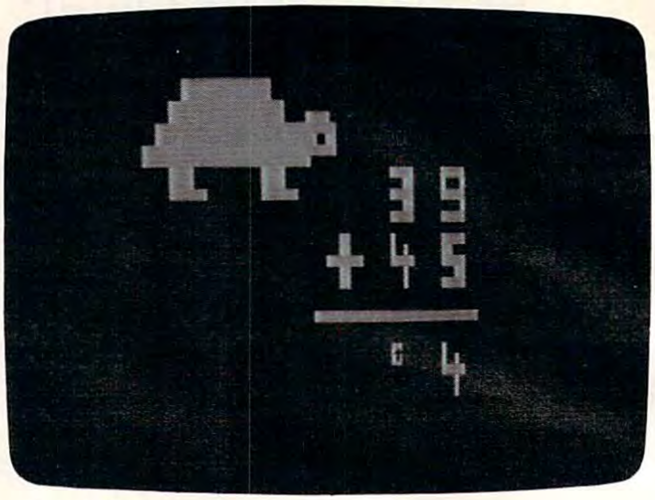
```
2340 PRINT TAB(6);"(ENTER 1 TO 14)"
     ::
2350 INPUT K
2360 IF (K<1)+(K>14)THEN 2350
2370 S=0
2380 R=14
2390 GOTO 490
2400 REM  DRAW THE SHELL
2410 R5=5
2420 COL=13
2430 FOR I=1 TO 4
2440 CALL HCHAR(I,COL,96,R5)
2450 R5=R5+2
2460 COL=COL-1
2470 NEXT I
2480 CALL HCHAR(5,9,96,12)
2490 RETURN
2500 REM  DRAW THE HEAD
2510 CALL HCHAR(3,21,97)
2520 CALL HCHAR(3,22,96,2)
2530 CALL HCHAR(4,21,96,3)
2540 CALL HCHAR(4,22,128)
2550 CALL HCHAR(5,21,96,3)
2560 RETURN
2570 REM  DRAW THE FEET AND TAIL
2580 FOR I=1 TO 8
2590 READ R5,C
2600 CALL HCHAR(R5,C,96)
2610 NEXT I
2620 RESTORE
2630 DATA 6,9,6,12,6,18,7,12,7,13,7
     ,18,7,19,5,22
2640 RETURN
2650 REM  ERASE THE HEAD
2660 FOR I=3 TO 5
2670 CALL HCHAR(I,21,32,3)
2680 NEXT I
2690 RETURN
2700 REM  DEFINE CHARS & COLORS
2710 CALL CHAR(96,"FFFFFFFFFFFFFFFF
     ")
2720 CALL CHAR(97,"0103070F1F3F7FFF
     ")
2730 CALL CHAR(104,"000000FFFF00000
     0")
2740 CALL CHAR(128,"000000000F0F0F0
     F")
2750 CALL CHAR(136,"3030180C0703000
     0")
2760 CALL COLOR(9,3,1)
2770 CALL COLOR(13,6,16)
2780 CALL COLOR(14,14,3)
2790 CALL CHAR(112,"00000000FFFFFFF
     F")
2800 CALL CHAR(113,"F0F0F0F0F0F0F0F
     0")
2810 CALL CHAR(114,"070707070707070
     7")
2820 CALL CHAR(115,"FFFFFFFFFFFFFFF
     F")
2830 RETURN
2840 IF LEN(F$)=2 THEN 2890
2850 P=VAL(SEG$(F$,1,1))
2860 X=W+4
2870 GOSUB 1430
2880 RETURN
2890 P=VAL(SEG$(F$,1,1))
2900 X=W
2910 GOSUB 1430
2920 P=VAL(SEG$(F$,2,1))
2930 X=W+4
2940 GOSUB 1430
2950 RETURN
2960 CALL VCHAR(14,11,115,3)
2970 CALL HCHAR(15,10,115)
2980 CALL HCHAR(15,12,115)
2990 IF Q=2 THEN 3010
3000 RETURN
3010 CALL HCHAR(14,11,32)
3020 CALL HCHAR(16,11,32)
3030 RETURN
3040 CALL HCHAR(14,9,115)
3050 CALL HCHAR(14,11,115)
3060 CALL HCHAR(15,10,115)
3070 CALL HCHAR(16,9,115)
3080 CALL HCHAR(16,11,115)
3090 RETURN
3100 END
```

Program 5: Snertle For The Color Computer

```
100 CLS(1):B$=CHR$(32)
110 PRINT@74,"**SNERTLE**"
120 PRINT@138,"SELECT 1"
130 PRINT@202,"1) ADDITION"
140 PRINTTAB(10)"2) SUBTRACTION"
150 PRINTTAB(10)"3) MULTIPLICATION"
155 PRINTTAB(10)"4) END"
160 PRINTTAB(10)"(ENTER 1,2,3 OR 4)
    ";:INPUTQ:IF Q>4 OR Q<1 THEN 16
    0
185 C=14:IF Q=1 OR Q=2 THEN C=99
187 IF Q=3 THEN 1000
188 IF Q=4 THEN END
190 CLS(1):PRINT@37,"ENTER LARGEST
    VALUE"
200 PRINTTAB(5)"(MIN.:1  MAX.:";C;"
    )";:INPUTR:IF R<1 OR R>C THEN 2
    00
230 PRINT@133,"ENTER SMALLEST VALUE
    "
240 PRINTTAB(5)"(MIN.:0  MAX.:";R;"
    )";:INPUTS:IF S<0 OR S>R THEN 2
    40
263 CLS:PRINT@227,"PRESS X TO RETUR
    N TO MENU";:FORI=1TO750:NEXTI:C
    LS(0)
270 Z=0:ZZ=0
275 GOSUB 1100:GOSUB 1170:GOSUB1230
301 TR=0:ZZ=ZZ+1
305 L=INT(RND(R-S)+S)
310 IF Q=3ANDT=1THEN320
315 K=INT(RND(R-S)+S)
320 F$=STR$(K):W=0
325 IF K<L AND Q=2  THEN TR=0:GOTO3
    05
330 W=0:GOSUB3000
335 W=64
340 F$=STR$(L)
345 W=96:GOSUB 3000
346 ON Q GOSUB 6000,6000,6004
350 IF Q=1 THEN M=K+L
355 IF Q=2 THEN M=K-L
360 IF Q=3 THEN M=K*L
380 MM=1:IF M>9 THEN MM=2
385 IF M>99 THEN MM=3
390 GOSUB 740
393 V=0:GOSUB 1100
395 FOR J=0 TO MM-1
397 POKE 1466-(4*J),94
399 HH$=INKEY$
400 H$=INKEY$
405 IF H$=""THEN 400
410 IF H$="X" AND ZZ=1 THEN 100
```

```
411 IF H$="X" THEN CLS(1):PRINT@68,"YOUR PERCENTAGE IS ";INT(Z/(ZZ-1)*100):GOTO120
413 IF H$<>"0" AND VAL(H$)=0 THEN 400
415 P=VAL(H$)
420 V=V+(P*10^J):X=1466-(4*J):GOSUB 480:NEXTJ
450 IF INT(M)=INT(V) THEN 470
451 SOUND 80,6:FORI=1TO20:NEXTI:SOUND 80,6:FORI=1TO20:NEXTI:SOUND60,12
452 FOR I=1439 TO 1535:POKEI,128:NEXT I
456 IF TR=1 THEN 460
458 TR=1:GOSUB 1500:GOSUB 770:GOTO393
460 M$=STR$(M)
461 FORI=1 TO 11-MM:READA:NEXTI
462 FOR OO=1 TO MM
464 P=VAL(MID$(M$,(OO+1),1))
465 READX:GOSUB 480:NEXT OO:RESTORE
470 GOSUB1170:IF TR=0THEN GOSUB 2500: GOSUB 755:Z=Z+1:GOSUB 6500
471 GOSUB 2225:GOTO301
480 IF P=0 THEN 720
485 ON P GOSUB 500,525,555,585,610,633,660,680,700:RETURN
500 POKEX,143:POKEX+32,143:POKEX+64,140:RETURN
525 POKEX,140:POKEX+1,143:POKEX+33,140:POKEX+32,143:POKEX+64,140:POKEX+65,140
530 RETURN
555 POKEX,140:POKEX+32,140:POKEX+64,140:POKEX+65,140
560 POKE X+1,143:POKEX+33,143:RETURN
585 POKEX,138:POKEX+32,140:POKEX+1,130:POKEX+33,142
590 POKEX+64,128:POKEX+65,136:RETURN
610 POKEX,143:POKEX+32,140:POKEX+64,140
615 POKEX+1,140:POKEX+33,143:POKEX+65,140:RETURN
633 POKEX,143:POKEX+32,143:POKEX+64,140:POKEX+1,140
635 POKE X+33,141:POKEX+65,140:RETURN
660 POKE X,140:POKEX+32,129:POKEX+64,132
670 POKEX+65,128:POKE X+1,141:POKEX+33,138:RETURN
680 POKEX,142:POKEX+32,142:POKEX+64,140:POKEX+65,140
685 POKEX+1,141:POKEX+33,141:RETURN
700 POKEX,142:POKEX+32,140:POKEX+64,140
710 POKEX+1,141:POKEX+33,141:POKEX+65,140:RETURN
720 POKEX,142:POKEX+1,141:POKEX+32,138:POKEX+33,133
725 POKEX+64,140:POKEX+65,140:RETURN
740 FORI=1392 TO 1404:POKEI,131:NEXTI:RETURN
755 PRINT@103,"GOOD";:FORI=1TO500:NEXTI:RETURN
770 PRINT@72,"TRY";:PRINT@103,"AGAIN";:FOR I=1 TO 500:NEXTI:RETURN
```



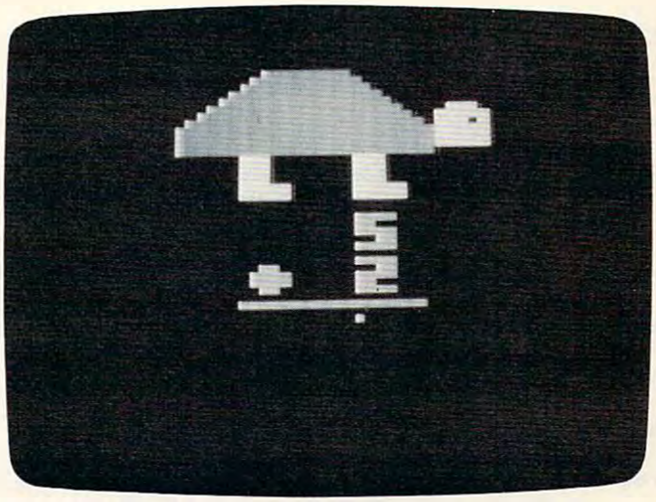
"Snertle," Color Computer version.

```
1000 CLS(1):PRINT@66,"DO YOU WISH TO:"
1010 PRINT@130,"1) PRACTICE TIMES TABLES"
1020 PRINT@162,"2) RANDOM NUMBERS"
1030 PRINT@224,"(ENTER 1 OR 2)";:INPUTT:IF T<1 OR T>2 THEN 1030
1050 IF T=2 THEN 1090
1060 CLS(1):PRINT@66,"ENTER TIMES TABLE"
1070 PRINT@100,"(1-14)";:INPUT K:IF K<1 OR K>14 THEN 1070
1090 S=0:R=14:GOTO 263
1100 FOR I=1056 TO 1152 STEP 32
1110 READ A,B
1120 FOR J=1TOB
1130 POKEI+J+A,143
1140 NEXTJ:NEXTI:RESTORE:RETURN
1170 POKE 1169,140:POKE1167,140:POKE1168,140
1180 POKE 1103,129:POKE1104,131:POKE1105,130
1190 POKE1135,143:POKE1136,142:POKE 1137,143:RETURN
1230 POKE 1196,143:POKE1197,143:POKE1189,143:POKE1190,143
1240 POKE 1228,140:POKE1229,140:POKE1230,140:POKE1221,140:POKE1222,140:POKE1223,140:RETURN
1500 FORI=1103 TO 1167 STEP 32:FOR J=0 TO 3:POKE I+J,128:NEXTJ:NEXTI:POKE 1167,143:RETURN
2225 FOR I=1140 TO 1236 STEP 32
2230 FOR J=1 TO 11:POKEJ+I,128:NEXTJ:NEXTI:FOR I=1260 TO 1535 STEP 32
2235 FOR J=1 TO 16:POKE J+I,128:NEXTJ:NEXTI:RETURN
2500 POKE 1167,139:RETURN
3000 IF LEN(F$)>2 THEN 3030
3015 P=VAL(MID$(F$,2,1))
3020 X=1210+W:GOSUB480
3025 RETURN
3030 P=VAL(MID$(F$,2,1))
3035 X=1206+W:GOSUB480
3040 P=VAL(MID$(F$,3,1))
3045 X=1210+W:GOSUB480
3050 RETURN
5000 DATA 5,7,4,9,3,11,2,13,1458,1462,1466
```

```
6000 POKE 1298,143:POKE1330,143:POK
     E 1362,140:POKE 1331,140:POKE1
     329,140
6001 IF Q=2 THEN POKE 1298,128:POKE
     1330,140:POKE1362,128
6003 RETURN
6004 POKE 1297,131:POKE1299,131:POK
     E1330,140:POKE1329,131:POKE133
     1,131:RETURN
6500 SOUND 100,7:SOUND130,10
6510 RETURN
```

Program 6: Snertle For Apple

```
110  TEXT : HOME : VTAB 2: HTAB 15: PRINT
     "**SNERTLE**": VTAB 5
120  PRINT : VTAB 5: HTAB 10: PRINT "SE
     LECT ONE:"
130  PRINT : PRINT : HTAB 10: PRINT "1)
      ADDITION"
140  PRINT : HTAB 10: PRINT "2) SUBTRAC
     TION"
150  PRINT : HTAB 10: PRINT "3) MULTIPL
     ICATION"
155  PRINT : HTAB 10: PRINT "4) END PRO
     GRAM"
160  PRINT : PRINT : HTAB 10: PRINT "(E
     NTER 1,2,3 OR 4) ";: INPUT Q: IF Q
      < 1 OR Q > 4 THEN 160
185 C = 14: IF Q = 1 OR Q = 2 THEN C =
     99
187  IF Q = 3 THEN 1000
188  IF Q = 4 THEN  END
190  HOME : VTAB 3: HTAB 10: PRINT "ENT
     ER LARGEST VALUE"
200  HTAB 10: PRINT "(MIN.:1  MAX.:";C;
     ")";: INPUT R: IF R < 1 OR R > C THEN
     200
230  HTAB 10: VTAB 10: PRINT "ENTER SMA
     LLEST VALUE"
240  HTAB 10: PRINT "(MIN.:0  MAX.:";R;
     ")";: INPUT S: IF S < 0 OR S > R THEN
     240
263  HOME : VTAB 10: HTAB 7: PRINT "TYP
     E ";: INVERSE : PRINT "X";: NORMAL
     : PRINT " TO RETURN TO THE MENU"
265  FOR I = 1 TO 2000: NEXT I: HOME
270 Z = 0:ZZ = 0: GR
275  GOSUB 1100: COLOR= 12: GOSUB 1170:
      GOSUB 1230
301 TR = 0:ZZ = ZZ + 1
305 L =  INT ( RND (1) * (R - S + 1)) +
     S
310  IF Q = 3 AND T = 1 THEN 320
315 K =  INT ( RND (1) * (R - S + 1)) +
     S
320 F$ =  STR$ (K):W = 0
325  IF K < L AND Q = 2 THEN 305
330 W = 0: GOSUB 3000
340 F$ =  STR$ (L)
345 W = 6: GOSUB 3000
346  ON Q GOSUB 6000,6000,6004
350  IF Q = 1 THEN M = K + L
355  IF Q = 2 THEN M = K - L
365  IF Q = 3 THEN M = K * L
380  GOSUB 740:MM = 1: IF M > 9 THEN MM
      = 2
385  IF M > 99 THEN MM = 3
393 V = 0: COLOR= 12: GOSUB 1170
395  FOR J = 0 TO MM - 1
397  COLOR= 1: PLOT 21 - (5 * J),34
399  POKE  - 16368,0
400 H$ = "":H =  PEEK ( - 16384) - 128:
      IF H > 0 THEN H$ =  CHR$ (H)
```



"Snertle," Apple version.

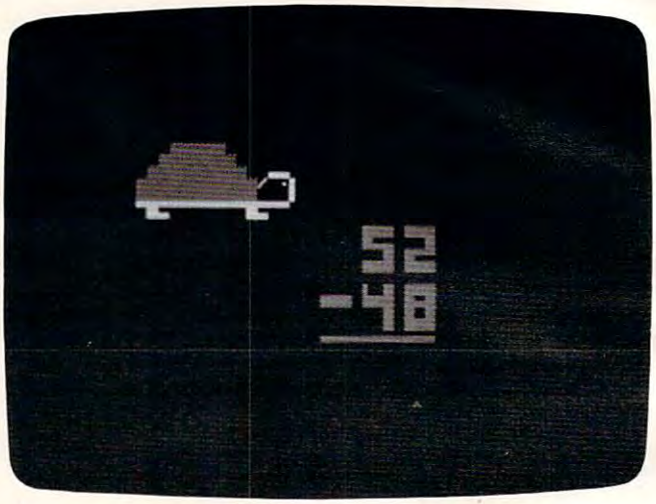
```
407  IF H$ = "X" AND ZZ = 1 THEN  POKE
      - 16368,0: GOTO 110
410  IF H$ = "X" THEN  TEXT : HOME : HTAB
     15: PRINT "PERCENTAGE="; INT (Z /
     (ZZ - 1) * 100): POKE  - 16368,0: GOTO
     120
412  IF H < 48 OR H > 57 THEN 400
415 P =  VAL (H$)
420 V = V + (P * 10 ^ J):W = 14:X = 21 -
     (5 * J): GOSUB 480: NEXT J
450  IF M = V THEN 470
451  FOR I = 1 TO 40: FOR J = 1 TO 2: NEXT
     J:L =  PEEK ( - 16336): NEXT I
452  COLOR= 0: FOR I = 33 TO 38: HLIN 7
     ,34 AT I: NEXT I: COLOR= 1
456  IF TR = 1 THEN 460
458 TR = 1: COLOR= 0: GOSUB 1170: GOSUB
     770:V = 0: GOTO 395
460 M$ =  STR$ (M)
461  IF MM < 3 THEN  FOR I = 1 TO 3 - M
     M: READ X: NEXT I
462  FOR OO = 1 TO MM
464 P =  VAL ( MID$ (M$,OO,1))
465  READ X: GOSUB 480: NEXT OO: RESTORE

467  FOR I = 1 TO 900: NEXT
470  COLOR= 12: GOSUB 1170: IF TR = 0 THEN
      GOSUB 2500: GOSUB 755:Z = Z + 1: GOSUB
     6500: HOME
471  GOSUB 2225: GOTO 301
480  COLOR= 1: IF P = 0 THEN  GOSUB 720

485  ON P GOSUB 500,525,555,585,610,633
     ,660,680,700: RETURN
500  VLIN 20 + W,24 + W AT X: VLIN 20 +
     W,24 + W AT X + 1: RETURN
525  HLIN X,X + 3 AT 20 + W: PLOT X + 2
     ,21 + W: PLOT X + 3,21 + W: HLIN X
     ,X + 3 AT 22 + W
530  VLIN 23 + W,24 + W AT X: VLIN 23 +
     W,24 + W AT X + 1: PLOT X + 2,24 +
     W: PLOT X + 3,24 + W: RETURN
555  VLIN 20 + W,24 + W AT X + 2: PLOT
     X,20 + W: PLOT X,22 + W: PLOT X,24
      + W
560  PLOT X + 1,20 + W: PLOT X + 1,22 +
     W: PLOT X + 1,24 + W: RETURN
585  VLIN 20 + W,22 + W AT X: PLOT X +
     1,22 + W: VLIN 20 + W,24 + W AT X +
     2: PLOT X + 3,22 + W: RETURN
```

```
610  HLIN X,X + 3 AT 20 + W: HLIN X,X +
     3 AT 22 + W: HLIN X,X + 3 AT 24 +
     W: PLOT X + 2,23 + W: PLOT X + 3,2
     3 + W
615  PLOT X,21 + W: PLOT X + 1,21 + W: RETU
     RN
633  VLIN 20 + W,24 + W AT X: VLIN 20 +
     W,24 + W AT X + 1: VLIN 22 + W,24 +
     W AT X + 3: HLIN X + 2,X + 3 AT 20
      + W
635  PLOT X + 2,22 + W: PLOT X + 2,24 +
     W: RETURN
660  HLIN X + 1,X + 3 AT 20 + W: PLOT X
      + 3,21 + W: PLOT X + 2,22 + W
665  VLIN 23 + W,24 + W AT X + 1: RETURN

680  GOSUB 720: HLIN X + 1,X + 2 AT 22 +
     W: RETURN
700  HLIN X,X + 3 AT 20 + W: HLIN X,X +
     3 AT 22 + W: HLIN X,X + 3 AT 24 +
     W: VLIN 20 + W,24 + W AT X + 3
705  VLIN 21 + W,22 + W AT X: RETURN
720  VLIN 20 + W,24 + W AT X: VLIN 20 +
     W,24 + W AT X + 3: HLIN X + 1,X +
     2 AT 20 + W: HLIN X + 1,X + 2 AT 2
     4 + W: RETURN
740  HLIN 10,27 AT 32: RETURN
755  VTAB 21: HTAB 19: PRINT "GOOD!": FOR
     I = 1 TO 300: NEXT I: RETURN
770  VTAB 21: HTAB 16: PRINT "TRY AGAIN
     ": FOR I = 1 TO 1000: NEXT I: HOME
     : RETURN
1000  HOME : VTAB 4: HTAB 13: PRINT "DO
      YOU WISH TO:"
1010  PRINT : HTAB 9: PRINT "1) PRACTIC
     E TIMES TABLES"
1020  PRINT : HTAB 9: PRINT "2) PRACTIC
     E RANDOM NUMBERS"
1030  PRINT : HTAB 9: PRINT "(ENTER 1 O
     R 2) ";: INPUT T: IF T < 0 OR T >
     2 THEN 1030
1050  IF T = 2 THEN 190
1060  HOME : VTAB 5: HTAB 11: PRINT "EN
     TER TIMES TABLE (1-14)"
1070  INPUT K: IF K < 1 OR K > 14 THEN
     1070
1090 S = 0:R = 14: GOTO 263
1100 J = 12:JJ = 20: COLOR= 4: FOR I =
     0 TO 8: HLIN J,JJ AT I:J = J - 1:J
     J = JJ + 1
1110  NEXT I: FOR I = 8 TO 11: HLIN J +
     1,JJ - 1 AT I: NEXT I: RETURN
1170  HLIN 30,32 AT 5: FOR I = 6 TO 10:
      HLIN 29,33 AT I: NEXT I: COLOR= 0
     : PLOT 32,7: RETURN
1230  COLOR= 12: FOR I = 12 TO 15: HLIN
     10,12 AT I: HLIN 21,23 AT I: NEXT
     I
1240  FOR I = 16 TO 17: HLIN 10,14 AT I
     : HLIN 21,25 AT I: NEXT I: RETURN
2225  COLOR= 0: FOR I = 20 TO 38: HLIN
     10,39 AT I: NEXT I: COLOR= 1: RETURN
2500  COLOR= 0: PLOT 32,10: PLOT 31,9: COLO
      R= 1: RETURN
3000  IF  LEN (F$) > 1 THEN 3030
3015 P =  VAL ( MID$ (F$,1,1))
3020 X = 21: GOSUB 480
3025  RETURN
3030 P =  VAL ( MID$ (F$,1,1))
3035 X = 16: GOSUB 480
3040 P =  VAL ( MID$ (F$,2,1))
3045 X = 21: GOSUB 480
3050  RETURN
5000  DATA   12,16,22
6000  HLIN 11,14 AT 29: HLIN 11,14 AT 2
     8: IF Q = 1 THEN  VLIN 27,30 AT 12
     : VLIN 27,30 AT 13
6001  RETURN
6004  PLOT 12,27: PLOT 14,27: PLOT 13,2
     8: PLOT 12,29: PLOT 14,29: RETURN
6500  FOR I = 1 TO 20:L =  PEEK ( - 163
     36): NEXT I: FOR I = 1 TO 10: NEXT
     I: FOR I = 1 TO 40:L =  PEEK ( - 1
     6336): NEXT I: RETURN
```



"Snertle," PC/PCjr version.

Program 7: Snertle For PC/PCjr

```
10 DEF SEG=0:POKE 1047,192
20 SCREEN 0,1:WIDTH 40:KEY OFF
25 S$=CHR$(219):D$=CHR$(31):L$=CHR$(29):
R$=CHR$(28):U$=CHR$(30):TB$=CHR$(223):BB
$=CHR$(220):LB$=CHR$(221):RB$=CHR$(222):
SP$=CHR$(32)
100 B$=CHR$(13):C$=CHR$(9)
110 COLOR 12:CLS:LOCATE 24,9,0:PRINT"***
**** SNERTLE *******"
120 PRINT B$B$B$B$B$C$"       SELECT ONE:
"
130 COLOR 2:PRINT B$C$"1) ADDITION"
140 COLOR 4:PRINT B$C$"2) SUBTRACTION"
150 COLOR 6:PRINT B$C$"3) MULTIPLICATION
"
155 COLOR 14:PRINT B$C$"4) END PROGRAM"
160 PRINT B$B$B$B$C$"(ENTER 1,2,3 OR 4)"
;
170 Q$=INKEY$:X=RND(1):Q=VAL(Q$):IF Q<1
OR Q>4 THEN 170
175 C=14:IF Q=1 OR Q=2 THEN C=99
185 C=14:IF Q=1 OR Q=2 THEN C=99
187 IF Q=3 THEN 1000
188 IF Q=4 THEN END
190 CLS:LOCATE 10,12:PRINT "ENTER LARGES
T VALUE"
200 PRINT:PRINT    "(MIN.:0 MAX.:";C;")";
:INPUT R:IF R<0 OR R>C THEN PRINT U$U$U$
:GOTO 200
230 PRINT:PRINT   "ENTER SMALLEST VALUE"
240 PRINT:PRINT    "(MIN.:0 MAX.:";R;")";
:INPUT S :IF S<0 OR S>R THEN PRINT U$U$U
$:GOTO 240
263 CLS:LOCATE 12,5:PRINT "PRESS ' X ' T
O RETURN TO MENU":FOR I =1 TO 1000 :NEXT
 I
```

```
265 CLS
270 Z=0:ZZ=0
275 COLOR 2:GOSUB 1100:GOSUB 1170:GOSUB
1230:GOSUB 1260: COLOR Q *2
301 TR=0:ZZ=ZZ+1
305 L=INT(RND(1)*(R-S+1))+S
310 IF Q=3 AND T=1 THEN 320
315 K=INT(RND(1)*(R-S+1))+S
320 F$=STR$(K):W=0
325 IF K<L THEN W=5
330 GOSUB 3000
335 W=5
337 IF L>K THEN W=0
340 F$= STR$(L)
345 GOSUB 3000
346 ON Q GOSUB 6000,6000,6004
350 IF Q=1 THEN M=K+L
355 IF Q=2 AND K>=L THEN M=K-L
360 IF Q=2 AND K<L THEN M=L-K
365 IF Q=3 THEN M=K*L
380 GOSUB 740:MM=1:IF M>9 THEN MM=2
385 IF M>99 THEN MM=3
390 GOSUB 740
393 V=0:COLOR 2 :GOSUB 1100:COLOR Q*2
394 FOR A=1 TO 10:B$=INKEY$:NEXT
395 FOR J=0 TO (MM-1)
397 LOCATE 24,30-4*J:PRINT"^";
400 H$=INKEY$
405 IF H$="X"AND ZZ=1 THEN 100
406 IF H$="X" THEN CLS:PRINT B$"PERCENTA
GE:";INT(Z/(ZZ-1)*100):GOTO 120
407 IF H$="" OR H$<"0" OR H$>"9" THEN 40
0
412 FOR I= 21 TO 31:LOCATE 24,I:PRINT SP
$;:NEXT
415 P=VAL (H$):Y=20
420 V=V+(P*10^J):X=29-J*4:GOSUB 475:NEXT
 J
450 IF M=V THEN 470
452 FOR I= 20 TO 23:LOCATE I,21:FOR J=1
TO 11:PRINT SP$;:NEXT J,I
456 IF TR =1 THEN 460
458 TR =1:GOSUB 1500:GOSUB 770:GOTO 393
460 M$ =STR$(M):X =33:Y=20
462 FOR OO=MM TO 1 STEP -1
464 P = VAL (MID$ (M$,(OO+1),1))
465 X=X-4:GOSUB 475:NEXT OO:RESTORE
470 FOR I=1 TO 750:NEXT:GOSUB 1230: IF T
R=0 THEN GOSUB 2500::GOSUB 755: Z=Z+1:GO
SUB 6500
471 GOSUB 2225: GOTO 301
475 LOCATE Y,X
480 IF P=0 THEN GOSUB 720
485 ON P GOSUB 500,525,555,585,610,633,6
60,680,700:RETURN
500 PRINT R$R$;:FOR I=1 TO 4 :PRINT S$D$
L$;:NEXT :RETURN
525 PRINT S$S$S$D$L$S$D$L$TB$L$L$TB$L$L$
S$D$L$S$S$S$:RETURN
555 PRINT S$S$S$D$L$S$D$L$S$L$L$TB$D$L$L
$S$S$S$:RETURN
585 PRINT LB$R$S$D$L$L$L$S$S$S$D$L$S$D$L
$S$:RETURN
610 PRINT S$S$S$D$L$L$L$S$BB$BB$D$L$S$D$
L$L$L$S$S$S$:RETURN
633 PRINT S$S$S$D$L$L$L$S$BB$BB$D$L$L$L$
S$R$S$D$L$L$L$S$S$S$:RETURN
660 PRINT S$S$S$D$L$S$D$L$L$S$D$L$L$S$:R
ETURN
680 PRINT S$S$S$D$L$L$L$S$BB$S$D$L$L$L$S
$R$S$D$L$L$L$S$S$S$:RETURN
700 PRINT S$S$S$D$L$L$L$S$BB$S$D$L$S$D$L
$S$:RETURN
720 PRINT S$S$S$D$L$L$L$S$R$S$D$L$L$L$S$
R$S$D$L$L$L$S$S$S$:RETURN
740 LOCATE 18,21:FOR I=1 TO 11:PRINT BB$
;:NEXT:RETURN
755 LOCATE 4,7:PRINT "GOOD!":RETURN
770 LOCATE 3,8:PRINT "TRY" D$L$L$L$L$ "A
GAIN"
780 FOR I=1000 TO 500 STEP -250:SOUND I,
4:NEXT:FOR TD=1 TO 500:NEXT:RETURN
960 FOR I=1 TO 4:LOCATE X,I:PRINT S$:NEX
T:RETURN
1000 CLS:LOCATE 7,10:PRINT"DO YOU WISH T
O:"
1010 PRINT:PRINT:PRINT C$"1) PRACTICE TI
MES TABLE"
1020 PRINT:PRINT C$"2) RANDOM NUMBERS
1030 PRINT:PRINT:PRINT C$"(ENTER 1 OR 2)
";:INPUT T:IF T<1 OR T>2 THEN PRINT U$U$
U$U$:GOTO 1030
1050 IF T=2 THEN GOTO 190
1060 CLS:PRINT:PRINT:PRINT C$"ENTER TIME
S TABLE"
1070 PRINT:PRINT C$"(1-14)";:INPUT K:IF
K<1 OR K>14 THEN PRINT U$U$U$:GOTO 1070
1090 S=0:R=14:GOTO 263
1100 FOR I= 2 TO 6
1110 READ A :READ B
1120 FOR J= 1 TO B
1130 LOCATE I,J+A :PRINT CHR$(176)
1140 NEXT J:NEXT I:RESTORE:RETURN
1170 LOCATE 7,4:FOR I= 1 TO 11 :PRINT TB
$;:NEXT :RETURN
1230 COLOR 2:LOCATE 5,15:PRINT CHR$(47)U
$BB$BB$D$L$CHR$(249)LB$D$L$LB$D$L$L$L$L$
TB$TB$TB$:COLOR Q*2:RETURN
1240 LOCATE 7,5:PRINT S$ :LOCATE 7,14:PR
INT S$
1250 RETURN
1260 COLOR 2:GOSUB 1240:LOCATE 8,5:PRINT
 TB$TB$:LOCATE 8,14:PRINT TB$TB$:RETURN:
COLOR Q *2
1270 RETURN
1500 FOR I=4 TO 7:LOCATE I,15:FOR J=1 TO
 4:PRINT SP$;:NEXT J,I:RETURN
2225 FOR I= 9 TO 23:LOCATE I,21: FOR J=
1 TO 11 :PRINT SP$;:NEXT J,I:RETURN
2500 COLOR 2:LOCATE 6,17:PRINT CHR$(126)
:RETURN:COLOR Q*2
3000 COLOR Q*2:X=29:IF LEN (F$)>2 THEN 3
030
3015 P=VAL (MID$(F$,2,1))
3020 Y=9+W:GOSUB 475
3025 RETURN
3030 P=VAL(MID$(F$,3,1))
3035 Y=9+W:GOSUB 475
3040 P=VAL(MID$(F$,2,1))
3045 X=X-4:GOSUB 475
3050 RETURN
5000 DATA 6,5,5,7,4,9,3,11,3,11
6000 LOCATE 14,22:PRINT S$D$L$L$S$S$S$D$
L$L$S$;
6002 IF Q=2 THEN PRINT L$SP$U$U$U$L$SP$
6003 RETURN
6004 LOCATE 14,21:PRINT S$D$S$U$S$D$D$L$
L$L$S$R$S$:RETURN
6500 FOR I=500 TO 1000 STEP 250:SOUND I,
4:NEXT:RETURN
```

©

PENTOMINOS
A Puzzle-Solving Program

Jim Butterfield, Associate Editor

Computers can solve puzzles. With the right set of instructions, a program will follow the same logic as humans, trying things to see if they fit. It's interesting to watch the computer working in this way.

This famous puzzle is dealt with at some length in Arthur C. Clarke's novel Imperial Earth. *The characters of the novel don't use a computer to solve the puzzle.*

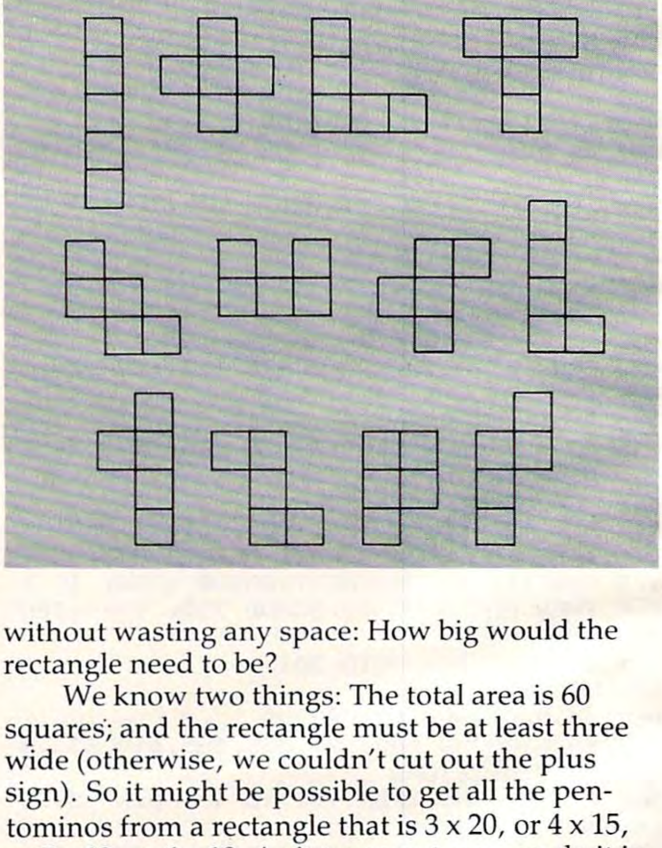
The original program works on all Commodore computers. Additional versions are included here for the Atari, IBM PC and PCjr, TI-99/4A, Radio Shack Color Computer, and Apple.

NOTE: *IBM, TI, Color Computer, and Apple users should insert lines 110–860 from Program 1, the Commodore version, into their programs. The* rem *statements at the ends of these lines should be ignored.*

Pentominos are like dominos, except that they are made up of five elements rather than two. If we put five squares end to end and glued them together, we'd get a long strip, often called the I pentomino. On the other hand, if we took a central square and glued the other four squares to the sides, top, and bottom, we'd get something that looks like a plus sign, which many people call the X pentomino.

Allowing for the differences that are caused by rotating or turning over a piece, there are 12 different pentominos. They are shown in Figure 1; but you might find it fun to try discovering them yourself by drawing them out on a piece of paper. Most of them look a little like letters—you can see a T, an X, and a W among them, for example.

Figure 1: The 12 Pentominos



What's The Puzzle?

The 12 different pentominos, each with an area of 5 squares, give a total of 60 squares. Suppose you had to cut these pentominos out of a rectangle without wasting any space: How big would the rectangle need to be?

We know two things: The total area is 60 squares; and the rectangle must be at least three wide (otherwise, we couldn't cut out the plus sign). So it might be possible to get all the pentominos from a rectangle that is 3 x 20, or 4 x 15, or 5 x 12, or 6 x 10. As it turns out, we can do it in any of these ways.

We can turn the question inside out and put it this way: Can you fit all 12 pentominos into a rectangle of size: 3 x 20, or 4 x 15, or 5 x 12, or 6 x 10?

The Brain Bender

Don't let the following computer program take the fun out of the puzzle for you. Cut the pieces out of cardboard and try your hand at the puzzle.

4 Color
80 COLUMN Letter Quality
PRINTER / PLOTTER

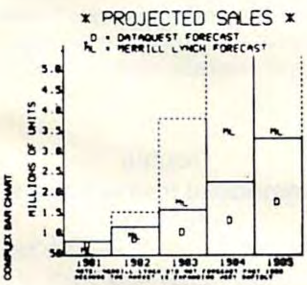
Super Special

$99 $99

1/2 PRICE SALE **1/2 PRICE SALE**

• **LOWEST PRICE IN U.S.A.**

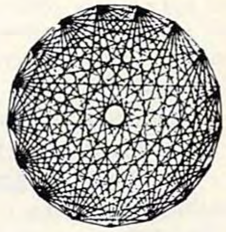
• List your programs • High resolution graphics for bar charts and geometric figures (like spirograph) • Plugs directly into VIC 20 and Commodore 64 — Interface included • Lowest cost letter quality printer in the country.



ACTUAL PRINT SAMPLES

UPPER CASE – ABCDEFGHIJKLMNOPQRSTUVWXYZ

LOWER CASE – abcdefghijklmnopqrstuvwxyz



At last you can list your programs (even control characters) and make beautiful high resolution graphics at an affordable price. This **80 column letter quality printer/plotter** is great for making complex bar charts for business plus fancy greeting cards and geometric designs. Great for homework too. **Everyone must have a 4 color printer plotter for their VIC-20 or Commodore-64.** List $199.00. **Sale $99.00.**

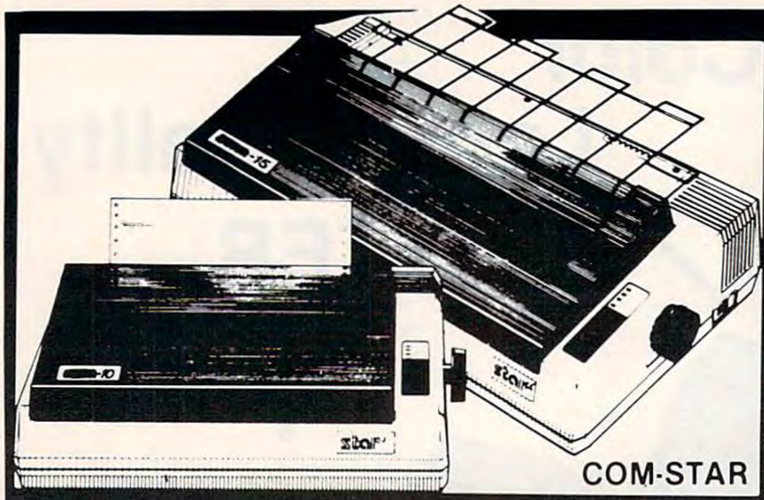
• LOWEST PRICES • 15 DAY FREE TRIAL • 90 DAY FREE REPLACEMENT WARRANTY
• BEST SERVICE IN U.S.A. • ONE DAY EXPRESS MAIL • OVER 500 PROGRAMS • FREE CATALOGS

Add $10.00 for shipping, handling and insurance. Illinois residents please add 6% tax. Add $20.00 for CANADA, PUERTO RICO, HAWAII orders. WE DO NOT EXPORT TO OTHER COUNTRIES.

Enclose Cashiers Check, Money Order or Personal Check. Allow 14 days for delivery, 2 to 7 days for phone orders, 1 day express mail! Canada orders must be in U.S. dollars. Visa - MasterCard - C.O.D.

PROTECTO
ENTERPRIZES (WE LOVE OUR CUSTOMERS)
BOX 550, BARRINGTON, ILLINOIS 60010
Phone 312/382-5244 to order

80 COLUMN PRINTER SALE—$149.00*



COM-STAR T/F

Tractor Friction Printer

only $199**

•15 Day Free Trial -180 Day Immediate Replacement Warranty

- **Lowest Priced, Best Quality, Tractor-Friction Printers in the U.S.A.**
- **Fast 80-120-160 Characters Per Second**
- **40, 46, 66, 80, 96, 132 Characters Per Line Spacing**
- **Word Processing**
- **Print Labels, Letters, Graphs and Tables**
- **List Your Programs**
- **Print Out Data from Modem Services**
- **"The Most Important Accessory for Your Computer"**

*STX-80 COLUMN PRINTER—$149.00

Prints full 80 columns. Super silent operation, 60 CPS, prints Hi-resolution graphics and block graphics, expanded character set, exceptionally clear characters, fantastic print quality, uses inexpensive thermal paper! Best thermal printer in the U.S.A.! (Centronics Parallel Interface).

**DELUXE COMSTAR T/F 80 CPS PRINTER—$199.00

The COMSTAR T/F (Tractor Friction) PRINTER is exceptionally versatile. It prints 8½" x 11" standard size single sheet stationary or continuous feed computer paper. Bi-directional, impact dot matrix, 80 CPS, 224 characters. (Centronics Parallel Interface).

Premium Quality—120 CPS COMSTAR T/F SUPER-10X PRINTER—$289.00

COMSTAR T/F (Tractor Friction) SUPER-10X PRINTER gives you all the features of the COMSTAR T/F PRINTER plus a 10" carriage, 120 CPS, 9 x 9 dot matrix with double strike capability for 18 x 18 dot matrix (near letter quality), high resolution bit image (120 x 144 dot matrix), underlining, back spacing, left and right margin settings, true lower decenders with super and subscripts, prints standard, italic, block graphics and special characters, plus 2K of user definable characters! The COMSTAR T/F SUPER-10X PRINTER was Rated No. 1 by "Popular Science Magazine." It gives you print quality and features found on printers costing twice as much!! (Centronics Parallel Interface) (Better than Epson FX 80).

Premium Quality—120 CPS COMSTAR T/F SUPER-15½" PRINTER—$379.00

COMSTAR T/F SUPER 15½" PRINTER has all the features of the COMSTAR T/F SUPER-10X PRINTER plus a 15½" carriage and more powerful electronics components to handle large ledger business forms! (Better than Epson FX 100).

Superior Quality SUPER HIGH SPEED—160 CPS COMSTAR T/F 10" PRINTER—$489.00

SUPER HIGH SPEED COMSTAR T/F (Tractor Friction) PRINTER has all the features of the COMSTAR SUPER-10X PRINTER plus SUPER HIGH SPEED PRINTING—160 CPS, 100% duty cycle, 8K buffer, diverse character fonts, special symbols and true decenders, vertical and horizontal tabs. RED HOT BUSINESS PRINTER at an unbelievable low price!! (Serial or Centronics Parallel Interface)

Superior Quality SUPER HIGH SPEED—160 CPS COMSTAR T/F 15½" PRINTER—$579.00

SUPER HIGH SPEED COMSTAR T/F 15½" PRINTER has all the features of the SUPER HIGH SPEED COMSTAR T/F 10" PRINTER plus a 15½" carriage and more powerful electronics to handle larger ledger business forms! Exclusive bottom paper feed!!

PARALLEL INTERFACES

For VIC-20 and COM-64—$49.00
For All Apple Computers—$79.00

NOTE: Other printer interfaces are available at computer stores!

Double Immediate Replacement Warranty

We have doubled the normal 90 day warranty to 180 days. Therefore if your printer fails within "180 days" from the date of purchase you simply send your printer to us via United Parcel Service, prepaid. We will IMMEDIATELY send you a replacement printer at no charge, prepaid. This warranty, once again, proves that WE LOVE OUR CUSTOMERS!

Add $17.50 for shipping, handling and insurance. WE DO NOT EXPORT TO OTHER COUNTRIES EXCEPT CANADA.

Enclose Cashiers Check, Money Order or Personal Check. Allow 14 days for delivery, 2 to 7 days for phone orders, 1 day express mail! Canada orders must be in U.S. dollars. VISA — MASTER CARD ACCEPTED. We ship C.O.D.

PROTECTO

ENTERPRIZES (WE LOVE OUR CUSTOMERS)

BOX 550, BARRINGTON, ILLINOIS 60010
Phone 312/382-5244 to order

SUPER-10" ABCDEFGHIJKLMNOPQRSTUVWXYZ
ABCDEFGHIJKLMNOPQRSTUVWXYZ 1234567890

Olympia

COMBINATION "DAISY WHEEL" PRINTER / TYPEWRITER

"WORLD'S FINEST"
LIST $799.00
SALE $489.00

- SUPERB COMPUTER PRINTER COMBINED WITH WORLD'S FINEST ELECTRONIC TYPEWRITER!
- BETTER THAN IBM SELECTRIC — USED BY WORLD'S LARGEST CORPORATIONS!
- TWO MACHINES IN ONE — JUST A FLICK OF THE SWITCH!
- SUPERB EXECUTIVE CORRESPONDENCE — HOME, OFFICE, WORD PROCESSING!
- EXTRA LARGE CARRIAGE — ALLOWS 14-1/8" PAPER USAGE!
- DROP IN CASSETTE RIBBON — EXPRESS LIFT OFF CORRECTION OR ERASER UP TO 46 CHARACTERS!
- PRECISION DAISY WHEEL PRINTING — MANY TYPE STYLES!
- PITCH SELECTOR — 10, 12, 15 CPS, AUTOMATIC RELOCATE KEY!
- AUTOMATIC MARGIN CONTROL AND SETTING! KEY IN BUFFER!
- ELECTRONIC RELIABILITY, BUILT IN DIAGNOSTIC TEST!
- CENTRONICS PARALLEL INTERFACE BUILT-IN (SERIAL OPTIONAL)!
- 15 DAY FREE TRIAL — 90 DAY FREE REPLACEMENT WARRANTY!

Add $17.50 for shipping and handling!!

Enclose Cashiers Check, Money Order or Personal Check. Allow 14 days for delivery, 2 to 7 days for phone orders, 1 day express mail! Canada orders must be in U.S. dollars. VISA — MASTER CARD ACCEPTED. We ship C.O.D.

PROTECTO
ENTERPRIZES (WE LOVE OUR CUSTOMERS)

BOX 550, BARRINGTON, ILLINOIS 60010
Phone 312/382-5244 to order

ONE CALL GETS IT ALL!

TELECOMMUNICATIONS SALE! $49

Make a connection!

for Commodore VIC-20 and 64

REACH OUT AND ACCESS SOMEONE

Wander the wires of bulletin board systems gathering information and making new computer friends!

DELUXE 40 CHARACTER MODEM SALE **$49.00** *(List $99.00)*

- Saves on-line time; Easier to read!
- Direct connect, originate/answer, half and full duplex!
- Terminal program (tape included) Disk—Add $5.00!
- One year free access to Compuserve, 2 hours free on-line time!
- 15 day free trial — 180 day free replacement warranty!
- Instruction manual!

80 COLUMN TERMINAL CARTRIDGE SALE **$59.00**

- No cassette or disk drive needed!
- VIC-20 40-80 Column . . . . $59.00 (List $99.00)
- COM-64 80 Column Board . . . . $99.00 (List $199.00)

Converts your computer screen to 40 or 80 columns! You can add a 40-80 column word processor, mailmerge and electronic spreadsheet for only $24.95 (tape or disk).

SMART 64 MODEM PROGRAM SALE **$26.95** *(List $39.95)*

(DISK)

- One key password automatic entry!
- On-line alarm timer!
- Prints out all information received!
- Record and send programs on disk!
- Use with Protecto 40-80 column terminal!

• LOWEST PRICES • 15 DAY FREE TRIAL • 90 DAY FREE REPLACEMENT WARRANTY
• BEST SERVICE IN U.S.A. • ONE DAY EXPRESS MAIL • OVER 500 PROGRAMS • FREE CATALOGS

Add $3.00 for postage. Add $6.00 for CANADA, PUERTO RICO, HAWAII orders. WE DO NOT EXPORT TO OTHER COUNTRIES.

Enclose Cashiers Check, Money Order or Personal Check. Allow 14 days for delivery, 2 to 7 days for phone orders, 1 day express mail! Canada orders must be in U.S. dollars. We accept Visa and MasterCard. We ship C.O.D.

PROTECTO
ENTERPRIZES (WE LOVE OUR CUSTOMERS)
BOX 550, BARRINGTON, ILLINOIS 60010
Phone 312/382-5244 to order

SANYO MONITOR SALE!!

9″ Data Monitor

- 80 Columns × 24 lines
- Green text display
- Easy to read - no eye strain
- Up front brightness control
- High resolution graphics
- Quick start - no preheating
- Regulated power supply
- Attractive metal cabinet
- UL and FCC approved

- ***15 Day Free Trial - 90 Day Immediate Replacement Warranty***

9″ Screen - Green Text Display	*$ 69.00
12″ Screen - Green Text Display (anti-reflective screen)	*$ 99.00
12″ Screen - Amber Text Display (anti-reflective screen)	*$ 99.00
14″ Screen - Color Monitor (national brand)	*$249.00

*PLUS $9.95 for Connecting Cable.

Display Monitors From Sanyo

With the need for computing power growing every day, Sanyo has stepped in to meet the demand with a whole new line of low cost, high quality data monitors. Designed for commercial and personal computer use. All models come with an array of features, including upfront brightness and contrast controls. The capacity 5 × 7 dot characters as the input is 24 lines of characters with up to 80 characters per line.

Equally important, all are built with Sanyo's commitment to technological excellence. In the world of Audio/Video, Sanyo is synonymous with reliability and performance. And Sanyo quality is reflected in our reputation. Unlike some suppliers, Sanyo designs, manufactures and tests virtually all the parts that go into our products, from cameras to stereos. That's an assurance not everybody can give you!

SANYO

Official Video Products of the Los Angeles 1984 Olympics

• LOWEST PRICES • 15 DAY FREE TRIAL • 90 DAY FREE REPLACEMENT WARRANTY
• BEST SERVICE IN U.S.A. • ONE DAY EXPRESS MAIL • OVER 500 PROGRAMS • FREE CATALOGS

Add $10.00 for shipping, handling and insurance. Illinois residents please add 6% tax. Add $20.00 for CANADA, PUERTO RICO, HAWAII orders. WE DO NOT EXPORT TO OTHER COUNTRIES.

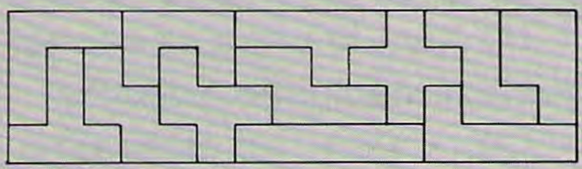
Enclose Cashiers Check, Money Order or Personal Check. Allow 14 days for delivery, 2 to 7 days for phone orders, 1 day express mail! Canada orders must be in U.S. dollars. Visa - MasterCard - C.O.D.

PROTECTO
ENTERPRIZES (WE LOVE OUR CUSTOMERS)
BOX 550, BARRINGTON, ILLINOIS 60010
Phone 312/382-5244 to order

It's an interesting way to wile away the hours. 6 x 10 and 5 x 12 are not too hard; 4 x 15 will make you work; and 3 x 20, which seems at first to be the easiest, proves to be a real brain bender.

A sample solution to the 4 x 15 problem is given in Figure 2.

Figure 2: A 4 x 15 Solution



If humans can waste time trying to fit the pieces, computers can do it too. "Pentominos" does not run at blinding speed; it tries the pieces at about the same speed as humans do. It's dumber than human puzzle solvers: It will try to make a piece fit in places we know instinctively are hopeless. But the computer has no intuition: It will plod along, making dumb moves until it finds a combination that fits.

The program tries the pieces "visibly"—that is, you can see it putting the pieces in place, thinking about its next move, and then taking a piece back out when it becomes obvious (even to the dumb computer) that it can't work there.

In a moment we'll get to more detail on how it works. The computer always thinks about fitting the upper-leftmost empty square, and it will tell you which piece it is trying to fit there; that piece's identity will be shown in a corner of the screen. So you can track the computer's thoughts if you wish.

It can take a few minutes or several hours to find the next solution. This program is a good one to set up for an overnight run. You might want to turn off your TV set or monitor and let the computer hum away quietly all by itself.

When a solution is found, you can type CONT at any blank place on the screen, and the computer will go after the next solution.

How It Works

The pentominos and all their possible rotations are stored in DATA statements. Only four squares need to be described for each pentomino rotation, since the information gives coordinates based upon the starting square.

After reading in the data, the computer uses the following logic. Line numbers are given for those who would like to try examining the program.

1. (Line 2010) The computer looks through the list of pieces to find the first one that isn't being used. Then it searches the board for a blank square, starting at the left and searching each column top to bottom. That's the next place it will try to fit a piece. If it can't find a blank, we have a solution and will go to step 5.

2. (Line 2030) The piece just picked is set to its first rotation.

3. (Line 2060) The computer tries to fit the piece starting at the square it has identified. If it doesn't fit, it will skip ahead to step 7.

4. (Line 2120) The piece fits, so the computer puts it onto the board, onto the screen, and marks off the piece as used. It then goes back to step 1 to look for a new place to fit pieces.

5. (Line 2170) We have a solution! Stop and wait for the user to admire us. If the user types CONT, we'll keep going into step 6.

6. (Line 2190) We've reached a dead end, so we go back and remove the last piece placed on the board. If there are no pieces left, we quit; at this point we will have found all the solutions.

7. (Line 2260) Let's rotate the current piece so that we can try it in a different way. If we can find a new rotation, we go back to step 3 to try the piece. If not, we continue to step 8.

8. (Line 2300) The computer looks through the list of pieces to find the next piece to be tried. Then it goes back to step 2.

Variables And Arrays

If you're trying to read the program, it will be worthwhile to have some information on variables and arrays. Here are some useful ones:

Array B(X,Y) is the board. If the value is zero, that part of the board is blank. When a board square is used, the appropriate value in this array is set to the number of the occupying piece; but the important thing to remember is that it's set to nonzero.

The DATA statements show all rotations of all pieces. They are transferred to arrays X and Y:

Arrays X(rotation,C) and Y(rotation,C) tell where to find the squares (X and Y) of each piece's rotation. The rotation is taken from the DATA statements.

Array P(rotation) tells which piece is involved for each rotation of the above table.

Each Piece Has Data

Array P$(piece) is the name of the piece.

Array S(piece) tells where to find the starting rotation for piece X.

Array T(piece) tells which rotation is currently being used (or tried) for piece X.

Arrays X2(piece) and X2(piece) list the starting square where piece A has been placed.

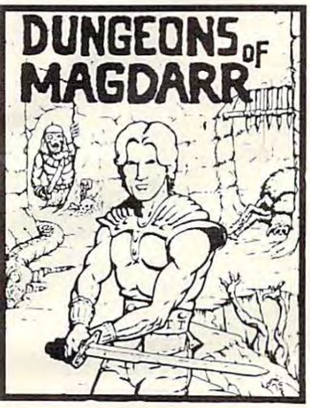
Tracking The Moves

Array U(move) lists the pieces in the order in which we tried them.

The piece under consideration is designated

AARDVARK LTD.

VIDEO ADVENTURES™



DUNGEONS OF MAGDARR - Serious D of D for up to 8 players. Features full 3d GRAPHICS! You get a choice of several characters that grow from game to game and are interchangeable with characters from our famous Dungeons of Death game. A real dungeon with level after level of monsters to conquer and treasures to find - all in hi-res 3d graphics.

Available On: TRS80C, IBM PC, CMD64

TAPE $19.95 DISK $24.95

BAG-IT-MAN - The ultimate arcade game for TRS80C or MCD64. This one has three screens full of BAGS OF GOLD, CARTS & ELEVATORS TO RIDE IN, MINE SHAFTS, and TWO NASTY GUARDS. Great sound and color and continuous excitement.

Available On: TRS80C 32K, CMD64

TAPE $19.95 DISK $24.95

QUEST - A different kind of Graphic Adventure, it is played on a computer generated mape of Alesia. You'll have to build an army and feed them through combat, bargaining, exploration of ruins and temples, and outright banditry! Takes 2 - 5 hours to play and is different each time.

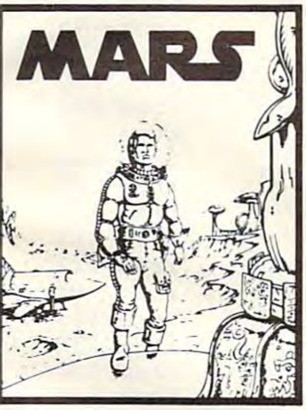
Available On: TRS80C 16K, CMD64, VIC20 13K, MC10 16K, TI99 (EXT. BASIC), IBMPC

TAPE $14.95 DISK $19.95

STARFIRE - If you enjoyed StarRaiders or StarWars, you will love Starfire. It is not a copy, but the best shoot-em-up, see them in the window space game on the CMD64 or TRS80C. The fantastic graphics will put you right in the control room as you hyperspace from quadrant to quadrant fighting the aliens and protecting your bases.

Available On: TRS80C 16K, CMD64

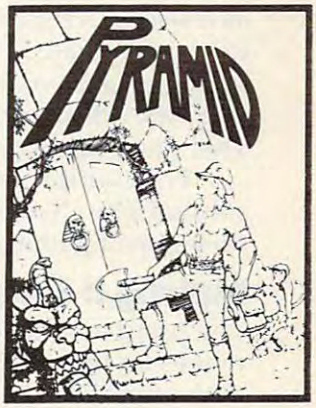
TAPE $19.95 DISK $24.95



MARS - Your ship crashed on the Red Plane and you have to get home. You will have to explore a Martian City, repair your ship, and deal with possibly hostile aliens to get home again. This is recommended as a first Adventure. It is in no way simple - playing time normally runs from 30 to 50 hours, but it lets you try out Adventuring before you battle the really tough ones. Full Graphics Adventure.

Available On: TRS80C, CMD 64, IBM PC

TAPE $19.95 DISK $24.95



PYRAMID - ONE OF THE TOUGHEST ADVENTURES. Average time through the pyramid is 50 - 70 hours. Clues are everywhere and some ingenious problems make this popular around the world. FULL GRAPHIC ADVENTURE.

Available On: TRS80C 16K, CMD64, MC10 16K, IBM PC

TAPE $19.95 DISK $24.95

NEW! GRAPHIC ADVENTURES

AARDVARK offers over 120 original high quality programs. Send one dollar for a current catalog and receive a $1.00 gift certificate good towards your next purchase.

Authors - **AARDVARK** pays top dollar for high quality programs. Send a copy today for a personal review and editorial help.

TO ORDER: Send amount indicated plus $2.00 shipping, per order. Include quantity desired and your preference of tape or disk. Be sure to indicate type of system and amount of memory. When using charge card to order by mail, be sure to include expiration date.

MasterCard CHARGE CARDS WELCOME VISA

1-313-669-3110

PHONE ORDERS ACCEPTED
8:00 a.m. to 8:00 p.m. E.S.T., MON-FRI

AARDVARK *Action Software*

2352 S. COMMERCE • WALLED LAKE, MI 48088 • (313) 669-3110

CMD64 / TRS80C / IBMPC / VIC20 / TI99

Figure 3: Partial Solution. The Program Will Be Trying To Fit The Point Marked X.

by P; its current rotation, of course, will be T(P).

When we place a piece, we log it into array U and use P1 to keep track of how many pieces have been used.

Program Variations

The program could be speeded up significantly by using a compiler or by converting it to machine language. I have chosen not to do that for two reasons: compatibility and readability.

A machine language version would nevertheless be quite straightforward to write. No special math or other logic is involved. Such a program would be very fast. But it would not be universal, since different machines would need to load the program into different memory locations.

If you go for many solutions, you should realize that some of the solutions are transformations of others. Given one solution, others can be found by inverting it left to right or top to bottom. This means that each solution is really four solutions; but the computer will find each of the four as it works. If this is not desired, the extra solutions can be eliminated by removing all but two of the rotations of a single eight-rotation piece. That way, the reflected solutions couldn't happen: That piece can appear in only one orientation.

For example, we could eliminate reflected solutions by changing line 770 to DATA R,2 and then deleting lines 800 to 850 inclusive.

Making It Smarter

The program would run faster if it didn't show its moves on the screen, but watching it work is most of the fun. For one thing, it may remind you of an important aspect of computers: They're dumb, but they're faithful.

The computer will lumber along, trying dumb moves. But it won't get tired, and it will eventually reach the solution.

Yes, we could add extra logic to make the computer smarter. We could ask the computer to scan for some of the obviously impossible situations that it does not recognize at all with the present program. But there's a danger: The computer could waste more time being smart than it does being dumb.

Program 1: Pentominos For Commodore

Refer to the "Automatic Proofreader" article before typing this program in.

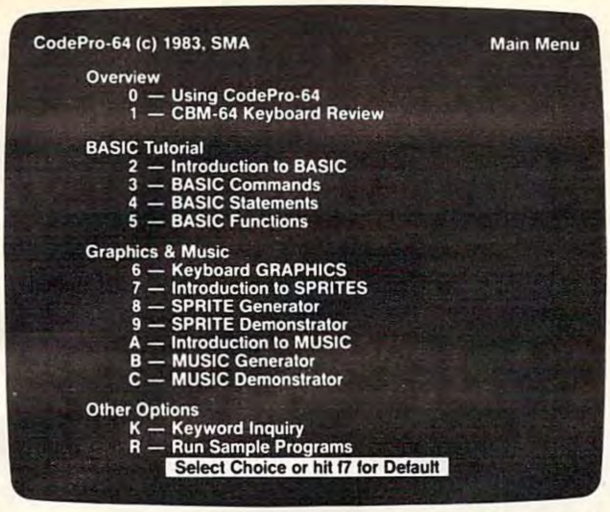
```
100 PRINT CHR$(142)"{CLR}{5 RIGHT}PENTOMI
    NOS{DOWN}"                      :rem 140
110 DATA I,2                        :rem 83
120 DATA Ø,1,Ø,2,Ø,3,Ø,4            :rem 107
130 DATA 1,Ø,2,Ø,3,Ø,4,Ø            :rem 108
140 DATA X,1                        :rem 100
150 DATA 1,-1,1,Ø,2,Ø,1,1           :rem 152
160 DATA V,4                        :rem 103
170 DATA Ø,1,Ø,2,1,Ø,2,Ø            :rem 108
180 DATA Ø,1,Ø,2,1,2,2,2            :rem 113
190 DATA 1,Ø,2,Ø,2,1,2,2            :rem 114
200 DATA 1,Ø,2,Ø,2,-1,2,-2          :rem 196
210 DATA T,4                        :rem 97
220 DATA Ø,1,Ø,2,1,1,2,1            :rem 106
230 DATA 1,Ø,1,1,2,Ø,1,2            :rem 107
240 DATA 1,Ø,2,Ø,1,-1,1,-2          :rem 198
250 DATA 2,-1,2,Ø,2,1,1,Ø           :rem 155
260 DATA W,4                        :rem 105
270 DATA Ø,1,1,1,1,2,2,2            :rem 113
280 DATA 1,Ø,1,1,2,1,2,2            :rem 114
290 DATA Ø,1,1,-1,1,Ø,2,-1          :rem 202
300 DATA 1,-1,1,Ø,2,-2,2,-1         :rem 242
310 DATA U,4                        :rem 99
320 DATA Ø,2,1,Ø,1,1,1,2            :rem 107
330 DATA 2,Ø,Ø,1,1,1,2,1            :rem 108
340 DATA Ø,1,1,Ø,2,Ø,2,1            :rem 108
350 DATA 1,Ø,Ø,1,Ø,2,1,2            :rem 109
360 DATA F,8                        :rem 93
370 DATA Ø,1,1,-1,1,Ø,2,Ø           :rem 155
380 DATA 1,-1,2,-1,1,Ø,1,1          :rem 203
390 DATA 1,-1,1,Ø,1,1,2,1           :rem 159
400 DATA 1,-1,1,Ø,2,Ø,2,1           :rem 151
410 DATA Ø,1,1,1,1,2,2,1            :rem 108
420 DATA 1,Ø,1,1,2,1,1,2            :rem 109
430 DATA 1,Ø,1,1,2,-1,2,Ø           :rem 154
440 DATA 1,-2,1,-1,2,-1,1,Ø         :rem 246
450 DATA L,8                        :rem 99
460 DATA 1,Ø,2,Ø,3,Ø,3,1            :rem 114
470 DATA Ø,1,Ø,2,Ø,3,1,3            :rem 115
480 DATA 1,-3,1,-2,1,-1,1,Ø         :rem 251
490 DATA 1,Ø,2,Ø,3,Ø,3,-1           :rem 162
500 DATA 1,Ø,2,Ø,3,Ø,Ø,1            :rem 106
510 DATA Ø,1,Ø,2,Ø,3,1,Ø            :rem 107
520 DATA Ø,1,1,1,2,1,3,1            :rem 111
530 DATA 1,Ø,1,1,1,2,1,3            :rem 112
540 DATA Y,8                        :rem 112
550 DATA Ø,1,Ø,2,Ø,3,1,1            :rem 112
560 DATA 1,Ø,2,Ø,3,Ø,1,1            :rem 113
570 DATA 1,-1,1,Ø,1,1,1,2           :rem 159
580 DATA 1,-1,1,Ø,2,Ø,3,Ø           :rem 160
590 DATA Ø,1,Ø,2,Ø,3,1,2            :rem 117
600 DATA 1,Ø,2,Ø,3,Ø,2,1            :rem 109
610 DATA 1,-2,1,-1,1,Ø,1,1          :rem 199
620 DATA 1,Ø,2,Ø,3,Ø,2,-1           :rem 156
630 DATA Z,4                        :rem 109
640 DATA Ø,1,1,1,2,1,2,2            :rem 114
650 DATA 1,Ø,1,1,1,2,2,2            :rem 115
660 DATA 1,-2,1,-1,1,Ø,2,-2         :rem 251
670 DATA 2,-1,1,Ø,2,Ø,Ø,1           :rem 159
680 DATA P,8                        :rem 108
690 DATA Ø,1,1,Ø,1,1,2,Ø            :rem 115
700 DATA 1,Ø,Ø,1,1,1,Ø,2            :rem 107
710 DATA Ø,1,1,Ø,1,1,1,2            :rem 109
720 DATA 1,Ø,Ø,1,1,1,2,1            :rem 110
730 DATA 1,-1,1,Ø,2,-1,2,Ø          :rem 202
740 DATA 1,-1,1,Ø,Ø,1,1,1           :rem 156
750 DATA Ø,1,Ø,2,1,1,1,2            :rem 114
```

```
760 DATA 1,0,2,0,1,1,2,1              :rem 115
770 DATA R,8                          :rem 110
780 DATA 0,1,0,2,1,2,1,3              :rem 119
790 DATA 1,0,2,0,2,1,3,1              :rem 120
800 DATA 1,-1,1,0,2,-1,3,-1           :rem 247
810 DATA 1,-1,1,0,0,1,0,2             :rem 154
820 DATA 0,1,1,1,1,2,1,3              :rem 114
830 DATA 1,0,1,1,2,1,3,1              :rem 115
840 DATA 1,0,2,-1,2,0,3,-1            :rem 206
850 DATA 1,-2,1,-1,1,0,0,1            :rem 204
860 DATA A,0                          :rem 85
870 V$="{HOME}{13 DOWN}"              :rem 138
880 H$="{23 RIGHT}"                   :rem 40
1000 DIM X(63,4),Y(63,4),P(64),P$(13),S(1
     3),T(13),B(6,20)                 :rem 36
1001 DIM X1(5),Y1(5),X2(12),Y2(12),U(12)
                                      :rem 241
1010 READ P$,N:IF N=0 GOTO 1070       :rem 81
1020 T=T+1:P$(T)=P$:S(T)=V+1          :rem 41
1030 FOR J=V+1 TO V+N:P(J)=T          :rem 12
1040 FOR K=0 TO 3:READ X(J,K),Y(J,K):NEXT
      K,J                             :rem 203
1050 V=V+N:PRINT P$;                  :rem 158
1060 GOTO 1010                        :rem 194
1070 PRINTLEFT$(V$,5);:PRINT"CHOOSE:
     {DOWN}"                          :rem 34
1080 FOR J=3 TO 6:PRINT J;"BY";60/J;"
     {DOWN}":NEXT J                   :rem 219
1090 INPUT "SELECT 3 THRU 6";W1       :rem 205
1100 IF W1<3 OR W1>6 OR W1<>INT(W1) GOTO
     {SPACE}1070                      :rem 77
1110 W2=60/W1                         :rem 166
1120 PRINT "{CLR}"                    :rem 40
2000 REM FIND NEW SPACE TO FILL       :rem 231
2010 GOSUB 3000:P=J:GOSUB 3200:IF X1>W2 G
     OTO 2170                         :rem 178
2020 REM GET A NEW PIECE              :rem 25
2030 T(P)=S(P)                        :rem 235
2040 PRINT "{HOME}";P$(P);"{11 DOWN}"
                                      :rem 52
2050 REM TRY FITTING PIECE            :rem 37
2060 C$=P$(P):X1(0)=X1:Y1(0)=Y1:FOR J=1 T
     O 4                              :rem 71
2070 X=X(T(P),J-1)+X1:Y=Y(T(P),J-1)+Y1:X1
     (J)=X:Y1(J)=Y                    :rem 100
2080 IF X<1 OR Y<1 OR X>W2 OR Y>W1 GOTO 2
     260                              :rem 8
2090 IF B(Y,X)<>0 GOTO 2260           :rem 119
2100 NEXT J                           :rem 76
2110 REM IT FITS - PUT PIECE IN PLACE
                                      :rem 3
2120 B=P:FOR J=0 TO 4                 :rem 67
2130 X=X1(J):Y=Y1(J):GOSUB 3500       :rem 246
2140 NEXT J                           :rem 80
2150 X2(P)=X1:Y2(P)=Y1:P1=P1+1:U(P1)=P:GO
     TO 2010                          :rem 223
2160 REM BOARD FILLED                 :rem 197
2170 PRINT "{HOME}{2 SPACES}SOLUTION";:EN
     D                                :rem 119
2180 REM UNDRAW LAST ONE              :rem 150
2190 P=U(P1):U(P1)=0:P1=P1-1:IF P1<0 THEN
      PRINT"THAT'S ALL":END           :rem 112
2200 B=0:X=X2(P):Y=Y2(P):C$=" ":GOSUB 350
     0                                :rem 13
2210 X1=X:Y1=Y:FOR J=1 TO 4           :rem 237
2220 X=X(T(P),J-1)+X1:Y=Y(T(P),J-1)+Y1:X1
     (J)=X:Y1(J)=Y                    :rem 97
2230 GOSUB 3500                       :rem 15
2240 NEXT J                           :rem 81
2250 REM ROTATE THE PIECE             :rem 195
2260 T(P)=T(P)+1:IF P(T(P))=P GOTO 2060
                                      :rem 58
2270 REM GIVE UP ON PIECE             :rem 130
2280 T(P)=0                           :rem 46
2290 REM LOOK FOR NEW PIECE           :rem 29
2300 P=P+1:IF P>12 GOTO 2190          :rem 189
2310 IF T(P)<>0 GOTO 2300             :rem 242
2320 GOTO 2030                        :rem 197
3000 FOR J=1 TO 12:IF T(J)<>0 THEN NEXT J
                                      :rem 130
3010 RETURN                           :rem 164
3200 FOR X1=1 TO W2:FOR Y1=1 TO W1:rem 19
3210 IF B(Y1,X1)=0 GOTO 3230          :rem 149
3220 NEXT Y1,X1                       :rem 69
3230 RETURN                           :rem 168
3500 PRINT LEFT$(V$,Y+2);LEFT$(H$,X);C$:B
     (Y,X)=B                          :rem 231
3510 RETURN                           :rem 169
```

Program 2: Pentominos For Atari

Refer to the "Automatic Proofreader" article before typing this program in.

```
FE 100 PRINT "{CLEAR}PLEASE WAIT...   I
       NITIALIZING ARRAYS":POKE 752,1:
       POSITION 0,0
FD 110 DATA I,2
GL 120 DATA 0,1,0,2,0,3,0,4
GM 130 DATA 1,0,2,0,3,0,4,0
GE 140 DATA X,1
JI 150 DATA 1,-1,1,0,2,0,1,1
GH 160 DATA V,4
GM 170 DATA 0,1,0,2,1,0,2,0
HB 180 DATA 0,1,0,2,1,2,2,2
HC 190 DATA 1,0,2,0,2,1,2,2
ME 200 DATA 1,0,2,0,2,-1,2,-2
GB 210 DATA T,4
GK 220 DATA 0,1,0,2,1,1,2,1
GL 230 DATA 1,0,1,1,2,0,1,2
MG 240 DATA 1,0,2,0,1,-1,1,-2
JL 250 DATA 2,-1,2,0,2,1,1,0
GJ 260 DATA W,4
HB 270 DATA 0,1,1,1,1,2,2,2
HC 280 DATA 1,0,1,1,2,1,2,2
MK 290 DATA 0,1,1,-1,1,0,2,-1
PC 300 DATA 1,-1,1,0,2,-2,2,-1
GD 310 DATA U,4
GL 320 DATA 0,2,1,0,1,1,1,2
GM 330 DATA 2,0,0,1,1,1,2,1
GM 340 DATA 0,1,1,0,2,0,2,1
GN 350 DATA 1,0,0,1,0,2,1,2
FN 360 DATA F,8
JL 370 DATA 0,1,1,-1,1,0,2,0
ML 380 DATA 1,-1,2,-1,1,0,1,1
JP 390 DATA 1,-1,1,0,1,1,2,1
JH 400 DATA 1,-1,1,0,2,0,2,1
GM 410 DATA 0,1,1,1,1,2,2,1
GN 420 DATA 1,0,1,1,2,1,1,2
JK 430 DATA 1,0,1,1,2,-1,2,0
PG 440 DATA 1,-2,1,-1,2,-1,1,0
GD 450 DATA L,8
HC 460 DATA 1,0,2,0,3,0,3,1
HD 470 DATA 0,1,0,2,0,3,1,3
PL 480 DATA 1,-3,1,-2,1,-1,1,0
KC 490 DATA 1,0,2,0,3,0,3,-1
GK 500 DATA 1,0,2,0,3,0,0,1
GL 510 DATA 0,1,0,2,0,3,1,0
GP 520 DATA 0,1,1,1,2,1,3,1
HA 530 DATA 1,0,1,1,1,2,1,3
HA 540 DATA Y,8
HA 550 DATA 0,1,0,2,0,3,1,1
HB 560 DATA 1,0,2,0,3,0,1,1
JP 570 DATA 1,-1,1,0,1,1,1,2
KA 580 DATA 1,-1,1,0,2,0,3,0
HF 590 DATA 0,1,0,2,0,3,1,2
```



A new concept in interactive visual learning.

CodePro-64™

Now you can learn to code in BASIC and develop advanced programming skills with graphics, sprites and music—*visually*. You learn by interacting with CodePro-64, a new concept in interactive visual learning.

SEE PROGRAM EXECUTION

Imagine actually seeing BASIC statements execute. CodePro-64 guides you through structured examples of BASIC program segments. You enter the requested data or let CodePro-64 do the typing for you. (It will not let you make a mistake.)

You step through and actually see the execution of sample program statements by simply pressing the space bar. CodePro-64 does the rest. You see statements with corresponding *graphics* and variable value displays.

EXTENSIVE TUTORIAL

CodePro-64's extensive tutorial guides you through each BASIC command, program statement, and function. You get clear explanations. Where appropriate, you invoke BasicView to see examples execute and watch their flow charts and variables change.

By seeing graphic displays of program segment execution you learn by visual example. You *learn faster and grasp programming concepts easier* with CodePro-64 because you immediately see the results of your input.

You control your learning. You can go through the tutorial sequentially, or return to the main menu and select different topics, or *use keywords* to select language elements to study. You can page back and forth between screens within a topic at the touch of a function key.

Once you have practiced and mastered the BASIC language elements you move on to more advanced concepts. You learn about sprite and music programming.

SPRITE GENERATOR & DEMONSTRATOR

CodePro-64's sprite generator lets you *define your own sprites* on the screen. You learn how to define sprites and what data values correspond to your sprite definitions. (You can then save your sprite data to a diskette file for use in your own programs.) You can *easily experiment* with different definitions and make changes to immediately see the effects.

We also help you learn to program with sprites by giving you a *sprite demonstrator* so you can see the effect of changing register values. You can experiment by moving your sprite around in a screen segment, change its color and see the effects of your changes. You learn by visual examples.

MUSIC GENERATOR & DEMONSTRATOR

Our *Music Generator* and *Music Demonstrator* will provide hours of instruction and creative enjoyment. From the beginning of your instruction you can *compose simple tunes* on the screen using the generator. Once you've completed a composition you can *save the tune* and its associated SID parameters to a diskette file. Our music sample program can be used alone or incorporated into your own programs to *read the saved music file* and replay your songs.

Our music demonstrator *lets you experiment* with various combinations of music programming parameters and hear the results. All you do is enter rows of SID parameters on the screen to create a particular sound. Then you *hear each sound by playing the "keyboard organ"* in real time as you shift from row to row of SID parameters. By seeing your input and hearing the result you quickly learn how to create new musical sounds and special sound effects.

Whether you're a beginning programmer or an experienced professional, CodePro-64 will help you improve you Commodore 64 programming skills. We're sure because CodePro-64 was developed by a team of two professionals with *over 25 years* of software development experience.

CodePro-64 is a professional quality educational program for the serious student of personal computing. And it's *fully guaranteed*. Order yours today.

OUR GUARANTEE

We guarantee your satisfaction. You must be satisfied with CodePro-64 for the Commodore-64. Try it for 10 days and if for any reason you are not satisfied return it to us (undamaged) for a full refund. *No risk.*

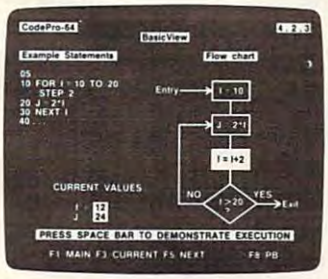
HOW TO ORDER

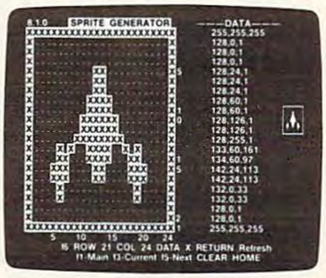
Order your copy of CodePro-64 today by mail or phone. Send only $59.95 plus $3.00 shipping and handling to:

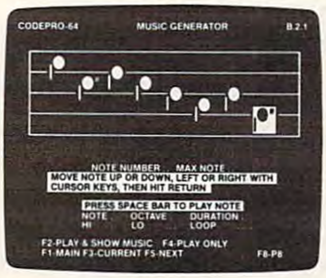
SYSTEMS MANAGEMENT ASSOCIATES
3700 Computer Drive, Dept. G-1
Raleigh, N.C. 27609

Available on diskette only. MasterCard/-VISA accepted. For faster service on credit card orders, call toll free 1-800 SMA-RUSH. (1-800-762-7874). *Dealer inquiries invited.*

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







SYSTEMS MANAGEMENT ASSOCIATES

```
GN 6ØØ  DATA 1,Ø,2,Ø,3,Ø,2,1
MH 61Ø  DATA 1,-2,1,-1,1,Ø,1,1
JM 62Ø  DATA 1,Ø,2,Ø,3,Ø,2,-1
GN 63Ø  DATA Z,4
HC 64Ø  DATA Ø,1,1,1,2,1,2,2
HD 65Ø  DATA 1,Ø,1,1,1,2,2,2
FL 66Ø  DATA 1,-2,1,-1,1,Ø,2,-2
JP 67Ø  DATA 2,-1,1,Ø,2,Ø,Ø,1
GM 68Ø  DATA F,8
HD 69Ø  DATA Ø,1,1,Ø,1,1,2,Ø
GL 7ØØ  DATA 1,Ø,Ø,1,1,1,Ø,2
GN 71Ø  DATA Ø,1,1,Ø,1,1,1,2
GO 72Ø  DATA 1,Ø,Ø,1,1,1,2,1
MK 73Ø  DATA 1,-1,1,Ø,2,-1,2,Ø
JM 74Ø  DATA 1,-1,1,Ø,Ø,1,1,1
HC 75Ø  DATA Ø,1,Ø,2,1,1,1,2
HD 76Ø  DATA 1,Ø,2,Ø,1,1,2,1
GO 77Ø  DATA R,8
HH 78Ø  DATA Ø,1,Ø,2,1,2,1,3
HI 79Ø  DATA 1,Ø,2,Ø,2,1,3,1
PH 8ØØ  DATA 1,-1,1,Ø,2,-1,3,-1
JK 81Ø  DATA 1,-1,1,Ø,Ø,1,Ø,2
HC 82Ø  DATA Ø,1,1,1,1,2,1,3
HD 83Ø  DATA 1,Ø,1,1,2,1,3,1
MO 84Ø  DATA 1,Ø,2,-1,2,Ø,3,-1
MM 85Ø  DATA 1,-2,1,-1,1,Ø,Ø,1
FF 86Ø  DATA A,Ø
HE 1ØØØ DIM X(63,4),Y(63,4),P(64),PP$(
        13),S(13),T(13),B(6,20)
CI 1ØØ1 DIM X1(5),Y1(5),X2(12),Y2(12),
        U(12),C$(1),P$(1)
MP 1ØØ2 Z=Ø:FOR I=Ø TO 63:P(I)=Z:FOR J
        =Ø TO 4:X(I,J)=Z:Y(I,J)=Z:NEXT
         J:NEXT I
DA 1ØØ3 P(64)=Z:FOR I=Ø TO 12:S(I)=Z:T
        (I)=Z:X2(I)=Z:Y2(I)=Z:U(I)=Z:N
        EXT I:S(13)=Z:T(13)=Z
BK 1ØØ4 FOR I=Ø TO 6:FOR J=Ø TO 2Ø:B(I
        ,J)=Z:NEXT J:NEXT I:FOR I=Ø TO
         5:X1(I)=Z:Y1(I)=Z:NEXT I
GD 1ØØ5 PRINT "{CLEAR}":POSITION 15,Ø:
        PRINT "PENTOMINOS":PRINT
EH 1Ø1Ø READ P$,N:IF N=Ø THEN 1Ø7Ø
PJ 1Ø2Ø T=T+1:PP$(T,T)=P$:S(T)=V+1
AM 1Ø3Ø FOR J=V+1 TO V+N:P(J)=T
DI 1Ø4Ø FOR K=Ø TO 3:READ L,M:X(J,K)=L
        :Y(J,K)=M:NEXT K:NEXT J
JO 1Ø5Ø V=V+N:PRINT P$;
MC 1Ø6Ø GOTO 1Ø1Ø
JM 1Ø7Ø POSITION 1,5:PRINT "CHOOSE:":P
        RINT
EL 1Ø8Ø FOR J=3 TO 6:PRINT J;" BY ";6Ø
        /J:NEXT J
JF 1Ø9Ø PRINT :PRINT "SELECT 3 THRU 6:
         ";:INPUT W1
HM 11ØØ IF W1<3 OR W1>6 OR W1<>INT(W1)
         THEN GOTO 1Ø7Ø
KG 111Ø W2=6Ø/W1
BC 112Ø PRINT "{CLEAR}"
OH 2ØØØ REM FIND NEW SPACE TO FILL
OB 2Ø1Ø GOSUB 3ØØØ:P=J:GOSUB 32ØØ:IF X
        1>W2 THEN GOTO 217Ø
BJ 2Ø2Ø REM GET A NEW PIECE
OL 2Ø3Ø T(P)=S(P)
NP 2Ø4Ø POSITION 1,1:PRINT PP$(P,P):PO
        SITION Ø,12
CF 2Ø5Ø REM TRY FITTING PIECE
BD 2Ø6Ø C$=PP$(P,P):X1(Ø)=X1:Y1(Ø)=Y1:
        FOR J=1 TO 4
GE 2Ø7Ø X=X(T(P),J-1)+X1:Y=Y(T(P),J-1)
        +Y1:X1(J)=X:Y1(J)=Y
DH 2Ø8Ø IF X<1 OR Y<1 OR X>W2 OR Y>W1
        THEN GOTO 226Ø
KG 2Ø9Ø IF B(Y,X)<>Ø THEN GOTO 226Ø
EM 21ØØ NEXT J
AD 211Ø REM IT FITS - PUT PIECE IN PLA
        CE
ED 212Ø B=P:FOR J=Ø TO 4
PG 213Ø X=X1(J):Y=Y1(J):GOSUB 35ØØ
FA 214Ø NEXT J
NP 215Ø X2(P)=X1:Y2(P)=Y1:P1=P1+1:U(P1
        )=P:GOTO 2Ø1Ø
MF 216Ø REM BOARD FILLED
DF 217Ø POSITION Ø,12:PRINT "SOLUTION"
        ;:POKE 752,Ø:END
JG 218Ø REM UNDRAW LAST ONE
HA 219Ø P=U(P1):U(P1)=Ø:P1=P1-1:IF P1<
        Ø THEN PRINT "THAT'S ALL":END
AN 22ØØ B=Ø:X=X2(P):Y=Y2(P):C$=" ":GOS
        UB 35ØØ
DN 221Ø X1=X:Y1=Y:FOR J=1 TO 4
GB 222Ø X=X(T(P),J-1)+X1:Y=Y(T(P),J-1)
        +Y1:X1(J)=X:Y1(J)=Y
AP 223Ø GOSUB 35ØØ
FB 224Ø NEXT J
MD 225Ø REM ROTATE THE PIECE
GJ 226Ø T(P)=T(P)+1:IF P(T(P))=P THEN
        GOTO 2Ø6Ø
IC 227Ø REM GIVE UP ON PIECE
CO 228Ø T(P)=Ø
BN 229Ø REM LOOK FOR NEW PIECE
OM 23ØØ P=P+1:IF P>12 THEN GOTO 219Ø
OI 231Ø IF T(P)<>Ø THEN 23ØØ
MF 232Ø GOTO 2Ø3Ø
IC 3ØØØ FOR J=1 TO 12:IF T(J)<>Ø THEN
        NEXT J
KE 3Ø1Ø RETURN
BD 32ØØ FOR X1=1 TO W2:FOR Y1=1 TO W1
IL 321Ø IF B(Y1,X1)=Ø THEN 323Ø
JC 322Ø NEXT Y1:NEXT X1
KI 323Ø RETURN
MO 35ØØ POSITION X,Y+2:PRINT C$:B(Y,X)
        =B
KJ 351Ø RETURN
```

Program 3: Pentominos For IBM PC/PCjr

Insert lines 110–860 from the Commodore version (Program 1).

```
100 CLS:PRINT "                PENTOMINOS":P
RINT
1000 DIM X(63,4),Y(63,4),P(64),P$(13),S(
13),T(13),B(6,20)
1001 DIM X1(5),Y1(5),X2(12),Y2(12),U(12)

1010 READ P$,N:IF N=0 GOTO 1070
1020 T=T+1:P$(T)=P$:S(T)=V+1
1030 FOR J=V+1 TO V+N:P(J)=T
1040 FOR K=0 TO 3:READ X(J,K),Y(J,K):NEX
T K,J
1050 V=V+N:PRINT P$;
1060 GOTO 1010
1070 LOCATE 5,1:PRINT"CHOOSE:":PRINT
1080 FOR J=3 TO 6:PRINT J;"BY";60/J;"":P
RINT:NEXT J
1090 INPUT "SELECT 3 THRU 6";W1
1100 IF W1<3 OR W1>6 OR W1<>INT(W1) GOTO
 1070
1110 W2=60/W1
1120 CLS
2000 REM FIND NEW SPACE TO FILL
2010 GOSUB 3000:P=J:GOSUB 3200:IF X1>W2
GOTO 2170
2020 REM GET A NEW PIECE
2030 T(P)=S(P)
2040 LOCATE 1,1:PRINT P$(P)
2050 REM TRY FITTING PIECE
2060 C$=P$(P):X1(0)=X1:Y1(0)=Y1:FOR J=1
TO 4
```

SOFTWARE PURCHA$ING POWER

Now you can get deep discounts on micro software by taking advantage of Software City's tremendous Purchasing Power. We buy from leading software publishers in wholesale quantities, so you benefit with substantial savings — whether buying for home or business.

Low prices, great selection and friendly service have made us the nation's largest chain of software stores. Start putting our Purchasing Power to work for you today!

SOFTWARE ALWAYS DISCOUNTED

Programs, peripherals, disks, accessories, and custom forms plus a complete selection of books and magazines.

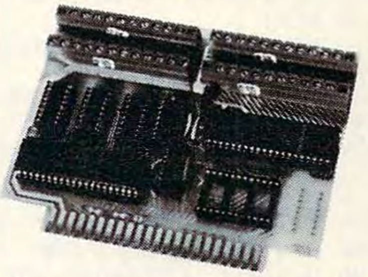
Software City®

VISIT A **SOFTWARE CITY** PROGRAM DISCOUNT CENTER NEAR YOU -

ALABAMA: Birmingham (205) 591-8314 • Huntsville (205) 536-9456 **ARIZONA:** Phoenix (602) 264-1422 Tucson (602) 721-1008 **COLORADO:** Westminster (303) 430-8708 **CONNECTICUT:** Hartford (coming soon) • Orange (203) 799-2119 • Stamford (203) 359-1414 **DELAWARE:** Wilmington (302) 478-4880 **FLORIDA:** Orange Park (904) 264-3102 • St. Petersburg (813) 345-5132 Sarasota (813) 923-4040 • Tampa (813) 961-8081 **GEORGIA:** Atlanta (404) 257-1833 **ILLINOIS:** Arlington Heights (312) 259-4260 • Chicago (coming soon) **INDIANA:** Fort Wayne (coming soon) Indianapolis (coming soon) **IOWA:** Davenport (319) 386-2345 **KENTUCKY:** Louisville (502) 893-3838 **MARYLAND:** Bethesda (301) 468-1001 **MASSACHUSETTS:** Lexington (617) 861-0116 • W. Springfield (413) 739-5101 **MICHIGAN:** Ann Arbor (313) 996-4500 Okemos (coming soon) • Southfield (313) 559-6966 • Sterling Hts. (313) 978-3700 **MISSOURI:** Independence (816) 461-1260 • St. Louis (coming soon) **NEBRASKA:** Omaha (coming soon) **NEW JERSEY:** Bergenfield (201) 387-8388 • Cherry Hill (609) 424-8155 Englishtown (201) 972-1150 • Fair Lawn (201) 791-8793 • Green Brook (201) 968-7780 Hamilton (609) 890-1066 • Linwood (609) 927-3393 • Little Falls (coming soon) Midland Park (201) 447-9794 • Montvale (201) 391-0931 • Morristown (201) 267-3353 Pine Brook (201) 575-4574 • Pompton Lakes (201) 831-1004 • Red Bank (201) 747-6490 Ridgefield (201) 943-9444 • Summit (201) 273-7904 • Teaneck (201) 692-8298 **NEW YORK:** Albany (coming soon) • Brooklyn (coming soon) • Buffalo (coming soon) • Fairport (716) 223-3723 Forest Hills (212) 261-1141 • Great Neck (516) 482-4929 • Long Island (coming soon) Manhattan (212) 832-0760 • Mt. Kisco (914) 666-6036 • North White Plains (914) 946-1800 Pomona (914) 354-3706 • Poughkeepsie (coming soon) • Scarsdale (coming soon) Staten Island (212) 351-9217 • Syracuse (315) 445-2577 **NORTH CAROLINA:** Charlotte (704) 366-5218 • Fayetteville (919) 864-7293 **OHIO:** Centerville (513) 439-1237 Columbus (614) 888-6660 • Mayfield Hts. (216) 473-8124 **OKLAHOMA:** Tulsa (918) 744-0558 **PENNSYLVANIA:** Bethel Park (412) 854-1777 • Exton (215) 524-1483 Fairless Hills (215) 943-4544 • King of Prussia (coming soon) • Pittsburgh (412) 367-0441 Whitehall (215) 434-3060 **SOUTH CAROLINA:** Spartanburg (803) 574-4713 **TEXAS:** Austin (512) 458-2101 **VERMONT:** Burlington (802) 864-6981 **VIRGINIA:** Fairfax (coming soon) Falls Church (703) 845-9393 • Richmond I (804) 740-8400 • Richmond II (804) 320-2244 Virginia Beach (804) 463-8220 **WASHINGTON:** Bellevue (206) 451-1141 **CANADA:** (coming soon) **ENGLAND:** London 013529220 **PUERTO RICO:** San Juan (809) 781-9357

BUSINESS SOFTWARE: Catalog at all stores.
RETAIL STORE FRANCHISES: $50,000 estimated investment. Offering by prospectus only. Direct inquiries to SOFTWARE CITY FRANCHISE DEPT. • 1415 Queen Anne Road, Teaneck, NJ 07666

NEW!
Universal Input/Output Board for VIC-20/64



- 16 channel 8-bit A/D converter with 100 microsecond sampling time.
- 1 D/A output.
- 16 high voltage/high current discrete outputs.
- 1 EROM socket.
- Use multiple boards for additional channels up to 6 boards.

VIC-20 uses MW-311V $205.00
CBM-64 uses MW-311C $225.00

MW-302: VIC-20/64 Parallel Printer Interface.

Works with all centronics type parallel matrix & letter printers and plotters— Epson, C.Itoh, Okidata, Nec, Gemini 10, TP-I Smith Corona, and most others. Hardware driven; works off the serial port. Quality construction: Steel DIN connectors & shielded cables. Has these switch selectable options: Device 4, 5, 6 or 7; ASCII or PET ASCII; 7-bit or 8-bit output; upper & lower case or upper only. Recommended by PROFESSIONAL SOFTWARE for WordPro 3 Plus for the 64, and by City Software for PaperClip.

MW-302 .... $119.95

Dealer inquiries invited.

VISA MasterCard

Micro World Electronix, Inc.
3333 S. Wadsworth Blvd. #C105,
Lakewood, CO 80227
(303) 987-9532 or 987-2671

```
2070 X=X(T(P),J-1)+X1:Y=Y(T(P),J-1)+Y1:X
1(J)=X:Y1(J)=Y
2080 IF X<1 OR Y<1 OR X>W2 OR Y>W1 GOTO
2260
2090 IF B(Y,X)<>0 GOTO 2260
2100 NEXT J
2110 REM IT FITS - PUT PIECE IN PLACE
2120 B=P:FOR J=0 TO 4
2130 X=X1(J):Y=Y1(J):GOSUB 3500
2140 NEXT J
2150 X2(P)=X1:Y2(P)=Y1:P1=P1+1:U(P1)=P:G
OTO 2010
2160 REM BOARD FILLED
2170 LOCATE 15,1:PRINT "  SOLUTION";:END
2180 REM UNDRAW LAST ONE
2190 P=U(P1):U(P1)=0:P1=P1-1:IF P1<0 THE
N PRINT"THAT'S ALL":END
2200 B=0:X=X2(P):Y=Y2(P):C$=" ":GOSUB 35
00
2210 X1=X:Y1=Y:FOR J=1 TO 4
2220 X=X(T(P),J-1)+X1:Y=Y(T(P),J-1)+Y1:X
1(J)=X:Y1(J)=Y
2230 GOSUB 3500
2240 NEXT J
2250 REM ROTATE THE PIECE
2260 T(P)=T(P)+1:IF P(T(P))=P GOTO 2060
2270 REM GIVE UP ON PIECE
2280 T(P)=0
2290 REM LOOK FOR NEW PIECE
2300 P=P+1:IF P>12 GOTO 2190
2310 IF T(P)<>0 GOTO 2300
2320 GOTO 2030
3000 FOR J=1 TO 12:IF T(J)<>0 THEN NEXT
J
3010 RETURN
3200 FOR X1=1 TO W2:FOR Y1=1 TO W1
3210 IF B(Y1,X1)=0 GOTO 3230
3220 NEXT Y1,X1
3230 RETURN
3500 LOCATE Y+2,X:PRINT C$:B(Y,X)=B
3510 RETURN
```

Program 4: Pentominos For TI-99/4A

Insert lines 110–860 from the Commodore version (Program 1).

(**Note:** *If using a disk drive, type CALL FILES(1) before loading and running this program.)*

```
40 CALL CLEAR
50 PRINT "{8 SPACES}PENTOMINOS": :
60 GOTO 870
70 FOR I=1 TO LEN(A$)
80 CALL HCHAR(ROW,COL+I,ASC(SEG$(A$
   ,I,1)))
90 NEXT I
100 RETURN
870 DIM XX(63,4),YY(63,4),PP(64),PP
   $(13),SS(13),TT(13),BB(6,20)
880 DIM XX1(5),YY1(5),XX2(12),YY2(1
   2),UU(12)
890 CT=5
900 READ P$,N
910 IF N=0 THEN 1040
920 T=T+1
930 PP$(T)=P$
940 SS(T)=V+1
950 FOR J=V+1 TO V+N
960 PP(J)=T
970 FOR K=0 TO 3
980 READ XX(J,K),YY(J,K)
990 NEXT K
1000 NEXT J
1010 V=V+N
1020 PRINT P$;
1030 GOTO 900
1040 CALL CLEAR
1050 PRINT " CHOOSE:": :
1060 FOR J=3 TO 6
1070 PRINT J;" BY ";60/J
1080 NEXT J
1090 PRINT
1100 INPUT " SELECT 3 THRU 6: ":W1
1110 IF (W1<3)+(W1>6)+(W1<>INT(W1))
     THEN 1040
1120 W2=60/W1
1130 CALL CLEAR
1140 REM FIND NEW SPACE TO FILL
1150 GOSUB 1930
1160 P=J
1170 GOSUB 1970
1180 IF X1>W2 THEN 1500
1190 REM GET A NEW PIECE
1200 TT(P)=SS(P)
1210 ROW=CT
1220 COL=5+CT
1230 A$=PP$(P)
1240 GOSUB 70
1250 REM TRY FITTING PIECE
1260 C$=PP$(P)
1270 XX1(0)=X1
1280 YY1(0)=Y1
1290 FOR J=1 TO 4
1300 X=XX(TT(P),J-1)+X1
1310 Y=YY(TT(P),J-1)+Y1
1320 XX1(J)=X
1330 YY1(J)=Y
1340 IF (X<1)+(Y<1)+(X>W2)+(Y>W1)TH
     EN 1840
1350 IF BB(Y,X)<>0 THEN 1840
1360 NEXT J
1370 REM IT FITS - PUT PIECE IN PLA
     CE
1380 B=P
1390 FOR J=0 TO 4
1400 X=XX1(J)
1410 Y=YY1(J)
1420 GOSUB 2030
1430 NEXT J
1440 XX2(P)=X1
1450 YY2(P)=Y1
1460 P1=P1+1
1470 UU(P1)=P
1480 GOTO 1150
1490 REM BOARD FILLED
1500 ROW=15
1510 COL=5+CT
1520 A$="SOLUTION"
1530 GOSUB 70
1540 ROW=17
1550 COL=5
1560 A$="FIND ANOTHER SOLUTION?"
1570 GOSUB 70
1580 CALL KEY(3,K,S)
1590 IF S<>1 THEN 1580
1600 IF CHR$(K)="Y" THEN 1620
1610 END
1620 REM UNDRAW LAST ONE
1630 P=UU(P1)
1640 UU(P1)=0
1650 P1=P1-1
1660 IF P1>=0 THEN 1690
1670 PRINT "THAT'S ALL"
1680 STOP
1690 B=0
1700 X=XX2(P)
```

```
1710 Y=YY2(P)
1720 C$=" "
1730 GOSUB 2030
1740 X1=X
1750 Y1=Y
1760 FOR J=1 TO 4
1770 X=XX(TT(P),J-1)+X1
1780 Y=YY(TT(P),J-1)+Y1
1790 XX1(J)=X
1800 YY1(J)=Y
1810 GOSUB 2030
1820 NEXT J
1830 REM ROTATE THE PIECE
1840 TT(P)=TT(P)+1
1850 IF PP(TT(P))=P THEN 1260
1860 REM GIVE UP ON PIECE
1870 TT(P)=0
1880 REM LOOK FOR NEW PIECE
1890 P=P+1
1900 IF P>12 THEN 1630
1910 IF TT(P)<>0 THEN 1890
1920 GOTO 1200
1930 FOR J=1 TO 12
1940 IF TT(J)=0 THEN 1960
1950 NEXT J
1960 RETURN
1970 FOR X1=1 TO W2
1980 FOR Y1=1 TO W1
1990 IF BB(Y1,X1)=0 THEN 2020
2000 NEXT Y1
2010 NEXT X1
2020 RETURN
2030 ROW=Y+1+CT
2040 COL=X+CT
2050 A$=C$
2060 GOSUB 70
2070 BB(Y,X)=B
2080 RETURN
```

Program 5:
Pentominos For The Color Computer

Insert lines 110–860 from the Commodore version (Program 1).

```
100 CLS:PRINT"{11 SPACES}PENTOMINOS"
999 PCLEAR 1
1000 DIM X(63,4),Y(63,4),P(64),P$(1
     3),S(13),T(13),B(6,20)
1001 DIM X1(5),Y1(5),X2(12),Y2(12),
     U(12)
1010 READ P$,N:IF N=0 GOTO 1070
1020 T=T+1:P$(T)=P$:S(T)=V+1
1030 FOR J=V+1 TO V+N: P(J)=T
1040 FOR K=0 TO 3:READ X(J,K),Y(J,K
     ):NEXT K,J
1050 V=V+N:PRINT P$;
1060 GOTO 1010
1070 PRINT@64,"CHOOSE:"
1080 FOR J=3 TO 6:PRINT J;" BY";60/
     J:NEXT J
1090 INPUT "SELECT 3 THRU 6";W1
1100 IF W1<3 OR W1>6 OR W1<>INT(W1)
      GOTO 1070
1110 W2=60/W1
1120 CLS
2000 REM FIND NEW SPACE TO FILL
2010 GOSUB 3000:P=J:GOSUB 3200:IF X
     1>W2 GOTO 2170
2020 REM GET A NEW PIECE
2030 T(P)=S(P)
2040 PRINT@33,P$(P)
2050 REM TRY FITTING PIECE
2060 C$=P$(P):X1(0)=X1:Y1(0)=Y1:FOR
      J=1 TO 4
2070 X=X(T(P),J-1)+X1:Y=Y(T(P),J-1)
     +Y1:X1(J)=X:Y1(J)=Y
2080 IF X<1 OR Y<1 OR X>W2 OR Y>W1
     GOTO 2260
2090 IF B(Y,X)<>0 GOTO 2260
2100 NEXT J
2110 REM IT FITS - PUT PIECE IN PLA
     CE
2120 B=P:FOR J=0 TO 4
2130 X=X1(J):Y=Y1(J):GOSUB 3500
2140 NEXT J
2150 X2(P)=X1:Y2(P)=Y1:P1=P1+1:U(P1
     )=P:GOTO 2010
2160 REM BOARD FILLED
2170 PRINT@385,"SOLUTION":END
2180 REM UNDRAW LAST ONE
2190 P=U(P1):U(P1)=0:P1=P1-1:IF P1<
     0 THEN PRINT"THAT'S ALL":END
2200 B=0:X=X2(P):Y=Y2(P):C$=" ":GOS
     UB 3500
2210 X1=X:Y1=Y:FOR J=1 TO 4
2220 X=X(T(P),J-1)+X1:Y=Y(T(P),J-1)
     +Y1:X1(J)=X:Y1(J)=Y
2230 GOSUB 3500
2240 NEXT J
2250 REM ROTATE THE PIECE
2260 T(P)=T(P)+1:IF P(T(P))=P GOTO
     2060
2270 REM GIVE UP ON PIECE
2280 T(P)=0
2290 REM LOOK FOR NEW PIECE
2300 P=P+1:IF P>12 GOTO 2190
2310 IF T(P)<>0 GOTO 2300
2320 GOTO 2030
3000 FOR J=1 TO 12:IF T(J)<>0 THEN
     NEXT J
3010 RETURN
3200 FOR X1=1 TO W2:FOR Y1=1 TO W1
3210 IF B(Y1,X1)=0 GOTO 3230
3220 NEXT Y1,X1
3230 RETURN
3500 PRINT @X+(Y+2)*32,C$;:B(Y,X)=B
3510 RETURN
```

Program 6: Pentominos For The Apple

Insert lines 110–860 from the Commodore version (Program 1).

```
1000  DIM X(63,4),Y(63,4),P(64),P$(13),
     S(13),T(13),B(6,20)
1001  DIM X1(5),Y1(5),X2(12),Y2(12),U(1
     2)
1003  HOME : HTAB 16: PRINT "PENTOMINOS
     ": PRINT
1010  READ P$,N: IF N = 0 GOTO 1070
1020 T = T + 1:P$(T) = P$:S(T) = V + 1
1030  FOR J = V + 1 TO V + N:P(J) = T
1040  FOR K = 0 TO 3: READ X(J,K),Y(J,K
     ): NEXT K,J
1050 V = V + N: PRINT P$;
1060  GOTO 1010
1070  PRINT : VTAB (5): PRINT "CHOOSE:"
     : PRINT
1080  FOR J = 3 TO 6: PRINT J;" BY ";60
      / J: PRINT : NEXT J
1090  INPUT "SELECT 3 THRU 6? ";W1
1100  IF W1 < 3 OR W1 > 6 OR W1 <  >  INT
     (W1) GOTO 1070
1110 W2 = 60 / W1
1120  HOME
2000  REM  FIND NEW SPACE TO FILL
2010  GOSUB 3000:P = J: GOSUB 3200: IF
     X1 > W2 GOTO 2170
```

```
2020  REM  GET A NEW PIECE
2030 T(P) = S(P)
2040  VTAB 1: PRINT P$(P): VTAB 12
2050  REM  TRY FITTING PIECE
2060 C$ = P$(P):X1(Ø) = X1:Y1(Ø) = Y1: FOR
     J = 1 TO 4
2070 X = X(T(P),J - 1) + X1:Y = Y(T(P),
     J - 1) + Y1:X1(J) = X:Y1(J) = Y
2080  IF X < 1 OR Y < 1 OR X > W2 OR Y >
     W1 GOTO 2260
2090  IF B(Y,X) <  > Ø GOTO 2260
2100  NEXT J
2110  REM  IT FITS - PUT PIECE IN PLACE

2120 B = P: FOR J = Ø TO 4
2130 X = X1(J):Y = Y1(J): GOSUB 3500
2140  NEXT J
2150 X2(P) = X1:Y2(P) = Y1:P1 = P1 + 1:
     U(P1) = P: GOTO 2010
2160  REM  BOARD FILLED
2170  VTAB 1: PRINT "  SOLUTION";: END

2180  REM  UNDRAW LAST ONE
2190 P = U(P1):U(P1) = Ø:P1 = P1 - 1: IF
     P1 < Ø THEN  PRINT "THAT'S ALL": END

2200 B = Ø:X = X2(P):Y = Y2(P):C$ = " "
     : GOSUB 3500
2210 X1 = X:Y1 = Y: FOR J = 1 TO 4
2220 X = X(T(P),J - 1) + X1:Y = Y(T(P),
     J - 1) + Y1:X1(J) = X:Y1(J) = Y
2230  GOSUB 3500
2240  NEXT J
2250  REM  ROTATE THE PIECE
2260 T(P) = T(P) + 1: IF P(T(P)) = P GOTO
     2060
2270  REM  GIVE UP ON PIECE
2280 T(P) = Ø
2290  REM  LOOK FOR NEW PIECE
2300 P = P + 1: IF P > 12 GOTO 2190
2310  IF T(P) <  > Ø GOTO 2300
2320  GOTO 2030
3000  FOR J = 1 TO 12: IF T(J) <  > Ø THEN
      NEXT J
3010  RETURN
3200  FOR X1 = 1 TO W2: FOR Y1 = 1 TO W
    1
3210  IF B(Y1,X1) = Ø GOTO 3230
3220  NEXT Y1,X1
3230  RETURN
3500  VTAB Y + 4: HTAB X: PRINT C$:B(Y,
    X) = B
3510  RETURN
```

©

Pitstop

Shay Addams

Racing games are nothing new, but *Pitstop* from Epyx incorporates a realistic element of the sport that sets it apart from everything else on the track. In addition to zooming around the course as fast as possible, you must develop a solid plan for maneuvering your three-man pit crew when you're forced to pull in for fresh tires and refueling. The game is available on cartridge for Atari, Commodore 64, and Coleco Adam computers.

The action takes place on one of six speedways, all based on genuine tracks such as Le Mans and Monaco. You can race at any one, or opt for the "Mini-Circuit," in which the program picks three courses at random for you to complete, one after the other. Hardcore speed demons will prefer the "Grand Circuit"—it requires you to cover all six tracks in succession, a grueling marathon event. The number of laps per race can be set to three, six, or nine; skill levels include Rookie, Semi-Pro, and Pro. Up to four players can compete by taking turns.

The race kicks off as you push forward on the stick to accelerate. The perspective and graphics are similar to *Enduro*, but unfortunately not as detailed as *Pole Position*. While you accelerate, the gears shift automatically, accompanied by authentic sound effects. The screen scrolls vertically, with a green background and yellow cars. Your speed, elapsed time, and current lap are constantly displayed.

No more than two other cars are on the track simultaneously, but they are programmed to swerve into your path or travel side by side to prevent your passing them. The main thing to watch out for is bumping into other cars or the sides of the road. An accident won't cause a colorful explosion the way it does in *Pole Position*, but it will reduce your speed as in *Baja Buggies*.

Trouble With Tires

This is where *Pitstop* takes a detour from the familiar "race around the track" scenario of similar games. When you smash into another car or the railing alongside the road, the corresponding tire is damaged. Starting off a deep blue, the tires change to a different hue each time you have an accident. Sustain too much damage and the tire explodes, knocking you out of the race. You've got to keep an eye on the color of all four tires and be ready to pull into the pits when they turn a bright red (indicating that they'll burst on the next collision).

The pit area is located to the right of the finish line. An inset map on the left displays an overhead view of the course, with your car's current position and the finish line prominently marked. Turn into the off-ramp on the right as you pass the finish line, and the scene cuts to a three-quarter perspective of your car sitting in the pits. Now your vehicle is revealed as one of those low-slung, Indy 500-type racers, and is larger and much more detailed.

Action In The Pits

A member of your pit crew waits on each side of the car, standing by to change the tires. Another is behind you, gas hose in hand. If the horizontal fuel gauge says you're running low, it's best to get the gas pumping immediately. This is done by using the joystick to move a cursor over the man, then hitting the fire button. Now you can steer him into place, where he automatically starts refilling your tank.

Tires are changed by activating one of the other men and moving him to the tire you want removed. He'll latch onto it, and you can guide him to a stack of fresh tires. When he touches the stack, the tire he's holding turns a deep blue to indicate that he's got a new tire, which he can then attach to the car. But keep your eyes on the gas gauge, because if you don't remove the nozzle when the tank's topped off, the gas spills over and you have to fill it up again.

While all this is going on, a timer at the top right of the screen shows the seconds ticking away to remind you how much time you're losing in the pits. Another digital display at bottom left tells you how much overall time has elapsed since the race began. To underscore the urgency of getting out of the pits as quickly as possible, the rest of the cars keep racing past in the background, their engines buzzing as they gain distance on you. When you're ready to roll, position the cursor over the man in front of the car and he'll raise his flag to wave you back onto the track.

Multiplayer Competition

You can make it through three laps around most tracks without a stop for gas or tire changes, but the only fun involved in this is trying to beat your best time for the same course. *Pitstop*'s more enjoyable in group play. When one driver completes the set number of laps, the next one takes a whirl around the track. After the race, each player's time is posted, along with his portion of the $94,000 prize money. If you're competing in a Mini- or Grand circuit, the overall winnings are displayed at the bottom. If two or more players tie, the one who started first wins, so flip a coin to determine who goes first.

In addition to the exciting competition and action, *Pitstop* requires strategy and split-second decision-making that are missing in other racing games. Should you try to finish the race in spite of a severely damaged tire, or pull into the pits and at least insure that you complete the race? Is there time to change all four tires? Situations like these put a real edge on the game play. Since veteran race car drivers agree that many professional races are won in the pits, not on the track, *Pitstop* has to be one of the most realistic and playable racing simulations available.

Pitstop
Epyx Computer Software
1043 Kiel Court
Sunnyvale, CA 94089
Atari, 64 versions, $39.95
Coleco Adam version, $53

©

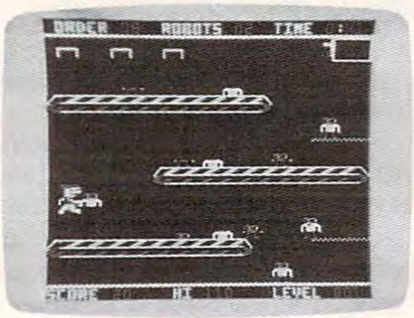
Panic Button For VIC And TRS-80 Color Computer

Michael B. Williams

Not wanting to imitate the other arcade games on the market, First Star has introduced a game which is refreshingly original—and very entertaining.

In *Panic Button*, you have been hired to assemble various objects whose parts parade on three continuously moving conveyor belts. On the first level, robot parts are ejected from the three chutes at the top of the screen. Not only must you catch up to them, but you must assemble them *in the proper order* to be given credit for the item. Should you accidentally place the robot's feet on its head (an improper sequence), no credit is given for the item, nor are its parts reusable, since there is no way to separate any two joined parts. I found it frustrating: No sooner had I completed two-thirds of an object than an incorrect part dropped from a chute and attached itself to mine. Surprisingly, this occurred in my favor as often as it did against me.

After a while, especially during the harder screens, these



Parts continuously flow from three conveyor belts in Panic Button *(Color Computer version).*

"rejected" objects (obviously thrown by your boss in anger) begin to fly around the screen, at times bumping into you and making your job even more difficult. I almost found it more than I could handle, having to race around the screen to retrieve objects moving nearly as quickly as I was.

Houses, Telephones, And Lamps

In later screens, you will find three-layered cakes, houses, telephones, televisions, and finally lamps dropping from the chutes. After every screen, it becomes progressively more challenging to complete your minimum order as the conveyor belts move faster and the number of objects you must assemble within the two-minute limit increases. After completing the second screen, you have a much-awaited opportunity to fling a pie into your boss's face—but that is not what gets you fired.

Panic Button breaks away from the three-man tradition and provides you with only one worker. Should he fail to fulfill his minimum order of assembled items, the boss spares no time in firing him (where's another pie?).

You have only one thing going for you in this game—the "panic button." You activate it by using the joystick button to move your character over to the operating switch. This slows the conveyors to a halt, allowing you to freely gather the objects around the factory. (Unfortunately, it has no effect on the clock, which continues to run down.) But your enraged boss soon comes to restart the conveyor belt, and you continue your frantic race against time.

An "external" panic button not mentioned in the rather skimpy documentation is the space bar: Pressed at any time during the game, it pauses the action indefinitely. I found myself using this panic button more than the other.

First Star's decision to develop a game with a unique concept is refreshing, but an original game is not always a good game. With *Panic Button*, however, First Star has succeeded. I recommend it to anyone who enjoys nonstop action—and even to those who do not. After all, that is the reason the "panic button" exists.

Panic Button
First Star Software, Inc.
22 East 41st Street
New York, NY 10017
Color Computer or VIC (8K expansion) tape, $24.95
VIC cartridge $34.95
Color Computer cartridge $39.95

©

ATARI

600XL .... CALL
800XL .... CALL
1200XL $409*
*Reflects $100 Atari Rebate

1050 DRIVE $335
1025 PRINTER $399
1020 COLOR PTR .. $219
1027 PRINTER $279
1010 RECORDER $72
MICROBITS INFC $78
80 COLUMN BD $249
TECHNICAL NOTES $25
REAL TIME CLK $38
810 DRIVE $419

DRIVES

ASTRA 1620 .... $469
RANA 1000 .... $295
PERCOM 88-S1 ..... $298
PERCOM 40-S1 ..... $418
PERCOM 40-S2 ..... $718
PERCOM 44-S1 ..... $510
PERCOM 44-S2 ..... $929

MEMORIES

48K RAM (INTEC) $95
64K RAM (INTEC) $119
48K RAM (MOSAIC) . $109
64K RAM (MOSAIC) . $145
128K RAM DISK $299
32K RAM (MOSAIC) .. $68

ATARI SOFTWARE

ADVENTURE INT'L
Adv. 1-12 each (C) $18
Preppie (C/D) $20
Preppie II (C/D) $23
Diskey (D) $33
Sea Dragon (C/D) $23

APX
Eastern Front (C/D) .. $23
747 Land Sim. (C/D) .. $17
Fig-Forth (C) $30

ATARI INC.
Microsoft Basic II (R) . $62
Mickey in Great Outdoors (C/D) $36
Paint (D) $30
Speed Reading (C) ... $54
Qix (R) $30
Dig Dug (R) $30
Atari Writer (R) $68
Donkey Kong (R) $30
Time Wise (D) $23
Visicalc (D) $139
Juggles House (C/D) .. $22
Juggles Rnbw (C/D) .. $22
Pilot (Home) $55
Galaxian $30
Defender $30
ET $34
Microsoft Basic (D) .. $62
Macro Ass. & Edit (D) . $62
Assembler Editor (R) . $42
Basic Cartridge (R) .. $45
Pac Man (R) $30
Centipede (R) $30
Caverns of Mars (D) .. $28
Star Raiders (R) $30
Conv. Lang. Ea. (C) ... $42
Music Composer (R) .. $31
Super Breakout (R) ... $26
My First Alphabet (D) . $26
Prog. 2 & 3 (ea.)(C) $21
Word Processor (D) . $102
Pilot (Educ.) $92
Touch Typing (C) $19
Home File Mngr (D) ... $36

AUTOMATED SIMUL.
Hellfire Warrior (C/D) . $27
Kng Arthr's Heir (C/D) $20
Invasion Orion (C/D) .. $17
Temple of Aps. (C/D) . $27
Star Warrior (C/D) $27
Dragon's Eye (D) $20
Crush Crumble (C/D) . $20

AVALON HILL
VC (D) $17
B-1 Nuc. Bomber (C) .. $12
Legionnaire (C) $23

BRODERBUND
Sky Blazer (D) $22
Bank St. Writer (D) ... $46
A.E. (D) $23
Arcade Machine (D) ... $39
Choplifter (D) $23

CBS
Mountain King (R) $27
Boulders & Bombs (R) $27
Krazy (each) $34

CONTINENTAL SOFT.
Home Accountant (D) $48
Tax Advantage (D) ... $39

DATASOFT
Text Wizard (D) $65
Graphic Master (D) ... $27
Micro Painter (D) $23
Lisp Interpreter (D) .. $79
Graphics Gen. (D) $17
Basic Compiler (D) ... $65
Zaxxon (C/D) $27

DON'T ASK
Sam (D) $39
P.M. Animator (D) $23
Teletari (D) $27

EDU-WARE
Prisoner II (D) $27
Spelling Bee (D) $27
Compu-Read (D) $20
Compu-Math Fr. (D) . $27
Compu-Math Dec. (D) . $27

EDUCATIONAL SOFT.
Tricky Tutorial 1,2,3 or 4 (C/D) $15
Tricky Tutorial 5,6 or 7 (C/D) $22

INFOCOM
Suspended (D) $34
Zork I, II or III (D) $27
Starcross (D) $27
Deadline (D) $34

JV SOFTWARE
Jrny to Plnts (C/D) $20
Action Quest (C/D) ... $20
Ghost Encount. (C/D) . $20

LJK
Letter Perfect (D) $74
Data Perfect (D) $74

ON-LINE
Ultima II (D) $39
Marauder (D) $23
Lunar Leeper (D) $20
Wiz & Princess (D) ... $22
Frogger (C/D) $23
Crossfire (R) $23

OPTIMIZED SYSTEMS
C-65 (D) $58
Bug-65 (D) $23
Max-65 (D) $58
Basic A + (D) $58

ROKLAN
Gorf (D) $27
Gorf (R) $30
Wizard of Wor (D) $27
Wizard of Wor (R) $30

SIRIUS
Alpha Shield (R) $27
Wavy Navy (D) $23
Bandits (D) $23

SPINNAKER
Snooper Troop 1,2 (D) . $30
Kindercomp (D) $20
Rhymes & Riddles (D) $20
Hey Diddle Diddle (D) $20
Srch Amzng Thngs (D) $27
Story Machine (D) $23
Face Maker (D) $23

STRATEGIC SIM.
Cosmic Balance (D) .. $27
Cosmic Balance II (D) $27
Tigers In Snow (C/D) .. $27
Battle of Shiloh (C/D) . $27
Battle of Norm. (C/D) . $27
Galactic Gladiator (D) $27
Cytron Masters (D) ... $27

SYNAPSE SOFTWARE
File Mngr 800 + ... $65
Protector II (D) $23 (R) $29
Shamus ... (D) $23 (R) $29
Fort Apocalypse (C/D) $23
Shamus II (C/D) $23
Necromancer (C/D) ... $23
Pharoh's Curse (C/D) . $23

THORN EMI
Soccer (R) $34
Jumbo Jet (R) $34
Submarine Comm. (R) $34

USA
Atari World (D) $39
3-D Sprgrphcs (C/D) .. $27

MISCELLANEOUS
Sargon II ... (C) $20 (D) $23
Financial Wizard (D) .. $41
Castle Wolfenstein (D) $20
Master Type (D) $27
Millionaire (D) $52
Astro Chase (D) $22
Ali Baba (D) $22
Miner 2049er (R) $34
Sammy Sea Serp. (C) . $13
Pinball (D) $20

SPECIALS

Gemini 15X Printer $399
Axiom AT-100 Printer (with interface $229
Astra Double Density Dual Drive $469
Rana 1000 Drive $319
Bit-3 80 Column Board $245
Mannesmann Talley 160L Printer $589
Atari 400 Keyboard (In Home) $35
Programmer Kit .. $48 Entertainer Kit ... $64
Wico Joystick ... $23 Wico Trackball ... $49

ASTRA 1620 DISK DRIVE SYSTEM

MORE FOR YOUR MONEY
DOUBLE OR SINGLE DENSITY
TWO DRIVES

SPECIAL $469

Printers/Etc.

GEMINI 10X .. $279
GORILLA $199
PROWRITER . $345
SMITH TPI ... $488

CITOH
Prowriter $345
Prowriter II $629
Starwriter $1149
Printmaster $1448

NEC
8023 A-C $409
3510 .... $1375
3530 .... $1579
3550 .... $1779
7710/7730 .... $1998

SILVER REED P $669
QUME 11/40 + ... $1299

OKI-DATA
Microline 82A $398
Microline 83A $638
Microline 84P $958
Microline 92 .... $488
Microline 93 .... $858

DIABLO
620R $939
630R $1719

MONITORS

AMDEK
Color I $289
V300 .... $139
V300A $149
Color II $449

NEC
GRN (JB1260) $115
GRN (JB1201) $155
Color Composite $298
RGB Color $598

MODEMS

HAYES
Smartmodem $209
Smartmodem 1200 .. $498
Micromodem II $259

NOVATION
J-Cat $99
Apple Cat II $259
D-Cat $149

D = Disk
R = ROM (Cartridge)
C = Cassette

COSMIC COMPUTERS
UNLIMITED

727 BREA CANYON RD., SUITE 16
WALNUT, CA 91789

ORDER LINES OPEN MON-SAT 8 am - 8 pm

(800) 626-7642
PLEASE FOR ORDERS ONLY
SORRY, NO COD'S

(714) 594-5204
FOR TECHNICAL INFO, ORDER INQUIRIES, OR FOR CALIFORNIA ORDERS

Add $2.50 shipping per software order in continental U.S. Add $5.00 shipping per software order for AK, HI, FPO-APO. Add $10.00 or 15% (whichever is greater) per software order for non-U.S. Call for cost of hardware shipping. Calif. residents add 6½% sales tax. Cashiers checks or money orders filled within 24 hours for items in stock. Personal checks require 4 weeks to clear. MasterCard and Visa OK for software only within continental U.S., add 3% surcharge. Include card no., expiration date and signature. Due to our low prices, all sales are final. All defective returns must have a return authorization number. Please call to obtain one before returning goods for replacement or repair. Prices & availability subject to change.

COMMODORE

CBM 64 .... CALL
1541 DISK DRIVE ... $239

1701 Color Monitor .. $255
1525 Printer $239
1520 Color Ptr $169
Card ? (Infc) $60
Light Pen $29
Cassette Infc $29
Card ? Software $16
1530 Recorder $59
1600 Modem $59
1650 Auto Modem $89
CMB 64 Ref Guide $18
The Connection (Infc) .. $85
MSD Disk Drive $339
PTI 45 Lot Board $59

64 SOFTWARE 64

Multiplan .... CALL
Script 64 .... $77
Calc Result Prof $114
Calc Result Easy $68
The Home Accountant $48
Delphis Oracle $89
Word Pro 3 with Spell $78

ACCESS SOFTWARE
Neutral Zone (C/D) ... $26
Sprite Master (C/D) ... $27

AVALON HILL
Nukewar (C) $12
Planet Miners (C) $12
Androm. Conquest (C) $14
Midway Campaign (C) $12
North Atl. Convoy (C) $12
Comp. Stcks/Bnds (C) $15
Computer Football (C) $18
Telengard (C) $16

BATTERIES INCLUDED
Paper Clip (D) $85
Delphis Oracle (D) 89

BRODERBUND
Choplifter (R) $29
Serpentine (R) $27
Seafox (R) $27
David's Midnight (D) .. $23

COMMODORE
Easy File (D) $75
Easy Finance (D) $38
Easy Mail (D) $38
Easy Script (D) $75
Easy Schedule (D) ... $59
Logo (R) $75
Pilot (D) $75
Assembler (D) $38
Music Machine (D) ... $25
Music Composer (D) .. $25
Meza Music (D) $75
Video/Music Supt. (D) $38
Jupiter Lander (R) $25
Radar Rat Race (R) ... $25
Sea Wolf (R) $25
Kickman (R) $25

COMM-DATA
Pakacuda . (C) $14 (D) $18
Escp. MCP . (C) $14 (D) $18
Centropods (C) $14 (D) $18

COMPUTERMAT
Arcade-Pak (C) $18
Education-Pak (C) ... $18

CREATIVE SOFTWARE
Moondust (R) $25
Trashman (R) $25
Save New York (R) ... $25
Astroblitz (R) $25
Household Fin. (D) ... $25

DATA 20
Video Pak 80 ...... $139
Z80 Video Pak $229

EN-TECH
Finance Calc 64 ...... $34
Data Base 64 .... $56
Invoice Ease 64 ..... $56

EPYX
Temple of APS (D) ... $27
Upper Reach. APS (D) $14
Jumpman (D) $27

HES
HES Modem $59
6502 Prof.Dev.Sys.(C) . $22
Hesmon 64 (R) $27
Turtle Grapics II (R) .. $41
Heswriter 64 (R) $32
Gridrunner (R) $27
Retroball (R) $27

INFOCOM
Zork I, II or III (D) $27
Deadline (D) $35
Starcross (D) $27

JIN SAM
Mini-Jini (R) $75

LITTLE WIZARD
Pro.Mail.List (C)$22 (D)$25
Stockmaster (Inventory) (C)$25 (D)$28

LOGISTIC
Datacalc 64 (C)$55 (D)$59
Home Journal (D) $55

MICROSPEC
Payroll System (D) ... $73
Inventory Pkg (D) $73
General Ledger (D) ... $73
Disk Data Mgr (D) $62
Mail List Mgr (D) $41
Checkbook Mgr (D) ... $39

M-SOFT
M-File (D) $89

ON-LINE
Frogger (D) $23
Jawbreaker (D) $20

PACIFIC COAST SOFT.
PCS (80 Col BD, Word Proc, D.Base, Spreadsheet) CALL
Account PAC (C/D) ... $34
File PAC (D) $30
Editor PAC (D) $39
Inquire PAC (D) $57
Happy Tutor Typng (D) $18

PROFESS. SOFTWARE
Wordpro 3 + /64 (D) ... $68

QUICK BROWN FOX
Prof.Word Proc. (R) ... $50

RAINBOW
Writers Assistant $95
Spreadsheet Assist. .. $95
File Assistant $95

SIRIUS
Blade/Blackpoodle (D) $27
Type Attack (D) $27
Repton (D) $27
Critical Mass (D) $27
Snake Byte (D) $23
Way Out (D) $27
Fast Eddie (D) $23
Turmoil (D) $23
Spider City (D) $27
Squish'Em (D) $23
Final Orbit (D) $27
Alpha Shield (D) $27

SKYLES ELEC. WORKS
Busicalc (C/D) $52
Busiwriter (D) $72

SPINNAKER
Snooper Troops 1 (D) . $29
Facemaker (D) $23
Kindercomp (D) $20
Hey Diddle (D) $20
Most Amaz. Thing (D) . $27

SYNAPSE
Fort Apocalypse (C/D) $23
Survivor (C/D) $23
Drelbs (C/D) $23
Pharoh's Curse (C/D) . $23
Protector II (D) $23
Morgal (D) $23
Shamus (D) $23

TAYLORMADE
Touch Typing Tutor 3.0 (D) $21

TIMEWORKS
Rbbrs/Lost Tomb (C/D) $21
Wall Street (C/D) $21
Money Manager (C/D) . $21
Data Master (C/D) $21
Dungeons of Alg. Dragons (C/D) $21

TOTL
Text 2.6 ... (C) $32 (D) $34
Label 2.6 .. (C) $15 (D) $17
Time Manager 2.6 (C) . $24
Time Manager 2.6 (D) . $27
Resrch Assist. 2.0 (C) . $24
Resrch Assist. 2.0 (D) . $27

UMI
Motor Mania (C) $20
Renaissance (C) $27

VICTORY
Annihilator (C/D) $16
Kongo Kong (C/D) $16
Trek (C/D) $14
Adv. Pack #1 (C/D) ... $16
Adv. Pack #2 (C/D) ... $16
Grave Robbers (C/D) . $13
Chomper Man (C/D) .. $18

Questions Beginners Ask

Tom R. Halfhill, Features Editor

Are you thinking about buying a computer for the first time, but you don't know much about computers? Or maybe you just purchased a computer and are still a bit baffled. Each month in this column, COMPUTE! *will answer questions frequently asked by beginners.*

Q What is a motherboard?

A A motherboard is the main circuit board of a computer. All other boards are connected to the motherboard.

The most important component on the motherboard is the central processing unit (CPU)—the central brain of the computer. The CPU is a microprocessor chip which performs or supervises all computer operations. It fetches each program instruction one at a time, executes it, stores the result, and then fetches the next instruction.

The motherboard also contains support chips required by the CPU: usually a video chip to control the TV display; input/output chips to handle the exchange of data with such peripherals as the disk drive, tape recorder, or printer; and perhaps a sound chip for music and sound effects.

In some computers—such as the Apple, Atari 800, and IBM PC/PCjr—the motherboard has long, narrow sockets called slots into which accessory boards can be plugged. Memory boards full of RAM chips (Random Access Memory) often fit into these slots. Other accessory boards (or cards) might include operating systems, disk drive controllers, printer interfaces, direct-connect modems, 80-column video expanders, graphics expanders, and even piggyback processors (boards with another CPU to allow the computer to run different types of software). That's why motherboards with several internal slots make a computer more versatile.

Some computers, including most home computers these days, contain only one circuit board—the motherboard. All the components are contained on this main board: the CPU, support chips, RAM chips, and ROM chips (Read Only Memory). Consolidating all the boards into one motherboard makes the computer smaller, lighter, and—most important from the manufacturer's point of view—cheaper to produce. For example, original Atari 800s contain six boards, and that's even before all the slots are filled with accessory boards. But the new Atari 800XL, which replaces the 800, contains only one board, even though it has more memory (64K RAM versus 8K–48K). Obviously, the 800XL costs less to manufacture.

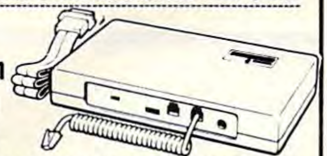
Of course, a computer without slots for accessory boards would not be as versatile. So single-board computers generally have an expansion slot or *system bus* on the rear. This allows accessory boards to be added externally. The accessory boards resemble large cartridges because they are enclosed in protective plastic or metal housings.

This still leaves one problem. How can more than one accessory board be plugged in at once? Naturally, there's a solution—an expansion box or motherboard extender. Both devices convert a lone expansion slot into several slots. For instance, you can expand a Commodore VIC-20 from the standard 5K RAM to 24K RAM by plugging a motherboard extender into the rear expansion slot, and then plugging 3K and 16K expanders into the motherboard extender.

Occasionally this is necessary even on computers with internal slots on the motherboard, such as the IBM PC. To fully equip a PC, sometimes the five internal slots just aren't enough. ©

SPECIALS on INTEGRATED CIRCUITS
6502 @ 4.90 6520 @ 4.00 6522 @ 5.00 4116 @ 1.85
2532 @ 5.90 2716 @ 4.45 6116 @ 6.45 4164 @ 6.90

Anchor Automation Signalman MODEMS



FREE SOURCE MEMBERSHIP WITH SIGNALMAN
All Signalman Modems are Direct Connect, and provide the best price-performance values. **Dealer and OEM inquiries invited**

Volksmodem with computer cable	68
Mark VII Auto Dial/Auto Answer	99
Mark XII Smart Model 1200/300	279
DC HAYES Smartmodem	219
DC Hayes Smartmodem 1200/300	519

PROM QUEEN for C64 or VIC	130
Apple Emulator for Commodore 64	Call
STAT Statistics Package for C64	95
Solid Oak 2 Level Stand for C64 or VIC	29
C64/VIC Switch (networking)	129
BACKUP V1.0 tape copier for C64 or VIC	20
CARDBOARD/6 Motherboard - VIC	64
CARDBOARD/5 Motherboard - C64	56
CARD PRINT G Printer Int. with Graphics	72
CARD PRINT B Printer Interface—C64/VIC	48
CARDBOARD/3s Motherboard - VIC	22
CARDCO C64/VIC Calculator Keypad	32
CARDRAM/16 RAM Expansion - VIC	49
Complete CARDCO Line in stock	
CIE and VIE IEEE Interfaces in stock	
MSD SuperDrive for C64 or IEEE	365
MAE Assembler for C64	50
Koala Pad Touch Tablet—C64 or VIC	75
CBC 4/12 Analog to Digital 4 chan/12 bit	179
MULTIPLAN for C64	79
Dust Cover for C64 or VIC	6
Grand Master Chess for C64	24
COMAL Language for C64 with sprites, color graphics, sound, turtle graphics.	14
BusCard II from Batteries Included	159
ULTRA BASIC – 64 with Turtle Graphics	37
Super Disk Utility – C64 – includes backup	19
MicroChess – C64 – 8 levels of play	17
HES MODEM with software for C64	45
Commodore 64 Programmers Reference Guide	16
WordPro 3+/64 with Spellright	85
VIController (also C64) – BSR Controller	50
COM VOICE Synthesizer for C64 or VIC	139
VIC products in stock – call for extra discounts.	
Victory Software for VIC and C64 in stock.	

APPLE—FRANKLIN ITEMS

FRANKLIN—complete line in stock	
QUENTIN Drives for Apple/Franklin	189
Swapper Stopper automatic switch between paddles and joystick	26
KRAFT Apple Joystick	40
Kraft Apple Paddle Pair	30
Koala Pad Touch Tablet-Apple/Franklin	90
SPINNAKER Software in stock	
Broderbund Software in stock	
16K RAM Card for Apple	59
Multiplan—Microsoft	185
Solid Oak 2 Level Stand for Apple	29
Serial Card for Apple	89
MPC RAM/80 column card for IIe (AP/TXT)	139
Z80 Softcard and CP/M (Microsoft)	235
RANA Elite I with Controller	389
Parallel Printer Interface/Cable	69
Microtek and MPC Interfaces in stock	
Grappler + Interface	135
DC Hayes Micromodem II, IIe with Smartcom	245
PFS: File or PFS: Report or PFS: Graph	95
Videx 80 Column Card	209
Apple Blue Book	19

commodore

See us for Personal, Business, and Educational requirements. Educational Discounts available.

PETSCAN I — $245 base price

Allows you to connect up to 30 CBM/PET Computers to shared disk drives and printers. Completely transparent to the user. Perfect for schools or multiple word processing configurations. Base configuration supports 2 computers. Additional computer hookups $100 each.

COMPACK/STCP — $115

Intelligent Terminal Package for PET, CBM, C64
Includes ACIA Hardware / STCP Software

SCREENMAKER 80 Column Adapter for C64 Provide big screen capability for business applications.	139
Copy-Writer Word Processor for C64 Full-featured package with 800 lines of text in memory. Includes double column printing, graphic capability, full printer support.	49
Special Screenmaker/Copy-Writer Combo	179

VICTORY Software for VIC and C64

Metamorphosis	16	Creator's Revenge	16
Labyrinth of Creator	16	Galactic Conquest	16
Kongo Kong	16	Annihilator	16
Chomper Man	16	Grave Robbers	13
Bounty Hunter	16	Adventure Pack I or II	16

PAPER CLIP Word Processor – CBM/C64	60
ORACLE Data Base from Batteries Included	89
SPINNAKER Software C64, Apple, IBM, Atari	
Compute!'s First Book of PET/CBM	11
POWER ROM Utilities for PET/CBM	78
WordPro 4+ - 8032, disk, printer	285
VISICALC for PET, ATARI, or Apple	189
Compute's First Book of 64 Sound & Graphics	11
SM-KIT enhanced PET/CBM ROM Utilities	40
PET Spacemaker II ROM Switch	36
Compute's First Book of Games	11
Dust Cover for PET, CBM, 4040, or 8050	8
CmC Interfaces (ADA1800, ADA1450, SADI in stock)	
Compute!'s Reference Guide to 64 Graphics	11
Compute!'s Machine Language for Beginners	11
HES Software and Hardware in stock	
UMI products in stock	

FlexFile for PET/CBM/C64 — $59

DataBase, Report Writer with calculations, Mailing Lists. Easy to use, and can be modified.

FORTH for PET/C64 full FIG model – Cargile/Riley — includes all FORTH 79 Standard extensions, structured 6502 assembler with nested decision macros, standard 16x64 screens, ability to read/write BASIC sequential files, sample programs, introductory + reference manual.	50
Metacompiler for FORTH for independent object code	30
Floating Point for FORTH	20
KMMM PASCAL IV for PET/CBM/C64 Virtually full Jensen-Wirth implementation is now suitable for advanced placement courses.	99
EARL for PET/CBM Disk-based ASSEMBLER	65
SuperGraphics – BASIC Language Extensions Fast graphics, sound, turtle graphics routines for PET/CBM.	45
RAM/ROM for PET/CBM	4K $75 8K $90
COMAL Language for C64, CBM, PET	14
CBM Public Domain Software – C64 27 disks	75
STAT for PET/CBM/C64 and Apple — **Comprehensive Statistical Analysis Routines** Includes complete file handling capabilities, summary statistics, confidence intervals, hypothesis tests, exponential mean tests, multiple and power series regression, analysis of variance, histograms, and non-parametric tests.	95
PageMate 60 Command Word Processor Full-featured package for all Commodore computers. Full screen editing, and supports disk, tape, and all printers.	20

DISK SPECIALS

Scotch (3M) 5" ss/dd	10/ 2.10	50/ 1.90	100/ 1.86
Scotch (3M) 5" ds/dd	10/ 2.65	50/ 2.45	100/ 2.40
Scotch (3M) 8" ss/sd	10/ 2.20	50/ 2.00	100/ 1.98
Scotch (3M) 8" ss/dd	10/ 2.80	50/ 2.50	100/ 2.47

We stock VERBATIM DISKS

Write for Dealer and OEM prices.

Sentinal 5" ss/dd	10/ 1.80	50/ 1.75	100/ 1.65
Sentinal 5" ds/dd	10/ 2.40	50/ 2.35	100/ 2.25

We stock Dysan disks

Wabash 5" ss/sd	10/ 1.50	50/ 1.45	100/ 1.40
Wabash 5" ss/dd	10/ 1.80	50/ 1.75	100/ 1.65
Wabash 5" ds/dd	10/ 2.50	50/ 2.45	100/ 2.35

We stock MAXELL DISKS

Write for dealer and OEM prices.

Disk Storage Pages 10 for $4 Hub Rings 50 for $6
Disk Library Cases 8"—3.00 5"—2.25
Head Disk Cleaning Kits 12
AMARAY Disk Storage Systems in stock.
Innovative Concepts FLIP 'N' FILES in stock.

CASSETTE TAPES—AGFA PE-611 PREMIUM

C-10	10/ .61	50/ .58	100/ .50
C-30	10/ .85	50/ .82	100/ .70

ZENITH data systems

ZVM-122A	99	ZVM-123G	89
ZVM-131	300	ZVM-135	490

Z100 16-bit/8-bit System	CALL
Z29 Terminal (DEC and ADM compatible)	680
Z-150 IBM PC COMPATIBLE	CALL
Z-160 PORTABLE PC	CALL
We stock entire Zenith line.	

USI Video Monitors – Green or AMBER 20 MHz hi-res
Dealer and OEM inquiries invited

WRITE FOR IBM PC COMPATIBLE PRICES	
MultiPlan—IBM or Apple	185
Quadboard for IBM available	
KOALA PAD Touch Tablets—Apple, Atari, IBM, CBM	
Peachtext 5000 Software Package	209
PFS Software for IBM and Apple in stock	
SPINNAKER Software C64/VIC, Apple, IBM, Atari	
VOTRAX Personal Speech System	269
BMC 9191 Color Monitor	229
BMC 12A 12" Green Monitor	79
Dynax (Brother) DX-15 Daisy Wheel Printer	469
Brother HR-25 Daisy Wheel Printer (25 cps)	769
Itoh Prowriter Parallel Printer	379
Panasonic 1090 Printer with Correspondence Mode	279
Gemini 10X	289
EPSON, Okidata, Star Micronics printers in stock	
USI CompuMOD 4 R F Modulator	29
We Stock AMDEK Monitors	
A P Products	15% OFF
COMPUTER COVERUPS IN STOCK	
BROOKS 6 Outlet Surge Suppressor/Noise Filter	54
Surge Suppressor-6 outlet	29
Electrohome 1302-2 13" Hi-res RGB Monitor	335
Panasonic 12" Monitor (20 MHz) with audio	135
Synertek SYM-1 Microcomputer	189

Hewlett Packard

Write or call for prices.

DATASHIELD BACKUP POWER SOURCE — $265
Battery back up Uninterruptible Power Supply with surge and noise filtering. The answer to your power problems.

ATARI – WE STOCK ENTIRE LINE
SPINNAKER and Broderbund Software in stock.

215-822-7727
252 Bethlehem Pike
Colmar, PA 18915

A B Computers

WRITE FOR CATALOG. Add $1.50 per order for United Parcel. We pay balance of UPS surface shipping charges on all prepaid orders (add extra for mail, APO/FPO, air). Prices include cash discount. Regular prices slightly higher. Prices subject to change.

David D. Thornburg, Associate Editor

Computers In The Workplace

I can't remember the exact occasion, but about three years ago my son (who was then seven years old) was being taken to lunch by a friend of mine in downtown Palo Alto. As the two of them walked down the street, my boy looked in the window of an office where he saw a woman typing some correspondence. "What is she doing?" my son asked. "She is typing a letter," my friend replied. At that, my son looked again and said, "That's funny; I thought only men typed."

I thought it was pretty funny too—for a while. My son knows that I spend a lot of time at a keyboard, much of it writing articles and books. In fact, I am happy he sees that keyboards are not the sole domain of female typists, but are becoming increasingly used by men. But any stereotype is dangerous; it is as dangerous for my son to think of men as typists as it is for women to be typecast in that role.

A Difference In Use

As I thought about the incident some more, it became apparent that there was perhaps a distinction in the ways that keyboards were being used by men and women, especially in business. In most businesses it appears that male keyboard users are using spreadsheet programs, or performing other analytical or forecasting activities with computers, while the majority of women employees are using keyboards connected to nothing more sophisticated (or career-enhancing) than an electric typewriter. In general, it appears that men compute and women type.

Because those who compute tend to earn more than those who type, it is worth exploring the potential of the business computer in eliminating sex-stereotyped jobs. I refer to sex stereotypes rather than discrimination because, as we shall see, a good portion of the job-selection process is induced by the very people who end up perpetuating the stereotype of women as typists.

No Access To The Professions

It is one of my pleasures to spend part of my time as a teacher. Sometimes my students range from third to sixth grade, and other times they are first-year graduate students in product design. In my graduate classes, I will often have only four or five women among my 40 students. Since product design is among the more "artsy" of the engineering fields, you would expect this number to be higher (assuming that you believe women are more interested in the arts than men).

In fact, I find it quite disappointing that there's such a small percentage of women. But the reasons for it are not hard to discern. In order to gain entrance to graduate school in an engineering field, students must have majored in engineering or the physical sciences in college. This, of course, requires a very solid background in mathematics.

As I look at the younger children I sometimes work with, I find that many of the girls are turned off to mathematics by the time they reach fourth grade, and that those who are not turned off have spent time with teachers who have a deep love and understanding of mathematics themselves. The mathphobia that sets in at an early age has a significant destructive power.

To allow any group to consider itself incapable of mastering mathematics is to essentially deny that group access to the professions. For whatever reasons, most of the high-paying technical, business, and medical professions require a significant number of advanced mathematics courses in col-

David Thornburg is an author and speaker who has been heavily involved with the personal computer field since 1978. His main interest is in making computers responsive to people's needs. He is the inventor of the KoalaPad graphics tablet and is the author of nine books about programming. His recent series Computer Art and Animation *(Addison-Wesley) includes four books on Logo for the Atari, Commodore, Radio Shack and TI computers.* Discovering Apple Logo *(Addison-Wesley) shows how Logo can be used as a tool for exploring the art and pattern of nature. He has been called "an enthusiastic advocate for a humanistic computer revolution," and his editorial opinions have appeared in* COMPUTE! *since its inception.*

lege. By allowing some of our youngsters to become math illiterate, we are confining them to the lower end of the wage scale years before they seek their first jobs.

Working In A Man's Field

Unfortunately, mathematics is generally considered a man's field. In an attempt to counter this perception, Teri Perl wrote a book several years ago that should be on the shelves of every bookstore in the nation. This book is *Math Equals* (Addison-Wesley), a brief history of women in mathematics. Rather than presenting a dry historical treatise, Teri Perl portrays the women of her study as complete human beings and talks about their frustrations of being good in a man's field when they were expected instead to tend to matters of the home.

Of all the people who should read this book, among the most important would be the teachers of grammar school who pass on their own frustration and fear of mathematics to their female students, who in turn embrace them as their own.

But what does mathphobia have to do with men using computers while women type? The answer can be found in a myth that is as wrong as the belief that women aren't good at mathematics—that you need to be good at math in order to use computers. I would venture a guess that many of you are "good at computers," but are probably not "good at math." You already know that mathematics is not a prime requisite for computer literacy. And yet you are viewing the problem from the other side of the bridge—you have already made the passage.

Reinforcing The Myth

Imagine the plight of the woman with a degree in the arts or the humanities who wants to find a job in business. When offered an opportunity to learn about computers, many women say, "Oh, I couldn't learn how to use computers, I never was good at math"; or "I never was good at technical subjects." By making statements of this sort, these women are removing themselves from career paths that lead to high-paying jobs.

Because these fears are, in fact, unfounded, those who express them are allowing the persistence of a myth to restrict their professional growth.

While I don't know a sure-fire way to break through to people who hold themselves back in this way, two authors have done a marvelous job in trying to show working women the road to computer confidence and higher-paying jobs.

These authors are Dorothy Heller and June Bower, and their book is *Computer Confidence—A Woman's Guide,* published by Acropolis Books ($9.95 paperback). Because of the timeliness of its topic and its lucid style, this book deserves a wide readership. You could do your community a favor by seeing that your local bookstore has plenty of copies in stock.

A Highly Personal Book

As women who entered the computer field from backgrounds in the humanities, the authors have the rare perspective of those who have walked both sides of the street. The book is a highly personal account; in fact, it is the book they wish they had had (but couldn't find) when they entered the computer field. Topics range from a short history of women who "made it big" in computers, to case histories of working women who use computers without knowing how to solve partial differential equations. By blending case histories with enough technical data to make the reader a savvy shopper for computer technology the authors prepare the reader for the main goal of the book: to show women how they can enter career paths with unlimited upward potential.

This assistance covers the spectrum from worksheets to help the reader identify appropriate career choices, to practical tips on how to handle job interviews, and especially how to handle the inevitable objections that arise when the interviewer finds that the educational and work background of the applicant doesn't include the "right" degrees from the "right" schools. ©

Software Unlimited

KOALA SALE!

Koala Pads
Atari, C-64, Vic-20 w/Disk Software *$65.00*
Atari, C-64 w/ Cartridge Software *$73.00*
Apple II, IIe, II+ *$88.00*
IBM P.C. *$108.00*
Instant Programmers Guide All Computers *$12.00*

THE Clone Machine
with "Unguard" C-64 .... *$32.00*
S.A.M., A, C, Disk *$39.95*
Chatterbee A, C, Disk *$27.00*

SOFTWARE UNLIMITED, 27 LOOKOVER LN., YARDLEY PA 19067. VISA & MC ADD 4%. PLEASE ADD $1.50 P&H. PERSONAL CHECKS REQUIRE 2 WEEKS TO CLEAR. CALL FOR A COMPLETE LIST OF OUR SOFT & HARDWARE.
(215) 493-1372
OUT OF STATE 1-800-225-3656

SAVE BIG ON COMPUTER PRODUCTS

MODEMS

HAYES

Smartmodem 300	209
Smartmodem 1200	489
Smartmodem 1200B	425
Micromodem II E	240
Smartcom II Software (for IBM PC)	90

NOVATION

J. Cat 300B direct	99
103 SmartCat 300B Smart	159
103/212 Smart 300/1200 B	379
AppleCat II 300 baud for Apple	200
212 AppleCat 300/1200B for Apple	390
Access 1-2-3 1200B for IBM	379

MONITORS

GORILLA

12" Green	88.00
12" Amber	94.00

TAXAN

12" KG-12N (green)	120
12" KG-12N-UY (amber)	125
RGB vision III	499

AMDEK

V300G 12" green	130
V300A 12" amber	145
V310A 12" amber (for IBM)	160
Color I+ 13" composite	280
Color II 13" RGB	425
Color IV 13" RGB analog	740

Monitor Cables

PA 400 for TI-99/4A or Commodore	15

QUADRAM

MICROFAZER

QRMP-8 Par/Par	135
QRMSP-8 Ser/Par	145
QRMMSS-8 Ser/Ser	145
QRMPS-8 Par/Ser	145

QUADBOARD (for IBM PC)

QR5064 64K	270
QR4064 64K	270

QUADLINK

QR 3000 for IBM	475
QR 3010 for Compaq	475
QR 3020 for Columbia	475
QR 8201 Quadcolor-1	200
QR 8202 Quadcolor-2 (upgrade kit)	200
e Ram 80 — 80 col. card for Apple IIe	115

wabash maxell Dysan

DISKETTES

10 — 5¼" Floppy Diskettes (packed with a storage box)

	SS /SD	SS / DD	DS / DD
Wabash	$19	$22.50	$30
Maxell	NA	27.00	39
Dysan	NA	27.00	39
10 — 8" floppy diskettes			
Maxell	NA	36.00	42

PRINTERS

Tremendous Discounts

Gemini 10X 275
Gemini 15X 400
Delta 10 390
Epson RX80 275
Other Epson models . . . CALL
Panasonic 1090 . 275
Most models IN STOCK of DIABLO • TTX MANNESMANN TALLY CALL FOR PRICES

HEWLETT-PACKARD CALCULATORS

HP-11C	70	HP-15C	90
HP-12C	90	HP-16C	90
HP-41C	145	HP-41CX	245
HP-41CV	200	HP-97	560

all software & accessories too

PORTABLE COMPUTERS

HP-71B	399	HP-75	719

PERSONAL COMPUTATION DEALER

CABLES — INTERFACES

accessories for Computer Printers

GRAPPLER PLUS	120
16K BUFFERED GRAPPLER	165
Apple Dumpling	70
Cardco G	65
10 ft. Par. Cable for IBM	32
10 ft. 36x36 Parallel	32
10 ft. 25x25 RS-232	25
Y Cable (for TI-99/4A)	25
6 ft. TI-99/4A parallel cable	25
13x16 printer stand for 80 col. printers)	15
16x22 printer stand (heavy duty) for wide carriage printers	40

RIBBONS, PAPER

Paper — 1000 Sheet Pack

9½ x 11 white, 20 lb.	12
9½ x 11 green bar, 20 lb.	12
9½ x 11 trim edge, 20 lb.	15
14⅞ x 11 green bar, 20 lb.	19

Ribbon Cartridges

for Epson 80 col.	4
for Epson 132 col.	7
for Mann. Tally 160	11
for Mann. Tally 180	14
for M.T. Spirit 80	7.50
for Panasonic 1090	6.50
for Diablo Hytype II	5
for C. Itoh Starwriter	5

Twin spool ribbons for Gemini 10/10X, 15/15X; Okidata 80, 82, 83
6 for 15.00 or 12 for 24.00

CALL TOLL FREE 800-621-1269 EXCEPT Illinois, Alaska, Hawaii

Corp. Accts. Invited. Min. Ord. $15.00. Mastercard or Visa by mail or phone. Mail Cashier's Check, Mon. Ord., Pers. Check (2 wks. to clr.) Add $4.00 1st item. (**AK, HI, P.R., Canada add $10.00 first item**) $1.00 ea. add'l shpg. & handl. Shipments to IL address add 7% tax. Prices subj. to change. **WRITE for free catalog. RETURN POLICY: Defectives Only: Most products replaced within 30 days of purchase with identical merchandise only. Computer and large peripherals replaced only when defective on arrival (within 3 work days of purchase). Other problems covered by mfr. warranty. ALL ELEK-TEK MERCHANDISE IS BRAND NEW, FIRST QUALITY AND COMPLETE.**

ELEK-TEK, inc. 6557 N. Lincoln Ave., Chicago, IL 60645
(312) 631-7800 (312) 677-7660

SAVE MORE THAN EVER ON... 3M Scotch® DISKETTES AND OTHER COMPUTER NEEDS!

NEW! Lower Prices

3M Scotch® Diskettes

are boxed in 10's with labels, envelopes and reinforced hubs on 5¼" diskettes.

LIFETIME WARRANTY!

$1.85 ea. 5¼" SSDD (744)
Qty. 20

5¼" DSDD (745) **$2.35** ea.
Qty. 20

5¼" SSQD-96TPI (746) **$2.60** ea.
5¼" DSQD-96TPI (747) **$3.25** ea.
8" SSSD (740) **$2.05** ea.
8" SSDD (741) **$2.50** ea.
8" DSDD (743) **$3.10** ea.

DISK CADDIES

...the original flip-up disk holder for 10 diskettes. Beige or grey only.

5¼" **$1.65** ea. + .20 Shpng. 8" **$2.29** ea. + .20 Shpng.

RIBBONS!

at bargain prices.

EPSON MX-70/80	**$3.58** ea. + .25 Shpng.
EPSON MX-100	**$6.99** ea. + .25 Shpng.
Okidata Micro 84	**$3.66** ea. + .25 Shpng.
Diablo 630 Mylar	**$2.60** ea. + .25 Shpng.
Diablo 630 Nylon	**$2.93** ea. + .25 Shpng.

DISKETTE 70 STORAGE

The best buy we've seen! Dust-free storage for 70 5¼" diskettes with six dividers.

$14.95 + $3.00 Shpng.

FLIP 'N FILE

Redesigned for better appearance and greater ease of use. FLIP 'N FILE 25 holds 25 5¼" diskettes; FLIP 'N FILE 50 holds 50 5¼" diskettes.

FLIP 'N FILE 50
Retail $39.95
...DW Price
$26.95
+$5.00 Shpng.

WE WILL BEAT ANY NATIONALLY ADVERTISED PRICE!
Nationwide: 1-800-621-6827
Illinois: 1-312-944-2788
Hours: 9 AM-5 PM Central Time

Minimum Order: $35.00

Shipping: 5¼" DISKETTES—Add $3.00 per 100 or fraction thereof. 8" DISKETTES—Add $4.00 per 100 or fraction thereof. OTHER ITEMS: Shipping charges as shown in addition to diskette shipping charges. **Payment:** VISA or MasterCard. COD orders only, add $3.00. **Taxes:** Illinois customers, please add 8%.

DISK WORLD!

SUITE 4806
30 EAST HURON STREET
CHICAGO, ILLINOIS 60611

Authorized Distributor Information Processing Products 3M

The Morning After:
Anti-Computer Backlash And The Arrival Of The Mass-Market Home Computer
Part 1

This is the text of the speech Fred delivered at the West Coast Computer Faire in late March. We are printing the speech in two parts.

We are at a watershed in home computing. The watershed has been caused by the computer price wars of 1983, the introduction of simple and inexpensive, yet powerful, new computer programs and peripherals, and the entry of IBM into the home computer market.

Over the next year, home computing users, vendors, and enthusiasts will divide into two major camps: the *computer intimates* and the *computer literates*. By the end of 1986 these two groups will have fused into a third camp: the *neo-programmers,* who will represent the bulk of the users of home computers through the next decade.

Literates Vs. Intimates

Hackers, computer professionals, old-line computer educators, programming teenagers, and computer hobbyists will make up the bulk of computer *literates*. Computer literates will stress the importance of learning how to program and learning how computers work. The computer itself will continue to be the prime concern of this group.

Computer *intimates* will far outnumber the computer literates. Computer intimates will consist of all the millions of Americans who were roped or forced into using computers and who demand that they be easier to use and more practical.

Computer intimates will believe that software and computer input devices are far more important than the computer itself. As a group they will preach ignorance of computer programming and ignorance of the computer's insides as virtues. The motto of the computer intimates will be: "You don't have to know how a computer works, only how to make it do work for *you*."

The Computer Freight Train

On December 6, 1983, I appeared on ABC's *Good Morning America* TV show as a computer expert. My task was to advise families on the type of computer they should purchase for Christmas. In less than seven and a half minutes I led the show's viewers and its two hosts, David Hartman and Joan Lunden, through a bewildering array of computer hardware and computer programs.

I am sure that when the segment was over, most viewers still couldn't tell the difference between a disk drive, a program recorder, or a touch pad. But I'll wager that they did have a better feeling for the risk involved in investing in a personal computer, for the daunting complexity of becoming a first-time user, and for the flood of computer products and the dearth of reliable guidelines for making a purchase.

"Most consumers see personal computers as a high-speed freight train," I told viewers. "They feel they have to take the risk of hopping on now,

or they feel they will be run over or left behind."

The Hottest Thing Under The Christmas Tree

More computers were sold as Christmas gifts this year than in any year prior to 1983. By early 1984 over eight million Americans had personal computers.

Unfortunately, soon after Christmas, many of these Americans began suffering from "morning after" regrets and resentments. Too many Americans who had seen the slick commercials on TV and who had heard the daily press reports about the computer revolution were now wondering what they had gotten themselves into.

Most Americans have heard the word *software* but have only a vague idea what the word means. They have no understanding of what comprises a "complete" computer system. They have no appreciation of what operating or programming a computer entails.

Most Americans don't even know how to hook up a computer's cables, plug it in, or turn it off. I know of one family who finally turned their computer off at one in the morning, but who only did so after hours of agonized, fruitless searching of the manual. They were afraid they might break the computer if they turned it off the wrong way.

The Computer Kit

Why do people buy computers? Most Americans buy computers out of curiosity, for their work, to play games, or as an educational aid and tool for their children.

Most Americans buy computers at bargain-basement prices, usually at discount houses. Most Americans get their basic knowledge about computers from news stories and TV commercials.

When a person buys a computer, he thinks he has bought something equivalent to what he has seen on TV. He expects his computer to be able to do roughly the same things as the TV computer.

The average new-computer purchaser brings his computer home, struggles with the manuals, cables, and plugs, and finally powers the computer up. After all this effort, what does he get?

A blank screen.

After still more struggling with his manual, the astute newcomer finally realizes that what he has bought is a *kit*—like a bicycle or a puzzle that comes in a million pieces. Only it's worse. The kit's pieces are invisible. You don't get to see them until they appear on the computer's display screen after you have typed them in at the keyboard.

The pieces, of course, are the commands in the computer's BASIC programming language. Computer commands are more difficult to use than puzzle pieces for two reasons. First, puzzle pieces are combined in some sort of visual order to make up a picture. Second, pieces in a puzzle can usually be combined in only one way. And the picture fragment on each piece is a clue to where the piece belongs.

But computer commands are different. They carry no picture fragment that helps you see where in a picture (or a program) they belong. And they can be combined in an infinite number of ways. There is no set order to reach any given solution.

Most kits—for a bicycle, a lawn chair, a toaster oven, a sandbox, or swing set—come with explicit, printed directions. Computer kits don't usually come with printed directions. Instead, they come with a *dictionary* of commands organized, alphabetically, from A to Z. You get all the building blocks, but little or no help in how to put them together. And, before long, you realize, with a sinking feeling, that they can be put together in a million ways.

But where do you start?

Buying Half A Computer

It finally dawns on the consumer that what he has bought is only *half a computer*. Until he buys some software and some more equipment—a program recorder or disk drive, cassettes, disks, cartridges, and a printer—he can't do anything useful.

Of course this isn't exactly true. He can always assemble the kit himself. There are dozens of magazines and hundreds of books with pre-recorded programs for his kind of computer. All he has to do is follow the blueprints—the listings—in the books and magazines, and soon he will be the proud owner of a real computer.

Of course he will need to spend dozens of hours entering in the programs, and dozens of hours more poring over the listings, trying to figure out why his programs don't work.

And he will have to invest in a storage device, so he can save his delicate, precious programs.

And he still needs a printer if he plans to use the computer as an electronic typewriter, bookkeeper, or filing cabinet, the three most popular home computer applications.

Voting No To The Home Computer

After the average consumer has forked over from $50 to $300, is he likely to invest another $100 to $1000 for additional hardware and software to "finish off" his computer?

After the consumer has made his purchase and found that he has only half a computer, is he likely to feel positively toward computers and computer companies?

After the average consumer has realized that he has bought a kit, is he likely to roll up his sleeves, master a programming language, or patiently enter in hundreds of lines of unintelligible commands?

The answer to all these questions, *for the average consumer*, is no.

The After-Christmas Backlash

Under these circumstances, the average person who bought or received a computer for Christmas is not likely to become a computer enthusiast. Instead, he is likely to become part of a growing anticomputer backlash.

More and more individuals and groups in society are coming to the conclusion that personal computers have not lived up to their promise. At the very least, they have not lived up to their commercials.

These individuals and groups are becoming more organized and outspoken. Like me, they see personal computers as a high-speed freight train, and they are set on derailing that train.

The other night I was listening to National Public Radio's "All Things Considered." A so-called computer expert was on the show decrying the use of computers in education. In his opinion, most people were using computers as fancy, expensive, electronic flash cards. He warned American parents and teachers that the computer industry was deceiving them in a major way.

Two nights later I read in *USA Today* that the American Academy of Pediatricians was warning against using computers with small children. The Academy reaffirmed its decade-old statement that "Advertising that promotes ... learning environments, programs, or systems is often guilt-producing, misleading and potentially destructive of human development and values." The Academy scolded parents who create a "superbaby syndrome" in which parents buy computers for small children and enroll them in computer classes even before they are toilet-trained.

Fighting Back

The American public has been dazzled by the glamour and high-tech chic of personal computers. On the surface, the public's attitude toward computers seems to have undergone a dramatic change. On the surface, it appears that most Americans approve of computers, if not for themselves, at least for their children. And even if they don't approve of them, they see them as inevitable.

This is, indeed, how Americans feel—*on the surface*. But what is going on beneath the surface?

I submit that the public's current attitude toward computers is superficial and can easily be changed. I further submit that the situation is becoming increasingly ripe for public opinion to take a swing in the opposite direction. This swing may be dramatic and quick.

The American public has been put on the defensive by the rapid spread of personal computers. But the public is likely to regain the offensive at the first opportunity. Beneath the thin veneer of approval lurk people's old prejudices and stereotypes against computers. These prejudices and stereotypes are fortified and aggravated by the bad experiences millions of people are having, firsthand, with computers.

The American public just needs a champion. As soon as groups and individuals appear who can articulate the public's feelings against computers, the public will rally around them. And then a major backlash against computers will begin.

A Consumer Uprising

People who are alienated by computers are not ignorant Luddites who oppose computers just because they are new and different.

Many people already oppose computers out of ignorance and prejudice. But many more may soon oppose computers because they feel computers have been misrepresented and oversold.

An anticomputer backlash may be in the cards. If so, it should not be viewed by those of us in the computer industry as an ignorant neo-Luddite rebellion. We should see it for what it is: a legitimate uprising by irate, unhappy consumers. ©

J. B. Shelton and Glenn M. Kleiman

Ready-to-Run Magazines

We met our first personal computer, an 8K PET, back in 1978. Soon thereafter we purchased one of the "new" PETs—a-state-of-the-art machine with 16K RAM memory, a full-size keyboard and a cassette recorder for external memory.

In those long gone days of almost six years ago, we eagerly sought information about our new machine, but little was available. It came with very little documentation, and what was provided was barely understandable. Today almost every bookstore has a large selection of computer books and even some drugstores carry computer magazines, but no books or magazines were readily available back then.

One source of valuable information was *Cursor* magazine, published by Ron Jeffries. Not a traditional magazine, *Cursor* arrived, somewhat irregularly, on a cassette tape. Each issue contained six programs that we could load and run right away. The programs were a mix of graphics and sound demonstrations, games, puzzles, programming utilities, educational programs, and simple applications programs (for example, for calculating mortgage rates). All the programs were at least reasonable; some were true gems.

A First Look

The programs in *Cursor* magazine gave us our first sense of the potential uses of personal computers. In addition, we could list and analyze the programs to learn new programming techniques. *Cursor* also has claim to being the all-time best buy in the personal computer industry. The price of a six-issue subscription was originally $20.

Cursor magazine continued publishing through May 1982. Copies of all 30 back issues are still available, and some of the programs have been made available for the Commodore 64. Another early cassette magazine for TRS-80 computers, *CLOAD*, continues to publish and is now available on disk also.

The idea of "magazines" of ready-to-run programs has grown. Two new magazines on disk have recently appeared, both focusing on education about and with computers. In this column, we review and compare *Microzine* and *Window*. Our reviews are based on the first three issues of *Microzine* and the second and third issues of *Window*. Both magazines are now available for Apple computers, and versions for other computers are being developed.

Dr. Glenn M. Kleiman is an educational psychologist and software developer. He is the author of Brave New Schools: How Computers Can Change Education *(Reston/Prentice Hall) and the designer of* Square Pairs, *an educational game program (Scholastic, Inc.).*

Microzine, Captivating For Children

Microzine, published five times a year by Scholastic, Inc., is designed for children ages 10 and up. Each issue contains four programs and a 48-page printed manual that supplements the onscreen instructions and provides additional ideas for using some of the programs.

One of the four programs in each issue is a Twistaplot story. These are stories in which the plot details and outcome are controlled by decisions the reader makes. For example, one issue contains a crime-solving adventure called "Mystery at Pinecrest Manor." This is an old-fashioned whodunit which makes the reader an active participant in the story. As the reader and participant, you study files containing background information about each of the suspects, search for clues, and spy on suspects. You play the part of a character in the story, deciding where to go and what to do at each choice point. You can reread the story many times, changing your responses and thereby encountering different events and outcomes each time.

The flexibility of the stories, excellent graphics, and the active role played by the reader make Twistaplots captivating for children. Interactive stories are an exciting new genre of fiction, and Twistaplots demonstrate some of the advantages of using computers to present these stories.

Educational Programs

Each *Microzine* also contains one or two computer *tool* programs. These provide a means for children to explore and learn about different uses of computers.

A Poster program provides a simple computer language for creating colorful, low-resolution

CHART YOUR PROGRESS

SPECIALS OF THE MONTH

Atari 600XL	$189
Atari 800XL	$299
Atari 1030 Modem	$115
Rana 1000 Disk Drive	$339
1027 Letter Quality Printer	$299
Commodore SX64 Portable Computer	$959
Commodore Automodem	$ 95
Commodore Graphics Printer	$269
Verbatim Twin Pak	$ 5.25
Elephant Disks S/S	$19.50

HARDWARE SPECIALS

Printers

Leading Edge Gorilla	$189
NEC 8023A	$459
Okidata ML 82A	Call
Okidata ML 83A	Call
Okidata ML 84P	Call
Okidata ML 92P	Call
C. ITOH 8510 Prowriter	$399
C. ITOH 8510 SP	$665
Mannesmann Tally 160L	$629
Gemini 10X	$315
Mannesmann Tally Spirit	$345
Alphacom 81	$189

Monitors

Leading Edge Green Hi-Res 12"	$ 89
USI Amber	$169
NEC Green Screen	$169
Leading Edge Amber Hi-Res 12"	$ 89
Amdek Color 1+	$315

RECREATION

Q Bert/CRT	$ 35
Frogger/CRT	$ 35
Popeye/CRT	$ 35
Witness	$ 36
Planetfall	$ 36
Deadline	$ 36
Zork I, II, III	ea. $ 29
Enchanter	$ 36
Infidel	$ 36
Temple of Apshai	$ 27
Pitstop	$ 27
Starleague Baseball	$ 23
Starleague Football	$ 23
Blue Max	$ 23
Joust	$ 43
Robotron 2084	$ 33
Pole Position	$ 36
Ms. Pac Man	$ 36
B-1 Nuclear Bomber (C)	$ 12
Midway Campaign (C)	$ 12
Nukewar (C)	$ 12
Telengard (C)	$ 16
Jumpman Jr./CRT	$ 27
Pitstop/CRT	$ 27
Flight Simulator II/D	$ 39
Night Mission Pinball	$ 20
Sammy Light Foot (CRT)	$ 20
Apple Cider Spider	$ 23

EDUCATION

Atari

Harcourt Brace SAT/D	$59
Chaterbee/D	$29
Match Wits/D	$22
Early Games/D	$22
Early Games Music/D	$22
Spellicopter	$29
Creature Creator	$29
Math Maze	$29
Alien Addition/D	$25
Meteor Multiplication	$25
Compu-Read/D	$22
Compu-Math/Fractions	$29
Compu-Math/Decimals	$29
Spelling Bee Games	$29
Speed Read +/D	$45
Spider Eater/Koala	$22
Bumble Bee/Learning Co.	$29
Snooper Troops 1, 2 ea	$32
Face Maker/CRT	$25
Kids on Keys/CRT	$25
Alphabet Zoo/CRT	$25
Kid Writer/CRT	$25
Delta Drawing/CRT	$29

Commodore

CBS Addition/Subt.	$17
CBS Multiplication/Division	$17
Speed Reader II/D	$49
Word Attack/D	$36
Mathblaster/D	$36
Spellakazan/D	$29
Crypto Cube/D	$29
Master Type/D-CRT	$29
Songwriter/D	$29
Alphabet Zoo/CRT	$22
Fraction Fever/CRT	$22
Delta Drawing/CRT	$22
Facemaker/CRT	$25
Trains/D	$29
Kidwriter/CRT	$25
Dungeons of Algebra Dragons	$17
Juggles Rainbow	$22
Bumble Bee	$29
Early Games Piece of Cake	$22
Early Games Match Maker	$22

BUSINESS

Atari

Atariwriter	$ 79
Visicalc	$179
Financial Wizard	$ 45
Letter Perfect	$ 69
Data Perfect	$ 79
Spell Perfect	$ 65
Spell Wizard	$ 45
Text/Spell Wizard Combo	$ 65
Bank Street Writer	$ 49
Bank Street Speller	$ 49
Homeword	$ 49
Syn File +	$ 65
Syn Text	$ 23
Syn Mail	$ 34
Home Accountant	$ 48
Tax Advantage	$ 39
Complete Personal Accountant	$ 65
Atspeller	$ 33
Diskette Mailing List	$ 21
Miles Payroll System	$119
Atari Accounting	$169
Family Finance	$ 36
File Manager 800+	$ 65

Commodore

Easy Script 64/D	$ 35
Easy Spell 64/D	$ 16
Easy Finance I, II ea./D	$ 16
Accts. Receivable/D	$ 35
Payroll/Checkwriting/D	$ 35
Assembler 64/D	$ 15
Logo/D	$ 49
Simon's Basic/CRT	$ 17
Bank Street Writer/D	$ 49
Home Accountant/D	$ 48
Tax Advantage	$ 45
FCM	$ 35
Paper Clip Word Processor	$ 89
Delphi's Oracle	$ 89
Practicalc 64	$ 40
Calc Result/Easy-CRT	$ 59
Calc Result/Advanced-D	$109
Multiplan/D	$ 65
General Ledger/D	$ 59
Data Manager/D	$ 17
Electronic Checkbook	$ 17
Swif tax/D	$ 35
Quick Brown Fox/D	$ 59
Quick Brown Fox/CRT	$ 56

COMPUTER OUTLET

Computer Outlet

1095 East Twain, Las Vegas, NV 89109 Mon.–Fri. 8 a.m. to 6 p.m., Sat. 9 a.m. to 5 p.m.

VISA

1-800-634-6766

Information and Inquiries: (702) 796-0296 Order Status: (702) 369-5523.

SHIPPING: For fast delivery, cashier checks, money orders or direct bank wires. Personal and company checks, allow 3 weeks to clear. C.O.D. charges: $3 minimum or 1% on orders over $300. Nevada residents add 5¾% sales tax. Shipping charges based on weight $3 minimum. APO and FPO orders: $10 minimum and 15% on all orders over $100. School and business purchase orders welcome. All returns must be accompanied by return authorization number. Call (702) 369-5523 before returning goods for replacement. Prices reflect cash discount only and are subject to change. Catalogs: 50¢ U.S., $1.00 foreign.

pictures. This program is a good introduction to both computer graphics and some rudimentary programming concepts.

An Electronic Card Filer program demonstrates how computers can be used to store, sort, and retrieve information. This program is well designed for introducing data base and information retrieval concepts, but it is limited to small amounts of information. Each card, or record, can contain only five fields of information, with up to 25 letters or numbers per field.

Another tool program, Melody Maker, is for creating music on the computer. With Melody Maker you can enter notes over a two-octave range and have the computer play your song. You can also have the computer create a visual display to go with your music. One type of display shows a musical staff and the notes; other types of displays create colorful patterns. You can save your songs on disk to play again later.

No Editing Feature

The Melody Maker program can be very useful in helping children learn about reading music. Its main drawback is that it is difficult to change a song once you have entered it. You can go back and change any note to another note, but you cannot insert or delete new notes. Therefore, if you want to insert or delete a note at the beginning, you have to reenter the entire song.

There is also a program called Amazing Robot that is intended to introduce programming concepts. As you might expect, the commands the robot follows are like those of turtle graphics. You can instruct the robot to move forward or back a number of steps, or turn left or right a number of degrees. However, this robot does not draw with a pen, as turtles do. Instead, you command it to maneuver through different mazes and patterns displayed on the screen. This aspect of Amazing Robot is similar to *Karel the Robot*, which was reviewed in this column in January 1983.

Amazing Robot does introduce some programming concepts. But we found it to be awkward to enter and edit procedures. For example, if you make a typing mistake while entering a procedure or accidentally direct the robot to touch a wall, you are thrown out of the edit mode and have to use a reedit command. Amazing Robot does not encourage learning and exploration nearly as well as more complete programs such as Scholastic's *Turtle Tracks*, Spinnaker's *Delta Drawing*, or any of the available versions of Logo.

The remaining programs include one in which you select questions to see the answers actor Robert Macnaughton gave; a tutorial and simulation game about hot air balloons; a word game; and a chase game. None of these will teach children much or draw their attention away from *Pac-Man*, *Frogger*, or whatever videogame is their current favorite.

Window Is A Screen Magazine

Window, intended for adults as well as children, takes seriously its status as a magazine using the new medium of computers. No print materials are provided, except for a note about booting the disk and accessing the help screens. Everything else you need to know is shown on the computer screen.

Window provides a great deal of flexibility. It lets you take a guided tour of each issue. This is similar to skimming through a printed magazine. You control the speed of progress through the screens and you can stop, back up, or continue at any time. You can choose to explore any program further. While working with a program you can always stop and return to skimming or to the table of contents.

Each issue of *Window* has a central theme which is the focus of a feature program, one or more other programs related to the theme, several software reviews, columns, and some smaller programs called "window dressing." The themes of the two issues we have reviewed are data base programs and music programs.

Sample Data Bases

The feature program of the data base issue is called Notebook. It allows up to 20 fields in each record, and it lets you obtain hard copies if you have a printer.

Window also provides a variety of sample data bases for you to explore and extend. Several are examples of data bases students and teachers have created. There is also a data base called *clues*. This is used in conjunction with another program called Adventurefile, which is a computer mystery. To solve the mystery, you have to use the Notebook program and the clue data base. The sample data bases provide a good starting point for novices learning about data base programs and the varied functions they can serve.

The same issue contains reviews of two software packages, Geography Search and Dueling Digits. Magazines on disk are an ideal vehicle for software reviews. Not only are the programs described and evaluated, but you also get to see actual screen displays and use interactive demonstrations of parts of the programs. These reviews gave us a much better sense of the programs than any written review ever could.

Some Fun Features

The disk also contains two games, one a variation of Monopoly and the other a variation of Simon. The games are appropriately referred to as "window dressing," as they do not add a great deal to the magazine. Finally, there is a *VisiCalc* column. This provides a template for multiplication tables, but you have to have *VisiCalc* to use it.

The feature program on the music issue of *Window* is called Mini-Songwriter. This program overlaps in function with *Microzine*'s Melody Maker, but is different in style. You enter notes by moving a marker on a piano-like keyboard displayed on the screen and specifying the length of each note. You can play your songs, varying the speed as you go. You can easily edit and save songs. *Window* also provides sample songs and another program that uses the Mini-Songwriter. This is a Mystery Melody program that presents "name that tune" riddles.

There are comprehensive reviews of MECC's *Music Theory* program, Spinnaker's *Snooper Troops*, and Earthware's *Volcanoes* program. In the reviews, you get to try a set of "which note is wrong" problems like those presented by the MECC program; search for clues as you would in the actual *Snooper Troops* program; and see the type of data you would collect in the *Volcanoes* simulation program.

The rest of the disk contains an editorial about work with computer music and Logo at MIT; a sample of music created with the *Songwriter* program (the full version of the Mini-Songwriter, available from Scarborough Software); and a graphic demonstration of sorting algorithms. These are all interesting additions to the main features. There are also columns that provide *VisiCalc* templates and Logo procedures. These columns can be used only by people who have *VisiCalc* or MIT Logo.

Comparison of *Microzine* And *Window*

Both *Microzine* and *Window* are exploring new terrain. So far, *Window* has been more innovative in its attempt to use the new medium of the computer without support of any printed materials. We had no difficulty using any of their programs with the information available on disk. We enjoyed skimming through the programs and viewing *Window*'s experiments with different formats of displaying information on the computer screen. *Window* is inventively interactive—you interact with the computer in flexible ways with several programs.

Microzine is more conservative in its approach and depends upon printed materials to provide the instructions necessary for many programs. However, the print materials also provide useful suggestions for extending the computer activities.

In their first few issues, *Microzine* and *Window* have each provided simple data base and music programs, so these programs provide a good basis for direct comparisons. The programs in both magazines are suitable for introducing novices to using computers for data bases and for creating music. However, none of the programs can replace full data base or music creating programs.

Overall, the programs in the two magazines are comparable. *Window* has an edge in the flexibility of its data base program and the ease of editing in the music program. *Microzine*'s music program has more visual display options than *Window*'s.

While we do not find major advantages in either magazine's programs, there are important differences in the overall presentations of how computers can be used for data bases and music. *Window* provides sample data bases, songs, and games that use the data base and music programs; *Microzine* does not. These extras provide good demonstrations, help people get started, and show how each program can be used in many ways. So we tend to favor *Window*'s presentations of data base and music programs.

As for the other programs, *Microzine*'s Twistaplots provide good examples of interactive fiction and contain excellent graphics. There is nothing in *Window* that is directly comparable. On the other hand, *Window* contains useful reviews of programs and ongoing columns for *VisiCalc* and Logo users.

The producers of both magazines can be expected to continue to experiment, explore, and improve. In fact, improvements are already evident within the first few issues. Our reviews and comparison should be read as a report on the status of these magazines as of the first few issues. Exciting prospects lie ahead for both, and we expect to see many more ready-to-run magazines in the near future.

Cursor *Magazine*
The Code Works
P.O. Box 6905
Santa Barbara, CA 93160

CLOAD *Magazine*
P.O. Box 1448
Santa Barbara, CA 93102

Microzine
Scholastic, Inc.
P.O. Box 641
Lyndhurst, NJ 07071

Window, *Inc.*
469 Pleasant St.
Watertown, MA 02172

©

Richard Mansfield, Senior Editor

A Random Leap

One of the enjoyable things you can do with a computer is simulate real events: things which might be too dangerous, too expensive, or too time-consuming to try in real life. The Air Force and some commercial airlines use a flight simulator so true-to-life that it can serve for all but the most advanced pilot training.

We don't have enough RAM memory, or the computation speed, or the ultrahigh resolution screens necessary to create a flight simulation of breathtaking realism. But we can try a simple simulation and get a feel for how they are programmed. The basis of the simulation will be accidental, unpredictable events created by the RND (random) command in BASIC.

Lurching Across A Bridge

Imagine a frog, lurching across a bridge. Every time he leaps, you don't know if it will be to the left or to the right. He doesn't know either. The one thing you can count on is that he will never leap straight ahead.

There are three possibilities in this game. He will either fall off the left or right side of the bridge, or safely reach the other side of the river. For this simulation, we're going to assume that the bridge is as wide as your computer screen and that the frog starts his journey midway between the left and right sides. That gives him a fair chance to make it across.

By setting up this simulation, we'll learn how to make use of the RND command as well as a way to animate characters on the screen. Let's look at the program line by line, to see what each BASIC command contributes to the overall effect. (Atari computers don't have a TAB command, so the animation technique discussed below will not work on them.)

First, we've got to define the size of the bridge, its width. Leave line 100 as it is if you have a Commodore 64 or any other computer which allows 40 characters per screen line. If you have a VIC, you should change line 100 to read: COLS =22. The VIC has 22 characters per screen line. If you have a TI, change it to: COLS=32.

The variable Y in line 110 is going to signify the position of the frog each time it leaps. If Y is raised to a higher number, the frog will appear further to the right on the screen (and be nearer the right side of the "bridge"). If Y goes down, if something is subtracted from Y, the frog moves left. At the start of the game, though, we want to put the frog in the middle between the left and right sides of the bridge so we divide COLS by 2. If you've got a 40-column screen, Y starts off equaling 20. That means that the frog is 20 from the right edge and 20 from the left—smack in the middle.

Rounding Numbers

The variable X in line 130 will tell us whether the frog should leap to the right or the left each time he leaps. This is the only complicated-looking line in the program, but it contains an important trick: the INT command. It "rounds off" a decimal number. INT (12.3) becomes 12. INT (12.7) becomes 12. Wait a minute. That's not *rounding off* as we usually think of it. 12.7 should become 13 since .7 is closer to 13 than to 12.

In fact, INT merely throws away anything to the right of the decimal point. This isn't true rounding. That's why we need to add the +.5 in line 120. By adding .5, we force a number to be rounded correctly by INT. 12.7+.5 would be 13.2 and INT (13.2) would give us the right answer: 13. Likewise, 12.3+.5 would be 12.8 and INT (12.8) would give us the correctly rounded answer: 12.

It's not important to remember *why* you need to add .5 to any number you want rounded by INT; just remember to do it. In line 120 we're not rounding off 12 or 13, all we want is an answer that tells us to go in one of two directions, to go either left or right. This is like tossing a coin, you get heads or tails. So here X will be either a 0 or a 1 after INT gets through rounding off RND(0). But what does RND(0) do for us? It creates a random number. But, by itself, the random number is a decimal fraction between 0 and 1. Try this:

10 PRINT RND(0):GOTO 10

When you RUN this, you'll see a series of decimal fractions, all kinds of different numbers. How would you get higher random numbers? Just multiply RND(0) by something. Try: PRINT RND(0) * 10. If you just want whole numbers (called *integers*), use INT.

Anyway, in our frog simulation we don't need these higher random numbers. If X becomes a 0 in line 120, we move the frog to the left (in line 160). If X becomes a 1 in line 120, we move the frog to the right (in line 140). Line 130 is the test to see which number is in X.

Notice that we don't need to write a line like: IF X=1 THEN 140. You *could* write that test and put it in line 135 if you wanted to. It wouldn't do any harm. But you don't need to. The computer will go to line 140 all by itself if X is anything other than a 0 when it's tested in line 130. The computer always performs each action in the order listed *unless you force it not to* with a GOTO, IF, or GOSUB command. If it doesn't come across one of those commands, it will go from line 140 to 150 to 160 and on up the list in simple line-number order.

Also on line 120 is another counter, the variable C. It will keep track of the total number of leaps the frog has made (either left or right). This lets us know how far he got before he fell off. It also sometimes shows that he's won the game. If he manages to leap a certain distance without falling, he's crossed the bridge.

But back to our simulation. After lines 130–160 make an adjustment to variable Y (our "position-of-the-frog" counter) we come to a series of tests in lines 170–190. Each of these tests will end the program in a different way. In 170, if the frog position is greater than (>) the total number of columns, he has fallen off the right side. In 180, if his position is less than 1, he has fallen off the left side. And, finally, in line 190, if he has taken more leaps than the width of the bridge, he made it across. You can change this line if you want to make it harder for him to cross the bridge. Just replace COLS with a higher number.

Line 200 prints the frog symbol on the screen to show us his position. The TAB command is just like a TAB key on a typewriter: It moves over a certain number of spaces from the left side of the screen. In this case, the number of spaces is controlled by the position variable Y.

Finally, to slow the frog down a bit, we put in line 210. This is often called a *delay loop* or a *do-nothing loop* because it simply takes up some time and serves no other purpose. Here we're asking the computer to count from 1 to 10 before going back down to line 120 and figuring out the frog's next leap.

```
100 COLS=40:REM PUT YOUR SCREEN SIZE HERE
                                    :rem 232
110 Y=COLS/2                        :rem 186
120 X=INT(RND(0)+.5):C=C+1          :rem 176
130 IFX=0THEN160                    :rem 174
140 Y=Y+3                           :rem 226
150 GOTO170                         :rem 103
160 Y=Y-3                           :rem 230
170 IFY>COLSTHENPRINT" >>>FROG FELL OFF R
    IGHT SIDE. IN"C"LEAPS.":END     :rem 120
180 IFY<1THENPRINT" <<<FROG FELL OFF LEFT
     SIDE. IN"C"LEAPS.":END          :rem 30
190 IFC>COLSTHENPRINT"FROG SAFELY CROSSED
     THE BRIDGE!":END               :rem 160
200 PRINTTAB(Y)"*"                   :rem 14
210 FORT=1TO10:NEXTT                 :rem 13
220 GOTO120                          :rem 96
```

BASIC Style—Program Evolution

Jim Butterfield, Associate Editor

Sometimes you see programs that are so crisp and neat that you wonder how the programmer's mind can be so orderly. The statements come out in an elegant, incisive style. Every line zeros in on exactly the right thing to do.

How does a programmer develop an elegant style? Why can't you write like that? Sometimes a lowly hacker can feel inferior when facing such immaculate programming style. Yet the program you see is often a matter of evolution—rewriting and tidying up. It's not always written that way from the beginning.

I have been accused of writing "squeaky clean" programs. It seems to me that you might like to see how my murky first programs get reworked and tightened up into their final version. In some ways, programming style isn't what you write (at least at first)—it's knowing what to look for when you clean up.

A Simple Lister

I needed to do an almost trivial job: list a file from disk to the printer. I had a minor extra feature to add: I wanted individual pages, so that the lines needed to be counted; I needed a title on each page; and at the end of the run, for the sake of neatness, I wanted the printer to eject the page.

It's not a demanding task, but I'd like to show you how I went about it. Even a simple job like that can be revised and tightened up extensively.

Here's my first program: I'll talk my way through the listing.

```
100 OPEN 4,3
```

Open file number four to the screen. Why? So I can send the program's output to the screen and see that it's working right. After the program looks good, I'll change the above line to OPEN 4,4.

```
105 OPEN 1,8,3,"CONTROL"
```

That's my input file to be listed.

```
110 REM START OF PAGE
120 FOR J=1 TO 2:PRINT#4:L=L+1:NEXT J
130 PRINT#4,"{5 SPACES}TITLE{3 SPACES}":L
    =L+1
140 PRINT#4:L=L+1
```

This prints the page title. I know I'll come back here for each new page, so I'm placing a REM statement here to mark the place. I rigorously add 1 to the line count, L, each time I print a line.

```
150 INPUT#1,A$:SW=ST
170 PRINT#4,A$:L=L+1
```

Here's where I input from disk and output (to the screen first, later to the printer). I need to save the value of ST (the status variable) so that later I can check to see if this is the last line from the file. ST will be changed by the PRINT# command, so I save its input value in variable SW.

```
180 IF L<62 GOTO 250
190 IF L=66 THEN L=0:GOTO 250
200 PRINT#4:L=L+1:GOTO 190
```

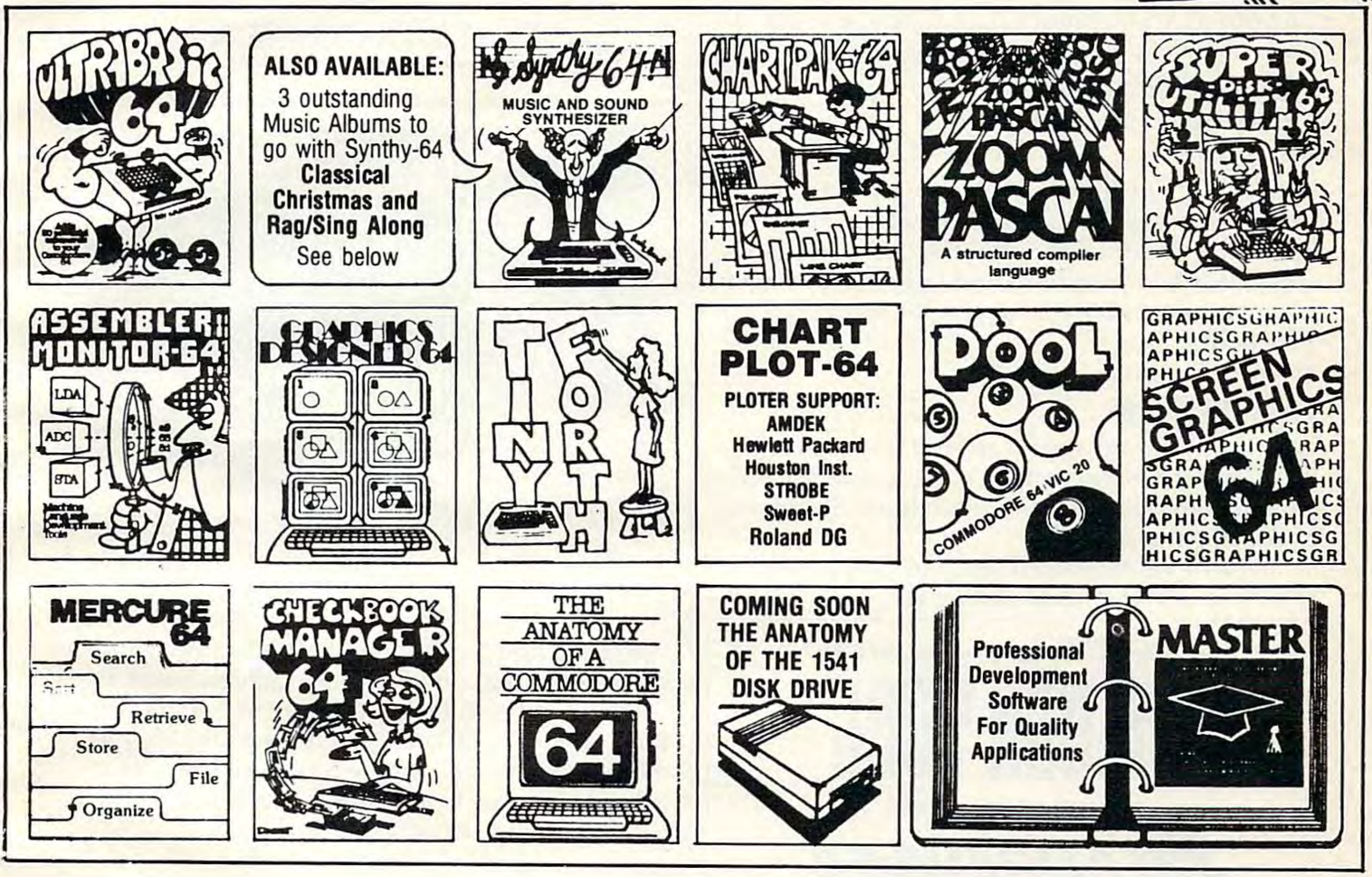
If I have printed the maximum number of lines desired, I want to eject the paper by printing until the line count L equals 66. Since each page has 66 lines, I'm now at the start of the next page and can set L back to zero.

```
250 IF SW<>0 GOTO 300
260 IF L=0 GOTO 110
270 GOTO 150
```

If I'm at the end of the input file (SW=0), I'll go to line 300 and wind things up. Otherwise, I want to go back.

Here's a cute touch—perhaps too cute for some tastes. Variable L can only be equal to zero if I've just ejected a page. If so, I want to go back

GET THE MOST OUT OF YOUR COMMODORE-64 OR VIC-20 COMPUTER



ULTRABASIC-64...Add 50 commands: graphics, music, TURTLE and game features.Tutorial,demo plus.
TAPE $39.95 DISK $42.95

SYNTHY-64... Sets the standard for all of the rest. Best 64-synthesizer anywhere. Samples and manual. **CASSETTE $29.95 DISK $32.95.** Also available: 3 great companion music albums; **Classical, Christmas,** and **Ragtime Sing-Along. DISK $12.95 Each.**

CHARTPAK-64...Professional qualtiy pie, line and bar charts. Menu driven, interactive, hardcopy.
DISK $42.95

ZOOM PASCAL-64...Produces 6502 machine code for speed. Floating point, Integers, strings File handling.
DISK $39.95

SUPER DISK UTILITY-64... Speed copy 4 ways: Total, Bam, Append or File. Dump or modify sectors. More.
DISK $22.95

ASSEMBLER-MONITOR-64 High speed language development. Eleven function monitor. Screen editing of source file. **DISK $32.95**

GRAPHICS DESIGNER-64... Menu-driven drawings, floor plans and illustrations etc.. Slide program capability.
DISK $32.95

TINY FORTH-64/20...Exciting language-low price. Powerful, extensible, 200 + word vocabulary.
TAPE $24.95 DISK $27.95

CHARTPLOT-64...Same fine features as CHARTPAK-64 with high quality output to plotters.
DISK $84.95

POOL-64/20...Play Fullrack or nine ball using hires graphics. Vic-20 required 8K expander.
TAPE $14.95 DISK $17.95

SCREEN GRAPHICS-64 Adds 24 hires, multicolor, sprite commands to 64-BASIC. Demo, tutorial and manual.
TAPE $24.95 DISK $27.95

MERCURE-64...Simple, powerful file management with fast design, entry search report capabilities. Tutorial. **DISK $32.95**

CHECKBOOK MANAGER-64 Simple check account maintainance. Optional screen or printer report and backup.
DISK $22.95

ANATOMY OF A COMMODORE-64 Complete guide. Full comment ROMS list, detailed internals, descriptions.
300 PAGE BOOK $19.95

MASTER-64...Full ISAM file management; powerful screen management; excellent printer generator; programmer's aid; BASIC 4.0 commands; machine language monitor; Software developers: NO RUNTIME ROYALTIES; With 150 page manual in three-ring binder and development software.
SOFTWARE ON DISK $84.95

✱ DEALER INQUIRIES INVITED

FREE CATALOG Ask for a listing of other Abacus Software for Commodore-64 or Vic-20

DISTRIBUTORS

Great Britain:
ADAMSOFT
18 Norwich Ave.
Rochdale, Lancs.
01-788-8963

Belgium:
Inter. Services
AVGuillaume 30
Brussel 1160, Belgium
2-660-1447

France:
Micro Application
147 Avenue Paul-Doumer
Rueill Malmaison, France
1-732-9254

Canada East:
KING MICROWARE LTD.
5950 Cote des Neiges
Montreal, Quebec H3S 1Z6
514/737-9335

West Germany:
DATA BECKER
Merowingerstr 30
4000 Dusseldorf
0211/312085

Sweden:
TIAL TRADING
PO 516
34300 Almhult
476-12304

Australia:
CW ELECTRONICS
416 Logan Road
Brisbane, Queens.
07-397-0808

New Zealand:
VISCOUNT ELECTRONICS
306-308 Church Street
Palmerston North
63-86-696

AVAILABLE AT COMPUTER STORES, OR WRITE:

Abacus Software

P.O. BOX 7211 GRAND RAPIDS, MICH. 49510

For postage & handling, add $1.50 (U.S. and Canada), add $3.00 for foreign. Make payment in U.S. dollars by check, money order or charge card. (Michigan Residents add 4% sales tax).

FOR QUICK SERVICE PHONE 616-241-5510

to 110 and print a new title. If not, get another line from the input file starting at line 150.

```
300 IF L<>0 GOTO 190
```

Here's a supercute trick. I pondered this one for a while, since it's almost too clever; that sort of thing can trip up your logic. Here's the objective: If we're finished, but the paper hasn't been ejected, go back to line 190 and eject the paper. The program will branch back here again, but this time variable L will be zero and we can finish the job by closing the files.

```
310 CLOSE 1
320 CLOSE 4
```

That's it. It's really rather messy. It works, and for a temporary job that's all we would need.

But it doesn't feel right. The code feels messy: It seems to jump around, and I don't get a feeling of smoothness in the program. It's time to pick at the coding.

First Revision

The first awkward spot is around lines 190 and 200. The routine to eject the paper works but looks clumsy. Besides, we call it twice (once at 62 lines, and again at end of file).

I have feelings about this part of the program, too. It's a unit to do a particular job. I would feel better moving it to a separate subroutine where it can stand out as an identifiable action. Sometimes I create a subroutine out of some in-line code and then move it back later; it helps me identify the modules that make up the program. Let's move the eject routine to a subroutine at line 500, clean it up a bit, and see what we get:

```
100 OPEN 4,3
105 OPEN 1,8,3,"CONTROL"
110 REM START OF PAGE
120 FOR J=1 TO 2:PRINT#4:L=L+1:NEXT J
130 PRINT#4,"{5 SPACES}TITLE{3 SPACES}":L
=L+1
140 PRINT#4:L=L+1
150 INPUT#1,A$:SW=ST
170 PRINT#4,A$:L=L+1
180 IF L<62 GOTO 250
190 GOSUB 500:GOTO 250
250 IF SW<>0 GOTO 300
260 IF L=0 GOTO 110
270 GOTO 150
300 IF L<>0 GOTO 190
310 CLOSE 1
320 CLOSE 4
330 END
500 FOR J=L TO 66:PRINT#4:NEXT J
510 L=0:RETURN
```

We can see that the GOTO 250 on line 190 is now redundant since we'll go there anyway. But we have other things to do. We're still trimming the program and have some distance to go yet.

Digging Deeper

Around lines 250 to 270, we jump around a lot. We have one jump forward to 300 and two jumps back to 110 or 150. The logic seems scattered.

I have a thing about loops: I like to see them neatly nested, with short jumps entirely within longer jumps. It might even be summarized as a rule of thumb: Where possible, make short jumps as short as possible.

Using this rule, I want to get the loop back to 150 into logical order first. Then we'll work in the longer loop to 110 and finally the forward branch to 300. We'll need to expand the logic using an AND operator, but that's not too hard.

As the routine is written, certain logical things start to fall together. For example, we don't have to GOTO forward to line 300. When we're finished writing the two loops, we'll fall into 300 naturally. ("Naturally" seems to be a key word in how programs seem to come together as you tighten them up.)

We can also tighten up the page eject conditions. If we write line 180 correctly, there will be no need to go back to get a page ejection. One option would be to call the subroutine at 500 twice. But if we think of what our objective really is at line 180, we can do it all correctly the first time through. Inverting the logic and adding an OR connective does the trick nicely.

Look at how far the original program has come:

```
100 OPEN 4,4
105 OPEN 1,8,3,"CONTROL"
110 REM START OF PAGE
120 FOR J=1 TO 2:PRINT#4:L=L+1:NEXT J
130 PRINT#4,"{5 SPACES}TITLE{3 SPACES}":L
    =L+1
140 PRINT#4:L=L+1
150 INPUT#1,A$:SW=ST
170 PRINT#4,A$:L=L+1
180 IF L>61 OR SW<>0 THEN GOSUB 500
250 IF SW=0 AND L>0 GOTO 150
260 IF SW=0 GOTO 110
310 CLOSE 1
320 CLOSE 4
330 END
500 FOR J=L TO 66:PRINT#4:NEXT J
510 L=0:RETURN
```

This is pleasing, but we can do even more. The repeated SW = 0 test in lines 250 and 260 still irks a little: It seems clumsy. The whole business is tied up with whether to print a title or not. Is there a better way? Could the test of L>0 be somehow shuttled up to the top of the loop instead of sitting at the bottom?

The Header Module

While we're thinking about it, that whole business of printing a header is really a module—we must do the whole thing, title and all, or nothing. If we move it out to a subroutine, we might see the logic flow more clearly. Let's do it and work on the logic flow. We end up with this:

```
100 OPEN 4,3
105 OPEN 1,8,3,"CONTROL"
110 IF L=0 THEN GOSUB 600
150 INPUT#1,A$:SW=ST
170 PRINT#4,A$:L=L+1
180 IF L>61 OR SW<>0 THEN GOSUB 500
260 IF SW=0 GOTO 110
310 CLOSE 1
320 CLOSE 4
330 END
500 FOR J=L TO 66:PRINT#4:NEXT J
510 L=0:RETURN
600 FOR J=L TO 2:PRINT#4:L=L+1:NEXT J
610 PRINT#4,"{5 SPACES}TITLE{3 SPACES}":L
=L+1
620 PRINT#4:L=L+1
630 RETURN
```

Look at that main section from lines 100 to 330. It now seems tight and concise like a finely tuned instrument.

Both subroutines—at lines 500 and 600—are called only once. If it seemed important, we could put them back into the main program stream. But I'm happy to see them as clearly isolated modules. At this stage I would add comments (line 499: REM PAGE EJECT and line 599: REM PAGE TITLE) to neaten things up.

Moral

First, what you see published is not always the first idea that popped into the author's head. The programmer is not always smarter than you. Time has been taken to groom the program into its final shape. When many people are going to read your code, you like to take a few extra pains with its appearance.

Second, don't be afraid to revise your programs, even if they work correctly. Sure, a one-shot program often doesn't warrant picking over; use it and forget it. But sometimes the exercise can reveal, almost accidentally, powerful and effective programming methods.

Third, *style* isn't an inborn talent that some people have and some don't. You learn it as you go. Some things you will discover for yourself, and others you'll pick up by looking at other people's programs.

The odd thing is that we instinctively recognize better writing when we have written it. You may not know exactly why, but you often feel good about a certain piece of programming. Usually, it's because it has style. ©

VIC/64 Memdata

Michael M. Milligan

"Memdata" converts a machine language routine into DATA statements and then erases itself, allowing you to save the DATA to disk or tape for later use.

Transferring a machine language routine into DATA statements involves a lot of work. To simplify the job, "Memdata" takes memory bytes between two addresses, inclusively, and returns DATA statements complete with BASIC line numbers. Once the program has generated those statements, it automatically erases itself, leaving only the DATA—as you will see by typing LIST after the program is run.

The first part of Memdata is a modified version of Jim Wilcox's "Automatic Line Numbers" (*COMPUTE!'s First Book of VIC*). The line numbers are the decimal value for the address of the first byte in each line. This serves as a marker to be sure that every location is accounted for. Also, because many machine language subroutines are located at the top of RAM, it makes the data line numbers high enough to be appended to an existing BASIC program. The appending can be done with the Datassette or disk files, thus eliminating a lot of typing.

Once you save the DATA statements you have created, enter NEW and PRINT PEEK (43),PEEK(44). Write down these two numbers. LOAD the program to which you wish to append the DATA. Next, enter POKE 43,PEEK(45)–2:POKE 44,PEEK(46). Then, LOAD the DATA statements right in there with the first program. When it is loaded, POKE 43 and 44 with the numbers you wrote down after the earlier PEEK(43) and PEEK(44). This will merge the two programs if the DATA statement line numbers are higher than the highest line numbers in the original program.

Memdata erases itself in a novel way. Because line numbers used in Memdata are so high, the DATA statements will be the first lines in the BASIC program area. After the DATA statements are created, Memdata searches memory for DATA (token 131) following a line number. When it finds something besides a DATA token, it POKEs zeros into the high and low bytes of the link address for that line. These two zeros, plus the zero byte that signals end-of-line, make up the three zero bytes that convince the LIST and SAVE functions that the end of the BASIC program has been reached. Because of this, it's important to save the program before you run it for the first time.

Memdata

Refer to the "Automatic Proofreader" article before typing this program in.

```
63720 PRINT"{CLR}{21 I}":PRINT"{RVS}TO
      {SPACE}CONVERT MEMORY TO {OFF}"
                                :rem 159
63723 PRINT"{RVS}DATA STATEMENTS ENTER
      {OFF}":PRINT"{21 U}"      :rem 61
63730 PRINT"INCLUSIVE DECIMAL":PRINT"MEMO
      RY LOCATIONS":PRINT:INPUT"FROM";A
                                :rem 138
63733 PRINT:INPUT"TO";C:PRINT:INPUT"BYTES
       PER LINE";B              :rem 170
63735 C=C/256:POKE251,(C-INT(C))*256:POKE
      252,C                     :rem 60
63740 POKE2,B:PRINT"{CLR}";     :rem 172
63750 B=A/256:POKE253,(B-INT(B))*256:POKE
      254,B                     :rem 55
63755 PRINT:PRINTMID$(STR$(A),2,LEN(STR$(
      A))-1);"DATA ";           :rem 247
63760 FORI=0TOPEEK(2)-1         :rem 76
63763 A$=STR$(PEEK(A+I))+","    :rem 223
63765 IFA+I>PEEK(251)+256*PEEK(252)GOTO63
      780                       :rem 221
63768 PRINTMID$(A$,2,LEN(A$)-1);  :rem 7
63770 IFA+I=PEEK(251)+256*PEEK(252)GOTO63
      830                       :rem 212
63775 NEXTI:GOTO63830           :rem 11
63780 PRINT"{LEFT} ":GOTO 63870  :rem 241
63830 PRINT"{LEFT} ":POKE631+PEEK(198),13
                                :rem 72
63840 PRINT"GO63850":FORA=631TO634:POKEA,
      145:NEXT:POKEA,13:POKE636,13:POKE19
      8,6                       :rem 147
63841 END                       :rem 221
63850 PRINT"{2 UP}":FORA=1TO3:PRINT"
      {8 SPACES}":NEXT:PRINT"{3 UP}";
                                :rem 28
63860 A=PEEK(253)+256*PEEK(254)+PEEK(2):G
      OTO63750                  :rem 227
63870 Q=PEEK(43):U=PEEK(44)     :rem 29
63880 IFPEEK(Q+4+256*U)<>131GOTO63900
                                :rem 79
63890 Q1=PEEK(Q+256*U):U1=PEEK(Q+1+256*U)
      :Q=Q1:U=U1:GOTO63880      :rem 86
63900 P=Q+256*U:POKEP,0:POKEP+1,0:rem 173
63910 PRINT"{CLR}{21 I}"        :rem 177
63920 PRINT"{RVS}TYPE LIST TO SEE DATA
      {OFF}"                    :rem 145
63930 PRINT"{21 U}"             :rem 238
```

©

INSIGHT: Atari

Bill Wilkinson

Learning How

A month or two ago, I stated that I couldn't possibly teach beginning machine language programming in this column—it would consume my entire output for a year or more. And yet I continue to get letters that ask me "How do you learn to write programs?"

I believe that those who ask the question are not asking for a tutorial on the foibles and pitfalls of the FOR-NEXT loop. Nor are they really asking about the intricacies of the 6502 instruction set. Most of them have already mastered the tutorial-level material on their chosen language. What these perplexed people are really asking is "What good is all this programming stuff, anyway?"

And that is not really surprising. So many tutorials tell you *how* to write a program to do such and such. So few discuss *why*. Too often, learning to program is approached like learning a foreign language. Memorize the conjugations and punctuation; put sentences together like this; and if someone asks you "G'dye moya k'neega?" you know what to answer (providing you were studying Russian instead of Spanish).

Computer Conversations

But the need to learn human languages is obvious: The first time you feel hungry in Paris, you can ask for directions to a restaurant in your best Berlitz French. You don't have to "design" a conversation. Not so with learning to program: "Okay, now I know all these neat keywords and syntax and punctuation. How do I start a conversation?" Well, as I hinted above, the secret is that you must *design* a program.

To some, this design process is simple and obvious. Others never really get the hang of it. (Would it surprise you to learn that many professional programmers never become expert at designing? They make their living implementing other people's designs.) And many, like myself, become somewhat proficient at a few kinds of designs while remaining incompetent at others. (My lament: I don't think I will ever achieve the level of creativity necessary to design a really good game.)

Now, all the above philosophizing surely has some purpose, you hope. Indeed, I think it does.

Kibitzing

I have been promising for a few months now that I would provide patches to allow the Atari 1050 drive to work in enhanced mode with good old Atari DOS 2.0s. Well, I finally gathered enough information to begin the task, and I thought you might enjoy looking over my shoulder while I tackle the problem.

This will be a kind of short diary of what I have gone through. There have been more side-tracks and bugs and flat-out boo-boos than I can find room for here. And I won't even tell you how many assemblies I have made (though I will say I made about 10 or 12 just looking for the best of several possibilities for a series of shift instructions).

Even though I admire and strive for a "clean" design, I am apt to take the course of least resistance if I am confident it will work properly. With that in mind, then, let us begin tackling our task.

Note: I will make frequent reference to the listing of Atari DOS 2.0s as published in the book *Inside Atari DOS* from COMPUTE! Books. Page numbers and line numbers in square brackets [131: 1350] refer to the book.

It will *not* be necessary to own the book to understand most of what is going on, but having the book available will make it easier. Also, if you do not understand machine language, neither the book nor my explanations will be easy to follow, but you can still use the results (which will appear next month).

The 1050 And DOS 2.0s

The first thing we must always do is define the task. Here, that is deceptively simple to do: Make the enhanced density mode of the Atari 1050 drive work with Atari DOS 2.0s.

The next step is much harder: Design the implementation of the task. And, actually, this single step consists of many substeps. For example, let's first investigate the facts which I knew when I started.

The drives:

Item: An Atari 810 drive has 40 tracks of 18 sectors of 128 bytes each. That's a total of 720 sectors.

ATARI®

ATARI is a trademark of ATARI, INC.

NEW LOWER PRICES!!

ATARI 800XL .... Call

- Atari 1050 Disk Drive ...... Call
- Atari 1010 Recorder 77.00
- Atari 1027 Printer .... Call
- Atari 850 Interface .... Call

MOSAIC

- 64K Ram/400 ..... 149.00
- 64K Ram +Cable Kit 400/800 .... 169.00
- 48K Ram Kit 94.00
- 16/32 Expander 64.95
- 32K Ram 77.95
- Mosaic Adaptor 49.95

Memory Expansion for Atari 600XL .. Call

3RD Party Printer Interfaces

- Interfast I 139.00
- Apeface w/cable included 69.95

PRINTERS

- Axiom AT-100 .. 229.00
- Axiom AT-550 .. 329.00
- Riteman .... Call
- Gemini 10X ..... Call
- Alphacom ...... Call
- Okidata 92A .... Call

Call for assorted Printer Packages

DISK DRIVES

- Percom ....
- Trak .... Call
- Concorde .... For
- Indus GT .... Prices

RS232 Modem Adaptor (Through serial port) 39.95

Koala Touch Tablet Cart or Disk . . . 69.95

MONITORS

- USI .... Call
- AMDEK .... Call

STIMUTECH SUBLIMINAL SOFTWARE FOR ATARI AND CBM 64

Expando-Vision Interface w/one **Free** Cart 99.00

- Weight Control
- Study Habits
- Stress Control
- Drinking Control
- Smoking Control
- Career Success
- Sexual Confidence

Addit'l. Rom Carts $29.95 ea.

COMMODORE 64

- **CBM 64 .... Call**
- **1541 Disk Drive 249.00**
- **1525 80 Column Printer 229.00**
- **1530 Datasette 66.00**
- **1702 Color Monitor 249.00**
- **1650 AD/AA Modem 89.00**
- **RS 232 Interface 44.00**

COMMODORE 64 SOFTWARE

- Program Ref. Guide ... 19.95
- Assembler - D 17.95
- Easy Finance I, II, III, IV - D 19.95
- Easy Calc - D 64.95
- Easy Mail - D 17.95
- Easy Script - D 39.95
- Easy Spell - D 19.95
- Logo - D 39.95
- The Manager - D ... 87.95
- General Ledger - .. 37.95
- Accts. Rec. - D 37.95
- Accts. Pay. - D 37.95
- Code Writer - D 37.95
- Zork I, II or III D 29.95
- Suspended - D 29.95
- Starcross - D 29.95
- Deadline - D 29.95

ATARI SOFTWARE

ATARI

- Conversational Lang. - T .. 44.95
- My First Alphabet - D 26.95
- Touch Typing - T 19.95
- Home Filing Mgr. - D 37.95
- Star Raiders - Cart 32.95
- Assembly Editor - Cart 46.95
- Macroassembler - D 67.95
- Invitation to Program I T ... 19.95
- Basketball - Cart 26.95
- Graph-it - T 15.95
- Qix - Cart 32.95
- Dig Dug - Cart 32.95
- Atariwriter - Cart 74.95
- Donkey Kong - Cart 37.95
- Ms. Pac-Man - Cart 39.95
- Tennis - Cart 35.95
- Eastern Front - Cart 32.95
- Donkey Kong Jr. - Cart ... 39.95
- Pengo - Cart 35.95
- Logo - Cart 79.95
- Robitron - Cart 35.95
- Pole Position - Cart 39.95
- Microsoft Basic II - C 67.95
- Paint - D 33.95
- Caverns of Mars - Cart 32.95
- Joust - Cart 39.95
- Visicalc - D 159.95

SIERRA ON-LINE

- Homeword - D 49.95
- Dark Crystal - D 27.95
- Frogger - D/T 23.95
- Quest For Tires - D 23.95
- Ultima I - D 23.95
- Ultima II - D 41.94
- Wizard/Princess - D 22.95

INFOCOM

- Deadline - D 34.95
- Witness - D 34.95
- Zork I, II, III - D 27.95
- Starcross - D 27.95
- Planetfall - D 34.95
- Suspended - D 34.95
- Enchanter - D 34.95
- Infidel - D 34.95
- Sorcerer - D 34.95

BRODERBUND

- AE - D 23.95
- Arcade Machine - D 41.95
- Bank St. Writer - D 49.95
- Lode Runner - D 23.95
- Drol - D 23.95
- Spare Change - D 23.95
- Choplifter - D 23.95

PARKER BROS.

- Astrochase - Cart 34.95
- Frogger - Cart 34.95
- Q-Bert - Cart 34.95
- Popeye - Cart 34.95

EPYX

- Gateway To Apshai - Cart 27.95
- Pitstop - Cart 27.95
- Swat Rescue - Cart 27.95
- Silicon Warrior - Cart 27.95
- Fun With Music - Cart 27.95
- Jumpman Jr. - Cart 27.95
- Dragonriders of Pern - D/T 27.95
- Temple of Apshai - D/T ... 27.95
- Hellfire Warrior - D/T 20.95
- Fun With Art - Cart 27.95

DATA SOFT

- Money Wizard - D 49.95
- Letter Wizard - D 49.95
- Spell Wizard - D 34.95
- Genesis - D/T 20.95
- Heathcliff - D/T 20.95
- Pooyan - D/T 20.95
- Zaxxon - D/T 27.95
- Micropainter - D 23.95

SPINNAKER

- Up For Grabs - Cart 27.95
- Cosmic Life - Cart 23.95
- Facemaker - Cart 23.95
- Alphabet Zoo - Cart 23.95
- Fraction Fever - Cart 23.95
- Kids on Keys - Cart 23.95
- Delta Drawing - Cart 27.95
- Story Machine - Cart 27.95
- Rhymes & Riddles - D 20.95
- Amazing Things - D 27.95
- Hey Diddle Diddle - D 20.95
- Snooper Troops 1 or 2 - D .. 30.95
- Trains - D 27.95
- Aerobics - D 30.95

LJK

- Letter Perfect - D 74.95
- Data Perfect - D 74.95
- Spell Perfect - D 64.95

MISCELLANEOUS FOR ATARI

- Castle-Wolfenstein - D 20.95
- Home Accountant - D 52.95
- Master Type - D/Cart 27.95
- Spelunker - D 27.95
- Flight Simulator II - D 37.95
- Zombies - D/T 23.95
- Mr. Robot - D 23.95
- River Raid - Cart 31.95
- Diskey - D 34.95
- Chatterbee - D 27.95
- Rally Speedway - Cart 34.95
- Ultima III - D 41.95
- Miner 2049'er - Cart 34.95
- Scraper Caper - Cart 34.95
- Basic XL - Cart 74.95
- Monkey Wrench II - Cart .. 49.95
- Omnimon 82.95
- Star Trek - Cart 27.95
- Starbowl Football - D/T ... 22.95
- Hockey - Cart 27.95
- Computer War - Cart 27.95
- Flip Flop - D/T 20.95
- Boulder Dash - D/T 20.95
- Carrier Force - D 41.95
- Combat Leader - D/T 27.95
- Encounter - D/T 23.95
- Quasimoto - D/T 23.95
- N.Y.C. - D/T 23.95
- Compute Books .... Call

Hundreds of items available for the Atari and CBM 64 . . . please call.

COMMODORE 64 SOFTWARE

ATARISOFT

- Pac-Mac 34.95
- Centipede 34.95
- Defender 34.95
- Dig Dug 34.95
- Donkey Kong 34.95
- Stargate 34.95
- Robotron 34.95

ACCESS

- Neutral Zone - D/T 26.95
- Spritemaster - D/T 27.95
- Beachhead - D/T 26.95

BLUE SKY

- Calc Result/Easy - Cart .... Call
- Calc Result/Advanced - D ... Call
- Script 64 - D .... Call

BATTERIES INCLUDED

- Paper Clip 99.95
- Delphi Oracle 114.95

DATA 20

- Video Pak 80 .... 139.95
- Z80 Video Pack 229.95

HES

- Multiplan - D 74.95
- 64 Forth - Cart 41.95
- HES Writer - Cart 32.95
- HES Mon - Cart 27.95
- Turtle Graphics - Cart 41.95
- Mutant Camels 20.95
- HES Modem 59.95

CIMMARON

- Instaspeed - D/T 74.95
- Instawriter - Cart 55.95
- Instamail - D/T 29.95

SIRIUS

- Blackpoole - D 27.95
- Repton - D 27.95
- Type Attack - D 27.95
- Wayout - D 27.95
- Critical Mass - D 27.95

SYNAPSE

- Blue Max - D/T 23.95
- Zaxxon - D/T 23.95
- Morgul - D/T 23.95
- Shamus II - D/T 23.95
- Sentinal - D/T 23.95
- Pharoah's Curse - D/T 23.95
- Slamball - D/T 23.95

MISCELLANEOUS FOR CBM 64

- Flight Simulator II - D 37.95
- Night Mission/Pinball - D .. 20.95
- Praticalc PS - D 59.95
- M-File - D 84.95
- Word Pro 3+/Spell D 74.95
- Home Accountant - D 52.95
- Step by Step - D/T 44.95
- Barron's Sat - D 59.95
- Money Manager - D/T 19.95
- Bristles - D/T 20.95
- Telestar 64 - Cart 37.95
- Star League Baseball - D/T 22.95
- Zombies - D/T 23.95
- Castle Wolfenstein - D 20.95
- Swashbuckler - D 23.95
- Sorcerer - D 34.95
- Computer Baseball - D 27.95
- Ringside Seat - D 27.95
- Professional Golf - D 27.95
- Congo Bongo - Cart 27.95
- Mastertype - Cart 27.95
- Solo Flight - D 23.95
- Vic Switch 124.95
- Cardco + G 64.95

ACCESSORIES

- WICO Joystick .... Call
- Flip'n' File Trunks 20.95
- Flip'n' File Cart 20.95
- Joysensor 24.95
- Elephant Disks (Box of 10) 20.95
- WICO Trakball 37.95
- KRAFT Joystick 15.95

D - Disk
T - Cassette
Cart - Cartridge

ComputAbility™

To Order Call Toll Free
800-558-0003

For Technical Info, Order Inquiries, or for Wisc. Orders -
414-351-2007

NO SURCHARGE FOR MASTERCARD OR VISA

COMPUTABILITY
P.O. Box 17882
Milwaukee, WI 53217

MasterCard — VISA

ORDER LINES OPEN
Mon-Fri 11 AM - 7 PM CST
Sat 12 PM - 5 PM CST

ORDERING INFORMATION For fast delivery send cashier's check, money order or direct bank transfers Personal and company checks allow 2 weeks to clear. Charges for COD are $3.00. School Purchase Orders welcome. In CONTINENTAL USA, include $3.00 shipping per software order. Include 3% shipping on all Hardware orders, minimum $3.00. Mastercard & Visa please include card # and expiration date. WI residents please add 5% sales tax. HI, AK, FPO, APO, Canadian orders - add 5% shipping, minimum $5.00. All other foreign orders, please add 15% shipping, minimum $10.00. All goods are new and include factory warranty. Due to our low prices, all sales are final. All defective returns must have a return authorization number. Please call 414-351-2007 to obtain an RA# or your return will NOT be accepted for replacement or repair. Prices and availability are subject to change without notice.

Item: An Atari 1050 drive has 40 tracks of 26 sectors of 128 bytes each, for a total of 1040 sectors.

Item: A 1050 will automatically read either density diskette (single or enhanced), but it formats a new diskette according to the format command it receives. In particular, a ! command ($21) causes single-density formatting, while a " command ($22) causes enhanced density.

The software:

Item: DOS 2 is capable of accessing both 810 drives and their double-density equivalents (drives with 40 tracks of 18 sectors of 256 bytes each).

Item: There is an inherent limit of 1024 sectors in DOS 2, since it allows only a 10-bit sector number in the link field of each sector. Also, on a single density diskette, DOS 2 accesses only 719 of the 720 sectors.

Item: The listing of Atari DOS. Actually, this is *not* a "known" item, and much of what follows is a discussion of what I learned and applied from reading the listing several times.

Finding The Format

Armed with these knowns, let's tackle the unknowns. It seemed to me that the first point to attack was the disparity between what the 1050 was capable of and what DOS 2 would request of it. All of a sudden, DOS 2 must be able to understand three different kinds of disk formats. Question: How can DOS tell what format a particular diskette is?

The answer is to be found in the DOS listing [66: 2213–2222]. During initialization, a status request is made of each drive. When the drive responds, one of the bytes it returns to the computer describes the drive's type. In particular, the listing makes it clear that a double-density disk has bit 5 ($20) set on. DOS 2 uses this bit to differentiate between 128-byte and 256-byte sectors.

All very well, even assuming that an enhanced mode 1050 returns a zero bit here (which it does, thus properly indicating 128-byte sectors). But what distinguishes an enhanced density diskette? I confess that I obtained the answer to this question through a simple experiment: I simply booted a system with an Indus 1050-compatible drive as D2 and looked at the status value it returned during DOS initialization. Lo and behold, it returned $80. Not surprisingly, the high bit is off in 810 and double-density modes. Voilà.

Sector Limits

The second major question to investigate is "How many of the 1050's sectors can we make DOS 2 utilize?" Well, we already know that 1024 is an upper limit. Is there any other limiting factor? The answer is in the layout of the Volume Table Of Contents (VTOC) under DOS 2. The VTOC contains a single bit for each accessible sector on the disk (a scheme known as a *bitmap*, though Atari literature often uses *VTOC* and *bitmap* interchangeably). If a bit is on (1), the corresponding sector is available. If a bit is off (0), the sector is in use. With eight bits per byte, then, there must be 90 bytes in the bitmap.

DOS 2 allows only a single sector (in this case, 128 bytes) for the VTOC of each diskette. While we could circumvent this restriction, it would require a lot of work, and might cause some secondary problems. (I don't want to go into this subject more now, but it cost me four to six hours of investigation before I decided against a two-sector VTOC.)

In 128 bytes, there are 1024 bits. So it would seem that the limit on number of sectors is indeed 1024. Alas, it is not to be. The description of the VTOC clearly calls out usages for the first six bytes (DOS type, maximum number of sectors, current number of sectors, write-required flag) and reserves the next four. So now we are down to 118 bytes and 944 sectors. Is that our limit?

A Final Of 976 Sectors

At first, I was inclined to say it is. But I pored over the listing a couple more times, checked every memory reference that was related, and finally concluded that we could use the four reserved bytes. Which gives us 122 bytes and a final maximum of 976 sectors. Well, that doesn't seem too bad. We are only 64 sectors away from the theoretical maximum and surely a lot better off than with a limit of 720 sectors.

So this is our plan: Use the upper bit ($80) of the drive status to recognize an enhanced density diskette; allow 975 sectors (DOS 2 always throws away the first possible sector); displace the bitmap in the VTOC by 4 bytes on the low end and lengthen it to 122 bytes.

Implementing Our Plan

By the time I had decided on a plan, over half the time I had allotted to this project had elapsed. As I write this, all the allotted time is gone, and I am not done yet. Sounds like a typical software project. Anyway, this month I will tell you of the difficulties I faced. Next month we can decide how well I faced them. In any case, let's begin the next step.

Before I could start the actual coding of the modifications, I had to find all the places in DOS which would be affected by my scheme. While many parts of DOS are affected by a change in density (from 128- to 256-byte sectors), there are only a few routines which actually care about such things as disk status, where the VTOC's bitmap is, and how many sectors are available.

Some of the routines I could successfully ignore. For example, when you delete a file and free up its sectors for later use, you must bump the count of free sectors. But if the rest of DOS is working, you don't have to check for validity of the bumped value. The same thing is true when we allocate a free sector and must decrement the count. And the boot process cares whether we are using 128- or 256-byte sectors, but it doesn't care how many sectors are on the disk.

Some Areas Need Patching

But there *are* several spots which definitely need attention, so let's discuss them now (next month we discuss the solutions).

1. In the BSIO (Basic Sector Input Output) routine, there is a check for a format command [65: 2144]. DOS 2 simply compares the current command with $21 (!) and makes a decision according to an exact match. Now, though, we must allow for either $21 or $22 (") as format commands.

2. In DOS initialization [66: 2218], each accessible drive is checked for its status. DOS 2 ignores all bits of the status except bit 5 ($20) and stores a 1 or 2 (single or double density) in the drive table (DRVTBL) for each drive so checked. We need to find a way to capture and use bit 7 ($80), preferably by keeping it in DRVTBL, also. Fortunately, the only other routine which accesses DRVTBL is SETUP, which we discuss below.

3. In XFORMAT [79: 3510], the actual format command is stored in the DCB (for use by BSIO, as above). We need to allow for either $22 or $21, while DOS 2 allows only $21.

4. Also in XFORMAT [79: 3547, 3552], the maximum number of sectors and number of sectors available are stored in the VTOC which is being created (for the newly formatted disk). Currently, DOS 2 simply uses LDA # (load immediate value) to store what it thinks is the only possible count (707). We must provide for the enhanced density count as well.

5. Again in XFORMAT [80: 3559–3570], there are several assumptions made about how big the bitmap is and where the directory and boot sectors are to be represented in the map. Since we will move the base of the map down four bytes, we must provide for variable numbers here, as well.

6. In FRESECT [90: 5166], the base of the bitmap is assumed to be byte 10 ($0A) of the VTOC. We must change the assumption.

7. In GETSECTOR [91: 5199, 5202, 5239], similar assumptions about the bitmap are coded via immediate loads.

8. In SETUP [92: 5288], which is called by every major routine in DOS 2, the type byte stored in DRVTBL (see item 2, above) is simply transferred to a global location (DRVTYP) for use by other routines. If we change what is stored in DRVTBL, we need to change how and what we store in DRVTYP.

Keeping The Patches Small

And that's it. Not too bad, right? If only that were true. Remember, our goal here is to *patch* the standard version of DOS without affecting its normal operations and without requiring a reassembly of the whole thing to make our patches fit. In general, then, the smaller and fewer the patches, the better.

The real problem here is the number of load immediate instructions, used to implement what are now to become invalid assumptions. If these were three-byte instructions (such as loads from a non-zero page memory location), we would have a simple task: Change the values in the locations being loaded.

Since they are load immediate instructions, though, our only choices are to either make large and cumbersome patches (generally JSRs to subroutines which will do the work, but remember that JSR occupies three bytes), use loads from zero page (a neat alternative, but we have no zero page available to us), or to continue to use load immediate.

Self-Modifying Routines

My choice? Continue to use load immediate. But how? By producing some (shudder at this next phrase, please) self-modifying routines. Remember how I said at the beginning that I sometimes took the path of least resistance? This is one of those sometimes.

The "trick" which allows my scheme to work is relatively simple: Every routine which needs a load immediate changed is only used by DOS 2 after a call has been made to SETUP. Basically, SETUP examines the disk number and drive type and produces various pointers and values in fixed locations for use by other, higher-level routines. What would be more appropriate than for SETUP to also set up the needed values which will be loaded in immediate mode?

And this is, indeed, the plan I tried. At the point where SETUP stores the drive type [92: 5288], I placed a JSR to my patch-it routine. And my patch-it routine used the disk type information to determine which of a pair of immediate values would be used in each of the cases noted above. It looked like it would work.

Fitting The Patch Into DOS.SYS

Except (You knew that was coming, didn't you?) where do I put the patch? I have discussed this subject before, so let me succinctly say that the only sizable patch area in DOS.SYS is at location

$1501, in the gap between DOS.SYS and Mini-DUP (the root of DUP.SYS). There are exactly 63 bytes available there. And my routine was about 85 bytes long.

The story of how I pared my patch down to fit (just barely) will have to wait for next month. Fortunately, it is a short patch. Also fortunately, there are a couple of small patch spaces still floating around in DOS.

Incidentally, if you were looking for the continuation of my notes on how to load saved binary files, keep looking. It turns out that the subject has direct bearing on what we are doing here, so it seemed not inappropriate to postpone it a month (or possibly two). ©

A BASIC Cross-Reference

Jim Butterfield, Associate Editor

"Cross-Ref" is a valuable programming tool that serves several purposes. Not only does it locate all line number and variable references in a program, but it also helps you prepare documentation and even tighten up your program. It's for BASIC programs stored on disk and will output to the screen or printer. For PET/CBM (Upgrade and 4.0 BASIC) and Commodore 64.

"Cross-Ref" and "Cross-Ref64" will analyze a BASIC program stored on disk and give you information on all line number references and all variable references.

It works only with programs written in BASIC; it does not work with programs stored on tape. A program SAVEd on disk may be manipulated as if it were a data file; but a program on tape cannot be handled in that way.

All types of variables are detected and listed: regular variables, strings, integer variables, and arrays. This includes special variables such as TI, TI$, or ST. If a variable name contains more than two characters, only the first two will be shown. (They're the only ones used by BASIC.) So HOUSE is the same variable as HONK.

While Everything Is Fresh In Your Mind

If you have completed writing a program, the Cross-Ref output will serve as a valuable piece of documentation. As each line and variable is listed, you may note its purpose while everything is fresh in your mind: "Line 300 is the start of the analysis: variable A$ is the name of the input file...."

Even if your program is not complete, Cross-Ref can be useful. In large programs, you may wonder what variable names have been used; you want to pick a fresh variable name that won't conflict with anything else. Alternatively, a test run may reveal a problem that shows up within the subroutine that starts at line 750: You can find all calls to that subroutine.

If you're thinking of tightening up your program, you may want to pack two or three lines of code together into a single line. But you can't do this if some of the lines are referenced elsewhere in the program. Cross-Ref will tell you the story.

And if you're looking at somebody else's program, and don't know, say, what variable V3 is being used for, you can run Cross-Ref and find every occurrence of V3.

Running The Program

LOAD and RUN Cross-Ref. Be sure you place the disk with the program you want to cross-reference into the disk drive.

When Cross-Ref asks PROGRAM?, type in the name of the program you wish to analyze. You may use pattern matching if you wish: For example, BAG* will match program name BAGELS.

Everything happens very fast. The disk runs for about the same amount of time that is needed to load the program in question. Then you are asked PRINTER? At that time, the cross-reference is complete; the program wants to know where to deliver the results. Answer Y or N.

Output may be to screen or printer. The line number cross-reference appears first. The referenced line number appears, followed by a colon, then the lines where it is used.

Then the variable cross-reference appears, in alphabetical order. Arrays are shown with a single left parenthesis, so that A(M+NV%) will be shown as A(—and there will also be other entries for M and NV%, of course.

Sometimes a variable or line number will be used more than once on a single line of your program, for example, "100 X=X+7:IF X>20 THEN X=0". In this case, the cross-reference for X will show line 100 only once.

Machine Language For Speed

It's written mostly in machine language for speed. An early BASIC version of this program appeared in COMPUTE!, May/June 1980 (that's Issue 4); being a BASIC program, it ran slo-o-o-owly. But it worked on identical principles to this version of Cross-Ref.

If you're interested in the mechanics, the next few paragraphs give an insight into the unusual logic of both the original BASIC version and the machine language program presented here.

Because of the plethora of characters to be analyzed, an unusual approach was taken. It might be called a "state transition" program.

Here's the general idea. When we begin the analysis of a BASIC line, we start in state A. In this state, we are interested in only a few characters: an alphabetic, which signals the start of a variable; a GOTO, THEN, or GOSUB, which signals that a line number may be coming; a REM, which indicates we should ignore everything up to the end of the line; quote marks, which tell us that the next few characters will not be of interest to us; and binary zero, which signals end of line.

If we don't see any of these characters, we remain in state A and get the next character, throwing the old one away. But if we do see a character of interest, we switch to a new state.

Suppose we're looking at a line that says:

```
FOR J=1 TO 9:X35$="HELLO":GOTO 500
```

We start in state A. The first thing we get is the FOR—it's not a character, but a specially coded *token*. Throw it away; it's not on our list. Continuing on our line, we see a space, which we trash, followed by the letter J. Aha! It's an alphabetic, which tells us "we're in a variable—start collecting characters." At this point we don't know if the variable is called J, J5, JEEPERS, or JR$. We collect the J and switch to state B.

In state B, we are looking for a whole different set of characters. Alphabetic and numeric characters will be collected into our variable name and will move us to state C. On the other hand, a dollar or percent sign will also be collected, but will move us to state E, where we look for a possible array. Continuing the options: a left parenthesis would signal an array; collect it and wrap up this label. A space will be ignored. Almost anything else (in our example, the equals token) will cause the label to be wrapped up and put away, returning us to state A.

Back in state A again, we throw away the equals, the 1 character, the space, the TO token, the 9, and the colon. Suddenly we hit the X: Collect it, and we're off to state B again. This time, state C finds a numeric, collects it, and switches us to state D. State D throws away the 5. We stay in state D and discover the dollar sign, which is duly collected, and we flip to state E. The equals sign drops us back to state A; but we wrap up the collected characters X3$ and enter them into the results table. And so on. Each individual state searches for its own set of characters which trigger an action and a movement to another state.

The program to do all this is surprisingly small. The state transition table that directs the program from one state to another is surprisingly big.

There are tricky bits, some of which involve the strange syntax of the PRINT statement. It's possible to write BASIC lines such as:

```
PRINT A$B$C%D(3)E
```

I'd much rather use semicolons to separate those variables, but since we're allowed to code that way, extra programming must be added to Cross-Ref to pick out the variables when they are mushed together like that.

Typing Cross Reference

Both the PET/CBM and 64 versions of this program use a special technique to attach the machine language to the BASIC portion of the program. The ML is located immediately following the end of the BASIC program, then the zero-page pointer to the end of the program is changed to point to the end of the ML. This fools the computer into treating the ML as part of the BASIC program.

To enter the PET/CBM version, first type in Program 1. You must enter it *exactly* as it is shown because the ML must begin at exactly the end of BASIC. You can check by typing the following line in direct mode:

```
PRINT PEEK(1261),PEEK(1262),PEEK(1263)
```

If you have entered Program 1 correctly, you'll see:

```
58    160    52
```

If these are not the values you get, check for spaces added or left out. When you have Program 1 entered correctly, type the following line in direct mode:

```
POKE 41,10:POKE 2560,0:NEW
```

Then type in and RUN Program 2. Program 2 will check for DATA statement errors as it POKEs the ML into the proper locations. If no errors are detected, the program will change the pointers in zero page to attach the ML to the BASIC from Program 1. When you type LIST after Program 2 is finished, you should see the lines from Program 1. Although it doesn't show, the ML POKEd by Program 2 is also in place. You should immediately SAVE a copy of the completed Cross-Ref program. You will not need the old Program 1 or 2 again.

The 64 Version

To enter the 64 version (Program 3), you *must* use the MLX machine language editor. If you have not already typed in MLX from a previous issue of COMPUTE!, there's a copy elsewhere in this issue. Be sure you read the accompanying article and understand how to use MLX before you begin typing in the data from Program 3. The MLX listing in Program 3 contains the BASIC as well as the ML portions of Cross-Ref, so no separate BASIC program must be typed in. MLX makes things *much* easier—it's a program worth SAVEing for this, and future, programs.

Because Cross-Ref begins at the default start-of-BASIC address (where MLX would normally be located), you must adjust the 64 so that the BASIC area for MLX is above the area of memory which Cross-Ref will occupy. Do this by typing the following line in direct mode (no line number):

POKE 44,16:POKE 642,16:POKE 4096,0:NEW

If you do not finish typing all of Program 3 in one session, see the instructions in the MLX article on saving an unfinished version of your work. Note that you must also type the direct mode line above *before* loading MLX again to continue your work.

When MLX is first RUN, it will ask you for a starting and ending address. For Cross-Ref, the proper values are:

starting address	**2049**
ending address	**3398**

Use the MLX Save option to make a copy of your work. The version of Cross-Ref created by MLX can then be LOADed and RUN like a regular BASIC program.

An early version of Cross-Ref for PET/CBM, called XREF, was published in *Cursor* magazine (which came on cassette tape), issue 25. The details are different, but the program's general speed and other characteristics are about the same.

Could Cross-Ref be expanded to analyze other features? For example, FOR/NEXT loop matches or OPEN and CLOSE statements together with associated file usage? Perhaps, but I think not. Whether or not it's a good idea, BASIC allows a single FOR statement to be matched with more than one NEXT (and vice versa, for that matter). Files can be opened, closed and used with variable logical file numbers—for example, PRINT#X, "HELLO"—so that a single file's activity is difficult to trace. Cross-Ref wasn't constructed to follow the logic of your program, only the mechanics. You should find Cross-Ref a very useful programming support tool. You might discover that it leads to better programming.

The programs are set up for normal Commodore printers. If you have a printer that specifically needs a line feed character to be sent, you should modify Cross-Ref64 only as follows:

POKE 3181,10
POKE 3223,10

Program 1: BASIC Portion Of PET/CBM Version

```
100 PRINT"{CLR}CROSS REF":PRINT"
    {SHIFT-SPACE}{4 SPACES}JIM BUTTERFIEL
    D"
115 W=6:IFPEEK(32808)=32THENW=11
120 CLOSE1:INPUT"NAME OF PROGRAM";N$
130 OPEN1,8,3,N$+",P,R":GET#1,X$,Y$:IFX$<
    >CHR$(1)GOTO120
190 SYS1668:CLOSE1:INPUT"PRINTER";Z$:P=3:
    IFASC(Z$)=89THENP=4:W=11
200 OPEN4,P:PRINT#4,"CROSS-REF: ";N$:POKE
    208,W:SYS2102:PRINT#4:CLOSE4
```

Program 2: Loader For PET/CBM ML Portion

```
100 SA=1267:SL=200
110 FOR I=0 TO 8
120 CK=0:AD=SA+(I*120):LN=SL+(I*150)
130 FOR J=0 TO 119
140 READ BY:CK=CK+BY:POKE AD+J,BY
150 NEXT J:READ CV:IF CK<>CV THEN 190
160 NEXT I:PRINT "MACHINE LANGUAGE IS LOA
    DED"
170 POKE 40,1:POKE 41,4:POKE 42,43:POKE 4
    3,9
180 POKE 44,43:POKE 45,9:POKE 46,43:POKE
    {SPACE}47,9:END
190 PRINT "DATA ERROR IN LINES";LN;"-";LN
    +140:STOP
200 DATA 0,0,0,0,0,0,0,0
210 DATA 0,0,0,0,0,0,11,11
220 DATA 11,11,11,11,11,11,11,11
230 DATA 11,11,11,11,11,11,11,11
240 DATA 11,11,11,11,11,11,11,11
250 DATA 11,11,11,11,11,11,11,5
260 DATA 11,3,3,3,11,4,11,11
270 DATA 11,9,11,11,11,2,2,2
280 DATA 2,2,2,2,2,2,2,8
290 DATA 11,11,11,11,11,11,1,1
300 DATA 1,1,1,1,1,1,1,1
310 DATA 1,1,1,1,1,1,1,1
320 DATA 1,1,1,1,1,1,1,1
330 DATA 11,11,11,11,11,11,11,11
340 DATA 11,11,11,11,11,11,11,11
345 DATA 774
350 DATA 11,11,11,11,11,11,11,11
360 DATA 11,11,11,11,11,11,11,11
370 DATA 11,11,11,11,11,11,11,11
380 DATA 7,11,11,11,11,11,10,10
390 DATA 11,11,10,11,6,11,11,11
400 DATA 11,11,11,11,11,11,11,11
410 DATA 11,11,11,11,11,11,11,11
420 DATA 11,9,11,11,10,11,11,11
430 DATA 11,11,11,11,11,11,11,11
440 DATA 11,11,11,11,11,11,11,11
450 DATA 11,11,11,11,11,11,11,11
460 DATA 11,11,11,11,11,11,11,11
470 DATA 10,11,11,11,11,11,11,11
480 DATA 11,11,11,11,11,11,11,11
490 DATA 11,11,11,11,11,11,11,11
495 DATA 1304
500 DATA 11,11,11,11,11,11,11,11
510 DATA 11,11,11,11,11,11,11,11
520 DATA 11,11,11,11,11,11,11,11
530 DATA 11,11,11,11,11,0,12,12
540 DATA 12,12,12,12,12,12,12,12
550 DATA 12,0,224,72,12,12,24,36
560 DATA 48,12,12,60,12,0,24,24
570 DATA 24,24,12,24,24,24,24,24
580 DATA 24,0,36,36,36,36,36,36
590 DATA 36,36,36,36,36,0,48,48
600 DATA 48,48,48,48,48,12,48,48
610 DATA 48,0,224,212,12,12,24,36
620 DATA 48,12,60,60,12,0,72,72
630 DATA 12,12,24,36,48,12,12,60
640 DATA 12,0,12,212,12,12,24,36
645 DATA 3507
650 DATA 48,12,60,60,12,0,236,236
660 DATA 248,140,24,36,48,12,12,60
670 DATA 12,0,108,108,248,140,24,36
```

```
680 DATA 48,12,12,60,12,0,120,12
690 DATA 12,140,24,36,48,12,12,60
700 DATA 12,162,1,32,198,255,32,54
710 DATA 7,169,0,133,190,169,11,133
720 DATA 191,169,6,133,185,162,13,189
730 DATA 29,9,157,249,10,202,16,247
740 DATA 48,7,32,204,255,96,32,179
750 DATA 7,32,228,255,32,228,255,240
760 DATA 241,169,0,133,192,169,10,133
770 DATA 193,32,228,255,133,90,32,228
780 DATA 255,133,89,162,12,134,184,32
790 DATA 228,255,201,32,240,249,170,189
795 DATA 12998
800 DATA 0,5,168,177,184,16,3,32
810 DATA 11,7,41,127,164,184,133,184
820 DATA 201,84,176,7,192,84,144,3
830 DATA 32,64,7,201,120,208,19,192
840 DATA 120,208,15,142,122,2,32,64
850 DATA 7,174,122,2,169,12,133,184
860 DATA 208,205,201,0,240,160,208,191
870 DATA 41,127,72,201,84,240,20,138
880 DATA 162,0,180,84,192,32,240,7
890 DATA 232,224,5,208,245,240,18,149
900 DATA 84,240,14,138,162,0,180,85
910 DATA 148,84,232,224,4,208,247,133
920 DATA 88,104,96,162,4,169,32,149
930 DATA 84,202,16,251,96,72,165,192
940 DATA 164,193,56,233,7,133,186,176
945 DATA 14445
950 DATA 1,136,132,187,201,0,152,233
960 DATA 10,144,20,160,4,185,84,0
970 DATA 209,186,208,5,136,16,246,48
980 DATA 73,165,186,164,187,208,219,165
990 DATA 192,164,193,133,188,132,189,56
1000 DATA 233,7,176,1,136,133,186,132
1010 DATA 187,201,0,152,233,10,144,21
1020 DATA 160,6,56,177,186,145,188,249
1030 DATA 84,0,136,16,246,144,6,165
1040 DATA 186,164,187,208,214,160,6,185
1050 DATA 84,0,145,188,136,16,248,24
1060 DATA 165,192,105,7,133,192,144,2
1070 DATA 230,193,32,54,7,104,96,96
1080 DATA 165,190,164,191,133,186,132,187
1090 DATA 56,165,192,233,0,141,122,2
1095 DATA 15395
1100 DATA 165,193,233,10,141,123,2,13
1110 DATA 122,2,240,227,24,173,122,2
1120 DATA 101,186,133,190,133,188,173,123
1130 DATA 2,101,187,133,191,133,189,32
1140 DATA 39,8,165,192,56,233,7,164
1150 DATA 193,176,1,136,133,192,132,193
1160 DATA 201,0,152,233,10,144,184,165
1170 DATA 188,164,189,56,233,7,176,1
1180 DATA 136,133,188,132,189,160,6,56
1190 DATA 177,186,145,188,241,192,136,16
1200 DATA 247,144,6,32,39,8,76,250
1210 DATA 7,160,6,177,192,145,188,136
1220 DATA 16,249,48,190,165,186,164,187
1230 DATA 56,233,7,176,1,136,133,186
1240 DATA 132,187,96,162,4,134,84,32
1245 DATA 15168
1250 DATA 201,255,169,0,160,11,133,186
1260 DATA 132,187,160,4,185,84,0,209
1270 DATA 186,208,5,136,16,246,48,34
1280 DATA 169,13,32,210,255,169,10,32
1290 DATA 210,255,160,0,177,186,153,84
1300 DATA 0,32,210,255,200,192,5,144
1310 DATA 243,169,58,32,210,255,169,0
1320 DATA 133,188,230,188,165,188,197,208
1330 DATA 144,22,169,13,32,210,255,169
1340 DATA 10,32,210,255,160,5,169,32
1350 DATA 32,210,255,136,16,248,48,222
1360 DATA 160,5,177,186,133,90,200,177
1370 DATA 186,133,89,32,225,255,164,151
1380 DATA 200,208,248,32,192,8,24,165
1390 DATA 186,164,187,105,7,144,1,200
1395 DATA 16229
1400 DATA 133,186,132,187,197,190,165,187
1410 DATA 229,191,144,134,96,169,0,162
1420 DATA 2,157,122,2,202,16,250,120
1430 DATA 248,160,15,6,89,38,90,162
1440 DATA 2,189,122,2,125,122,2,157
1450 DATA 122,2,202,16,244,136,16,235
1460 DATA 216,88,162,0,169,48,133,189
1470 DATA 134,192,189,122,2,72,74,74
1480 DATA 74,74,9,48,32,16,9,104
1490 DATA 41,15,9,48,224,2,208,2
1500 DATA 198,189,32,16,9,166,192,232
1510 DATA 224,3,144,220,96,197,189,208
1520 DATA 4,169,32,208,2,198,189,76
1530 DATA 210,255,0,0,0,0,0,0
1540 DATA 0,78,79,78,69,32,0,0
1545 DATA 12648
```

Program 3: MLX Listing For 64

```
2049 :043,008,100,000,153,034,083
2055 :032,067,082,079,083,083,177
2061 :032,082,069,070,034,058,102
2067 :153,034,160,032,032,032,206
2073 :032,074,073,077,032,066,123
2079 :085,084,084,069,082,070,249
2085 :073,069,076,068,034,000,101
2091 :052,008,115,000,087,178,227
2097 :048,054,000,081,008,120,104
2103 :000,160,049,058,133,034,233
2109 :078,065,077,069,032,079,205
2115 :070,032,080,082,079,071,225
2121 :082,065,077,034,059,078,212
2127 :036,000,126,008,130,000,123
2133 :159,049,044,056,044,051,232
2139 :044,078,036,170,034,044,241
2145 :080,044,082,034,058,161,044
2151 :035,049,044,088,036,044,143
2157 :089,036,058,139,088,036,043
2163 :179,177,199,040,049,041,032
2169 :137,049,050,048,000,176,069
2175 :008,190,000,158,050,054,075
2181 :057,050,058,160,049,058,053
2187 :133,034,080,082,073,078,107
2193 :084,069,082,034,059,090,051
2199 :036,058,080,178,051,058,100
2205 :139,198,040,090,036,041,189
2211 :178,056,057,167,080,178,111
2217 :052,058,087,178,049,049,130
2223 :000,224,008,200,000,159,254
2229 :052,044,080,058,152,052,107
2235 :044,034,067,082,079,083,064
2241 :083,045,082,069,070,058,088
2247 :032,034,059,078,036,058,240
2253 :151,049,057,048,044,087,129
2259 :058,158,051,049,051,054,120
2265 :058,152,052,058,160,052,237
2271 :000,000,000,000,000,000,223
2277 :000,000,000,000,000,000,229
2283 :000,000,000,000,000,000,235
2289 :000,000,000,000,000,000,241
2295 :000,000,000,000,000,000,247
2301 :000,000,000,000,011,011,019
2307 :011,011,011,011,011,011,069
2313 :011,011,011,011,011,011,075
2319 :011,011,011,011,011,011,081
2325 :011,011,011,011,011,011,087
```

```
2331 :011,011,011,011,011,011,093
2337 :011,005,011,003,003,003,069
2343 :011,004,011,011,011,009,096
2349 :011,011,011,002,002,002,084
2355 :002,002,002,002,002,002,063
2361 :002,008,011,011,011,011,111
2367 :011,011,001,001,001,001,089
2373 :001,001,001,001,001,001,075
2379 :001,001,001,001,001,001,081
2385 :001,001,001,001,001,001,087
2391 :001,001,001,001,011,011,113
2397 :011,011,011,011,011,011,159
2403 :011,011,011,011,011,011,165
2409 :011,011,011,011,011,011,171
2415 :011,011,011,011,011,011,177
2421 :011,011,011,011,011,011,183
2427 :011,011,011,011,011,011,189
2433 :011,011,007,011,011,011,191
2439 :011,011,010,010,011,011,199
2445 :010,011,006,011,011,011,201
2451 :011,011,011,011,011,011,213
2457 :011,011,011,011,011,011,219
2463 :011,011,011,011,011,009,223
2469 :011,011,010,011,011,011,230
2475 :011,011,011,011,011,011,237
2481 :011,011,011,011,011,011,243
2487 :011,011,011,011,011,011,249
2493 :011,011,011,011,011,011,255
2499 :011,011,011,011,011,011,005
2505 :011,011,010,011,011,011,010
2511 :011,011,011,011,011,011,017
2517 :011,011,011,011,011,011,023
2523 :011,011,011,011,011,011,029
2529 :011,011,011,011,011,011,035
2535 :011,011,011,011,011,011,041
2541 :011,011,011,011,011,011,047
2547 :011,011,011,011,011,011,053
2553 :011,011,011,011,011,011,059
2559 :011,000,012,012,012,012,058
2565 :012,012,012,012,012,012,077
2571 :012,000,224,072,012,012,087
2577 :024,036,048,012,012,060,209
2583 :012,000,024,024,024,024,131
2589 :012,024,024,024,024,024,161
2595 :024,000,036,036,036,036,203
2601 :036,036,036,036,036,036,001
2607 :036,000,048,048,048,048,019
2613 :048,048,048,012,048,048,049
2619 :048,000,224,212,012,012,055
2625 :024,036,048,012,060,060,049
2631 :012,000,072,072,012,012,251
2637 :024,036,048,012,012,060,013
2643 :012,000,012,212,012,012,087
2649 :024,036,048,012,060,060,073
2655 :012,000,236,236,248,140,199
2661 :024,036,048,012,012,060,037
2667 :012,000,108,108,248,140,211
2673 :024,036,048,012,012,060,049
2679 :012,000,120,012,012,140,159
2685 :024,036,048,012,012,060,061
2691 :012,162,001,032,198,255,023
2697 :032,054,011,169,000,133,024
2703 :075,169,015,133,076,169,012
2709 :010,133,070,162,013,189,214
2715 :052,013,157,249,014,202,074
2721 :016,247,048,007,032,204,203
2727 :255,096,032,179,011,032,004
2733 :228,255,032,228,255,240,131
2739 :241,169,000,133,077,169,200
2745 :014,133,078,032,228,255,157
2751 :133,093,032,228,255,133,041
2757 :092,162,012,134,069,032,186
2763 :228,255,201,032,240,249,128
2769 :170,189,000,009,168,177,154
2775 :069,016,003,032,011,011,101
2781 :041,127,164,069,133,069,056
2787 :201,084,176,007,192,084,203
2793 :144,003,032,064,011,201,176
2799 :120,208,019,192,120,208,082
2805 :015,142,060,003,032,064,049
2811 :011,174,060,003,169,012,168
2817 :133,069,208,205,201,000,049
2823 :240,160,208,191,041,127,206
2829 :072,201,084,240,020,138,000
2835 :162,000,180,087,192,032,160
2841 :240,007,232,224,005,208,173
2847 :245,240,018,149,087,240,242
2853 :014,138,162,000,180,088,107
2859 :148,087,232,224,004,208,178
2865 :247,133,091,104,096,162,114
2871 :004,169,032,149,087,202,186
2877 :016,251,096,072,165,077,226
2883 :164,078,056,233,007,133,226
2889 :071,176,001,136,132,072,149
2895 :201,000,152,233,014,144,055
2901 :020,160,004,185,087,000,029
2907 :209,071,208,005,136,016,224
2913 :246,048,073,165,071,164,096
2919 :072,208,219,165,077,164,240
2925 :078,133,073,132,074,056,143
2931 :233,007,176,001,136,133,033
2937 :071,132,072,201,000,152,237
2943 :233,014,144,021,160,006,193
2949 :056,177,071,145,073,249,136
2955 :087,000,136,016,246,144,000
2961 :006,165,071,164,072,208,063
2967 :214,160,006,185,087,000,035
2973 :145,073,136,016,248,024,031
2979 :165,077,105,007,133,077,215
2985 :144,002,230,078,032,054,197
2991 :011,104,096,096,165,075,210
2997 :164,076,133,071,132,072,061
3003 :056,165,077,233,000,141,091
3009 :060,003,165,078,233,014,234
3015 :141,061,003,013,060,003,224
3021 :240,227,024,173,060,003,164
3027 :101,071,133,075,133,073,029
3033 :173,061,003,101,072,133,248
3039 :076,133,074,032,039,012,077
3045 :165,077,056,233,007,164,163
3051 :078,176,001,136,133,077,068
3057 :132,078,201,000,152,233,013
3063 :014,144,184,165,073,164,223
3069 :074,056,233,007,176,001,032
3075 :136,133,073,132,074,160,199
3081 :006,056,177,071,145,073,025
3087 :241,077,136,016,247,144,108
3093 :006,032,039,012,076,250,180
3099 :011,160,006,177,077,145,091
3105 :073,136,016,249,048,190,233
3111 :165,071,164,072,056,233,032
3117 :007,176,001,136,133,071,057
3123 :132,072,162,009,181,069,164
3129 :157,080,003,202,016,248,251
3135 :096,162,009,189,080,003,090
3141 :149,069,202,016,248,162,147
3147 :004,134,087,032,201,255,020
3153 :169,000,160,015,133,071,117
3159 :132,072,160,004,185,087,215
3165 :000,209,071,208,005,136,210
3171 :016,246,048,034,169,013,113
3177 :032,210,255,169,032,032,067
3183 :210,255,160,000,177,071,216
```

```
3189 :153,Ø87,ØØØ,Ø32,21Ø,255,Ø86
3195 :2ØØ,192,ØØ5,144,243,169,Ø52
32Ø1 :Ø58,Ø32,21Ø,255,169,ØØØ,Ø85
32Ø7 :133,Ø73,23Ø,Ø73,165,Ø73,114
3213 :197,19Ø,144,Ø22,169,Ø13,1Ø8
3219 :Ø32,21Ø,255,169,Ø32,Ø32,1Ø9
3225 :21Ø,255,16Ø,ØØ5,169,Ø32,216
3231 :Ø32,21Ø,255,136,Ø16,248,Ø32
3237 :Ø48,222,16Ø,ØØ5,177,Ø71,Ø8Ø
3243 :133,Ø93,2ØØ,177,Ø71,133,21Ø
3249 :Ø92,Ø32,225,255,24Ø,Ø31,Ø28
3255 :165,2Ø3,Ø1Ø,Ø1Ø,144,245,192
3261 :Ø32,215,Ø12,Ø24,165,Ø71,196
3267 :164,Ø72,1Ø5,ØØ7,144,ØØ1,176
3273 :2ØØ,133,Ø71,132,Ø72,197,238
3279 :Ø75,165,Ø72,229,Ø76,144,2ØØ
3285 :131,Ø96,169,ØØØ,162,ØØ2,ØØ5
3291 :157,Ø6Ø,ØØ3,2Ø2,Ø16,25Ø,139
3297 :12Ø,248,16Ø,Ø15,ØØ6,Ø92,Ø98
33Ø3 :Ø38,Ø93,162,ØØ2,189,Ø6Ø,ØØ7
33Ø9 :ØØ3,125,Ø6Ø,ØØ3,157,Ø6Ø,133
3315 :ØØ3,2Ø2,Ø16,244,136,Ø16,Ø92
3321 :235,216,Ø88,162,ØØØ,169,Ø95
3327 :Ø48,133,Ø74,134,Ø77,189,142
3333 :Ø6Ø,ØØ3,Ø72,Ø74,Ø74,Ø74,1Ø6
3339 :Ø74,ØØ9,Ø48,Ø32,Ø39,Ø13,226
3345 :1Ø4,Ø41,Ø15,ØØ9,Ø48,224,2Ø2
3351 :ØØ2,2Ø8,ØØ2,198,Ø74,Ø32,Ø27
3357 :Ø39,Ø13,166,Ø77,232,224,Ø12
3363 :ØØ3,144,22Ø,Ø96,197,Ø74,ØØ1
3369 :2Ø8,ØØ4,169,Ø32,2Ø8,ØØ2,152
3375 :198,Ø74,Ø76,21Ø,255,ØØØ,Ø92
3381 :ØØØ,ØØØ,ØØØ,ØØØ,ØØØ,ØØØ,Ø53
3387 :Ø78,Ø79,Ø78,Ø69,Ø32,ØØØ,139
3393 :ØØØ,Ø13,Ø13,Ø13,Ø13,Ø13,13Ø
```

©

C. Regena

File Processing
Part 3

This month C. Regena concludes her three-part discussion on creating data files.

A Birthday List

Program 1 prints a birthday list of the students in a class. The same data file is used, and the information is arranged in order by birthdate. Line 180 is the OPEN statement for the printer (use your own printer configuration). Line 190 is the OPEN statement for the disk drive to read in information.

Line 210 reads in the date—again, in the same order that the items were saved. We will ignore some of the information, but all the items must be read in order. Line 250 combines several of the items into one variable T$. The birthday BD and T$ are actually arrays, so the items may be sorted. Lines 280–350 contain the sorting procedure to sort by birthday.

Line 360 and lines 510–560 print the header. Lines 370–480 print the information. Lines 380–400 print the month and day from the BD number that was saved. Line 410 prints a blank line between months. Lines 420–450 use POS and SEG$ to separate the T$ item back into its parts, then line 460 prints the information in columns using the IMAGE statement of line 200.

The Report Writer

Program 2 generates reports using the data saved in Program 1 of Part 2 (April 1984). Lines 160–200 present the option to print the report for one of the reading groups or for the whole class.

These reports will use a 132-column line, or *compressed print* (16.5 characters per inch). Line 210 OPENs device #1 for the printer. The previous reports used an 80-column line, which is the default value for most printers. VARIABLE 132 is used to designate a longer line before a carriage return. Line 230 sets *my* printer (TI 825, which is like the TI 840) to use compressed print. You will probably need a different command.

Some printers can use a certain CHR$ number. Other printers may require you to set certain hardware switches. I have used compressed print and the 132-column line so more can fit on the one line. The other two reports in this program may be printed with the regular printing.

Line 240 is the OPEN statement to read the data from the data file created by Program 1 (Part 2, April). Again, the variables are in the same order as they were saved. Line 280 checks for the end of the file. Lines 290–300 check to see if a particular group was chosen or if the whole class is to be printed. Lines 310–480 then print the first report. The student's R$ tally is separated using SEG$. Line 360 and line 410 are used to print information if only part of the ten weeks is used. If you have a different number of weeks in your report, you can change the 10 in lines 130, 410, 520, 560, 600, and 670, and the titles in lines 140 and 930–950.

Total Values

Variable names starting with T are total values. Lines 440–450 print total presentations divided by total possible weeks and the individual's percentage. Lines 500–630 print the totals for each week.

A bar graph report is printed in lines 640–700. Each asterisk represents a report, and the appropriate number of asterisks is printed for each week as a graph.

The final report in this program is to rank the students from high score to low score by percentage. Lines 720–780 contain the sort routine. The percentages were stored in the P array with the corresponding names in NN$. Lines 790–850 print the percents and names. Line 810 and the subroutine in lines 1000–1150 alphabetize the names of all students who have a zero score.

Console BASIC

You can, in fact, do file processing without Extended BASIC and all the peripherals. I used Extended BASIC mainly because of the ease in formatting the printing—lining up the columns. In regular console BASIC you can use subroutines to line up columns of numbers and the TAB function to start the columns right. See my January 1984 column in COMPUTE! for some suggestions on formatting and screen scrolling.

To use a printer you need the RS-232 interface plus the printer. A number of different name brands of printers can be used with the TI-99/4A. The printer manuals should tell you what features the printer has and how to control different features, such as the number of characters per inch and form feeds. Using the printer and RS-232 manuals, you can determine the appropriate printer configuration necessary for the OPEN statement. Without a printer, you can print on the screen—just keep within the 28 print columns and print a screen at a time or use a scrolling delay method so you can read the information as it is printed.

To use a disk drive you also need the disk controller or disk controller card for the Peripheral Expansion box. The disk controller or card comes with a command module and a manual that describes disk procedures. To use a cassette, simply change the "DSK1.---" statements to "CS1" and change the VARIABLE to FIXED. The cassette system works fine—it just takes longer than the disk system.

Program 1: Birthday List

```
80 REM TI EXTENDED BASIC
90 REM DISK, PRINTER
100 REM BIRTHDAY LIST
110 CALL CLEAR
120 DISPLAY AT(12,5):"BIRTHDAY LIST
    "
130 OPTION BASE 1
140 DIM T$(140),BD(140),M$(12)
150 FOR I=1 TO 12 :: READ M$(I):: N
    EXT I
160 DATA JAN,FEB,MAR,APR,MAY,JUN,JU
    L,AUG,SEP,OCT,NOV,DEC
170 L=0 :: I=1 :: L$="---"
180 OPEN #1:"RS232.BA=600"
190 OPEN #3:"DSK1.SAMPLE",INTERNAL,
    INPUT ,VARIABLE 192
200 IMAGE "{5 SPACES}### ##
    {3 SPACES}######################
    ##  ########{3 SPACES}#########
    #################"
210 INPUT #3:G,N$,F$,A$,P$,BD(I),R$
    ,C$
220 IF C$="MOVED" THEN 210
230 IF N$="ZZZ" THEN 270
240 IF P$="" THEN P$="{4 SPACES}"
250 T$(I)=F$&" "&N$&"/"&P$&A$
260 I=I+1 :: GOTO 210
270 I=I-1 :: CLOSE #3
280 DISPLAY AT(23,1):"SORTING"
290 B=1
300 B=2*B :: IF B<=I THEN 300
310 B=INT(B/2):: IF B=0 THEN 360
320 FOR J=1 TO I-B :: C=J
330 D=C+B :: IF BD(C)<=BD(D)THEN 35
    0
340 AA=BD(C):: TT$=T$(C):: BD(C)=BD
    (D):: T$(C)=T$(D):: BD(D)=AA ::
     T$(D)=TT$ :: C=C-B :: IF C>0 T
    HEN 330
350 NEXT J :: GOTO 310
360 GOSUB 510
370 FOR J=1 TO I
380 IF BD(J)=0 THEN B$="---" :: D=0
     :: GOTO 420
390 BD$=STR$(BD(J)):: M=VAL(SEG$(BD
    $,1,LEN(BD$)-2)):: D=VAL(SEG$(B
    D$,LEN(BD$)-1,2))
400 B$=M$(M):: IF B$=L$ THEN 420
410 L=L+1 :: PRINT #1 :: L$=B$
420 P=POS(T$(J),"/",8)
430 N$=SEG$(T$(J),1,P-1)
440 P$="586-"&SEG$(T$(J),P+1,4)
450 A$=SEG$(T$(J),P+5,LEN(T$(J))-P+
    4)
460 PRINT #1,USING 200:B$,D,N$,P$,A
    $
470 L=L+1 :: IF L=48 THEN PRINT #1:
    CHR$(12):: L=0 :: GOSUB 510
480 NEXT J
490 PRINT #1:CHR$(12)
500 STOP
510 PRINT #1:TAB(34);"SAMPLE CLASS"
520 PRINT #1: :TAB(34);"BIRTHDAY LI
    ST"
530 PRINT #1: :TAB(33);"APRIL 15, 1
    984"
540 PRINT #1: : :TAB(5);"BIRTHDAY";
    TAB(15);"NAME";TAB(41);"PHONE";
    TAB(54);"ADDRESS"
550 PRINT #1:TAB(5);"--------";TAB(
    15);"----";TAB(41);"-----";TAB(
    54);"-------": : :
560 RETURN
570 END
```

Program 2: Report Writer

```
80 REM TI EXTENDED BASIC
90 REM DISK, PRINTER
100 REM REPORT WRITER
110 OPTION BASE 1
120 DIM D$(10),T(10),TT(10),NN$(140
    ),P(140)
130 FOR I=1 TO 10 :: READ D$(I):: N
    EXT I
140 DATA JAN 1,JAN 8,JAN 15,JAN 22,
    JAN 29,FEB 5,FEB 12,FEB 19,FEB
    26,MAR 4
150 DISPLAY AT(4,6)ERASE ALL:"REPOR
    T WRITER"
160 DISPLAY AT(7,3):"CHOOSE:" :: DI
    SPLAY AT(8,5):"1 GROUP 1" :: DI
    SPLAY AT(9,5):"2 GROUP 2"
170 DISPLAY AT(10,5):"3 GROUP 3" ::
     DISPLAY AT(12,5):"4 COMPLETE C
    LASS"
180 CALL KEY(0,KEY,ST)
190 IF KEY<49 OR KEY>52 THEN 180
200 G1=KEY-48 :: CALL HCHAR(7,3,32,
    192)
210 OPEN #1:"RS232.BA=600",VARIABLE
     132
220 REM SET FOR COMPRESSED PRINT
230 ESC$=CHR$(27):: PRINT #1:ESC$&"
    P"&"D"&ESC$&"\"
240 OPEN #3:"DSK1.SAMPLE",INTERNAL,
    INPUT ,VARIABLE 192
```

```
250 I=Ø :: L$="A"
260 GOSUB 880 :: GOSUB 930
270 INPUT #3:G,N$,F$,A$,P$,BD,R$,C$
280 IF N$="ZZZ" THEN 490
290 IF G1=4 THEN 310
300 IF G1<>G THEN 270
310 IF SEG$(C$,1,5)="AUDIT" THEN 270
320 C$=SEG$(N$,1,1):: IF L$<>C$ THEN L$=C$ :: PRINT #1 :: L=L+1
330 PRINT #1:TAB(1Ø);N$;", ";F$;TAB(44);
340 TA=Ø :: TP=Ø
350 IF R$="" THEN R$="ØØØØØØØØØØ"
360 FOR J=1 TO LEN(R$)
370 A$=SEG$(R$,J,1):: IF A$="1" THEN TA=TA+1
380 IF A$="1" OR A$="Ø" THEN TP=TP+1 :: T(J)=T(J)+VAL(A$):: TT(J)=TT(J)+1
390 PRINT #1:A$;"{4 SPACES}";
400 NEXT J
410 FOR JJ=J TO 1Ø :: PRINT #1:"{5 SPACES}";:: NEXT JJ
420 I=I+1 :: NN$(I)=F$&" "&N$
430 IF TP=Ø THEN P(I)=Ø :: GOTO 450
440 P(I)=INT(TA*100/TP)
450 PRINT #1,USING "{16 SPACES}##/##{5 SPACES}###":TA,TP,P(I)
460 L=L+1 :: IF L=48 THEN GOSUB 870 :: GOSUB 930
470 IF A$="-" THEN I=I-1
480 GOTO 270
490 GOSUB 950
500 PRINT #1
510 PRINT #1:TAB(1Ø);"REPORTS: ";TAB(42);
520 FOR J=1 TO 10
530 PRINT #1,USING "###  ":T(J);
540 TAT=TAT+T(J):: NEXT J
550 PRINT #1: :TAB(1Ø);"ENROLLED: ";TAB(42);
560 FOR J=1 TO 1Ø
570 PRINT #1,USING "###  ":TT(J);
580 TE=TE+TT(J):: NEXT J
590 PRINT #1: : :TAB(1Ø);"PERCENT REPORTS:";TAB(42);
600 FOR J=1 TO 1Ø
610 PRINT #1,USING "###  ":T(J)*100/TT(J);
620 NEXT J
630 PRINT #1:TAB(120);INT(TAT*100/TE)
640 GOSUB 870
650 PRINT #1: :TAB(1Ø);"DATE";TAB(3Ø);"REPORTS"
660 PRINT #1:TAB(1Ø);"----";TAB(3Ø);"-------": :
670 FOR J=1 TO 1Ø
680 A$=RPT$("*",T(J))
690 PRINT #1: :TAB(1Ø);D$(J);TAB(3Ø);T(J);"  ";A$
700 NEXT J
710 GOSUB 870
720 B=1
730 B=2*B :: IF B<=I THEN 730
740 B=INT(B/2):: IF B=Ø THEN 790
750 FOR J=1 TO I-B :: C=J
760 D=C+B :: IF P(C)<=P(D)THEN 780
770 AA=P(C):: AA$=NN$(C):: P(C)=P(D):: NN$(C)=NN$(D):: P(D)=AA :: NN$(D)=AA$ :: C=C-B :: IF C>Ø THEN 760
780 NEXT J :: GOTO 740
790 GOSUB 970
800 FOR J=I TO 1 STEP -1
810 IF P(J)=Ø AND FL=Ø THEN GOSUB 1000
820 PRINT #1:TAB(46);
830 PRINT #1,USING "###{8 SPACES}########################":P(J),NN$(J)
840 L=L+1 :: IF L=48 THEN GOSUB 870 :: GOSUB 970
850 NEXT J
860 STOP
870 PRINT #1:CHR$(12)
880 PRINT #1:TAB(58);"SAMPLE CLASS"
890 IF G1=4 THEN 910
900 PRINT #1: :TAB(60);"GROUP";G1
910 PRINT #1: :TAB(53);"BOOK REPORTS PRESENTED"
920 PRINT #1: :TAB(57);"FIRST TERM 1984" :: RETURN
930 PRINT #1: : :TAB(43);"JAN  JAN  JAN  JAN  JAN  FEB  FEB  FEB  FEB  MAR"
940 PRINT #1:TAB(1Ø);"NAME";TAB(43);" 1{4 SPACES}8{3 SPACES}15{3 SPACES}22{3 SPACES}29{4 SPACES}5{3 SPACES}12{3 SPACES}19{3 SPACES}26{4 SPACES}4";TAB(11Ø);"TOTAL";TAB(121);"%"
950 PRINT #1:TAB(1Ø);"----";TAB(43);"---  ---  ---  ---  ---  ---  ---  ---  ---  ---";TAB(110);"-----";TAB(120);"---": :
960 L=Ø :: RETURN
970 PRINT #1: : :TAB(44);"PERCENT";TAB(57);"NAME"
980 PRINT #1:TAB(44);"-------";TAB(57);"----": : :
990 L=Ø :: RETURN
1000 FOR K=1 TO J
1010 S=POS(NN$(K)," ",1)
1020 S1=POS(NN$(K)," ",S+1):: IF S1=Ø THEN 1030 ELSE S=S1
1030 NN$(K)=SEG$(NN$(K),S+1,LEN(NN$(K))-S)&", "&SEG$(NN$(K),1,S-1)
1040 NEXT K
1050 B=1
1060 B=2*B :: IF B<=J THEN 1060
1070 B=INT(B/2):: IF B=Ø THEN 1120
1080 FOR K=1 TO J-B :: C=K
1090 D=C+B :: IF NN$(C)>=NN$(D)THEN 1110
1100 A$=NN$(C):: NN$(C)=NN$(D):: NN$(D)=A$ :: C=C-B :: IF C>Ø THEN 1090
1110 NEXT K :: GOTO 1070
1120 FOR K=1 TO J :: S=POS(NN$(K),",",1)
1130 NN$(K)=SEG$(NN$(K),S+2,LEN(NN$(K))-S+1)&" "&SEG$(NN$(K),1,S-1)
1140 NEXT K
1150 FL=1 :: RETURN
1160 END
```

©

MACHINE LANGUAGE

Jim Butterfield, Associate Editor

A Program Critique

Part 2

This month we continue with comments on Bud Rasmussen's program to copy files on the Commodore 64 with a single disk unit. At this point the program has obtained a filename. The filename is kept in two forms: the short form ("FILENAME") and the longer form for writing ("FILENAME,P,W"). We will use the short form when we open the file for reading.

In this session, we'll track the mnemonics that open the error channel, initialize the disk, and input the file into RAM memory.

```
                        ;
                        ; DISK I/O ROUTINE
                        ;
                        ;
C18A  A9 00        DIOR LDA #0      ;CLEAR
C18C  8D 60 03          STA ISF     ;INPUT STAT FLAG
C18F  8D 61 03          STA IEC     ;INPUT ERR CODE
                        ;
```

This is probably overkill. The flags should be zeroed close to where they are used, if necessary.

```
C192  A2 22             LDX #IPBML  ;PRINT
C194  A0 C1             LDY #>IPBM  ;'INPUT
C196  A9 AD             LDA #<IPBM  ;PHASE BEGUN'
C198  20 75 C1          JSR PR      ;MSG
                        ;
```

A Friendly Message

In keeping with the friendly style, a message is printed telling the user what's going on. We'll find the message in-line very shortly.

```
C19B  A9 0F             LDA #15     ;SET
C19D  A2 08             LDX #8      ;COMMAND
C19F  A0 0F             LDY #15     ;CHANNEL
C1A1  20 BA FF          JSR SETLFS
C1A4  20 C0 FF          JSR OPEN    ;OPEN COMMD CH
                        ;
```

The command channel is opened. This is quite important: We'll get all our error messages from this channel. It should always be opened before other disk activities are started.

```
C1A7  20 3F C4          JSR ID      ;INIT DISK
C1AA  4C CF C1          JMP SNI     ;GOTO SET NAME
                                     INPUT
```

We send the initialize command to the disk over the command channel. This is not vital, but a good precaution. It's a subroutine within the program; we'll meet it much later.

We need to jump over the message to continue with the program. Here's the message:

```
                        ;
                        ;
                        ; INPUT PHASE BEGUN MESSAGE
                        ;
                        ;
C1AD  0D 0D 12     IPBM     .BYTE$0D,$0D,$12
C1B0  2A 2A 2A              .ASC "*** INPUT PHASE BEGUN ***"
C1CD  0D 0D                 .BYTE$0D,$0D
                   IPBML    =   *-IPBM
                        ;
```

Now we're ready to open the input file in preparation for reading it. We use the short name, since the last four characters (,S,W) aren't needed or wanted for an input file.

```
                        ;
                        ; OPEN INPUT
                        ;
                        ;
C1CF  AD AA 02     SNI  LDA IFNL    ;LOAD INPUT
                                     FNAME LEN
C1D2  A2 40             LDX #<FNA   ;LOAD FILENAME LO
C1D4  A0 03             LDY #>FNA   ;LOAD FILENAME HI
C1D6  20 BD FF          JSR SETNAM
                        ;
```

We're doing things backwards from the equivalent BASIC coding. If we code OPEN 2,8,2,"HOTDOG" in BASIC, we've now placed the "HOTDOG" part of the command. Now let's put in the 2,8,2 sequence:

```
                        ;
                        ; SET LOGICAL FILE (INPUT)
                        ;
C1D9  A9 02        SLFI LDA #2      ;LOAD LOGICAL
                                     FILE #
C1DB  A2 08             LDX #8      ;LOAD DEVICE
```

```
                                        ADDRESS
C1DD A0 02          LDY #2           ;LOAD SEC.
                                        ADDRESS
C1DF 20  BA FF      JSR  SETLFS
               ;
```

And finally, the OPEN itself:

```
               ;
               ; OPEN FILE (INPUT)
               ;
               ;
C1E2  20  C0  FF  OFI    JSR  OPEN
               ;
```

Error Check

Now we'll check to see if the OPEN took place without error:

```
C1E5  A5  90          LDA  IOS      ;TEST
C1E7  F0  0B          BEQ  OCI      ;STATUS
C1E9  8D  60  03      STA  ISF      ;STORE STATUS
                                      FLAG
C1EC  A9  01          LDA  #1       ;SET/STORE
C1EE  8D  61  03      STA  IEC      ;ERROR CODE
C1F1  4C  4F  C2      JMP  IE       ;INPUT ERROR
               ;
```

Location $90—called IOS here—is the familiar BASIC ST flag. If it's zero, we are OK and can proceed to read the file. If not, we must advise, abort, or take other appropriate action.

But this flag is not enough. ST, or hex 90, tells us only if the transfer of information (in this case, filename) has been passed to the disk correctly. After the information gets to the disk, there may be other problems.

If the file does not exist, or for any other reason cannot be opened, the disk will know there's an error; but the computer will not. The computer must ask the disk to deliver information on possible errors over its command channel. The command channel is open and ready to receive this data (we opened 15, remember), but we must ask for it.

To do the job right, we must think about coding along the following lines:

```
LDX  #15      ; command channel
JSR  $FFC6    ; input
JSR  $FFE4    ; get a character
PHA           ; stash it
JSR  $FFCC    ; close channel
PLA           ; unstash character
CMP  #$30     ; is it 0?
BNE  ERROR    ; nope, we have problem
```

A Better Way

The above is minimum coding. It would be better to create a more elaborate subroutine which brings in the whole message from the error channel and stores it in memory. (The message would end with $0D, the Return character.) Then we could check the first character for $30 (ASCII zero, start of the OK message); if not, we'd be able to print the whole error message.

Here comes the coding for a good OPEN:

```
               ; OPEN CHANNEL (INPUT)
               ;
               ;
C1F4  A2  02      OCI   LDX  #2       ;OPEN
C1F6  20  C6  FF        JSR  CHKIN    ;CHANNEL #2
               ;
C1F9  A5  90            LDA  IOS      ;TEST
C1FB  F0  0B            BEQ  LBSA     ;STATUS
C1FD  8D  60  03        STA  ISF      ;STORE STATUS
                                        FLAG
C200  A9  02            LDA  #2       ;SET/STORE
C202  8D  61  03        STA  IEC      ;ERROR CODE
C205  4C  4F  C2        JMP  IE       ;INPUT ERROR
               ;
```

I wish the comments said "connect channel" rather than "open channel." The OPEN (as we know it in BASIC) has been performed successfully. Now, we're establishing a connection to the input file preparatory to reading.

```
               ;
               ; LOAD BUFFER START ADDRESS
               ;
               ;
C208  A9  00      LBSA  LDA  #0       ;LOAD BFR
C20A  85  FB            STA  BAL      ;ADDR LO
C20C  AD  3D  C4        LDA  SP       ;LOAD BFR
C20F  85  FC            STA  BAH      ;ADDR HI
               ;
C211  A0  00            LDY  #0       ;BUFFER INDEX=0
               ;
```

Just before reading, we set up the memory address into which we will start to read. The low part of the address is zero; the high part is stored as a constant in the program (SP undoubtedly stands for Start Page). Immediate addressing could be used to set the start page if preferred.

```
               ;
               ; INPUT LOOP
               ;
               ;
C213  20  CF  FF  IL    JSR  CHRIN    ;GET CHARACTER
C216  91  FB            STA  (BAL),Y  ;STORE CHARACTER
C218  E6  FB            INC  BAL      ;INCR LO BYTE
C21A  D0  0C            BNE  TIS      ;IF NOT 0, TEST STAT
C21C  E6  FC            INC  BAH      ;INCR HI BYTE
C21E  A5  FC            LDA  BAH      ;LOAD HI BYTE AND,
C220  CD  3E  C4        CMP  EP       ;CHECK FOR END
                                        ADDR
C223  90  03            BCC  TIS      ;IF LO, TEST STAT
C225  4C  3B  C2        JMP  DSP
```

CHRIN Or CHRGET

Rasmussen uses the CHRIN routine ($FFCF) to get from the file. I prefer CHRGET ($FFE4), but the difference is minor with files. Either call gets from the file rather than keyboard/screen because we have switched the input channel with our call to CHKIN ($FFC6).

Some programmers would prefer to step the Y register through its range rather than change the indirect address each time. In principle, the Y register technique is faster; but in this case, it's doubtful that the speed difference could be observed. Timing of this whole section is governed almost totally by disk speed.

The program checks carefully to make sure that the data does not overrun the memory space available.

```
                    ;
                    ; TEST INPUT STATUS
                    ;
C228  A5 90      TIS    LDA  IOS      ;LOAD STATUS
C22A  F0 E7             BEQ  IL       ;IF 0, CARRY ON
C22C  C9 40             CMP  #EOFI    ;TEST FOR
C22E  F0 23             BEQ  EOF      ;EOF
C230  8D 60 03          STA  ISF      ;STORE STATUS
                                       FLAG
C233  A9 03             LDA  #3       ;SET/STORE
C235  8D 61 03          STA  IEC      ;ERROR CODE
C238  4C 4F C2          JMP  IE       ;INPUT ERROR
                    ;
```

Again we test the ST status byte (IOS); in this case, we're primarily interested in an end-of-file signal which would be flagged by a value of hex 40 (decimal 64) in ST.

Once again, the error routines are quite elaborate. It's my opinion that there is little need to check the disk error channel during the read phase; error notices will wait until we ask for them at end of file.

Opening The File

If we run out of memory, we come to DSP:

```
                    ;
                    ; DECREMENT START PG BY HEX 10
                    ; AND TRY AGAIN,
                    ; TO GIVE YOU 16 MORE BLKS.
                    ;
                    ;
C23B  38         DSP    SEC
C23C  AD 3D C4          LDA  SP       ;LOAD START PG
C23F  E9 10             SBC  #H10     ;SUBT HEX 10
C241  8D 3D C4          STA  SP       ;STORE IT BACK
C244  20 CC FF          JSR  CLRCHN   ;CLEAR CHANNEL
C247  A9 02             LDA  #2       ;SET CH 2
C249  20 C3 FF          JSR  CLOSE    ;FOR CLOSE
C24C  4C CF C1          JMP  SNI      ;START ALL OVER
```

I'm not sure what is going on here. The coding intention is this: If it doesn't fit, allocate an extra 4K and try again.

An Endless Loop

This is puzzling. If the 4K was available, why not make it available in the first read and save the trouble?

There's also a pitfall here. Suppose we allocate the extra 4K, and the program still doesn't fit into memory. We'll end up in an endless loop, since we will come back to DSP, do it again, and so on, and so on.

I'd prefer to allocate as much memory as possible right away, and quit if the program doesn't fit.

```
                    ;
                    ; INPUT ERROR
                    ;
                    ;
C24F  20 E7 FF   IE     JSR  CLALL    ;CLOSE ALL FILES
C252  00                BRK
                    ;
```

This is a programmer's error termination. The program will stop and break to the monitor, if there is a monitor in place. The programmer can then examine memory locations to see what the trouble is.

If there is not a monitor in the machine, the program will terminate with a READY statement and no other explanation.

Extra Work

For general use, the program would benefit from additional work in this area so that the user would see a meaningful message. This is almost out of character: The messages are so well presented in other parts of the program that their absence here is very noticeable indeed.

```
                    ;
                    ; END OF FILE
                    ;
                    ;
C253             EOF    =    *
                    ;
C253  A5 FB             LDA  BAL      ;SAVE
C255  85 FD             STA  EAL      ;LAST
C257  A5 FC             LDA  BAH      ;ADDRESS
C259  85 FE             STA  EAH      ;OF FILE
C25B  20 CC FF          JSR  CLRCHN   ;CLEAR CHANNEL
                    ;
```

Wrapping It Up

The end address (plus one, of course) is stored away, and the file disconnected. I would check the disk error channel at this point. Any errors that may have accumulated during the input phase will be waiting.

Now we may close the file and print an advisory message:

```
C25E  A9 02             LDA  #2       ;SET CH 2
C260  20 C3 FF          JSR  CLOSE    ;FOR CLOSE
                    ;
C263  A2 88             LDX  #IPFML   ;PRINT
C265  A0 C2             LDY  #>IPFM   ;'INPUT
C267  A9 6F             LDA  #<IPFM   ;PHASE FINISHED'
C269  20 75 C1          JSR  PR       ;MSG
                    ;
C26C  4C F7 C2          JMP  SOP      ;GOTO START OUT
                                       PHASE
                    ;
                    ;
                    ; INPUT PHASE FINISHED MESSAGE
                    ;
                    ;
C26F  12         IPFM   .BYTE$12
C270  20 20 49          .ASC " INPUT PHASE FINISHED.   "
C28F  0D 0D 12          .BYTE$0D,$0D,$12
C292  20 20 52          .ASC " REMOVE INPUT DISKETTE. "
C2B1  0D 0D 12          .BYTE$0D,$0D,$12
C2B4  20 20 49          .ASC " INSERT OUTPUT DISKETTE. "
C2D3  0D 0D 12          .BYTE$0D,$0D,$12
C2D6  20 20 50          .ASC " PRESS RETURN KEY WHEN
                        READY. "
C2F5  0D 0D             .BYTE$0D,$0D
C2F7             IPFML  =    *-IPFM
                    ;
                    ;
                    ; START OUTPUT PHASE
                    ;
C2F7             SOP    =    *
```

The input phase is complete. Next time, we'll take a look at output. ©

Atari Softkey

Thomas A. Marshall

This utility allows you to GOTO any line in a program while it's running, simply by pressing a console key. See the "Automatic Proofreader" article on page 180 before typing in programs.

To access the OPTION, SELECT, and START keys on the Atari keyboard console, you can use the following BASIC program:

```
MK Ø GOTO 1Ø
GC 1 ? "OPTION":GOTO 2Ø
EK 2 ? "SELECT":GOTO 2Ø
BJ 3 ? "START ":GOTO 2Ø
FG 1Ø ? "This is a demonstration of th
      e"
GP 11 ? "use of Atari's console keys."
HK 2Ø IF PEEK(53279)=3 THEN GOTO 1
HO 3Ø IF PEEK(53279)=5 THEN GOTO 2
IB 4Ø IF PEEK(53279)=6 THEN GOTO 3
AA 5Ø GOTO 2Ø
```

However, this requires that the computer be tied up in a loop, lines 20 to 50.

A much better way to accomplish the same thing is for a machine language program to check the console keys during the *vertical blank period*. (This is the time that the television's electron beam ends at the lower right corner of the screen until it begins again at the top left corner of the screen.) If a console key is pressed, the machine language program will execute a "GOTO line number" where the line number corresponds to the following keys pressed:

GOTO 1 for OPTION
GOTO 2 for SELECT
GOTO 3 for START
GOTO 4 for SHIFT & OPTION
GOTO 5 for SHIFT & SELECT
GOTO 6 for SHIFT & START

Note that we have doubled the effective number of console keys by adding the SHIFT key. Using this technique, the BASIC programmer can go directly to any portion of his program without stopping the program and typing GOTO *line number*.

An Automatic RUN

If you are really lazy, you can have the BASIC line, 3 RUN, so that your BASIC program will RUN when the START key is pressed, regardless of whether the BASIC program was running beforehand or not.

Program 1 creates an AUTORUN.SYS file. Note that this file resets the memory location, MEMLO, that points to the beginning of a BASIC program. Thus, the vertical blank machine language routine resides safely below the BASIC program. The drawback to this technique is that the machine language program will be erased when you go to DOS.

Also Autoruns

An additional feature included in the disk version of "Atari Softkey" is the ability to autorun any BASIC program saved on the disk. Program 2 is a demonstration program which will be RUN automatically by the AUTORUN.SYS file. So, Program 2 should be saved on the disk with the filename as in the AUTORUN.SYS file. Program 2 currently has the filename GOTO.BAS, defined in line 40 of Program 1 by F$="RUN D:GOTO.BAS".

The Tape Version

For Atari owners who do not have a disk drive, Program 3 POKEs Softkey into page 6. You need to initialize the machine language (ML) routine with the USR statement in line 120. Program 3 is essentially the same as Program 1, but with the autorun feature removed. Again, whenever the console keys are pressed, lines 1–6 in Program 2 will be executed as described above.

However, remember that if there is no line number in the BASIC program corresponding to the console key pressed, an "ERROR 12", line not found, will occur.

The ML program is initialized by placing the low and high address of the start of the ML program into memory addresses 736–737 (RUNAD $2E0–$2E1). Upon completion of DOS.SYS load, the computer will run the ML program pointed to by this address. After resetting several vectors, the ML program sets the Vertical Blank Interrupt (VBI) vector using the deferred mode.

The Deferred Mode

I have used the deferred mode (accumulator = 7), since there are about 20,000 machine cycles available versus about 3800 cycles in the immediate mode (accumulator = 6). Thus, the ML routine checks whether the SHIFT and the console keys are pressed during the vertical blank period. Once the keys are pressed, the ML program jumps to the subroutine that sounds the keyboard click and resets the pointer to the editor routine so that the ML can perform the GOTO *line number* input. It then simulates a press of the BREAK key so that the editor buffer is emptied and the new editor pointers are executed. Once the BASIC G.*line number* is in the editor buffer, the editor pointer is reset. A RETURN, CHR$(155), is placed in the editor buffer to execute the GOTO *line number* statement.

Softkey has many applications. I have found it most useful in a program that required the modification of DATA statements. You can RUN the BASIC program simply by pressing the START key. Another application is to go directly to subroutines without going through a menu selection.

Program 1: Atari Softkey

```
BL 10 REM Atari Softkey
EK 20 GRAPHICS 0:? "Insert a DOS 2.0S
      diskette":? "with DOS.SYS in dri
      ve 1"
FN 30 ? :? "Press RETURN when you have
       done this"
ND 40 DIM F$(18):B=0:F$="RUN D:GOTO.BA
      S":F$(4,4)=CHR$(34):REM 34=ASCII
       FOR "
EM 50 IF PEEK(764)=12 THEN POKE 764,25
      5:GOTO 70
AE 60 GOTO 50
NH 70 ? :? "Now writing the AUTORUN.SY
      S file"
DE 80 TRAP 100:CLOSE #1
KC 90 OPEN #1,8,0,"D:AUTORUN.SYS":TRAP
       4:GOTO 110
FM 100 CLOSE #1:? :? "Can't open AUTOR
       UN.SYS file":END
JH 110 FOR I=1 TO 292:TRAP 180:READ A:
       B=B+A:TRAP 210:PUT #1,A:NEXT I:
       TRAP 40000
BD 120 IF A<>96 THEN 170
KE 130 IF B<>30720 THEN 190
DA 140 FOR I=1 TO 18-LEN(F$):PUT #1,32
       :NEXT I
LA 150 FOR I=LEN(F$) TO 1 STEP -1:PUT
       #1,ASC(F$(I)):NEXT I:CLOSE #1
FN 160 ? :? " DATA ok, write successfu
       l.":END
OE 170 ? :? "There are too many DATA e
       ntries":GOTO 200
DM 180 ? "There are not enough DATA en
       tries":GOTO 200
FP 190 ? :? "Bad number in DATA statem
       ents"
HL 200 CLOSE #1:? "RECHECK the entries
       !":END
BG 210 ? :? :? "Error-";PEEK(195);" wh
       en attempting disk write.":CLOS
       E #1:END
HI 220 REM
FH 230 REM The following is the decimal
KD 240 REM equivalent of the machine
GA 250 REM language.  It must be typed
CA 260 REM perfectly in order to
BB 270 REM function.
HO 280 REM
FI 290 DATA 255,255,0,30,243,30
GP 300 DATA 165,12,141,57,30,165,13,14
       1,58,30,169,56,133,12,169,30,13
       3,13,32,63,30,169,244,141,231,2
       ,169,30,141,232
LF 310 DATA 2,173,243,30,240,10,169,20
       5,141,89,30,169,6,141,90,30,160
       ,105,162,30,169,7,32,92,228,96,
       32,64,21,32
FN 320 DATA 10,30,96,169,85,141,33,3,1
       69,30,141,34,3,96,169,0,141,33,
       3,169,228,141,34,3,96,251,243,5
       1,246,220
HA 330 DATA 30,163,246,51,246,60,246,7
       6,228,243,51,46,71,0,7,169,8,14
       1,31,208,173,31,208,205,104,30,
       240,100,141,104
CI 340 DATA 30,201,7,240,93,141,104,30
       ,173,103,30,208,85,173,104,30,2
       01,3,208,19,169,49,141,100,30,1
       73,15,210,41,8
GD 350 DATA 208,51,169,52,141,100,30,2
       08,44,201,5,208,19,169,50,141,1
       00,30,173,15,210,41,8,208,28,16
       9,53,141,100,30
PF 360 DATA 208,21,201,6,208,32,169,51
       ,141,100,30,173,15,210,41,8,208
       ,5,169,54,141,100,30,169,3,141,
       103,30,32,216
MK 370 DATA 252,32,63,30,169,0,133,17,
       76,98,228,172,103,30,240,9,185,
       99,30,206,103,30,160,1,96,32,74
       ,30,169,155
NJ 380 DATA 160,1,96,18
AK 390 DATA 224,2,225,2,0,30,206,6,255
       ,6
FC 400 DATA 172,243,30,240,9,185,237,6
       ,206,243,30,160,1,96,32,74,30,1
       69,220,141,89,30,169,30,141,90,
       30,169,155,160
HL 410 DATA 1,96
```

Program 2: Atari Softkey Test Program

```
MK 0 GOTO 10
DJ 1 ? " {TAB}OPTION{UP}":END
CB 2 ? " {TAB}SELECT{UP}":END
PA 3 ? " {TAB}START {UP}":END
OH 4 ? " {TAB}SHIFT-OPTION{UP}":END
MP 5 ? " {TAB}SHIFT-SELECT{UP}":END
JO 6 ? " {TAB}SHIFT-START {UP}":END
EO 10 ? "This is a test of"
NM 11 ? "Atari Softkey!"
```

Program 3: Atari Softkey (ML) For Tape Drive Users

```
EA 100 FOR I=0 TO 204:READ A:B=B+A:POK
       E 1536+I,A:NEXT I
```

```
GC 110 IF B<>19990 OR I<>205 THEN ? "R
       echeck DATA statements.":? "The
       y do not correctly total":END
CL 120 A=USR(1536)
EG 200 DATA 104,169,1,133,2,169,6,133,
       3,165,9,9,2,133,9,160,67,162,6,
       169,7,32,92,228,96,169,47,141,3
       3,3
KD 210 DATA 169,6,141,34,3,96,169,0,14
       1,33,3,169,228,141,34,3,96,251,
       243,51,246,182,6,163,246,51,246
       ,60,246,76
CG 220 DATA 228,243,49,46,71,0,7,169,8
       ,141,31,208,173,31,208,205,66,6
       ,240,100,141,66,6,201,7,240,93,
       141,66,6
DM 230 DATA 173,65,6,208,85,173,66,6,2
       01,3,208,19,169,49,141,62,6,173
       ,15,210,41,8,208,51,169,52,141,
       62,6,208
FF 240 DATA 44,201,5,208,19,169,50,141
       ,62,6,173,15,210,41,8,208,28,16
       9,53,141,62,6,208,21,201,6,208,
       32,169,51
JA 250 DATA 141,62,6,173,15,210,41,8,2
       08,5,169,54,141,62,6,169,3,141,
       65,6,32,216,252,32,25,6,169,0,1
       33,17
HH 260 DATA 76,98,228,172,65,6,240,9,1
       85,61,6,206,65,6,160,1,96,32,36
       ,6,169,155,160,1,96
NL 270 ? " Now type in program listing
       "
GB 280 ? " number 2 to demonstrate"
CC 290 ? " Atari Softkey."
```

©

64 EXPLORER

Larry Isaacs

In this month's column we will complete our look at line drawing in the 64's bitmapped graphics mode. We will deal with both hi-res and multicolor bitmapped graphics. Fortunately, the same general principles apply to both. Last month we saw how a routine to draw lines might look in BASIC. Actually executing the routine would show that BASIC is much too slow to be of much use for this task. At the end of last month's article we took the first step in putting together a set of machine language routines. This month we will complete the set.

First, here is a summary of the features of these drawing routines. The range of coordinates supported is 0 to 319 for X, and 0 to 199 for Y, when in hi-res mode. For multicolor mode, the range is 0 to 159 for X, and 0 to 199 again for Y. It is up to the user to insure that coordinates are within these ranges. Using coordinates which are too far out of range could cause the 64 to crash. In both hi-res and multicolor mode, the location of 0,0 is found at the lower left corner of the display.

Saving Memory For BASIC

The bitmap memory is placed at 57344 ($E000), underneath the operating system ROM. This avoids taking memory away from BASIC. Since this makes the bitmap data difficult to PEEK directly from BASIC, a routine is provided to perform this function. The screen memory is placed at 51200 ($C800), just below where the DOS Wedge loads. Use of these graphics routines should not conflict with the DOS Wedge, but may conflict with other BASIC enhancement software.

Last month we began by writing four of the required routines. This month we are going to upgrade two of those to accept arguments, and add six more. As was mentioned last time, we will execute these routines via a jump table at the beginning of the machine code. This will provide us fixed locations to SYS to, even if modifications or additions are made later. The following is a list of the routines found in the jump table:

Loc.	Description
JT+0	Save screen parameters
JT+3	Restore saved screen parameters
JT+6	Enable graphics screen
JT+9	Clear graphics screen
JT+12	Move graphics cursor to X,Y
JT+15	Plot pixel at X,Y
JT+18	Draw line to X,Y
JT+21	Set drawing mode
JT+24	Set drawing color (multicolor)
JT+27	Read bitmap byte (a function)

The jump vector location of these routines is shown as the variable JT plus an offset. To obtain the actual address, JT should be set to the base of the jump table, which is 49152 or $C000. The following table gives the syntax for using each of the routines in the jump table.

```
SYS JV                      :REM SAVE SCREEN
SYS JV+3                    :REM RESTORE SCREEN
SYS JV+6,MODE               :REM ENABLE GRAPHICS
      MODE: 0=HI-RES, 1=MULTICOLOR
SYS JV+9,C0,C1              :REM CLEAR SCREEN
      C0="OFF" COLOR, C1="ON" COLOR
      USE IF HI-RES BITMAP MODE
SYS JV+9,C0,C1,C2,C3        :REM CLEAR SCREEN
      C0=BACKGROUND, C1=FOREGROUND 1
      C2=FOREGROUND 2, C3=FOREGROUND 3
      USE IF MULTICOLOR MODE
SYS JV+12,X,Y               :REM MOVE
SYS JV+15,X,Y               :REM PLOT
SYS JV+18,X,Y               :REM DRAW
SYS JV+21,DM                :REM SET DRAWING MODE
      DM: 0=FLIP, 1=DRAW, 2=ERASE
SYS JV+24,C                 :REM SELECT COLOR
      WORKS ONLY FOR MULTICOLOR MODE
```

The last routine in the jump table (offset =27) is handled differently because it should be called by the USR function. To set it up as the USR function, execute the statement:

```
POKE 785,PEEK(JV+28) :POKE 786,PEEK(JV+29)
```

Once this is done, you may read bytes from the bitmap memory with the statement

```
BYTE=USR( OFFSET )
```

where OFFSET is the offset from the base address of the byte you wish to fetch.

A Graphics Cursor

The philosophy behind this is that these graphics commands differ slightly for other graphics enhancements to BASIC. Typically, enhancements will add a line-drawing command which always requires both end points. In the routines above, an internal graphics cursor is maintained. Lines are drawn from this graphics cursor to a specified end point. Whenever a line is drawn, the new end point becomes the graphics cursor location. Thus, successive executions of the DRAW routine will create a series of connected lines.

Also, you have a choice of three drawing modes, flip, draw, and erase. The draw mode causes points along the lines to be set to the on state, or to the selected color if in multicolor graphics. Erasing causes dots to be set to the off state or background color. The flip mode involves switching the pixels to their opposite state In the case of multicolor mode, pixels of the selected color are flipped to the background color, and vice versa. Pixels not of the selected color are flipped to the other nonselected color.

To provide a simple example of how to put these routines to use in a program, the following program draws an interesting circular pattern in hi-res mode. Once the pattern is drawn, the program will wait for you to press a key

```
10 JT=49152:SYS JT:REM SAVE SCREEN
20 SYS JT+6,0:SYS JT+9,1,2:REM INIT SCREEN
30 SYS JT+21,0:REM FLIP MODE
40 FOR I=0 TO 6.24 STEP .035
50 X=50*COS(I):Y=50*SIN(I)
60 SYS JT+12,160+X,100+Y:REM MOVE
70 SYS JT+18,160-X,100-Y:REM DRAW
80 NEXT
90 GET Z$:IF Z$="" THEN 90
100 SYS JT+3:REM RESTORE TEXT SCREEN
```

To put the required machine code into memory, run the BASIC program shown below.

Next month we'll explore some of the more interesting aspects of the machine language source code listing.

BASIC Program

Refer to the "Automatic Proofreader" article before typing this program in.

```
1 READ LN,SA,EA:LN=LN+30                  :rem 146
10 FOR I=0 TO EA-SA                       :rem 232
20 READ BY:POKE SA+I,BY:SUM=SUM+BY        :rem 120
30 IF INT((I+1)/8)*8<>(I+1) THEN 60       :rem 242
40 READ CS:IF CS<>SUM THEN 80             :rem 123
50 SUM=0:LN=LN+10                         :rem 254
60 NEXT                                   :rem 165
70 PRINT "SUCCESSFUL LOAD":END            :rem 105
80 PRINT "ERROR IN LINE";LN:END           :rem 104
500 DATA 500                              :rem 68
510 DATA 49152                            :rem 181
520 DATA 50087                            :rem 181
530 DATA 76,47,192,76,72,192,76,9,740     :rem 57
540 DATA 193,76,90,193,76,156,193,76,1053 :rem 255
550 DATA 59,194,76,192,194,76,101,195,1087 :rem 53
560 DATA 76,115,195,76,137,195,0,0,794    :rem 99
570 DATA 0,0,0,0,0,0,255,128,383          :rem 11
580 DATA 0,7,248,0,0,0,0,173,428          :rem 21
590 DATA 0,221,141,43,192,173,24,208,1002 :rem 212
600 DATA 141,44,192,173,17,208,141,45,961 :rem 230
610 DATA 192,173,22,208,141,46,192,96,1070 :rem 25
620 DATA 173,43,192,141,0,221,173,44,987  :rem 182
630 DATA 192,141,24,208,173,45,192,141,1116 :rem 68
640 DATA 17,208,173,46,192,141,22,208,1007 :rem 19
650 DATA 96,72,173,14,220,41,254,141,1011 :rem 218
660 DATA 14,220,165,1,41,253,133,1,828    :rem 69
670 DATA 104,96,72,165,1,9,2,133,582      :rem 242
680 DATA 1,173,14,220,9,1,141,14,573      :rem 225
690 DATA 220,104,96,164,254,240,13,160,1251 :rem 65
700 DATA 0,145,251,200,208,251,230,252,1537 :rem 53
710 DATA 198,254,208,243,164,253,240,10,1570 :rem 123
720 DATA 136,240,5,145,251,136,208,251,1372 :rem 67
730 DATA 145,251,96,32,97,192,160,0,973   :rem 144
740 DATA 132,251,160,200,132,252,160,232,1519 :rem 153
750 DATA 132,253,160,3,132,254,32,131,1097 :rem 12
760 DATA 192,44,40,192,16,20,160,0,664    :rem 75
770 DATA 132,251,160,216,132,252,160,232,1535 :rem 161
780 DATA 132,253,160,3,132,254,138,32,1104 :rem 11
790 DATA 131,192,169,0,133,251,169,224,1269 :rem 84
800 DATA 133,252,169,64,133,253,169,31,1204 :rem 72
810 DATA 133,254,169,0,32,131,192,76,987  :rem 192
820 DATA 114,192,32,253,174,32,158,173,1128 :rem 75
830 DATA 32,170,177,170,152,96,32,234,1063 :rem 24
840 DATA 192,141,34,192,142,35,192,32,960 :rem 234
850 DATA 234,192,141,36,192,142,37,192,1166 :rem 82
860 DATA 96,32,234,192,240,2,169,128,1093 :rem 241
870 DATA 141,40,192,173,0,221,9,3,779     :rem 34
880 DATA 73,3,141,0,221,173,24,208,843    :rem 76
890 DATA 41,7,9,8,9,32,141,24,271         :rem 92
900 DATA 208,173,17,208,9,32,141,17,805   :rem 131
910 DATA 208,44,40,192,16,12,173,22,707   :rem 125
920 DATA 208,9,16,141,22,208,169,3,776    :rem 90
930 DATA 208,10,173,22,208,41,239,141,1042 :rem 10
940 DATA 22,208,169,7,141,41,192,73,853   :rem 141
950 DATA 255,141,42,192,169,255,141,38,1233 :rem 82
960 DATA 192,96,32,246,192,44,40,192,1034 :rem 241
970 DATA 48,21,173,36,192,10,10,10,500    :rem 65
980 DATA 10,141,36,192,173,34,192,41,819  :rem 188
```

```
990 DATA 15,13,36,192,76,163,192,173,860
                                   :rem 199
1000 DATA 36,192,10,10,10,10,141,36,445
                                    :rem 96
1010 DATA 192,32,234,192,41,15,13,36,755
                                   :rem 171
1020 DATA 192,141,36,192,32,234,192,170,1
     189                           :rem 121
1030 DATA 173,34,192,141,33,208,173,36,99
     0                              :rem 24
1040 DATA 192,76,163,192,32,246,192,162,1
     255                           :rem 129
1050 DATA 3,189,34,192,157,30,192,202,999
                                   :rem 243
1060 DATA 16,247,96,56,169,199,237,32,105
     2                              :rem 43
1070 DATA 192,72,74,74,74,133,252,160,103
     1                              :rem 20
1080 DATA 0,132,251,74,102,251,74,102,986
                                   :rem 220
1090 DATA 251,101,252,133,252,173,30,192,
     1384                          :rem 161
1100 DATA 174,31,192,45,42,192,44,40,760
                                   :rem 172
1110 DATA 192,16,6,10,72,138,42,170,646
                                   :rem 122
1120 DATA 104,24,101,251,133,251,138,101,
     1103                          :rem 133
1130 DATA 252,133,252,104,41,7,24,101,914
                                   :rem 207
1140 DATA 251,144,2,230,252,24,105,0,1008
                                   :rem 198
1150 DATA 133,251,165,252,105,224,133,252
     ,1515                         :rem 207
1160 DATA 173,30,192,45,41,192,170,96,939
                                   :rem 242
1170 DATA 169,0,168,44,39,192,16,7,635
                                    :rem 94
1180 DATA 80,2,177,251,77,38,192,44,861
                                   :rem 145
1190 DATA 40,192,48,10,61,47,194,133,725
                                   :rem 183
1200 DATA 97,189,47,194,208,8,61,55,859
                                   :rem 161
1210 DATA 194,133,97,189,55,194,73,255,11
     90                             :rem 94
1220 DATA 49,251,5,97,145,251,96,128,1022
                                   :rem 234
1230 DATA 64,32,16,8,4,2,1,192,319
                                   :rem 126
1240 DATA 48,12,3,32,156,193,32,97,573
                                    :rem 85
1250 DATA 192,32,171,193,32,0,194,76,890
                                   :rem 186
1260 DATA 114,192,169,1,149,106,169,0,900
                                   :rem 228
1270 DATA 149,107,56,189,34,192,253,30,10
     10                             :rem 73
1280 DATA 192,149,98,189,35,192,253,31,11
     39                             :rem 98
1290 DATA 192,149,99,16,20,169,255,149,10
     49                             :rem 99
1300 DATA 106,149,107,56,169,0,245,98,930
                                   :rem 238
1310 DATA 149,98,169,0,245,99,149,99,1008
                                     :rem 4
1320 DATA 96,21,98,208,4,149,106,149,831
                                   :rem 192
1330 DATA 107,96,165,99,74,133,103,165,94
     2                              :rem 39
1340 DATA 98,106,133,102,24,169,0,229,861
                                   :rem 230
1350 DATA 98,133,104,169,0,229,99,133,965
                                   :rem 250
1360 DATA 105,96,24,165,102,101,100,133,8
     26                             :rem 56
1370 DATA 102,170,165,103,101,101,133,103
     ,978                          :rem 151
1380 DATA 197,99,144,19,208,4,228,98,997
                                   :rem 224
1390 DATA 144,13,138,56,229,98,133,102,91
     3                              :rem 32
1400 DATA 165,103,229,99,133,103,56,96,98
     4                              :rem 40
1410 DATA 32,246,192,32,97,192,162,0,953
                                   :rem 184
1420 DATA 32,74,194,162,2,32,74,194,764
                                   :rem 137
1430 DATA 165,98,197,100,165,99,229,101,1
     154                           :rem 137
1440 DATA 144,62,32,130,194,36,107,16,721
                                   :rem 221
1450 DATA 10,32,159,193,56,169,0,229,848
                                   :rem 194
1460 DATA 108,133,108,32,171,193,32,0,777
                                   :rem 227
1470 DATA 194,230,104,208,4,230,105,240,1
     315                           :rem 103
1480 DATA 102,238,30,192,208,3,238,31,104
     2                              :rem 11
1490 DATA 192,32,154,194,144,9,24,173,922
                                   :rem 241
1500 DATA 32,192,101,108,141,32,192,32,83
     0                               :rem 3
1510 DATA 171,193,32,0,194,76,241,194,110
     1                              :rem 15
1520 DATA 162,1,181,98,180,100,149,100,97
     1                              :rem 17
1530 DATA 148,98,202,16,245,32,130,194,10
     65                             :rem 74
1540 DATA 36,107,16,10,32,159,193,56,609
                                   :rem 184
1550 DATA 169,0,229,108,133,108,32,171,95
     0                              :rem 23
1560 DATA 193,32,0,194,230,104,240,31,102
     4                               :rem 1
1570 DATA 24,173,32,192,101,108,141,32,80
     3                              :rem 10
1580 DATA 192,32,154,194,144,8,238,30,992
                                   :rem 246
1590 DATA 192,208,3,238,31,192,32,171,106
     7                              :rem 27
1600 DATA 193,32,0,194,76,60,195,32,782
                                   :rem 137
1610 DATA 159,193,76,114,192,32,234,192,1
     192                           :rem 132
1620 DATA 41,3,73,3,106,106,106,141,579
                                   :rem 120
1630 DATA 39,192,96,32,234,192,41,3,829
                                   :rem 144
1640 DATA 170,189,133,195,44,40,192,16,97
     9                              :rem 45
1650 DATA 3,141,38,192,96,0,85,170,725
                                    :rem 88
1660 DATA 255,32,170,177,170,152,24,105,1
     085                           :rem 121
1670 DATA 0,133,251,138,105,224,133,252,1
     236                           :rem 109
1680 DATA 32,97,192,160,0,177,251,32,941
                                   :rem 187
1690 DATA 114,192,168,169,0,108,5,0,756
                                   :rem 139
```
©

Atari Line Check Utility

Ed Sisul

"Atari Line Check" lets you use a joystick to perform a line-by-line search for program bugs.

Quite often, the most effective way to debug a program is to check each line, one at a time, for mistakes. For those of us who are not fortunate enough to own a printer, this can be a very tedious task. The lines can be examined using LIST and CTRL-1 to scroll through the program, but it is difficult to find minor mistakes while staring at a whole screen filled with GRAPHICS 0 text. The lines can be displayed one at a time using the sequence LIST *line number*, SHIFT CLEAR, LIST *line number*, SHIFT CLEAR, etc.; but this approach is too slow and cumbersome.

Scrolling With A Joystick

This program will step through a listing and display each line, one at a time, in large GRAPHICS 2 print. The best part is that the scrolling is controlled with a joystick. Pulling back on the stick advances through the listing, and pushing forward on the stick backtracks through the listing. With the stick centered, the displayed line stays on the screen for scrutiny. If a mistake is spotted, press the trigger button, and the line containing the mistake is redisplayed in the normal screen editing mode so it can be corrected. Once the error is dispatched, typing CONT will resume the line-by-line check, or typing RUN will terminate the line check and execute the main program. After typing in "Atari Line Check," LIST it to disk or cassette. Then, using the ENTER command, append it to the program to be checked. Plug a joystick into Port 1 and type GOTO 32000 to start checking lines.

Array Storage

The heart of the program is lines 32010–32030. Lines 32010–32020 retrieve the program line numbers stored in memory and store them in the array LINUM. A complete explanation of the PEEKs used to do this can be found in Larry Isaacs' article "Inside Atari BASIC" in *COMPUTE!'s First Book of Atari*. Line 32025 opens the screen editor for input and output, lists a line on the screen, then retrieves the entire line, including its line number, and stores it in the variable LINE$. The POKEs in line 32025 blank the screen during these operations. Line 32030 then reprints LINE$ on the screen in GRAPHICS 2 text in black letters on a white background.

Lines 32035–32055 contain the joystick controller routines to increment or decrement the subscript of the line number array or to redisplay a line for editing. Line 32000 initializes the variables, dimensions LINE$ to the maximum number of characters in a logical line, and dimensions the LINUM array to accommodate a 200-line program. The POKE in line 32000 standardizes the left-hand margin on all systems. Line 32005 initially sets all elements of the LINUM array to zero. Should you encounter a program with more than 200 lines, simply change the dimensioned size of LINUM in line 32000 and the maximum increment of the loop in line 32005 accordingly.

Storage Characters

Because each line is displayed in graphics mode 2, which uses the internal character set, some characters won't be displayed as originally typed. For instance, the special graphics characters will be displayed as numeric or punctuation symbols, and lowercase letters will be displayed as green uppercase letters. Also, the CLEAR character, CHR$(125), will cause the screen to clear when it is printed. When this happens, just press the trigger button to see the characters in their original form.

Atari Line Check

Refer to the "Automatic Proofreader" article before typing this program in.

```
DM 32000 POKE 82,0:ST=0:ZZ=1:TRAP 32005:DIM LINE$(120),LINUM(200):TRAP 40000
IC 32005 FOR N=0 TO 200:LINUM(N)=0:NEXT N
ML 32010 AD=PEEK(136)+256*PEEK(137)
JB 32015 LINUM(ZZ)=PEEK(AD)+256*PEEK(AD+1):IF LINUM(ZZ)=32000 THEN END
OE 32020 IF LINUM(ZZ)=0 THEN AD=AD+PEEK(AD+2):GOTO 32015
HL 32025 OPEN #1,13,0,"E:":POKE 709,8:POKE 710,8:POKE 712,8:LIST LINUM(ZZ):POSITION 0,1:INPUT #1;LINE$:CLOSE #1
BF 32030 GRAPHICS 18:POKE 708,2:POKE 712,8:POSITION 0,2:?#6;LINE$
IC 32035 IF STRIG(0)=0 THEN ST=1:GRAPHICS 0:LIST LINUM(ZZ):STOP
EP 32040 IF ST=1 THEN ST=0:GOTO 32025
MH 32045 IF STICK(0)=13 THEN ZZ=ZZ+1:GOTO 32020
MA 32050 IF STICK(0)=14 AND ZZ>0 THEN ZZ=ZZ-1:GOTO 32025
DF 32055 GOTO 32035
```

©

Commodore Word Wizard

Joe W. Rocke

"Word Wizard" improves your writing skills by checking the readability of any written material. For the VIC-20, Commodore 64, and PET/CBM computers.

The term *foggy writing* was originated by Robert Gunning. Seeking ways to improve the readability of written text, he developed a *fog index* formula. The formula is based on counting the number of words and sentences in a sample paragraph of text. Long words and long sentences produce a high index number. This type of writing is called foggy because it can be harder to read and understand. Writing that is easy to read (and understand) should have a low fog index.

The fog index formula uses a 100- to 200-word sample of text. Words of three syllables or more are considered "long." Dividing the word count by the number of sentences provides the average sentence length. Adding the number of long words and performing a simple computation produce the fog index. Although the index number is rather arbitrary, it does provide a standard for measuring text readability.

Researchers have since learned that people prefer to read below their educational level. Thus the fog formula has been expanded to produce a reading level index number. The result is a number that represents the approximate grade level at which written material can be read and understood.

People are comfortable reading text that has a reader index ranging from 6 to 8. Most of the writing in popular magazines and newspapers has an index in this range. People are capable of reading at a higher level, but the concentration required can make such writing tedious. Even college professors find it uncomfortable to read something with an index of 12 or higher.

Computerized Word Check

The computer is an ideal tool for checking text for readability. Large companies have developed programs of this type to check their product manuals. When used with word processing systems, this checking process takes little additional time.

Using "Word Wizard" is as simple as typing text onto a video screen instead of on paper, as with a typewriter. A 100-word sample is all that is required. Almost all text-reading analysis is based on this sample size.

The program begins with a prompt. There is no cursor, but whatever is typed appears on the screen. The left arrow can be used to correct a typo without affecting the program. Use the RETURN key only when you are finished entering the sample. The screen then clears, and the text that has been typed to memory will begin to march across the screen. The text display will then be formatted to improve readability.

Type in the text sample without worrying how it looks on the video display. The text will wrap around the screen, causing some words to be broken midway and to continue on the next line. The display is primarily for reference so you can see what was originally typed.

The Display Phase

Next, during its display phase, the program counts characters, words, and sentences. It also counts the number of words containing more than nine characters, which are presumed to consist of three or more syllables. Word groups ending with a semicolon or colon are counted as one sentence. This prevents a compound sentence from being counted as a single sentence. Naturally, any word group ending with a period, question mark, or exclamation mark is counted as a sentence.

The word-checking data is stored in simple variables and is then used to compute the reading index at the end of the display cycle. A continuation prompt concludes the display cycle to permit you to read the last display page.

Finally the word, sentence, and long word counts are displayed. The reading index, rounded to two decimal places, completes the text analysis. The program then asks you to repeat the analysis or exit the program.

An index of 6–9 indicates a good readability level. A higher index indicates that the text might benefit from some editing. You may want to use two shorter sentences which carry the same thought as a long one, or try to find shorter words. For example, it is easier to read *city* than the word *metropolis*.

Variables

A$	The input string is confined to one character.
BE	Beginning address of the memory storage area.
C	ASCII value of A$, and the character counter.
C$	Character string used for the display cycle.
ID	Reading index. L is the display line length counter.
LW	Long word count storage.
MS	Memory storage ending address.
P	PEEK value of MS contents.
S	Sentence count storage.
T	Display cycle loop counter.
W	Word count storage.
WC	Input cycle word count.
Z & Z$	Prompts.

Housekeeping Chores

Lines 10–30: Housekeeping chores are performed at the beginning of the program. The formula used to round the reading index is defined in line 10. Major variables are set to zero to prevent errors if the program is rerun. Variable MS in line 20 denotes the beginning memory storage address. A second variable is set to the same value for use in the display loop.

The value currently in the program works with an unexpanded VIC-20. Use MS=2300 in line 20 if you have a PET/CBM or a 3K expanded VIC. (Ignore the color commands if you have a PET.) For a VIC with 8K or more of expansion memory, use MS=5900. Try MS=3300 for the Commodore 64. For other systems you will have to use an address above the BASIC program area.

Lines 35–150: The input cycle begins at line 60 with the GET A$ keyboard scan for a key input. When a key is pressed, the input is checked for a backspace (left cursor). If it is a backspace, the invisible cursor moves one space to the left, and the memory storage is decreased by one. This is to prevent counting the backspace as part of the text. The program then loops back for a new key input.

If the key pressed is a text character, the key is displayed and converted to its ASCII equivalent. The ASCII value is then POKEd in memory address MS for storage. The input is then tested for a carriage return (CR); if not a CR, storage address MS is incremented by one, and the program loops back for another key input. Note that a CR breaks the input loop, jumping program flow to the continuation GOSUB.

The Word Count

Line 110 performs a word count during the input cycle. The count value of 125 in line 120 limits input to a maximum of 125 words. These two lines are optional, but do insure keeping the input within sample limits. A smaller number of words can be used for a sample, of course.

Lines 160–300: The display and checking cycle begins upon user response to the continuation prompt. Variables used to accumulate word-checking data are set to zero to prevent errors if the program is repeated. A FOR-NEXT loop is used for the display cycle, since storage beginning address BE and ending address MS were established during the input cycle.

The stored ASCII data is PEEKed from each memory address, converted to a string, and temporarily stored in string variable C$ for display. C$ now represents the keyboard character entered during the input cycle. The individual characters are counted and the count is stored in C. L is used to count characters for line display formatting.

Word-checking functions are performed by IF statements. These lines check for the space character that denotes a word end, or punctuation indicating a sentence end. A space increments the word count, W. A sentence end increments the sentence count stored in S and decreases the character count by one. The decrease prevents the punctuation from being counted as a word character. If the character count in C is equal to or greater than 9, and a space indicates a word, then long word counter LW is incremented. The character counter is returned to zero value whenever a space or sentence end is encountered.

Screen Formatting

Line 220 formats the text to reduce word wraparound.

Lines 320–400: The text analysis is performed in this portion of the program. The reading index is computed in line 320. Text data accumulated during the word-check cycle are displayed, followed by the reading index (ID). The rounding function is performed by the FNA(ID) formula which was established at the beginning of the program.

Lines 410–480: The remaining lines contain the user prompts. Conventional INPUT statements are used to keep the program short. END is used between the REPEAT prompt and the continuation GOSUB to prevent an error message when exiting the program. Line 470 prints the word input count and returns control to the continuation prompt of line 150.

Word Wizard

Refer to the "Automatic Proofreader" article before typing this program in.

```
5 REM... * WORD CHECK *                   :rem 145
10 DEF FNA(B)=INT(B*100+.5)/100            :rem 92
20 MS=5300: BE=MS                         :rem 165
30 C=0:L=0:LW=0:S=0:W=0:WC=0              :rem 137
35 REM...INPUT CYCLE                      :rem 214
50 PRINT"{CLR}{7}BEGIN INPUT ...":PRINT
                                          :rem 169
60 GETA$:IFA$=""THEN 60                   :rem 239
70 IFA$=CHR$(157)THEN PRINTA$;:MS=MS-1:GO
   TO60                                   :rem 209
80 PRINT A$;                              :rem 149
90 C=ASC(A$)                              :rem 118
100 POKE MS,C                             :rem 207
110 IFA$=" "THEN WC=WC+1                    :rem 3
120 IFWC=>125 THEN 470                    :rem 153
130 IFA$=CHR$(13)THEN 150                  :rem 64
140 MS=MS+1: GOTO 60                       :rem 71
150 GOSUB 440                             :rem 174
155 REM...DISPLAY CYCLE                   :rem 143
160 C=0:L=0:LW=0:S=0:W=1                  :rem 125
170 PRINT"{CLR}"                          :rem 252
180 FOR T=BE TO MS                        :rem 219
190 P=PEEK(T)                             :rem 241
200 C$=CHR$(P)                            :rem 216
210 C=C+1:L=L+1                            :rem 29
220 IFC$=" "AND L=>15THEN GOSUB 460
                                           :rem 84
230 PRINTC$;                              :rem 196
240 IFC$=" "THEN W=W+1:C=C-1              :rem 222
250 IFC$="."ORC$="!"ORC$="?"ORC$=":"ORC$=
    ";"THEN S=S+1:C=C-1:C$=" "             :rem 32
260 IFC$=" "ANDC=>9 THEN LW=LW+1          :rem 231
270 IFC$=" "THEN C=0                      :rem 239
280 IFC$=CHR$(13)THEN 310                  :rem 70
290 NEXT                                  :rem 218
300 PRINT                                  :rem 32
310 GOSUB 440                             :rem 172
315 REM...* ANALYSIS *                    :rem 191
320 ID=.4*(W/S+LW*100/W)                   :rem 36
330 PRINT"{CLR}"                          :rem 250
340 PRINTSPC(4)"** ANALYSIS **":PRINT
                                          :rem 166
350 PRINT"WORDS{2 SPACES}=";W             :rem 199
360 PRINT"SENTENCES{2 SPACES}=";S:rem 221
370 PRINT"AVG.WD/SENT =";INT(W/S)           :rem 8
380 PRINT"LONG WORDS{2 SPACES}=";LW
                                           :rem 70
390 PRINT                                  :rem 41
400 PRINT"READER INDEX =";FNA(ID):rem 187
410 PRINT:INPUT"REPEAT (Y/N)";Z$ :rem 209
415 IFZ$<>"N"ANDZ$<>"Y"THEN410          :rem 223
420 IFZ$="Y"GOTO 20                        :rem 24
430 PRINT"{BLU}{CLR}":END                  :rem 43
440 INPUT"PRESS <RETURN>";Z               :rem 232
450 RETURN                                :rem 121
460 PRINTC$;CHR$(13):L=0:RETURN           :rem 140
470 PRINT:PRINT"WORDS INPUT=";WC :rem 100
480 GOTO150                               :rem 107
```

©

The Automatic Proofreader

For VIC, 64, And Atari

Charles Brannon, Program Editor

At last there's a way for your computer to help you check your typing. "The Automatic Proofreader" will make entering programs faster, easier, and more accurate.

The strong point of computers is that they excel at tedious, exacting tasks. So why not get your computer to check your typing for you?

With "The Automatic Proofreader" nestled in your VIC-20, Commodore 64, or Atari computer, every line you type in will be verified. It displays a special code, called a *checksum*, at the top of the screen. The checksum, either a number (VIC/64) or a pair of letters (Atari), corresponds to the line you've just typed. It represents every character in the line summed together. A matching code in the program listing lets you compare it to the checksum which the Proofreader displays. A glance is all it takes to confirm that you've typed the line correctly.

Entering The Automatic Proofreader

Commodore (VIC/64) owners should type in Program 1. Program 2 is for Atari users. Since the Proofreader is a machine language program, be especially diligent. Watch out for typing extra commas, or a letter O for a zero, and check every number carefully. If you make a mistake when typing in the DATA statements, you'll get the message "Error in DATA statements" when you RUN the program. Check your typing and try again.

When you've typed in The Automatic Proofreader, SAVE it to tape or disk at least twice *before running it for the first time*. If you mistype the Proofreader, it may cause a system crash when you first run it. By SAVEing a copy beforehand, you can reLOAD it and hunt for your error. Also, you'll want a backup copy of the Proofreader because you'll use it again and again—every time you enter a program from COMPUTE!.

When you RUN the Proofreader, the program will be POKEd safely into memory, then it will activate itself. If you ever need to reactivate it (RUN/STOP—RESTORE or SYSTEM RESET will disable it), just enter the command SYS 886 (VIC/64) or PRINT USR(1536) for the Atari.

Using The Proofreader

Now, let's see how it works. LIST the Proofreader program, move the cursor up to one of the lines, and press RETURN. If you've entered the Proofreader correctly, a checksum will appear in the top-left corner of your screen.

Try making a change in the line and hit RETURN. Notice that the checksum has changed. All VIC and 64 listings in COMPUTE! now have a number appended to the end of each line, for example, :rem 123. *Don't enter this statement*. It is just for your information. The rem is used to make the number harmless if someone does type it in. It will, however, use up memory if you enter it, and it will cause the checksum displayed at the top of the screen to be different, even if you entered the rest of the line correctly.

The Atari checksum is found immediately to the left of each line number. This makes it impossible to type in the checksum accidentally, since a program line must start with a number.

Just type in each line (without the printed checksum), and check the checksum displayed at the top of the screen against the checksum in the listing. If they match, go on to the next line. If they don't, there's a mistake. You can correct the line immediately, instead of waiting to find the error when you RUN the program.

The Proofreader is not picky with spaces. It will not notice extra spaces or missing ones. This is for your convenience, since spacing is generally not important. Occasionally proper spacing is important, but the article describing the program will warn you to be careful in these cases.

Nobody's Perfect

Although the Proofreader is an important aid, there are a few things to watch out for. If you enter a line by using abbreviations for commands, the checksum will not match up. This is because the Proofreader is very literal: It looks at the individual letters in a line, not at tokens such as PRINT. There is a way to make the Proofreader check such a line. After entering the line, LIST it. This makes the computer spell out the abbreviations. Then move the cursor up to the line and press RETURN. It should now match the checksum. You can check whole groups of lines this way. Atari users should beware of using ? as an abbreviation for PRINT—they're not the same thing in the Proofreader's eyes.

The checksum is a sum of the ASCII values of the characters in a line. VIC and 64 owners may wonder why the numbers are so small, never exceeding 255. This is because the addition is done only in eight bits. A result over 255 will roll over past zero, like an odometer past 99999. On the Atari, the number is turned into two letters, both for increased convenience and to make the Proofreader shorter. For the curious, the letters correspond to the values of the left and right nybbles added to 33 (to offset them into the alphabet). This number is then stored directly into screen memory.

Due to the nature of a checksum, the Proofreader will not catch all errors. Since $1+3+5=3+1+5$, the Proofreader cannot catch errors of transposition. In fact, you could type in the line in any order, and the Proofreader wouldn't notice. Anytime the Proofreader seems to act strange, keep this in mind. Since the ASCII values of the number 18 (49+56) and 63 (54+51) both equal 105, these numbers are equal according to the Proofreader. There really is no simple way to catch these kinds of errors. Fortunately, the Proofreader will catch the majority of the typing mistakes most people make.

If you want the Proofreader out of your way, just press SYSTEM RESET or RUN/STOP—RESTORE. If you need it again, enter SYS 828 (VIC/64) or PRINT USR(1536) (Atari). You must disable the Proofreader before doing any tape operations on the VIC or 64.

Hidden Perils

The Proofreader's home in the VIC and 64 is not a very safe haven. Since the cassette buffer is wiped out during tape operations, you need to disable the Proofreader with RUN/STOP—RESTORE before you SAVE your program. This applies only to tape use. Disk users or Atari owners have nothing to worry about.

Not so for VIC and 64 owners with tape drives. What if you type in a program in several sittings? The next day, you come to your computer, LOAD and RUN the Proofreader, then try to LOAD the partially completed program so you can add to it. But since the Proofreader is trying to hide in the cassette buffer, it is wiped out!

What you need is a way to LOAD the Proofreader after you've LOADed the partial program. The problem is, a tape load to the buffer destroys what it's supposed to load.

After you've typed in and RUN the Proofreader, enter the following lines in direct mode (without line numbers) *exactly* as shown:

```
A$="PROOFREADER.T": B$="{1Ø SPACES}": FOR
    X = 1 TO 4: A$=A$+B$: NEXTX
FOR X = 886 TO 1Ø18: A$=A$+CHR$(PEEK(X)):
    NEXTX
OPEN 1,1,1,A$:CLOSE1
```

After you enter the last line, you will be asked to press record and play on your cassette recorder. Put this program at the beginning of a new tape. This gives you a new way to load the Proofreader. Anytime you want to bring the Proofreader into memory without disturbing anything else, put the cassette in the tape drive, rewind, and enter:

OPEN1:CLOSE1

You can now start the Proofreader by typing SYS 886. To test this, PRINT PEEK(886) should return the number 173. If it does not, repeat the steps above, making sure that A$ ("PROOFREADER.T") contains 13 characters and that B$ contains 10 spaces.

You can now reload the Proofreader into memory whenever LOAD or SAVE destroys it, restoring your personal typing helper.

Incidentally, you can protect the cassette buffer on the Commodore 64 with POKE 178,165. This POKE should work on the VIC, but it has caused numerous problems, probably due to a bug in the VIC operating system. With this POKE, the 64 will not wipe out the cassette buffer during tape LOADs and SAVEs.

Program 1: VIC/64 Proofreader

```
1ØØ PRINT"{CLR}PLEASE WAIT...":FORI=886TO
    1Ø18:READA:CK=CK+A:POKEI,A:NEXT
11Ø IF CK<>17539 THEN PRINT"{DOWN}YOU MAD
    E AN ERROR":PRINT"IN DATA STATEMENTS.
    ":END
12Ø SYS886:PRINT"{CLR}{2 DOWN}PROOFREADER
     ACTIVATED.":NEW
886 DATA 173,Ø36,ØØ3,2Ø1,15Ø,2Ø8
892 DATA ØØ1,Ø96,141,151,ØØ3,173
898 DATA Ø37,ØØ3,141,152,ØØ3,169
9Ø4 DATA 15Ø,141,Ø36,ØØ3,169,ØØ3
91Ø DATA 141,Ø37,ØØ3,169,ØØØ,133
916 DATA 254,Ø96,Ø32,Ø87,241,133
922 DATA 251,134,252,132,253,ØØ8
928 DATA 2Ø1,Ø13,24Ø,Ø17,2Ø1,Ø32
934 DATA 24Ø,ØØ5,Ø24,1Ø1,254,133
94Ø DATA 254,165,251,166,252,164
946 DATA 253,Ø4Ø,Ø96,169,Ø13,Ø32
952 DATA 21Ø,255,165,214,141,251
958 DATA ØØ3,2Ø6,251,ØØ3,169,ØØØ
964 DATA 133,216,169,Ø19,Ø32,21Ø
97Ø DATA 255,169,Ø18,Ø32,21Ø,255
976 DATA 169,Ø58,Ø32,21Ø,255,166
982 DATA 254,169,ØØØ,133,254,172
988 DATA 151,ØØ3,192,Ø87,2Ø8,ØØ6
994 DATA Ø32,2Ø5,189,Ø76,235,ØØ3
1ØØØ DATA Ø32,2Ø5,221,169,Ø32,Ø32
1ØØ6 DATA 21Ø,255,Ø32,21Ø,255,173
1Ø12 DATA 251,ØØ3,133,214,Ø76,173
1Ø18 DATA ØØ3
```

Program 2: Atari Proofreader

```
1ØØ GRAPHICS Ø
11Ø FOR I=1536 TO 17ØØ:READ A:POKE I
    ,A:CK=CK+A:NEXT I
12Ø IF CK<>19Ø72 THEN ? "Error in DA
    TA statements. Check typing":END
13Ø A=USR(1536)
14Ø ? :? "Automatic Proofreader now
    activated."
15Ø END
1536 DATA 1Ø4,16Ø,Ø,185,26,3
1542 DATA 2Ø1,69,24Ø,7,2ØØ,2ØØ
1548 DATA 192,34,2Ø8,243,96,2ØØ
1554 DATA 169,74,153,26,3,2ØØ
156Ø DATA 169,6,153,26,3,162
1566 DATA Ø,189,Ø,228,157,74
1572 DATA 6,232,224,16,2Ø8,245
1578 DATA 169,93,141,78,6,169
1584 DATA 6,141,79,6,24,173
159Ø DATA 4,228,1Ø5,1,141,95
1596 DATA 6,173,5,228,1Ø5,Ø
16Ø2 DATA 141,96,6,169,Ø,133
16Ø8 DATA 2Ø3,96,247,238,125,241
1614 DATA 93,6,244,241,115,241
162Ø DATA 124,241,76,2Ø5,238,Ø
1626 DATA Ø,Ø,Ø,Ø,32,62
1632 DATA 246,8,2Ø1,155,24Ø,13
1638 DATA 2Ø1,32,24Ø,7,72,24
1644 DATA 1Ø1,2Ø3,133,2Ø3,1Ø4,4Ø
165Ø DATA 96,72,152,72,138,72
1656 DATA 16Ø,Ø,169,128,145,88
1662 DATA 2ØØ,192,4Ø,2Ø8,249,165
1668 DATA 2Ø3,74,74,74,74,24
1674 DATA 1Ø5,161,16Ø,3,145,88
168Ø DATA 165,2Ø3,41,15,24,1Ø5
1686 DATA 161,2ØØ,145,88,169,Ø
1692 DATA 133,2Ø3,1Ø4,17Ø,1Ø4,168
1698 DATA 1Ø4,4Ø,96
```

©

CAPUTE!

Modifications Or Corrections To Previous Articles

Atari Super Directory

The character which appears as a grave (`) in lines 5010 and 5020 of this program from the April issue (p. 176) should actually be {.}, CTRL-period. You may find it easier to replace these lines with the lines below, which build M$ from DATA statements.

```
OM 5000  DIM M$(40):RESTORE 5040
NJ 5010  FOR I=1 TO 40:READ A:M$(I)=C
         HR$(A):NEXT I
KI 5030  RETURN
FG 5040  DATA 104,201,2,240,9,170,240
         ,5,104,104,202,208,251,96,10
         4,133,204,104,133,203,104
IK 5050  DATA 104,133,205,160,0,177,2
         03,9,128,145,203,200,196,205
         ,208,245,96,0,0
```

Roader For Atari And Color Computer

The Atari version of this game from the March issue (p. 66) may stop with an ERROR 141 message. To prevent this, Edward Rybczyk suggests the following corrections:

```
380 IF A=43 THEN CLR: RUN
390 POKE 764,255: END
```

The Color Computer version requires Extended BASIC to run as published. Ron Crail suggests changes to allow the program to run in standard Color BASIC: Change the value of XLOC to 304 in line 220 and to 308 in line 230, and change COS to SIN in lines 260 and 310. Also, adding the line 245 N$ = "X" will prevent an OS error.

VIC Barrier Battle

A testing loop was inadvertently left in line 200 of this game program from the March issue (p. 84). Troy Pibus points out that the line should read:

```
200 DD=37154:P1=37151:P2=37152
```

64 MLX And Trident

There is an error in the version of the "MLX" machine language editor from the March issue (p. 182). In line 765, K = S + 1 should be replaced with K = S. This error will prevent the "Trident" game (p. 100), published in MLX format, from working properly. Fortunately, the problem is quite easy to fix. First, load and correct MLX and save the corrected version. Then run MLX and use the MLX Load option to load in Trident. Use the starting and ending addresses given in the Trident article. Retype the first line of Trident (49152), then use the MLX Save option to create a new copy of the game, which should now work properly.

Atari Trident

Reader Jim Davis suggests the following improvement to this game from the March issue (p. 94):

```
105 Z=USR(ADR(M$),M,M+1,128):FOR I=15 TO
    0 STEP -0.08:SOUND 0,10,8,I:NEXT I:Z
    =USR(ADR(A$),48+C,1,144,51)
```

This adds an explosion sound when an incoming missile is destroyed.

Commodore Floating Subroutines

Programs 1, 2, and 3 for this article from the March issue (p. 164) will print a range of hex address values which is one greater than the correct range, as shown in decimal. To correct this, Paul Montognese suggests changing the H = C in line 63994 to H = C –1.

Chopperoids

Some readers tried to create a binary file (MLX option F) for this Atari machine language program (December 1983, p. 122). As stated in the article, "Chopperoids" must be put on a boot disk or boot tape. If you made a binary file, follow these steps to create a boot disk from your work:

1. Load the MLX program and make the following *temporary* changes:

```
750 IF  NOT READ THEN 1040
850 TRAP 40000:CLOSE #2:? "Finished.
    ":LET READ=0:BUFFER$(FIN-BEG+31)
    =CHR$(0):BUFFER$(31)=BUFFER$(61)
    :GOTO 360
1000 H=INT(ADR(BUFFER$)/256):L=ADR(B
     UFFER$)-H*256:L=L+30:POKE ICBAD
     R+X,L:POKE ICBADR+X+1,H
```

2. Run the modified MLX and use the addresses given in the original article. Specify the boot disk option.

3. Use the MLX Load command to load your binary file. All the data will be moved up five lines, as described in the February "CAPUTE!" corrections.

4. Use the MLX New Address command to begin typing at line 6092 and enter the additional lines from February "CAPUTE!" (p. 181). Insert a new disk in the drive and use the MLX Save option to create a boot disk with the corrected data. ©

MLX Machine Language Entry Program For Commodore 64

Charles Brannon, Program Editor

MLX is a labor-saving utility that allows almost fail-safe entry of machine language programs published in COMPUTE!. *You need to know nothing about machine language to use MLX—it was designed for everyone.*

MLX is a new way to enter long machine language (ML) programs with a minimum of fuss. MLX lets you enter the numbers from a special list that looks similar to BASIC DATA statements. It checks your typing on a line-by-line basis. It won't let you enter illegal characters when you should be typing numbers. It won't let you enter numbers greater than 255 (forbidden in ML). It won't let you enter the wrong numbers on the wrong line. In addition, MLX creates a ready-to-use tape or disk file. You can then use the LOAD command to read the program into the computer:

LOAD "filename",1,1 (for tape)
LOAD "filename",8,1 (for disk)

To start the program, you enter a SYS command that transfers control from BASIC to machine language. The starting SYS number appears in the article.

Using MLX

Type in and save MLX for your 64 (you'll want to use it in the future). When you're ready to type in an ML program, run MLX. MLX asks you for two numbers: the starting address and the ending address. These numbers are given in the article accompanying the ML program.

You'll see a prompt corresponding to the starting address. The prompt is the current line you are entering from the listing. It increases by six each time you enter a line. That's because each line has seven numbers—six actual data numbers plus a *checksum number*. The checksum verifies that you typed the previous six numbers correctly. If you enter any of the six numbers wrong, or enter the checksum wrong, the computer rings a buzzer and prompts you to reenter the line. If you enter it correctly, a bell tone sounds and you continue to the next line.

MLX accepts only numbers as input. If you make a typing error, press the INST/DEL key; the entire number is deleted. You can press it as many times as necessary back to the start of the line. If you enter three-digit numbers as listed, the computer automatically prints the comma and goes on to accept the next number. If you enter less than three digits, you can press either the comma, SPACE bar, or RETURN key to advance to the next number. The checksum automatically appears in inverse video for emphasis.

To simplify your typing, MLX redefines part of the keyboard as a numeric keypad (lines 581–584):

	U	I	O			7	8	9
H	J	K	L	become	0	4	5	6
	M	,	.			1	2	3

MLX Commands

When you finish typing an ML listing (assuming you type it all in one session), you can then save the completed program on tape or disk. Follow the screen instructions. If you get any errors while saving, you probably have a bad disk, or the disk is full, or you've made a typo when entering the MLX program itself.

You don't have to enter the whole ML program in one sitting. MLX lets you enter as much as you want, save it, and then reload the file from tape or disk later. MLX recognizes these commands:

SHIFT-S: Save
SHIFT-L: Load
SHIFT-N: New Address
SHIFT-D: Display

When you enter a command, MLX jumps out of the line you've been typing, so we recommend you do it at a new prompt. Use the Save command to save what you've been working on. It will save on tape or disk as if you've finished, but the tape or disk won't work, of course, until you finish the typing. Remember what address you stop at. The next time you run MLX, answer all the prompts as you did before, then insert the disk or tape. When you get to the entry prompt, press SHIFT-L to reload the partly completed file into memory. Then use the New Address command to resume typing.

To use the New Address command, press SHIFT-N and enter the address where you previously stopped. The prompt will change, and you can then continue typing. Always enter a New Address that matches up with one of the line numbers in the special listing, or else the checksum won't work. The Display command lets you display a section of your typing. After you press SHIFT-D, enter two addresses within the line number range of the listing. You can abort the listing by pressing any key.

What if you forgot where you stopped typing? Use the Display command to scan memory from the beginning to the end of the program. When you reach the end of your typing, the lines will contain a random pattern of numbers. When you see the end of your typing, press any key to stop the listing. Use the New Address command to continue typing from the proper location.

MLX: Machine Language Entry

```
10 REM LINES CHANGED FROM MLX VERSION 2.0
   0 ARE 750,765,770 AND 860        :rem 50
100 PRINT"{CLR}[6]";CHR$(142);CHR$(8);:
    POKE53281,1:POKE53280,1         :rem 67
101 POKE 788,52:REM DISABLE RUN/STOP
                                   :rem 119
110 PRINT"{RVS}{39 SPACES}";       :rem 176
120 PRINT"{RVS}{14 SPACES}{RIGHT}{OFF}
    [*]£{RVS}{RIGHT} {RIGHT}{2 SPACES}
```

```
    {*}{OFF}{*}£{RVS}£{RVS}
    {14 SPACES}"";                       :rem 250
130 PRINT"{RVS}{14 SPACES}{RIGHT} {G}
    {RIGHT} {2 RIGHT} {OFF}£{RVS}£{*}
    {OFF}{*}{RVS}{14 SPACES}";           :rem 35
140 PRINT"{RVS}{41 SPACES}"             :rem 120
200 PRINT"{2 DOWN}{PUR}{BLK} MACHINE LANG
    UAGE EDITOR VERSION 2.01{5 DOWN}"
                                         :rem 237
210 PRINT"{5}{2 UP}STARTING ADDRESS?
    {8 SPACES}{9 LEFT}";                 :rem 143
215 INPUTS:F=1-F:C$=CHR$(31+119*F)
                                         :rem 166
220 IFS<256OR(S>40960ANDS<49152)ORS>53247
    THENGOSUB3000:GOTO210                :rem 235
225 PRINT:PRINT:PRINT .                  :rem 180
230 PRINT"{5}{2 UP}ENDING ADDRESS?
    {8 SPACES}{9 LEFT}";:INPUTE:F=1-F:C$=
    CHR$(31+119*F)                       :rem 20
240 IFE<256OR(E>40960ANDE<49152)ORE>53247
    THENGOSUB3000:GOTO230                :rem 183
250 IFE<STHENPRINTC$;"{RVS}ENDING < START
    {2 SPACES}":GOSUB1000:GOTO 230
                                         :rem 176
260 PRINT:PRINT:PRINT                    :rem 179
300 PRINT"{CLR}";CHR$(14):AD=S:POKEV+21,0
                                         :rem 225
310 A=1:PRINTRIGHT$("0000"+MID$(STR$(AD),
    2),5);":";                           :rem 33
315 FORJ=ATO6                            :rem 33
320 GOSUB570:IFN=-1THENJ=J+N:GOTO320
                                         :rem 228
390 IFN=-211THEN 710                     :rem 62
400 IFN=-204THEN 790                     :rem 64
410 IFN=-206THENPRINT:INPUT"{DOWN}ENTER N
    EW ADDRESS";ZZ                       :rem 44
415 IFN=-206THENIFZZ<SORZZ>ETHENPRINT"
    {RVS}OUT OF RANGE":GOSUB1000:GOTO410
                                         :rem 225
417 IFN=-206THENAD=ZZ:PRINT:GOTO310
                                         :rem 238
420 IF N<>-196 THEN 480                  :rem 133
430 PRINT:INPUT"DISPLAY:FROM";F:PRINT,"TO
    ";:INPUTT                            :rem 234
440 IFF<SORF>EORT<SORT>ETHENPRINT"AT LEAS
    T";S;"{LEFT}, NOT MORE THAN";E:GOTO43
    0                                    :rem 159
450 FORI=FTOTSTEP6:PRINT:PRINTRIGHT$("000
    0"+MID$(STR$(I),2),5);":";           :rem 30
451 FORK=0TO5:N=PEEK(I+K):PRINTRIGHT$("00
    "+MID$(STR$(N),2),3);",";            :rem 66
460 GETA$:IFA$>""THENPRINT:PRINT:GOTO310
                                         :rem 25
470 NEXTK:PRINTCHR$(20);:NEXTI:PRINT:PRIN
    T:GOTO310                            :rem 50
480 IFN<0 THEN PRINT:GOTO310             :rem 168
490 A(J)=N:NEXTJ                         :rem 199
500 CKSUM=AD-INT(AD/256)*256:FORI=1TO6:CK
    SUM=(CKSUM+A(I))AND255:NEXT          :rem 200
510 PRINTCHR$(18);:GOSUB570:PRINTCHR$(146
    );                                   :rem 94
511 IFN=-1THENA=6:GOTO315                :rem 254
515 PRINTCHR$(20):IFN=CKSUMTHEN530
                                         :rem 122
520 PRINT:PRINT"LINE ENTERED WRONG : RE-E
    NTER":PRINT:GOSUB1000:GOTO310:rem 176
530 GOSUB2000                            :rem 218
540 FORI=1TO6:POKEAD+I-1,A(I):NEXT:POKE54
    272,0:POKE54273,0                    :rem 227
550 AD=AD+6:IF AD<E THEN 310             :rem 212
560 GOTO 710                             :rem 108
570 N=0:Z=0                              :rem 88
580 PRINT"{£}";                          :rem 81
581 GETA$:IFA$=""THEN581                 :rem 95
582 AV=-(A$="M")-2*(A$=",")-3*(A$=".")-4*
    (A$="J")-5*(A$="K")-6*(A$="L"):rem 41
583 AV=AV-7*(A$="U")-8*(A$="I")-9*(A$="O"
    ):IFA$="H"THENA$="0"                 :rem 134
584 IFAV>0THENA$=CHR$(48+AV)             :rem 134
585 PRINTCHR$(20);:A=ASC(A$):IFA=13ORA=44
    ORA=32THEN670                        :rem 229
590 IFA>128THENN=-A:RETURN               :rem 137
600 IFA<>20 THEN 630                     :rem 10
610 GOSUB690:IFI=1ANDT=44THENN=-1:PRINT"
    {OFF}{LEFT} {LEFT}";:GOTO690         :rem 62
620 GOTO570                              :rem 109
630 IFA<48ORA>57THEN580                  :rem 105
640 PRINTA$;:N=N*10+A-48                 :rem 106
650 IFN>255 THEN A=20:GOSUB1000:GOTO600
                                         :rem 229
660 Z=Z+1:IFZ<3THEN580                   :rem 71
670 IFZ=0THENGOSUB1000:GOTO570           :rem 114
680 PRINT",";:RETURN                     :rem 240
690 S%=PEEK(209)+256*PEEK(210)+PEEK(211)
                                         :rem 149
691 FORI=1TO3:T=PEEK(S%-I)               :rem 67
695 IFT<>44ANDT<>58THENPOKES%-I,32:NEXT
                                         :rem 205
700 PRINTLEFT$("{3 LEFT}",I-1);:RETURN
                                         :rem 7
710 PRINT"{CLR}{RVS}*** SAVE ***{3 DOWN}"
                                         :rem 236
715 PRINT"{2 DOWN}(PRESS {RVS}RETURN{OFF}
    ALONE TO CANCEL SAVE){DOWN}":rem 106
720 F$="":INPUT"{DOWN} FILENAME";F$:IFF$=
    ""THENPRINT:PRINT:GOTO310            :rem 71
730 PRINT:PRINT"{2 DOWN}{RVS}T{OFF}APE OR
    {RVS}D{OFF}ISK: (T/D)"              :rem 228
740 GETA$:IFA$<>"T"ANDA$<>"D"THEN740
                                         :rem 36
750 DV=1-7*(A$="D"):IFDV=8THENF$="0:"+F$:
    OPEN15,8,15,"S"+F$:CLOSE15           :rem 212
760 T$=F$:ZK=PEEK(53)+256*PEEK(54)-LEN(T$
    ):POKE782,ZK/256                     :rem 3
762 POKE781,ZK-PEEK(782)*256:POKE780,LEN(
    T$):SYS65469                         :rem 109
763 POKE780,1:POKE781,DV:POKE782,1:SYS654
    66                                   :rem 69
765 K=S:POKE254,K/256:POKE253,K-PEEK(254)
    *256:POKE780,253                     :rem 17
766 K=E+1:POKE782,K/256:POKE781,K-PEEK(78
    2)*256:SYS65496                      :rem 235
770 IF(PEEK(783)AND1)OR(191ANDST)THEN780
                                         :rem 111
775 PRINT"{DOWN}DONE.{DOWN}":GOTO310
                                         :rem 113
780 PRINT"{DOWN}ERROR ON SAVE.{2 SPACES}T
    RY AGAIN.":IFDV=1THEN720             :rem 171
781 OPEN15,8,15:INPUT#15,E1$,E2$:PRINTE1$
    ;E2$:CLOSE15:GOTO720                 :rem 103
790 PRINT"{CLR}{RVS}*** LOAD ***{2 DOWN}"
                                         :rem 212
795 PRINT"{2 DOWN}(PRESS {RVS}RETURN{OFF}
    ALONE TO CANCEL LOAD)"               :rem 82
800 F$="":INPUT"{2 DOWN} FILENAME";F$:IFF
    $=""THENPRINT:GOTO310                :rem 144
810 PRINT:PRINT"{2 DOWN}{RVS}T{OFF}APE OR
    {RVS}D{OFF}ISK: (T/D)"              :rem 227
820 GETA$:IFA$<>"T"ANDA$<>"D"THEN820
                                         :rem 34
830 DV=1-7*(A$="D"):IFDV=8THENF$="0:"+F$
                                         :rem 157
840 T$=F$:ZK=PEEK(53)+256*PEEK(54)-LEN(T$
    ):POKE782,ZK/256                     :rem 2
```

```
841 POKE781,ZK-PEEK(782)*256:POKE780,LEN(
    T$):SYS65469                    :rem 107
845 POKE780,1:POKE781,DV:POKE782,1:SYS654
    66                              :rem 70
850 POKE780,0:SYS65493              :rem 11
860 IF(PEEK(783)AND1)OR(191ANDST)THEN870
                                    :rem 111
865 PRINT"{DOWN}DONE.":GOTO310      :rem 96
870 PRINT"{DOWN}ERROR ON LOAD.{2 SPACES}T
    RY AGAIN.{DOWN}":IFDV=1THEN800
                                    :rem 172
880 OPEN15,8,15:INPUT#15,E1$,E2$:PRINTE1$
    ;E2$:CLOSE15:GOTO800            :rem 102
1000 REM BUZZER                     :rem 135
1001 POKE54296,15:POKE54277,45:POKE54278,
     165                            :rem 207
1002 POKE54276,33:POKE 54273,6:POKE54272,
     5                              :rem 42
1003 FORT=1TO200:NEXT:POKE54276,32:POKE54
     273,0:POKE54272,0:RETURN       :rem 202
2000 REM BELL SOUND                 :rem 78
2001 POKE54296,15:POKE54277,0:POKE54278,2
     47                             :rem 152
2002 POKE 54276,17:POKE54273,40:POKE54272
     ,0                             :rem 86
2003 FORT=1TO100:NEXT:POKE54276,16:RETURN
                                    :rem 57
3000 PRINTC$;"{RVS}NOT ZERO PAGE OR ROM":
     GOTO1000                       :rem 89
```

©

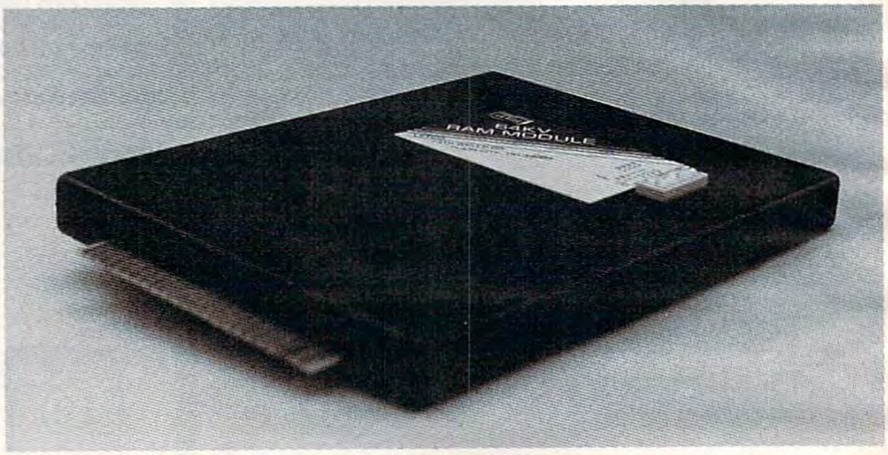
Memory Expander For VIC-20

Letco has announced the 64KV Memory Module, which adds more than 64K of memory to your VIC-20.

The 64KV houses 8K in each of the VIC's blocks 1, 2, and 3. Block 3 can also be paged, or swapped, under program control, with five other separate 8K sections of memory. Each block has a separate enable switch and a write-protect switch, and there is a switch to make block 3 respond as though it is block 5 (the normal game block).

The module is priced at $109.95

Letco
7310 Wells Road
Plain City, OH 43064
(614) 873-4410



The Letco 64KV Memory Module adds more than 64K RAM memory to your VIC-20.

Authoring System And Teaching Tool

CLAS, a teaching tool and authoring system for educators, has been released by Touch Technologies for the Apple II+ and IIe, the IBM PC and PCjr, and the Commodore 64.

The software package functions as a teaching tool for any subject. Authoring procedures allow instructors to create lessons in their own teaching style. Up to 30 problem sets can be offered with each lesson. Questions take the form of true/false, multiple choice, short answer, or matching.

If desired, the questions can be presented in a different order each time the lesson is used.

Sound is used to give feedback when a response is made to a question. A help mode is provided for the student, along with a review of problem areas and a summary of performance at the end of the lesson.

Memory requirement for Apple computers is 48K. The IBMs must use DOS 2.0/2.1. *CLAS* is available for $89.95.

Touch Technologies
609 S. Escondido Blvd.
Ste. 101
Escondido, CA 92025
(619) 743-0494

Interface For TI-99/4A

Mikel Laboratories, Inc., has announced an RS-232-C interface system for the TI-99/4A.

The $145.95 system is a free-standing unit which allows the TI-99/4A to use a printer and modem without a peripheral expansion unit.

The company also offers cassette interface systems ($49.95), TI cassette cables ($11.95), and printers and monitors. A line of personal computer accessories for the TI-99/4A will soon be available from Mikel Laboratories.

Mikel Laboratories
3341 W. El Segundo Blvd.
Hawthorne, CA 90250
(213) 679-2542

Life Insurance Program For Atari, Commodore

Advanced Financial Planning has released *Life Insurance Planning*, a software package for the Atari 400 and 800 computers and the Commodore 64.

The program will calculate the inflation rate applicable to a user's budget; the user's total estate needs reduced into terms of today's dollars (such as future living expenses for the family, college expenses, and funeral expenses); the total estate provided by all sources of income and assets; and the total shortfall needed to be provided by life insurance.

Life insurance needs can be calculated for any year over the planning period in order to help the user select the proper type of insurance policy.

Life Insurance Planning supports virtually any printer, and requires a disk drive. The Atari version requires the Atari BASIC cartridge and 32K RAM. The package is priced at $29.95. When purchased with Advanced Financial Planning's *Retirement Planning* program, the total price is $49.95 (shipping prices are included in this total).

Advanced Financial Planning
20922 Paseo Olma
El Toro, CA 92630
(714) 855-1578

Music Adventure Games For Apple II

Syntauri Corporation has introduced Musicland, an advanced set of musical games for the Apple II.

The package is built from four basic games—*Sound Factory, Timbre Painting, Music Doodles,* and *Music Blocks*. The four games are integrated. Musicland is a foundation program from which advanced musical concepts and structures may be taught to young children.

The system attempts to maintain the interest of young students, while providing musical challenges for adult musicians as well. Aimed at musically untrained children, Musicland lets youngsters use joystick controls to discover musical form, timbre, orchestration, composition, and transposition.

Children can compose, edit, and play music as well. Interactive graphics aid exploration, from sketching a simple melody to inverting a complex musical passage. Multipart pieces can be composed, orchestrated, and played back in stereo. Each of the four games covers a different range of musical learning experiences.

Musicland requires a 64K Apple II computer system with one disk drive, plus synthesizers (the Mountain Computer MusicSystem) which plug into the Apple. The Musicland set with manuals sells for $150. The synthesizers are available for under $400.

Syntauri Corporation
4962 El Camino Real
Suite 112
Los Altos, CA 94022
(415) 966-1273

Three Learning Programs For Atari, Commodore

Three learning programs from Carousel Software have been released for the Commodore 64 and Atari computers on disk or cassette.

Telly Turtle is an introduction to computer programming which uses drawing routines and emphasizes logical thinking, problem solving, numbers sequencing, and visual discrimination.

Brain Strainers includes three learning games for from one to four players: *Clef Climber*, a multilevel, animated note recognition game; *Finders Keepers*, a multiscreen and multilevel concentration game; and *Follow the Leader*, a music and graphic pattern recognition game with up to 44 levels of difficulty.

Simulated Computer is an animated simulation of a computer in operation. Programs written by the user can be seen and heard flowing through the component parts of the computer. The program serves as a teaching tool about the way a computer works.

Telly Turtle (34.95) and *Brain Strainers* ($29.95) are meant for ages five to adult. *Simulated Computer* ($29.95) is directed toward ages 12 to adult.

Carousel Software, Inc.
877 Beacon Street
Boston, MA 02215
(617) 437-9419

Games, Tutorial For Commodore 64

Advanced Microware has introduced two new software products for the Commodore 64.

Casino Pac includes four games—Blackjack, Poker, Keno, and Slot Machine. Each simulates the new videogaming machines being used in gambling centers such as Las Vegas and Atlantic City. The games let you practice your betting strategy, try your own betting systems, or play for fun.

Casino Pac sells on tape or disk for $39.

64Tour is a tour of the features and capabilities of the Commodore 64, with demonstrations of all the graphics modes, as well as music and sound effects. The package is priced at $12.

Advanced Microware
P.O. Box 6143
Santa Ana, CA 92706
(714) 554-6470

New Product releases are selected from submissions for reasons of timeliness, available space, and general interest to our readers. We regret that we are unable to select all new product submissions for publication. Readers should be aware that we present here some edited version of material submitted by vendors and are unable to vouch for its accuracy at time of publication.

COMPUTE! *welcomes notices of upcoming events and requests that the sponsors send a short description, their name and phone number, and an address to which interested readers may write for further information. Please send notices at least three months before the date of the event, to: Calendar, P.O. Box 5406, Greensboro, NC 27403.* ©

COMPUTER MAIL ORDER

NEC PRINTERS
NEC 2050 .... $999.00
NEC 3550 .... $1679.00

PERCOM/TANDON DRIVES
5¼" 320K Floppy $229.00
5 Meg Hard w/Controller ...... CALL
10 Meg Hard w/Controller ...... CALL
15 Meg Hard w/Controller ...... CALL
20 Meg Hard w/Controller ...... CALL

AMDEK
310A Amber Monitor $169.00
DXY 100 Plotter $599.00
Color II $399.00

AST RESEARCH
Six Pak Plus...from $279.00
Combo Plus II...from $279.00
Mega Plus...from $309.00
I/O Plus...from $139.00

QUADRAM
Quadlink $479.00
Quadboard...as low as $289.00
Quad 512 Plus...as low as... $249.00
Quadcolor...as low as $219.00
Chronograph $89.00
Parallel Interface Board $89.00
64K RAM Chips Kit $59.00

MICROPRO
WordStar/MailMerge $349.00
InfoStar $299.00
SpellStar $159.00
CalcStar $99.00

MICROSTUF
Crosstalk $105.00

MICROSOFT
Multiplan $159.00

ASHTON TATE
dBASE II $389.00
Friday! $185.00

IUS
EasyWriter II $249.00
EasySpeller $119.00
EasyFiler $229.00

CONTINENTAL SOFTWARE
1st Class Mail/Form Letter...$79.00
The Home Accnt. Plus $88.00

IBM

VISICORP

	IBM	APPLE
VisiCalc		159.00
VisiCalc 4	159.00	
VisiCalc-Advanced		269.00
VisiWord/Spell	249.00	
Visitrend/Plot	199.00	199.00
VisiLink		169.00
VisiFile	199.00	169.00
VisiSchedule	199.00	199.00
Visidex		159.00
VisiPlot		135.00
VisiTerm		75.00
Desktop Plan	199.00	169.00
Bus. Forecast Model	75.00	75.00
Stretch Calc	75.00	75.00
VisiTutor Calc	59.00	59.00
VisiTutor Advanced	75.00	75.00
VisiTutor Word	59.00	59.00
Vision Calc	249.00	
Vision Graph	129.00	
Vision Mouse	159.00	
Vision Host	319.00	

pfs

	APPLE	IBM
Write:	79.00	89.00
Graph:	79.00	89.00
Report:	79.00	79.00
File:	79.00	89.00
Solutions*: as low as	16.00	16.00

*Call On Titles

LOTUS
1-2-3 .... $329.00

PROFESSIONAL SOFTWARE
PC Plus/The Boss $349.00

SYNAPSE
File Manager $89.00

MBC-550 PC ... CALL
MBC-555 PC ... CALL
MBC 1100 .... $1499.00
FDD 3200-320K Drive $389.00
MBC 1150 .... $1899.00
MBC 1200 .... $1849.00
FDD 6400-640K Drive $469.00
MBC 1250 .... $2099.00
PR 5500 Printer $599.00

APPLE/FRANKLIN DISK DRIVES

MICRO-SCI
A2 .... $219.00
A40 .... $299.00
A70 .... $319.00
C2 Controller $79.00
C47 Controller $89.00

RANA
Elite 1 .... $279.00
Elite 2 .... $389.00
Elite 3 .... $569.00

APPLE IIe STARTER PACK
64K Apple IIe, Disk Drive & Controller, 80 Column Card, Monitor II & DOS 3.3
COMPLETE .... CALL

TERMINALS
914 .... $569.00
924 .... $689.00
925 .... $739.00
950 .... $929.00
970 .... $1039.00

COMPUTERS
Teleport Portable .... CALL
800A $1099.00
802 .... $2699.00
803 .... $1949.00
802H $4695.00
806/20 .... $4999.00
816/40 .... $9199.00
1602 .... $3399.00
1603 .... CALL

SANYO

FRANKLIN

ACE 1000 Color Computer .... CALL
ACE Family Pack System ...... CALL
ACE PRO PLUS System .... CALL
ACE 1200 Office Mgmt. System ... CALL
"NOT IIE EXPENSIVE"

TeleVideo

PRINTERS

AXIOM
AT-100 Atari Interface $239.00
GP-100 Parallel Interface ... $199.00
AT-846 Interface $89.00

AXIOM BMC
401 Letter Quality $589.00
BX-80 Dot Matrix $269.00

CENTRONICS
122 Parallel $299.00
739-1 Parallel $199.00
739-3 Serial $249.00

C.ITOH
Gorilla Banana $209.00
Prowriter 8510P $379.00
Prowriter 1550P $599.00
A10 (18 cps) $569.00
8600P $929.00
F10-40 .... $999.00
F10-55 .... $1499.00

COMREX
ComWriter II Letter Quality ... $499.00

DIABLO
620 Letter Quality $949.00
630 Letter Quality $1749.00

DAISYWRITER
2000 .... $999.00
Tractor Feed $109.00

EPSON
MX-80FT, MX-100, RX-80, RX-80FT, FX-80, FX-100 .... CALL

IDS
Prism 80...For Configurations...CALL
Prism 32...For Configuration...CALL

MANNESMAN TALLY
160L $589.00
180L $799.00
Spirit 80 .... $309.00

NEC
2010/2030 .... $899.00
8023 Dot Matrix $379.00
8025 Dot Matrix $669.00
3510 Serial/Letter Quality ... $1449.00
3530 Parallel/Letter Quality ... $1499.00
7710/7730 Serial/Parallel ... $1949.00

OKIDATA
82, 83, 84, 92, 93, 2350, 2410 ... CALL

SMITH CORONA
TP-2 .... $399.00
Tractor Feed $119.00

SILVER REED
500 Letter Quality $469.00
550 Letter Quality $699.00

STAR
Gemini 10X $299.00
Gemini P15X $399.00
Delta 10 .... $559.00
Serial Board $75.00

TOSHIBA
1350 .... $1599.00
TRANSTAR .... CALL

PRINTER CABLES
Available for Atari, Commodore, IBM, Apple, Epson, Kaypro, Televideo, Franklin, Eagle, Sanyo, Osborne, NEC, Zenith and many others. We supply all your computer needs!

PAPER SUPPLIES
1000 shts. 8½x11 Tractor Paper...19.99
1000 shts. 14½x11 Tractor Paper..$24.99
1 or 2" Address Labels $9.99

MONITORS

AMDEK
300 Green $149.00
300 Amber $159.00
310 Amber $169.00
Color 1 .... $279.00
Color 1 Plus $299.00
Color 2 .... $399.00
Color 2 Plus $419.00
Color 3 .... $349.00
Color 4 .... $699.00

BMC
12" Green $88.99
12" Green HI-RES $119.99
9191-13" Color $249.99

GORILLA
12" Green $88.99
12" Amber $95.99

NEC
JB 1260 Green $109.00
JB 1201 Green $149.99
JB 1205 Amber $159.99
JC 1215 Color $299.99
JC 1216 RGB $429.00
JC 1460 Color RGB $359.00

PRINCETON GRAPHICS
HX-12 RGB $519.00

SAKATA
100 .... $269.00

TAXAN
210 Color RGB $299.00
400 Med-Res RGB $319.00
415 Hi-Res RGB $439.00
420 Hi-Res RGB (IBM) $489.00
100 12" Green $125.00
105 12" Amber $135.00

USI
Pi 1, 9" Green $99.99
Pi 2, 12" Green $119.99
Pi 3, 12" Amber $149.99
Pi 4, 9" Amber $139.99
1400 Color $269.99

QUADRAM
Quadchrome 8400 .... $579.00

ZENITH
ZVM 122 Amber $109.00
ZVM 123 Green $89.99
ZVM 135 Color/RGB $469.99

MODEMS

ANCHOR
Mark I (RS-232) $79.00
Mark II (Atari) $79.00
Mark III (TI-99) $109.00
Mark IV (CBM/PET) $125.00
Mark V (Osborne) $95.00
Mark VI (IBM-PC) $169.00
Mark VII (Auto Ans/Auto Dial) ... $119.00
Mark XII (1200 Baud) $299.00
TRS-80 Color Computer $99.00
9 Volt Power Supply $9.00

HAYES
Smartmodem 300 .... $219.00
Smartmodem 1200 .... $509.00
Smartmodem 1200B $459.00
Micromodem II $265.00
Micromodem II Plus $299.00
Micromodem IIE $269.00
Micromodem 100 .... $299.00
Smart Com II $89.00
Chronograph $199.00

NOVATION
J-Cat $99.99
SmartCat 103 .... $179.00
SmartCat 103/212 .... $399.00
AutoCat $219.00
212 AutoCat $549.00
Apple Cat II $249.00
212 Apple Cat $569.00
Apple Cat 212 Upgrade $309.00
Cat $139.99
D-Cat $149.00
PC-Cat $399.00

ZENITH
ZT-1 .... $309.00
ZT-10 .... $339.00
ZT-11 .... $369.00

APPLE INTERFACE CARDS & BUFFERS
Choose from PKASO, Orange Micro, MPC, MicroMax, Tymac, Quadram & Practical Peripherals .... CALL

WEST
800-648-3311
VISA
In NV call (702)588-5654, Dept. 0506
Order Status Number: 588-5654
P.O. Box 6689, Stateline, NV 89449

CANADA
800-268-4559
In Toronto call (416)828-0866, Dept. 0506
Order Status Number: 828-0866
2505 Dunwin Drive, Unit 1B
Mississauga, Ontario, Canada L5L1T1

EAST
800-233-8950
In PA call (717)327-9575, Dept. 0506
Order Status Number: 327-9576
Customer Service Number: 327-1450
477 E. Third St., Williamsport, PA 17701

No risk, no deposit on C.O.D. orders. Pre-paid orders receive free shipping within the UPS Contintental United States with no waiting period for certified checks or money orders. Add 3% (minimum $5.00) shipping and handling on all C.O.D. and credit card orders. Larger shipments may require additional charges. NV and PA residents add sales tax. All items subject to availability and price change. We stock manufacturer's and third party software for most all computers on the market. Call today for our new catalog.

COMPUTER MAIL ORDER

Koala
Technologies Corporation

KOALA PADS

Atari (Disk)	$75.00
Atari (ROM)	$82.00
C-64 (DISK)	$75.00
C-64 (ROM)	$82.00
IBM	$99.00
AP/Franklin	$85.00

commodore

CBM 8023... $599

MSD SD1 Disk Drive	$349.00
MSD SD2 Disk Drive	$599.00
CBM 4032	$599.00
CBM 8096	$869.00
CBM 9000	$999.00
B128-80	$769.00
CBM 64K Memory Board	$269.00
8032 to 9000 Upgrade	$269.00
2031LP Disk Drive	$299.00
8050 Disk Drive	$949.00
8250 Disk Drive	$1199.00
4023 Printer	$379.00
8023 Printer	$569.00
6400 Printer	$1399.00
Z-RAM	$499.00
Silicon Office	$699.00
The Manager	$199.00
Soft ROM	$125.00
VisiCalc	$159.00

PROFESSIONAL SOFTWARE

Word Pro 2 Plus	$159.00
Word Pro 3 Plus	$189.00
Word Pro 4 Plus/5 Plus...each	$279.00
InfoPro	$179.00
Administrator	$399.00
Power	$79.00

SX-64 PORTABLE
$839.00

VIC 20 .... CALL
CBM 64 .... $199

C1541 Disk Drive	$249.00
C1530 Datasette	$69.00
C1520 Color Printer/Plotter	$129.00
M-801 Dot Matrix Printer	$219.00
C1526 Dot Matrix/Serial	$299.00
C1702 Color Monitor	$249.00
C1311 Joystick	$4.99
C1312 Paddles	$11.99
C1600 VIC Modem	$59.00
C1650 Auto Modem	$89.00
Logo 64	$49.00
Pilot 64	$39.00
Simon's Basic	$19.00
Word Pro 64 Plus	$59.00
Parallel Printer Interface	$49.00
Calc Result 64	$129.00
Codewriter 64	$75.00
Quick Brown Fox	$49.00
Word Pro 64 Plus	$59.00
MCS 801 Color Printer	$499.00
DPS 1101 Daisy Printer	$459.00
Magic Voice Speech Module	$54.00
Desk Organizer Lock	$49.00

We stock a full inventory of software for Commodore, such as: Artworx, Broderbund, Commercial Data, Creative Software, Epyx, HES, MicroSpec, Nufekop, Romox, Sirius, Synapse, Thorn EMI, Tronix, UMI, Victory, Spinnaker, Rainbow & Timeworks!

ATARISOFT
Games for IBM, Apple, C64 & VIC 20

	IBM/Apple	C64/VIC 20
Pac Man	29.99	37.99
Centipede	29.99	37.99
Dig Dug	29.99	37.99
Donkey Kong	29.99	37.99
Defender	29.99	37.99
Robotron	29.99	37.99
Star Gate	29.99	37.99

CMO'S PORTABLE CORNER

hp

HP 71B
$499.99

41 CV........ $129.99
41 CX........ $199.99

HP 10C	$51.99
HP 11C	$69.99
HP 12C	$88.99
HP 15C	$88.99
HP 16C	$88.99
HP 75C	$749.99
HPIL Module	$98.99
HPIL Cass. or Printer	$359.99
Card Reader	$143.99
Extended Function Module	$63.99
Time Module	$63.99

TIMEX/SINCLAIR

Timex/Sinclair 1000	$24.99
Timex/Sinclair 2086	CALL
16K Memory	$25.00
2040 Printer	$99.99
VuCalc	$17.99
Mindware Printer	$99.99

PC-8201
$599

NEC

PC-8221A Thermal Printer	$149.00
PC-8281A Data Recorder	$99.00
PC-8201-06 8K RAM Chips	$105.00
PC-8206A 32K RAM Cartridge	$329.00

SHARP

PC-1500A... $165.99
PC-1250A.... $88.99

CE-125 Printer/Cassette	$128.99
CE-150 Color Printer/Cassette	$171.99
CE-155 8K RAM	$93.99
CE 161 16K RAM	$134.99
CE 500 ROM Library	$29.99

ATARI®

HOME COMPUTERS

600XL
$99✲

✲With Purchase of 1010 Program Recorder, E.T. Phone Home & C1311 Joysticks

TOTAL SYSTEM PRICE $259

1010 Recorder	$74.00
1020 Color Printer	$249.00
1025 Dot Matrix Printer	$339.00
1027 Letter Quality	$309.00
1030 Direct Connect Modem	$119.00
1050 Disk Drive	$339.00
CX30 Paddle	$12.00
CX40 Joystick...each	$8.00
CX77 Touch Tablet	$64.00
CX80 Trak Ball	$48.00
CX85 Keypad	$105.00
488 Communicator II	$229.00
4003 Assorted Education	$47.00
4011 Star Raiders	$33.00
4012 Missile Command	$29.00
4013 Asteroids	$29.00
5049 VisiCalc	$159.00
7097 Logo	$79.00
7101 Entertainer	$69.00
7102 Arcade Champ	$75.00
8026 Dig Dug	$33.00
8030 E.T. Phone Home	$33.00
8031 Donkey Kong	$39.00
8033 Robotron	$35.00
8034 Pole Position	$39.00
8036 Atari Writer	$79.00
8040 Donkey Kong, Jr.	$39.00
8043 Ms. PacMan	$39.00
8044 Joust	$39.00

DISKETTES

MAXELL

5¼" MD-1	$29.00
5¼" MD-2	$39.00
8" FD-1 (SS/DD)	$39.00
8" FD-2 (DS/DD)	$49.00

VERBATIM

5¼" SS/DD	$26.99
5¼" DS/DD	$36.99

ELEPHANT

5¼" SS/SD	$18.49
5¼" SS/DD	$22.99
5¼" DS/DD	$28.99

HEAD

5¼" Disk Head Cleaner	$14.99

DISK HOLDERS

INNOVATIVE CONCEPTS

Flip-n-File 10	$3.99
Flip-n-File 50	$17.99
Flip-n-File(400/800ROM)Holder	$17.99

LJK ENTERPRISES

Atari Letter Perfect-Disk(40/80)	$79.99
Atari Letter Perfect-ROM(40 col)	$79.99
Atari Letter Perfect-ROM(80 col)	$79.99
Atari Data Perfect-ROM(80 col)	$79.99
Atari Spell Perfect-DISK	$59.99
Atari Utility/MailMerge	$21.00
Apple Letter Perfect	$99.00
Apple Data Perfect	$75.00
Apple LJK Utility	$21.00
Apple LowerCase Generator	$19.00

600XL...... $189
800XL...... $299
1200XL .... CALL
1400XL .... CALL

PERCOM

AT 88-S1	$369.00
AT 88-A1	$259.00
AT 88-S1PD	$449.00
AT 88-DDA	$119.00
RFD 40-S1	$449.00
RFD 40-A1	$269.00
RFD 40-S2	$699.00
RFD 44-S1	$539.00
RFD 44-S2	$869.00

TEXAS INSTRUMENTS

TX 99-S1	$279.00

RANA

1000	$329.00

TRAK

AT-D2	$389.00

INDUS

GT-Drive	$379.00

MEMORY BOARDS

Axlon 32K	$59.00
Axlon 48K	$99.00
Axlon 128K	$299.00
Intec 32K	$59.00
Intec 48K	$85.00
Intec 64K	$99.00
Intec Real Time Clock	$29.00

ALIEN VOICE BOX

Atari	$119.00
Apple	$149.00

CONTROLLERS & JOYSTICKS

WICO

Joystick	$21.99
3-way Joystick	$22.99
Famous Red Ball	$23.99
Power Grip	$21.99
BOSS Joystick	$17.99
ATARI/VIC Trak Ball	$34.99
Apple Trak Ball	$54.99
Apple Adapter	$15.99
Apple Analog	$37.99

KRAFT

Joystick	$41.99
Atari Single Fire	$12.99
Atari Switch Hitter	$15.99
Apple Paddles	$34.99
IBM Paddles	$34.99
IBM Joystick	$46.99

AMIGA

3100 Single	$13.99
3101 Pair	$19.99
Joyboard	$37.99

TG

Atari Trak Ball	$47.99
Apple Joystick	$47.99
Apple Trak Ball	$47.99

WEST
800-648-3311
In NV call (702)588-5654, Dept. 0506
Order Status Number: 588-5654
P.O. Box 6689, Stateline, NV 89449

CANADA
800-268-4559
In Toronto call (416)828-0866, Dept. 0506
Order Status Number: 828-0866
2505 Dunwin Drive, Unit 1B
Mississauga, Ontario, Canada L5L1T1

EAST
800-233-8950
In PA call (717)327-9575, Dept. 0506
Order Status Number: 327-9576
Customer Service Number: 327-1450
477 E. Third St., Williamsport, PA 17701

MasterCard

CANADIAN ORDERS: All prices are subject to shipping, tax and currency fluctuations. Call for exact pricing in Canada.
INTERNATIONAL ORDERS: All shipments outside the Continental United States must pre-pay by certified check only. Include 3% (minimum $5.00) shipping and handling. EDUCATIONAL DISCOUNTS: Additional discounts are available to qualified Educational Instutions.
APO & FPO: Add 3% (minimum $5.00) shipping and handling.

Lyco Computer Marketing & Consultants

TO ORDER CALL US **TOLL FREE 800-233-8760**

In PA 1-717-327-1824

PERCOM

FOR ATARI COMPUTERS

- AT88S1 .... $299.00
- AT88S2 .... $535.00
- AT**S1PD ... $439.00
- RFD40S1 ... $399.00
- RFD40S2 ... $675.00
- RFD44S1 ... $449.00
- AT88 doubler

HARD DISK DRIVES for APPLE IBM-PC

- 5MEG $1349.00
- 10MEG ... $1599.00
- 15MEG ... $1999.00
- 20MEG ... $2359.00

*Add $30.00 for TRS 80 Drives

TEXAS INSTRUMENT

Disk Drive ... $245.00

TRAK DISK DRIVES

- AT-D1 .... $379.00
- AT-D2 .... $399.00
- PRINTER CABLE $22.95
- Software for ATD-2 .... $22.95

RANA DISK DRIVE

COMPUTER CARE

BIB

- 5¼ DISK DRIVE CLEANER $12.75
- COMPUTER CARE KIT $19.75

BLANK DISKETTES

ELEPHANT

- Single Side SD (10) $17.75
- Single Side DD (10) $21.75
- Double Side DD (10) $26.75

MAXELL

- MD I (10) $28.75
- MD II (10) $38.75

CERTRON CASSETTES

- CC-10 12 for $15.99
- CC-20 12 for $17.99

INNOVATIVE CONCEPTS

- Disk Storage (holds 10) $4.95
- Disk Storage (holds 15) $9.95
- Disk Storage (holds 50) $26.95

apple®

SSI

- Battle of Shilo $26.75
- Tigers in the Snow $26.75
- Cosmic Balance $26.75
- Knights of the Desert . $26.75
- Battle for Normandy .. $26.75
- Germany 1985 .... $36.75

RANA DISK DRIVES

- Elite 1 .... $295.00
- Elite 2 .... $449.00
- Elite 3 .... $559.00

MICRO-SCI

- A2 .... call
- A40 .... call
- A70 .... call

MUSE

- Castle Wolfenstein $20.75
- Caverns of Frietag $20.75
- Robot War $26.75

CONTINENTAL

- Home Accountant $51.75
- Book of Apple Software . $16.75

BRODERBUND

- Bank Street Writer $49.75
- AE $24.75
- LODE RUNNER D $24.75
- Choplifter $24.75
- David's Midnight $24.75

SPINNAKER

- Kindercomp $21.75
- Story Machine $23.75
- FaceMaker $23.75
- Snooper Trooper $29.75
- Delta Drawing $34.75

EPYX

- Temple of Apshai $26.95
- Star Warrior $26.95
- Crush, Crumble & Chomp . $22.75

ADVENTURE

- Saga#1 Adventureland ... $29.95
- Saga#2 Pirate Adventure . $29.95
- Saga#3 Secret Mission ... $29.95
- Stone of Sisyphus $24.95

ALIEN GROUP

- Atari Voice Box $99.00
- Apple Voice Box $129.00

commodore

- 1212 Programmers Ad. $44.75
- 1213 Vicmon $44.75
- Vic 20 dust cover $6.99
- Vic 64 dust cover $6.99

TIMEWORKS

- INVENTORY $59.75
- ACCOUNTS REC. $59.75
- ACCOUNTS PAY. $59.75
- GENERAL LEDGER $59.75
- PAYROLL $59.75
- CASH FLOW $59.75
- SALES ANALYSIS $59.75
- ELEC. CHECKBOOK $59.75
- MONEY MANAGER $59.75
- DATA MANAGER $59.75
- WALL STREET $59.75

HES 64

- Sound Box $9.95
- 64Forth $55.75
- Hesmon $25.75
- Turtle Graphics $37.75
- Heswriter $28.75
- Gridrunner $19.75
- Attack of M C $22.75
- Turtle Trainer $22.75
- Turtle Tutor $22.75
- Paint Brush $22.75
- Benji $25.75
- Home Manager $28.75
- Time Money Mgr $44.97
- OmniCalc $33.75
- Sword Point $19.95
- Hes Modem $52.75

CARDCO

- Cardprinter / LQ1 .... $499.00
- Cardprint DM1 .... $109.00
- 5 Slot Expansion 64 ..... $54.00
- 64 Write NOW $39.00
- 64 Mail NOW $29.00
- 20 Write NOW $29.00
- 64 Keypad $29.00
- Universal Cass. Int. $29.75
- Printer Utility $19.75
- 6 Slot Expansion $79.95
- 3 Slot Expansion $24.95
- PRINTER INTERFACE $39.75
- PRINTER INTERFACE with full graphics $65.75
- LIGHT PEN $29.75

PARKER 20

- Frogger (ROM) $33.75
- QBert (ROM) $33.75
- Tutankham (rom) $33.75

SPINNAKER 64

- Kindercomp $21.75
- Story Machine $23.75
- Face Maker $23.75
- Snooper Trooper $29.75
- Delta Drawing $34.75
- Shamus II c/d $24.95
- Pinhead c/d $22.95

QUICK BROWN FOX

- QBF Word Processor $49.95

LJK

- Letter Perfect $105.00
- Data Perfect $95.00

ADVENTURE INTERNATIONAL

- S. Adams Adventure $28.75

VIC-64

- Household Finance C/D .. $24.75

VIC 20

- King Arthurs Heir Cass $24.75
- Monster Maze Rom $24.75

ATARI

Computers for people.

- 600XL ... $CALL
- 800XL .... for
- 1400XL ... Lowest
- 1450 ..... Prices
- 1020 PRINTER .... NOW
- 1025 PRINTER .... IN
- 1027 PRINTER .... STOCK
- 1050 DISK DRIVE .... $SAVES
- 1010 RECORDER $74.75

SYNAPSE

- BLUE MAX C/D $24.75
- Ft. APOCALYPSE C/D ... $24.75
- PHAROAH'S CURSE C/D ... $24.75

FIRST STAR

- ASTRO CHASE C/D $22.75
- BRISTOLS C/D $22.75
- FLIP FLOP C/D $22.75

ALIEN GROUP

- Voice Box 2 . $99.75

DON'T ASK

- Sam $41.75
- Abuse $15.95
- Teleatri $27.95
- Poker Sam .. $24.95

APX

- 3R Math $19.95
- Typo Attack $24.95
- Family Budget $19.95
- F. Cash Flow $19.95

BRODERBUND

- Bank Street Writer D $49.75
- AE D $24.75
- Apple Panic D $23.75
- Choplifter ROM $32.75
- David's Midnight $24.75
- Stellar Shuttle C/D $18.75
- Ft. Apocalypse $24.75

EPYX

- GATEWAY TO ASPHI R $28.75
- JUMPMAN JR R $28.75
- PIT STOP R $28.75
- GATEWAY TO

SSI

- Battle of Shilo C/D $26.75
- Tigers in the Snow C/D $26.75
- Battle for Normandy C/D .. $26.75
- Knights of the Desert C/D . $26.75
- Cosmic Balance C/D $26.75

ON-LINE

- Frogger $24.95
- Wizard & Prin $26.95

ROKLAN

- Wizard of War $29.75
- Gorf $29.75
- Delux Invader $27.95

BIG 5

- Miner 2049 .... $32.75

EASTERN HOUSE

- Monkey Wrench 2 .... $52.75

BRODERBUND

- LODE RUNNER D $24.75
- OPERATION WIRLWIND D $29.75
- DROL D $24.75

PARKER BROTHERS

- Tutankham R $33.75
- Super Cobra R $33.75
- Astro Chase R $33.75
- Frogger R $33.75
- QBert R $33.75
- Popeye R $33.75
- Risk R $42.75
- Chess R $42.75

SPINNAKER

- Story Machine R $26.75
- Face Maker R $24.75
- Kinderomp R $20.75
- Fraction Fever R $24.75
- Delta Drawing R $26.75

Lyco Computer Marketing & Consultants

TO ORDER CALL US TOLL FREE **800-233-8760**
In PA 1-717-327-1824

PRINTER INTERFACING — PRINTER PAPER AVAILABLE
Available for IBM PC, Apple, Atari, Vic 20 & Vic 64

SAVE on these in-stock PRINTERS

LETTER QUALITY

SMITH CORONA TP2	$449.00
DIABLO 630	$1719.00
ALPHACOM 42	$89.00
ALPHACOM 81	$129.00
NEC 8023	$369.00
NEC 8025	$699.00
NEC PC-8200 COMPUTER	$CALL

EPSON

RX-80	$SAVE$
RX-80FT	ON
FX-80	In-Stock
FX-100	EPSON
MX-80FT	PRINTERS
MX-100	$$CALL$$

MANNESMANN TALLY

SPIRIT 80	$CALL
MT 160L	$CALL

OKIDATA

80	$SAVE$
82A	CALL for
83A	LOWEST
84	PRICES
92	on these
93	In-Stock
PACEMARK 2350	PRINTERS

ATARI 850 REPLACEMENTS IN-STOCK

CITOH

GORILLA GX100	$179.00
PROWRITER 8510	$339.00
PROWRITER II	$659.00
8600	$1025.00
STARWRITER	$1099.00
PRINTMASTER	$1499.00

STAR MICRONTICS

GEMINI 10X	$269.00
GEMINI 15X	$CALL
DELTA 10	$479.00

MODEMS

ANCHOR MARK I	$79.00
ANCHOR MARK II	$79.00
HAYES SMART	$239.00
HAYES MICRO II	$309.00
Micro Bit MPP-1000	$129.75
NOVATION	
CAT	$144.00
D-CAT	$155.00
J-CAT	$115.00
APPLE CAT II	$279.00
212 APPLE CAT	$589.00

CORDLESS TELEPHONES from...$69.75

MONITORS

Sakata Color	$229.00
Amdek Color I	$275.00
Amdek 300 Green	$149.00
Amdek 300 Amber	$149.00
Gorilla Green	$99.00

DUST COVERS

800	$3.99
400	$3.99
1200	$3.99
410	$3.99
810	$3.99
1050	$5.99
PROWRITER	$5.99
GEMINI 10X	$5.99
PERCOM DISK	$5.99
VIC 20/64	$5.99

apple

APPLE DUMPLING GX	$99.75
APPLE DUMPLING 64 (16 Buffer)	$179.75
INFOCOM	
Zork I, II, or III	$26.75
Deadline	$33.75

commodore

HES 20

HES MON Rom	$25.75
Turtle Graphics	$25.75
HES Writer	$25.75
Shamus	$25.75
Protector	$25.75

VIC 64/20

PACMAN	$33.75
DONKEY KONG	$33.75
DIG DUG	$33.75
DEFENDER	$33.75
CENTIPEDE	$33.75

FIRST STAR 64

BRISTOLS C/D	$22.75
FLIP FLOP C/D	$22.75

SYNAPSE 64

ZEPPELIN C/D	$24.75
BLUE MAX C/D	$24.75
DIMENSION X C/D	$24.75

EPYX 64

ASPHI R	$28.75
JUMPMAN JR R	$28.75
PIT STOP R	$28.75

BRODERBUND 64

BANK STREET WRITER	$49.75
CHOPLIFTER	$24.75
LODE RUNNER	$24.75
DROL	$24.75

KOALA TOUCH TABLET...$69.75

PERCOM

AT88S1	$299.00
RDF44SI	$449.00

GEMINI 10 ...... $269.00

	IBM	ATARI	APPLE
PACMAN	$27.95	$29.75	$27.95
DONKEY KONG	$27.95	$29.75	$27.95
DIG DUG	$27.95	$29.75	$27.95
DEFENDER	$27.95	$29.75	$27.95
CENTIPEDE	$27.95	$29.75	$27.95
KOALA TOUCH TABLET	$99.75	$69.75	$84.75
DEADLINE	$34.75	$34.75	$34.75
ENCHANTER	$34.75	$34.75	$34.75
INFIDEL	$34.75	$34.75	$34.75
PLANETFALL	$34.75	$34.75	$34.75
STAR CROSS	$34.75	$34.75	$34.75
SUSPENDED	$34.75	$34.75	$34.75
WITNESS	$34.75	$34.75	$34.75
ZORK I	$34.75	$34.75	$34.75
ZORK II	$34.75	$34.75	$34.75
ZORK III	$34.75	$34.75	$34.75

48K RAM	$75.00
64K RAM	$99.00
ATARI WRITER	$79.00

BUSINESS

Visicalc	$159.75
Letter Perfect	$89.75
Letter Perfect	$89.75
Data Perfect	$89.75
TEXT WIZZARD	$34.75
SPELL WIZZARD	$34.75
File Manager	$69.75
Home File Mgr	$69.75
Bookeeper	$119.75
C.R.I.S.	$199.75

ATARI Computers for people.

TO ORDER

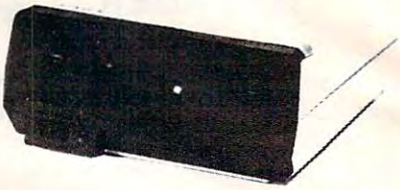
CALL TOLL FREE
800-233-8760
Customer Service 1-717-327-1825

or send order to
Lyco Computer
P.O. Box 5088
Jersey Shore, PA 1774C

POLICY

In-stock items shipped within 24 hours of order. Personal checks require four weeks clearance before shipping. No deposit on C.O.D. orders. Free shipping on prepaid cash orders within the continental U.S. PA residents add sales tax. All products subject to availability and price change. Advertised prices show 4% discount offered for cash, add 4% for Master Card or Visa. DEALER INQUIRIES INVITED.

YOU CAN'T TELL A DISK DRIVE BY ITS COVER!!



WITH A **HAPPY ENHANCEMENT** INSTALLED THESE ARE
THE MOST POWERFUL DISK DRIVES FOR YOUR ATARI COMPUTER

WARP SPEED SOFTWARE DISK READING AND WRITING 500% FASTER

HAPPY BACKUP — Easy to use backup of even the most heavily protected disks
HAPPY COMPACTOR — Combines 8 disks into 1 disk with a menu
WARP SPEED DOS — Improved Atari DOS 2.0S with WARP SPEED reading & writing
SECTOR COPIER — Whole disk read, write and verify in 105 seconds
1050 ENHANCEMENT — Supports single, 1050 double, and true double density
810 ENHANCEMENT — Supports single density

SPECIAL SUGGESTED RETAIL PRICE: Get the HAPPY ENHANCEMENT 810 or 1050 version with the HAPPY BACKUP PROGRAM, plus the multi drive HAPPY BACKUP PROGRAM, plus the HAPPY COMPACTOR PROGRAM, plus the HAPPY DRIVE DOS, plus the HAPPY SECTOR COPY, all with WARP DRIVE SPEED, including our diagnostic, a $350.00 value for only $249.95, for a limited time only! Price includes shipping by air mail to U.S.A. and Canada. Foreign orders add $10.00 and send an international money order payable through a U.S.A. bank. California orders add $16.25 state sales tax. Cashiers check or money order for immediate shipment from stock. Personal checks require 2-3 weeks to clear. Cash COD available by phone order and charges will be added. No credit card orders accepted. ENHANCEMENTS for other ATARI compatible drives coming soon, call for information. Specify 1050 or 810 ENHANCEMENT, all 1050s use the same ENHANCEMENT. Please specify -H model for all 810 disk drives purchased new after February 1982, call for help in 810 ENHANCEMENT model selection. Dealers now throughout the world, call for the number of the dealer closest to you.

ATARI is a registered trademark of Atari Computer Inc.

HAPPY COMPUTERS, INC. • P. O. Box 1268 • Morgan Hill, California 95037 • (408) 779-3830

PRODUCT MART

DUST COVERS

For Personal Computers, Peripherals, Game Units–Protective, Long-Lasting Vinyl Resists Both Dust and Liquids.

— CHOICE OF COLORS —

Amdek	IBM PC
Apple	Mattel
Atari	Rana Systems
BMC	Sanyo
Commodore	Star Micronics
Coleco	TI 99/4
Epson	TRS 80
Franklin Ace	*PLUS OTHERS*

GROUP/VOLUME DISCOUNTS AVAILABLE

FOR **FREE BROCHURE** WRITE:

ENCHANTED FOREST
P.O. Box 5261, Newport Beach, CA 92662
(1129 W. Balboa Blvd.)

Dealer Inquiries Invited

RENT SOFTWARE

1-2-3 Lotus 99.00*
Home Accountant 15.00*
PFS: Write 28.00*
VisiCalc 50.00*

FREE CATALOG

***Ask about our membership**

Games / Business / Utilities / Education

1-800-221-1031 California
1-800-221-4568 All Others
619-481-0559 San Diego

SOFTRENT™

COMPUCAT

Catalog of Computers and Supplies

Our prices are **WHOLESALE + 10%**

Samples!!!

ATARI 850 INTERFACE — $220

Compucat — $163

ATARI 1027 PRINTER — $350

Compucat — $265

We support the complete **ATARI** and **COMMODORE** product lines.
Ask for our free price list.

You may order in the regular manner or download our *TeleCatalog* and order from your computer or terminal.

(408) 353-1836

Instant shipping (or as fast as we can). *Mastercard & Visa Accepted* (no extra charge). Shipping & handling add 5%. California customers add 6.5% sales tax. Order by *phone* (Mon. - Fri. 10 am - 5 pm PST). Order by *modem* (daily 6 pm - 9 am) from our online TeleCatalog.

COMPUCAT

24500 Glenwood Hwy., Los Gatos, CA 95030

(M)agreeable™

software

STOCK HELPER™

Commodore 64™

Stock HELPER is a tool to maintain a history of stock prices and market indicators on diskette, to display charts, and to calculate moving averages. Stock HELPER was designed and written by a "weekend investor" for other weekend investors.

VISA $30.00 plus $1.25 shipping. MasterCard

Canadians may pay $37.50 CDN plus $1.55 shipping.

(M)agreeable software, inc.

5925 Magnolia Lane • Plymouth, MN 55442

(612) 559-1108

HELPER is a trademark of (M)agreeable Software, Inc.
Commodore is a trademark of Commodore Electronics Ltd.

BREAK-THRU

ONLY **$34.95**

A compact and inexpensive Eprom eraser for the hacker. It erases two chips per exposure, so if you are one of those smart people who only makes little mistakes and only needs to erase two Eproms at a time, this eraser is for you.

WALLING CO.

7755 E. Evans • Suite 400 • Scottsdale, AZ 85260
(602) 998-7550 Arizona residents add 6% tax

SAVE MORE THAN EVER ON 3M Scotch® DISKETTES

LIFETIME WARRANTY!

$1.85 ea. 5¼" SSDD ← (744) Qty. 20

$2.35 ea. 5¼" DSDD (745) → Qty. 20

5¼" SSQD—96TPI (746)	$2.60 ea.
5¼" DSQD—96TPI (747)	$3.25 ea.
8" SSSD (740)	$2.05 ea.
8" SSDD (741)	$2.50 ea.
8" DSDD (743)	$3.10 ea.

Shipping: 5¼" DISKETTES—Add $3.00 per 100 or fraction thereof. 8" DISKETTES—Add $4.00 per 100 or fraction thereof. OTHER ITEMS: Shipping charges as shown in addition to diskette shipping charges. **Payment:** VISA or MasterCard. COD orders only, add $3.00. **Taxes:** Illinois customers, please add 8%.

Hours: 9 AM-5 PM Central Time

For fast service call

Nationwide: 1-800-621-6827

In Illinois: 312-944-2788

DISK WORLD!

Suite 4806 • 30 East Huron Street • Chicago, Illinois 60611

Authorized Distributor Information Processing Products 3M

Data base Management

C= 64 Tape/Disk

N★ Horizon Disk

TRS-80 Disk

Define your own database;
Add; Change; Delete; Print.
Easy to use. Menu driven.

Only $39.00

Melcomp
P.O. Box 1085
Melbourne, FL 32901

NEC® 2000/3500 Replacement Ribbons

Multistrike & Nylon

Manufactured by Aspen Ribbons, Inc. Buy direct from manufacturer & save. Standard ink color is black. Red, green, blue, & brown colors are available for $2.00 extra— nylon.

PRICES

NEC® 2000/3500 MS	NEC® 2000/3500 Nylon
$3.25 to $6.00 ea.	$4.75 to $9.00 ea.

Price depends on quantity ordered.

CALL FOR FREE CATALOGUE.

*Aspen Ribbons, Inc. is not affiliated with any company mentioned in this ad.

Aspen Ribbons, Inc. 1700 N. 55th St.
Boulder, CO 80301-2796 (303)444-4054
Telex: 45-0055 End User: 800-525-0646
Wholesale: 800-525-9966

Commodore

B128-80

QUICK RECORD
$95.00

Full cursor editing
750 characters per record
Unlimited fields per record

Relative access	Math abilities
Sort by any field	Dymanic menu
Full screen format	Field replication

Quick Record is a fast and easy method of handling data. It is easy to learn yet retains the power needed in a data base.
SPECIFY DISK FORMAT

B128-80	$860.00
Transtar 130	$810.00
Transtar 120	$540.00
Prowriter 8510	$390.00

Tri S/C Software

The Computer GIN
2002A First Ave.
Dodge City, KS 67801
(316) 225-2628

COD's (add $5.00) FOB (Dodge City, KS)
2 weeks for checks KS Residents add 4%

VIC-20 COMMODORE 64

THE RECIPE BOX

Now you can easily store and recall your favorite recipes on your Commodore computer. THE RECIPE BOX is a complete menu-driven disk system that comes with these additional features:

SEARCH BY INGREDIENT - Only have a pound of hamburger in the freezer? Let THE RECIPE BOX show you all the recipes that you have on file that use hamburger, or any other ingredient you choose.

SEARCH BY CATEGORY/INGREDIENT - Any combination of the above.

AUTOMATIC MEASUREMENT - THE RECIPE BOX will automatically scale up or down the amount of ingredients you need according to how many servings you want.

SCREEN OR PRINTED OUTPUT - Have printed copies to use in the kitchen or give to friends.

THE RECIPE BOX requires one disk drive and will run on a 5K VIC-20, Commodore 64. Please specify. Send check or money order for 21.95 to:

Aries Marketing Co.
P.O. Box 4196
4200 Shannon Drive
Baltimore, MD 21205
MD residents add 5% sales tax

GREAT DISKETTES Super low prices SYNCOM

The low priced, high quality diskette with a LIFETIME WARRANTY. Packed in polybags of 10 with Tyvek envelopes, labels and reinforced hubs.

One of the best buys we've seen. **LIFETIME WARRANTY!**

$1.39 ◄ ea. 5¼"SSDD QTY. 20

$1.85 5¼" DSDD ea. ► QTY. 20

OTHER GREAT VALUES:

DISKETTE 70—Holds 70 5¼" diskettes in dust free safety .... **$14.95** ea. + $3.00 Shpng.

DISK CADDIES—Flip up style holds 10 5¼" diskettes ... **$1.65** ea. + .20 Shpng.

Shipping: 5¼" DISKETTES—Add $3.00 per 100 or fraction thereof. OTHER ITEMS: Shipping charges as shown in addition to diskette shipping charges. **Payment:** VISA or MC. COD orders only, add $3.00. **Taxes:** Illinois customers, please add 8%.

Nationwide: 1-800-621-6827
In Illinois: 1-312-944-2788
Minimum Order: $35.00
WE WILL BEAT ANY NATIONALLY ADVERTISED PRICE!
DISK WORLD!
Suite 4806 • 30 East Huron Street • Chicago, Illinois 60611

SYNCOM

Advertisers Index

COMPUTE!'s FREE Reader Information Service

Use these cards to request FREE information about the products advertised in this issue. Clearly print or type your full name and address. Only one card should be used per person. Circle the numbers that correspond to the key number appearing in the advertisers index.

Send in the card and the advertisers will receive your inquiry. Although every effort is made to insure that only advertisers wishing to provide product information have reader service numbers, COMPUTE! cannot be responsible if advertisers do not provide literature to readers.

Please use these cards *only* for subscribing or for requesting product information. Editorial and customer service inquiries should be addressed to: COMPUTE!, P.O. Box 5406, Greensboro, NC 27403. Check the expiration date on the card to insure proper handling.

Use these cards and this address only for COMPUTE!'s Reader Information Service. Do not send with payment in any form.

COMPUTE!

101	102	103	104	105	106	107	108	109	110	111
112	113	114	115	116	117	118	119	120	121	122
123	124	125	126	127	128	129	130	131	132	133
134	135	136	137	138	139	140	141	142	143	144
145	146	147	148	149	150	151	152	153	154	155
156	157	158	159	160	161	162	163	164	165	166
167	168	169	170	171	172	173	174	175	176	177
178	179	180	181	182	183	184	185	186	187	188
189	190	191	192	193	194	195	196	197	198	199
200	201	202	203	204	205	206	207	208	209	210
211	212	213	214	215	216	217	218	219	220	221
222	223	224	225	226	227	228	229	230	231	232
233	234	235	236	237	238	239	240	241	242	243
244	245	246	247	248	249	250	251	252	253	254
255	256	257	258	259	260	261	262	263	264	265
266	267	268	269	270	271	272	273	274	275	276
277	278	279	280	281	282	283	284	285	286	287
288	289	290	291	292	293	294	295	296	297	298
299	300	301	302	303	304	305	306	307	308	309
310	311	312	313	314	315	316	317	318	319	320
321	322	323	324	325	326	327	328	329	330	331
332	333	334	335	336	337	338	339	340	341	342
343	344	345	346	347	348	349	350			

Circle 101 for a one year new U.S. subscription to COMPUTE!: you will be billed for $24.

Please print or type your full name and address. Limit one card per person.

Name ______________________

Address ______________________

City ______________________

State/Province ______________ Zip ________

Country ______________________

Please include zip code. Expiration 7/31/84 CO584

COMPUTE!

101	102	103	104	105	106	107	108	109	110	111
112	113	114	115	116	117	118	119	120	121	122
123	124	125	126	127	128	129	130	131	132	133
134	135	136	137	138	139	140	141	142	143	144
145	146	147	148	149	150	151	152	153	154	155
156	157	158	159	160	161	162	163	164	165	166
167	168	169	170	171	172	173	174	175	176	177
178	179	180	181	182	183	184	185	186	187	188
189	190	191	192	193	194	195	196	197	198	199
200	201	202	203	204	205	206	207	208	209	210
211	212	213	214	215	216	217	218	219	220	221
222	223	224	225	226	227	228	229	230	231	232
233	234	235	236	237	238	239	240	241	242	243
244	245	246	247	248	249	250	251	252	253	254
255	256	257	258	259	260	261	262	263	264	265
266	267	268	269	270	271	272	273	274	275	276
277	278	279	280	281	282	283	284	285	286	287
288	289	290	291	292	293	294	295	296	297	298
299	300	301	302	303	304	305	306	307	308	309
310	311	312	313	314	315	316	317	318	319	320
321	322	323	324	325	326	327	328	329	330	331
332	333	334	335	336	337	338	339	340	341	342
343	344	345	346	347	348	349	350			

Circle 101 for a one year new U.S. subscription to COMPUTE!: you will be billed for $24.

Please print or type your full name and address. Limit one card per person.

Name ______________________

Address ______________________

City ______________________

State/Province ______________ Zip ________

Country ______________________

Please include zip code. Expiration 7/31/84 CO584

COMPUTE!

101	102	103	104	105	106	107	108	109	110	111
112	113	114	115	116	117	118	119	120	121	122
123	124	125	126	127	128	129	130	131	132	133
134	135	136	137	138	139	140	141	142	143	144
145	146	147	148	149	150	151	152	153	154	155
156	157	158	159	160	161	162	163	164	165	166
167	168	169	170	171	172	173	174	175	176	177
178	179	180	181	182	183	184	185	186	187	188
189	190	191	192	193	194	195	196	197	198	199
200	201	202	203	204	205	206	207	208	209	210
211	212	213	214	215	216	217	218	219	220	221
222	223	224	225	226	227	228	229	230	231	232
233	234	235	236	237	238	239	240	241	242	243
244	245	246	247	248	249	250	251	252	253	254
255	256	257	258	259	260	261	262	263	264	265
266	267	268	269	270	271	272	273	274	275	276
277	278	279	280	281	282	283	284	285	286	287
288	289	290	291	292	293	294	295	296	297	298
299	300	301	302	303	304	305	306	307	308	309
310	311	312	313	314	315	316	317	318	319	320
321	322	323	324	325	326	327	328	329	330	331
332	333	334	335	336	337	338	339	340	341	342
343	344	345	346	347	348	349	350			

Circle 101 for a one year new U.S. subscription to COMPUTE!: you will be billed for $24.

Please print or type your full name and address. Limit one card per person.

Name ______________________

Address ______________________

City ______________________

State/Province ______________ Zip ________

Country ______________________

Please include zip code. Expiration 7/31/84 CO584

COMPUTE! Reader Service
P.O. Box 11747
Philadelphia, PA 19101

Place
Postage
Here

COMPUTE! Reader Service
P.O. Box 11747
Philadelphia, PA 19101

Place
Postage
Here

COMPUTE! Reader Service
P.O. Box 11747
Philadelphia, PA 19101

"Commodore-ready", . . . and ready for you NOW!

"Cardcorder" DC/1, Data Cassette Recorder/Player

Introducing the "CARDCORDER", Model DC/1, the Computer Cassette that is "Commodore-ready," designed for storage and retrieval of computer data efficiently, economically; with consistent performance. Yet, this fine CARDCO product is priced lower than any similar product with special quality features.

Includes standard connector which is "Commodore-ready"; LED "save" indicator light which confirms data recording on to the tape; handles up to 120 minutes (60 minutes on each side) of any standard tape including existing pre-recorded commercial as well as personal data tapes intended for use with Commodore Personal Computers; ready to go . . . just plug it in and record efficiently.

CARDCO's "CARDCORDER" COMPUTER CASSETTE is a quality data cassette recorder/player in an attractive polystyrene case, with all the standard cassette functions: record . . . play . . . rewind . . . fast forward . . . stop and eject . . . pause. A solid-state designed product of the finest components with auto-stop.

The "CARDCORDER" DC/1 carries a 90 day warranty to original owners.

All CARDCO products are available at your local dealers.

cardco, inc.

313 Mathewson Wichita, Kansas 67214 (316) 267-6525

"The world's largest manufacturer of Commodore accessories."

Commodore 64 Magic Desk I

Only From Commodore—The Excitement and Simplicity of Magic Desk!

Only Commodore brings you the magic of MAGIC DESK . . . the next generation of "user friendly" software! Imagine using your computer to type, file and edit personal letters and papers *without learning any special commands!* All MAGIC DESK commands are PICTURES. Just move the animated hand to the picture of the feature you want to use (like the TYPEWRITER) and you're ready to go.

The MAGIC DESK Typewriter works just like a real ELECTRIC TYPEWRITER . . . and it's COMPUTERIZED. All the filing is *electronic*. Excellent sound effects and screen animation make typing fun, whether you're typing letters, reports or memos . . . and the built-in filing feature makes MAGIC DESK useful for keeping names and addresses, home inventory lists, insurance information and more.

Your COMMODORE 64, COMMODORE DISK DRIVE and MAGIC DESK are an unbeatable combination. Filing operations are automatically linked to your Commodore disk drive—but you don't have to know any commands—just "file" the pages you type in the file cabinet and your text is automatically saved on diskette. There are 3 file drawers with 10 file folders in each drawer and 10 pages in each folder.

To PRINT a page you've typed, just "point" at the picture of the printer and your pages are automatically printed on your COMMODORE PRINTER or PRINTER/PLOTTER. If you want to erase what you've typed, the WASTEBASKET under the desk lets you "throw away" pages. There's even a DIGITAL CLOCK which helps you keep track of time while you're typing.

Not only is MAGIC DESK easy to use . . . it's hard to make a mistake! Just press the COMMODORE key and one of several "help menus" appears to tell you exactly what to do next. Special messages show you how the various picture commands work and help you when you make a mistake. Help messages also show you how to use the printer, filing cabinet, digital clock and wastebasket.

Another reason why Commodore is number one in the world of microcomputers—Come join us.

commodore COMPUTERS

First In Quality Software