=1
603 C=66:IF A=2 THEN C=79
604 IF X=1 OR Y=1 THEN 700
605 IF N(X-W,Y-W)=B THEN 650
610 IF N(X-W,Y-W)=45 OR X-W=1 OR Y-W=1 THEN 700
615 W=W+1:GOTO 605
650 FOR V=1 TO W
655 N(X-V,Y-V)=B
656 IF W<>1 THEN H=1:NEXT V
700 W=1
701 IF Y=1 THEN 800
705 IF N(X,Y-W)=B THEN 750
710 IF N(X,Y-W)=45  OR Y-W=1 THEN 800
715 W=W+1:GOTO 705
750 FOR V=1 TO W
751 N(X,Y-V)=B:IF W<>1 THEN H=1
752 NEXT V
800 W=1
801 IF X=8 OR Y=1 THEN 900
805 IF N(X+W,Y-W)=B THEN 850
810 IF N(X+W,Y-W)=45 OR Y-W=1 OR X+W=8 THEN 900
815 W=W+1:GOTO 801
850 FOR V=1 TO W
851 N(X+V,Y-V)=B:IF W<>1 THEN H=1
852 NEXT V
900 W=1:S=1:IF X=1 THEN 1000
901 IF N(X-W,Y)=B THEN 950
902 IF N(X-W,Y)=45 OR X-W=1 THEN 1000
903 IF X-W=8 THEN 1100
905 W=W+S:GOTO 901
950 FOR V=S TO W STEP S
951 N(X-V,Y)=B:IF W<>S THEN H=1
952 NEXT V
1000 IF S=-1 THEN 1100
1001 S=-1:W=-1:IF X=8 THEN 1100
1002 GOTO 901
1005 W=1:IF X=8 THEN 1100
1100 W=1:IF X=1 OR Y=8 THEN 1200
1101 IF N(X-W,Y+W)=B THEN 1150
1102 IF N(X-W,Y+W)=45 OR X=1 OR Y=8 THEN 1200
1105 W=W+1:GOTO 1101
1150 FOR V=1 TO W
1151 N(X-V,Y+V)=B:IF W<>1 THEN H=1
1152 NEXT V
1200 W=1:IF Y=8 THEN 1300
1201 IF N(X,Y+W)=B THEN 1250
1202 IF N(X,Y+W)=45 OR Y+W=8 THEN 1300
1205 W=W+1:GOTO 1201
1250 FOR V=1 TO W:IF W<>1 THEN H=1
1251 N(X,Y+V)=B:NEXT V
1300 W=1:IF Y=8 OR X=8 THEN 1400
1301 IF N(X+W,Y+W)=B THEN 1350
1302 IF N(X+W,Y+W)=45 OR X+W=8 OR Y+W=8 THEN 1400
1305 W=W+1:GOTO 1301
1350 FOR V=1 TO W:IF W<>1 THEN H=1
1351 N(X+V,Y+V)=B:NEXT V
1400 IF H=0 THEN 1550
1405 N=N+1:IF N=64 THEN 1600
1410 FOR X=1 TO 24:PRINT:NEXT X:RETURN
1500 PRINT"PLACE ALLREADY OCCUPIED":GOTO 110
1550 PRINT"MOVE NOT ALLOWED":N(X,Y)=45:GOTO 110
1600 FOR J=1 TO 8:FOR W=1 TO 8
1605 IF N(J,W)=79 THEN I=I+1
1610 NEXT W:NEXT J:PRINT"O'S SCORE IS:";I
1620 PRINT"B'S SCORE:";64-I
1700 END
```

CHESSBOARD ----

This program will run on all machines except the C1P. This program does not actually play chess or do any error checking, it simply displays the chess board and pieces. The players simply input their moves and the program updates the board. Typing KC or QC would indicate a king side or queen side castle. Typing BOARD will display the board. You move pieces by typing locations (example: D2-D4). You can change a pawn to a queen by typing the change after the move command (example: E2-E1Q).

```
 1 REM***DANIEL GLASSER
 2 REM***CHESS BOARD
 10 DIM B1(8,3),B2(8,8),P$(16)
 30 FOR X=1 TO 2: FOR Y=1 TO 8: READ S
 42 B1(X,Y)=S: B1(X+2,Y)=S: B1(X+4,Y)=S: B1(X+6,Y)=S
 50 NEXT Y: NEXT X:FOR X=1 TO 2:FOR Y=1 TO 8:READ S:B2(X,Y)=S
 90 NEXT Y:NEXTX:FORX=7TO8:FOR Y=1 TO 8:READ S:B2(X,Y)=S:NEXT Y
 100 NEXTX:FORX=1TO16:READS$:P$(X)=S$:NEXTX:GOTO800
 140 IF W=1 THEN 160
 141 INPUT"W)";M$:GOTO 170
 160 INPUT"B)";M$
 170 IF M$="END"THEN2000
 177 IF M$="KC" THEN 250
 178 IF M$="QC" THEN 300
 179 IF M$="BOARD" THEN 799
 180 A$=LEFT$(M$,1):B$=MID$(M$,2,1):A1$=MID$(M$,4,1):B1$=MID$(M$,5,1)
 190 B=VAL(B$):C=VAL(B1$):IFW=1THEN210
 200 W=1:GOTO220
 210 W=0:W2$(K)=M$
 220 P1=ASC(A$)-64:P2=ASC(A1$)-64
 230 U=B2(B,P1):B2(B,P1)=0:B2(C,P2)=U
 235 IF RIGHT$(M$,1)="Q" THEN 900
 240 GOTO 799
 250 IF W=1 THEN 280
 260 W=1
 265 B2(1,1)=0:B2(1,4)=0:B2(1,2)=7:B2(1,3)=4:GOTO799
 280 W=0:B2(8,1)=0:B2(8,4)=0:B2(8,2)=14:B2(8,3)=11:GOTO799
 300 IFW=1THEN330
 310 W=1
 320 B2(1,8)=0:B2(1,4)=0:B2(1,6)=7:B2(1,5)=4:GOTO799
 330 W=0:B2(8,8)=0:B2(8,4)=0:B2(8,6)=14:B2(8,5)=11:GOTO 799
 799 FOR M5=1 TO 4:PRINT:NEXT M5
 800 PRINT"-A----B----C----D----E----F----G----H--"
 801 PRINT
 810 FOR X=1 TO 8: FOR Y=1 TO 8
 820 S1=B2(X,Y)
 830 IF S1=0 THEN S1=B1(X,Y)
 840 S$=P$(S1):PRINTS$;"  ";
 850 NEXTY:PRINT"   ";X:PRINT:PRINT:NEXT X
 851 PRINT"-A----B----C----D----E----F----G----H--"
 855 GOTO 140
 900 IF C=8 THEN B2(C,P2)=8:GOTO 799
 910 IF C=1 THEN B2(C,P2)=15
 912 GOTO 799
 1000 DATA 1,2,1,2,1,2,1,2,2,1,2,1,2,1,2,1
 1010 DATA 4,5,6,7,8,9,5,4,3,3,3,3,3,3,3,3
 1020 DATA 10,10,10,10,10,10,10,10,11,12,13,14,15,16,12,11
 1030 DATA "   "," ##"," WP"," WR"," WN"," WB"," WK"," WQ"," WB"
 1040 DATA " BP"," BR"," BN"," BB"," BK"," BQ"," BB"
 2000 END

OK
```

Word Processor ---

This program is a simple word processor for BASIC-in-ROM computers. The program allows tape storage of text, but requires 10K+ RAM to process full pages of text.

```
10 REM--D. TEWKSBARY; WORD PROCESSOR
21 DIML$(32),C$(13):IP=1:LP=1:C$(0)="IT"
22 C$(1)="IL":C$(2)="DL":C$(3)="IC":C$(4)="DC"
23 C$(5)="U":C$(6)="A":C$(7)="G":C$(8)="PT"
24 C$(9)="PD":C$(10)="C":C$(11)="F":C$(12)="S":C$(13)="R"
40 PRINT:INPUT"COMMAND";A$:FORI=0TO13:IFC$(I)=A$GOTO50
41 IFIP=33THENIP=1
45 NEXTI:GOSUB1400:GOTO40
50 ONI+1GOTO170,100,200,300,400,500,600,700,800,900,1000,1100,1200
55 INPUT"OLD STRING";A$:INPUT"NEW STRING";B$:PRINT
60 PRINT"REPLACE ";A$;" WITH ";B$:PRINT:INPUT"CORRECT";Y$
65 IFLEFT$(Y$,1)="N"GOTO55
70 LA=LEN(A$):FORI=1TO32:FORI2=1TOLEN(L$(I))
80 IFMID$(L$(I),I2,LA)=A$GOTO90
81 NEXTI2,I:GOTO40
90 IFI2=1THENL$(I)=B$+MID$(L$(I),I2+LA):GOTO81
95 L$(I)=LEFT$(L$(I),I2-1)+B$+MID$(L$(I),I2+LA):GOTO81
100 PRINT"("IP")";:INPUTA$:IFA$="**"GOTO40
105 IFIP>32THENPRINT"FILE OVERFLOW":GOTO40
110 C=0:FORI=1TOLEN(A$):IFMID$(A$,I,1)="/"GOTO150
115 L$(IP)=L$(IP)+MID$(A$,I,1)
120 NEXTI:IP=IP+1:GOTO100
150 IFLEN(L$(32))<>0THENPRINT"FILE OVERFLOW":GOTO40
151 IFIP>31THENPRINT"FILE OVERFLOW":PRINT:GOTO40
155 FORI2=31TOIPSTEP-1:L$(I2+1)=L$(I2):NEXT
160 IP=IP+1:L$(IP)="":GOTO120
170 POKE515,254:PRINT"START TAPE":PRINT:INPUTA$
175 PRINT"("IP")";:INPUTA$:IFA$="**"THENPOKE515,0:GOTO40
176 IFIP>32THENPRINT"FILE OVERFLOW":POKE515,0:GOTO40
180 L$(IP)=A$:IP=IP+1:GOTO175
200 PRINT:INPUT"DID YOU WANT TO REMOVE A SERIES";Y$
210 IFLEFT$(Y$,1)="N"THENGOTO250
220 INPUT"LOWEST LINE NUMBER";A
230 INPUT"HIGHEST LINE NUMBER";B
231 IFA>BGOTO220
235 IFA<1ORA>31ORB<1ORB>31THENPRINT"OUT OF RANGE":GOTO40
240 FORI=ATOB:L$(I)="":NEXT
245 FORI=B+1TO32:L$(A+I-B-1)=L$(I):NEXT
246 FORI=32-B+ATO32:L$(I)="":NEXT:GOTO40
250 INPUT"WHICH ONE THEN";A
260 IFA<1ORA>32THENPRINT"OUT OF RANGE":GOTO250
270 L$(A)="":PRINT:INPUT"IS THAT ALL";Y$
280 IFLEFT$(Y$,1)="Y"GOTO40
290 GOTO250
300 PRINT:INPUT"WHAT STRING WOULD YOU LIKE TO INSERT HERE";A$
305 B$(1)="":B$(2)="":C=1
315 FORI=1TOLEN(A$):IFMID$(A$,I,1)="/"THENGOSUB350:NEXTI:GOTO317
316 B$(C)=B$(C)+MID$(A$,I,1):NEXTI
317 IFC+IP>31THENPRINT"OVERFLOW":GOTO40
318 IFC=2THENL$(IP+1)=B$(2)+" "+L$(IP+1)
320 IFLP=1THENL$(IP)=B$(1)+L$(IP):GOTO335
330 L$(IP)=LEFT$(L$(IP),LP-1)+B$(1)+MID$(L$(IP),LP)
335 LP=LP+LEN(B$(1)):IFC=2THENLP=LEN(B$(2))+1:IP=IP+1
340 GOTO40
350 C=C+1:IFC>2THENPRINT"ONLY ONE / ALLOWED":GOTO40
355 L$(IP+1)=MID$(L$(IP),LP)+" "+L$(IP+1)
360 IFLP+I<3THENL$(IP)="":RETURN
```

```
370 L$(IP)=LEFT$(L$(IP),LP-1):RETURN
400 INPUT"HOW MANY CHARACTERS DO YOU WANT TO DELETE";C:IFC=0GOTO40
420 IFLP=1THENL$(IP)=MID$(L$(IP),C+1):GOTO40
430 L$(IP)=LEFT$(L$(IP),LP-1)+MID$(L$(IP),LP+C)
440 GOTO40
500 PRINT:PRINT"("IP")"L$(IP):GOTO40
600 PRINT:INPUT"ADVANCE HOW MANY";A
610 IFIP+A>32ORIP+A<1THENPRINT"OUT OF RANGE":GOTO40
620 IP=IP+A:LP=1
630 PRINT:PRINT"CURRENTLY AT ";IP:GOTO40
700 PRINT:INPUT"WHAT STRING ARE YOU LOOKING FOR";A$
701 INPUT"ON WHICH OCCURANCE";C:IFC=0GOTO701
710 J=0:LA=LEN(A$):FORI=IPTO32:FORI2=1TOLEN(L$(I))
720 IFMID$(L$(I),I2,LA)=A$GOTO740
730 NEXTI2,I:PRINT:PRINT"NOT FOUND":GOTO40
740 J=J+1:IFJ<CGOTO730
750 IP=I:LP=I2:GOTO630
800 PRINT:INPUT"THE PRINT SHOULD START WITH WHICH LINE";L1
810 INPUT"WHERE SHOULD THE PRINT STOP";L2
815 IFL1<1ORL2<1ORL2>32ORL1>32GOTO800
816 INPUT"DID YOU WANT LINE NUMBERS";Y$:IFLEFT$(Y$,1)="Y"GOTO830
820 PRINT:POKE517,1:PRINT"FOR PLAYBACK VIA THE IT MODE, BEGIN THE";
821 PRINT" TAPE AFTER THIS LINE    BEGINS!!"
825 FORI=L1TOL2:PRINTL$(I):NEXT:PRINT"**":POKE517,0:GOTO40
830 FORI=L1TOL2:PRINT"("I")";L$(I):NEXT:GOTO40
900 INPUT"THE PRINT SHOULD START WITH WHICH LINE";L1
901 INPUT"WHERE SHOULD THE PRINT STOP";L2
902 IFL1<1ORL2<1ORL1>32ORL2>32THENPRINT"OUT OF RANGE":GOTO40
905 INPUT"ON WHAT LINE WOULD YOU LIKE YOUR DATA TO START";A
910 INPUT"AND WHAT INCREMENT";B:POKE517,1:PRINT
915 PRINT"FOR PLAYBACK VIA THE IT MODE, BEGIN THE TAPE AFTER THIS";
916 PRINT" LINE    BEGINS!!"
920 FORI=L1TOL2:PRINTA"DATA"CHR$(34);L$(I);CHR$(34):A=A+B:NEXT
925 PRINT"**":POKE517,0:GOTO40
1000 FORI=1TO31:IFLEN(L$(I))<>0GOTO1020
1010 FORI2=I+1TO32:L$(I2-1)=L$(I2):NEXTI2:L$(32)=""
1020 NEXTI:GOTO40
1100 PRINT:INPUT"WHICH LINE DID YOU WANT TO BE ON";A$
1110 A=VAL(A$):IFA=0GOTO40
1120 IP=A:GOTO630
1200 PRINT:INPUT"ENTIRE TEXT";Y$:IFLEFT$(Y$,1)="Y"GOTO1250
1210 INPUT"WHICH LINE";C:IFC<1ORC>31THENPRINT"OUT OF RANGE":GOTO40
1230 L$(C)=L$(C)+" "+L$(C+1):L$(C+1)=""
1240 INPUT"COMPRESS";Y$:IFLEFT$(Y$,1)="Y"GOTO1000
1245 GOTO40
1250 FORI=1TO31STEP2:L$(I)=L$(I)+" "+L$(I+1):L$(I+1)="":NEXT:GOTO1240
1400 PRINT"LIST OF COMMANDS        IL> INSERT LINE"
1410 PRINT" DL> DELETE LINE        IC> INSERT CHARACTER"
1420 PRINT" DC> DELETE CHARACTER    G> GO FOR STRING"
1430 PRINT"  V> VERIFY LINE         F> FIND LINE"
1440 PRINT"  A> ADVANCE             C> COMPRESS"
1450 PRINT"  S> STACK LINE          R> REPLACE STRING"
1460 PRINT" PT> PRINT AS TEXT      PD> PRINT AS DATA"
1470 RETURN
```

Trax ---

This program will run on all OSI computers with 540 video boards. This is a two player game where the one who lasts the longest wins!

```
5 FOR LF=1 TO 32:PRINT:NEXT
10 INPUT "DO YOU WANT INSTRUCTIONS (Y/N)";IANS$
12 IF IANS$="N" THEN 52
20 PRINT" THE OBJECT OF THIS GAME IS NOT TO RUN INTO ANYTHING."
22 PRINT" INCLUDING THE BOUNDARY, THE ENEMIES TRACK,OR YOUR OWN"
24 PRINT" TRACK. ( DO NOT GO BACK ON YOURSELF!)"
26 PRINT" PLAYER 1 STARTS IN THE TOP LEFT HAND CORNER ; "
28 PRINT" PLAYER 2 STARTS IN THE BOTTOM RIGHT HAND CORNER."
30 PRINT"  THE MOVEMENT KEYS ARE AS FOLLOWS :"
32 PRINT
34 PRINT"       W                    O"
36 PRINT"     A S                  K L"
38 PRINT"       Z                    ,"
40 PRINT
42 PRINT"   PLAYER 1              PLAYER 2"
44 PRINT
46 PRINT"  THEY MOVE EACH PLAYER NORTH SOUTH EAST WEST AS APPROPRIATE"
48 PRINT" MAY BOTH OF YOU HAVE EQUAL LUCK!"
50 INPUT "  ARE YOU READY";IANS$
52 D1=1:D2=-1
53 P1=53509:P2=55032
54 POKE 530,1
56 FOR LC=1 TO 32:PRINT:NEXT LC
58 KB=57088
60 DP=30*64
62 BC=161
64 REM SET UP BOARD
66 TL=53313
68 FOR BD=55040 TO 55295:POKE BD,32:NEXT BD
70 FOR BD=TL TO TL+60
72 POKE BD,BC:POKE BD+DP,BC:NEXT BD
74 FOR BD=TL TO TL+DP STEP 64
76 POKE BD,BC:POKE BD+60,BC:NEXT BD
100 REM  NOW PLAY LOOP
110 GOSUB 1000
120 NP=P1+D1
130 IF PEEK(NP)<>32 THEN 500
140 POKE NP,232:P1=NP
150 GOSUB 1000
160 NP=P2+D2
170 IF PEEK(NP)<>32 THEN 600
180 POKE NP,233:P2=NP
190 GOTO 110
500 FP=55027
510 GOTO 610
600 FPp53509
610 FOR FL=1 TO 10
620 POKE FP,73:POKE FP+1,32:POKE FP+3,87:POKE FP+4,73
625 POKE FP+2,32
630 POKE FP+5,78
640 FOR DM=1 TO 50
650 NEXT DM
660 POKE FP,32:POKE FP+1,32:POKE FP+3,32:POKE FP+4,32
670 POKE FP+5,32
680 FOR DM=1 TO 50
690 NEXT DM
700 NEXT FL
710 INPUT "PLAY AGAIN";IANS$
720 IF IANS$="Y" THEN 52
730 STOP
999 REM KBOARD POLLER
1000 POKE KB,2
1010 Q=PEEK(KB)
1020 IF Q=64 THEN D1=-1
1030 IF Q=32 THEN D1=64
1040 POKE KB,4
1050 IF PEEK(KB)=2 THEN D2=64
1060 POKE KB,8
1070 Q=PEEK(KB)
1080 IF Q=128 THEN D1=1
1090 IF Q=2 THEN D2=-1
1100 IF Q<>130 THEN 1120
1110 D1=1:D2=-1
1120 POKE KB,16
1130 IF PEEK(KB)=128 THEN D1=-64
1140 POKE KB,32
1150 Q=PEEK(KB)
1160 IF Q=64 THEN D2=1
1170 IF Q=32 THEN D2=-64
1180 RETURN
```

Star Wars ---

This program is a game that will run on any graphics machine except the C1P. You are controlling a turret and an invader is after you.

```
1 REM STAR WARS
2 REM JERRY DURBAK
19 FORI=1TO30:PRINT:NEXTI
20 POKE2073,98:POKE56832,0
30 GOSUB20000
35 T=57088
40 Z=55168
45 W=55199
50 P=55180
55 XX=32:YY=64:ZZ=128
60 N4=48:N3=48:N2=48:N1=48
70 M4=48:M3=48:M2=48:M1=48
80 EX=161:S2=0
30 U=1:Q=1
97 POKET,128
100 REM MAIN BODY AND SAUCER DRIVER
150 D=INT(24*RND(1)+1)
200 X=53440+64*D
300 Y=53471+64*D
400 ON INT(2*RND(1)+1)GOTO450,1000
450 REM LEFT ENTRANCE
475 I=X-1
500 I=I+1
600 POKEI,4
700 IFI-X=P-ZTHENGOSUB2000
800 GOSUB5000
850 IFQ=2THENQ=1:GOTO150
900 POKEI,32
950 IFI>X+32THENGOTO150
970 GOTO500
1000 REM RIGHT ENTRANCE
1025 I=Y+1
1050 I=I-1
1100 POKEI,4
1200 IFY-I=W-PTHENGOSUB2000
1300 GOSUB5000
1350 IFQ=2THENQ=1:GOTO150
1400 POKEI,32
1500 IFI<Y-32THENGOTO150
1600 GOTO1050
2000 REM SAUCER FIRING
2100 FORK=1TO33-D
2200 POKEI+64*K,149
2300 IFK>2THENPOKEI+64*(K-2),32
2375 IFQ=2THENGOTO2999
2400 IFI+64*K=PTHENR=P:GOSUB3000
2450 IFSW$="2"THENGOSUB5000
2500 NEXTK
2600 POKEI+64*(K-1),32
2700 POKEI+64*K,32
2999 RETURN
3000 REM EXPLOSIONS
3100 EX=161
3200 POKER+64,EX:POKER+64+1,EX:POKER+64+2,EX
3300 POKER+64-1,EX
3400 POKER-3,EX:POKER-2,EX:POKER-1,EX:POKER,EX:POKER+1,EX
3500 POKER-64-2,EX:POKER-64-1,EX:POKER-64,EX:POKER-64+1,EX
```

```
3550 POKER-64+2,EX
3600 POKER-128-1,EX:POKER-128,EX:POKER-128+1,EX:POKER-128+2,EX
3700 POKER-192-3,EX:POKER-192-1,EX:POKER-192,EX:POKER-192+2,EX
4100 POKER-256-2,EX
4200 POKER-64+4,EX
4250 IFU=2THENQ=2
4300 IFEX=32THENGOTO4500
4400 EX=32:GOTO3200
4500 GOSUB10000
4999 RETURN
5000 REM TURRET DRIVER
5100 POKEP,176:POKEP+1,178
5200 C=PEEK(T)
5300 IFC=YYANDP<W-4THENGOTO5700
5400 IFC=ZZANDP>ZTHENGOTO5900
5500 IFC=XXTHENGOSUB7000
5600 GOTO6100
5700 P=P+1:IFP>ZTHENPOKEP-1,32:POKEP,32
5800 GOTO6000
5900 P=P-1:IFP<W-4THENPOKEP+1,32:POKEP+2,32
6000 POKEP,176:POKEP+1,178
6100 RETURN
7000 REM TURRET FIRING
7100 FORF=1TO27
7150 IFU=2THENGOTO7400
7200 POKEP-64*F,149
7300 POKE(P+1)-64*F,149
7400 IFF>2THENPOKEP-64*(F-2),32
7500 IFF>2THENPOKE(P+1)-64*(F-2),32
7600 IFP-64*F=ITHENR=I:U=2:GOSUB3000
7650 IF(P+1)-64*F=ITHENR=I:U=2:GOSUB3000
7700 NEXTF
7800 U=1
7900 POKEP-64*(F-2),32
7950 POKE(P+1)-64*(F-2),32
8000 POKEP-64*(F-1),32
8050 POKE(P+1)-64*(F-1),32
8100 POKEP-64*F,32
8150 POKE(P+1)-64*F,32
8200 RETURN
10000 REM SCORING
10100 IFQ=2THENN3=N3+1
10200 IFQ=1THENM3=M3+1
10300 IFN3=58THENN4=N4+1:N3=48
10400 IFM3=58THENM4=M4+1:M3=48
10500 POKEZ-128,N4:POKEZ-127,N3:POKEZ-126,N2:POKEZ-125,N1
10600 POKEZ-101,M4:POKEZ-100,M3:POKEZ-99,M2:POKEZ-98,M1
10650 IFN4=50RM4=50THENGOTO30000
10700 FORAB=1TO1500:NEXT
10800 POKEZ-128,32:POKEZ-127,32:POKEZ-126,32:POKEZ-125,32
10900 POKEZ-101,32:POKEZ-100,32:POKEZ-99,32:POKEZ-98,32
11000 RETURN
20000 REM INTRODUCTION
20050 IFS2=1THENGOTO21100
20100 PRINT"STAR WARS":FORI1=1TO15:PRINT:NEXTI1
20200 PRINT"DO YOU NEED":INPUT"INSTRUCTIONS (Y/N)";A$
20300 IFLEFT$(A$,1)="N"THENGOTO21100
20400 FORI2=1TO30:PRINT:NEXT
```

```
20500 PRINT"YOU ARE EQUIPPED WITH":PRINT"DOUBLE LASER FIRE AND"
20600 PRINT"MOBILITY.":PRINT:PRINT"YOUR CONTROLS ARE:"
20700 PRINT"1> LEFT":PRINT"2> RIGHT":PRINT"3> FIRE"
20800 PRINT:PRINT"YOUR ENEMY HAS A":PRINT"SINGLE LASER EQUIPPED"
20900 PRINT"WITH COMPUTERIZED ":PRINT"TRACKING."
21000 PRINT:PRINT:PRINT"USE THE FORCE!!"
21050 PRINT:PRINT"YOUR SCORE IS ON THE":PRINT"LEFT.  PLAY IS TO 2000"
21100 PRINT:PRINT"YOU CAN PLAY:":PRINT"1> HARD":PRINT"2> EASY"
21200 PRINT:PRINT"READY??":INPUT"(ENTER A 1 OR 2)";SW$
21500 FORI3=1TO30:PRINT:NEXT
29999 RETURN
30000 REM ENDING AND RERUN
30050 POKEP,32:POKE(P+1),32
30100 FORI5=1TO30:PRINT:NEXT
30150 IFM4=50THENGOTO31000
30200 PRINT"GOOD SHOOTING!!!":PRINT:PRINT:PRINT
30300 IFSW$="2"THENPRINT"TRY THE HARD VERSION."
30400 IFSW$="1"THENPRINT"THE FORCE WAS WITH YOU!"
30500 GOTO31200
31000 PRINT"THE IMPERIAL FORCES":PRINT"WHERE TOO MUCH FOR"
31100 PRINT"YOU!!":PRINT:PRINT:PRINT
31200 PRINT:PRINT:PRINT"WOULD YOU LIKE":INPUT"ANOTHER RUN (Y/N)";A$
31300 IFLEFT$(A$,1)="Y"THENCLEAR:S2=1:GOSUB20000:GOTO35
40000 POKE2073,173:POKE56832,1
49999 END
```

Graphic V ---

This program will run on all graphics machines except the C1P. The program is a graphic pattern generator.

```
2 REM----GRAPHIC V
3 REM--CHALLENGER II
5 FORI=1TO30:PRINT:NEXT:POKE55104,32:POKE56832,0
10 FORM=0TO7:READI(M):NEXT
20 C=INT(RND(1)*100)+128
21 IFINT(RND(1)*7)=1THENC=32
25 IFC>178ANDC<183ORC>210ANDC<215GOTO20
30 P=INT(RND(1)*1800)+53300
35 IFP/64-INT(P/64)>.5GOTO30
40 M=INT(RND(1)*8)
50 P=P+I(M):POKEP,C
60 IFINT(P/32)=P/32ORP>55104ORP<53330GOTO20
70 IFINT(RND(1)*9)<>1GOTO50
80 MN=INT(RND(1)*8):IFI(MN)<>-I(M)THENM=MN:GOTO50
100 GOTO80
110 DATA64,-64,65,-65,63,-63,1,-1
```

ESCAPE ----

This program will run on all graphics machines except the C1P. The program demonstrates randomness and, trial & error of a rat trying to escape from a self made maze.

```
 10 REM  ESCAPE  4/15/79
 20 K1=1:K2=2:K6=64:KH=64:KV=26:K0=0:TT=0:LL=5000
 30 G=1:F=1:DIMBB$(KV,KH)
 40 KA=53536:KB=54399:KC=55136:KD=54336
 50 HH$="-1":GOSUB60000
 70 FORI=1TOPEEK(8996)+2:B=RND(1):NEXT
 80 GOSUB60000:GOSUB20000
160 A$="PRESENTING...":B$="T H E   R A T   R A C E !"
162 A=53545:FORI=ATOA+LEN(A$)-1:B=B+1
164 POKEI,ASC(MID$(A$,B,1)):NEXTI
166 B=0:FORI=1TO1000:NEXT
168 A=54290:FORI=ATOA+LEN(B$)-1:B=B+1
170 POKEI,ASC(MID$(B$,B,1)):NEXTI
180 FORI=1TOKV:FORJ=1TOKH:BB$(I,J)=HH$:NEXTJ:NEXTI
190 GOTO280
200 GOSUB60000:PRINT"THIS RUN TOOK":T;"MOVES":PRINT
222 Q=K0:IFT>TTTHENTT=T
224 IFT<LLTHENLL=T
226 PRINT"THIS WAS RUN NUMBER";G:PRINT
227 PRINT"THE HIGHEST SCORE IN THESE RUNS WAS";TT:PRINT
228 PRINT"THE LOWEST SCORE IN THESE RUNS WAS";LL:PRINT
229 GG=GG+T:AV=INT(GG/G)
230 PRINT"THE AVERAGE RUN TOOK";AV;"MOVES":T=0:PRINT
232 PRINT"(THE RAT GOT OUT";E;"TIMES)":PRINT
240 G=G+1:GG$=STR$(G):FORI=1TO3000:NEXT
280 GOSUB60000:GOSUB20000:C$="READY?":D$="
380 A=54290:B=0:IFF=1THENF=0:GOTO400
382 FORI=ATOA+LEN(C$)-1:B=B+1
384 POKEI,ASC(MID$(C$,B,1)):NEXTI:FORI=1TO1000:NEXT
400 X=INT(RND(K1)*KV):Y=INT(RND(K1)*KH)
410 N=53439+64*X+Y
415 IFN<53504THEN400
470 B=0:FORI=ATOA+LEN(D$)-1:B=B+1
480 POKEI,ASC(MID$(D$,B,1)):NEXTI
1000 D=INT(RND(1)*4+1):A=INT(RND(1)*10+1):X1=X:Y1=Y
1100 ONDGOSUB2000,3000,4000,5000
1110 IFQ<>K1THEN1150
1120 X$="H E   E S C A P E D !"
1122 A=54356:B=0:FORI=ATOA+LEN(X$)-1:B=B+1
1124 POKEI,ASC(MID$(X$,B,1)):NEXTI:FORI=1TO1000:NEXT
1126 E=E+K1:GOTO200
1150 IFX1=XANDY1=YTHENC=C+K1:GOTO1180
1160 C=K0
1180 IFC>15THENGOSUB50000
1200 GOTO1000
2000 FORJ=K1TOA:IFY+K1>KHTHENRETURN
2300 IFBB$(X,Y+K1)=GG$THENRETURN
2400 Y=Y+K1:N=N+K1:POKEN,18:GOSUB10000
2450 GOSUB30000:IFQ=K1THENRETURN
2500 NEXTJ:RETURN
3000 FORJ=K1TOA/K2:IFX+K1>KVTHENRETURN
3300 IFBB$(X+K1,Y)=GG$THENRETURN
3400 X=X+K1:N=N+K6:POKEN,20:GOSUB10000
```

```
3450 GOTO2450
4000 FORJ=K1TOA:IFY-K1<K1THENRETURN
4300 IFBB$(X,Y-K1)=GG$THENRETURN
4400 Y=Y-K1:N=N-K1:POKEN,22:GOSUB10000
4450 GOTO2450
5000 FORJ=K1TOA/K2:IFX-K1<K1THENRETURN
5300 IFBB$(X-K1,Y)=GG$THENRETURN
5400 X=X-K1:N=N-K6:POKEN,16:GOSUB10000
5450 GOTO2450
10000 T=T+K1:BB$(X,Y)=GG$:RETURN
20000 L=INT(RND(1)*255+1)
20010 IFL=32THEN20000
20050 FORI=53504TO53567:POKEI,L:NEXT
20100 FORI=53631TO55167STEP64:POKEI,L:NEXT
20200 FORI=55166TO55105STEP-1:POKEI,L:NEXT
20300 FORI=55104TO53568STEP-64:POKEI,L:NEXT
20400 POKE53536,32:POKE54399,32:POKE55136,32:POKE54336,32
20500 RETURN
30000 IFN=KAORN=KBORN=KCORN=KDTHENQ=1
30010 RETURN
50000 FORD=K1TO4:A=INT(RND(K1)*10+K1)
50100 ONDGOSUB2000,3000,4000,5000
50200 IFY1<>YORX1<>XTHENC=0:RETURN
50300 NEXTD:X$="A H A ! . . . G O T C H A !"
50400 A=54356:B=0:FORI=ATOA+LEN(X$)-1:B=B+1
50500 POKEI,ASC(MID$(X$,B,1)):NEXTI:FORI=1TO1000:NEXT
50600 GOTO200
60000 FORI=1TO32:PRINT:NEXT:RETURN
```

BREAKOUT ----

THIS PROGRAM WILL RUN ONLY ON MACHINES WITH ASCII KEYBOARDS. THIS GAME CAN BE CONVERTED TO A POLLED KEYBOARD SYSTEM. (SEE PROGRAM CONVERTIONS.)

```
1 REM----------------DAVID E. TEWKSBARY
2 REM---------------BREAKOUT
5 INPUT"DO YOU NEED INSTRUCTIONS";Y$:IFLEFT$(Y$,1)="N"GOTO18
6 PRINT"  HOKAY":PRINT:PRINT"THIS IS THE GAME BREAKOUT.":PRINT
7 PRINT" THE IDEA IS TO WIPE OUT ENOUGH BARRIERS TO ALLOW"
8 PRINT"THE BOUNCING BALL TO REACH THE TOP OF THE SCREEN."
9 PRINT"YOUR PADDLE IS ON THE BOTTOM OF THE SCREEN AND IS "
10 PRINT"MOVED IN THE FOLLOWING MANNER:":PRINT
11 PRINT"  1) LEFT":PRINT"  2) RIGHT":PRINT
12 PRINT" IN ORDER TO SERVE A BALL, PRESS 4.  SIX BALLS WILL"
13 PRINT"BE THE LIMIT.  POINTS CAN BE SCORED BY KNOCKING DOWN"
14 PRINT"THE BARRIERS, THE DEEPER ONES BEING MORE VALUABLE."
15 PRINT:PRINT"WHEN YOU ARE READY, TYPE 'R'.":INPUTY$
16 IFY$<>"R"GOTO15
18 FORI=1TO30:PRINT:NEXT:POKE56832,0
19 PRINT"  BALL #          SCORE"
20 FORI=53534TO54942STEP64:POKEI,47:NEXT:POKEI,35
30 FORI=55005TO54977STEP-1:POKEI,95:NEXT:POKEI,35
40 FORI=54912TO53504STEP-64:POKEI,92:NEXT:POKEI,35
50 FORI=53441TO53469:POKEI,42:NEXT:POKEI,35
51 FORI=53505TO53533:POKEI,51:NEXT
52 FORI=53569TO53597:POKEI,50:NEXT
53 FORI=53633TO53661:POKEI,49:NEXT
59 REM---------------INITIALIZE VARIABLES
60 M(1)=-65:M(2)=-64:M(3)=-63:M(4)=63:M(5)=64:M(6)=65
70 BC=79:B0=32:PC=61:T=57343:P=54925
75 POKEP,61:POKE55048,48:POKE55062,48:POKE55063,48
80 REM---------------SERVE
85 IFX=6GOTO1000
86 IFPEEK(T)<>180GOTO86
90 X=X+1:POKE55048,48+X
100 BP=INT(RND(BC)*25)+53827:GOSUB500:L=6
105 REM--------------BALL'S MOTION
110 FORA=1TOL:IFA<>1GOTO230
120 POKEBP,B0:Q=PEEK(BP+M(IB)):IFQ=B0GOTO200
125 IFQ=224GOTO200
130 L=L*.87:IFQ=220ORQ=92THENIB=IB+2:GOTO180
140 IFQ=47ORQ=239THENIB=IB-2:GOTO180
144 Q2=PEEK(BP+64)
145 IFQ2=61ORQ2=253THENIB=INT(RND(BC)*3)+1:GOTO180
150 IFQ=223ORQ=95GOTO85
155 IFQ=234ORQ=42GOTO1000
167 IFQ=241ORQ=49THENZ=Z+1:GOSUB500:GOTO190
168 IFQ=242ORQ=50THENZ=Z+2:GOSUB500:GOTO190
169 IFQ=243ORQ=51THENZ=Z+3:GOSUB500:GOTO190
170 IB=INT(RND(BC)*6)+1
180 Q2=PEEK(BP+M(IB)):IFQ2<>32ANDQ2<>224GOTO170
190 POKE55062,INT(Z/10)+48:POKE55063,Z+48-INT(Z/10)*10
200 BP=BP+M(IB):POKEBP,BC
210 REM--------------PADDLE MOTION
230 C=PEEK(T):IFC=178ANDP<54941GOTO260
250 IFC=177ANDP>54913GOTO270
255 GOTO280
260 P=P-1:IFP>54912THENPOKEP,B0
262 P=P+2:POKEP,PC:GOTO280
270 P=P+1:IFP<54942THENPOKEP,B0
275 P=P-2:POKEP,PC
280 NEXTA:GOTO110
500 POKEBP+M(IB),B0:IB=INT(RND(BC)*3)+4:RETURN
999 REM--------------END MESSAGES
1000 FORI=1TO30:PRINT:NEXT:POKE56832,1
1001 IFQ<>234ANDQ<>32GOTO1040
1010 PRINT"WELL, WELL.  SO NOW I SUPPOSE YOU THINK YOU'RE"
1020 PRINT"TOUGH, EH?  WELL, YOU ONLY HAD";Z;"POINTS!"
1030 PRINT"  SO THERE!!":GOTO1080
1040 PRINT"NOBODY SAID IT WAS GOING TO BE EASY!!"
1050 PRINT"SERIOUSLY THOUGH,";Z;"ISN'T TOO BAD!"
1060 PRINT" (BUT I GUESS THERE'S ALWAYS ROOM FOR IMPROVEMENT.)"
1080 PRINT:INPUT"WOULD YOU LIKE TO TRY AGAIN";Y$
1200 IFLEFT$(Y$,1)="Y"THENRUN18
1210 PRINT:PRINT"HAVE IT YOUR WAY.":END
```

TORPEDO -----

THIS PROGRAM WILL RUN ONLY ON MACHINES WITH ASCII KEYBOARDS. THIS GAME CAN BE CONVERTED TO A POLLED KEYBOARD SYSTEM. (SEE PROGRAM CONVERTIONS.)

```
10 REM--R. VAN SCOY--TORPEDOE
20 INPUT"DO YOU WANT INSTRUCTIONS";A$:IFLEFT$(A$,1)="N"THEN130
25 FORX=1TO15:PRINT:NEXT
30 PRINT"HELLO ADMIRAL,":PRINT"YOUR ORDERS HAVE PUT YOU IN CHARGE OF"
40 PRINT"AN UNDERWATER, ANTI-SUBMARINE INSTALLATION."
50 PRINT"YOU ARE TO DESTROY ALL SUBMARINES WHICH PASS OVER"
60 PRINT"COMMAND!!! BECAUSE OF THE IMPORTANCE OF THIS ASSIGN-"
65 PRINT"MENT, YOUR PREFORMANCE WILL BE RATED, YOU WILL RECEIVE"
70 PRINT"25 MERIT POINTS FOR EACH SUB YOU DESTROY."
75 PRINT"GOOD LUCK ADMIRAL!"
80 PRINT"OH, YES. I ALMOST FORGOT TO TELL YOU THAT DUE TO BUDGET"
85 PRINT"CUTS, YOU MUST MAKE EVERY SHOT COUNT, BECAUSE EACH MISS"
90 PRINT"WILL COST YOU 5 POINTS, AND EVERY SUB THAT GETS AWAY,"
95 PRINT"WILL COST YOU TEN POINTS!!"
100 PRINT"YOU MUST ALSO KNOW THAT THE SUBMARINES ARE EQUIPED WITH"
105 PRINT"AN AUTO-PILOT WHICH IS DESIGNED TO CONFUSE THE ENEMY!"
130 M1$="TORPEDOES:":M2$="SCORE: 000"
150 FORX=1TO5:READT(X):T(X)=T(X)-64:NEXT
170 DIMR(15):FORX=1TO15:READR(X):NEXT
200 PRINT:PRINT:PRINT:INPUT"ENTER A RANDOM SEED";Z1
215 FORX=1TO32:PRINT:NEXT
240 KE=57088:T=55040:C1=49
245 NU=25:SC=0:T1=TP+45
250 TP=53440:H=1344:POKE55104,32
270 FORX=1TOLEN(M1$):POKETP+X,ASC(MID$(M1$,X,1)):NEXT
280 FORX=1TOLEN(M2$):POKETP+X+32,ASC(MID$(M2$,X,1)):NEXT
290 FORX=53504TO53566:POKEX,94:NEXT
300 FORX=TTOT+64STEP15:POKEX,91:POKEX+2,93:
310 POKEX+65,C1:C1=C1+1:NEXT
350 Z=INT(18*RND(Z1))*64+53697:Z2=15
400 FORC=ZTOZ+60
405 POKET1,32:POKET1-1,32:T1=C+R(Z2)
410 POKET1,62:POKET1-1,62
420 S1=INT(NU/10):S2=INT(NU-S1*10):POKETP+12,S1+48:POKETP+13,S2+48
450 S=PEEK(KE)-176
470 IFS<1ORS>5THEN680
500 FORQ=T(S)TOT(S)-HSTEP-64
580 C7=PEEK(Q-64):IFC7=62ORC7=174THENNU=NU-1:GOTO900
600 POKEQ-64,33:POKEQ,32
660 FORX=1TO15:NEXT
670 NEXTQ:NU=NU-1:SC=SC-5:POKEQ,32
675 GOSUB920
680 Z2=INT(14*RND(Z1)+1)
700 NEXTC:POKET1-2,32:POKET1-1,32
750 SC=SC-10:GOSUB920:GOTO350
899 REM---EXPLOSION OF SUB
900 POKEQ,32
910 FORQ1=0TO32:POKET1-1,Q1:POKET1,Q1:NEXT
915 SC=SC+25:GOSUB920:IFNU=0THEN1000
916 GOTO350
920 L=ABS(SC):S3=INT(L/100):S4=INT((L-100*S3)/10):S5=L-100*S3-10*S4
925 IFSC<0THENPOKETP+39,45
926 IFSC>=0THENPOKETP+39,32
930 POKETP+40,S3+48:POKETP+41,S4+48:POKETP+42,S5+48
950 RETURN
1000 PRINT:PRINT:PRINT:IFSC<150THEN1100
1010 PRINT"CONGRATULATIONS ADMIRAL, WITH ";SC;"POINTS"
1020 PRINT"YOU'RE AMERICA'S FIRST SUBMARINE ACE!!!":GOTO1600
1100 IFSC<100THEN1200
1110 PRINTSC;"POINTS IS EXCELLENT SHOOTING ADMIRAL"
1120 PRINT"IT'S GOOD ENOUGH TO GET YOU PROMOTED.":GOTO1600
1200 IFSC<50THEN1300
1210 PRINT"WELL ADMIRAL, ";SC;"POINT IS PRETTY GOOD SHOOTING,"
1220 PRINT"BUT THE TOP MAN THINKS YOU NEED MORE PRACTICE.":GOTO1600
1300 IFSC<0THEN1400
1310 PRINTSC;"POINT IS NOT VERY GOOD ADMIRAL."
1320 PRINT"THE COUNTRY IS LUCKY IT'S NOT DEPENDING ON YOU!!"
1330 GOTO1600
1400 PRINT"WHEN YOU ONLY GET ";SC;"POINTS, THE ENEMY"
1420 PRINT"IS GLAD YOU'RE NOT ON THEIR SIDE. YOU'LL"
1430 PRINT"BE MOPPING THE DECKS TOMORROW, IF YOU DON'T"
1440 PRINT"IMPROVE THAT SCORE!!!!"
1450 PRINT1450:PRINT"SC=";SC
```

```
1600 PRINT:PRINT:PRINT:INPUT"DO YOU WANT TO TRY AGAIN";A$
1650 IFLEFT$(A$,1)="Y"THENCLEAR:RUN130
1900 END
2000 DATA55041,55056,55071,55086,55101
2100 DATA-128,-127,-126,-63,-63,-62,1,2,64,65,66,128,129,130,0
```

STELLAR PERSUIT ----

THIS PROGRAM WILL RUN ONLY ON MACHINES WITH ASCII KEYBOARDS. THE PROGRAM CAN BE CONVERTED TO A POLLED KEYBOARD SYSTEM (SEE PROGRAM CONVERTIONS). IN THIS GAME YOU ARE CHASING AN INVADER HOME AND TRYING TO DESTROY HIM. TO CONTROL YOUR TURRET DEPRESS "1" TO MOVE DOWN, "2" TO MOVE UP, "3" TO MOVE RIGHT, "4" TO LEFT, AND "5" TO STOP. DEPRESSING THE "T" WILL FIRE ROCKETS. DEPRESS THE "F" TO FIRE THE LASERS.

```
1 REM ***ROBERT L.
2 REM ***COPPEDGE
5 POKE2073,96
9 S1=32:S2=32:S3=32
10 DIM X(20),Y(20),Z(20)
11 INPUT Z1
12 FOR Y=53250 TO 55295:POKE Y,32:NEXT Y
14 K0=INT(53316+RND(Z1)*1915)
16 FOR Y=1 TO 75
17 C=INT(53316+RND(Z1)*1915)
18 POKE C,46:NEXT Y
20 FOR Y=1 TO 10
30 X(Y)=INT(53316+RND(Z1)*1915)
40 Z(Y)=INT((X(Y)-53316)/64)
45 W(Y)=X(Y)-53316-64*Z(Y)
55 IF W(Y)<30 THEN W(Y)=-1
56 IF Z(Y)<14 THEN Z(Y)=-1
57 IF W(Y)<34 AND W(Y)>30 THEN W(Y)=0
58 IF Z(Y)<18 AND Z(Y)>14 THEN Z(Y)=0
59 IF W(Y)>33 THEN W(Y)=1
60 IF Z(Y)>17 THEN Z(Y)=1
65 IF M1=1 THEN 155
70 POKE X(Y),46:NEXT Y
110 K=55167:J=55231:G=55199
120 FOR Y=1 TO 10
121 GOTO 300
125 POKE X(Y),32
126 POKE 54177,45:POKE 54173,45:POKE 54047,73:POKE 54303,73
130 X(Y)=X(Y)+W(Y)+64*Z(Y)
135 POKE X(Y),46
140 IF X(Y)>55231 OR X(Y)<53316 THEN 150
145 IF X(Y)-53316-64*INT((X(Y)-53316)/64)>63 THEN 150
146 IF X(Y)-53316-INT((X(Y)-53316)/64)<1 THEN 150
147 GOTO 155
150 POKE X(Y),32:M1=1:GOTO 30
155 M=PEEK(57343)
156 IF M=212 OR L=2 THEN 190
157 IF M=198 OR L=1 THEN 162
158 IF M>176 AND M<182 THEN 550
159 IF M>0 AND M<5 THEN 551
160 IF M=198 OR L=1 THEN 162
161 GOTO 200
162 IF H=16 THEN 178
163 H=H+1:L=1
164 IF S8=20 THEN 900
165 POKE J,94:POKE K,94
166 POKE J+198,32:POKE K+186,32
167 J=J-66:K=K-62
168 GOTO 200
178 S8=S8+1
180 POKE 54373,32:POKE 54361,32:POKE 54307,32
181 POKE 54299,32:POKE 54241,32:POKE 54237,32
182 FOR I=1 TO 80:POKE 54175,I:NEXT I
183 H=0:L=0:K=55167:J=55231
184 POKE 54175,32
186 IF K0>54107 AND K0<54114 THEN 500
187 IF K0>54171 AND K0<54178 THEN 500
188 IF K0>54235 AND K0<54242 THEN 500
189 GOTO 200
190 IF H=8 THEN 195
```

```
192 IF S6=10 THEN 800
193 POKE G,32:G=G-128:POKE G,34
194 L=2:H=H+1:GOTO 200
195 POKE G,32:G=55199
196 S6=S6+1:GOTO 182
200 NEXT Y
210 GOTO 120
300 N=INT(16*RND(Z1)+1)
301 IF N>8 THEN 125
305 ON N GOTO 310,315,320,325,330,335,340,345
310 S=64:GOTO 347
315 S=-64:GOTO 347
320 S=63:GOTO 347
325 S=-63:GOTO 347
330 S=65:GOTO 347
335 S=-65:GOTO 347
340 S=1:GOTO 347
345 S=-1
347 K4=K0-53312-64*INT((K0-53312)/64)
348 IF K0+S>55231 OR K0+S<53312 THEN 300
349 IF K4<2 OR K4>61 THEN 300
350 POKE K0,S1:POKE K1,S2:POKE K2,S3
355 K0=K0+S:K1=K0+1:K2=K0-1:K3=K0-64
360 S1=PEEK(K0):S2=PEEK(K1):S3=PEEK(K2)
365 POKE K0,79:POKE K1,92:POKE K2,47
400 GOTO 125
500 PRINT"ENEMY DESTROYED!":S7=S7+1
501 FOR Y1=55039 TO 55061:POKE Y1,32:NEXT Y1
502 S1=32:S2=32:S3=32
505 POKE K0-64,32:POKE K0-63,32:POKE K0-65,32
510 K0=INT(53316+RND(Z1)*1915)
515 IF S7=6 THEN 1000
516 POKE 54113,32:POKE 54109,32
517 POKE 53983,32:POKE 54239,32
520 GOTO 120
550 M=M-176
551 ON M GOTO 555,560,570,580,585
555 S=64:GOTO 590
560 S=-64:GOTO 590
570 S=1:GOTO 590
580 S=-1:GOTO 590
585 S=0
590 POKE K0,S1:POKE K1,S2:POKE K2,S3
592 K0=K0+S:K1=K0+1:K2=K0-1
594 S1=PEEK(K0):S2=PEEK(K1):S3=PEEK(K2)
595 IF K0+S>55231 OR K0+S<53312 THEN 605
596 K4=K0-53312-64*INT((K0-53312)/64)
597 IF K4<2 OR K4>61 THEN 605
599 POKE K0,79:POKE K1,92:POKE K2,47
600 GOTO 200
605 K0=K0-S:K1=K0+1:K2=K0-1
610 POKE K0,79:POKE K1,92:POKE K2,47
615 GOTO 200
800 PRINT"TUBES EMPTY...":FOR Y2=1 TO 500:NEXT
801 FOR Y1=55039 TO 55053:POKE Y1,32:NEXT Y1
802 POKE K0-64,32:POKE K0-63,32:POKE K0-65,32
803 POKE 54113,32:POKE 54109,32:POKE 53983,32
804 POKE 54239,32
806 FOR Y1=1 TO 5
807 C=INT(55039+RND(Z1)*128)
808 POKE C,46:NEXT Y1:GOTO 120
900 PRINT"LASERS DRY...":FOR Y1=1 TO 500:NEXT Y1:L=0:GOTO 801
1000 PRINT"YEEEE-HAHHH!!!"
1010 END
```

NEW YORK TAXI ----

THIS PROGRAM WILL RUN ONLY ON MACHINES WITH ASCII KEYBOARDS. THIS GAME CAN BE CONVERTED TO A POLLED KEYBOARD SYSTEM. (SEE PROGRAM CONVERTIONS.)

```
1 REM--------B. BENNETT---N.Y. TAXI---
2 PRINT"DO YOU WANT DIRECTIONS (Y OR N)":INPUT E$
3 IF ASC(E$)=78 THEN 17:PRINT:PRINT:PRINT"NEW YORK TAXI"
4 PRINT:PRINT:PRINT"THE OBJECT OF THIS GAME IS TO CATCH A TAXI..."
5 PRINT:PRINT"WITH OUT BEING HIT BY SAID TAXI (ITS HARD TO GET N.Y."
6 PRINT:PRINT"CABBIES ATENTION !)"
7 PRINT:PRINT:PRINT"YOUR CONTROLS ARE":PRINT:PRINT"1 - MOVE UP "
```

```
8 PRINT:PRINT"2 - MOVE DOWN":PRINT:PRINT"3 - LEFT"
9 PRINT:PRINT"4 - RIGHT":PRINT:PRINT"AND MOST IMPORTANT"
10 PRINT:PRINT"5 - FLAGS DOWN TAXI"
11 PRINT:PRINT"OH, I ALMOST FORGOT, YOU HAVE TO BE DIRECTLY"
12 PRINT:PRINT"IN FRONT OF THE CAB FOR HIM TO NOTICE YOU."
13 PRINT:PRINT"READY (Y AND RETURN)":INPUT R$
17 FOR N=1 TO 32:PRINT:NEXT N
18 K=1:H=54112:I=H:GOTO700
19 FOR N=CA TO CA+8:POKE N,32:NEXT N:POKE CA-61,32
20 POKE CA-60,32:POKE CA-59,32:POKE CA+66,32:POKE CA+71,32
21 CA=CA+2:IF LN>35 THEN 700
22 FOR N=CA TO CA+8:POKE N,62:NEXT N
25 POKE CA-61,47:POKE CA-60,61:POKE CA-59,92
27 POKE CA+66,79:POKE CA+71,79
28 LN=LN+2
29 IF VI>0 THEN 310
30 FOR N=11 TO12:HI=PEEK(CA+N)
32 IFHI<>32ANDHI<>224THEN300
33 NEXT N
40 FOR V=I-1 TO I+1:POKE V,32:NEXT V:POKEI-64,32:POKE I+63,32
45 POKE I+65,32
50 POKE H,48:POKE H-64,79
60 POKE H+63,47:POKE H+65,92
70 POKE H+1,92:POKE H-1,47
80 I=H
90 M=PEEK(57343):IFM<176 OR M>182 THEN 19
95 IF M=177 THEN H=H-64
100 IF M=178 THEN H=H+64
105 IF M=179 THEN H=H-1
110 IF M=180 THEN H=H+1
115 IF M=181 THEN 200
130 M=0:GOTO 19
200 POKE H-1,32:POKEH+1,32
210 FOR N=0 TO 70:NEXTN:IF K/2=INT(K/2) THEN230
220 POKE H-63,47:POKE H-65,92:FOR N=0 TO 70:NEXT N
225 POKE H-63,32:POKE H-65,32:K=K+1:GOTO 19
230 POKEH-1,47:POKEH+1,92:FOR N=0TO70:NEXTN
240 K=K+1:GOTO 19
300 HI=0:VI=3
305 IF K/2=INT(K/2) THEN 400
310 VI=VI-1:IF VI<1 THEN 320:GOTO19
315 GOTO19
320 POKEH-64,83:POKE H-63,80:POKE H-62,76:POKE H-61,65:POKEH-60,84
321 U=INT(10*RND(H)):IF U>3 THEN 500
325 SC=SC-10
326 PRINT"WELL, WHATS A FEW MONTHS IN THE HOSPITAL..."
330 PRINT:PRINT"(DONT ANSWER)  YOU LOSE 10 POINTS ON THAT ONE":PRINT
331 PRINT"   WANT TO TRY AGAIN ?":INPUT D$
332 IF ASC(D$)=89 GOTO 17
333 PRINT:PRINT"SCARED HUH ?   TAKE A BUS NEXT TIME !"
336 PRINT
340 PRINT"TOTAL SCORE :  ";SC
350 END
400 POKE H+1,92:POKE H-1,47
425 SC=SC+25
430 IF SC>200 THEN 600
450 PRINT,"CONGRATULATIONS !"
452 PRINT:PRINT"YOU CAUGHT A CAB WITH NO BLOOD LOST !"
457 PRINT"SCORE :  ";SC
458 FOR B=1 TO 1500:NEXT B
460 VI=0:GOTO 17
500 PRINT"OUCH !   I BET THAT HURT !"
505 PRINT:PRINT
510 PRINT:PRINT"IT COST YOU 10 POINTS TOO.   IN VIEW OF ALL YOU'VE"
515 PRINT:PRINT"GONE THROUGH, I'LL GIVE YOU ANOTHER CHANCE"
517 SC=SC-10
520 GOTO 331
600 PRINT"SAY ARE YOU BY ANY CHANCE FROM NEW YORK ...?"
605 PRINT:PRINT"SCORE :  ";SC
610 GOTO 458
700 CA=64*INT(10*RND(H))+53760
702 LN=0
705 GOTO22
```

Memory Dump

Purpose

The program permits the user to dump contiguous blocks of memory of any length (65A will not dump blocks of more than 256_{10} words). This program also enables the user to specify the end address of a dump. This is very valuable when running at high baud rates!

Memory Allocation

0F7C - 0FB4 and 00FA-00FB
Starting Address- 0F7C

Relocation Information

The only position dependent instruction is the JMP at 0FA4

Use

Load the "GO" locations and load FA with the ending address low and FB with the ending address high. Type a "G". The program replies with an "X" and expects a four digit hex starting address for the dump. The program prints the block up to and including the last byte specified (unlike the 65A, this program will stop in the middle of a line). The program types an "R" and returns to the 65A.

Memory Dump

Location	Op Code		Mnemonic		Notes
F7C	A9 58		DUMP	LDA #'X	Type an
F7E	20 0B	FE		JSR OUTCH	"X"
F81	20 C7	FE		JSR BUILD	Accept Starting Address for Dump
F84	A0 00			LDY #$00	Clear Index
F86	A2 09		DUØ	LDX #$9	Set Byte/Line Count
F88	A9 0D			LDA #'CR	Output
F8A	20 0B	FE		JSR OUTCH	a Carriage
F8D	A9 0A			LDA #'LF	Return, Line Feed
F8F	20 0B	FE		JSR OUTCH	
F92	CA		DU1	DEX	Decrement Byte/Line Count
F93	F0 F1			BEQ DUØ	Reset if Zero
F95	20 E0	FE		JSR PRTBYT	Print Current byte
F98	A5 FA			LDA FINL	Low Address
F9A	C5 FC			CMP FC	Match?
F9C	F0 09			BEQ DU4	
F9E	E6 FC		DU3	INC FC	Increment Pointer
FA0	D0 F0			BNE DU1	Carry?
FA2	E6 F			INC FD	Yes, Carry!
FA4	4C 92	0F		JMP DU1	Loop Back
FA7	A5 FB		DU4	LDA FINH	High Address
FA9	C5 FD			CMP FD	Match?
FAB	D0 F1			BNE DU3	
FAD	A9 52			LDA #'R	Print an
FAF	20 0B	FE		JSR OUTCH	"R" and
FB2	4C 40	FE		JMP CONTROL + B	Exit

FINL FA
FINH FB

PRINTER PATCH ----

This program is a patch for OS-65U operating system to enable the use of a Selecterm printer on C3 machines.

```
 10 REM  CREATED BY FOUR STATE MICRO COMPUTER SYSTEMS INC.
 20 REM  JOPLIN,  MISSOURI  64801
 30 REM  JOE F. LINDEN
 40 REM  NAME -- PRINTER PATCH FOR SELECTERM PRINTER
 50 REM  EXECUTES IN BASIC MODE OF OPERATION
 60 REM  RE-CREATION DATE -- 7/18/78
 70 REM
 80 REM  EXECUTIVE OVERLAY ROUTINE AS FOLLOWS --
 90 REM
 100 DIM A(53)
 110 FOR I=1 TO 53:READ A(I):NEXT I
 120 I=1
 130 FOR J= 8668 TO 8696:POKE J,A(I):I=I+1:NEXT J
 140 FOR K= 8576 TO 8599:POKE K,A(I):I=I+1:NEXT K
 150 END
 160 DATA 169,0,141,03,247,141,02,247,169,4,141,03,247,173,02,247
 170 DATA 41,128,201,128,208,247,169,0,76,128,33,36,36,141,03,247
 180 DATA 169,127,141,02,247,169,44,141,03,247,173,205,34,41,127
 190 DATA 141,02,247,76,143,34
OK
POKE 8708,129
```

BACK SPACE ----

This program is for enabling the backspace for the OS-65U operating system on C3 systems. NOTE: this routine interferes with the new 65U editor.

```
290 REM FOR BACK SPACE THIS LINE SHOULD SAY GOSUB 5000

5000 REM POKE OUTPUT TO ALLOW PRINTING OF BACKSPACE
5010 POKE 21340,8: REM BACSPC=8
5020 REM  POKE PATCH INTO INPUT ROUTINE
5030 POKE 1373,76:POKE 1374,0:POKE 1375,92
5040 REM POKE PATCH INTO $5C00
5050 FOR PTR=23552 TO 23571
5060 READ MAC: POKE PTR,MAC
5070 NEXT PTR:RETURN
5080 DATA 201,8,208,13,32,8,48,72
5090 DATA 169,32,32,8,48,104,76,76,5,76,97,5
```

REPORT WRITER ---

This program is a Report Writer which can produce generalized listings and selective reports from disk files. The program will run on C3 systems under 65U.

```
  3 FORX=1 TO 32: PRINT:NEXT
  4 PRINT"REPORT WRITER"
  5 PRINT"-------------"
  6 POKE2888,0:REM ACCEPT NULL INPUT
  7 FLAG9:FLAG11:REM FLG9-50000 ERR CHK  FLG11-SPACE SUPP
 10 DV(1)=PEEK(9832):IF DV(1)>127 THEN DV(1)=DV(1)-128+4
 30 DV$(1)=CHR$(DV(1)+65)
 40 INPUT"DEVICE FILE STORED ON ";DV$(2)
 41 DV$(2)=LEFT$(DV$(2),1)
 42 IF DV$(2)>="A" OR DV$(2)<="E" GOTO 51
 43 PRINT:GOTO 40
 49 GOTO 51
 50 PRINT:PRINT"WHAT!":PRINT
 51 INPUT"FILE NAME, PASSWORD ";MN$,P$
 60 IF LEN(MN$)<5 THEN MN$=MN$+" ":GOTO 60
 65 IF LEN(MN$)>6 THEN GOTO 50
 70 F$=LEFT$(MN$,5)+"0"
110 IF P$=" " THEN P$="ANAN"
130 IF LEN(P$)<>4 GOTO 50
190 INPUT"CONSOLE(C) OR PRINTER(P)";V$
200 PRINT
210 IF LEFT$(V$,1)="P" THEN VD=5:SW=132:GOTO 290
230 IF LEFT$(V$,1)="C" THEN VD=PEEK(11665):SW=80:GOTO 290
250 GOTO 190
290 INPUT"ENTER HEADER COMMENT ";H$
295 PRINT
300 NE$="":IF H$=NE$ GOTO 330
310 IF LEN(H$)>SW-10 THEN PRINT"MAXIUM LENGTH OF COMMENT ";SW-10:GOTO290
330 Y=100:DIM L$(Y):DIM FP(Y):DIM FL(Y):DIM LL$(Y)
335 DIM LF(Y)
349 REM
350 PRINT"ENTER LABELS (WHEN FINISHED TYPE 'DONE' FOR THE NEXT LABEL)";L$(X)
355 NF=0:X=1:Y=100
360 PRINT:FORX=1 TO Y:PRINT"LABEL # ";X; :INPUT L$(X)
380 IF L$(X)="DONE" AND NF=0 THEN GOTO 2000
390 IF L$(X)="DONE" THEN L$(X)="":GOTO 460
395 INPUT"LABEL CHANGE (Y OR N) ";C$
396 IF C$<>"Y" AND C$<>"N" THEN GOTO 395
397 IF C$="N" GOTO 410
400 PRINT"ENTER CHANGE FOR LABEL # ";X:INPUT LL$(X)
410 PRINT
450 NF=NF+1:NEXT X
460 REM
470 INPUT"ARE ALL LABELS ENTERED CORRECTLY NOW (Y OR N)";Q$
475 PRINT:PRINT
485 IF LEFT$(Q$,1)="Y" THEN GOTO 600
490 IF LEFT$(Q$,1)="N" THEN FORX=1 TO NF:L$(X)=""
500 NF=NF-1:NEXTX:GOTO 350
600 DEV DV$(2)
650 OPEN F$,P$,1
665 INDEX(1)=9: INPUT%1,EODF
670 INDEX(1)=20:INPUT %1,BODF
675 INDEX(1)=31:INPUT %1,RL
680 INDEX(1)=42:INPUT %1,NR
690 FOR X=1 TO NF
800 FP(X)=1E8
850 NEXT X
900 GOSUB 5000
910 FOR Y=1 TO NF
```

```
920 IF FP(Y)<1E8 GOTO930
921 PRINT"ERROR **FIELD NOT FOUND**":FORX=1 TO NF+2:L$(X)=NE$:LL$(X)=NE$
922 FL(X)=0:FP(X)=0:TT=0:B=0:CLOSE 1:GOTO 349
930 NEXT Y
950 PRINT
1000 LR=0
1200 IF CNT=0 GOTO 1240
1220 IF LC<55 GOTO 1410
1240 PG=PG+1:PRINT#VD
1250 PRINT#VD,H$;TAB(SW-10);:PRINT#VD," PAGE ";PG
1280 PRINT#VD
1290 IF CNT>0 GOTO 1330
1300 REM SECTION TO CALCULATE TABS FOR OUTPUT
1305 FL=FL+1:IF FL>1 THEN GOTO 1320
1310 DIM PP(NF+1): REM PP=PRINT POSTION(S)
1320 FOR X=1 TO NF
1325 IF LEN(LL$(X))=>1 THEN L$(X)=LL$(X)
1330 IF FL(X)<LEN(L$(X)) GOTO 1360
1340 REM FIELD LONGER THAN COLUMN HEADER
1350 PP(X+1)=FL(X)+PP(X)+3: GOTO 1370
1360 PP(X+1)=LEN(L$(X))+PP(X)+3: REM HEADER > FIELD
1370 NEXT: REM TABS ARE SET AT THIS POINT
1371 IF PP(NF+1)<SW THEN GOTO 1380
1372 PRINT"ERROR **WIDTH OF REPORT WRITER IS ";PP(NF+1)-SW;" CHARACTERS TO";
1373 PRINT"O LONG**":FORX=1 TO NF+2:L$(X)="":D$(X)="":FP(X)=0:FL(X)=0:TT=0
1374 PG=0:LL$(X)="":IF X<>F THEN PP(X)=0
1375 NEXT X:CLOSE1:GOTO 349
1380 FOR X=1 TO NF: PRINT#VD,TAB(PP(X));L$(X);: NEXT: PRINT#VD: PRINT#VD
1390 REM
1400 REM
1410 FOR X=1 TO NF: INDEX(1)=BODF+FP(X)+LR: INPUT %1,D$(X): NEXT
1420 FOR X=1 TO NF: PRINT#VD,TAB(PP(X));D$(X);: NEXT: PRINT#VD
1850 LR=LR+RL:CNT=CNT+1:IF CNT=NR THEN GOTO 2000
1860 IFINDEX(1)>=EODF GOTO 2000
1880 LC=LC+1:IF LC>54 THEN LC=0:GOTO 1240
1900 FOR R=1 TO NF:D$(X)="":NEXT R
1950 GOTO 1410
2000 DEVDV$(1):CLOSE:FLAG10:FLAG12:
2010 PRINT:INPUT"ENTER A  C  TO CONTINUE ";QA$:FORX=1TO32:PRINT:NEXT:END
5000 TT=0:INDEX(1)=53
5050 INPUT %1,X$
5100 FOR X=1 TO NF
5101 Y=X
5150 IF L$(X)=X$ THEN U=U+1:GOTO 5300
5200 NEXT X:GOTO 5350
5300 FP(X)=TT
5350 INPUT %1,LF(Y):REM GET FILE LENGTH
5400 TT=TT+LF(Y):REM KEEP RUNNING TOTAL INTO RECORD
5425 IF L$(X)=X$ THEN FL(Y)=LF(Y)
5450 B=B+LEN(L$(X)):IF B+NF<SW THEN GOTO 5550
5460 PRINT"WIDTH OF REPORT WRITER > ";SW;:PRINT" CHAR":FOR X=1 TO NF
5470 L$(X)="":NF=NF-1:FP(X)=0:NEXT X
5480 TT=0:B=0
5500 CLOSE 1:GOTO 349
5550 IF INDEX(1)<BODF GOTO 5050
5600 IF INDEX(1)>BODF THEN PRINT"ERROR **FILE HEADER ERROR**"
5700 RETURN
50000 REM ERROR HANDLING IS DONE HERE
50010 ER=PEEK(18226):EL=PEEK(1174)+PEEK(11775)*256
50020 REM CHK FOR CHANNEL ALREADY OPEN
```

```
50030 IF ER=133 THEN CLOSE
50040 REM CHK FOR NON-EXISTANT ENTRY ERROR
50050 IF ER=128 THEN ERR$="INVALID FILE NAME": GOTO 52000
50060 REM CHK FOR END OF FILE HIT ERRORR
50070 IF ER=132 THEN ERR$="END OF FILE ERROR":CLOSE:GOTO 52000
50080 REM CHK FOR SYSTEM ERROR I.E. CHANNEL NOT OPEN ERROR
50090 IF ER=129 THEN ERR$="CANNOT ACCESS FILE":GOTO 52000
50100 ERR$="DISC ERROR CODE "+STR$(ER)+"IN LINE "+STR$(EL)
50200 CLOSE
52000 PRINT:PRINT"******ERROR******":PRINT:PRINT
53000 PRINT ERR$
54000 REM PICK UP DISC ADDRESS WHERE ERROR OCCURED
54010 EA=0:FORI=4 TO 1 STEP -1:EA=EA*256+PEEK(9889+I):NEXT I
54020 D=PEEK(9832):IF D>127 THEN D=D-128+4
54030 PRINT"ON DEVICE "+CHR$(D+65)+" AT DISC ADDRESS";EA
54040 REM COMMON EXIT PATH
54050 DEV CHR$(DV(1)+65):REM SELECT ORGINAL DEVICE
54060 FLAG10:FLAG12
54070 PRINT:INPUT"ENTER A  C  TO CONTINUE ";QA$:FORX=1TO32:PRINT:NEXT:END
```

LOCATE FILE ON DEVICE ----

This program locates any file under OS-65U operating system on C3 machines.

```
48997 REM
48998 REM  -----  LOCATE FILE ON ANY DEVICE  -----
48999 REM
49000 DV$="A":IF CH=0 GOTO 49020
49010 DEV DV$:OPEN F$,P$,CH:GOTO RL
49020 DEV DV$: RUN F$,P$,SL
50000 ER=PEEK(10226):EL=PEEK(11774)+PEEK(11775)*256
50010 IF EL<>49010 AND EL<>49020 GOTO 50100
50020 DV$=CHR$(1+ASC(DV$)):IF DV$<"I" GOTO EL
50030 PRINT:PRINT"FILE OR PROGRAM NOT FOUND ON LINE":PRINT:PRINT
50040 PRINT"PUT THE FILE < ";F$;" > ON LINE":PRINT
50050 POKE 2838,0:INPUT"ENTER A  C  TO CONTINUE ";X$
50060 GOTO 49000
50100 PRINT:PRINT"DISC ERROR":STOP:REM ***** STANDARD DISC ERROR POINT
59997 REM
59998 REM      -----  ROUTINE DESCRIPTION  -----
59999 REM
60000 REM      ENTRY-- F$,P$ = FILENAME AND PASWORD
60010 REM               CH=0   RUN AS A PROGRAM
60020 REM               CH<>0  OPEN AS A DATA FILE
60025 REM               RL =   RETURN LINE NUMBER
60026 REM                 (STACK LOST ON ERROR--NO SUB ROUTINES)
60027 REM                                    (NO FOR LOOPS)
60030 REM
60040 REM      EXIT     RETURN TO LINE RL (RETURN LINE NUMBER)
60050 REM               IF NO FIND THEN ALLOWS RETRYS UNTIL FOUND
```

```
1000 PRINT "***** CASH FLOW BUDGET ANALYSIS *****"
1001 PRINT "*                                   *"
1002 PRINT "*  PROGRAM BY P LINDQUIST 04/15/79  *"
1003 PRINT "*                                   *"
1004 PRINT "*************************************"
1010 REM
1020 REM PROGRAM PERFORMS CASH FLOW BUDGET ANALYSIS AS A
1030 REM FUNCTION OF CALENDAR PERIODS BASED ON WEEKLY START
1040 REM DATES IN DATA STATEMENTS BEGINNING AT L/N 1600.
1050 REM AVAILABLE DISK FILES : "SAMPLE"
1052 D1$="SAMPLE"
1055 NP=0
1060 DIM P$(26),PS(26),BAL(26)
1065 DIM P(30),DATE$(30),IE$(30),AMT(30),DES$(30)
1120 PRINT"DO YOU WANT INSTRUCTIONS (Y OR N)?";
1130 K=57088
1140 POKEK,4 : IFPEEK(K)=8THEN2000
1150 POKEK,16 : IFPEEK(K)=8THENGOSUB1165:GOTO2000
1160 GOTO1140
1165 FORI=1TO10:PRINT:NEXT
1170 REM  INSTRUCTIONS
1180 PRINT:PRINT"PROGRAM PERFORMS CASH FLOW BUDGET ANALYSIS."
1190 PRINT"PROGRAM WILL ASK WHAT ACTION IS REQUIRED BY"
1200 PRINT"PRINTING 'REQUEST?'.  YOU MAY RESPOND BY"
1210 PRINT"TYPING H, T, P, B, L, S, R, OR D AS FOLLOWS:"
1220 PRINT
1230 PRINT"H--REPEAT INSTRUCTIONS"
1240 PRINT"T--INCOME OR EXPENSE TRANSACTION"
1250 PRINT"P--SHOW DETAIL FOR PERIOD"
1260 PRINT"B--DISPLAY BALANCE FORECAST, HISTOGRAM, AND STATISTICS"
1265 PRINT"L--LIST ALL TRANSACTIONS"
1266 PRINT"S--SAVE TRANSACTION FILE ON DISK FILE"
1267 PRINT"R--RECALL TRANSACTION FILE FROM DISK"
1270 PRINT"D--DELETE TRANSACTION"
1275 PRINT"E--WILL EXIT TO THE SYSTEM MONITOR"
1280 PRINT
1290 PRINT"T--(INCOME OR EXPENSE TRANSACTION) WILL BE FOLLOWED"
1300 PRINT"BY '?'.  TYPE 'DATE, I OR E, AMOUNT, DESCRIPTION'"
1310 PRINT"FOR EACH TRANSACTION, WHERE 'I' INDICATES INCOME,"
1320 PRINT"AND 'E' INDICATES EXPENSE.
1350 PRINT
1360 PRINT"TO CONTINUE, HIT SPACE BAR"
1370 K=57088 : POKEK,2 : IF PEEK(K)=0 THEN 1370
1380 PRINT
1390 PRINT"DATE MUST BE IN FORMAT MM/DD/YY.  IF DATE IS"
1400 PRINT"'00/00/00', THE TRANSACTION WILL BE REPEATED EACH"
1410 PRINT"PERIOD.  IF MM IS '00', TRANSACTION WILL BE"
1420 PRINT"REPEATED FOR EACH MONTH."
1430 PRINT
1440 PRINT"P--(DETAIL FOR PERIOD) WILL BE FOLLOWED BY '?'."
1450 PRINT"TYPE IN THE PERIOD START DATE (MM/DD/YY)."
1460 PRINT
1470 PRINT"B--(BALANCE FORECAST) WILL BE FOLLOWED BY '?'.  TYPE"
1480 PRINT"IN THE BALANCE FORWARD AND THE NUMBER OF PERIODS TO"
1490 PRINT"BE FORECAST."
1500 PRINT
1510 PRINT"D--(DELETE TRANSACTION) WILL BE FOLLOWED BY '?'."
1520 PRINT"TYPE IN THE 'DESCRIPTION' OF THE TRANSACTION TO BE"
1530 PRINT"DELETED EXACTLY AS TYPED WHEN THE TRANSACTION WAS"
1540 PRINT"ENTERED."
1550 PRINT
1560 RETURN
1600 DATA12/31/78,01/07/79,01/14/79,01/21/79,01/28/79,02/04/79
1610 DATA02/11/79,02/18/79,02/25/79,03/04/79,03/11/79,03/18/79
1620 DATA03/25/79,04/01/79,04/08/79,04/15/79,04/22/79,04/29/79
1630 DATA05/06/79,05/13/79,05/20/79,05/27/79,06/03/79,06/10/79
1640 DATA06/17/79,06/24/79,07/01/79,07/08/79,07/15/79,07/22/79
1650 DATA07/29/79,08/05/79,08/12/79,08/19/79,08/26/79,09/02/79
1660 DATA09/09/79,09/16/79,09/23/79,09/30/79,10/07/79,10/14/79
1670 DATA10/21/79,10/28/79,11/04/79,11/11/79,11/18/79,11/25/79
1680 DATA12/02/79,12/09/79,12/16/79,12/23/79,12/30/79,01/06/80
1990 REM  INITIALIZE PERIOD START DATES
1998 DATA"NULL"
2000 RESTORE : PRINT : PRINT
2002 PRINT"INPUT THE START DATE FOR THE FIRST WEEK 'MM/DD/YY'"
2004 INPUT"SHOULD START WITH A SUNDAY:";D$
2006 PRINT
2008 GOSUB 2300: SD=SI
2010 INPUT"NUMBER OF WEEKS IN A PERIOD:";NW
2012 PRINT
2014 READ D$: IF D$<>"NULL" THEN2020
```

```
2016 PRINT"START DATE NOT FOUND - RESTART PROGRAM"
2018 RUN"BEXEC*
2020 GOSUB 2300 : IF ABS(SI-SD)>2 THEN 2014
2022 P$(1)=D$ : PS(1)=SI : NP=1
2030 FORJ=2TO26
2032 FORI=1TONW
2034 READ D$ : IF D$="NULL" THEN 2050
2036 NEXT I
2038 GOSUB 2300 : P$(J)=D$ : PS(J)=SI : NP=NP+1
2040 NEXT J
2050 PRINT"PERIOD START DATES ARE AS FOLLOWS:":PRINT
2052 FOR I=1TONP STEP 2
2054 PRINT P$(I),
2056 IF I<NP THEN PRINTTAB(25); P$(I+1)
2058 NEXT I : PRINT : PRINT
2060 PRINT"IS THIS OK? ";
2062 K=57088
2064 POKEK,16 : IF PEEK(K)=8THENPRINT"YES":GOTO2100
2066 POKEK,4 : IF PEEK(K)=8THENPRINT"NO?":GOTO2000
2070 GOTO2064
2100 PRINT:PRINT:PRINT
2110 NT=0 :REM  NUMBER OF TRANSACTIONS
2120 ST=100 :REM ST IS HISTOGRAM STEP SIZE
2130 PRINT:FOR I=1TO500:NEXT:PRINT"REQUEST? ";
2135 K=57088
2140 POKEK,16:X=PEEK(K)
2150 IFX=64THENPRINT"E":RUN"BEXEC*
2155 IFX=32THENPRINT"R":GOSUB4100:GOTO2130:REM INPUT FILE
2160 IFX=16THENPRINT"T":GOSUB2350:GOTO2130:REM TRANSACTION
2170 POKEK,32:X=PEEK(K)
2175 IFX=64THENPRINT"L":GOSUB4000:GOTO2130:REM LIST
2180 POKEK,8:X=PEEK(K)
2185 IFX=128THENPRINT"S":GOSUB5000:GOTO2130:REM DISK SAVE
2190 IFX=64THENPRINT"D":GOSUB3600:GOTO2130:REM DELETE
2195 IFX=8THENPRINT"HELP?":GOSUB1165:GOTO2130:REM HELP!!!
2200 POKEK,4:X=PEEK(K)
2202 IFX=16THENPRINT"B":GOSUB2700:GOTO2130:REM BALANCE
2205 POKEK,2:X=PEEK(K)
2207 IFX=2THENPRINT"P":GOSUB3300:GOTO2130:REM PERIOD DETAIL
2210 GOTO2140
2290 REM  INTEGER-FOR-DATE SUBROUTINE
2300 D$=RIGHT$(D$,8)
2305 DD=VAL(MID$(D$,4,2))
2310 MM=VAL(LEFT$(D$,2))
2320 YY=VAL(RIGHT$(D$,2))
2330 SI=DD+100*(MM+100*YY)
2340 RETURN
2345 REM  TRANSACTIONS SUBROUTINE
2350 FORI=1TO100:NEXTI:  NT=NT+1
2360 INPUTDATE$(NT),IE$,AMT(NT),DES$(NT)
2370 D$=DATE$(NT)
2380 GOSUB 2460
2390 IFFLAG=1THENNT=NT-1:RETURN
2400 P(NT)=IN
2410 IE$(NT)=LEFT$(IE$,1)
2420 IFIE$(NT)="I"ORIE$(NT)="E"THENRETURN
2430 PRINT"INCOME OR EXPENSE? ";
2435 X=USR(X):IFX=27THENSTOP
2436 PRINTCHR$(X)
2440 IFX=69THENIE$(NT)="E":RETURN
2445 IFX=73THENIE$(NT)="I":RETURN
2446 GOTO2430
2450 REM  CHECK-DATE SUBROUTINE
2460 FLAG=0
2470 IFLEN(D$)<>8THEN2600
2480 IFMID$(D$,3,1)<>"/"THEN2600
2490 IFMID$(D$,6,1)<>"/"THEN2600
2500 GOSUB2300
2510 IFMM>12THEN2600
2520 IFDD>31THEN2600
2530 IFSI=0THENIN=0:RETURN
2540 IFMM=0THENIN=-1:RETURN
2550 FORIN=1TONP-1
2560 IFPS(IN)<=SIANDSI<PS(IN+1)THENRETURN
2570 NEXTIN
2580 PRINT"DATE NOT WITHIN RANGE OF PERIODS IN DATA TABLE"
2600 PRINT"DATE MUST BE IN FORMAT 'MM/DD/YY'.  FOR EXAMPLE, "
2610 PRINT"4TH OF JULY 1979 WOULD BE '07/04/79'."
2620 FLAG=1:RETURN
2630 REM  EXPENSE-INCOME SUBROUTINE
2640 IF IE$(J)="E"THENBAL(I)=BAL(I)-AMT(J)
2650 IF IE$(J)="I"THENBAL(I)=BAL(I)+AMT(J)
```

```
2660 RETURN
2690 REM  BALANCE SUBROUTINE
2700 FOR I=1TO100 : NEXT
2710 INPUT BAL, NS
2720 IF NS>NP THEN NS=NP
2750 PRINT
2760 PRINT"BALANCE FORWARD :"; BAL
2770 PRINT
2780 PRINT "PERIOD       BALANCE";
2785 PRINTTAB(25);"-------------++++++++++++++++++++"
2790 BAL(1)=BAL
2800 FOR I=1TONS-1
2810 IF I>1 THEN BAL(I)=BAL(I-1)
2820 FORJ=1TONT
2830 IF P(J)=0 THEN GOSUB 2640 : GOTO2900
2860 IF P(J)=I THEN GOSUB2640 : GOTO2900
2870 IF P(J)>0 THEN 2900
2880 GOSUB3000
2890 IFFLAG=1THENGOSUB2640
2900 NEXTJ
2910 PRINTP$(I+1);"    ";BAL(I);
2920 GOSUB3100
2930 NEXTI
2940 GOSUB3800
2950 RETURN
2990 REM  EACH-MONTH SUBROUTINE
3000 FLAG=0
3010 DD=VAL(MID$(DATE$(J),4,2))
3020 D1=VAL(MID$(P$(I),4,2))
3030 D2=VAL(MID$(P$(I+1),4,2))
3040 IF LEFT$(P$(I),2)=LEFT$(P$(I+1),2)THEN3070
3050 IF DD>=D1THENFLAG=1
3060 IF DD<D2THENFLAG=1
3070 IF DD>=D1ANDDD<D2THENFLAG=1
3080 RETURN
3090 REM  HISTOGRAM SUBROUTINE
3100 IFBAL(I)<ST/2THEN3160
3110 PRINTTAB(40);
3120 FORJ=1TOINT((BAL(I)/ST)+.5)
3130 PRINT "$";
3135 IFPOS(I)>59THEN3210
3140 NEXTJ
3150 GOTO3210
3160 IFBAL(I)>=-ST/2THEN3210
3170 PRINTTAB(40+INT((BAL(I)/ST)+.5))
3180 PRINT "#";
3190 IFPOS(I)>=40THEN3210
3200 GOTO3180
3210 PRINT
3220 RETURN
3290 REM  PERIOD-DETAIL SUBROUTINE
3300 FORI=1TO100:NEXTI: INPUT D$
3310 GOSUB2460
3320 IFFLAG=1GOTO3300
3330 I=IN
3340 BAL(I)=0
3350 SJ=SI
3360 PRINT"DATE","EXP/INC","AMOUNT","DESCRIPTION"
3370 PRINT
3380 FORJ=1TONT
3390 IFP(J)=0THEN3460
3400 D$=DATE$(J)
3420 GOSUB2300
3430 IFP(J)=IANDSI>=SJTHEN3460
3440 IFP(J)>0THEN3480
3450 GOSUB3000:IFFLAG=0THEN3480
3460 GOSUB2640
3470 PRINTDATE$(J),IE$(J),AMT(J),DES$(J)
3480 NEXTJ
3490 PRINTTAB(28);"----------"
3500 PRINT"BALANCE FOR PERIOD:";TAB(28);BAL(I)
3510 RETURN
3590 REM  DELETE-TRANSACTION SUBROUTINE
3600 FORI=1TO100:NEXTI: INPUT DES$
3610 FORI=1TONT
3620 IFDES$=DES$(I)THEN3660
3630 NEXTI
3640 PRINT"TRANSACTION RECORD NOT FOUND:  ";DES$
3650 RETURN
3660 P(I)=P(NT)
3670 DATE$(I)=DATE$(NT)
3680 IE$(I)=IE$(NT)
```

```
3690 AMT(I)=AMT(NT)
3700 DES$(I)=DES$(NT)
3710 NT=NT-1
3720 RETURN
3790 REM  STATISTICS SUBROUTINE
3800 IFNS<1THENRETURN
3810 SN=0
3820 SX=0
3830 SY=0
3840 X2=0
3850 XY=0
3860 FORI=I1TOI1+NS
3870 SN=SN+1
3880 SX=SX+I
3890 X2=X2+I*I
3900 SY=SY+BAL(I)
3910 XY=XY+I*BAL(I)
3920 NEXTI
3930 A=SN*X2-SX*SX
3940 B=(SN*XY-SX*SY)/A
3950 PRINT "SLOPE IS ";INT(B);" (DOLLARS) PER PERIOD"
3960 RETURN
3990 REM  TRANSACTION-LIST SUBROUTINE
4000 PRINT"  TOTAL NUMBER OF TRANSACTIONS: ";NT
4010 PRINT"DATE","INC/EXP","AMOUNT","DESCRIPTION"
4020 FORI=1TONT
4030 PRINTDATE$(I),IE$(I),AMT(I),DES$(I)
4040 NEXTI
4050 RETURN
4090 REM  DISK RETREIVAL SUBROUTINE
4100 FOR I=1TO100:NEXTI
4110 INPUT"NAME";NAME$
4120 IFNAME$<>D1$ANDNAME$<>D2$ANDNAME$<>D3$THENRETURN
4130 DISK OPEN,6,NAME$
4135 PRINT"DISK FILE FOR ";NAME$;" OPEN"
4140 INPUT#6, NT
4150 FORI=1TONT
4160 INPUT#6, P(I)
4161 INPUT#6, DATE$(I)
4162 INPUT#6, IE$(I)
4163 INPUT#6, AMT(I)
4164 INPUT#6, DES$(I)
4165 PRINT DATE$(I),IE$(I),AMT(I),DES$(I)
4170 NEXT I
4200 DISK CLOSE, 6
4220 PRINT
4230 PRINT"FILE LOAD COMPLETE - ";NAME$:PRINT
4240 RETURN
4990 REM  DISK FILE SAVE ROUTINE
5000 FORI=1TO100:NEXT
5010 INPUT"NAME";NAME$
5020 IFNAME$<>D1$ANDNAME$<>D2$ANDNAME$<>D3$THENRETURN
5030 DISK OPEN, 6, NAME$
5035 PRINT"DISK FILE FOR ";NAME$;" OPEN"
5040 PRINT#6, NT
5050 FOR I=1TONT
5060 PRINT#6, P(I)
5070 PRINT#6, DATE$(I)
5080 PRINT#6, IE$(I)
5090 PRINT#6, AMT(I)
5100 PRINT#6, DES$(I)
5110 NEXT I
5115 DISK CLOSE, 6
5120 PRINT
5130 PRINT"FILE SAVE COMPLETE - ";NAME$:PRINT
5240 RETURN
```

JOYSTICK ----

This program will run on any machine with joysticks. The progam is a demonstration of the programming and operation of the joysticks.

```
10 REM PGM NAME = JOYSTK
20 REM TESTS 2 JOYSTICKS SIMULTANEOUSLY !
30 REM PROGRAMMED BY L. KREBS ON 03/01/79
31 REM
32 REM
33 REM
40 DIM K(20),P$(10)
50 FOR I=1 TO 10  :REM INITIALIZE ARRAYS
60 READ P$(I)
70 NEXT I
80 READ Q$(1),R$(1)  :REM GET ACTION MSGS + PLAYER MSGS
90 READ Q$(2),R$(2)
92 FOR I=1 TO 18  :REM GET DECODE COMPARE VALUES
94 READ K(I)
96 NEXT I
97 Y1=50
100 POKE 2073,96  :REM DISABLE CTRL-C
110 POKE 57088,128  :REM STROBE PLAYER "A", ROW7
120 AA=PEEK(57088) AND 31  :REM USE ONLY BITS 0-4
124 POKE 57088,16  :REM STROBE PLAYER "B", ROW 4
126 BB=PEEK(57088) AND 248  :REM USE ONLY BITS 3-7
129 Z=0
130 A=AA AND 30  :REM REMOVE ACTION KEY ON BIT 0
135 LA=1  :REM ACTION KEY IS BIT 0
138 X=1  :REM COMPARE VALUES IN K(1) - K(9)
140 GOSUB1000  :REM DO PLAYER "A"
150 A1=C1  :REM SAVE PLAYER "A" POSITION
160 A2=C2  :REM SAVE PLAYER ACTION KEY STATUS
180 AA=BB
190 A=AA AND 120  :REM REMOVE ACTION KEY ON BIT 7
200 LA=128  :REM ACTION KEY IS BIT 7
208 X=10  :REM COMPARE VALUES IN K(10) - K(18)
210 GOSUB 1000  :REM DO PLAYER "B"
220 B1=C1
230 B2=C2
250 IF A1<>Y1 THEN Z=1
255 IF A2<>Y2 THEN Z=1
260 IF B2<>Z2 THEN Z=1
265 IF B1<>Z1 THEN Z=1
270 Y1=A1:Y2=A2
280 Z1=B1:Z2=B2
290 IF Z=1 THEN 350
295 GOTO 400  :REM IF NO CHANGE, THEN DON'T PRINT
350 PRINT R$(1),P$(A1),Q$(A2)
360 PRINT R$(2),P$(B1),Q$(B2)
365 PRINT "PRESS 'A' TO ABORT"
370 PRINT "  "
400 POKE 57088,2  :REM PRESS "A" TO ABORT
420 IF PEEK(57088)=64 THEN 900
440 GOTO 100  :REM GO DO EVERYTHING AGAIN !
900 PRINT "ABORT"
910 RUN"BEXEC*
1000 REM SUBTROUTINE TO CHECK A JOY STICK
1010 C1=10  :REM ASSUME ERROR UNLESS CHANGED
1020 FOR I=1 TO 9
1030 Y=X+I-1
1040 IF A=K(Y) THEN C1=I
1050 NEXT I
1110 C2=1  :REM ASSUME ACTION NOT PRESSED
1120 IF(AA AND LA) <> 0 THEN C2=2  :REM CHECK ACTION KEY
```

```
1130 RETURN
1990 REM
1995 REM
2000 REM POSITIONS AVAILABLE
2001 REM NOTE: POSITION "Z" IS THE ERROR POSITION
2010 DATA A,B,C,D,E,F,G,H,I,Z
2020 DATA "NORMAL",'A' POSITION =
2030 DATA " ACTION ",'B' POSITION =
2040 REM  PLAYER "A" DECODE A-I
2050 DATA 16,20,4,12,8,10,2,18,0
2060 REM PLAYER "B" DECODE A-I
2080 DATA 32,48,16,80,64,72,8,40,0
3000 END
```

SET TIME OF DAY ----

This program will run on all machines with graphics and real time capabilities. It shows updated time every second. OS-65D V3.1 HC operating system is needed to set up internal clock hardware.

```
10 REM  SET TIME OF DAY PROGRAM
20 REM
30 PRINT: INPUT "TIME (H,M,S) "; H,M,S
40 T = 9480
50 POKE T,H: POKE T+1,M: POKE T+2,S
60 REM
100 REM  DISPLAY TIME OF DAY & TRIP TIMER IF ON
110 REM
120 T = 9480
122 C = 224
130 H = PEEK(T): M = PEEK(T+1): S = PEEK(T+2)
134 IF PEEK(C-1) = 0 GOTO 140
136 HT= PEEK(C): MT= PEEK(C+1): ST= PEEK(C+2)
138 PRINT HT;":";MT;":";ST,
140 PRINT H;":";M;":";S
150 IF PEEK(T+2) = S GOTO 150
160 GOTO 130
170 REM
200 REM  TURN TRIP TIMER ON
210 REM
220 C = 224
230 POKE C,0: POKE C+1,0: POKE C+2,S
240 POKE C-1,1
250 GOTO 100
260 REM
300 REM  ACTIVATE PIA EVENT INPUTS
310 REM
320 REM    SETUP "STATE OF INTEREST" BITS
330 POKE 9392,254: REM  PIA 1 A, LOOK FOR BIT 0 = 0
340 POKE 9396,253: REM  PIA 2 A, LOOK FOR BIT 1 = 0
350 POKE 9400,251: REM  PIA 3 A, LOOK FOR BIT 2 = 0
360 REM    SETUP MASKS
370 POKE 230,1: REM  PIA 1 A, BIT 0
380 POKE 234,2: REM  PIA 2 A, BIT 1
390 POKE 238,4: REM  PIA 3 A, BIT 2
400 REM    PERMIT 'RTMON' TO RUN
410 POKE 222,1
420 END
```

DIGITAL CLOCK ----

This program will run on machines with real time clocks and graphics. It is a digital clock updating time every minute. The clock will appear on the screen after time has been set. OS-65D V3.1 HC operating system is needed to run the internal time hardware.

```
5 DIMZ(3,1),G(3,3),C(9,3,3):M=1:POKE2073,96:FORI=0TO3:READZ(I,1):NEXT
32 FORI=0TO3:FORJ=0TO3:READG(I,J):NEXTJ,I
33 FORI=0TO9:FORJ=0TO3:FORK=0TO3:READC(I,J,K):NEXTK,J,I
40 FORI=1TO30:PRINT:NEXT:T=9480:GOSUB30000:FORI=1TO30:PRINT:NEXT
50 GOSUB2000:GOTO50
2000 H0=PEEK(T):M0=PEEK(T+1):S0=PEEK(T+2)
2001 C$=" A.M.":IFH0>12THENH0=H0-12:C$=" P.M."
2002 IFH0=12THENC$=" P.M."
2003 IFH0=0THENH0=12
2005 IFM0=M1THENPOKE54053,32:FORI=1TO100:NEXT:POKE54053,42:GOTO2020
2006 M1=M0
2010 GOSUB9000
2011 IFC$=" A.M."ANDPEEK(54123)=32THENPOKE54123,146
2012 IFC$=" P.M."ANDPEEK(54123)=146THENPOKE54123,32
2020 IFPEEK(T+2)<>S0THENRETURN
2030 GOTO2020
9000 Z=0
9010 Z(3,0)=INT(H0/10):Z(2,0)=H0-Z(3,0)*10
9020 Z(1,0)=INT(M0/10):Z(0,0)=M0-Z(1,0)*10
9025 IFQS=0THEN9100
9030 IFZ=4THENRETURN
9040 IFZ=3ANDZ(3,0)=0THENPOKE54036,32:POKE54100,32:RETURN
9050 FORI=0TO3:FORJ=0TO3:POKEZ(Z,1)+G(I,J),C(Z(Z,0),I,J)
9060 NEXTJ,I:IFZ(Z,0)=0THENZ=Z+1:GOTO9030
9065 IFZ(3,0)=0ANDPEEK(54100)<>32THENZ=3:GOTO9040
9070 RETURN
9100 POKE54042,174:POKE54106,174:FORZ=0TO3
9110 IFZ=3ANDZ(3,0)=0THENPOKE54036,32:POKE54100,32:GOTO9130
9120 FORI=0TO3:FORJ=0TO3:POKEZ(Z,1)+G(I,J),C(Z(Z,0),I,J)
9125 NEXTJ,I,Z
9130 QS=1:L=54056:POKEL-63,128:POKEL-62,128:POKEL,147
9140 POKEL+1,144:POKEL+2,144:POKEL+3,146:POKEL+64,147
9150 IFC$=" A.M."THENPOKEL+67,146
9160 L=54060:POKEL,147:POKEL+1,190:POKEL+2,189:POKEL+3,146
9170 POKEL+64,147:POKEL+67,146:RETURN
10000 DATA54047,54043,54038,54034,-64,-63,-62,-61,0,1,2,3,64,65
10010 DATA66,67,129,129,130,131
10020 DATA32,128,128,32,147,32,32,146,147,32,32,146,32,135,135,32
10025 DATA32,32,32,32,32,32,146,32,32,32,146,32,32,32,32,32
10030 DATA32,128,128,32,32,144,144,146,147,32,32,32,32,135,135,32
10040 DATA32,128,128,32,32,144,144,146,32,32,32,146,32,135,135,32
10050 DATA32,32,32,32,147,144,144,146,32,32,32,146,32,32,32,32
10060 DATA32,128,128,32,147,144,144,32,32,32,32,146,32,135,135,32
10070 DATA32,128,128,32,147,144,144,32,147,32,32,146,32,135,135,32
10080 DATA32,128,128,32,32,32,32,146,32,32,32,146,32,32,32,32
10090 DATA32,128,128,32,147,144,144,146,147,32,32,146,32,135,135,32
10100 DATA32,128,128,32,147,144,144,146,32,32,32,146,32,135,135,32
30000 INPUT"WHAT IS THE TIME (H,M,S)";H,M,S
30010 IFH<10RH>12ORH<>INT(H)GOTO30000
30020 IFM<0ORM>59ORM<>INT(M)GOTO30000
30030 IFS<0ORS>59ORS<>INT(S)GOTO30000
30040 PRINT:PRINTH;"HOURS,"M"MINUTES, AND"S"SECONDS":PRINT
30050 INPUT"IS THAT A.M. OR P.M.";Y$
30060 IFLEFT$(Y$,1)="P"THENH=H+12
30065 IFH=24THENH=12
30066 IFH=12ANDLEFT$(Y$,1)="A"THENH=0
30070 ST=9480:POKEST,H:POKEST+1,M:POKEST+2,S:RETURN
```

FLASHER ----

This program runs on all graphics machines except the C1P. This BASIC program loads a machine level program into memory by the use of DATA statements and then runs it. The program flashes random graphics and color (if machine has color hardware) on the screen.

```
1 FL=PEEK(57088):GOSUB60000
2 POKE2073,96:POKE57088,1
10 PRINT"TO ABORT PROGRAM, PRESS 'CTRL .'
20 PRINT"PRESS ANYTHING TO BEGIN?!";:INPUTY$
30 BE=INT(RND(1)*9)+208:LE=INT(RND(1)*(216-BE))+1
40 CH=INT(RND(1)*128)+128:IFRND(1)>.45THENBE=BE+16
50 POKE16336,BE:POKE16337,LE:POKE16338,CH
55 IFPEEK(57088)<>FLGOTO70
60 POKE56832,5:X=USR(X):GOTO30
70 POKE56832,1:END
60000 POKE8955,211:POKE8956,63
60010 FORI=16339TO16372:READA
60020 POKEI,A:NEXT:RETURN
60030 DATA172,209,63,174,210,63,173,208,63,141,225,63
60040 DATA142,0,208,173,224,63,24,105,1,141,224,63,144
60050 DATA242,238,225,63,136,152,208,235,96
```

THEME ----

This short program plays the Star Wars theme through the tone generator. The tone generator is the only special hardware that is needed to run the program.

```
1 POKE56832,7
10 DEFFNA(C)=(49152/(2^(C/12)*220))
20 FORJ=1TO2:FORI=1TO48
30 READA:POKE57089,FNA(A)
40 NEXTI:RESTORE:NEXTJ
50 END
100 DATA1,1,1,8,8,8,8,8,8,6,5,3,13,13,13
110 DATA8,8,8,8,8,8,6,5,3,13,13,13,8,8,8,8,8,8
120 DATA6,5,6,3,3,3,3,3,3,1000,1000,1000,1000,1000,1000
```

KEYS ----

This program requires DAC hardware to operate. It is an assembler program and must be typed in through the assembler editor, assembled to memory by typing A3, and then started by typing "!GO 317E". After the program is started the keyboard becomes musical, sending various sounds through the DAC output. Note, there is no exit for this program.

```
 10              *=$317E
 20       LDA #$FF
 30       STA $DF00
 40START  LDA $DF00
 50       TAX
 60LOOP   ADC $20,X
 70       STA $DF01
 80       DEX
 90       BNE LOOP
100       STA LOOP+1
110       JMP START
```

PLANE BANNER ----

This program can be run on any graphics machine except the C1P. The program is a demonstration of animation. If the program is to be run on a color system, please refer to USR(X) for color backgrounds. NOTE: On non disk based systems you must delete all disk calls (DISK!"CA 4FD0=36,1").

```
15 REM** PLANE BANNER ****
20 DIMA(25)
25 FORI=1TO20:READA(I):NEXT
30 PRINT:PRINT:PRINT:PRINT
35 PRINT"     O.S.I. COMPUTERS"
40 FORI=1TO22:PRINT:NEXT
45 POKE56832,6
50 D=4096:TS=12:GR=6
60 DISK!"CA 4FD0=36,1":POKE20433,TS:X=USR(X)
80 FORI=54976TO55200
90 POKEI+D,GR:NEXT
94 K=53700
95 K=K+D:GOSUB1100
100 FORI=54915TO54937STEP2
110 POKEI+D,0:POKEI,15
120 NEXT
200 FORI=54464TO54494
210 POKEI,237
215 POKEI+D,2
220 FORJ=1TO50:NEXT
230 POKEI,32
240 POKEI+D,12
250 NEXT
300 FORI=53982TO53952STEP-1
310 POKEI,239
320 FORJ=1TO50:NEXT
330 POKEI,32
340 NEXT
400 FORI=54208TO54238
410 POKEI,237
420 POKEI+D,10
430 POKEI+D-5,7
440 GOSUB1000
445 POKEI-21,32
446 POKEI+D-23,12
450 FORJ=1TO50:NEXT
460 POKEI,32
470 POKEI+D,12
500 NEXT
600 GOTO2000
1000 FORV=7TO20
1010 POKEI-V,A(V)
1020 NEXT
1030 RETURN
1100 FORI=1TO16
1110 POKEK+I,I
1120 NEXT:RETURN
2000 FORI=1TO3000:NEXT
2005 FORI=54915TO54937:POKEI,32:NEXT
2010 X=USR(X)
2100 END
5000 DATA32,32,32,32,32,32
5100 DATA83,67,73,72,80,65
5200 DATA82,71,32,82,79,76,79,67
```

AC ----

This program will run on any machine with CA-12 remote AC control interface and OS-65D V3.1 HC operating system. The program needs an AC driver in memory before the program is typed into the computer.(See color and AC manual.)

```
10 REM  AC CONTROL PROGRAM
20 REM  (WITH EMBEDDED AC CONTROL DRIVER)
30 REM SETUP VECTOR TO 'ACTL' CODE
40 POKE 548,127: POKE 549,50: REM $0224 = $7F, $32
50 REM
60 REM PRESET STATUS TO 64 CHAR DISPLAY
70 POKE 249,1: REM $F9
80 REM
90 REM  OPERATOR INTERFACE FOR AC CONTROL
100 INPUT "DEVICE, COMMAND NUMBER "; N, M
110 ACTL N,M: GOTO 100
120 REM
130 REM  MINI LIGHT SHOW
140 NN=65: BR=66: LO=67: REM  ON: BRIGHT: LIGHTS ON
150 FF=68: DI=69: AF=70: REM  OFF: DIM: ALL OFF
200 ACTL 1,2,3,12
210 ACTL NN
220 ACTL 1,FF,2,FF,3,FF,12,FF
230 ACTL 12,NN,3,NN,2,NN,1,NN
240 ACTL 1,2,3,12
250 FOR I=1 TO 10
260 ACTL DI
270 NEXT I
```

MACHINE CODE KALEIDOSCOPE ----

This program will only run on color systems. It is an assembler program and must be typed in through the assembler editor, assembled to memory by typing A3, and then started by typing "!GO 4025". Depressing the "E" will return you to the assembler editor.

```
 10              ; KALEIDOSCOPE
 20              ; DAVID E. TEWKSBARY
 30        *=$4000
 40          TEMP=$3FD0
 50          DUMM=$3FD8
 60         COUNT=$3FEA
 70        COUNT2=$3FEB
 80            CH=$3FEC
 90          NUMB=$3FED
100;
110CLEAR   LDA #$D0
120        STA CHAR+2
130        LDA #$E0
140        STA COLO+2
150        LDX #0
160        LDY #8
170LOOP    LDA #32
180CHAR    STA $D000,X
190        LDA #8
200COLO    STA $E000,X
210        INX
220        BNE LOOP
230        INC CHAR+2
240        INC COLO+2
250        DEY
260        BNE LOOP
270        RTS
280;
290START   LDA #4
300        STA $DE00
310        JSR CLEAR
320        LDA #96
330        STA 2073
340;
350BCOL    LDX #1
360        LDA #$D4
370LOOP1   STA TEMP,X
380        INX
390        CPX #5
400        BNE CONT
410        LDA #$E
420CONT    CPX #9
430        BNE LOOP1
440        LDA #0
450        STA COUNT
460;
470        LDA #16
480        STA $DF00
490        LDA $DF00
500        CMP #64
510        BNE COLADD
520;
530        JSR CLEAR
540        LDA #1
550        STA $DE00
560        RTS
570;
580COLADD   INC COUNT
590        LDY COUNT
600        CPY #9
610        BEQ BCOL
620;
630        LDA #63
640        STA NUMB
650;
660        LDX #1
670        LDY #5
680        JSR SUB
690;
700        LDX #3
710        LDY #7
720        JSR ADD
730;
740        LDA #65
750        STA NUMB
760;
770        INX
780        INY
790        JSR SUB
800;
810        LDX #2
820        LDY #6
830        JSR ADD
840;
850BROW    LDX #1
860        LDY #1
870LOOP2   LDA TEMP,X
880        STA DUMM,Y
890        INY
900        STA DUMM,Y
910        INY
920        INX
930        CPX #9
940        BNE LOOP2
950        LDY #0
960        STY COUNT2
970;
980        LDY #5
990        LDA CH
1000RANDOM ADC DUMM,Y
1010       DEY
1020       BNE RANDOM
1030       ORA #128
1040       STA CH
1050;
1060;
1070ROWADD INC COUNT2
1080       LDY COUNT2
1090       CPY COUNT
1100       BNE OKAY
1110       JMP COLADD
1120;
1130OKAY   LDA #65
1140       STA NUMB
```

```
1150;
1160        LDX #9
1170        LDY #17
1180        JSR SUB
1190;
1200        INX
1210        INY
1220        JSR ADD
1230;
1240        LDX #13
1250        LDY #21
1260        JSR ADD
1270;
1280        INX
1290        INY
1300        JSR SUB
1310;
1320        LDA #63
1330        STA NUMB
1340;
1350        LDX #11
1360        LDY #19
1370        JSR SUB
1380;
1390        INX
1400        INY
1410        JSR ADD
1420;
1430        LDX #15
1440        LDY #23
1450        JSR ADD
1460;
1470        INX
1480        INY
1490        JSR SUB
1500;
1510DONER   LDA $DE00
1520        BPL DONER
1530TIGHT2  LDA $DE00
1540        BMI TIGHT2
1550        LDY #1
1560        LDX #9
1570MOVE1   LDA DUMM,Y
1580        STA POKE+2
1590        CLC
1600        ADC #16
1610        STA POKE2+2
1620MOVE2   LDA DUMM,X
1630        STA POKE+1
1640        STA POKE2+1
1650        INX
1660        INY
1670        LDA CH
1680POKE2   STA $E000
1690POKE    STA $D000
1700        CPY #9
1710        BNE MOVE1
1720        JMP ROWADD
1730;
1740ADD     LDA $3FD0,Y
1750        CLC
1760        ADC NUMB
1770        STA $3FD0,Y
1780        BCC RIGHT
1790        INC $3FD0,X
1800RIGHT   RTS
1810;
1820SUB     LDA $3FD0,Y
1830        SEC
1840        SBC NUMB
1850        STA $3FD0,Y
1860        BCS RIGHT2
1870        DEC $3FD0,X
1880RIGHT2  RTS
```

RANDOM SQUARE ----

This is a short graphics demonstration program. The program is compatible with all graphics machines except the C1P. If a color unit is used, then the program "USR(X) for color backgrounds" should also be used. If the computer system does not have a disk drive, then the disk call (DISK!"CA 4FD0=36,1) should be deleted from the program. NOTE: For more detail see USR(X) for color backgrounds.

```
10 REM ** PAUL A. JOVIAK **
12 REM ** RANDOM SQUARE  **
15 POKE56832,6:PRINT:PRINT:PRINT:PRINT
17 PRINT"      OHIO SCIENTIFIC:"
20 PRINT:PRINT"                COLOR GRAPHICS"
23 PRINT"                     AND"
25 PRINT"                INVERTED VIDEO"
30 FORI=1TO20:PRINT:NEXT
40 DISK!"CA 4FD0=36,1"
50 D=4096:B(1)=1:B(2)=-64:B(3)=-1:B(4)=64
80 FORPS=1TO5
90 FORL=1TO2:H=58318
100 FORI=1TO11STEP2
140 H=H+63:FORK=1TO4:Z=H
144 IFK=2THENZ=H+I
145 IFK=3THENZ=H+I+I*-64
146 IFK=4THENZ=H+I*-64
150 FORJ=0TOI-1
160 TS=INT(16*RND(Z))
170 RS=INT(255*RND(TS))
200 POKEZ+J*B(K),TS
220 F=PEEK(57088):IFF<>1THEN800
250 IFL=2THENPOKEZ+J*B(K)-D,32:GOTO500
300 POKEZ+J*B(K)-D,RS
500 NEXTJ,K,I,L
600 POKE20433,TS:X=USR(X)
650 NEXTPS
800 END
```

BLACKJACK ----

This program will run on any machine with a Votrax unit. Note, this program requires a votrax driver (see Votrax user's manual).

```
20 DIMC(13,4):FORI=1TO30:PRINT:NEXT
21 B$="PA1 PA1 THV UH3 UH3 UH3 PA0 D E L ER PA1"
25 PRINT"1 TO GET ANOTHER CARD":PRINT"2 TO DOUBLE YOUR BET"
30 PRINT"3 TO STAY"
40 S$=" OF ":S$(1)="HEARTS":S$(2)="DIAMONDS"
41 S$(3)="CLUBS":S$(4)="SPADES"
50 PRINT:INPUT"NUMBER OF PLAYERS (1-5)";N
55 IFN<>INT(N)ORN<1ORN>5GOTO50
70 FORI=1TO13:FORJ=1TO4:C(I,J)=0:NEXTJ,I
80 FORI=1TON+1:FORJ=1TO5:PC(I,J)=0:NEXTJ:SW(I)=0:NEXTI
81 PRINT"ENTER";:PRINT#5,"I N T ER PA1"
82 PRINT" YOUR";:PRINT#5,"Y U R PA1"
83 PRINT" BETS: ";:PRINT#5,"B EH T S PA"
85 FORI=1TO300:NEXT
90 PRINT:PRINT:FORI=1TON:PRINT:PRINT"PLAYER";
91 PRINT#5,"P L A ER PA1":PRINTI;:T=I+20:A$="":GOSUB1671
100 INPUTW(I):IFW(I)>0ANDW(I)<501GOTO110
101 PRINT:PRINT"GET";:PRINT#5,"G EH1 T PA0"
102 PRINT" SERIOUS!!";:PRINT#5,"S ER I AH2 S"
105 PRINT:PRINT:PRINT#5,"PA1 PA1 PA1":PRINT"PLAYER";:GOTO91
110 NEXT:FORI=1TO4:PRINT:NEXT
120 FORQ=1TON:FORI1=1TO2:GOSUB1100:PC(Q,I1)=IC:GOSUB1000
130 NEXTI1:PRINT
140 GOSUB1200:IFT=21THENGOSUB1305:SW(Q)=1:GOTO300
186 IFI1=3THENT=20+Q:A$="PA1 P L A ER PA1":GOSUB1671
187 IFI1=3THENGOSUB1200:GOSUB1450
190 PRINT"PLAYER"Q;:INPUTC:IFC<1ORC>3ORC<>INT(C)GOTO190
191 PRINT
200 IFC=1ORC=3GOTO230
220 W(Q)=W(Q)*2:PRINT#5,"D D UH1 UH1 B L PA Y U R PA B EH T PA0"
230 IFC=1GOTO236
235 IFC=3THENPRINT#5,"S T A A1 Y Y1 PA1":GOTO300
236 PRINT#5,"P L A ER PA0 T A K S PA0 W UH3 UH2 UH2 N N PA0"
240 GOSUB1100:PC(Q,I1)=IC:GOSUB1000
260 GOSUB1200:GOSUB1450:IFT>21THENGOSUB1400:GOTO300
265 IFI1=5THENSW(Q)=1:PRINT#5,"PA1 A A1 PA1 W I N ER PA1":GOTO300
270 I1=I1+1:PRINT:GOTO190
300 FORI=1TO300:NEXT:PRINT:NEXTQ
302 DO=0:DW=0
303 FORI1=1TO2:GOSUB1100:PC(Q,I1)=IC:GOSUB1000:NEXT
304 GOSUB1200:IFT=21THENGOSUB1305:SW(Q)=1:GOTO700
306 PRINT#5,RIGHT$(B$,12):GOSUB1450:T1=T
307 FORQ=1TON:GOSUB1200:IFT<=T1ORT>21ANDSW(Q)=0THENDO=DO+W(Q):GOTO309
308 IFSW(Q)=0THENDW=DW+W(Q)
309 NEXTQ:IFI1=6THENSW(N+1)=1:GOTO700
310 IFDW>0ANDT1<15GOTO325
316 IFN=1ANDDW<>0GOTO325
320 IFDO>DWORT1>18THENPRINT#5,B$+" S T A Y S PA0":GOTO690
325 PRINT#5,B$+" T A K S PA0 W UH3 UH2 UH2 N N PA0"
326 DO=0:DW=0
330 GOSUB1100:PC(Q,I1)=IC:GOSUB1000
340 GOSUB1200:GOSUB1450:IFT<22THENI1=I1+1:T1=T:GOTO307
350 GOSUB1400
690 REM
700 FORI=1TO5:PRINT:NEXT
710 FORI=1TON:H(I)=H(I)-W(I):NEXT
720 IFSW(I)=0GOTO726
721 PRINT#5,"H AH PA1 H AH PA1"
725 PRINT"THE HOUSE WINS (AGAIN)!!":GOTO790
726 PRINT"CONGRADULATIONS TO:":PRINT
730 S=0:FORI=1TON:IFSW(I)=0GOTO740
735 PRINT"  PLAYER"I:H(I)=H(I)+W(I)*2:S=1
740 NEXT
750 Q=N+1:GOSUB1200:PT=T
760 FORQ=1TON:GOSUB1200:IFPT<22ANDT<PT+1GOTO770
761 IFT>21ORSW(Q)=1GOTO770
765 PRINT"  PLAYER"Q:H(Q)=H(Q)+W(Q)*2:S=1
770 NEXT:IFS=0THENPRINT"THE HOUSE!":PRINT#5,"H AH PA1 H AH PA1"
790 PRINT:PRINT:PRINT
800 Q=N+1:GOSUB1200:PRINT"THE HOUSE HAD"T:PRINT:PRINT
810 FORQ=1TON:GOSUB1200
820 PRINT"PLAYER"Q,"SCORE"T,"WAGER"W(Q),"HOLDINGS"H(Q):NEXT
830 PRINT:PRINT:PRINT"DO";:PRINT#5,"D D U PA0"
831 PRINT" YOU";:PRINT#5,"Y U PA0"
832 PRINT" WANT";:PRINT#5,"W AH1 EH2 N T PA0"
833 PRINT" TO";:PRINT#5,"T OO O1 PA0"
834 PRINT" CONTINUE";:PRINT#5,"K UH N PA T I N Y U"
835 INPUTY$:IFLEFT$(Y$,1)="Y"GOTO70
```

```
850 PRINT:PRINT"YOU ";:PRINT#5,"Y U1 U PA0"
855 PRINT"MAY ";:PRINT#5,"M A Y PA0"
856 PRINT"PAY ";:PRINT#5,"P A Y PA0"
860 PRINT"YOUR ";:PRINT#5,"Y U R R PA0"
865 PRINT"DEBTS ";:PRINT#5,"D EH T S S PA1"
870 PRINT"AT ";:PRINT#5,"AE T T PA0"
875 PRINT"THE ";:PRINT#5,"TH UH UH1 PA0"
880 PRINT"FRONT ";:PRINT#5,"F R AH1 N T PA0"
885 PRINT"DESK";:PRINT#5,"D EH S K PA1 PA1"
886 PRINT", PLEASE!!":PRINT#5,"P L E S PA0"
890 RUN"BEXEC*
1000 PRINTC$;S$;S$(IS):RETURN
1100 IC=INT(RND(1)*13)+1:IS=INT(RND(1)*4)+1
1110 IFC(IC,IS)=1GOTO1100
1120 C(IC,IS)=1:C$=MID$(STR$(IC),2):IFIC=1THENC$="ACE"
1130 IFIC=11THENC$="JACK"
1140 IFIC=12THENC$="QUEEN"
1150 IFIC=13THENC$="KING"
1160 IFIC>10THENIC=10
1170 RETURN
1200 IC=0:T=0:FORI=1TO5:IFPC(Q,I)=1THENIC=IC+1
1210 T=T+PC(Q,I):NEXT
1220 IFIC=0THENRETURN
1230 FORI=1TOIC:IFT+10>21THENRETURN
1240 T=T+10:NEXT:RETURN
1305 PRINT#5,"B L AH1 E2 K PA0 J J EH3 AH1 K PA0"
1310 PRINT:PRINT"BLACKJACK!!":PRINT:RETURN
1400 PRINT#5,"B AH1 S T EH D":PRINT:PRINT"BUSTED!!":PRINT:RETURN
1450 A$="SH SH O I1 NG PA1 "
1455 ONT-1GOTO1470,1480,1490,1500,1510,1520,1530,1540,1550
1460 ONT-10GOTO1560,1570,1580,1590,1600,1610,1620,1630,1640
1465 ONT-19GOTO1670,1670,1670,1670,1670,1670,1670,1670,1670,1670
1466 ONT-28GOTO1670,1770,1780,1790
1470 A$=A$+"T IU IU U1 U1 PA1":GOTO1800
1480 A$=A$+"TH TH R E Y PA1":GOTO1800
1490 A$=A$+"F O1 O1 R PA1":GOTO1800
1500 A$=A$+"F AH2 AH1 I3 Y Y Y PA1":GOTO1800
1510 A$=A$+"S I I3 K PA0 S PA1":GOTO1800
1520 A$=A$+"S EH3 EH2 V EH2 N N PA1":GOTO1800
1530 A$=A$+"A2 A2 AY T PA1":GOTO1800
1540 A$=A$+"N AH1 EH2 Y N PA1":GOTO1800
1550 A$=A$+"T AE1 EH N PA1":GOTO1800
1560 A$=A$+"UH3 L EH1 V EH2 N PA1":GOTO1800
1570 A$=A$+"T W EH L V PA":GOTO1800
1580 A$=A$+"TH ER T E N PA":GOTO1800
1590 A$=A$+"F O1 R T E N PA":GOTO1800
1600 A$=A$+"F I V T E N PA":GOTO1800
1610 A$=A$+"S I1 K S T E1 E1 N":GOTO1800
1620 A$=A$+"S EH1 V EH3 N T E1 E1 N":GOTO1800
1630 A$=A$+"A1 AY Y1 T PA0 E1 E1 N PA":GOTO1800
1640 A$=A$+"N AH2 I3 Y N T E1 E1 N PA1":GOTO1800
1670 A$=A$+"T W EH1 N T E PA0"
1671 ONT-19GOTO1800,1680,1690,1700,1710,1720,1730,1740,1750,1760
1680 A$=A$+" W UH3 UH2 UH2 N N PA1":GOTO1800
1690 A$=A$+" T IU IU U1 U1 PA1":GOTO1800
1700 A$=A$+" TH TH R E Y PA1":GOTO1800
1710 A$=A$+" F O1 O1 R PA1":GOTO1800
1720 A$=A$+" F AH2 AH1 I3 Y Y Y PA1":GOTO1800
1730 A$=A$+" S I I3 K PA0 S PA1":GOTO1800
1740 A$=A$+" S EH3 EH2 V EH2 N N PA1":GOTO1800
1750 A$=A$+" A2 A2 AY Y T PA1":GOTO1800
1760 A$=A$+" N AH1 EH2 Y N PA1":GOTO1800
1770 A$=A$+"TH ER T E PA1":GOTO1800
1780 A$=A$+"TH ER T E PA0 W UH3 UH2 UH2 N N PA1":GOTO1800
1790 A$=A$+"TH ER T E PA0 ":GOTO1470
1800 PRINT#5,A$:RETURN
2000 IFC=1ORC=3GOTO230
```

Telecom Test ---

This program should be used with C2-4P's and C2-8P's with the universal telephone interface option. (CA-15) to test and demonstrate the versatility of this comprehenisve interface board.

```
1 REM TELCOM TEST PROGRAM
10 DIMN(16),N$(16),D(16)
50 DATA 238,222,190,237,221,189,235,219,187,215
52 DATA 231,183,126,125,123,119
55 DATA 1,2,3,4,5,6,7,8,9,0,*,#,A,B,C,D
56 FORI=1TO16:READN(I):NEXTI
57 FORI=1TO16:READN$(I):NEXTI
100 A=63488:FORI=ATOA+6STEP2:POKEI,255:NEXTI
105 FORI=A+1TOA+7STEP2:POKEI,0:NEXTI
110 DATA 63,15,240,255
120 FOR I=ATOA+7STEP2:READ8:POKE I,8:NEXTI
130 FORI=A+1TOA+7STEP2:POKEI,6:NEXTI:POKE63494,255
135 POKE 63496,3:POKE 63496,145:REM ACIA
140 PRINT"1 = ORIGINATE CALL PULSE DIALER
142 PRINT"2 = ORIGINATE CALL TONE  DIALER
144 PRINT"3 = HANG UP"
145 PRINT"4 = DECODE TONE INPUTS"
146 PRINT"5 = END PROGRAM & HANG UP"
147 PRINT"6 = CALL NATIONAL WEATHER SERVICE"
148 PRINT"7 = CALL TIME SERVICE"
149 PRINT"8 = AUTO ANSWER MODE"
150 PRINT"9 = FORCE OFF HOOK AND YOUR FUNCTION"
200 INPUT"COMMAND";C
210 ON C GOTO 1000,1000,2000,9000,9999,5000,5050,6000,6000
220 GOTO 200
1000 REM ORIGINATE
1010 INPUT"NUMBER PLEASE";A$
1020 L=LEN(A$):E=1:POKE 63490,0
1030 FORJ=1TOL
1040 FORI=1TO16:IFMID$(A$,J,1)=N$(I)THEND(J)=N(I):E=0
1050 NEXTI:IF E=1 THEN GOTO 1010
1060 E=1:NEXTJ
1070 REM # IS NOW IN D(0) - D(15) - # OF DIGITS IN L
1080 POKE 63490,1:REM OFF HOOK
1090 IF C=1 THEN POKE 63488,30:REM PULSE DIAL
1100 IF C=2 THEN POKE 63488,29:REM TONE DIAL
1110 REM WAIT 1 SECOND
1115 D=60:GOSUB8000
1120 REM BEGIN DIAL
1130 FORI=1TOL:POKE63494,D(I):D=1:GOSUB8000:POKE63494,255:D=1
1160 GOSUB8000:NEXTI:GOTO140
2000 REM HANG UP PHONE
2010 GOSUB 9090:GOTO140
5000 A$="9311212":C=2:GOTO1020
5050 A$="4711212":C=2:GOTO1020
6000 REM ANSWER MODE
6010 PRINT"3 = TAPE REC. "
6015 PRINT"4 = MODEM RCV. "
6020 PRINT"5 = TONE DECODER"
6050 INPUT"YOUR INPUT FUNCTION";R
6060 PRINT"0 = NO OUTPUT"
6065 PRINT"1 = VOTRAX OUTPUT"
6070 PRINT"2 = AUX OUTPUT"
6075 PRINT"3 = TAPE PLAYER OUTPUT"
6080 PRINT"4 = MODEM OUTPUT"
6090 PRINT"5 = TONE GEN. OUTPUT"
6094 INPUT"OUTPUT FUNCTION";X
6100 POKE 63488,((R*8)+X)
6105 IF C<>8 THEN GOTO 6120
6110 Z=PEEK(63488):Z=ZAND128:IF Z=0 THEN 6110
6120 GOSUB 9080
6130 IF R=3 GOTO 140
6132 IF R=4 THEN PRINT"INSERT YOUR MODEM PROGRAM":GOTO140
6134 IF R=5 GOTO 9001
6140 GOTO 6010
8000 FORT=1TOD:WAIT 56832,128,128:WAIT 56832,128:NEXTT:RETURN
9000 GOSUB 9080:POKE 63488,45
9001 PRINT:PRINT:PRINT"PRESS  (#)  TO EXIT THIS MODE & HANG UP"
9005 Z=PEEK(63493):Z=ZAND128:IFZ>0 THEN 9020
9010 GOTO9005
9020 F=PEEK(63492) AND 15:REM GET DATA AND RESET FLAG
9030 PRINT"YOU PRESSED ";N$(F):IFN$(F)="#"THENGOSUB9090:GOTO140
9040 GOTO9005
9080 POKE 63490,1:RETURN:REM LIFT HOOK
9090 POKE 63490,0:RETURN
9999 POKE 63490,0:END
```

Little Guy ---

This program will run on any graphics machine except the C1P with color (optional tone generator). Using keys 1-4 you move a little man around the screen trying to find the door.

```
5 C=3
10 LS=50    :REM LOW WHOOP
12 HS=100   :REM HIGH WHOOP
14 SS=2     :REM WHOOP STEP
16 OK=192     :REM GUY MUST MOVE!!!! ON START
20 DEF FNR(X)=INT(X*RND(SD))
80 INPUT"INPUT SEED";SD
90 GOTO 100
99 POKE 2073,173 : POKE 56832,1 : END
100 FORL=0TO35:PRINT:NEXTL
150 POKE56832,4
200 A=4096
300 BL=14
400 FORL=53248TO55295:POKEL+A,BL:NEXTL
500 GOSUB30000:REM *** MAKE A RANDOM DOOR ***
600 GOSUB30100:REM *** MAKE A LITTLE GUY  ***
700 GOSUB31000:REM *** LET THE USER MOVE
800 END
29999 REM*** RANDOM DOOR ***
30000 DX=FNR(32)
30010 DY=FNR(32)
30020 RD=53248+DY*64+DX
30030 POKERD+A,C
30050 RETURN
30099 REM*** LITTLE GUY ***
30100 GX=FNR(32)
30110 GY=FNR(32)
30120 LG=53248+GY*64+GX
30130 IF LG=RD THEN 30100
30140 POKELG,240
30150 RETURN
30200 REM*** MOVE LITTLE GUY UP(1) ***
30210 POKE LG,32
30220 IF GY<>0 THEN 30230
30222 GY=31
30224 LG=LG+1984
30226 POKELG,240
30228 RETURN
30230 GY=GY-1
30240 LG=LG-64
30250 POKELG,240
30260 RETURN
30300 REM*** MOVE LITTLE GUY DOWN(5) ***
30310 POKE LG,32
30320 IF GY<>31 THEN 30330
30322 GY=0
30324 LG=LG-1984
30326 POKE LG,240
30328 RETURN
30330 GY=GY+1
30340 LG=LG+64
30350 POKE LG,240
30360 RETURN
30400 REM*** MOVE LITTLE GUY RIGHT(3) ***
30410 POKE LG,32
30420 IF GX<>31 THEN 30430
30422 GX=0
30424 LG=LG-31
30426 POKE LG,240
30428 RETURN
30430 GX=GX+1
30440 LG=LG+1
30450 POKE LG,240
30460 RETURN
30500 REM*** MOVE LITTLE GUY LEFT(7) ***
30510 POKE LG,32
30520 IF GX<>0 THEN 30530
30522 GX=31
30524 LG=LG+31
30526 POKE LG,240
30528 RETURN
30530 GX=GX-1
30540 LG=LG-1
30550 POKE LG,240
30560 RETURN
30600 REM*** MOVE LITTLE GUY UP+RIGHT(2)
30610 POKELG,32
30620 IF GY<>0 THEN 30630
30622 GY=31
30623 GX=GX+1
30624 LG=LG+1985
30626 IF GX=31 THEN 30632
30628 POKE LG,240
30629 RETURN
30630 IF GX<>31 THEN 30640
30631 GY=GY-1
30632 GX=0
30634 LG=LG-95
30636 POKE LG,240
30638 RETURN
30640 GX=GX+1
30650 GY=GY-1
30660 LG=LG-63
30670 POKE LG,240
30680 RETURN
30700 REM*** MOVE LITTLE GUY DOWN+LEFT(6)
30710 POKE LG,32
30720 IF GY<>31 THEN 30730
30722 GY=0
30724 GX=GX-1
30726 LG=LG-1985
30727 IF GX=0 THEN 30732
30728 POKE LG,240
30729 RETURN
30730 IF GX<>0 THEN 30740
30731 GY=GY+1
30732 GX=31
30734 LG=LG+95
30736 POKE LG,240
30738 RETURN
30740 GX=GX-1
30750 GY=GY+1
30760 LG=LG+63
30770 POKE LG,240
30780 RETURN
```

```
30800 REM*** MOVE LITTLE GUY DOWN+RIGHT(4
30810 POKE LG,32
30820 IF GY<>31 THEN 30830
30822 GY=0
30823 GX=GX+1
30824 LG=LG-1983
30826 IF GX=31 THEN 30832
30828 POKE LG,240
30829 RETURN
30830 IF GX<>31 THEN 30840
30831 GY=GY+1
30832 GX=0
30834 LG=LG+33
30836 POKE LG,240
30838 RETURN
30840 GX=GX+1
30850 GY=GY+1
30860 LG=LG+65
30870 POKE LG,240
30880 RETURN
30900 REM*** MOVE LITTLE GUY UP+LEFT(8)
30910 POKE LG,32
30920 IF GY<>0 THEN 30930
30922 GY=31
30923 GX=GX-1
30924 LG=LG+1983
30926 IF GX=0 THEN 30932
30928 POKE LG,240
30929 RETURN
30930 IF GX<>0 THEN 30940
30931 GY=GY-1
30932 GX=31
30934 LG=LG-33
30936 POKE LG,240
30938 RETURN
30940 GX=GX-1
30950 GY=GY-1
30960 LG=LG-65
30970 POKE LG,240
30980 RETURN
31000 REM*** POLL KEYBOARD ***
31005 J=FNR(8)+1
31010 POKE2073,96:REM*** NO ↑C ***
31020 KB=57088:R1=128
31025 UP=128: DO=32: LT=16: RT=64
31027 UR=192: UL=144:DR=96: DL=48
31030 POKE KB,R1
31040 K=PEEK(KB)
31050 IF K=UP THEN 31200
31060 IF K=UR THEN 31210
31070 IF K=RT THEN 31220
31080 IF K=DR THEN 31230
31090 IF K=DO THEN 31240
31100 IF K=DL THEN 31250
31110 IF K=LT THEN 31260
31120 IF K=UL THEN 31270
31125 IF K=2  THEN 99
31140 K=OK : GOTO 31050
31200 GOSUB30200
31202 GOTO 31300
31210 GOSUB30600
31212 GOTO 31300
31220 GOSUB 30400
31222 GOTO 31300
31230 GOSUB 30800
31232 GOTO 31300
31240 GOSUB 30300
31242 GOTO 31300
31250 GOSUB 30700
31252 GOTO 31300
31260 GOSUB 30500
31262 GOTO 31300
31270 GOSUB 30900
31300 OK=K : REM SAVE OLD DIRECTION
31305 IF LG<>RD THEN 31030
31310 POKE2073,173:REM** ↑C WORKS!!! **
31320 REM** MAKE A WHOOP FOR FINDING DOOR
31330 POKE 56832,6
31340 POKE57089,0
31350 FOR I=1 TO 5
31360 FOR J=LS TO HS STEP SS
31370 POKE57089,J
31380 NEXT J
31390 FOR J=HS TO LS STEP -SS
31400 POKE 57089,J
31410 NEXT J
31420 NEXT I
31430 POKE56832,4
31440 FOR J=1 TO 300:NEXT J
31450 POKE RD+A,BL
31460 GOSUB30000
31465 POKE 2073,96:REM** NO ↑C AGAIN!! **
31470 GOTO31300
40000 RETURN
```

COPIER ----

This program will run on any disk based machine with a polled keyboard except the C1P. It will copy all tracks except track 0. Place new disk into the drive, and initialize it. Take out the initialized disk and place the master into the drive while copier is in memory. Run the copier. Depress the space bar when "OLD DISK" appears on the screen and the screen will fill with the contents of that track. Now, remove the master disk and place new disk into the drive. Depress the space bar and "OLD DISK" will appear again. Replace new disk with the master and repeat procedure until program runs to completion. After copying tracks 1-39 for 5" floppies (or 1-76 for 8" floppies), you must copy track 0 as follows. First type "EXIT" and A* will appear. Type "CA 0200=13,1" for 5 inch floppies or "CA 0200=01,2" for 8 inch floppies followed by RETURN and A* will reappear. Type "GO 0200" followed by RETURN. Type "2" ,then RETURN. Type "R4200",then RETURN while the master disk is in the drive. Remove the master disk from the drive and replace it with the new disk. Type "W4200/2200,8" ,then RETURN. Exit and depress the BREAK key and then "D". NOTE:copy drive A to drive A if you have are using a one drive system.

```
1 FORI=1TO30:PRINT:NEXT
2 POKE2073,96
5 PRINT"COPIER"
6 PRINT:PRINT"COPIES ALL TRACKS EXCEPT ZERO"
7 GOSUB1000
10 DISK!B$
20 FORI=A+100TOB+100
25 Z$=RIGHT$(STR$(I),2)
26 GOSUB5000
27 FORJ=1TOX
28 P$=RIGHT$(STR$(J),1)
30 A$="CA D000="+Z$+",1"+P$
40 DISK!A$
45 IFC$=B$THENPOKE57088,2:F=PEEK(57088):IFF<>16THEN45
50 DISK!C$
60 A$="SA "+Z$+",1"+P$+"=D000/"+X$(J)
70 DISK!A$
75 PRINT"***** OLD  DISK *******"
76 IFC$=B$THENPOKE57088,2:F=PEEK(57088):IFF<>16THEN76
80 DISK!B$
90 NEXTJ,I
100 END
1000 PRINT:PRINT
1010 INPUT"COPY FROM DRIVE";B$
1020 IFB$<>"A"ANDB$<>"B"ANDB$<>"C"ANDB$<>"D"THEN1010
1030 INPUT"COPY TO DRIVE";C$
1040 IFC$<>"A"ANDC$<>"B"ANDC$<>"C"ANDC$<>"D"THEN1030
1045 B$="SE "+B$:C$="SE "+C$
1050 PRINT:PRINT
1060 INPUT"FROM TRACK";A
1065 IFA>39ORA<1ORA<>INT(A)THEN1060
1070 INPUT"TO TRACK";B
1075 IFB>39ORB<0ORB<>INT(B)THEN1070
1080 IFA>BTHEN1050
1090 PRINT"******* OLD  DISK ********"
1100 INPUT"READY";Q$:IFLEFT$(Q$,1)<>"Y"THENRUN
2000 RETURN
5000 DISK!"DIR "+Z$
5100 X=VAL(CHR$(PEEK(55106)))
5200 FORJ=0TOX:X$(J+1)=CHR$(PEEK(55109-J*64)):NEXT
5300 RETURN
```

PROGRAM CONVERSIONS ----

The ASCII keyboard is located at the address 57343 in memory. Note, the polled keyboard is located at the address 57088. First, look through the program for all variables being assigned a value by PEEKing this location. The PEEK is usually followed by comparisons with that variable to determine which key was depressed. The two things that will differ in programs written for ASCII keyboards and polled keyboards are, the keyboard is at a different address in memory and the comparison values are different. Here are some conversions from ASCII to polled done with the help of "The Challenger Character Graphics Reference Manual".

```
100 Q=PEEK(57343)
200 IF Q=177 THEN 500
```

This is a sample of two lines from a program written for ASCII keyboards. You must now subtract 128 from the comparison value (this is a hardware consideration).

```
177-128=49
```

49 is the ASCII value for a "1" (i.e., ASC("1")=49), so the program is looking for a "1" to be depressed on the keyboard. Using page 11 of the character graphics manual, the code modified for polled keyboards becomes:

```
100 POKE57088,128 : Q=PEEK(57088)
200 IF Q=128 THEN 500
```

EXAMPLE #2

ASCII program:

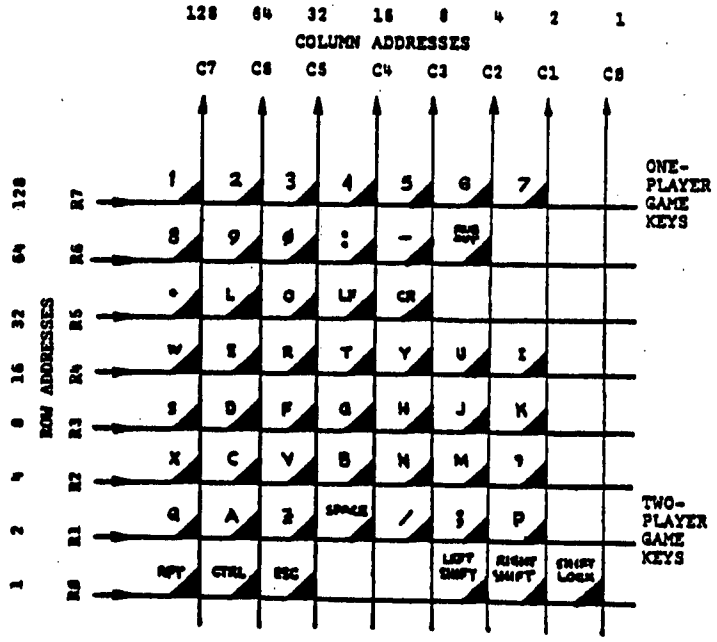
```
100 Q=PEEK(57343)
200 IF Q=212 THEN X=1
300 IF Q=198 THEN X=2
```

SUBTRACT:

```
212-128=84      ASC("T")=84      (looking for a "T")
198-128=70      ASC("F")=70      (looking for an "F")
```

POLLED program:

```
100 POKE57088,16 : Q=PEEK(57088)
200 IF Q=16 THEN X=1
210 POKE57088,8 : Q=PEEK(57088)
300 IF Q=32 THEN X=2
```



USR(X) FOR COLOR BACKGROUND ----

This is a BASIC program that sets up an ASSEMBLER subroutine under the USR(X) function. The subroutine changes the background color of the entire screen. Note, if a disk system is not used then the BASIC code; DISK!"CA 4FD0=36,1"; must be removed from the program.

To save the assembler program (created by this BASIC program) on disk, type DISK!"SA 36,1=4FD0/1" after running the program. This will let you call the program from disk in any other BASIC program by the command DISK!"CA 4FD0=36,1" instead of running this BASIC code.

Use the following code in BASIC (after the assembler program is loaded into memory) to exicute the assembler routine. NOTE: this must be done after the subroutine is in memory.

POKE8955,208:POKE8956,79

This is the high and low addresses to tell the computer where the USR(X) function is located in memory.

POKE20433,(your color choice, 0-16)

This is your choice of color background.

X=USR(X)

This is the BASIC command for jumping to an assembler subroutine specified by the previous POKEs.

```
100 FORI=20432TO20473:READX:POKEI,X:NEXT
200 DATA162,14,169,0,141,242,79,169,224,141,243,79,173,242,79
210 DATA24,105,1,141,242,79,173,243,79,105,0,141,243,79,201,232
220 DATA240,6,142,0,224,76,220,79,96,0,2
```

MEMORY TEST ---

The following memory test is written for 4 and 8K BASIC in ROM machines. By changing lines 0 and 1 the program is easily adapted to your particular requirements. This program should be run anytime your computer "acts funny" as a quick diagnostic.

```
0 Y=8191
1 INPUT"4 OR 8K";K:IFK=4THENY=4095
2 P=1025
5 Q=255
10 FORX=PTOY
12 POKEX,Q
13 NEXTX
20 FORX=PTOY
22 Z=PEEK(X)
23 IFZ<>QTHENGOSUB100
25 NEXTX:PRINT" PASS USING ";Q
26 IFQ=0THEN5
30 Q=0:GOTO10
100 PRINT"LOCATION";X;"WAS";Z;"NOT";Q:RETURN
```

Mini-Floppy Disk Directory

```
OS-65D VERSION 3.0
 -- DIRECTORY --

FILE NAME     TRACK RANGE
-----------------------
OS65D3        0  - 12
BEXEC*        14 - 14
CHANGE        15 - 16
CREATE        17 - 19
DELETE        20 - 20
DIR           21 - 21
DIRSRT        22 - 22
RANLST        23 - 24
RENAME        25 - 25
SECDIR        26 - 26
SEQLST        27 - 28
TRACE         29 - 29
ZERO          30 - 31
ASAMPL        32 - 32

 50 ENTRIES FREE OUT OF 64
```

Full Size Floppy Disk Directory

```
OS-65D VERSION 3.0
 -- DIRECTORY --

FILE NAME     TRACK RANGE
-----------------------
OS65D3        0  - 8
BEXEC*        9  - 9
CHANGE        10 - 10
CREATE        13 - 14
DELETE        15 - 15
DIR           16 - 16
DIRSRT        17 - 17
RANLST        18 - 19
RENAME        20 - 20
SECDIR        21 - 21
SEQLST        22 - 23
TRACE         24 - 24
ZERO          25 - 26
ASAMPL        27 - 27

 50 ENTRIES FREE OUT OF 64
```

```
10 REM BASIC EXECUTIVE
20 REM
24 REM SETUP INFLAG & OUFLAG FROM DEFAUL
25 X=PEEK(10950): POKE 8993,X: POKE 8994,X
30 PRINT : PRINT "BASIC EXECUTIVE FOR OS-65D VERSION 3.0" : PRINT
40 PRINT "13 OCT 1978 RELEASE"
100 PRINT
110 PRINT "FUNCTIONS AVAILABLE:"
120 PRINT "   CHANGE - ALTER WORKSPACE LIMITS"
130 PRINT "   DIR    - PRINT DIRECTORY"
140 PRINT "   UNLOCK - UNLOCK SYSTEM FROM END USER MODIFICATIONS"
10000 REM
10010 REM UNLOCK SYSTEM
10020 REM
10030 REM REPLACE "NEW" AND "LIST"
10040 POKE 741,76 : POKE 750,78
10050 REM
10060 REM ENABLE CONTROL-C
10070 POKE 2073,173
10080 REM
10090 REM DISABLE "REDO FROM START"
10100 POKE 2893,55 : POKE 2894,8
10110 PRINT : PRINT "SYSTEM OPEN" : END
```

```
10 REM CHANGE PARAMETER UTILITY UNDER OS-65D VERSION 3.0
20 REM
30 PRINT : PRINT "CHANGE PARAMETER UTILITY" : PRINT
35 GOSUB 1000 : REMARK  DO EVERYTHING ELSE FIRST
36 PRINT : INPUT "CHANGE BASIC'S WORKSPACE LIMITS (Y/N)";A$
37 IF A$<>"Y" THEN END
40 INPUT "HOW MANY 12 PAGE BUFFERS DO YOU WANT BEFORE THE WORKSPACE";B
50 IF B<0 OR B>2 THEN PRINT "ANSWER 0, 1, OR 2" : GOTO 40
60 L=12670+B*3072 : REM L=$317E PLUS B*$C00
70 IF B<>0 THEN 130
80 INPUT "WANT TO LEAVE ANY ROOM BEFORE THE WORKSPACE";A$
90 IF MID$(A$,1,1)<>"Y" THEN 170
100 INPUT "HOW MANY BYTES";B
110 L=L+B
120 GOTO 170
130 INPUT "WANT TO LEAVE ANY ADDITIONAL ROOM";A$
140 IF MID$(A$,1,1)<>"Y" THEN 170
150 INPUT "HOW MANY BYTES";B
160 L=L+B
170 PRINT:PRINT "THE BASIC WORKSPACE WILL BE SET TO START AT";L
180 PRINT "LEAVING";L-12670;"BYTES FREE IN FRONT OF THE WORKSPACE"
190 INPUT "IS THAT ALRIGHT";A$
200 IF MID$(A$,1,1)="Y" THEN 210
204 INPUT "NEW LOWER LIMIT";L : IF L<12670 THEN 204
206 PRINT L-12670;"BYTES WILL BE FREE BEFORE THE WORKSPACE"
210 MP=PEEK(8960)
220 PRINT:PRINT "YOU HAVE";(MP+1)/4;"K OF RAM"
230 U=(MP+1)*256
240 INPUT "DO YOU WANT TO LEAVE ANY ROOM AT THE TOP";A$
250 IF MID$(A$,1,1)<>"Y" THEN 280
260 INPUT "HOW MANY BYTES";B
270 U=U-B
280 PRINT:PRINT "THE BASIC WORKSPACE WILL BE SET TO END AT";U
290 PRINT "LEAVING";(MP+1)*256-U;"BYTES FREE AFTER THE WORKSPACE"
300 INPUT "IS THAT ALRIGHT";A$
310 IF MID$(A$,1,1)="Y" THEN 320
312 INPUT "NEW UPPER LIMIT";U : IF U>49152 OR U<L+3 THEN 312
314 PRINT (MP+1)*256-U;"BYTES WILL BE FREE AFTER THE WORKSPACE"
320 PRINT:PRINT"YOU WILL HAVE";U-L+1;"BYTES FREE IN THE WORKSPACE"
330 INPUT "IS THAT ALRIGHT";A$
340 IF MID$(A$,1,1)<>"Y" THEN RUN
350 REM
360 REM NOW DO THE ADJUSTMENTS
370 REM
380 POKE132,U-INT(U/256)*256:POKE133,INT(U/256)
390 POKE120,L+1-INT((L+1)/256)*256:POKE121,INT((L+1)/256):POKEL,0:NEW
1000 REM
1010 REM
1020 PRINT "THE TERMINAL WIDTH IS SET FOR";PEEK(23)
1030 INPUT "DO YOU WANT TO CHANGE IT (Y/N)";A$
1040 IF A$<>"Y" THEN 1100
1050 INPUT "NEW VALUE";WD
1060 IF WD<14 OR WD>255 THEN PRINT "BAD VALUE" : GOTO 1050
1070 POKE 23,WD
1080 NC=(INT(WD/14)-1)*14
1090 POKE 24,NC
1100 PRINT : PRINT "BASIC & THE ASSEMBLER USE";(PEEK(8960)+1)/4;
1110 PRINT "K WORKSPACES (";PEEK(8960);"PAGES"
1120 INPUT "WOULD YOU LIKE TO CHANGE THIS (Y/N)";A$
1130 IF A$<>"Y" THEN 1170
1140 INPUT "HOW MANY PAGES SHOULD THEY USE";KK
1150 IF KK<50 OR KK>191 THEN PRINT "BAD VALUE" : GOTO 1140
1160 POKE 8960,KK
1170 RETURN
```

```
10 REM FILE CREATE UTILITY OF OS-65D VERSION 3.0
20 REM
40 REM EXECUTE:
50 REM
100 REM 3D7E A000    ZERO    LDY #0
110 REM 3D80 A20C            LDX #12
120 REM 3D82 98              TYA
130 REM 3D83 997E31 ONE      STA $317E,Y
140 REM 3D86 C8              INY
150 REM 3D87 D0FA            BNE ONE
160 REM 3D89 EE853D          INC ONE+2
170 REM 3D8C CA              DEX
180 REM 3D8D D0F4            BNE ONE
190 REM 3D8F 60              RTS
200 REM
230 DATA 160,0,162,12,152,153,126,49,200,208,250
240 DATA 238,133,61,202,208,244,96
250 FORI=15742TO15759:READD:POKEI,D:NEXT
280 POKE8955,126:POKE8956,61:X=USR(X):POKE8955,212:POKE8956,34
320 PRINT:PRINT"FILE CREATION UTILITY":PRINT
330 PN=11897
335 INPUT"PASSWORD";A$:IFA$<>"PASS"THENEND
337 PRINT
340 DEF FNA(X)=16*INT(X/10)+X-10*INT(X/10)
350 DEF FNB(X)=10*INT(X/16)+X-16*INT(X/16)
360 DATA 1,2,3,4,5,6,7,8,9,A,B,C
370 PRINT "CREATES AN ENTRY IN DIRECTORY FOR A NEW FILE"
380 PRINT "AND INITIALIZES THE TRACKS THAT THE NEW FILE WILL"
390 PRINT "RESIDE ON.  THE TRACKS WILL CONTAIN NULLS WITH A"
400 PRINT "RETURN AT THE END OF THE TRACK."
410 PRINT
420 INPUT"FILE NAME";A$
430 IF LEN(A$)>6 THEN 410
440 IF LEN(A$)<6 THEN A$=A$+" ":GOTO 440
450 IFMID$(A$,1,1)<"A"ORMID$(A$,1,1)>"Z"THEN420
460 REM CHECK ALLOCATION + WATCH FOR NEW NAME
490 DIM AL%(76)
500 FORI=0TO76:AL%(I)=0:NEXT
510 DISK!"CA 2E79=08,2":GOSUB10000
520 DISK!"CA 2E79=08,1":GOSUB10000
530 REM
540 REM FIND FREE ENTRY IN DIRECTORY
560 GOSUB 20000
570 IF FL=1 THEN DISK!"SA 08,1=2E79/1":END
580 DISK!"CA 2E79=08,2":GOSUB10000
590 IF FL=1 THEN DISK!"SA 08,2=2E79/1":END
600 PRINT"** DIRECTORY FULL **":END
10000 REM
10010 REM SUBROUTINE USED BY ALLOCATION CHECKER
10030 FOR I=PN+6 TO PN+254 STEP 8
10040 IF PEEK(I-6)=35 THEN 10100
10050 C$="":FORK=1TO6:C$=C$+CHR$(PEEK(I-7+K)):NEXTK:IFC$<>A$THEN10090
10080 PRINT"** FILE NAME ";CHR$(34);A$;CHR$(34);" IN USE **":RUN
10090 T0=FNB(PEEK(I)) : T9=FNB(PEEK(I+1))
10095 FORK=T0TOT9:AL%(K)=-1:NEXTK
10100 NEXT I
10110 RETURN
20000 REM
20010 REM SUBROUTINE USED TO ENTER NEW NAME INTO DIRECTORY
```

```
20030 FOR I=PN TO PN+248 STEP 8
20040 IF PEEK(I)=35 THEN 20080
20050 NEXT I
20060 FL=0:RETURN
20080 PRINT : INPUT "FIRST TRACK OF FILE";T0
20090 IF T0<9 OR T0>76 THEN 20080
20100 INPUT "NUMBER OF TRACKS IN FILE";NT
20110 IF NT<1 OR NT>68 THEN 20100
20115 T9=T0+NT-1
20120 FK=0
20130 FOR K=T0 TO T9
20140 IF AL%(K) THEN PRINT"** TRACK";K;"IN USE **":FK=1
20150 NEXT K
20160 IF FK<>0 THEN RUN
20170 PG=12
20180 INPUT "12 PAGES PER TRACK.  IS THIS OK";B$
20190 IF MID$(B$,1,1)="Y" THEN 20220
20200 INPUT "HOW MANY PAGES PER TRACK THEN";PG
20210 IFPG<>12THENPRINT:PRINT"NOTE: ALL DEFAULTS ARE SET FOR 12 PAGES!"
20220 FORJ=0TO5:POKEI+J,ASC(MID$(A$,J+1,1)):NEXTJ
20250 POKE I+6,FNA(T0) : POKE I+7,FNA(T9)
20260 FORI=1TOPG:READP$:NEXTI
20270 FOR I=T0 TO T9
20280 T$=RIGHT$(STR$(I+100),2)
20290 DISK!"IN "+T$ : DISK!"SA "+T$+",1=317E/"+P$
20310 NEXT I
20320 FL=1:RETURN
```

```
10 REM DELETE FILE UTILITY UNDER OS-65D VERSION 3.0
20 REM
30 PRINT:PRINT"DELETE UTILITY":PRINT
32 PRINT "REMOVES AN ENTRY FROM THE DIRECTORY":PRINT
33 INPUT "PASSWORD";A$ : IF A$<>"PASS" THEN END
34 PRINT
35 FLAG=0 : PN=11897
40 INPUT "FILE NAME";A$
45 IF LEN(A$)>6 THEN 40
47 IF LEN(A$)<6 THEN A$=A$+" " : GOTO 47
50 DISK ! "CALL 2E79=08,1"
60 GOSUB 10000 : IF FLAG=1 THEN DISK ! "SAVE 08,1=2E79/1" : END
70 DISK ! "CALL 2E79=08,2"
80 GOSUB 10000 : IF FLAG=1 THEN DISK ! "SAVE 08,2=2E79/1" : END
90 PRINT "** ";CHR$(34);A$;CHR$(34);" NOT FOUND IN DIRECTORY **" : PRINT
100 END
10000 REM
10010 REM SEE IF FILE NAME A$ IS IN DIRECTORY BUFFER
10020 REM
10030 FOR I=PNT TO PN+248 STEP 8
10040 FOR J=I TO I+5
10050 IF PEEK(J)<>ASC(MID$(A$,J-I+1,1)) THEN 10100
10060 NEXT J
10070 FOR J=I TO I+5 : POKE J,ASC("#") : NEXT J
10080 POKE I+6,0 : POKE I+7,0
10090 FLAG=1 : RETURN
10100 NEXT I
10110 FLAG=0 : RETURN
```

```
10 REM DIRECTORY UTILITY FOR OS-65D VERSION 3.0
20 REM
30 NF=0
40 PN=11897
50 DEF FNA(X)=10*INT(X/16)+X-16*INT(X/16)
80 DV=1 : Y=1 : X=PEEK(8994)
90 IF X=<Y THEN 110
100 DV=DV+1 : Y=Y+Y : GOTO 90
110 PRINT "LIST ON LINEPRINTER INSTEAD OF DEVICE #";DV;
120 INPUT A$
130 IF MID$(A$,1,1)="Y" THEN DV=4
10000 REM
10010 REM PRINT A DIRECTORY OUT
10020 REM
10030 PRINT #DV : PRINT #DV,"OS-65D VERSION 3.0"
10035 PRINT #DV," -- DIRECTORY --" : PRINT #DV
10040 PRINT #DV,"FILE NAME    TRACK RANGE"
10050 PRINT #DV,"------------------------"
10060 DISK ! "CALL 2E79=08,1"
10070 GOSUB 11000
10080 DISK ! "CALL 2E79=08,2"
10090 GOSUB 11000
10130 PRINT #DV : PRINT #DV,NF;"ENTRIES FREE OUT OF 64" : PRINT #DV
10140 END
11000 REM
11010 REM READ DIRECTORY OUT OF BUFFER INTO ARRAYS
11020 REM
11040 FOR I=PN TO PN+248 STEP 8
11050 IF PEEK(I)=35 THEN NF=NF+1 : GOTO 11130
11060 N$=""
11070 FOR J=I TO I+5
11080 N$=N$+CHR$(PEEK(J))
11090 NEXT J
11100 PRINT #DV,N$;TAB(12);FNA(PEEK(I+6));TAB(16);"-";
11110 PRINT #DV,TAB(17);FNA(PEEK(I+7))
11130 NEXT I
11140 RETURN
```

```
10 REM SORTED DIRECTORY UTILITY FOR OS-65D VERSION 3.0
20 REM
30 NF=0
40 PN=11897
50 DEF FNA(X)=10*INT(X/16)+X-16*INT(X/16)
60 DIM NM$(64),T0%(64),T9%(64)
70 AV=0
80 DV=1 : Y=1 : X=PEEK(8994)
82 IF X=<Y THEN 86
84 DV=DV+1 : Y=Y+Y : GOTO 82
86 PRINT:PRINT "SORTED DIRECTORY UTILITY":PRINT
90 INPUT "SORTED BY NAME OR TRACK (N/T)";Z$
95 PRINT "LIST ON LINEPRINTER INSTEAD OF DEVICE #";DV;
96 INPUT A$ : IF MID$(A$,1,1)="Y" THEN DV=4
100 IF Z$="N" OR Z$="T" THEN 10000
110 PRINT "THEN IT WILL BE UNSORTED"
10000 REM
10010 REM PRINT A DIRECTORY OUT
10020 REM
10030 PRINT #DV : PRINT #DV,"OS-65D VERSION 3.0"
10035 PRINT #DV," -- DIRECTORY --" : PRINT #DV
10040 PRINT #DV,"FILE NAME    TRACK RANGE"
10050 PRINT #DV,"-----------------------"
10060 DISK ! "CALL 2E79=08,1"
10070 GOSUB 11000
10080 DISK ! "CALL 2E79=08,2"
10090 GOSUB 11000
10095 IF Z$="N" THEN GOSUB 20000
10097 IF Z$="T" THEN GOSUB 21000
10100 FOR I=0 TO AV-1
10110 PRINT #DV,NM$(I);TAB(12);T0%(I);TAB(16);"-";TAB(17);T9%(I)
10120 NEXT I
10130 PRINT #DV : PRINT #DV,64-AV;"ENTRIES FREE OUT OF 64" : PRINT #DV
10140 END
11000 REM
11010 REM READ DIRECTORY OUT OF BUFFER INTO ARRAYS
11020 REM
11040 FOR I=PN TO PN+248 STEP 8
11050 IF PEEK(I)=35 THEN 11130
11060 N$=CHR$(PEEK(I))+CHR$(PEEK(I+1))+CHR$(PEEK(I+2))+CHR$(PEEK(I+3))
11070 NM$(AV)=N$+CHR$(PEEK(I+4))+CHR$(PEEK(I+5))
11100 T0%(AV)=FNA(PEEK(I+6))
11110 T9%(AV)=FNA(PEEK(I+7))
11120 AV=AV+1
11130 NEXT I
11140 RETURN
20000 REM
20010 REM SORT DIRECTORY BY NAME
20015 REM  (SHELL METZNER SORT)
20020 REM
20022 M=AV-1
20025 M=INT(M/2)
20030 IF M=0 THEN RETURN
20032 J=0 : K=AV-1-M
20040 I=J
20050 L=I+M
20070 IF NM$(I)<NM$(L) THEN 20120
20080 T$=NM$(I):NM$(I)=NM$(L):NM$(L)=T$
20090 T%=T0%(I):T0%(I)=T0%(L):T0%(L)=T%
20100 T%=T9%(I):T9%(I)=T9%(L):T9%(L)=T%
20105 I=I-M
20110 IF I>=0 THEN 20050
20120 J=J+1
20130 IF J>K THEN 20025
20140 GOTO 20040
21000 REM
21010 REM SORT DIRECTORY BY TRACK
21020 REM
21022 M=AV-1
21025 M=INT(M/2)
21030 IF M=0 THEN RETURN
21032 J=0 : K=AV-1-M
21040 I=J
21050 L=I+M
21070 IF T0%(I)<T0%(L) THEN 21120
21080 T$=NM$(I):NM$(I)=NM$(L):NM$(L)=T$
21090 T%=T0%(I):T0%(I)=T0%(L):T0%(L)=T%
21100 T%=T9%(I):T9%(I)=T9%(L):T9%(L)=T%
21105 I=I-M
21110 IF I>=0 THEN 21050
21120 J=J+1
21130 IF J>K THEN 21025
21140 GOTO 21040
```

```
10 REM RANDOM ACCESS FILE LIST UTILITY UNDER OS-65D VERSION 3.0
20 REM
30 PRINT : PRINT "RANDOM ACCESS FILE READ" : PRINT
40 INPUT "FILE NAME";N$
50 IF LEN(N$)>6 THEN 40
70 DISK OPEN,6,N$
75 INPUT "EXAMINE SINGLE RECORDS OR GROUPS (S/G)";R$
77 IF R$="G" THEN 200
78 IF R$<>"S" THEN 75
80 PRINT : INPUT "RECORD NUMBER";R
90 DISK GET,R
100 INPUT #6,A$
110 PRINT : PRINT A$
120 GOTO 80
200 PRINT : INPUT "FIRST RECORD";R0
210 INPUT "LAST RECORD";R9
220 IF R9<R0 THEN 200
230 FOR R=R0 TO R9
240 DISK GET,R
250 INPUT #6,A$
260 PRINT : PRINT A$
270 NEXT R
280 GOTO 75
```

```
10 REM RENAME FILE UTILITY UNDER OS-65D VERSION 3.0
20 REM
30 PRINT : PRINT "RENAME UTILITY" : PRINT
35 FLAG=0 : PN=11897
40 INPUT "OLD NAME";A$
45 IF LEN(A$)>6 THEN 40
47 IF LEN(A$)<6 THEN A$=A$+" " : GOTO 47
50 DISK ! "CALL 2E79=08,1"
60 GOSUB 10000 : IF FLAG=1 THEN DISK ! "SAVE 08,1=2E79/1" : END
70 DISK ! "CALL 2E79=08,2"
80 GOSUB 10000 : IF FLAG=1 THEN DISK ! "SAVE 08,2=2E79/1" : END
90 PRINT "** ";CHR$(34);A$;CHR$(34);" NOT FOUND IN DIRECTORY **"
100 END
10000 REM
10010 REM SEE IF FILE NAME A$ IS IN DIRECTORY BUFFER
10020 REM
10030 FOR I=PNT TO PN+248 STEP 8
10040 FOR J=I TO I+5
10050 IF PEEK(J)<>ASC(MID$(A$,J-I+1,1)) THEN 10100
10060 NEXT J
10070 PRINT "RENAME ";CHR$(34);A$;CHR$(34); : INPUT " TO";AN$
10075 IF LEN(AN$)>6 THEN 10070
10080 IF LEN(AN$)<6 THEN AN$=AN$+" " : GOTO 10080
10082 IF MID$(AN$,1,1)<"A" OR MID$(AN$,1,1)>"Z" THEN 10070
10085 FOR J=I TO I+5 : POKE J,ASC(MID$(AN$,J-I+1,1)) : NEXT J
10090 FLAG=1 : RETURN
10100 NEXT I
10110 FLAG=0 : RETURN
```

```
10 REM SECTOR DIRECTORY UTILITY UNDER OS-65D VERSION 3.0
15 REM
20 PRINT : PRINT "SECDIR" : PRINT
22 PRINT "USES OS-65D'S DIR COMMAND TO PRINT OUT A SECTOR"
24 PRINT "MAP OF A GIVEN RANGE OF TRACKS" : PRINT
30 PRINT : INPUT "FIRST TRACK";T0
40 IF T0<1 OR T0>76 THEN 30
50 PRINT : INPUT "LAST TRACK";T9
60 IF T9<T0 OR T9>76 THEN 50
70 PRINT : PRINT "SECTOR MAP DIRECTORY" : PRINT
80 FOR I=100+T0 TO 100+T9
90 DISK ! "DIR "+RIGHT$(STR$(I),2)
95 PRINT
100 NEXT I
110 END
```

```
10 REM SEQUENTIAL FILE LISTER UTILITY UNDER OS-65D VERSION 3.0
20 REM
30 PRINT : PRINT "SEQUENTIAL FILE LISTER" : PRINT
40 PRINT "TYPE A CONTROL-C TO STOP"
60 PRINT : INPUT "FILE NAME";A$
70 IF LEN(A$)>6 THEN 60
90 DISK OPEN,6,A$
100 INPUT #6,D$
110 PRINT D$
120 GOTO 100
```

```
10 REM TRACE UTILITY UNDER OS-65D VERSION 3.0
20 REM
30 PRINT : PRINT "TRACE UTILITY" : PRINT
35 PRINT "WHEN BASIC'S TRACE FEATURE IS ENABLED, BASIC WILL PRINT"
37 PRINT "OUT EACH LINE NUMBER OF THE PROGRAM BEFORE IT IS EXECUTED"
38 PRINT
40 INPUT "ENABLE OR DISABLE (E/D)";A$
50 IF A$="E" THEN 100
60 IF A$="D" THEN 200
70 GOTO 40
100 REM
110 REM ENABLE
120 REM
130 REM THIS MUST ALL BE DONE AS ONE LINE!
140 REM
150 L=2011:POKEL,32:POKEL+1,216:POKEL+2,28:POKEL+3,234:POKEL+4,234
160 END
200 REM
210 REM DISABLE
220 REM
230 REM THIS MUST ALL BE DONE AS ONE LINE!
240 REM
250 L=2011:POKEL,24:POKEL+1,144:POKEL+2,2:POKEL+3,230:POKEL+4,200
260 END
```

```
10 REM FILE ZEROING UTILITY OF OS-65D VERSION 3.0
20 REM
40 REM TO ZERO OUT DATA BUFFER: EXECUTE THIS:
50 REM
100 REM 3D7E A000    ZERO   LDY #0
110 REM 3D80 A20C           LDX #12
120 REM 3D82 98             TYA
130 REM 3D83 997E31 ONE     STA $317E,Y
140 REM 3D86 C8             INY
150 REM 3D87 D0FA           BNE ONE
160 REM 3D89 EE853D         INC ONE+2
170 REM 3D8C CA             DEX
180 REM 3D8D D0F4           BNE ONE
190 REM 3D8F 60             RTS
200 REM
230 DATA 160,0,162,12,152,153,126,49,200,208,250
240 DATA 238,133,61,202,208,244,96
250 FORI=15742TO15759:READD:POKEI,D:NEXT
280 POKE8955,126:POKE8956,61:X=USR(X):POKE8955,212:POKE8956,34
285 DEF FNA(X)=10*INT(X/16)+X-16*INT(X/16)
290 PRINT:PRINT"FILE ZERO UTILITY":PRINT
300 PRINT"COMPLETELY ERASES THE CONTENTS OF A DATA FILE"
310 PRINT
320 INPUT "PASSWORD";A$ : IF A$<>"PASS" THEN END
330 INPUT "FILE NAME";A$
340 IF LEN(A$)>6 THEN 330
350 IF LEN(A$)<6 THEN A$=A$+" " : GOTO 350
360 PN=11897
370 DISK!"CA 2E79=08,1":GOSUB 10000
380 IF FL<>0 THEN 405
390 DISK!"CA 2E79=08,2":GOSUB 10000
400 IF FL=0 THEN PRINT "** FILE NAME NOT IN DIRECTORY **":END
405 PG=12
410 PRINT:INPUT"IS IT A NORMAL 12 PAGE DATA FILE";AA$
420 IF MID$(AA$,1,1)="Y" THEN 450
430 INPUT "THEN HOW MANY PAGES PER TRACK";PG
440 IF PG<1 OR PG>12 THEN 430
450 DATA 1,2,3,4,5,6,7,8,9,A,B,C
460 FOR I=1 TO PG : READ P$ : NEXT I
480 FOR I=T0 TO T9
490 T$=RIGHT$(STR$(I+100),2)
500 DISK!"SA "+T$+",1=317E/"+P$
510 NEXT I
520 END
10000 REM
10010 REM FIND A$ IN DIRECTORY
10020 REM
10030 FOR I=PN TO PN+248 STEP 8
10040 B$=""
10050 FOR K=I TO I+5 : B$=B$+CHR$(PEEK(K)) : NEXT K
10060 IF A$<>B$ THEN 10090
10070 T0=FNA(PEEK(I+6)) : T9=FNA(PEEK(I+7))
10080 FL=1 : RETURN
10090 NEXT I
10100 FL=0 : RETURN
```

OHIO SCIENTIFIC
1333 S. Chillicothe Road • Aurora, OH 44202
Phone: (216) 562-3101

Printed in U.S.A.