```
JA 17000 GOSUB 19000 'erase curs
         or
QK 17010 PUT (WX1-2,WY1),MSAVE%,
         PSET
AB 17020 LOCATE 1,XP-1:MSG$=" "+
         MTITLE$(MNID,0):GOSUB 1
         3000
IF 17030 RETURN
MA 17040 '
KN 17050 'Cursor ON
PN 18000 IF CURSOR=0 OR TOGGLE=1
          THEN RETURN 'no cursor
         , or cursor already on
HJ 18010 PUT (MX,MY),ARROW%:TOGG
         LE=1:RETURN
MJ 18020 'Cursor OFF
GE 19000 IF CURSOR=0 OR TOGGLE=0
          THEN RETURN 'no cursor
         , or cursor already off
FK 19010 PUT (MX,MY),ARROW%:TOGG
         LE=0:RETURN
MM 19020 '
DP 19030 'Following routine chec
         ks "mouse", returns scr
         een coords MX,MY, and b
         utton status MB (0 if n
         ot pressed, else 1).
BA 19040 'Requires that XRATIO#
         and YRATIO# set to rati
         o of horizontal/vertica
         l resolution divided by
          maximum value of joyst
         ick.  XOFF and YOFF are
          the lowest values retu
         rned by joystick or tab
         let.
BG 19050 'XRES and YRES are the
         width and height of the
          screen in pixels
HN 19060 'If CURSOR flag is set
         to -1, the routine upda
         tes an arrow cursor.
OH 19070 'if FROZEN flag is non-
         zero, disables joystick
         /tablet.
EK 19080 'Be sure to initialize
         ACC=1 and DACC.  ACC is
          an accelerating distan
         ce moved by cursor keys
         , reset to DACC to stop
          acceleration.  If DACC
          is negative, movement
         is constant, with no ac
         celeration.
MJ 19090 'Two flags returned are
          KEYMODE (0 if joystick
         /pad was just used, els
         e -1) and PENUP (0 for
         no contact with pad or
         joystick at far upper-l
         eft corner, else -1).
LH 19100 '
KK 19110 '*** GETXY ****
MP 19120 '
DN 20000 MB=0:PENUP=0
JL 20010 IF NOT FROZEN THEN S0=S
         TICK(0):S1=STICK(1):MB=
         STRIG(1):IF S0<>XOFF OR
          S1<>YOFF THEN NX=INT((
         S0-XOFF)*XRATIO#):NY=IN
         T((S1-YOFF)*YRATIO#):KE
         YMODE=0:ELSE PENUP=-1
JN 20020 MK$=INKEY$:KY=0:IF MK$=
         "" THEN IF  TIMER>=TM!
         THEN ACC=ABS(DACC):TM!=
         TIMER+.1:GOTO 20060 ELS
         E 20060
LM 20025 KY=ASC(MID$(MK$,2)+CHR$
         (0)):MB=MB OR -(KY=82):
         KEYMODE=-1
EE 20030 NX=-(NX+ACC*(KY=75)-ACC
         *(KY=77))*(KY<>71):NY=-
         (NY+ACC*(KY=72)-ACC*(KY
         =80))*(KY<>71)
MA 20040 IF KY=PK THEN ACC=ACC+2
         *(ACC<13)*(DACC>0):PK=K
         Y:ELSE ACC=ABS(DACC):PK
         =KY
DC 20050 KY=ASC(MK$):IF NOT (KY>
         47 AND KY<58) THEN WHER
         E=INSTR(CM$,CHR$(KY+32*
         (KY>96 AND KY<123))):IF
          WHERE THEN MNID=VAL(MI
         D$(CM$,WHERE+1,1)):MNIT
         =VAL(MID$(CM$,WHERE+2,1
         )):IF MFLAGS(MNID,MNIT)
         =0 THEN MNIT=0:MNID=0 E
         LSE GOSUB 21010
HK 20060 IF NX=MX AND NY=MY THEN
          RETURN
OP 20065 XBOUND=XRES-XARROW:YBOU
         ND=YRES-YARROW
EI 20070 NX=-NX*(NX>0 AND NX<=XB
         OUND)-XBOUND*(NX>XBOUND
         )-(NX<1)
FO 20080 NY=-NY*(NY>0 AND NY<=YB
         OUND)-YBOUND*(NY>YBOUND
         )-(NY<1)
PN 20090 GOSUB 19000:MX=NX:MY=NY
         :GOSUB 18000
HP 20100 RETURN
JJ 21010 XP=INT(MX(MNID-1)/8)+2:
         MSG$=" "+MTITLE$(MNID,0
         ):GOSUB 19000
GG 21015 LOCATE 1,XP-1:PRINT MSG
         $
NI 21020 IF SNDFX THEN SOUND 100
         00,.1
FL 21030 LOCATE 1,XP-1:GOSUB 130
         00
IO 21040 RETURN
```

Program 2: REMover

```
CI 10 'REMover-deletes REMS from
       a program
AF 20 CLS:PRINT"REMover: Deletes
       REMs"
CL 30 PRINT:PRINT"Enter name of
       ASCII program to edit":LIN
       E INPUT ": ";ASCFILE$
MK 40 PRINT:PRINT"Enter name of
       ASCII program to create":L
       INE INPUT ": ";CREATE$
OH 50 OPEN ASCFILE$ FOR INPUT AS
       #1
NO 60 OPEN CREATE$ FOR OUTPUT AS
       #2
PL 70 PRINT:PRINT"Level I change
       s REM statements to '"
FK 80 PRINT:PRINT"Level II delet
       es all REM lines and remov
       es REMs from end of line."
FD 90 PRINT:LINE INPUT "Level I
       (1) or II (2): ";LV$:LV=VA
       L(LV$):IF LV<1 OR LV>2 THE
       N 90
HK 100 WHILE NOT EOF(1)
IB 110  LINE INPUT#1,L$:PRINT"*"
       ;
DL 120  RP=INSTR(L$,"REM")
IL 130  IF RP=0 THEN RP=INSTR(L$
       ,"'")
LC 140  IF RP THEN L$=LEFT$(L$,R
       P-1):IF LV=1 THEN L$=L$+"
       '"
AG 150  IF LV=2 AND RP=INSTR(L$,
       " ")+1 THEN 170
BC 160  PRINT#2,L$
EC 170 WEND
HP 180 CLOSE#1
IM 190 CLOSE#2
QP 200 PRINT:PRINT:PRINT"Finishe
       d."
LN 210 END
```


64 UN-CRUNCHER

Larry Dinwiddie

This convenient Commodore 64 utility "uncrunches" crowded BASIC program lines into separate, easily readable and edited lines. The utility works with either disk or tape, and although it is written in machine language, no machine language expertise is required to use it. It runs on any Commodore 64 or 128 (in 64 mode).

One common programming technique in Commodore BASIC is to "crunch" programs into compact form by combining multiple statements on a single program line. As most programmers soon learn, crunching conserves memory and helps a program run faster. In addition, programs listed in magazines and books are usually crunched to save space.

However, crunching also makes the program more difficult to read and modify. Often, modifying a crunched program involves breaking up a long line into two or more shorter lines. This can be tedious, and it increases the risk of errors.

"64 Uncruncher" automatically uncrunches an entire program for

you, making each BASIC statement a separate program line. The resulting program is much easier to modify than the original. And because each statement is a separate line, it is simpler to follow the program's logic as well.

Figures 1 and 2 illustrate a simple BASIC program before and after uncrunching. Both programs are listed with a width of 40 columns so they appear just as they would on your screen. Notice how much easier it is to read the uncrunched version and decipher its logic.

Figure 1: Crunched Program

```
10 POKE 53281,15:POKE 53280,15:POKE 646,
0:PRINT CHR$(147):FOR J=1 TO 10
20 GOSUB 50:IF INT(K/2)=K/2 THEN PRINT "
, AN EVEN NUMBER":GOTO 40
30 PRINT ", AN ODD NUMBER"
40 NEXT J:PRINT:PRINT "FINISHED":END
50 READ K:PRINT "K =" K;:RETURN:DATA 123
,456,789,987,654,321,123,456,789,111
```

Figure 2: Uncrunched Program

```
100 POKE 53281,15
110 POKE 53280,15
120 POKE 646,0
130 PRINT CHR$(147)
140 FOR J=1 TO 10
150 GOSUB 220
160 IF INT(K/2)=K/2 THEN PRINT ", AN EVE
N NUMBER":GOTO 180
170 PRINT ", AN ODD NUMBER"
180 NEXT J
190 PRINT
200 PRINT "FINISHED"
210 END
220 READ K
230 PRINT "K =" K;
240 RETURN
250 DATA 123,456,789,987,654,321,123,456
,789,111
```

Using Uncruncher

Since Uncruncher is written in machine language, you must type it in using the "MLX" machine language entry program listed elsewhere in this issue. Be sure you read and understand the instructions for using MLX before you begin entering the data for Uncruncher.

When you run the MLX program, it asks you for a starting address and an ending address. Here are the addresses for Uncruncher:

Starting address: C000
Ending address: C60F

After you've typed in and saved all of the Uncruncher data, you can test it on any Commodore 64 BASIC program. Follow these steps:

1. Load Uncruncher into memory by typing LOAD "*filename*",8,1 for disk or LOAD "*filename*",1,1 for tape. Substitute your own filename, of course.
2. Type NEW and press RETURN.
3. Load (but do not run) the BASIC program you want to uncrunch.
4. To start Uncruncher, type SYS 49152 and press RETURN. The screen clears and Uncruncher displays messages informing you of its progress. It takes three passes through the BASIC code to uncrunch the program. When the READY prompt returns, the uncrunching is complete.

You may pause Uncruncher at any time by holding down the f1 function key. Do not interrupt Uncruncher by pressing RUN/STOP–RESTORE; if you do, the program may be left in a garbled, unusable form. No real harm is done, but you'll need to reload the program and restart Uncruncher.

Uniform Line Numbers

Uncruncher begins numbering the new program at line 100 using line increments of 10 (110, 120, and so on). Each BASIC statement is a separate line, except for lines containing IF-THEN statements. Because the THEN portion of such a statement must be on the same line as IF, IF-THEN lines are left unchanged except for renumbering. For any BASIC statement that references a line number (such as GOTO, GOSUB, IF-THEN, RUN, and LIST), the line reference is also renumbered.

During the third pass, the program prints the line numbers it is replacing. If Uncruncher finds a statement that refers to a nonexistent line number, it prints this error message:

UNREFERENCED BRANCH IN NEW LINE # xxxxx

When you see this message, *xxxxx* is replaced by the new line number where the nonexistent reference is located. To mark where the error occurred, Uncruncher replaces the meaningless line number with 63999, the highest legal line number. An unreferenced branch error indicates a logic error in the original program, so you should reload the original, correct the error, and then repeat the uncrunching process.

If uncrunching generates a large number of unreferenced line number errors, you may find it useful to divert Uncruncher's output to a printer. To do this, make sure the printer is turned on, then enter this statement in direct mode (without a line number):

OPEN 4,4:CMD 4:SYS 49152

Now everything that would have been printed on the screen is sent to the printer instead. When Uncruncher is finished, type this statement and press RETURN:

PRINT#4:CLOSE 4

DATA statements are uncrunched in a special way. After uncrunching, each DATA line contains approximately 60 characters per line, including the line number and the keyword DATA. However, no individual DATA item will be split across two lines.

After the program has been uncrunched, you may list it, resave it, or modify it as usual. If you have a crunch utility, you may wish to recrunch the program after making modifications.

Because the additional line numbers take up more memory, the uncrunched program is significantly larger than the original, leaving less memory for BASIC variables and arrays. In most cases this should not cause a problem other than slowing program execution somewhat. However, a very large BASIC program or one that requires a great deal of variable space may not run correctly in uncrunched form.

Similar problems may arise if the original program POKEs sprite shapes, custom characters, or other data into a reserved area within BASIC memory. If the uncrunched program text expands into the reserved area, the POKEs may destroy part of the program text. To be on the safe side, you may want to save the uncrunched program immediately before you try to run it.

64 Uncruncher

Please refer to the "MLX" article in this issue before entering the following listing.

```
C008:8D 8D C5 A5 74 8D 8E C5 83
C010:A5 75 8D 8F C5 A9 4C 85 5F
C018:73 A9 15 85 74 A9 C5 85 14
C020:75 20 3A C5 A9 FF 91 A3 1C
C028:C8 91 A3 A9 9E A0 C5 20 A4
C030:1E AB A9 C0 A0 C5 20 1E 67
C038:AB A9 EF 8D 28 03 A9 00 71
```

```
C040:8D 90 C5 20 15 C5 AA 20 9C
C048:15 C5 E0 00 D0 07 C9 00 18
C050:D0 03 4C 09 C1 20 15 C5 93
C058:20 15 C5 EE 90 C5 20 15 C7
C060:C5 C9 00 F0 D9 C9 A7 F0 7C
C068:38 C9 CB F0 2A C9 8A F0 7F
C070:30 C9 9B F0 2C C9 A4 F0 C1
C078:23 A2 00 8E 91 C5 C9 8D E1
C080:F0 2C C9 89 F0 28 C9 22 35
C088:D0 D4 20 15 C5 C9 00 F0 43
C090:AD C9 22 D0 F5 F0 C7 EE 9E
C098:91 C5 D0 C2 AD 91 C5 F0 CA
C0A0:BD 20 15 C5 C9 30 90 B9 F1
C0A8:C9 3A B0 B5 90 03 20 15 F4
C0B0:C5 20 6B A9 85 02 A9 0B B7
C0B8:85 A3 A9 C6 85 A4 A0 00 87
C0C0:B1 A3 AA C8 B1 A3 E0 FF C3
C0C8:D0 04 C9 FF F0 15 E4 14 A6
C0D0:D0 04 C5 15 F0 21 18 A5 A7
C0D8:A3 69 04 85 A3 90 DF E6 65
C0E0:A4 D0 DB 88 A5 14 91 A3 31
C0E8:A5 15 C8 91 A3 A9 FF A2 1B
C0F0:04 C8 91 A3 CA D0 FA A5 48
C0F8:02 F0 0B C9 2C F0 AF C9 04
C100:AB F0 AB 4C 61 C0 4C 3E B4
C108:C0 A9 C9 A0 C5 20 1E AB 30
C110:20 3A C5 A2 00 8E 90 C5 36
C118:A9 0B 85 A3 A9 C6 85 A4 36
C120:20 15 C5 AA 20 15 C5 E0 1E
C128:00 D0 07 C9 00 D0 03 4C F2
C130:FB C1 20 15 C5 85 FB 20 D3
C138:15 C5 85 FC A2 01 8E 90 FE
C140:C5 A0 00 B1 A3 AA C8 B1 F4
C148:A3 E0 FF D0 04 C9 FF F0 1B
C150:1F E4 FB D0 04 C5 FC F0 4B
C158:0D 18 A5 A3 69 04 85 A3 61
C160:90 DF E6 A4 D0 DB C8 A5 78
C168:A5 91 A3 C8 A5 A6 91 A3 B2
C170:20 15 C5 C9 00 F0 66 C9 F8
C178:3A F0 55 C9 22 F0 0C C9 53
C180:83 F0 15 C9 8B D0 E9 E6 9B
C188:A7 D0 E5 20 15 C5 C9 00 26
C190:F0 4B C9 22 D0 F5 F0 D8 D3
C198:A0 00 8C 90 C5 20 15 C5 A5
C1A0:C8 C9 22 F0 1D C9 00 F0 4F
C1A8:34 C9 3A F0 23 C9 2C D0 78
C1B0:EC C0 32 90 E8 18 A5 A5 C2
C1B8:69 0A 85 A5 90 DA E6 A6 E2
C1C0:D0 D6 20 15 C5 C8 C9 00 9C
C1C8:F0 13 C9 22 D0 F4 F0 CD EE
C1D0:A2 01 8E 90 C5 85 02 A5 AE
C1D8:A7 D0 95 F0 09 A2 01 8E 89
C1E0:90 C5 85 02 85 A7 18 A5 8F
C1E8:A5 69 0A 85 A5 90 02 E6 8D
C1F0:A6 A5 02 D0 03 4C 13 C1 AF
C1F8:4C 70 C1 A9 D1 A0 C5 20 4E
C200:1E AB 20 3A C5 A9 00 85 81
C208:A7 8D 90 C5 20 15 C5 AA BE
C210:20 15 C5 E0 00 D0 07 C9 CC
C218:00 D0 03 4C 67 C5 AD 24 C8
C220:C5 85 FB AD 25 C5 85 FC 8C
C228:A0 01 A5 A5 91 FB C8 A5 01
C230:A6 91 FB 18 A5 A5 69 0A 0F
C238:85 A5 90 02 E6 A6 20 15 43
C240:C5 8D 94 C5 20 15 C5 8D 69
C248:95 C5 EE 90 C5 20 15 C5 8F
C250:C9 00 F0 B1 C9 3A F0 4D 5A
C258:C9 22 F0 39 C9 CB F0 29 85
C260:C9 A4 F0 28 A2 00 8E 91 58
C268:C5 C9 83 F0 49 C9 8A F0 3A
C270:1E C9 9B F0 1A C9 A7 F0 32
C278:16 C9 8D F0 15 C9 89 F0 10
C280:11 C9 8B D0 C8 E6 A7 D0 81
C288:C4 4C ED C2 4C F3 C2 4C 71
C290:FB C2 4C 09 C3 20 15 C5 6D
C298:C9 00 F0 06 C9 22 D0 F5 EF
C2A0:F0 AB 4C 05 C2 A2 01 8E 94
C2A8:90 C5 A5 A7 D0 9F A9 04 73
C2B0:20 F9 C4 4C 05 C2 A0 00 96
C2B8:8C 90 C5 20 15 C5 C8 C9 7E
C2C0:00 F0 DF C9 3A F0 DE C9 38
C2C8:22 F0 14 C9 2C D0 EC C0 F9
C2D0:32 90 E8 A9 05 20 F9 C4 AC
C2D8:A9 83 91 A8 4C 05 C2 20 EC
C2E0:15 C5 C8 C9 00 F0 BB C9 1D
C2E8:22 D0 F4 F0 CE EE 91 C5 7C
C2F0:4C 4D C2 AE 91 C5 D0 03 7B
C2F8:4C 4D C2 20 15 C5 C9 30 D5
C300:90 04 C9 3A 90 06 4C 50 33
C308:C2 20 15 C5 AE 24 C5 86 10
C310:A8 AE 25 C5 86 A9 20 6B 1F
C318:A9 85 02 A9 0B 85 A3 A9 10
C320:C6 85 A4 A0 00 B1 A3 AA C3
C328:C8 B1 A3 E4 14 D0 04 C5 F4
C330:15 F0 0D 18 A5 A3 69 04 34
C338:85 A3 90 E7 E6 A4 D0 E3 4B
C340:C8 B1 A3 AA 85 63 C8 B1 B4
C348:A3 85 62 E0 FF D0 21 C9 AC
C350:FF D0 1D A9 DA A0 C5 20 4F
C358:1E AB AE 94 C5 AD 95 C5 CE
C360:20 CD BD A9 0D 20 D2 FF 4C
C368:A9 F9 85 62 A9 FF 85 63 D5
C370:A2 90 38 20 49 BC 20 DF D3
C378:BD 85 FB 84 FC A0 00 B1 24
C380:FB 99 98 C5 F0 03 C8 D0 D1
C388:F6 A9 FE A0 C5 20 1E AB 76
C390:EE 20 D0 A9 98 A0 C5 20 3F
C398:1E AB 38 AD 24 C5 E5 A8 A8
C3A0:85 FD A0 00 B9 98 C5 C9 04
C3A8:00 F0 03 C8 D0 F6 8C 97 6C
C3B0:C5 C4 FD F0 25 B0 0D A5 C6
C3B8:FD 84 FD 38 E5 FD 20 F8 04
C3C0:C3 4C DA C3 38 98 E5 FD C2
C3C8:20 6C C4 18 AD 24 C5 6D 8C
C3D0:96 C5 8D 24 C5 90 03 EE 6E
C3D8:25 C5 A0 00 AE 97 C5 B9 91
C3E0:98 C5 91 A8 C8 CA D0 F7 ED
C3E8:A5 02 C9 2C F0 07 C9 AB A2
C3F0:F0 03 4C 50 C2 4C 09 C3 5D
C3F8:A6 A8 86 FD A6 A9 86 FE 96
C400:8D 96 C5 18 65 FD 85 FD 5C
C408:90 02 E6 FE A5 2D 85 AA BE
C410:A5 2E 85 AB 38 A5 AA E5 F6
C418:FD 85 AC B0 02 C6 AB 38 5D
C420:A5 AB E5 FE AA A0 00 B1 9D
C428:FD 91 A8 C8 C4 AC D0 F7 29
C430:A0 00 E0 00 F0 0E E6 FE B2
C438:E6 A9 B1 FD 91 A8 C8 D0 47
C440:F9 CA D0 F2 38 A5 2D ED 63
C448:96 C5 85 2D B0 02 C6 2E 5B
C450:38 A5 2F ED 96 C5 85 2F 2A
C458:B0 02 C6 30 38 AD 24 C5 1D
C460:ED 96 C5 8D 24 C5 B0 03 B4
C468:CE 25 C5 60 8D 96 C5 A6 5A
C470:2D 86 A3 86 AA A6 2E 86 E1
C478:A4 86 AB A6 A8 8E 92 C5 40
C480:A6 A9 8E 93 C5 38 A5 AA D7
C488:E5 A8 85 FD B0 02 E6 A9 C4
C490:38 A5 AB E5 A9 85 FE 38 0D
C498:A5 A3 E5 FD 85 A3 85 AA EA
C4A0:B0 04 C6 A4 C6 AB AD 96 7D
C4A8:C5 A4 FD 18 65 A3 85 A3 E7
C4B0:90 02 E6 A4 B1 AA 91 A3 29
C4B8:88 D0 F9 B1 AA 91 A3 A6 9E
C4C0:FE F0 13 C6 AB C6 A4 88 1F
C4C8:B1 AA 91 A3 88 D0 F9 B1 6F
C4D0:AA 91 A3 CA D0 ED AD 96 65
C4D8:C5 18 65 2D 85 2D 90 02 CE
C4E0:E6 2E AD 96 C5 18 65 2F 11
C4E8:85 2F 90 02 E6 30 AE 92 1B
C4F0:C5 86 A8 AE 93 C5 86 A9 69
C4F8:60 AE 24 C5 86 A8 AE 25 98
C500:C5 86 A9 20 6C C4 A9 00 11
C508:A8 AE 96 C5 91 A8 C8 A9 2D
C510:FF CA D0 F8 60 A5 CB C9 F2
C518:40 D0 FA EE 24 C5 D0 03 23
C520:EE 25 C5 AD 01 08 AE 90 16
C528:C5 F0 0E C9 3A B0 0A C9 A3
C530:20 F0 E2 38 E9 30 38 E9 52
C538:D0 60 A9 0B 85 A3 A9 C6 FE
C540:85 A4 A9 64 85 A5 A9 00 49
C548:85 A6 85 A7 A5 2B 8D 24 84
C550:C5 A5 2C 8D 25 C5 38 AD E4
C558:24 C5 E9 01 8D 24 C5 B0 ED
C560:03 CE 25 C5 A0 00 60 A5 8D
C568:2F 85 31 A5 30 85 32 A9 13
C570:ED 8D 28 03 20 33 A5 AD 52
C578:8D C5 85 73 AD 8E C5 85 DC
C580:74 AD 8F C5 85 75 AD 0A 67
C588:C6 8D 20 D0 60 00 00 00 EE
C590:00 00 00 00 00 00 00 00 1C
C598:20 20 20 20 20 20 93 11 FB
C5A0:0E 2A 2A 2A 20 C3 CF CD 23
C5A8:D0 D5 D4 C5 21 27 53 20 75
C5B0:D5 4E 43 52 55 4E 43 48 FA
C5B8:45 52 20 2A 2A 2A 0D 00 36
C5C0:11 D0 41 53 53 20 31 0D F0
C5C8:00 D0 41 53 53 20 32 0D 72
C5D0:00 D0 41 53 53 20 33 0D 7C
C5D8:11 00 0D D5 4E 52 45 46 78
C5E0:45 52 45 4E 43 45 44 20 09
C5E8:42 52 41 4E 43 48 20 49 FB
C5F0:4E 20 4E 45 57 20 4C 49 E6
C5F8:4E 45 20 23 20 00 0D 91 DF
C600:20 20 20 20 20 20 20 0D 7A
C608:91 00 00 00 00 00 00 00 5E
```

©

Do-It-Yourself Movies On An Apple

Recently my ten-year-old daughter Catie asked if I'd like to help her with her school science project. Oh, boy! I thought. Here's a chance to show her how she could take advantage of a computer!

I was almost afraid to suggest that we use a computer, however. She's not quite as fanatical about the machines as I am, and she gets tired hearing how every family activity can somehow be tied to computers. So I didn't mention the word "computer" at all. Instead, I said, "Hey, Catie, how'd you like to make a movie for your science project?" This idea delighted her, so off we went.

The first step was to choose a subject. Catie chose black holes. "Okay," I said. "You have to do two things: Draw a bunch of squares like you see in the funnies in the newspaper, and draw pictures inside the squares of the black hole—how it's born, how it grows, and so on. Next, sit down and write a script for the movie. Match what you say in the script with the pictures in the squares."

Frame By Frame

Catie raced off and drew the pictures and wrote her script. When she came back, I was sitting in front of our Apple IIc. "Daddy," she said, "why are you sitting at the computer? We're supposed to be working on my movie."

"Aha!" I said. "The computer is going to help us make that movie." I introduced her to a program called *Fantavision*. *Fantavision* looks like a normal drawing program—it has a drawing window surrounded by lots of tools and menu options around the border. I showed Catie how she could draw things freehand or with rubberband lines, squares, circles, and so on. She could fill the objects with color, stretch them, rotate them, squish them, cut them, and paste them anywhere in the window.

But this was just the beginning. When she was done creating a picture of a happy face, I showed her how she had just created one frame in a cartoon. She could use the mouse to scroll the screen and begin creating the next frame. Catie then drew a face of a kitty cat.

"And now you've got a little movie," I told her. I pointed the mouse to a menu box labeled GO, and we watched a short cartoon of the happy face changing into the face of the cat.

A whole movie from only two frames? The secret is a complex technique that animators call *tweening* (derived from *between*). *Fantavision* automatically constructed dozens of new frames from Catie's first frame and second frame, then inserted them between her frames to smooth out the transition. These new frames, called tweeners, made the happy face in her first frame change gradually into the kitty's face of her second frame.

From Giant To Dwarf

With very little help from me, Catie sat down at the computer and learned how to use *Fantavision* in about half an hour. She copied her hand-drawn frames from her notebook onto the screen. The first frame was a picture of a normal, yellow-looking sun surrounded by stars in outer space. The next frame was of the same sun, now billions of years older, swollen to become a red giant star. The third frame showed the star shrunken into a tiny white dwarf star.

The white dwarf continued to shrink until it became a black hole. Catie drew a picture of the black hole that was straight out of Walt Disney—with swirling white clouds of cosmic gas spiraling around a dark center. Next, using the COPY, MOVE, and ROTATE commands, she drew successive frames of the black hole rotating and gobbling up stars.

Then Catie designed a title, which turned out to be one of the most spectacular parts of the movie. By using the ZOOM command, Catie was able to create several successive frames with the words "The Black Hole" growing larger and larger. And when the movie starts, the letters in the title break up into pieces which come together to form the stars and the sun. This looks like an amazing special effect, but it was completely unintended; it was just a by-product of *Fantavision*'s tweening capabilities.

Finally, Catie and I set up the Apple in the room with the stereo cassette player. We bought a copy of the soundtrack to the movie *Jaws* and aimed a video camera at the computer screen while the hungry shark music was playing in the background. Catie read her script as the movie progressed—from the opening title to the birth and growth of the black hole. It took us several tries to synchronize the music, Catie's narration, and the *Fantavision* movie, but it was well worth it. The next day, Catie took her project to school and won a blue ribbon for her efforts.

I was really proud of her, but my biggest thrill came when she ran up to me after the judging and said, "You know, Daddy, sometimes I'm glad that our family has a computer."

(*Fantavision* is available for $49.95 for all Apple II-series computers with at least 64K of RAM. For more information, contact Brøderbund Software, 17 Paul Drive, San Rafael, CA 94903.) ©

Speak Softly And Carry A Big RAM

Sooner or later it was inevitable that the grandiose and bizarre claims of some computer scientists would result in a critical response from someone with a radically different point of view. I just read such a response by Theodore Roszak, a history professor at California State University, Hayward. Roszak's book, *The Cult of Information* (Pantheon Books, 1986), lambastes an entire field for the excesses of a few. As a result, he is guilty of the same error as the people he criticizes—members of the artificial intelligensia who claim that computers accurately mimic human behavior and who feel we would be better off learning to think like machines.

I share his distaste for the extravagant and largely unsupported claims made by those who feel that silicon consciousness is our evolutionary destiny. What distresses me is that Roszak expresses the belief that the general acceptance of computers into our homes, schools, and workplaces is somehow damaging to our identity as human beings.

He is confusing the tool with the result and forgetting that technology is inherently neutral. Computers can be (and have been) used in inappropriate ways, as have fountain pens. How easy it is to use the wild exhortations of our field's fringe fanatics to damn an entire technology—one that most of us understand and use without feeling any loss of humanity.

Towards Holistic Thought

Rather than diminishing our human qualities, I think computers allow us to integrate our thinking—to become holistic learners who see knowledge as something more than a collection of facts stored in separately labeled boxes, each with its own content, and with little or no connection between them. The specialization of knowledge into fields was a result of an information explosion that made it impossible for one person to achieve mastery of all subjects. While this division will remain important for many experts in these fields, there is increasing evidence that it has negative consequences.

To take one example, many particle physicists are finding that advances in their field are aided by a study of Taoist philosophy. My own hobby of computational linguistics is populated by linguists, philosophers, and computer scientists, each willing to learn from the others. True knowledge is interdisciplinary. As soon as one draws a box around a topic to clarify the object of study, one risks excluding information or viewpoints that can end up being quite important.

I sometimes put on a multimedia event called *The Magic Universe of Recursion* in which I show the appearance of this mathematical concept in computer programming, music, art, literature, philosophy, and religion. My point is not to say that each of these topics is mathematical—mostly they are not. Rather, it is my purpose to show that recursion is an idea that breaks across traditional barriers of knowledge.

I am convinced that there are hundreds of general concepts that transcend the fields in which they were first used, if only we would look for them. Fortunately, one tool to aid in our search is readily at hand—the personal computer.

Enter The Computer

As a tool that lets us manipulate information and construct metaphorical worlds of our own design, the computer can help us chart a path across the boundaries of numerous disciplines in our quest for holistic education.

Most fields of endeavor are so complex and demanding that one has little time to search beyond the walls of one subject for ideas of value from another area. As computer technology becomes easier to use, the tedious aspects of at least some parts of many fields will be relegated to machinery, thus freeing people to stand back and take in a larger view of the subject. It is hard to take a reading on the stars while you are rowing the boat.

This is why I first became interested in Logo. I saw, in Logo's turtle graphics, a tool that would let me explore the mathematics of naturally occurring patterns. I have spent years exploring everything from cracks in drying mud to the delicate patterns in ferns. The ease with which I could generate, test, and evaluate hypotheses with the aid of the computer allowed me to ask questions I would not have dared to ask otherwise.

My point is that the really exciting uses of computers are likely to come from the interdisciplinary holistic thinkers—people who sense the unity behind the major ideas of our time and place. These people tend not to be technologists, because the intense study needed to master technology leaves (we are told) little room for anything else. The people I have in mind are those whose interests span many fields—physics and poetry, art and archaeology—people who probably have degrees in the "liberal arts."

In order for such people to use computer technology effectively, computers must have speed, lots of memory, excellent software, and a transparent user interface. Computers like the Macintosh and Amiga are stepping stones in the right direction. Software for these computers is being designed to rise to the level of the way people work rather than dragging the user down to the machine's level. ©

The Beginners Page

Tom R. Halfhill, Editor

Advanced String Features

To wrap up our long-running series on character strings in BASIC, let's take a look at some advanced string features which are finding their way into the latest and most sophisticated versions of the BASIC language. Although these features may not be found in the BASIC you're working with now, you'll probably encounter them sometime in the future.

If you want to keep up with these trends, pay attention to any new version of BASIC released by Microsoft, Inc. Microsoft certainly isn't the only company in the BASIC business, but it definitely *is* the market leader. Versions of Microsoft BASIC are either standard equipment or available as an option for almost every microcomputer ever made. When an advanced feature is introduced in a new version of Microsoft BASIC, it tends to cross over into the next version which is released, even if the next version is for a completely different computer.

For instance, the latest Microsoft BASIC to appear is Amiga BASIC. It shares numerous features with its nearest predecessor, Microsoft BASIC for the Macintosh. These two dialects are so much alike that some example programs in the Macintosh BASIC manual—even those with graphics—will run unchanged on the Amiga.

Super Strings

One new trend in Microsoft BASIC is to remove the 255-character limit on strings. Macintosh BASIC and Amiga BASIC both let you define strings up to 32,767 characters long.

So what? you might say. Who needs to display a message that's thousands of characters long? It probably won't fit on the screen, anyway.

But strings are good for lots of things besides displaying messages, of course, especially if they aren't limited to 255 characters. Programmers on the Atari 400/800/XL/XE computers know this well, because Atari BASIC has allowed super strings since 1979.

For instance, suppose you want to write a simple terminal program for downloading public domain software from information services and bulletin boards. Unless you're handy with a memory map, you might have trouble finding a large area of free memory in your computer to temporarily hold the downloaded data before storing it on disk. With super strings, it's no problem. Simply download everything into a single string, perhaps called BUFFER$. Since BASIC reserves and protects memory for the string, you don't have to worry about memory conflicts.

Best of all, the new Microsoft BASICs don't force you to give up anything in return for super strings—unlike Atari BASIC, which doesn't allow string arrays as a tradeoff for this feature.

Search And Replace

Another powerful feature of late-model BASICs is the INSTR function (pronounced *in-string*). INSTR searches through a longer string in search of any shorter string you specify. If INSTR finds the shorter string (*substring*), it returns a number indicating the substring's starting character position within the longer string. Example:

```
10 MAIN$="This is the longer string."
20 X=INSTR(MAIN$,"the")
```

When you run this program, INSTR returns the value 9 in the variable X, because the substring *the* begins at the ninth character position within MAIN$. Of course, you could also use a string variable for the substring parameter in the INSTR function. If INSTR can't find the substring, it returns a 0.

By adding another parameter, INSTR can be made to begin its search at any point within MAIN$. For example, X=INSTR(5,MAIN$,SUB$) would begin searching for SUB$ at the fifth character position of MAIN$. The INSTR function makes it a snap to write filing programs with rapid search-and-retrieve features, because it works at nearly machine language speed.

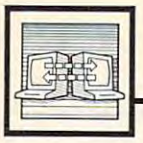
Some recent BASICs (including Macintosh BASIC, Amiga BASIC, and BASIC 7.0 on the Commodore 128) allow the use of MID$ as a statement as well as a function. You'll recall from the April column that the MID$ function lets you copy a substring from within a larger string. When used as a statement, MID$ lets you *replace* a specified substring with another string. And the replacement string isn't limited to the length of the substring it's replacing. When coupled with INSTR, the MID$ statement makes it easy to add a search-and-replace feature to a filing program.

Finally, another useful string command found in newer BASICs is UCASE$. This converts any string of lowercase characters to uppercase. Example:

```
PRINT UCASE$("capitalized")
```

results in CAPITALIZED. A logical application for the UCASE$ command is to make an INSTR search routine insensitive to case. For instance, the statement X=INSTR(UCASE$(MAIN$),UCASE$(SUB$)) will make certain that INSTR will find any matching SUB$ within MAIN$, even if some of the characters are mixed uppercase and lowercase.

Watch for more features like these to keep appearing in new versions of BASIC. Although it's over 20 years old, BASIC is only now experiencing its greatest growth spurt as programmers continue demanding more and more power from this popular language. ©



Telecomputing Today

Arlan R. Levitan

This Fido's No Dog

In June 1984, Tom Jennings of San Francisco and John Madill of Baltimore began developing and testing an MS-DOS-based electronic bulletin board system (BBS) called Fido. Although Fido sported the usual file upload and download facilities, its electronic mail system was far from typical. Fido systems were not designed to exist as separate, isolated entities like most BBSs. Instead, Jennings and Madill set out to create a BBS that could network with others of its own kind. Rather than requiring users and system operators to call each other's BBSs to leave messages, Fido would routinely store and forward messages to other Fidos via modem in the dead of the night, when long-distance phone rates are lowest.

By August 1984 there were almost 30 Fido systems (commonly referred to as *nodes* by Fido fans). Since then, Fido has grown faster and bigger than a Saint Bernard. Today, more than 100,000 users communicate over FidoNet, which consists of more than 1,000 Fido systems spread across the U.S., Europe, and Australia. Using FidoNet, these telecomputing enthusiasts can communicate with each other overnight. And in addition to the public FidoNet, internal Fido systems are being widely used by private industry and government bureaus.

The sheer magnitude of FidoNet easily qualifies it as the largest publicly owned and operated telecomputing network in the world. Other attempts at nationwide networking via BBS have collapsed under their own administrative weight. But the organizational talents of Fido's creators and a dedicated inner core of Fido system coordinators and directors have been put to good use. Careful planning and more sweat than expended in a dozen NBA playoffs have kept the Fido network functioning smoothly.

Global Party Line

If you live in a metropolitan area, your local Fido is likely to be a member of a group of Fido systems located relatively close to each other. Each group is considered to be a *local network.* One system within the group is designated as a *network host.* The system operator of the network host is charged with maintaining a list of the nodes in the local net.

How does Fido work? During the day, Fido users can leave messages for both local and remote users. At about 4:30 a.m., the nodes within the local network begin dialing their network host to transfer messages intended for remote Fido systems. Once all of the outgoing messages from the local net have been collected, the network host compresses them to shorten transmission time, then starts calling other network hosts to send the messages. From 5:00 to 5:30 a.m., the network hosts dial up their local nodes to deliver incoming messages. Heavily used local nets often have two network hosts, one each for outgoing and incoming traffic.

Fidos that are too isolated to be a member of a local network are called *independents* and are permitted to forward and receive mail directly to and from network hosts and other independents. Regional Fido coordinators are responsible for keeping track of independents and encouraging them to join existing nets or forming new ones.

At this writing, the U.S. is divided into 12 Fido regions. Europe has six regions; Australia, two. There are 82 network hosts worldwide. Each host has an average local net of about 13 systems. It's interesting to note that the network hosts in Europe are equipped with two different types of modems. To handle local traffic, they use modems adhering to the CCITT (Consultative Committee on International Telephony & Telegraphy) standard, which employs different frequencies than our domestic units; U.S.-type modems handle transfers to and from Fidos in North America.

Managing The FidoNet

What's truly amazing is that the cost of operating FidoNet is very low when spread out over the entire user base. There is no centralized billing. The local nodes are at liberty to recoup the long-distance charges incurred by their network host however they see fit, either footing the bill themselves or by charging a small yearly membership fee to their local users.

The logistics of keeping things straight within FidoNet could turn into a never-ending "Who's on first?" dilemma if everyone didn't have a constantly updated "phone book," or node list, for all of the systems. Network host operators and regional coordinators are responsible for notifying the national Fido coordinators of any changes in their networks. The national coordinators, in turn, forward a compiled list of changes to Ken Kaplan, executive director of the International Fido Association. A list of FidoNet changes is automatically transmitted to the network hosts every weekend from Kaplan's Fido.

There's also an excellent weekly FidoNet newsletter, managed by FidoFiend Tom Henderson, that's both compiled and distributed via FidoNet. For more information, write to the International FidoNet Association, P.O. Box 41143, St. Louis, MO 63141. ©

INSIGHT: Atari

Bill Wilkinson

Many people have asked me to discuss the use of the DOS 2.5 RAM disk with Atari computers other than the 130XE. Most are interested either in one of the 800XL memory upgrade kits now available or in simply using the extra 16K memory of an XL as a very small RAM disk. Since I've seen the subject treated incorrectly in several user group newsletters, I decided that some mildly technical discussion here would not be amiss.

Many months ago, in one of my columns, I described the memory map of an Atari XL computer. This time, let's see how a 130XE is a fairly simple expansion of the XL models.

An Atari 130XE has 126K—*not* 128K—of Random Access Memory. (Keep in mind that one kilobyte equals 1,024 bytes.) The first 62K is used and accessed exactly the same way as the 62K in the 1200XL and 800XL (that *62K* is not a typo, either—more on this later). Now, a 6502 microprocessor can address a total of only 64K of contiguous memory, because the address counter goes from 0 to 65535 (64 * 1024). In the hexadecimal (base 16) numbering system used by computers, those addresses are expressed as $0000 (0) to $FFFF (65535). When the address counter passes $FFFF, it rolls back to $0000 again. This is kind of like a car speedometer which only goes to 99,999.9 miles; another tenth of a mile and you have a new car again.

So, how does the 130XE access its extra 64K of memory? By a technique known as *bank selection.*

Cashing In At The Memory Bank

The extra 64K in the 130XE is divided into four separate 16K banks. The 6502 can access only one of these banks at a time. But wait, you say, if the main memory uses up the full addressable range of the 6502, where do these extra banks fit in?

The answer: 16K of the main memory (that is, of the regular 64K) is disabled. Effectively, then, a 130XE has *five* banks of RAM, each consisting of 16K, plus another 46K (not a typo) which is *not* bank selected.

Now comes the important part: Just where, within the 64K address space of the 6502, are these five banks addressed? As Appendix H of the 130XE owner's manual states, the selectable 16K bank falls between locations 16384 (hex $4000) and 32767 ($7FFF). This is the second quarter of the 6502's 64K memory space. Why was this 16K area chosen instead of some other area? Because the first quarter of memory includes zero page, and bank-selecting zero page is a tricky proposition in a computer which is handling interrupts. The other two quarters of memory share space with cartridges and/or the operating system ROM, which would make programming more complicated. Thus, the second quarter of memory wins by default.

Okay so far. Now let's consider where BASIC programs reside in memory. Generally they begin at a memory location called LOMEM, which can vary but is usually between $1C00 and $2400 (about 7000 and 9000) when DOS is booted. BASIC programs always end below screen memory, which in turn is below the BASIC ROM. In practice, this means that BASIC programs and their variables are limited to a length of about 31K.

Let's assume that LOMEM is at $2000 (8192). Let us also assume that we have loaded or typed in a BASIC program which is 12,000 bytes long. Where does that program end? Smack dab in the middle of the second quarter of memory, where the banks are selected.

You might think that this would cause a problem on a 130XE, since it has to switch that bank of memory on and off. But it's not a problem, because one of those five banks is assigned to be main memory—that is, the memory corresponding to the only memory at that address in a 1200XL or 800XL. The DOS 2.5 RAM disk never touches that bank; it limits itself to the other four banks.

Okay, enough background on the 130XE. Is there a way to use the extra 16K memory of the 800XL as a RAM disk? Yes, but it isn't easy. That extra memory is addressed from $C000 to $FFFF (but see below for an exception). Aside from the fact that DOS 2.5 wasn't designed to see a RAM disk in this address range, this range is shared with the operating system ROMs and the hardware input/output area. Shared? Yep, more bank selection. And this bank is even trickier to use.

To Be Continued

Just as things start to get interesting, I run out of room. There is much more to this topic. For example, we haven't even looked for the missing 2K of RAM in the XLs and XEs, have we? And wouldn't it be nice to consider the effects of some of the add-on memory kits for the XLs? Until next month, let me tantalize you with some tidbits.

The RAM disk which emulates drive 8 (D8:) is one of the nice features of DOS 2.5. One of the not-so-nice features is that the RAM disk is *always* D8:. Many, many programs which want two disk drives assume that the second drive is D2:. Wouldn't it be nice if we could change the RAM disk's drive number? Say no more. The BASIC program listings below accomplish this for you.

Program 1, "REPLACE.BAS,"

is for use with the RAMDISK.COM program supplied with DOS 2.5. After you boot the system with DOS 2.5 and RAMDISK.COM, this program simply changes all the magic memory locations in DOS so that the RAM disk is now addressed as D2:. (Or you can change lines 190 and 260 to make the RAM disk emulate any drive from D2: to D8:.) If you use Program 1, the DOS files DUP.SYS and MEM.SAV will be on D2:, but otherwise DOS 2.5 will be unchanged.

Program 2, "MAKERAM-.BAS," serves another purpose. As you've probably noticed, DUP.SYS and MEM.SAV take up a lot of room on the RAM disk. True, keeping them on the RAM disk does make DOS easier to use. However, if your program won't use DOS but could use more RAM disk space, why not leave them on D1:? That's exactly what MAKERAM.BAS does. It initializes and installs the RAM disk, but copies no files to it—all 499 RAM disk sectors are available for your use. Naturally, you may choose any drive number for the RAM disk (see lines 190 and 260 again). And, although we could change this program to allow it to work *after* RAMDISK.COM has booted, it is a waste of time since this program reinitializes the RAM disk, anyway. Therefore, you should erase or rename the RAMDISK.COM file when using MAKERAM.BAS (but don't erase your only copy of RAMDISK.COM).

Finally, Program 3, "MAKERAM.SUB," simply changes Program 2 into a subroutine which you can include in your own programs. Use it anytime you want your program to initialize a blank RAM disk.

For instructions on entering these listings, please refer to "COMPUTE!'s Guide to Typing In Programs" in this issue of COMPUTE!.

Program 1: REPLACE.BAS

```
HF 100 REM
DA 110 REM ==== REPLACE BAS
       ====
HH 120 REM
OC 130 REM A program to repl
       ace D8: with
CM 140 REM Dn: where n is an
       y drive
BL 150 REM number from 2 to
       7 (or even 8)
HL 160 REM
AI 170 IF PEEK(1802)<128 THE
       N PRINT "No RamDisk i
       nstalled!":STOP
HN 180 REM
CD 190 RAMDRIVENUM=2:REM Cha
       nge this as desired
HG 200 REM
AE 210 POKE 1920,RAMDRIVENUM
AM 220 POKE 2953,RAMDRIVENUM
KG 230 POKE 5439,48+RAMDRIVE
       NUM
MF 240 POKE 1802,PEEK(1802)-
       128
LK 250 REM (for changes to l
       ine 260, see "Mapping
        the Atari")
OK 260 IF PEEK(1802)=1 AND R
       AMDRIVENUM=2 THEN POK
       E 1802,3
FA 270 DIM INIT$(4)
HP 280 FOR I=1 TO 4:READ DAT
       A
PI 290 INIT$(I)=CHR$(DATA):N
       EXT I
AC 300 DATA 104,76,224,7
NE 310 JUNK=USR(ADR(INIT$))
HJ 320 REM
FI 330 REM Verify it worked
HL 340 REM
JH 350 DIM DRIVE$(6)
CB 360 DRIVE$="Dn:*.*"
IJ 370 DRIVE$(2,2)=CHR$(48+R
       AMDRIVENUM)
HP 380 REM
KK 390 OPEN #1,6,0,DRIVE$
BC 400 TRAP 430
LB 410 GET #1,BYTE:PRINT CHR
       $(BYTE);
GE 420 GOTO 410
GO 430 END
```

Program 2: MAKERAM.BAS

```
HF 100 REM
GA 110 REM ==== MAKERAM.BAS
       ====
HH 120 REM
IO 130 REM A program to set
       up a RAM disk on
FI 140 REM Dn:, where n is a
       ny drive
BL 150 REM number from 2 to
       7 (or even 8)
HL 160 REM
CO 170 IF PEEK(1802)>127 THE
       N PRINT "RamDisk alre
       ady installed!":STOP
HN 180 REM
CD 190 RAMDRIVENUM=2:REM Cha
       nge this as desired
HG 200 REM
AE 210 POKE 1920,RAMDRIVENUM
AM 220 POKE 2953,RAMDRIVENUM
DC 230 POKE 5439,49
MO 240 REM (line 230 forces
       DUP.SYS to drive 1)
LK 250 REM (for changes to l
       ine 260, see "Mapping
        the Atari")
OK 260 IF PEEK(1802)=1 AND R
       AMDRIVENUM=2 THEN POK
       E 1802,3
FA 270 DIM INIT$(4)
HP 280 FOR I=1 TO 4:READ DAT
       A
PI 290 INIT$(I)=CHR$(DATA):N
       EXT I
AC 300 DATA 104,76,224,7
NE 310 JUNK=USR(ADR(INIT$))
HJ 320 REM
JF 330 DIM DRIVE$(6)
CE 340 DRIVE$="Dn:*.*"
IH 350 DRIVE$(2,2)=CHR$(48+R
       AMDRIVENUM)
HN 360 REM
NL 370 REM Initialize our ne
       w disk
HP 380 REM
CJ 390 XIO 254,#1,0,0,DRIVE$
HI 400 REM
FH 410 REM Verify it worked
HK 420 REM
KF 430 OPEN #1,6,0,DRIVE$
BK 440 TRAP 470
LF 450 GET #1,BYTE:PRINT CHR
       $(BYTE);
GM 460 GOTO 450
HC 470 END
```

Program 3: MAKERAM.SUB

```
BD 10 GOSUB 9000:REM Your pr
      ogram here
DJ 20 END
KN 9000 REM
KM 9010 REM ==== MAKERAM.SUB
        ====
KP 9020 REM
HP 9030 REM Subroutine to se
        t up RAM disk
LB 9040 REM
EE 9050 IF PEEK(1802)>127 TH
        EN PRINT "Disk alrea
        dy installed!":STOP
CI 9060 POKE 1920,2
DA 9070 POKE 2953,2
GO 9080 POKE 5439,49
CL 9090 POKE 1802,3
FI 9100 DIM RAMDISK$(4)
PD 9110 FOR N=1 TO 4:READ X
EI 9120 RAMDISK$(N)=CHR$(X):
        NEXT N
DM 9130 DATA 104,76,224,7
OF 9140 JUNK=USR(ADR(RAMDISK
        $))
PM 9150 REM (any handy strin
        g can be used instea
        d of DRIVE$)
MP 9160 DIM DRIVE$(6)
CC 9170 DRIVE$="D2:*.*"
FP 9180 XIO 254,#1,0,0,DRIVE
        $
LD 9190 RETURN                  C
```

Attention Programmers

COMPUTE! magazine is currently looking for quality articles on Commodore, Atari, Apple, and IBM computers (including the Commodore Amiga and Atari ST). If you have an interesting home application, educational program, programming utility, or game, submit it to COMPUTE!, P.O. Box 5406, Greensboro, NC 27403. Or write for a copy of our "Writer's Guidelines."

Donald B. Trivette

WW II And KQ III

GATO is one of the most interesting games to come along for the IBM PC, PCjr, and compatibles in the last year. It's a strategy game that puts you in the captain's seat of a World War II *Gato*-class submarine. Your mission may be to rescue a downed pilot, resupply a friendly coast watcher, or sink an enemy fleet. Once you receive your orders, you must pilot your boat through enemy waters and around dangerous reefs using radar, charts, and the periscope—if you dare to risk detection.

Although *GATO* is billed as a submarine simulation, it's not a simulation like Microsoft's *Flight Simulator*. You won't actually learn to operate a sub or to navigate underwater. Nevertheless, there are ample controls—depth, speed, heading, fuel, battery, torpedo, periscope—to keep your fingers busy.

You won't master *GATO* in a few days—or even weeks. The level of difficulty is set by a program parameter: At level 0, where I play, Morse-code messages are translated into English and enemy ships leave a convenient trace on the patrol chart. (Even so, my record isn't good—I complete only half of my assigned missions.) At level 9 (for Annapolis graduates, I think), you'd better know Morse code and be able to make plots of enemy activity.

This isn't a game where you can shoot at everything in sight. Successfully completing the mission is the most important goal, and accomplishing that requires the use of strategy to survive.

GATO requires a PC with color/graphics adapter, 128K of RAM, and a color monitor, or a PCjr with a color monitor. It is produced by Spectrum HoloByte, Inc. ($39.95).

A Peek At A Sequel

The *King's Quest* series of adventure games has one of the largest followings of any entertainment program for the IBM. Whenever I write about *King's Quest*, I get lots of letters—some of them quite unique. (One lady wanted to give her husband the gnome's name for his birthday.) Anyway, someone on the inside has slipped me a copy of the design specifications and some memos between the designer and programmers for *King's Quest III*, which Sierra On-Line is working on for release in late fall. I won't spoil your fun by revealing too many secrets, but I'll drop some hints of what's to come in this eagerly awaited sequel. These notes also provide some insight into how a major adventure game is carefully planned and executed by a whole team of designers, artists, and programmers. It's almost like storyboarding a film script.

From the designer's notes: "I'm going to try to make *KQ3* more difficult to solve...I'd like it to be able to do its own mapping, but Ken and Jeff will have to be talked into this...I would like to try to add more arcade-type action, but still retain the flavor of an adventure game." The notes also indicate that there will be a new routine to draw the screens because some players (including myself) are getting important clues by watching what is drawn last in a scene.

"Included in the documentation will be the magic spell book, *Sorcery of Old*." The notes mention numerous spells, including one to transform someone into a cat and another to brew up a storm. It also mentions an invisibility ointment made from toad spittle, and the new cast of characters: Medusa, a huge spider, bandits, pirates, and an abominable snowman who lives in the mountains and will drag you into his cave and devour you for dinner. The notes indicate that the best way to deal with the snowman is to use a protective spell.

Here is the designer's description of Room 25: "Ocean side. Looks like north Calif. coastline. All, part, or none of the town will be in this picture, depending on how you draw it. There will be a dock or pier going out into the ocean from the town. Later on in the game, there will be a pirate ship that is tied to the dock. The pirate ship will probably be two screens long. You can get ocean water from this room for a spell...I'm not sure yet. Maybe, we will see a pirate walking around on the deck while it is tied to the dock and his mates are in the tavern. Or maybe we'll see an old man sitting outside the tavern, or maybe a woman coming out of the store or something. Just to make the town look like it is inhabited."

The notes also indicate that Room 38 (scene 38) is inside the bandit's hideout, and that a bandit will always be there to protect a bin. What the bin contains is unclear.

If my Sierra On-Line contact, known as Deep Ego, can come up with more, I'll let you know.

* * *

Here's a tip for those of you who are running Microsoft's *Flight Simulator* on the IBM PCjr. On some TV sets the colors will fade in and out. This occurs only with version 2.11 or earlier, only on the PCjr, and only with some TV sets. Nevertheless, what looks like a hardware problem is really a bug in *Flight Simulator*. If you call your Microsoft customer service number, they have a fix. ©

AmigaView

Charles Brannon, Program Editor

A New Operating System

Computer software continually evolves, and operating systems are no different. The operating system is the core software of your computer, responsible for managing the hardware and providing routines for other programs to draw upon. The Amiga operating system, for example, contains routines that support menus, windows, memory management, and multitasking.

Most computer operating systems are stored in Read Only Memory (ROM), a permanent, nonalterable form of memory. In contrast, most application software is provided on disk, which is loaded into Random Access Memory (RAM). When updates to the software become necessary (which is almost always the case), the publisher can simply ship new disks.

The only way to upgrade software stored on ROM, though, is to pry out the original ROM chips inside the computer and replace them with new ROMs. This usually requires dealer servicing.

RAM Emulating ROM

The Amiga uses a different technique. It contains only a small amount of ROM which loads the bulk of the operating system from the *Kickstart* disk into a special area of RAM called the Writeable Control Store (WCS). Once this RAM is filled, a special switch write-protects it—effectively turning the RAM into ROM as long as the computer is turned on. The WCS cannot be corrupted by an errant application program or even a system crash.

The WCS was originally intended as a stopgap measure until the operating system could be firmed up and burned into ROM chips. But soon after the computer was introduced, Amiga recognized the value of an easily upgradable operating system and decided to stick with the WCS. One upgrade has already been released: The original version 1.0 was replaced with version 1.1 in late 1985. Version 1.1 added new features and cured legions of bugs that plagued 1.0, but it is still not perfect.

Over the past few months, Amiga has been working very hard to finish version 1.2. This upgrade was developed at first to work with the European Amiga, but includes numerous bug fixes and improvements as well. At this writing (mid-May), we have been exploring a prerelease version of 1.2, which might be available by the time you read this. Note that some features we'll describe may be changed in the final release version.

The most noticeable improvement in 1.2 is the much faster disk access due to a technique known as *caching*. A disk cache buffers disk reads so that frequently accessed areas of the disk are copied into RAM. From then on, the frequently accessed files are read from RAM rather than from the drive. It's similar to using the RAM disk, except that output is always stored on disk, not in RAM, so this technique is much safer than using a RAM disk. If the power is interrupted, you haven't lost your data.

Version 1.2 lets you choose how much memory to allocate for this disk cache—the more memory you set aside, the faster the disk access. The disk directory is also buffered, so directory-based operations such as Open requesters or an AmigaDOS DIR command work much faster. As a tradeoff, the momentary disk access that takes place when you first insert a disk lasts a little longer, since all directories and subdirectories are buffered. And, of course, there's less RAM available for applications, since the cache consumes some memory.

A Better Workbench

The Workbench is improved, too. The horizontal lines in a window's title bar have been thickened to reduce flickering in the interlaced modes. When entering text into a text gadget, you can reposition the cursor by pointing and clicking the mouse. You can use Left Amiga-V and -B as shortcuts for the affirmative and negative choices in a two-button requester. When you drag icons, you move an actual copy of the icon rather than a crossed circle. This even works with multiple selections, and is really impressive when you are dragging dozens of icons. Opening a Workbench window is no longer an excuse for a coffee break: Icons now pop up quickly, with little disk access. Any reference to the RAM: device creates an icon for the RAM disk on the Workbench screen, especially handy for one-drive systems.

A new Preferences tool lets you select serial port parameters such as data bits, stop bits, and so on, greatly simplifying the use of a serial printer or modem. There's also a toggle switch for Workbench Interlace On/Off. When Interlace is turned on, the Workbench changes to a 400-line screen with twice the vertical resolution, giving 50 lines of text.

There is a new Notepad on the Workbench disk, enhanced with an Edit menu permitting copy/cut/paste and search-and-replace. You can set up the Notepad so it calls up only one font when loaded, then bring in the fonts later from a menu if you wish. You can select either character wrap or word-wrap, and you can intermix various fonts and styles in the same note. Scroll bars let you move quickly through your text. The Notepad is now almost a complete word processor.

All in all, the new operating system is very exciting. It almost makes the Amiga a whole different machine: faster, smoother, and more reliable than ever. ©

Bill Wilkinson

GEM Quirks

The Atari ST is a computer with excellent hardware, but all too often problems with its system software obscure this excellence. Admittedly, most users will never actually see these problems, since software developers work hard to circumvent them. Luckily, application programmers can make a real contribution to the users' perceptions of a machine.

For example, consider the ST's floppy disk drives. In theory they are among the fastest available for any microcomputer. And indeed, when you load a program, the speed is impressive. However, when a program starts performing file input/output using ordinary record sizes, there is so much operating system overhead to overcome that the ST performance is only fair. Creating a new file with 512-byte records is only a little more than twice as fast on an ST as it is on an Atari 400/800, XL, or XE.

Possible solution: The application program can read and write very large blocks to the disk (for example, 4K or bigger), performing the file buffering itself. Suddenly the performance is quite good again. This requires a little more work on the part of the application programmer, but the net effect is pleasing for the user.

Similarly, using a hard disk on the ST is an experience not to be forgotten. For example, compiling an average-length program with *Personal Pascal* usually takes one to two minutes using floppies. When using a hard disk, those times improve to 10 or 15 seconds. That's because the hard disk port on the ST is capable of transferring more than one megabyte per second.

But something happens as the hard disk starts filling up. Access times can double before the disk is even half full. Again, there's a solution: Partition the 20-megabyte disk into four smaller, five-megabyte "logical" drives. And, since the ST uses subdirectories so successfully, this is usually a practical solution.

Gullible GEM

Perhaps the biggest problem with GEM (the Graphics Environment Manager) is that it is too gullible—tell it a lie and it believes you. Consider what happens on an Atari 400/800, XL, or XE when an Atari BASIC programmer uses a PRINT statement to display a message which is wider than the screen: The text wraps around to the next line.

When programming with GEM, the easiest way to display something on the screen is via an *alert box.* This is the small window which pops up to report errors and so forth. To display an alert box, a programmer simply defines a string of the proper form and makes an easy call to a GEM routine. But if the programmer errs when defining that string (for example, by entering too many characters or leaving out some special characters), *crash!* Time to hit the old reset button.

Now, granted, the proper form of that string is easy to validate before calling GEM, so a well-written application program will never reveal this particular problem to its user. However, this is symptomatic of much of GEM. Application programmers must do a lot of work to insure that GEM is given only legal values to work with. GEM does not seem to follow the *GIGO* rule (Garbage In, Garbage Out); with GEM it is more like *GIC* (Garbage In, Crash!). So be careful if you're writing programs on the ST. Avoid crashes by double-checking all data before calling GEM routines.

The Software Explosion

To a beginner, the ST with its GEM operating system looks complex. And, truly, there is a *lot* to learn before you can write programs which show off all the capabilities of the ST. But, despite my comments above, experienced programmers find that GEM does so much of the work for them that they can develop fairly complex programs relatively quickly. Too, the capabilities and accessibility of higher-level languages for the ST (such as C, Pascal, and Modula-2) have made programmers more productive. As a result, there is arguably more software available for the ST, *at this point in its life*, than for any previous computer at a comparable point in its life.

For instance, one year after the Macintosh was introduced, it had far fewer programs available than the ST has about one year after its introduction. Not only that, the ST programs tend to be considerably less expensive than their Macintosh counterparts.

One of the reasons so much software is appearing is that the cost of developing for an ST is relatively low. A part-time ST programmer can have a full-blown ST development system for not much over $2,000 (including hard disk, printer, color and monochrome monitors, development software, and so forth). In the early days of the Mac, $10,000 was more the order of the day, so development tended to be restricted to established software companies.

The flip side of this coin is that the quantity of *high-quality* software for the ST is certainly *not* greater than what was available for the Macintosh. Since most early Mac developers were major software companies, their quality standards were generally higher than that of part-time hackers.

Bottom line: Try to see a demo of any ST software you are planning to purchase. There are a lot of excellent ST programs, but there are also some turkeys. ©

Programming the TI

C. Regena

An Amortization Schedule

Interest rates have been plunging lately, and it seems like home mortgages and refinancing are very popular topics for newspaper articles. Recently I was reading a question-and-answer article in which the reader asked for a program for his home computer to print an amortization schedule for a home mortgage. The columnist suggested a particular program which was easy to use and costs only $99. I couldn't believe someone would spend $99 for a program that uses one or two basic computations! So, for the price of this magazine, here is such a program: "Loan Amortization."

It's certainly easy to use. Just enter the amount of money you want to borrow, omitting the dollar sign and comma (i.e., type 50000 instead of $50,000). Next, enter the interest rate, such as 13 for 13 percent or 9.5 for nine and a half percent. Finally, enter the number of years for the loan. Most loans are for a certain number of whole years, such as 25 or 30, so this program is based on 12 monthly payments per year rather than calculating a number of months. The program then tells you what your monthly payment will be. (Of course, this figure doesn't include property taxes, insurance, or condominium fees.)

You may then choose to see the amortization schedule on the screen or print out a paper copy. If you have a printer, be sure to use the correct printer configuration in line 710, the OPEN statement. If you don't want to see the amortization schedule, you may calculate another loan or end the program.

Converting Math To BASIC

Among other things, Loan Amortization demonstrates how easy it can be to convert a mathematical formula into a BASIC program. Any ordinary formula can be converted by using the + and − signs for addition and subtraction, the * sign for multiplication, / for division, and sets of parentheses where necessary to group mathematical operations.

Use PRINT and INPUT statements to prompt numbers from the user. You may want to use some IF-THEN statements to make sure the INPUT values are within reasonable limits for the formula. In Loan Amortization, all numbers entered must be positive. The amount of the loan has to be six digits or less (not counting the cents) to help limit the printing variables. The number of years is from 1 to 50.

Once your program has all the numbers it needs, calculate the formula and PRINT the answer. The computer, of course, is ideal for handling repetitious calculations, such as this amortization schedule.

Any economics book has formulas for various calculations involving money—savings accounts, sinking fund deposits, present worth factors, and so forth. In this case, to find the monthly payment I used the capital recovery factor formula:

```
I(1+I)^N/(1+I)^N-1
```

where I = interest and N = the number of payments. To make it easier to type the program without errors, I used the variable D for interest, since the letter I can be confused with the numeral 1. Then the program converts the percentage to a monthly decimal, J=D/1200. The factor with the exponent is used twice, so I calculated it as F in line 490. Line 500 then calculates the capital recovery factor, CRF.

How To Pause Printing

The FOR-NEXT loop in lines 800–1050 prints the amortization schedule with the monthly payment PAY. Part of the payment goes to principal (the variable PR), and part is interest (the variable II). The balance is the original principal minus the principal part of the payment, P. Lines 1060–1200 calculate and print the last payment, which may be slightly different than the regular monthly payment because of rounding to the cent.

The printing on the screen includes only the month number, principal and interest, then balance. To pause the printing while it is scrolling, hold down any key. When you release the key, the schedule will continue. To make this work, lines 1010–1040 scan the keyboard in each loop. If a key is not pressed, the program goes to the next calculation. You may want to print different items or adjust the printing to better suit your needs.

All of the PRINT # statements send text to the printer. The variables L1, L2, and L3 are lengths used in the TAB functions to line up the columns. The variable R holds the user's choice: 1, 2, 3, or 4. If the choice is 1, the program skips all the statements that pertain to the printer.

The subroutine in lines 1250–1330 converts a number in the variable A to a string so that a number can be written in money form with two decimal places (using zeros where necessary). The numbers are rounded to the nearest cent.

If you have TI Extended BASIC or are converting this program to another version of BASIC, PRINT USING would be easier to use than this subroutine. For example, PRINT USING ####.## will round a number to two decimal places and will also right-justify numbers for printing straight columns.

Loan Amortization

```
100 REM  AMORTIZATION
110 CALL CLEAR
120 PRINT "THIS PROGRAM WIL
    L CALCULATE"
130 PRINT "A MONTHLY PAYMEN
    T FOR A"
```

```
140 PRINT "GIVEN PRINCIPAL
    BORROWED"
150 PRINT "AT A CERTAIN INT
    EREST RATE."
160 PRINT ::"ENTER AMOUNT B
    ORROWED."
170 INPUT PP
180 IF PP>0 THEN 210
190 PRINT "PLEASE ENTER AMO
    UNT > 0"
200 GOTO 160
210 IF PP<999999.01 THEN 25
    0
220 PRINT "THIS PROGRAM IS
    FOR LOANS"
230 PRINT "LESS THAN $99999
    9."
240 GOTO 160
250 PRINT ::"ENTER INTEREST
     RATE IN %."
260 INPUT D
270 IF D>=0 THEN 300
280 PRINT "PLEASE USE POSIT
    IVE PERCENT."
290 GOTO 250
300 PRINT ::"ENTER NUMBER O
    F YEARS FOR"
310 PRINT "LOAN."
320 INPUT Y
330 IF (Y>=1)+(Y<51)=-2 THE
    N 370
340 PRINT "THIS PROGRAM IS
    FOR LOANS"
350 PRINT "FROM 1 YEAR TO 5
    0 YEARS."
360 GOTO 300
370 IF Y=INT(Y)THEN 400
380 PRINT "NO FRACTIONAL YE
    ARS PLEASE."
390 GOTO 300
400 CALL CLEAR
410 PRINT "AMOUNT BORROWED:
     ";PP
420 PRINT :"INTEREST RATE:
    ";D;"PERCENT"
430 J=D/1200
440 PRINT "TIME IN YEARS: "
    ;Y
450 N=12*Y
460 IF D<>0 THEN 490
470 CRF=1/N
480 GOTO 510
490 F=(1+J)^N
500 CRF=J*F/(F-1)
510 PRINT STR$(N);" MONTHLY
     PAYMENTS"
520 A=PP*CRF
530 GOSUB 1250
540 PAY=A
550 PAY$=A$
560 PRINT "MONTHLY PAYMENT
    = $";A$
570 PRINT ::"PRINT AMORTIZA
    TION?"
580 PRINT :"1 YES, ON SCREE
    N"
590 PRINT "2 YES, ON PRINTE
    R"
600 PRINT "3 NO, TRY ANOTHE
    R LOAN"
610 PRINT "4 NO, END PROGRA
    M"
620 CALL KEY(0,K,S)
630 IF (K<49)+(K>52)THEN 62
    0
640 CALL CLEAR
650 R=K-48
660 ON R GOTO 670,670,110,1
    340
670 A=PP
680 GOSUB 1250
690 P=A
700 IF R=1 THEN 750
710 OPEN #1:"RS232.BA=600"
720 PRINT #1:"AMOUNT BORROW
    ED:  $";A$
730 PRINT #1:"INTEREST RATE
    : ";D;"PERCENT"
740 PRINT #1:::"MONTH    PA
    YMENT";TAB(30);"PRINCIP
    AL";TAB(50);"INTEREST";
    TAB(65);"BALANCE":::
750 PRINT "TO PAUSE PRINTIN
    G, HOLD ANY KEY DOWN.
    RELEASE KEY TO   CONTIN
    UE.":::
760 PRINT "AMOUNT BORROWED:
     ";A$
770 PRINT "INTEREST RATE: "
    ;D
780 PRINT :"MONTHLY PAYMENT
    : $";PAY$
790 PRINT :"     PRINCIPAL
       INTEREST":TAB(12);"B
    ALANCE"::
800 FOR M=1 TO N-1
810 M$="  "&STR$(M)
820 M$=SEG$(M$,LEN(M$)-2,3)
830 A=J*P
840 GOSUB 1250
850 I1$=A$
860 I1=A
870 L2=6-L
880 A=PAY-I1
890 GOSUB 1250
900 PR$=A$
910 PR=A
920 L1=6-L
930 A=P-PR
940 GOSUB 1250
950 P=A
960 P$=A$
970 L3=6-L
980 PRINT M$;"  ";PR$;TAB(1
    8+L2);I1$:TAB(10+L3);P$
990 IF R=1 THEN 1010
1000 PRINT #1:" ";M$;TAB(11
     );PAY$;TAB(31+L1);PR$;
     TAB(51+L2);I1$;TAB(65+
     L3);P$
1010 CALL KEY(0,K,S)
1020 IF S<1 THEN 1050
1030 CALL KEY(0,K,S)
1040 IF S<0 THEN 1030
1050 NEXT M
1060 M$="  "&STR$(M)
1070 M$=SEG$(M$,LEN(M$)-2,3)
1080 A=J*P
1090 GOSUB 1250
1100 I1$=A$
1110 I1=A
1120 L2=6-L
1130 A=I1+P
1140 GOSUB 1250
1150 PAY=A
1160 PAY$=A$
1170 L1=6-L
1180 PRINT M$;"  ";P$;TAB(1
     8+L2);I1$:TAB(15);"0"
1190 IF R=1 THEN 1220
1200 PRINT #1:" ";M$;TAB(11
     );PAY$;TAB(31+L1);P$;T
     AB(51+L2);I1$;TAB(68);
     "0"
1210 CLOSE #1
1220 PRINT ::"PRESS A KEY"
1230 CALL KEY(0,K,S)
1240 IF S=0 THEN 1230 ELSE
     570
1250 A=INT(A*100+.5)
1260 A$=STR$(A)
1270 L=LEN(A$)
1280 IF L>=2 THEN 1310
1290 A$="0"&A$
1300 L=2
1310 A$=SEG$(A$,1,L-2)&"."&
     SEG$(A$,L-1,2)
1320 A=VAL(A$)
1330 RETURN
1340 END
```

©

ATTENTION T.I. 99/4A OWNERS

- Diskettes - 59¢ each! Your choice SS or DD
- 512K Now Available for the 99/4A!
- 99/8 Level 4 Computer Upgrade Now Available
- Over 1500 Hardware and Software Accessories at Similar Savings

THE WORLD'S LARGEST COMPUTER ASSISTANCE GROUP

Now serving over 35,000 members worldwide with the best in technical assistance, service, and products for the Texas Instrument 99/4A Computer.

To become a member and receive newsletters, catalog, technical assistance and membership package, send $10.00 for a ONE Year Membership to:

99/4A National Assistance Group
National Headquarters
P.O. Box 290812
Ft. Lauderdale, Florida 33329
Attention Membership Division
For Further Information Call 24 Hours
(305) 583-0467

Penguin Software Announces Price Drop

Penguin Software has announced an across-the-board price drop for its software line. All programs in its COMPREHEND Interactive Novel series will be $17.95 for 5¼-inch disks (Apple, Commodore 64, IBM) and $19.95 for 3½-inch disks (Atari ST, Macintosh, and Amiga). This line includes such titles as *Crimson Crown, Oo-Topos,* and *Transylvania.* Suggested retail price for *Graphics Magician* and *Complete Graphics System* will be $39.95 ($49.95 for Macintosh version).

Other graphics utilities, such as *Paper Graphics, Transitions,* and *Cat Graphics* will be priced at $19.95. *Graphics Magician Junior* (Apple and Commodore) will be $19.95. *Disk Repair Kit* will be $12.95. In the Home series, *Home Data Manager* will be priced at $24.95, and *Home Connection* (with $15 free CompuServe time) will be $29.95. Also, some backlist titles will be available for $8.95.

Penguin Software, 2600 Keslinger Rd., P.O. Box 311, Geneva, IL 60134.

Circle Reader Service Number 220.

New Reading, Social Studies Software

CBS Interactive Learning has introduced *The Novel Approach: Lord of the Flies,* the first title in the Novel Approach computer software series developed by Media Basics for Apple, IBM, and Commodore eight-bit systems. Each program in the series focuses on a popular literary classic frequently studied in junior and senior high school. Designed to help students develop or maintain interest in reading and to build critical reading skills, each title in the Novel Approach can be used as a springboard for classroom discussion or independent study. Four additional Novel Approach titles are planned for release in the fall of 1986. They are *Animal Farm* by George Orwell, *A Tale of Two Cities* by Charles Dickens, *The Call of the Wild* by Jack London, and *Romeo and Juliet* by William Shakespeare.

The Novel Approach series motivates students to read by enhancing their understanding and appreciation of literature. Each program helps students focus on character motivation, plot development, symbolism, narrative techniques, and vocabulary. Rather than replacing the reading of the book itself, the programs are meant to be used before, during, and after reading. Each includes three separate learning activities: The Discoverer, designed to pique interest before reading; The Explorer, a self-paced series of questions and answers that enhance understanding; and The Master, designed to test students' knowledge of the story after it has been read.

Built into each program in the Novel Approach series is a comprehensive reference guide, *The Book Scanner.* It provides background information on each book, a profile of the author, and an annotated bibliography of related books. Errors are tracked, and corrections with explanations are provided.

The Novel Approach: Lord of the Flies comes with a program guide, teacher's guide, and backup disk. It is available for the Apple II series (48K RAM minimum), Commodore 64, and IBM-PC and PCjr with 128K RAM and graphics board for $59.95.

CBS also has introduced *Continents and Countries,* a program for use within the social studies curriculum in grades 5–12. Designed by Neosoft, the program helps students build and test their knowledge of the nations and peoples of the world through self-paced learning activities. Its database covers over 140 countries and includes facts on each country's major religion, language, per capita income, land area, form of government, and population. *Continents and Countries,* available for

the Apple II series with 48K RAM minimum, has a suggested retail price of $49.95.
CBS Interactive Learning, One Fawcett Pl., Greenwich, CT 06836.

Circle Reader Service Number 221.

Baseball And Bridge For Apple

Random House has announced Apple II conversion of two programs. *APBA Major League Players Baseball* uses actual statistics from the 1984 or 1985 baseball season and lets users start their own leagues, draft teams from a list of 676 big-league players, or play with the actual rosters for all 26 teams from each season. The program is now available for Apple IIe and IIc with 128K, 80-column card, and two disk drives for $59.95.

Tournament Bridge offers competition and practice for the serious bridge player. It is available for the Apple II+, IIe, and IIc for $49.95. Random House also is developing a word processor for Apple II computers that uses a Macintosh- style user interface. A fall release is planned.
Random House, Electronic Publishing Division, 201 E. 50th St., New York, NY 10022.

Circle Reader Service Number 222.

Foreign Language Detective Game

Gessler Educational Software has announced French, Spanish, and German versions of Tom Snyder's bestselling program *Snooper Troops*. The program helps children develop their foreign language vocabularies and reasoning skills by having them take notes, draw maps, and organize information.

The object of the Spanish and German versions is for the player to determine who committed a crime in the old mansion and why. The player must question the suspects and remember each correct password and clue in order to solve the mystery. Available for the Commodore 64, the Spanish and German versions retail for $39.95. In the French version, available for Apple II+, IIe, and IIc at $49.95, the player's mission is to find the villain who fled with Lily the Dolphin.
Gessler Educational Software, 900 Broadway, New York, NY 10003.

Circle Reader Service Number 223.

More New Releases From The U.K.

Firebird Licensees, a British software licensing company which made a nice entry into the U.S. market with *Elite*, recently introduced several new products.

The Pawn is a rich text-and-graphics adventure previously available for the Atari ST, but now shipping for the Commodore 64 and 128 (in native 128 mode). Set in the mythical world of Kerovnia, the game provides the player with an intricate network of plots and subplots with many objectives.

New members of the Firebird "flippy" family (disks with one program on each side) are *Battle of Britain/Battle of Midway* (Commodore 64, $19.95), strategy/war games that break out into arcade-style games at certain points in the action; *Iwo Jima/Falklands '82* (Commodore 64, $19.95); and *Chimera/Mercenary* (Atari 800/130, $19.95).
Firebird Licensees, P.O. Box 49, Ramsey, NJ 07446.

Circle Reader Service Number 224.

ST, Amiga Programs

Classic Image is releasing two programs each for the Atari ST-series computers and the Commodore Amiga.

Disk Library is a tool for keeping track of files on your disks. Files, folders, and subdirectories can be categorized and cross-referenced. Lists of files and folders can be displayed on the screen or dumped to a printer. Disks can be searched by any category, and the entire library is automatically updated as new disks are added. *Disk Library* works with single- or multiple-drive systems and is available for both the ST and Amiga for $49.95.

Diablo is an original game that combines animation with strategy. The screen is filled with mazelike tracks that disappear in sections as a ball rolls over them. The player's goal is to keep the ball rolling as long as possible without running out of track. Versions for the ST and Amiga retail for $29.95 each.
Classic Image, 510 Rhode Island Ave., Cherry Hill, NJ 08002.

Circle Reader Service Number 225.

Turbocharged Amiga

Computer System Associates has introduced a series of add-on circuit boards that modify a Commodore Amiga for high-speed operation using the Motorola 32-bit 68020 microprocessor.

A specially modified Turbo-Amiga runs at a CPU (central processing unit) clock speed of 14 megahertz, contains up to 2.5 megabytes of 32-bit memory, and can accept an optional Motorola 68881 math coprocessor. The 68020 modification alone increases overall performance by about 60 percent. By adding 512K of 32-bit memory, performance increases about 140 percent. Applications which use intensive floating-point math can run up to 40 times faster. Complete Turbo-Amiga systems start at $4,980.
Computer System Associates, 7564 Trade St., San Diego, CA 92121.

Circle Reader Service Number 226.

ST, Amiga Golf Game

Accolade has announced that versions of its golf-simulation game will be available this summer for the Atari ST-series and Commodore Amiga computers.

Mean 18: Ultimate Golf uses 3-D animation to simulate golfing on three famous courses—Pebble Beach, St. Andrews, and Augusta National. In addition, you can construct your own courses. A bird's-eye view shows the position of your ball after each shot. Different levels of difficultly accommodate all kinds of players. The ST and Amiga versions of *Mean 18* will retail for $49.95 each.
Accolade, 20833 Stevens Creek Blvd., Cupertino, CA 95014.

Circle Reader Service Number 227.

Recreational Software

Baudville is releasing three new home and educational programs for the Commodore 64, Atari 400/800/XL/XE, Apple II series, IBM PC and compatibles, Atari ST series, Commodore Amiga, and Apple Macintosh.

Video Vegas recreates authentic casino games such as blackjack, draw poker, keno, and slot machines. *Guitar Wizard* helps both novice and experienced musicians learn and analyze scales, chords, and tunings for all kinds of fretted string instruments. *Ted Bear's Rainy Day Games* is a three-in-one card game package for youngsters. It contains computer versions of concentration, old maid, and go fish.

All of the programs are scheduled for release this fall at prices ranging from $29.95 to $34.95.
Baudville, 1001 Medical Park Dr., SE, Grand Rapids, MI 49506.

Circle Reader Service Number 228.

Color Printer For Amiga, ST

Okidata has released adapters to make its Okimate 20 color thermal-transfer printer work with the Commodore Amiga and Atari ST-series computers.

The Plug 'N Print Modules for the Amiga and ST include a cable, cartridge ribbons, paper, and instructions. The Okimate 20 has a 24-element thermal printhead that reproduces more than

ATARI 130XE

ATARI 130XE Super Computer Package

130XE Computer — 1027 Printer
1050 Disk Drive — Atariwriter +

Call for individual & super package price

ATARI PRINTER INTERFACES

Uprint A ..... 54.95 MPP 1150 ..... 54.95

Supra 1000E Modem 44.95
Atari XM-301 Modem 39.95

ATARI 130-XE SUPER PRINTER PACKAGES

NX-10 Printer & U-Print A 299
Panasonic 1091 & U-Print A 299

Super Printer Packages have no extra shipping charges or credit card surcharges when shipped in Continental USA

ATARI 130XE SOFTWARE

BRODERBUND
Print Shop 28.95
Karateka 20.95
Print Shop Graph. I, II, or III ... 19.95
Print Shop Comp. 27.95

INFOCOM
See Commodore 64 section for items and prices

ELECTRONIC ARTS
Archon II 24.95
Archon 19.95
Seven Cit. of Gold 24.95
Skyfox 24.95
Pinball Const 19.95
One on One 24.95
Super Boulder Dash ... 19.95
Chessmaster 2000 ... 27.95
Racing Destruction ... 24.95

MICROPROSE
Silent Service 23.95
Gunship 23.95
Accrojet 23.95
F-15 Strike Eagle 23.95
Kennedy Approach ... 23.95
Conflict/Vietnam 27.95

SSI
See Commodore 64 section for items and prices

MISCELLANEOUS 130XE
Mind Pursuit 23.95
Never-Ending Story ... 23.95
Fooblitzky 27.95
Syncalc 32.95
Typesetter 27.95
Ultima III 37.95
Ultima IV 41.95
Raid Over Moscow ... 27.95
Fight Night 23.95
Hardball 23.95
Flight Simulator II ... 34.95
Alternate Reality 27.95
Fleet System II 49.95
Page Designer 23.95
Megafont II 19.95
Rubber Stamp 23.95
Halley Project 23.95
Synfile 32.95
Beachead II 27.95
Music Studio 20.95
Wizard's Crown 27.95
Gettysburg 39.95
Island Caper 23.95
Basic XE 52.95
Action 49.95
MAC 65XL 49.95
Paper Clip XE 39.95

ATARI 520ST*

Atari 520ST-RGB System ..... Call
Atari 520ST-Monochrome Sys ... Call
SF314DS/DD1 Megabyte Disk Dr ... 209

We warranty all 520ST computers purchased from ComputAbility for ninety days.

***Please call for stock availability on Atari ST products before ordering by mail.**

MISCELLANEOUS ST

Gato 34.95
Kings Quest II 34.95
ST Talk 17.95
H&D Base 69.95
Typesetter ST 27.95
Habadex Phone 39.95
Word Invaders 24.95
Mean 18 Golf 34.95
Financial Cookbook .. 34.95
Bratacccus 34.95
Michtron Utilities 41.95
Exploding Fist 34.95
Black Cauldron 27.95
Printmaster ST 27.95
P.C. Intercom 89.95
Final Word 94.95
Hex 27.95
Mudpies 27.95
Sundog 27.95
Flipside 27.95
Softspool 27.95
VIP Professional 119.95
Ultima II 39.95
Perry Mason 34.95
Degas 27.95
Farenheit 451 ..... 34.95
Personal Pascal 54.95
Checkminder 54.95
Printmaster 27.95
Art Gallery I 20.95
Compubridge 20.95
General Ledger 84.95
Major Motion 27.95
Pawn 27.95
DFT 34.95
Universe II 49.95
Macrodesk 20.95
Rubber Stamp 27.95
Kissed 34.95
Cards 27.95
Rouge 27.95

Amazon 34.95
DOS Shell 34.95
Kissed 27.95
9 Princess/Amber 34.95
Hacker 29.95
Dragonworld 34.95
Treasure Island 27.95
Wizard of Oz 27.95
Transylvannia 27.95
Borrowed Time 34.95
Mi-Term 34.95
Regent Word 34.95
Regent Spell 34.95
Regent Base ..... Call
Goldrunner 27.95
Time Bandit 27.95
Zoomracks 49.95
Easy Draw 99.95
Mindshadow 34.95
Phantasie 27.95
Hippopotamus ..... Call
Supra Hard Drive ..... Call
Supra 1200 ST Modem .. 159
QMI 1200 ST Modem ... 159
PC Board Designer .... Call
Infocom ..... See IBM
Apshai Trilogy 27.95
N-Vision 27.95
DB Man 69.95
Small Bus. Pkg. 84.95
Cornerman 34.95
ALT 34.95
M-Copy 54.95
Winter Games 27.95
Winnie The Pooh 19.95
Silent Service ..... Call
Flight Simulator II ..... Call
Music Studio 39.95
Phantasie 27.95
1 Megabyte Upgrade... 175
Extended Warranty .... Call

AMIGA

Call for Hardware and Add-on Peripherals prices

AMIGA SOFTWARE

Hacker 29.95
Mindshadow 29.95
Mastertype 27.95
VIP Professional 139.95
Transylvannia 27.95
Crimson Crown 27.95
Racter 29.95
Analyze 69.95
On-line 49.95
Artic Fox 29.95
Bratacccus 34.95
Adventure Const 29.95
Deluxe Video ..... Call
Deluxe Paint 69.95
Infocom ..... See IBM
Monkey Business 19.95
Keyboard Cadet 27.95
Rouge 27.95
Apshai Trilogy 27.95
Delta Patrol 19.95

Maxicom 39.95
Maxidesk 49.95
Maxiplan 119.95
One on One 29.95
7 Cities/Gold 29.95
Skyfox 29.95
Marble Madness 29.95
Return/Atlantis 29.95
Archon 29.95
Aegis Animator 99.95
Exploding Fist 27.95
Deluxe Print 69.95
Music Studio 39.95
Borrowed Time 29.95
Financial Cookbook .. 34.95
Scribble 69.95
Halley's Project 29.95
Kid Talk 39.95
Speller Bee 39.95
Gizmo 34.95

APPLE

SOFTWARE

BRODERBUND
Print Shop 39.95
Print Shop Graphics I, II, or III 17.95
Print Shop Comp 27.95
Karateka 21.95

ELECTRONIC ARTS
Bard's Tale 29.95
Auto-Duel 34.95
Skyfox 27.95
Lords of Conquest ... 27.95
Ultima IV 39.95
Moebius 39.95
Amnesia ..... Call

EPYX
Winter Games 24.95
Summer Games II 24.95

INFOCOM
See IBM section for items and prices.

MICROPROSE
See Atari 130XE section for items and prices.

SIR-TECH
Wizardry/Diam 23.95
Wizardry/Legacy 27.95
Wizardry/Proving 33.95

SSI
See Commodore 64 section for items & prices.

APPLE MISCELLANEOUS
Gamemaker 34.95
Hardball 24.95
Sundog 27.95
Newsroom 39.95
Clip Art Vol. I 20.95
Clip Art Vol. II 27.95
Gato 27.95
Kung Fu Master 25.95
Karate Champ 25.95
Strip Poker 23.95
Math Blaster 34.95
Fight Night 24.95
Phantasie II 27.95
Rambo 27.95
Amer. Challenge 27.95
Crossword Magic 34.95

IBM PC

Corona PC-400 Compatible ..... Call
Corona Portable PC Compatible ... Call

IBM PC SOFTWARE

BRODERBUND
Print Shop 39.95
Print Shop Graph. I ... 27.95
Bank Street Writer ... 49.95
Ancient Art of War ... 29.95

MICROPROSE
F-15 Strike Eagle 23.95
Kennedy Approach .. 27.95
Acrojet 27.95
Silent Service 27.95

IBM MISCELLANEOUS
Mean 18 ..... 34.95
Gato 27.95
Wizardry 39.95
Strip Poker 27.95
Jet 34.95
Newsroom 39.95
Superbowl Sunday .. 24.95
Home Pak 34.95
Alter Ego 34.95
Hacker 29.95
Borrowed Time 27.95
Isgur Portfolio 164.95
Typing Tutor 34.95

INFOCOM
Deadline 34.95
Starcross 34.95
Zork I 27.95
Zork II or III 29.95
Witness 27.95
Suspended 34.95
Planetfall 27.95
Sorcerer 29.95
Seastalker 27.95
Cutthroats 27.95
Hitchiker 27.95
Suspect 29.95
Wishbringer 27.95
Infidel 29.95
Enchanter 27.95
Spellbreaker 34.95
Mind Forever 34.95
Ballyhoo 27.95

SIERRA
King's Quest 34.95
King's Quest II 34.95
Ultima II 39.95

COMMODORE

COMMODORE 128

C-128 Computer ..... Call
1571 Disk Drive ..... Call
1902 Monitor ..... Call

1670 Modem 159
1350 Mouse 42.95
1750 512K Expander ... Call

COMMODORE 128 SOFTWARE

Swiftcalc w/Side. 49.95
Wordwriter + Spell ... 49.95
Data Manager II 49.95
Fleet System III 54.95
Mach 128 ..... 34.95
Superbase 128 ..... 69.95
King's Quest II 34.95
Home Pak 34.95

Superscript 128 ..... 59.95
Perfect Writer 49.95
Sylvia Porter-128 .... 49.95
Matrix 39.95
Perfect Filer 49.95
Paperback 128 ..... 34.95
Viziwrite 128 ..... Call
Vizistar 128 ..... Call

COMMODORE 64

NEW 64C Computer ... Call
C-64 Computer ..... Call
1541 Disk Drive ..... Call

1660 Modem 49.95
1670 Modem 159
1802 Monitor ..... Call

COMMODORE 64/128 SUPER PRINTER PKGS.

NX-10 & Xetec Supergraphic. . . .309
Panasonic 1091 & Xetec Supergraphic 309
Legend 1080 & Xetec Supergraphic 269

Super Printer Packages have no added shipping or charge card surcharges when shipped in Continental USA

While Supplies Last

B.I. 80 Column Card 59.95

XETEC Super Graphic 69.95
Cardco G-Wiz 54.95

COMMODORE 64 SOFTWARE

ACCESS
Mach V-Cart 21.95
Leader Board 27.95
Tenth Frame 27.95

SSI
Battle of Antietam 32.95
Fighter Command (No Atari) 37.95
Panzer Grenidier 24.95
USAAF 37.95
Breakthrough/ Ardennes 37.95
Kampgruppe 37.95
Phantasie II (No Atari) 27.95
Broadsides 24.95
Carrier Force 37.95
Comp. Ambush 37.95
Field of Fire (No Apple) 24.95
Gemstone Warrior ... 21.95
Imp. Galactum 24.95
Computer Baseball ... 24.95
Comp. Quarterback ... 24.95
Wizards Crown (No Atari) 27.95
Rings of Zilfin (No Atari) 27.95
Battle Group (No Atari) 39.95
Gettysburg 39.95

MICROPROSE
See Atari 130XE section for items and prices.

ABACUS
Call for Items and Prices

INFOCOM
Zork I 24.95
Zork II, or III 27.95
Deadline 29.95
Starcross 29.95
Witness 24.95
Planetfall 24.95
Hitchiker 24.95
Enchanter 24.95
Cutthroats 24.95
Sorcerer 24.95
Spellbreaker 29.95
Ballyhoo 27.95

ELECTRONIC ARTS
Adv. Construction 29.95
Mail Order Monster ... 24.95
Racing Destruction ... 24.95
Ultima III 37.95
Ultima IV 41.95
Bard's Tale 27.95
Lords of Conquest ... 27.95
Chessmaster 2000 ... 27.95
Touchdown Ftball 23.95
Ultimate Wizard 24.95
Super Boulderdash ... 19.95
Mind Mirror 24.95
See Atari 130XE section for rest of items & prices.

EPYX
Winter Games 24.95
Apshai Trilogy 24.95
Fast Load-Cart 24.95
Vorpal Utility 22.95
Multiplan 44.95
Movie Monster 24.95
World Karate 19.95
Street Sports 24.95
Super Cycle 24.95

MISCELLANEOUS COMMODORE 64
Freeze Frame 34.95
Print Shop 28.95
Cal-Kit 34.95
Superbase 64 ..... 47.95
Karateka 20.95
Hacker 20.95
Gamemaker 27.95
Ultima II 37.95
Karate Champ 25.95
Sticks of Death 20.95
Kung Fu Master 25.95
Paper Clip/Spell 54.95
Consultant 41.95
Spy vs. Spy Vol. II ... 23.95
Internat'l Hockey 19.95
Prt Shop Compan. 27.95
Prt Shop Graphics 19.95
Gato 20.95
Jet 29.95
Fleet System II 44.95
Printmaster 24.95
Newsroom 34.95
Fontmaster II 34.95
Sports Lib. Vol. I 19.95
Geos 39.95
Hidden Assets ..... Call
Adv. Music System ... 54.95
Spitfire 40 ..... 23.95

We stock hundreds of programs for the Apple, Atari, C-64 and IBM. If you don't see it here, don't hesitate to call

GENERAL HARDWARE

Star micronics-inc
THE POWER BEHIND THE PRINTED WORD.

NX-10 ..... Call
SG-10 ..... Call
SG-15 369
SD-10 339
SD-15 449
SR-10 489
SR-15 ..... Call
NL-10 ..... Call
NL-10C ..... Call

PRINTERS
Panasonic 1091 239
Legend 808 169
Legend 1080 209
Powertype 309
Juki 5510 389
Epson ..... Call
Panasonic 1080 215

PRINTER BUFFERS
Microfazer ..... From 169
U-Buff 16K 79.95
U-Buff 64K 99.95

MODEMS
Volksmodem 1200 189
Prometheus 1200 299
Maxwell 1200 ..... Call
Maxwell 2400 ..... Call
Prometheus 2400 ..... Call

MONITORS
Thommson RGB 289
Commodore 1802 ..... Call
Commodore 1902A 289
Gold Star ..... Call
Amdek ..... Call
Panasonic 1300DT 279
NEC 1225 139

— EST. 1982 —

ComputAbility™

P.O. Box 17882, Milwaukee, WI 53217
ORDER LINES OPEN
Mon-Fri. 11 a.m. - 7 p.m. CST
Sat. 12 p.m. - 5 p.m. CST

NO SURCHARGE FOR MASTERCARD & VISA

To Order Call Toll Free
800-558-0003

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

ORDERING INFORMATION: Please **specify system.** For fast delivery send cashier's check or money order. Personal and company checks allow 14 business days to clear. School P.O.'s welcome. **C.O.D. charges are $3.00** In Continental U.S.A. include $3.00 for software orders, 4% shipping for hardware, minimum $4.00. Master Card and Visa orders please include card #, expiration date and signature. WI residents please include 5% sales tax. HI, AK, FPO, APO, Puerto Rico and Canadian orders, please add 5% shipping, **minimum $5.00.** All other **foreign orders add 15% shipping, minimum $10.00.** All orders shipped outside the Continental U.S.A. are shipped first class insured U.S. mail. If foreign shipping charges exceed the minimum amount, you will be charged the additional amount to get your package to you quickly and safely. 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 R.A.# or your return will not be accepted. Priced and availability subject to change without notice.

100 colors, prints text at 80 characters per second in standard mode, or 40 characters per second in a near-letter-quality mode. Pages widths are 80 columns in standard mode or 132 columns in condensed mode. Other modes include expanded, boldface, italic, fine print, underlining, superscripts, and subscripts. An Okimate 20 with Plug 'N Print Module retails for $268.
Okidata, 532 Fellowship Rd., Mt. Laurel, NJ 08054.

Circle Reader Service Number 229.

ST MIDI Software

Electronic Music Publishing House has announced new software to take advantage of the Atari ST's built-in MIDI (Musical Instrument Digital Interface) ports.

Midiplay turns an ST into a 16-channel digital player/recorder that gives you control over the music's tempo, key, and timbre. It can play prerecorded music through the computer or a MIDI-equipped synthesizer, record music from a MIDI synthesizer, and display the music on the screen as it plays. It can also play music in slow motion—as much as ten times slower without altering the pitch. Depending on available memory, up to 250,000 MIDI notes/events can be recorded, and more than 150,000 can be stored on a single-sided 3½-inch disk. *Midiplay* responds to MIDI START, MIDI STOP, and MIDI CONTINUE commands from a remote MIDI device, and it supports playback looping. Playback time is accurate to 1/1000 second.

The synthesizer section turns the ST into a velocity-sensitive, three-voice, realtime synthesizer with eight envelopes, envelope-release control, vibrato speed/depth controls, and storage/playback of up to 26 programmable sound patches.

Three prerecorded music disks will also be available: *Classics Volume I—Music of Bach, Beethoven, Chopin, Debussy, and Mozart; Classics Volume II—The Music of Amadeus Mozart;* and *Music of the Beatles.* Other music disks are planned for the future.

Midiplay will retail for $49.95. It requires only an Atari ST; a MIDI-equipped synthesizer is optional.
Electronic Music Publishing House, 2210 Wilshire Blvd., Suite 488, Santa Monica, CA 90403.

Circle Reader Service Number 230.

Print Utility For Atari ST

Unison World has introduced an Atari ST version of its bestselling *PrintMaster,* a do-it-yourself print shop that allows easy creation of cards, signs, calendars, banners, and stationery. The program includes over 100 high-resolution graphics and many predefined border designs, type fonts, styles, and layout patterns. Menu-driven operation makes the program very easy to use, even for someone with no programming or drawing skills.

Suggested retail price for the ST version is $39.95. Other versions available include IBM-PC ($59.95), Commodore 64 ($34.95), and Apple II ($39.95). Art Gallery disks with additional graphics are available at additional cost.
Unison World, 3165 Adeline St., Berkeley, CA 94703.

Circle Reader Service Number 231.

More Stickybear Software

Weekly Reader Software has added several new products to its line of educational software featuring the familiar character Stickybear. *Stickybear Math 2* helps children practice multiplication and division ($39.95). *Stickybear BASIC* is a gentle introduction to the BASIC programming language ($39.95). *Stickybear Printer* is a sophisticated, easy-to-use graphic design program ($39.95). And *Stickybear Car Builder* helps familiarize you with all the machanics of car building by letting you design, construct, refine, and test sample automobiles ($39.95).
Weekly Reader Family Software, 245 Long Hill Rd., Middletown, CT 06457.

Circle Reader Service Number 232.

New Casio Keyboards

Casio has introduced several new electronic keyboards. The Model MT-55 ($149.50) is a 44-key mini-keyboard with twelve instrument sounds, twelve auto-rhythms, and auto-chording. This six-note polyphonic instrument has a real-time memory that holds 512 melody notes or can be used to store auto-chording for ease of performance. The Model MT-205 ($199) is a 49-key stereo mini-keyboard with twelve instrument sounds. It features twelve auto-rhythms with intro, fill-in, and ending patterns. Optional DP-1 drum pads can be hooked up for manual play of the PCM drum sound sources. The unit is battery powered. The Model MT-88 ($199) is a 49-key mini-keyboard with twelve instrument sounds, twelve auto-rhythms, and auto-chording. It allows auto-play of songs stored in ROM packs. The keyboard's Chord Guide feature teaches the user to play 3-note fingered chords easily by following lights over the keyboard.
Casio, Inc., 15 Gardner Rd., Fairfield, NJ 07006.

Circle Reader Service Number 233.

Briefly Noted

New products of all kinds were introduced at June's Consumer Electronics Show in Chicago. Here are some highlights:

• **SSI** introduced its latest tactical Civil War game, *Gettysburg: The Turning Point,* for Apple II series, Commodore 64, Atari 400/800/1200, and IBM-PC at $59.95 each. *Strategic Simulations, 1046 N. Rengstorff Ave., Mountain View, CA 94043.*

• **Star Micronics** premiered an upgrade of the popular Gemini 10X printer, the Gemini II. It combines the best features from the earlier model with the ease-of-use found in office printers ($329). *Star Micronics, 200 Park Ave., Suite 3510, New York, NY 10166.*

• **Keypunch Software** is distributing a line of inexpensive entertainment, educational, and productivity software for IBM, Apple, Commodore, and Atari. Prices range from $6.99 to $9.99. *Keypunch Software, 1221 Pioneer Bldg., St. Paul, MN 55101.*

• **Main Street Publishing** offers a similar budget line of packages previously sold by other publishers. Prices range from $4.95 to $9.95. *Main Street Publishing, 611 W. Travelers Trail; Burnsville, MN 55337.*

• **Mastertronic's** latest releases include *Ninja, Elektraglide,* and *Video Poker.* For the Commodore 64 ($9.99). *Mastertronic International, 7311B Grove Rd., Frederick, MD 21701.*

• **BCI** introduced *Mind Over Matter,* which contains four self-help programs: "Lose Weight," "Stop Smoking," "Be Successful," and "Conquer Stress." For IBM-PC, Apple II, Commodore 64, and Atari 8-bit computers ($9.95 each). *BCI Software, P.O. Box 730, Ringwood, NJ 07456.*

• **First Star Software** premiered *Spy vs. Spy III: Arctic Antics* for Commodore 64 and 48K Atari ($29.95) and 64K Apple ($34.95). *First Star Software, 18 E. 41st St., New York, NY 10017.*

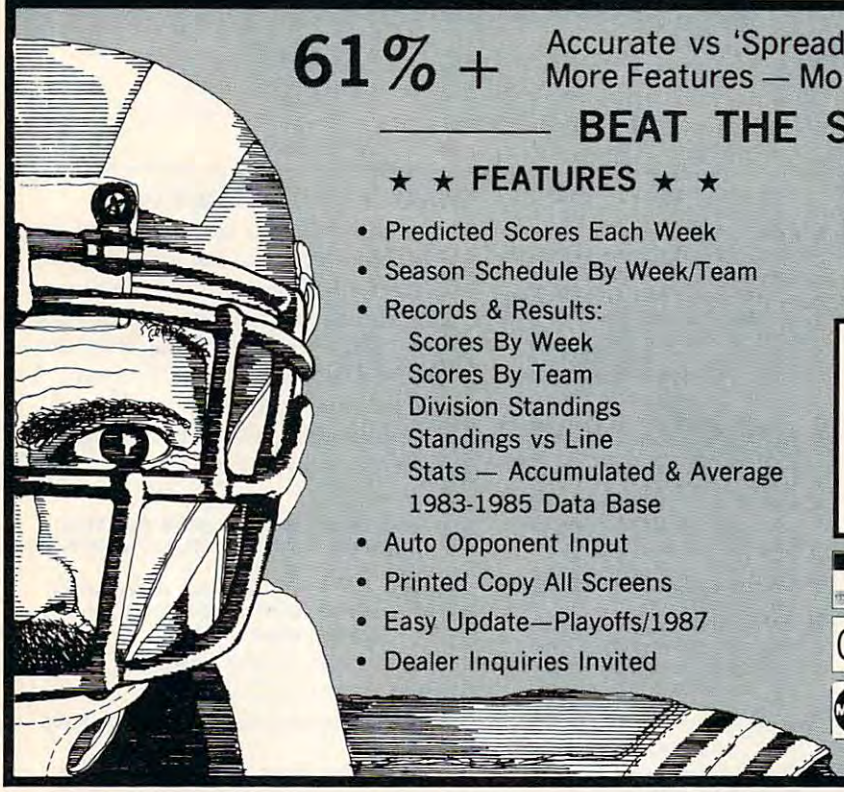
• **Intellicreations** (formerly Datasoft) introduced *Crosscheck,* a combination crossword puzzle/Scrabble game for Atari 8-bit, Commodore 64, Apple II ($29.95), and IBM-PC ($39.95). *Intellicreations, 19808 Nordhoff Pl., Chatsworth, CA 91311.*

• **Sharedata** premiered the Home Companion series, a line of $9.95 programs geared toward home repair and maintenance. *Sharedata, 7122 Shady Oak Rd., Eden Prairie, MN 55344.*

PIK'EM 86

★ ★ ★ ★

A Complete Pro Football Prediction Program For The 1986 NFL SEASON



61% + Accurate vs 'Spread' Since 1981
More Features — More Information In 1986 To .....

BEAT THE SPREAD

★ ★ FEATURES ★ ★

- Predicted Scores Each Week
- Season Schedule By Week/Team
- Records & Results:
  - Scores By Week
  - Scores By Team
  - Division Standings
  - Standings vs Line
  - Stats — Accumulated & Average
  - 1983-1985 Data Base
- Auto Opponent Input
- Printed Copy All Screens
- Easy Update—Playoffs/1987
- Dealer Inquiries Invited

54.95 INCLUDES DISK, DOCUMENTATION SHIPPING & HANDLING

- Apple II — IIc — IIe
- Commodore 64 — 128
- IBM PC, Tandy & Compatibles
- TRS-80 MIII/IV

Stats Needed To Run Program Available In Local Newspapers
Or
We Will Furnish Stats By Mail Or Modem All 20 Weeks — Season Price . . . 40.00

Program Comes Updated Thru Current Week Of Season . . . No Extra Charge

VISA
C.O.D.
MasterCard

ORDERS SHIPPED IN 2 DAYS
PHONE LINES OPEN 24 HRS.

TOLL FREE — 800-722-2277
TEXAS COLL. — 214-586-8212

MARATHON SOFTWARE DEPT. C
P. O. BOX 1349
JACKSONVILLE, TEXAS 75766

TOLL FREE & COLLECT FOR ORDERS ONLY

$5 TALKING DISK

OVER 100 WORDS in vocabularies that you can put into your own programs! No extra hardware required. Sample programs include:

- Talking four-function calculator — choose English, Spanish, or German.
- Talking keyboard — letters and punctuation in English.
- Demonstration of voice editing.

The $5 Talking Disk is available for Commodore 64, 128, Atari 800, 800XL, 130XE, and Apple II+ (64K), IIe, and IIc.

If you want to make your own vocabularies, in any language or accent, then you must have a VOICE MASTER for processing speech input. VOICE MASTER lets you do much more: YOU CAN RECOGNIZE SPOKEN COMMANDS and WRITE MUSIC AS YOU HUM! And affordable too — under $90 including the headset and microphone.

Send $5 for the talking disk. Prepaid orders only. Includes postage. ($7 outside USA, Canada, and Mexico.) Information on VOICE MASTER will accompany your disk. Or you can call or write for VOICE MASTER information separately. Please specify computer make and model with your order.

COVOX INC.
675 Conger St., Dept. C!
Eugene, OR 97402
Telephone (503) 342-1271

PROF JONES
SUMMER/FALL 1986

ORDER LINE
(208) 342-6939

THOROUGHBRED
GREYHOUND
TROTTER
LOTTERY/LOTTO
FOOTBALL
BASKETBALL

OFFICE HOURS
M-F 8 AM - 5 PM
SAT 9 AM - 1 PM
(Mountain Standard Time)

FREE CATALOG

1940 W. STATE
BOISE, IDAHO 85702

VISA
AMEX

PROFESSIONAL HANDICAPPING SYSTEMS

PROFESSIONAL SERIES™ (Tho/Grey/Trot)
The all new Professional Series™ represents the most advanced handicapping software available.

Analysis Module™
Complete bet analysis highlights this basic Professional Series™ module. Full 50 tracks/kennels/etc. $249.95

Factor Value/Multiple Regression Module™
Factor Value Weighting highlights this addition module™ $149.95

Data Base Manager Module™
Automatic storage of last 11 races highlights this module. ($99.95 with Factor Value Module) $149.95

GOLD EDITION™ (Tho/Grey/Trot)
The classic Gold Edition™ from Prof. Jones offers flexibility, results, and ease of use.

Gold Edition™	$159.95
Enhanced Gold Edition™	$199.95
Limited Gold Edition™	$299.95
Ultra Edition™	$399.95

Professor Picks Football™
$99.95; with win/loss power ratings $149.95; Professional Series™ $199.95

Expanded Lottery/Lotto Analysis
Lottery: 3-4 digits $79.95; Lotto: max. of 99 digits $99.95; Enhanced Lottery/ Lotto $129.95

PC-3 Portable Computer (4k)
Choice of Thoroughbred/Greyhound/Trotter Gold Edition™ software. $249.95

Model 100 Portable Computer (32k)
Choice of Thoroughbred/Greyhound/Trotter Gold Edition™ software with Master Bettor™ $649.95

Handicapper's Bulletin Board now available
VHS Training Tapes now available

Terms: Free shipping all software. Add $6.00 COD / $6.00 UPS Blue / $9.00 Out-of-country / ID residents add 5% / 3 weeks personal checks / cash price only add 2% Visa, MC, AMEX. Prices subject to change.

Atari Sound Development System

If you've tried to use the Sound Editor (Program 1, p. 71) from this article in the July issue, you've no doubt discovered that something is missing. The last 53 lines of the program were accidentally omitted. To complete the listing, add the following lines:

```
GC 2840 ? :? :? :POKE 752,0
CG 2850 ? "Enter name for LO
        AD file."
FH 2860 ? " or X to exit."
PG 2870 ? "{3 SPACES}{Q}{22 R}
        {E}"
OH 2880 ? "{3 SPACES}|Dn:fil
        ename.  Extender|"
AF 2890 ? "{3 SPACES}|automa
        tically attached|"
PH 2900 ? "{3 SPACES}{Z}{22 R}
        {C}"
HO 2910 GOSUB 2630:IF FL$="X
        " THEN RETURN
IB 2920 IF PEEK(195)<>170 TH
        EN 2940
AM 2930 ? :? FL$;" does not
        seem to exist...":PO
        KE 752,1:POSITION 0,
        20:? "{8 SPACES}PRES
        S ANY KEY":GET #1,K:
        GOTO 2830
IO 2940 ? "Okay, loading ";F
        L$;"."
MJ 2950 CLOSE #2:OPEN #2,4,0
        ,FL$:GET #2,BYTE
DM 2960 FOR X=0 TO 3:FOR Y=0
         TO 1:GET #2,Z:SD(X,
        Y)=Z:NEXT Y:NEXT X
DB 2970 FOR X=0 TO 3:FOR Y=1
         TO 35:GET #2,Z:S(X,
        Y)=Z:NEXT Y:NEXT X
DD 2980 BYT=BYTE
DC 2990 FOR X=7 TO 0 STEP -1
        :Y=INT(2^X+0.5):IF B
        YT>=Y THEN BYT=BYT-Y
        :BIT(X)=1
FK 3000 NEXT X
KE 3010 RETURN
NB 3020 REM INITIALIZATION
NB 3030 GRAPHICS 0:POKE 82,0
        :POKE 710,0:POKE 752
        ,1:DIM FLL$(20),FL$(
        20),BIT(7),VO(3),STA
        T(3):POKE 559,0
CD 3040 FOR J=0 TO 7:BIT(J)=
        0:NEXT J
GN 3050 FOR J=0 TO 3:VO(J)=8
        :STAT(J)=1:NEXT J
AA 3060 OPEN #1,4,0,"K:"
JN 3070 SOUND 0,50,10,10:FOR
         D=1 TO 5:NEXT D:SOU
        ND 0,0,0,0
DP 3080 DIM SD(3,2),S(3,35),
        G(15,4)
OJ 3090 RESTORE 3100:FOR X=0
         TO 3:FOR Y=0 TO 2:R
        EAD D:SD(X,Y)=D:NEXT
         Y:NEXT X
KB 3100 DATA 243,160,53760,1
        93,160,53762,144,160
        ,53764,121,160,53766
AF 3110 FOR X=0 TO 3:FOR Y=1
         TO 35:S(X,Y)=0:NEXT
         Y:NEXT X
MF 3120 SOUND 0,100,10,10:FO
        R D=1 TO 5:NEXT D:SO
        UND 0,0,0,0
NA 3130 RESTORE 3150:FOR X=1
         TO 15:FOR Y=1 TO 4:
        READ D:G(X,Y)=D:NEXT
         Y:NEXT X
MM 3140 SOUND 0,150,10,10:FO
        R D=1 TO 5:NEXT D:SO
        UND 0,0,0,0
IF 3150 DATA 1,1,1,1,1,1,1,0
        ,1,1,0,1,1,1,0,0,1,0
        ,1,1,1,0,1,0,1,0,0,1
        ,0,1,1,1,0,1,1,0,0,1
        ,0,1,0,0,1,1,1,0,0,0
        ,0,1,0,0
JK 3160 DATA 0,0,1,0,0,0,0,1
AL 3170 GRAPHICS 0:POKE 710,
        0:POKE 709,10:POKE 7
        52,1:POKE 559,0
KO 3180 DL=PEEK(560)+256*PEE
        K(561)
OP 3190 MEMTOP=PEEK(742)
HH 3200 SCREEN1=PEEK(89):SCR
        EEN2=MEMTOP-5
NL 3210 POKE 89,SCREEN2:POKE
         106,SCREEN2+4:POKE
        DL+5,SCREEN2:? CHR$(
        125)
CJ 3220 FOR D=5 TO 20:POSITI
        ON 3,D:? "!!!!!!!!!!
        !!!!!!!!!!!!!!!!!!
        !!!!!":NEXT D
MI 3230 SOUND 0,200,10,10:FO
        R D=1 TO 5:NEXT D:SO
        UND 0,0,0,0
KN 3240 POSITION 1,0:? "ENVE
        LOPE EDITOR"
OL 3250 POSITION 2,1:? "for
        Voice #"
NM 3260 POSITION 2,2:? "Pitc
        h value: "
AI 3270 FOR N=15 TO 10 STEP
        -1:Y=20-N:X=0:POSITI
        ON X,Y:? N:NEXT N
KL 3280 FOR N=9 TO 0 STEP -1
        :Y=20-N:X=0:POSITION
         X,Y:? N:NEXT N
OJ 3290 POSITION 3,22:? "123
        456789012345678901234
        56789012345"
BO 3300 POSITION 20,0:? "L -
         Listen"
EH 3310 POSITION 20,1:? "M -
         Menu"
AJ 3320 POSITION 20,2:? "S -
         Change Sound"
BL 3330 POSITION 20,3:? "C -
         Clear Bars"
NE 3340 SOUND 0,255,10,10:FO
        R D=1 TO 5:NEXT D:SO
        UND 0,0,0,0
HM 3350 POKE 89,SCREEN1:POKE
         106,SCREEN1+4:POKE
        DL+5,SCREEN1:? CHR$(
        125):POKE 559,34
GJ 3360 GOTO 40
```

Minding IBM Memory

The correction in last month's CAPUTE! column is not sufficient to correct all the bugs in the deallocation routine for this article from the June issue (p. 85). The **mov bx, [bp+6]** instructions in Program 2 should instead be **mov bx,[bp+8]**. To make this correction in Program 3, replace lines 100–110 with the following:

```
AH 100 DATA &h55,&h06,&h8b,&hec,
       &h8b,&h5e,&h08,&h8e,&h07,
       &hb4,&h49,&hcd
KF 110 DATA &h21,&h8b,&h5e,&h08,
       &h89,&h07,&h07,&h5d,&hca,
       &h02,&h00
```

The version of the program which appears on the COMPUTE! Disk for April–June includes all corrections.

Hex War For Amiga

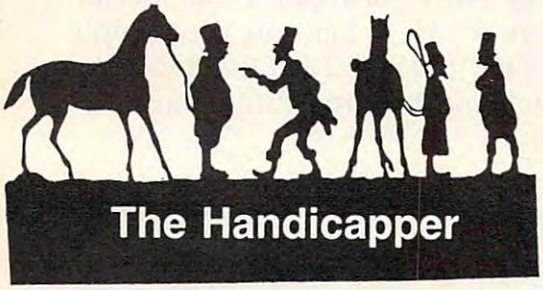
The Amiga version (Program 7, p. 55) of this game in the July issue uses the lowercase letter l as a variable name in several places. Unfortunately, on the printer used to make the listing there is no distinction between l and the numeral 1, so it's difficult to tell when to use the letter and when to use the number. Here are the places where the character should be an l (for clarity, change these to uppercase L): In the lines following the ones with labels 710, 715, and 718, the expressions should be **L = cit(j,1)**, **x = (k−L)*2+19**, and **y = (12−(k+L))*2+3.** Following the DATA statements in the Strengths routine, there is a loop that should use **FOR L=1 TO 5** and **NEXT L.** In the Reveille routine, there is a loop that should use **FOR L=0 to 6, army(k,L,pn) = army(k+1,L,pn), army(k+1,L,pn) = 0,** and **NEXT L.** In the Prisoners subroutine there is a loop that should use **FOR L=0 TO 6** and **army(k,L,j) = 0.** The lines labeled 3480 and 3490 should both start with **L=**, and just below those are two other statements that should read **IF c(1) => L THEN** and **IF c(2) => L THEN.**

Pigskin Predictions! Pro Handicapper

Tired of wrestling with Sunday point spreads? Let your IBM PC or Commodore 64/128 do it for you! Pigskin Predictions, the best-selling NFL handicapper, takes the hassle out of rating National Football League games. Forget about obscure, meaningless statistics. Just spend a few minutes typing in each week's scores and let our point spread software go to work. Here's what Pigskin Predictions has to offer:

- Predicts point spreads for all games--for the current week and the remainder of the season.
- Calculates projected won-lost records for all weeks.
- Menu-driven selection of schedules, ratings, division races, predictions or results by team or week. Seven different reports to screen or printer!
- Maintains home field advantage and power ratings for all NFL teams.
- 1986 Schedule data file included free. Yearly updates available.

Pigskin Predictions is only $39.95 on disk. Versions available for all Commodore 64/128 and IBM/Compatible computers. Get your copy now and be ready for the season!



Use your computer to improve your performance at the track! Separate programs for **Thoroughbreds, harness** horses and **greyhounds** rank the horses or dogs in each race quickly and easily, even if you've never handicapped before!

All the information you need is readily available in the thoroughbred Racing Form, harness or dog track program. We even provide a chart showing you exactly where to get the information you need! Our software puts the savvy of a veteran handicapper at your fingertips. Our complete instructions and wagering guide tell you how to bet, which races to bet and which ones to avoid—one of the real secrets of winning at the track!

Thoroughbred factors include speed, distance, past performance, weight, class, jockey's record, beaten favorite and post position. **Harness factors** include speed, post position, driver's record, breaking tendencies, class, parked-out signs and beaten favorite. **Greyhound factors** include speed, past performance, maneuvering ability, favorite box, class, kennel record, beaten favorite and breaking ability.

Thoroughbred, harness and greyhound programs are sold separately. **IBM/Compatible** and **Apple II** versions are $49.95 each, any two for $74.95, all three $99.95. **Commodore 64/128** and **Tandy Color Computer** versions are $39.95 each on tape or disk. Any two for $59.95, all three $79.95.

VISA **Federal Hill Software** MasterCard
8134 Scotts Level Rd.
Baltimore, MD 21208
Orders 800-628-2828 Ext. 850
Information 301-521-4886

BELOW DEALER COST

MONITORS

FULL COLOR **$99**

RGB

$259

MODEMS

300 BAUD

$28

1200 BAUD

$88

PRINTERS

EPSON

$78

NEAR LETTER QUALITY

$149

DISK DRIVE

1541 COMMODORE

$149

QUICK DELIVERY
1-800-345-5080

PRO-TECH-TRONICS
6870 Shingle Crk. Pkwy
#103
Minneapolis, MN 55430

From the publishers of *COMPUTE!*

August 1986 *COMPUTE!* Disk

All the exciting programs from the past three issues of *COMPUTE!* are on one timesaving, error-free, floppy disk that is ready to load on your Apple II, II+, IIe, and IIc computers. The August 1986 *COMPUTE! Disk* contains the entertaining and useful Apple programs from the June, July, and August 1986 issues of *COMPUTE!*.

The August 1986 *COMPUTE! Disk* costs $12.95 plus $2.00 shipping and handling and is available only from COMPUTE! Publications.

For added savings and convenience, you may also subscribe to the *COMPUTE! Disk.* At a cost of only $39.95 a year (a $12.00 savings), you'll receive four disks, one every three months. Each disk will contain all the programs for your machine from the previous three issues of *COMPUTE!*.

This is an excellent way to build your software library while you enjoy the quality programs from *COMPUTE!*.

Disks and subscriptions are available for Apple, Atari, Commodore 64 and 128, and IBM personal computers. Call for details.

For more information or to order the August 1986 *COMPUTE! Disk,* call toll free 1-800-346-6767 (in NY 212-265-8360) or write *COMPUTE! Disk*, P.O. Box 5038, F.D.R. Station, New York, NY 10150.

COMPUTE! Publications, Inc. abc

Part of ABC Consumer Magazines, Inc.
One of the ABC Publishing Companies
825 7th Avenue, 6th Floor, New York, NY 10019
Publishers of COMPUTE!, COMPUTE!'s Gazette, COMPUTE!'s Gazette Disk, COMPUTE! Books, and COMPUTE!'s Apple Applications

COMPUTE!'s Guide To Typing In Programs

Computers are precise—type the program *exactly* as listed, including necessary punctuation and symbols, except for special characters noted below. We have provided a special listing convention as well as a program to check your typing—"The Automatic Proofreader."

Programs for the IBM, TI-99/4A, and Atari ST models should be typed exactly as listed; no special characters are used. Programs for Commodore, Apple, and Atari 400/800/XL/XE computers may contain some hard-to-read special characters, so we have a listing system that indicates these control characters. You will find these Commodore and Atari characters in curly braces; *do not type the braces.* For example, {CLEAR} or {CLR} instructs you to insert the symbol which clears the screen on the Atari or Commodore machines. A complete list of these symbols is shown in the tables below. For Commodore, Apple, and Atari, a single symbol by itself within curly braces is usually a control key or graphics key. If you see {A}, hold down the CONTROL key and press A. This will produce a reverse video character on the Commodore (in quote mode), a graphics character on the Atari, and an invisible control character on the Apple.

Graphics characters entered with the Commodore logo key are enclosed in a special bracket: [<A>]. In this case, you would hold down the Commodore logo key as you type A. Our Commodore listings are in uppercase, so shifted symbols are underlined. A graphics heart symbol (SHIFT-S) would be listed as S. One exception is {SHIFT-SPACE}. When you see this, hold down SHIFT and press the space bar. If a number precedes a symbol, such as {5 RIGHT}, {6 S}, or [<8 Q>], you would enter five cursor rights, six shifted S's, or eight Commodore-Q's. On the Atari, inverse characters (white on black) should be entered with the inverse video

Atari 400/800/XL/XE

When you see	Type	See	
{CLEAR}	ESC SHIFT <		Clear Screen
{UP}	ESC CTRL -		Cursor Up
{DOWN}	ESC CTRL =		Cursor Down
{LEFT}	ESC CTRL +		Cursor Left
{RIGHT}	ESC CTRL *		Cursor Right
{BACK S}	ESC DELETE		Backspace
{DELETE}	ESC CTRL DELETE		Delete character
{INSERT}	ESC CTRL INSERT		Insert character
{DEL LINE}	ESC SHIFT DELETE		Delete line
{INS LINE}	ESC SHIFT INSERT		Insert line
{TAB}	ESC TAB		TAB key
{CLR TAB}	ESC CTRL TAB		Clear tab
{SET TAB}	ESC SHIFT TAB		Set tab stop
{BELL}	ESC CTRL 2		Ring buzzer
{ESC}	ESC ESC		ESCape key

Commodore PET/CBM/VIC/64/128/16/+4

When You Read:	Press:	See:
{CLR}	SHIFT + CLR/HOME	
{HOME}	CLR/HOME	
{UP}	SHIFT + ↑ CRSR ↓	
{DOWN}	↑ CRSR ↓	
{LEFT}	SHIFT + ← CRSR →	
{RIGHT}	← CRSR →	
{RVS}	CTRL 9	
{OFF}	CTRL 0	
{BLK}	CTRL 1	
{WHT}	CTRL 2	
{RED}	CTRL 3	
{CYN}	CTRL 4	
{PUR}	CTRL 5	
{GRN}	CTRL 6	
{BLU}	CTRL 7	
{YEL}	CTRL 8	

When You Read:	Press:	See:
[<1>]	COMMODORE 1	
[<2>]	COMMODORE 2	
[<3>]	COMMODORE 3	
[<4>]	COMMODORE 4	
[<5>]	COMMODORE 5	
[<6>]	COMMODORE 6	
[<7>]	COMMODORE 7	
[<8>]	COMMODORE 8	
{ F1 }	f1	
{ F2 }	SHIFT f1	
{ F3 }	f3	
{ F4 }	SHIFT f3	
{ F5 }	f5	
{ F6 }	SHIFT f5	
{ F7 }	f7	
{ F8 }	SHIFT f7	
←	←	

key (Atari logo key on 400/800 models).

Whenever more than two spaces appear in a row, they are listed in a special format. For example, {6 SPACES} means press the space bar six times. Our Commodore listings never leave a single space at the end of a line, instead moving it to the next printed line as {SPACE}.

Amiga program listings contain only one special character, the left arrow (←) symbol. This character marks the end of each program line. Wherever you see a left arrow, press RETURN or move the cursor off the line to enter that line into memory. Don't try to type in the left arrow symbol; it's there only as a marker to indicate where each program line ends.

The Automatic Proofreader

Type in the appropriate program listed below, then save it for future use. The Commodore Proofreader works on the Commodore 128, 64, Plus/4, 16, and VIC-20. Don't omit any lines, even if they contain unfamiliar commands or you think they don't apply to your computer. When you run the program, it installs a machine language program in memory and erases its BASIC portion automatically (so be sure to save several copies before running the program for the first time). If you're using a Commodore 128, Plus/4 or 16, do *not* use any GRAPHIC commands while the Proofreader is active. You should disable the Commodore Proofreader before running any other program. To do this, either turn the computer off and on or enter SYS 64738 (for the 64), SYS 65341 (128), SYS 64802 (VIC-20), or SYS 65526 (Plus/4 or 16). To reenable the Proofreader, reload the program and run it as usual. Unlike the original VIC/64 Proofreader, this version works the same with disk or tape.

On the Atari, run the Proofreader to activate it (the Proofreader remains active in memory as a machine language program); you must then enter NEW to erase the BASIC loader. Pressing SYSTEM RESET deactivates the Atari Proofreader; enter PRINT USR(1536) to reenable it.

The Apple Proofreader erases the BASIC portion of itself after you run it, leaving only the machine language portion in memory. It works with either DOS 3.3 or ProDOS. Disable the Apple Proofreader by pressing CTRL-RESET before running another BASIC program.

The IBM Proofreader is a BASIC program that simulates the IBM BASIC line editor, letting you enter, edit, list, save, and load programs that you type. Type RUN to activate. Be sure to leave Caps Lock on, except when typing lowercase characters.

Once the Proofreader is active, try typing in a line. As soon as you press RETURN, either a hexadecimal number (on the Apple) or a pair of letters (on the Commodore, Atari, or IBM) appears. The number or pair of letters is called a *checksum*.

Compare the value displayed on the screen by the Proofreader with the checksum printed in the program listing in the magazine. The checksum is given to the left of each line number. Just type in the program a line at a time (without the printed checksum), press RETURN or Enter, and compare the checksums. If they match, go on to the next line. If not, check your typing; you've made a mistake. Because of the checksum method used, do not type abbreviations, such as ? for PRINT. On the Atari and Apple Proofreaders, spaces are not counted as part of the checksum, so be sure you type the right number of spaces between quote marks. The Atari Proofreader does not check to see that you've typed the characters in the right order, so if characters are transposed, the checksum still matches the listing. The Commodore Proofreader catches transposition errors and ignores spaces unless they're enclosed in quotation marks. The IBM Proofreader detects errors in spacing and transposition.

IBM Proofreader Commands

Since the IBM Proofreader replaces the computer's normal BASIC line editor, it has to include many of the direct-mode IBM BASIC commands. The syntax is identical to IBM BASIC. Commands simulated are LIST, LLIST, NEW, FILES, SAVE, and LOAD. When listing your program, press any key (except Ctrl-Break) to stop the listing. If you enter NEW, the Proofreader prompts you to press Y to be especially sure you mean yes.

Two new commands are BASIC and CHECK. BASIC exits the Proofreader back to IBM BASIC, leaving the Proofreader in memory. CHECK works just like LIST, but shows the checksums along with the listing. After you have typed in a program, save it to disk. Then exit the Proofreader with the BASIC command, and load the program as usual (this replaces the Proofreader in memory). You can now run the program, but you may want to resave it to disk. This will shorten it on disk and make it load faster, but it can no longer be edited with the Proofreader. If you want to convert an existing BASIC program to Proofreader format, save it to disk with SAVE "*filename*",A.

Program 1: Atari Proofreader

By Charles Brannon, Program Editor

```
100 GRAPHICS 0
110 FOR I=1536 TO 1700:REA
    D A:POKE I,A:CK=CK+A:N
    EXT I
120 IF CK<>19072 THEN ? "E
    rror in DATA Statement
    s.  Check Typing.":END

130 A=USR(1536)
140 ? :? "Automatic Proofr
    eader Now Activated."
150 END
160 DATA 104,160,0,185,26,
    3,201,69,240,7
170 DATA 200,200,192,34,20
    8,243,96,200,169,74
180 DATA 153,26,3,200,169,
    6,153,26,3,162
190 DATA 0,189,0,228,157,7
    4,6,232,224,16
200 DATA 208,245,169,93,14
    1,78,6,169,6,141
210 DATA 79,6,24,173,4,228
    ,105,1,141,95
220 DATA 6,173,5,228,105,0
    ,141,96,6,169
230 DATA 0,133,203,96,247,
    238,125,241,93,6
240 DATA 244,241,115,241,1
    24,241,76,205,238
250 DATA 0,0,0,0,0,32,62,2
    46,8,201
260 DATA 155,240,13,201,32
    ,240,7,72,24,101
270 DATA 203,133,203,104,4
    0,96,72,152,72,138
280 DATA 72,160,0,169,128,
    145,88,200,192,40
290 DATA 208,249,165,203,7
    4,74,74,74,24,105
300 DATA 161,160,3,145,88,
    165,203,41,15,24
310 DATA 105,161,200,145,8
    8,169,0,133,203,104
320 DATA 170,104,168,104,4
    0,96
```

Program 2: IBM Proofreader

By Charles Brannon, Program Editor

```
10 'Automatic Proofreader Vers
   ion 3.0 (Lines 205,206 adde
   d/190 deleted/470,490 chang
   ed from V2.0)
100 DIM L$(500),LNUM(500):COLO
    R 0,7,7:KEY OFF:CLS:MAX=0:
    LNUM(0)=65536!
110 ON ERROR GOTO 120:KEY 15,C
    HR$(4)+CHR$(70):ON KEY(15)
     GOSUB 640:KEY (15) ON:GOT
    O 130
120 RESUME 130
130 DEF SEG=&H40:W=PEEK(&H4A)
140 ON ERROR GOTO 650:PRINT:PR
    INT"Proofreader Ready."
150 LINE INPUT L$:Y=CSRLIN-INT
    (LEN(L$)/W)-1:LOCATE Y,1
160 DEF SEG=0:POKE 1050,30:POK
    E 1052,34:POKE 1054,0:POKE
     1055,79:POKE 1056,13:POKE
     1057,28:LINE INPUT L$:DEF
     SEG:IF L$="" THEN 150
170 IF LEFT$(L$,1)=" " THEN L$
    =MID$(L$,2):GOTO 170
```

```
180 IF VAL(LEFT$(L$,2))=0 AND
    MID$(L$,3,1)=" " THEN L$=M
    ID$(L$,4)
200 IF ASC(L$)>57 THEN 260 'no
     line number, therefore co
    mmand
205 BL=INSTR(L$," "):IF BL=0 T
    HEN BL$=L$:GOTO 206 ELSE B
    L$=LEFT$(L$,BL-1)
206 LNUM=VAL(BL$):TEXT$=MID$(L
    $,LEN(STR$(LNUM))+1)
210 IF TEXT$="" THEN GOSUB 540
    :IF LNUM=LNUM(P) THEN GOSU
    B 560:GOTO 150 ELSE 150
220 CKSUM=0:FOR I=1 TO LEN(L$)
    :CKSUM=(CKSUM+ASC(MID$(L$,
    I))*I) AND 255:NEXT:LOCATE
     Y,1:PRINT CHR$(65+CKSUM/1
    6)+CHR$(65+(CKSUM AND 15))
    +" "+L$
230 GOSUB 540:IF LNUM(P)=LNUM
    THEN L$(P)=TEXT$:GOTO 150
    'replace line
240 GOSUB 580:GOTO 150 'insert
     the line
260 TEXT$="":FOR I=1 TO LEN(L$
    ):A=ASC(MID$(L$,I)):TEXT$=
    TEXT$+CHR$(A+32*(A>96 AND
    A<123)):NEXT
270 DELIMITER=INSTR(TEXT$," ")
    :COMMAND$=TEXT$:ARG$="":IF
     DELIMITER THEN COMMAND$=L
    EFT$(TEXT$,DELIMITER-1):AR
    G$=MID$(TEXT$,DELIMITER+1)
     ELSE DELIMITER=INSTR(TEXT
    $,CHR$(34)):IF DELIMITER T
    HEN COMMAND$=LEFT$(TEXT$,D
    ELIMITER-1):ARG$=MID$(TEXT
    $,DELIMITER)
280 IF COMMAND$<>"LIST" THEN 4
    10
290 OPEN "scrn:" FOR OUTPUT AS
     #1
300 IF ARG$="" THEN FIRST=0:P=
    MAX-1:GOTO 340
310 DELIMITER=INSTR(ARG$,"-"):
    IF DELIMITER=0 THEN LNUM=V
    AL(ARG$):GOSUB 540:FIRST=P
    :GOTO 340
320 FIRST=VAL(LEFT$(ARG$,DELIM
    ITER)):LAST=VAL(MID$(ARG$,
    DELIMITER+1))
330 LNUM=FIRST:GOSUB 540:FIRST
    =P:LNUM=LAST:GOSUB 540:IF
    P=0 THEN P=MAX-1
340 FOR X=FIRST TO P:N$=MID$(S
    TR$(LNUM(X)),2)+" "
350 IF CKFLAG=0 THEN A$="":GOT
    O 370
360 CKSUM=0:A$=N$+L$(X):FOR I=
    1 TO LEN(A$):CKSUM=(CKSUM+
    ASC(MID$(A$,I))*I) AND 255
    :NEXT:A$=CHR$(65+CKSUM/16)
    +CHR$(65+(CKSUM AND 15))+"
     "
370 PRINT #1,A$+N$+L$(X)
380 IF INKEY$<>"" THEN X=P
390 NEXT :CLOSE #1:CKFLAG=0
400 GOTO 130
410 IF COMMAND$="LLIST" THEN O
    PEN "lpt1:" FOR OUTPUT AS
    #1:GOTO 300
420 IF COMMAND$="CHECK" THEN C
    KFLAG=1:GOTO 290
430 IF COMMAND$<>"SAVE" THEN 4
    50
440 GOSUB 600:OPEN ARG$ FOR OU
    TPUT AS #1:ARG$="":GOTO 30
    0
450 IF COMMAND$<>"LOAD" THEN 4
    90
460 GOSUB 600:OPEN ARG$ FOR IN
    PUT AS #1:MAX=0:P=0
470 WHILE NOT EOF(1):LINE INPU
    T #1,L$:BL=INSTR(L$," "):B
    L$=LEFT$(L$,BL-1):LNUM(P)=
    VAL(BL$):L$(P)=MID$(L$,LEN
    (STR$(VAL(BL$)))+1):P=P+1:
    WEND
480 MAX=P:CLOSE #1:GOTO 130
490 IF COMMAND$="NEW" THEN INP
    UT "Erase program - Are yo
    u sure";L$:IF LEFT$(L$,1)=
    "y" OR LEFT$(L$,1)="Y" THE
    N MAX=0:LNUM(0)=65536!:GOT
    O 130:ELSE 130
500 IF COMMAND$="BASIC" THEN C
    OLOR 7,0,0:ON ERROR GOTO 0
    :CLS:END
510 IF COMMAND$<>"FILES" THEN
    520
515 IF ARG$="" THEN ARG$="A:"
    ELSE SEL=1:GOSUB 600
517 FILES ARG$:GOTO 130
520 PRINT"Syntax error":GOTO 1
    30
540 P=0:WHILE LNUM>LNUM(P) AND
     P<MAX:P=P+1:WEND:RETURN
560 MAX=MAX-1:FOR X=P TO MAX:L
    NUM(X)=LNUM(X+1):L$(X)=L$(
    X+1):NEXT:RETURN
580 MAX=MAX+1:FOR X=MAX TO P+1
     STEP -1:LNUM(X)=LNUM(X-1)
    :L$(X)=L$(X-1):NEXT:L$(P)=
    TEXT$:LNUM(P)=LNUM:RETURN
600 IF LEFT$(ARG$,1)<>CHR$(34)
     THEN 520 ELSE ARG$=MID$(A
    RG$,2)
610 IF RIGHT$(ARG$,1)=CHR$(34)
     THEN ARG$=LEFT$(ARG$,LEN(
    ARG$)-1)
620 IF SEL=0 AND INSTR(ARG$,".
    ")=0 THEN ARG$=ARG$+".BAS"
630 SEL=0:RETURN
640 CLOSE #1:CKFLAG=0:PRINT"St
    opped.":RETURN 150
650 PRINT "Error #";ERR:RESUME
     150
```

Program 3: Commodore Proofreader

By Philip Nelson, Assistant Editor

```
10 VEC=PEEK(772)+256*PEEK(773)
   :LO=43:HI=44
20 PRINT "AUTOMATIC PROOFREADE
   R FOR ";:IF VEC=42364 THEN
   {SPACE}PRINT "C-64"
30 IF VEC=50556 THEN PRINT "VI
   C-20"
40 IF VEC=35158 THEN GRAPHIC C
   LR:PRINT "PLUS/4 & 16"
50 IF VEC=17165 THEN LO=45:HI=
   46:GRAPHIC CLR:PRINT"128"
60 SA=(PEEK(LO)+256*PEEK(HI))+
   6:ADR=SA
70 FOR J=0 TO 166:READ BYT:POK
   E ADR,BYT:ADR=ADR+1:CHK=CHK
   +BYT:NEXT
80 IF CHK<>20570 THEN PRINT "*
    ERROR* CHECK TYPING IN DATA
    STATEMENTS":END
90 FOR J=1 TO 5:READ RF,LF,HF:
   RS=SA+RF:HB=INT(RS/256):LB=
   RS-(256*HB)
100 CHK=CHK+RF+LF+HF:POKE SA+L
   F,LB:POKE SA+HF,HB:NEXT
110 IF CHK<>22054 THEN PRINT "
   *ERROR* RELOAD PROGRAM AND
   {SPACE}CHECK FINAL LINE":EN
   D
120 POKE SA+149,PEEK(772):POKE
    SA+150,PEEK(773)
130 IF VEC=17165 THEN POKE SA+
   14,22:POKE SA+18,23:POKESA+
   29,224:POKESA+139,224
140 PRINT CHR$(147);CHR$(17);"
   PROOFREADER ACTIVE":SYS SA
150 POKE HI,PEEK(HI)+1:POKE (P
   EEK(LO)+256*PEEK(HI))-1,0:N
   EW
160 DATA 120,169,73,141,4,3,16
   9,3,141,5,3
170 DATA 88,96,165,20,133,167,
   165,21,133,168,169
180 DATA 0,141,0,255,162,31,18
   1,199,157,227,3
190 DATA 202,16,248,169,19,32,
   210,255,169,18,32
200 DATA 210,255,160,0,132,180
   ,132,176,136,230,180
210 DATA 200,185,0,2,240,46,20
   1,34,208,8,72
220 DATA 165,176,73,255,133,17
   6,104,72,201,32,208
230 DATA 7,165,176,208,3,104,2
   08,226,104,166,180
240 DATA 24,165,167,121,0,2,13
   3,167,165,168,105
250 DATA 0,133,168,202,208,239
   ,240,202,165,167,69
260 DATA 168,72,41,15,168,185,
   211,3,32,210,255
270 DATA 104,74,74,74,74,168,1
   85,211,3,32,210
280 DATA 255,162,31,189,227,3,
   149,199,202,16,248
290 DATA 169,146,32,210,255,76
   ,86,137,65,66,67
300 DATA 68,69,70,71,72,74,75,
   77,80,81,82,83,88
310 DATA 13,2,7,167,31,32,151,
   116,117,151,128,129,167,136
   ,137
```

Program 4: Apple Proofreader

By Tim Victor, Editorial Programmer

```
10 C = 0: FOR I = 768 TO 768 +
   68: READ A:C = C + A: POKE I
   ,A: NEXT
20 IF C < > 7258 THEN PRINT "ER
   ROR IN PROOFREADER DATA STAT
   EMENTS": END
30 IF PEEK (190 * 256) < > 76 T
   HEN POKE 56,0: POKE 57,3: CA
   LL 1002: GOTO 50
40 PRINT CHR$ (4);"IN#A$300"
50 POKE 34,0: HOME : POKE 34,1:
    VTAB 2: PRINT "PROOFREADER
   INSTALLED"
60 NEW
100 DATA 216,32,27,253,201,141
110 DATA 208,60,138,72,169,0
120 DATA 72,189,255,1,201,160
130 DATA 240,8,104,10,125,255
140 DATA 1,105,0,72,202,208
150 DATA 238,104,170,41,15,9
160 DATA 48,201,58,144,2,233
170 DATA 57,141,1,4,138,74
180 DATA 74,74,74,41,15,9
190 DATA 48,201,58,144,2,233
200 DATA 57,141,0,4,104,170
210 DATA 169,141,96
```

COMPUTE's Author Guide

Most of the following suggestions serve to improve the speed and accuracy of publication. COMPUTE! is primarily interested in new and timely articles on the Commodore 64/128, Atari, Apple, IBM PC/PCjr, Amiga, and Atari ST. We are much more concerned with the content of an article than with its style, but articles should be clear and well-explained.

The guidelines below will permit your good ideas and programs to be more easily edited and published:

1. The upper left corner of the first page should contain your name, address, telephone number, and the date of submission.

2. The following information should appear in the upper right corner of the first page. If your article is specifically directed to one make of computer, please state the brand name and, if applicable, the BASIC or ROM or DOS version(s) involved. In addition, *please indicate the memory requirements of programs.*

3. The underlined title of the article should start about 2/3 of the way down the first page.

4. Following pages should be typed normally, except that in the upper right corner there should be an abbreviation of the title, your last name, and the page number. For example: Memory Map/Smith/2.

5. All lines within the text of the article must be double- or triple-spaced. A one-inch margin should be left at the right, left, top, and bottom of each page. No words should be divided at the ends of lines. And please do not justify. Leave the lines ragged.

6. Standard typing paper should be used (no erasable, onionskin, or other thin paper) and typing should be on one side of the paper only (upper- and lowercase).

7. Sheets should be attached together with a paper clip. Staples should not be used.

8. If you are submitting more than one article, send each one in a separate mailer with its own tape or disk.

9. Short programs (under 20 lines) can easily be included within the text. Longer programs should be separate listings. *It is essential that we have a copy of the program, recorded twice, on a tape or disk.* If your article was written with a word processor, we also appreciate a copy of the text file on the tape or disk. Please use high-quality 10 or 30 minute tapes with the program recorded on both sides. The tape or disk should be labeled with the author's name, the title of the article, and, if applicable, the BASIC/ROM/DOS version(s). Atari tapes should specify whether they are to be LOADed or ENTERed. We prefer to receive Apple programs on disk rather than tape. Tapes are fairly sturdy, but disks need to be enclosed within plastic or cardboard mailers (available at photography, stationery, or computer supply stores).

10. A good general rule is to spell out the numbers zero through ten in your article and write higher numbers as numerals (1024). The exceptions to this are: Figure 5, Table 3, TAB(4), etc. Within ordinary text, however, the zero through ten should appear as words, not numbers. Also, symbols and abbreviations should not be used within text: use "and" (not &), "reference" (not ref.), "through" (not thru).

11. For greater clarity, use all capitals when referring to keys (RETURN, TAB, ESC, SHIFT), BASIC words (LIST, RND, GOTO), and three languages (BASIC, APL, PILOT). Headlines and subheads should, however, be initial caps only, and emphasized words are not capitalized. If you wish to emphasize, underline the word and it will be italicized during typesetting.

12. Articles can be of any length—from a single-line routine to a multi-issue series. The average article is about four to eight double-spaced, typed pages.

13. If you want to include photographs, they should be either 5×7 black and white glossies or color slides.

14. We do not consider articles which are submitted simultaneously to other publishers. If you wish to send an article to another magazine for consideration, please do not submit it to us.

15. COMPUTE! pays between $70 and $800 for published articles. In general, the rate reflects the length and quality of the article. Payment is made upon acceptance. Following submission (Editorial Department, COMPUTE! Magazine, P.O. Box 5406, Greensboro, NC 27403) it will take from four to eight weeks for us to reply. If your work is accepted, you will be notified by a letter which will include a contract for you to sign and return. *Rejected manuscripts are returned to authors who enclose a self-addressed, stamped envelope.*

16. If your article is accepted and you have since made improvements to the program, please submit an entirely new tape or disk and a new copy of the article reflecting the update. We cannot easily make revisions to programs and articles. It is necessary that you send the revised version as if it were a new submission entirely, but be sure to indicate that your submission is a revised version by writing, "Revision" on the envelope and the article.

17. COMPUTE! does not accept unsolicited product reviews. If you are interested in serving on our panel of reviewers, contact the Review Coordinator for details.

MLX Machine Language Entry Program For Commodore 64

Ottis Cowper, Technical Editor

"MLX" is a labor-saving utility that allows will help you enter machine language program listings without error. MLX is required to enter all Commodore 64 machine language programs published in COMPUTE!.

Type in and save some copies of MLX (you'll want to use it to enter future ML programs from COMPUTE!, COMPUTE!'s GAZETTE, and COMPUTE! books). When you're ready to enter an ML program, load and run MLX. You'll be asked for a starting address and an ending address. These addresses should appear in the article accompanying the MLX-format program listing you're typing.

If you're unfamiliar with machine language, the addresses (and all other values you enter in MLX) may appear strange. Instead of the usual decimal numbers you're accustomed to, these numbers are in *hexadecimal*—a base 16 numbering system commonly used by ML programmers. Hexadecimal—hex for short—includes the numerals 0–9 and the letters A–F. But don't worry—even if you know nothing about ML or hex, you should have no trouble using MLX.

After you enter the starting and ending addresses, you'll be offered the option of clearing the workspace. The data you enter with MLX is kept in a special reserved area of memory; clearing this workspace fills the reserved area with zeros, which makes it easier to find where you left off typing if you enter the listing in several sessions. Choose this option if you're starting to enter a new listing. If you're continuing a listing that's partially typed from a previous session, there's no point in clearing the workspace, since the data you load in will fill the area with whatever values were in workspace memory at the time of the last Save.

At this point, functions menu will appear. If you're just starting to type in a program, pick the first option, ENTER DATA, by pressing the E key. You'll be asked for an address; type the four-digit number at the start of the first line of the program listing. If you've already typed in part of a program, be sure to load the partially completed program before you resume entry, then choose the ENTER DATA option and type the line number where you left off typing at the end of the previous session. In any case, make sure the address you enter corresponds to the address of a line in the listing. Otherwise, you'll be unable to enter the data correctly. If you pressed E by mistake, you can return to the command menu by pressing RETURN alone when asked for the address. (You can get back to the menu from most options by pressing RETURN with no other input.)

Entering A Listing

Once you're in Enter mode, MLX prints the address for each program line for you. You then type in all nine numbers on that line, beginning with the first two-digit number after the colon (:). Each line represents eight data bytes and a checksum. Although an MLX-format listing appears similar to the "hex dump" listings from a machine language monitor program, the extra checksum number on the end allows MLX to check your typing.

When you enter a line, MLX recalculates the checksum from the eight bytes and the address and compares this value to the number from the ninth column. If the values match, you'll hear a bell tone, the data will be added to the workspace area, and the prompt for the next line of data will appear. But if MLX detects a typing error, you'll hear a low buzz and see an error message. The line will then be redisplayed for editing.

Invalid Characters Banned

Only a few keys are active while you're entering data, so you may have to unlearn some habits. You *do not* type spaces between the columns; MLX automatically inserts these for you. You *do not* press RETURN after typing the last number in a line; MLX automatically enters and checks the line after you type the last digit.

Only the numerals 0–9 and the letters A–F can be typed in. If you press any other key (with some exceptions noted below), you'll hear a warning buzz. MLX checks for transposed characters. If you're supposed to type in A0 and instead enter 0A, MLX will catch your mistake. There is one error that can slip past MLX: Because of the checksum formula used, MLX won't notice if you accidentally type FF in place of 00, and vice versa. And there's a very slim chance that you could garble a line and still end up with a combination of characters that adds up to the proper checksum. However, these mistakes should not occur if you take reasonable care while entering data.

Editing Features

To correct typing mistakes before finishing a line, use the INST/DEL key to delete the character to the left of the cursor. (The cursor-left key also deletes.) If you mess up a line really badly, press CLR/HOME to start the line over. The RETURN key is also active, but only before any data is typed on a line. Pressing RETURN at this point returns you to the command menu. After you type a character of data, MLX disables RETURN until the cursor returns to the start of a line. Remember, you can press CLR/HOME to quickly get to a line number prompt.

More editing features are available when correcting lines in which MLX has detected an error. To make corrections in a line that MLX has redisplayed for editing, compare the line on the screen with the one printed in the listing, then move the cursor to the mistake and type the correct key. The cursor left and right keys provide the normal cursor controls. (The INST/DEL key now works as an alternative cursor-left key.) You cannot move left beyond the first character in the line. If you try to move beyond the rightmost character, you'll reenter the line. During editing, RETURN is active; pressing it tells MLX to recheck the line. You can press the CLR/HOME key to clear the entire line if you want to start from scratch, or if you want to get to a line number prompt to use RETURN to get back to the menu.

Display Data

The second menu choice, DISPLAY DATA, examines memory and shows the contents in the same format as the program listing (including the checksum). When you press D, MLX asks you for a starting address. Be sure that the starting address you give corresponds to a line number in the listing. Otherwise, the checksum display will be meaningless. MLX displays program lines until it reaches the end of the program, at which point the menu is redisplayed. You can pause the display by pressing the space bar. (MLX finishes printing the current line before halting.) Press space again to restart the display. To break out of the display and

get back to the menu before the ending address is reached, press RETURN.

Other Menu Options

Two more menu selections let you save programs and load them back into the computer. These are SAVE FILE and LOAD FILE; their operation is quite straightforward. When you press S or L, MLX asks you for the filename. You'll then be asked to press either D or T to select disk or tape.

You'll notice the disk drive starting and stopping several times during a load or save. Don't panic; this is normal behavior. MLX opens and reads from or writes to the file instead of using the usual LOAD and SAVE commands. Disk users should also note that the drive prefix 0: is automatically added to the filename (line 750), so this should *not* be included when entering the name. This also precludes the use of @ for Save-with-Replace, so remember to give each version you save a different name.

Remember that MLX saves the entire workspace area from the starting address to the ending address, so the save or load may take longer than you might expect if you've entered only a small amount of data from a long listing. When saving a partially completed listing, make sure to note the address where you stopped typing so you'll know where to resume entry when you reload.

MLX reports the standard disk or tape error messages if any problems are detected during the save or load. (Tape users should bear in mind that Commodore computers are never able to detect errors during a save to tape.) MLX also has three special load error messages: INCORRECT STARTING ADDRESS, which means the file you're trying to load does not have the starting address you specified when you ran MLX; LOAD ENDED AT *address*, which means the file you're trying to load ends before the ending address you specified when you started MLX; and TRUNCATED AT ENDING ADDRESS, which means the file you're trying to load extends beyond the ending address you specified when you started MLX. If you see one of these messages and feel certain that you've loaded the right file, exit and rerun MLX, being careful to enter the correct starting and ending addresses.

The QUIT menu option has the obvious effect—it stops MLX and enters BASIC. The RUN/STOP key is disabled, so the Q option lets you exit the program without turning off the computer. (Of course, RUN/STOP–RESTORE also gets you out.) You'll be asked for verification; press Y to exit to BASIC, or any other key to return to the menu. After quitting, you can type RUN again and reenter MLX without losing your data, as long as you don't use the clear workspace option.

The Finished Product

When you've finished typing all the data for an ML program and saved your work, you're ready to see the results. The instructions for loading and using the finished product vary from program to program. Some ML programs are designed to be loaded and run like BASIC programs, so all you need to type is LOAD *"filename"*,8 for disk or LOAD *"filename"* for tape, and then RUN. Such programs will usually have a starting address of 0801. Other programs must be reloaded to specific addresses with a command such as LOAD *"filename"*,8,1 for disk or LOAD *"filename"*,1,1 for tape, then started with a SYS to a particular memory address. The most common starting address for such programs is 49152, which corresponds to MLX address C000. In either case, you should always refer to the article which accompanies the ML listing for information on loading and running the program.

An Ounce Of Prevention

By the time you finish typing in the data for a long ML program, you may have several hours invested in the project. Don't take chances—use our "Automatic Proofreader" to type MLX, and then test your copy *thoroughly* before first using it to enter any significant amount of data. Make sure all the menu options work as they should. Enter fragments of the program starting at several different addresses, then use the Display option to verify that the data has been entered correctly. And be sure to test the Save and Load options several times to ensure that you can recall your work from disk or tape. Don't let a simple typing error in MLX cost you several nights of hard work.

MLX

For instructions on entering this listing, please refer to "COMPUTE!'s Guide to Typing In Programs" in this issue of COMPUTE!.

```
EK 100 POKE 56,50:CLR:DIM IN$,
       I,J,A,B,A$,B$,A(7),N$
DM 110 C4=48:C6=16:C7=7:Z2=2:Z
       4=254:Z5=255:Z6=256:Z7=
       127
CJ 120 FA=PEEK(45)+Z6*PEEK(46)
       :BS=PEEK(55)+Z6*PEEK(56
       ):H$="0123456789ABCDEF"
SB 130 R$=CHR$(13):L$="{LEFT}"
       :S$=" ":D$=CHR$(20):Z$=
       CHR$(0):T$="{13 RIGHT}"
CQ 140 SD=54272:FOR I=SD TO SD
       +23:POKE I,0:NEXT:POKE
       {SPACE}SD+24,15:POKE 78
       8,52
FC 150 PRINT"{CLR}"CHR$(142)CH
       R$(8):POKE 53280,15:POK
       E 53281,15
EJ 160 PRINT T$" {RED}{RVS}
       {2 SPACES}{8 @}
       {2 SPACES}"SPC(28)"
       {2 SPACES}{OFF}{BLU} ML
       X II {RED}{RVS}
       {2 SPACES}"SPC(28)"
       {12 SPACES}{BLU}"
FR 170 PRINT"{3 DOWN}
       {3 SPACES}COMPUTE!'S MA
       CHINE LANGUAGE EDITOR
       {3 DOWN}"
JB 180 PRINT"{BLK}STARTING ADD
       RESS{4}";:GOSUB300:SA=A
       D:GOSUB1040:IF F THEN18
       0
GF 190 PRINT"{BLK}{2 SPACES}EN
       DING ADDRESS{4}";:GOSUB
       300:EA=AD:GOSUB1030:IF
       {SPACE}F THEN190
KR 200 INPUT"{3 DOWN}{BLK}CLEA
       R WORKSPACE [Y/N]{4}";A
       $:IF LEFT$(A$,1)<>"Y"TH
       EN220
PG 210 PRINT"{2 DOWN}{BLU}WORK
       ING...";:FORI=BS TO BS+
       EA-SA+7:POKE I,0:NEXT:P
       RINT"DONE"
DR 220 PRINTTAB(10)"{2 DOWN}
       {BLK}{RVS} MLX COMMAND
       {SPACE}MENU {DOWN}{4}":
       PRINT T$"{RVS}E{OFF}NTE
       R DATA"
BD 230 PRINT T$"{RVS}D{OFF}ISP
       LAY DATA":PRINT T$"
       {RVS}L{OFF}OAD FILE"
JS 240 PRINT T$"{RVS}S{OFF}AVE
        FILE":PRINT T$"{RVS}Q
       {OFF}UIT{2 DOWN}{BLK}"
JH 250 GET A$:IF A$=N$ THEN250
HK 260 A=0:FOR I=1 TO 5:IF A$=
       MID$("EDLSQ",I,1)THEN A
       =I:I=5
FD 270 NEXT:ON A GOTO420,610,6
       90,700,280:GOSUB1060:GO
       TO250
EJ 280 PRINT"{RVS} QUIT ":INPU
       T"{DOWN}{4}ARE YOU SURE
        [Y/N]";A$:IF LEFT$(A$,
       1)<>"Y"THEN220
EM 290 POKE SD+24,0:END
JX 300 IN$=N$:AD=0:INPUTIN$:IF
       LEN(IN$)<>4THENRETURN
KF 310 B$=IN$:GOSUB320:AD=A:B$
       =MID$(IN$,3):GOSUB320:A
       D=AD*256+A:RETURN
PP 320 A=0:FOR J=1 TO 2:A$=MID
       $(B$,J,1):B=ASC(A$)-C4+
       (A$>"@")*C7:A=A*C6+B
JA 330 IF B<0 OR B>15 THEN AD=
       0:A=-1:J=2
GX 340 NEXT:RETURN
CH 350 B=INT(A/C6):PRINT MID$(
       H$,B+1,1);:B=A-B*C6:PRI
       NT MID$(H$,B+1,1);:RETU
       RN
RR 360 A=INT(AD/Z6):GOSUB350:A
       =AD-A*Z6:GOSUB350:PRINT
       ":";
BE 370 CK=INT(AD/Z6):CK=AD-Z4*
       CK+Z5*(CK>Z7):GOTO390
PX 380 CK=CK*Z2+Z5*(CK>Z7)+A
JC 390 CK=CK+Z5*(CK>Z5):RETURN
QS 400 PRINT"{DOWN}STARTING AT
       {4}";:GOSUB300:IF IN$<>
       N$ THEN GOSUB1030:IF F
       {SPACE}THEN400
EX 410 RETURN
HD 420 PRINT"{RVS} ENTER DATA
       {SPACE}":GOSUB400:IF IN
       $=N$ THEN220
```

```
JK 430 OPEN3,3:PRINT
SK 440 POKE198,0:GOSUB360:IF F
       THEN PRINT IN$:PRINT"
       {UP}{5 RIGHT}";
GC 450 FOR I=0 TO 24 STEP 3:B$
       =S$:FOR J=1 TO 2:IF F T
       HEN B$=MID$(IN$,I+J,1)
HA 460 PRINT"{RVS}"B$L$;:IF I<
       24THEN PRINT"{OFF}";
HD 470 GET A$:IF A$=N$ THEN470
FK 480 IF(A$>"/"ANDA$<":")OR(A
       $>"@"ANDA$<"G")THEN540
MP 490 IF A$=R$ AND((I=0)AND(J
       =1)OR F)THEN PRINT B$;:
       J=2:NEXT:I=24:GOTO550
KC 500 IF A$="{HOME}" THEN PRI
       NT B$:J=2:NEXT:I=24:NEX
       T:F=0:GOTO440
MX 510 IF(A$="{RIGHT}")ANDF TH
       ENPRINT B$L$;:GOTO540
GK 520 IF A$<>L$ AND A$<>D$ OR
       ((I=0)AND(J=1))THEN GOS
       UB1060:GOTO470
HG 530 A$=L$+S$+L$:PRINT B$L$;
       :J=2-J:IF J THEN PRINT
       {SPACE}L$;:I=I-3
QS 540 PRINT A$;:NEXT J:PRINT
       {SPACE}S$;
PM 550 NEXT I:PRINT:PRINT"{UP}
       {5 RIGHT}";:INPUT#3,IN$
       :IF IN$=N$ THEN CLOSE3:
       GOTO220
QC 560 FOR I=1 TO 25 STEP3:B$=
       MID$(IN$,I):GOSUB320:IF
        I<25 THEN GOSUB380:A(I
       /3)=A
PK 570 NEXT:IF A<>CK THEN GOSU
       B1060:PRINT"{BLK}{RVS}
       {SPACE}ERROR: REENTER L
       INE [43]":F=1:GOTO440
HJ 580 GOSUB1080:B=BS+AD-SA:FO
       R I=0 TO 7:POKE B+I,A(I
       ):NEXT
QQ 590 AD=AD+8:IF AD>EA THEN C
       LOSE3:PRINT"{DOWN}{BLU}
       ** END OF ENTRY **{BLK}
       {2 DOWN}":GOTO700
GQ 600 F=0:GOTO440
QA 610 PRINT"{CLR}{DOWN}{RVS}
       {SPACE}DISPLAY DATA ":G
       OSUB400:IF IN$=N$ THEN2
       20
RJ 620 PRINT"{DOWN}{BLU}PRESS:
        {RVS}SPACE{OFF} TO PAU
       SE, {RVS}RETURN{OFF} TO
        BREAK[43]{DOWN}"
KS 630 GOSUB360:B=BS+AD-SA:FOR
       I=BTO B+7:A=PEEK(I):GOS
       UB350:GOSUB380:PRINT S$
       ;
CC 640 NEXT:PRINT"{RVS}";:A=CK
       :GOSUB350:PRINT
KH 650 F=1:AD=AD+8:IF AD>EA TH
       ENPRINT"{DOWN}{BLU}** E
       ND OF DATA **":GOTO220
KC 660 GET A$:IF A$=R$ THEN GO
       SUB1080:GOTO220
EQ 670 IF A$=S$ THEN F=F+1:GOS
       UB1080
AD 680 ONFGOTO630,660,630
CM 690 PRINT"{DOWN}{RVS} LOAD
       {SPACE}DATA ":OP=1:GOTO
       710
PC 700 PRINT"{DOWN}{RVS} SAVE
       {SPACE}FILE ":OP=0
RX 710 IN$=N$:INPUT"{DOWN}FILE
       NAME[43]";IN$:IF IN$=N$
       {SPACE}THEN220
PR 720 F=0:PRINT"{DOWN}{BLK}
       {RVS}T{OFF}APE OR {RVS}
       D{OFF}ISK: [43]";
FP 730 GET A$:IF A$="T"THEN PR
       INT"T{DOWN}":GOTO880
HQ 740 IF A$<>"D"THEN730
HH 750 PRINT"D{DOWN}":OPEN15,8
       ,15,"I0:":B=EA-SA:IN$="
       0:"+IN$:IF OP THEN810
SQ 760 OPEN 1,8,8,IN$+",P,W":G
       OSUB860:IF A THEN220
FJ 770 AH=INT(SA/256):AL=SA-(A
       H*256):PRINT#1,CHR$(AL)
       ;CHR$(AH);
PE 780 FOR I=0 TO B:PRINT#1,CH
       R$(PEEK(BS+I));:IF ST T
       HEN800
FC 790 NEXT:CLOSE1:CLOSE15:GOT
       O940
GS 800 GOSUB1060:PRINT"{DOWN}
       {BLK}ERROR DURING SAVE:
       [43]":GOSUB860:GOTO220
MA 810 OPEN 1,8,8,IN$+",P,R":G
       OSUB860:IF A THEN220
GE 820 GET#1,A$,B$:AD=ASC(A$+Z
       $)+256*ASC(B$+Z$):IF AD
       <>SA THEN F=1:GOTO850
RX 830 FOR I=0 TO B:GET#1,A$:P
       OKE BS+I,ASC(A$+Z$):IF(
       I<>B)AND ST THEN F=2:AD
       =I:I=B
FA 840 NEXT:IF ST<>64 THEN F=3
FQ 850 CLOSE1:CLOSE15:ON ABS(F
       >0)+1 GOTO960,970
SA 860 INPUT#15,A,A$:IF A THEN
        CLOSE1:CLOSE15:GOSUB10
       60:PRINT"{RVS}ERROR: "A
       $
GQ 870 RETURN
EJ 880 POKE183,PEEK(FA+2):POKE
       187,PEEK(FA+3):POKE188,
       PEEK(FA+4):IFOP=0THEN92
       0
HJ 890 SYS 63466:IF(PEEK(783)A
       ND1)THEN GOSUB1060:PRIN
       T"{DOWN}{RVS} FILE NOT
       {SPACE}FOUND ":GOTO690
CS 900 AD=PEEK(829)+256*PEEK(8
       30):IF AD<>SA THEN F=1:
       GOTO970
SC 910 A=PEEK(831)+256*PEEK(83
       2)-1:F=F-2*(A<EA)-3*(A>
       EA):AD=A-AD:GOTO930
KM 920 A=SA:B=EA+1:GOSUB1010:P
       OKE780,3:SYS 63338
JF 930 A=BS:B=BS+(EA-SA)+1:GOS
       UB1010:ON OP GOTO950:SY
       S 63591
AE 940 GOSUB1080:PRINT"{BLU}**
        SAVE COMPLETED **":GOT
       O220
XP 950 POKE147,0:SYS 63562:IF
       {SPACE}ST>0 THEN970
FR 960 GOSUB1080:PRINT"{BLU}**
        LOAD COMPLETED **":GOT
       O220
DP 970 GOSUB1060:PRINT"{BLK}
       {RVS}ERROR DURING LOAD:
       {DOWN}[43]":ON F GOSUB98
       0,990,1000:GOTO220
PP 980 PRINT"INCORRECT STARTIN
       G ADDRESS (";:GOSUB360:
       PRINT")":RETURN
GR 990 PRINT"LOAD ENDED AT ";:
       AD=SA+AD:GOSUB360:PRINT
        D$:RETURN
FD 1000 PRINT"TRUNCATED AT END
        ING ADDRESS":RETURN
RX 1010 AH=INT(A/256):AL=A-(AH
       *256):POKE193,AL:POKE1
       94,AH
FF 1020 AH=INT(B/256):AL=B-(AH
       *256):POKE174,AL:POKE1
       75,AH:RETURN
FX 1030 IF AD<SA OR AD>EA THEN
       1050
HA 1040 IF(AD>511 AND AD<40960
       )OR(AD>49151 AND AD<53
       248)THEN GOSUB1080:F=0
       :RETURN
HC 1050 GOSUB1060:PRINT"{RVS}
       {SPACE}INVALID ADDRESS
        {DOWN}{BLK}":F=1:RETU
       RN
AR 1060 POKE SD+5,31:POKE SD+6
       ,208:POKE SD,240:POKE
       {SPACE}SD+1,4:POKE SD+
       4,33
DX 1070 FOR S=1 TO 100:NEXT:GO
       TO1090
PF 1080 POKE SD+5,8:POKE SD+6,
       240:POKE SD,0:POKE SD+
       1,90:POKE SD+4,17
AC 1090 FOR S=1 TO 100:NEXT:PO
       KE SD+4,0:POKE SD,0:PO
       KE SD+1,0:RETURN
```

©

COMPUTE! Subscriber Services

Please help us serve you better. If you need to contact us for any of the reasons listed below, write to us at:

COMPUTE! Magazine
P.O. Box 10954
Des Moines, IA 50340

or call the Toll Free number listed below.

Change Of Address. Please allow us 6–8 weeks to effect the change; send your current mailing label along with your new address.

Renewal. Should you wish to renew your **COMPUTE!** subscription before we remind you to, send your current mailing label with payment or charge number or call the Toll Free number listed below.

New Subscription. A one year (12 month) US subscription to **COMPUTE!** is $24.00 (2 years, $45.00; 3 years, $65.00. For subscription rates outside the US, see staff page). Send us your name and address or call the Toll Free number listed below.

Delivery Problems. If you receive duplicate issues of **COMPUTE!**, if you experience late delivery or if you have problems with your subscription, please call the Toll Free number listed below.

COMPUTE!
1-800-247-5470
In IA 1-800-532-1272

Classified

SOFTWARE

TANDY 1000 PROGRAMS AND NEWSLETTER Send for free information on educational & entertainment programs & newsletter. Soda Pop Software, POB 653, Kenosha, WI 53141

FREE SOFTWARE for C64, C128, IBM, & CPM. Send SASE for info (specify computer) to: PUBLIC DOMAIN USERS GROUP PO Box 1442-A1, Orange Park, FL 32067

COMMODORE: TRY BEFORE YOU BUY. Top 25 best-selling games, utilities, new releases. Visa, MasterCard. Free brochure. Rent-A-Disk, 908 9th Ave., Huntington, WV 25701 (304) 522-1665

DISCOUNT SOFTWARE: Amiga/Apple/Atari/ C64-128/IBM PC-PCjr/TRS-80/Timex/Sinclair. Free Catalog: WMJ DATA SYSTEMS, 4 Butterfly Dr., Hauppauge, NY 11788

Genealogy for the 64/128. **Family Tree** produces Pedigree Charts, Fam. Group Records, Individ. Files, Indexes, Searches of Ancestors. LDS vers. avail.-$49.95. Genealogy Softw., POB 1151, Port Huron, MI 48061, (519) 344-3990

BRIDGE GAME PROGRAM $39.95 demo disk $5. 1 to 4 players for IBM, Apple, TI99, TRS80, C64, 128, +4, 16, VIC. Authors John & Lynda Allan, Box 313, Azilda, Ontario, Canada P0M 1B0

COMPUTER CASINO! 24 GAMES! BLACKJACK, 21, DRAW POKER, SLOTS, more! C64/128 Disk. $12.95: M. E. ADAMS, 6547 N. Acad. #446-A2, Co. Springs, CO 80907

AMIGA OR ATARI ST SOFTWARE. 20 Different disks $99, 10 disks $59, or $6.95 each. 30 C64 programs + catalog $2. Info-SASE FRUGAL-WARE, 23 E. Green St., W. Hazelton, PA 18201

Amiga and C64 Public Domain Software. For a list of available programs send a self-addressed, stamped envelope to MCA, P.O Box 5533, Katy, TX 774491-5533

FREE APPLE SOFTWARE
Over 1000 Public Domain programs on 50 diskettes, $5 ea. + $1 s/h per order. Send $1 for catalog (refundable). C&H Enterprizes, Box 29243, Memphis, TN 38127

FREE SOFTWARE CATALOG!
Call Toll-Free 1-800-554-1162, Tevex, Inc. Save 1/3 off retail prices. We carry SSI, Elect. Arts, Infocom and many more!

SELL YOUR PROGRAMS to Software Publishers. Software for the Apple, Atari, Commodore, IBM, TI, TRS-80, Zenith, Osborne, Kaypro, Others. Directory $5.95. I.W., Box 40581, Pas., CA 91104

ATARI 8 BIT: FULL FUNCTION BUSINESS DBMS. Any drive(s)/upgrades. GL/AR/AP/Inv/ Mail/W/P. 9000 records/disk. MICROMOD, 1635-A Holden, Orlando, FL 32809 (305) 857-6014

FREE EVERYTHING BOOKS for TI99/4A and C64/C128! Our "Everything Books" are packed full of the newest software, hardware, books, accessories and much more! The prices are low, the service is dependable, and you can order toll-free. Request your "Everything Book" today. Call 1-800-348-2778 (219-259-7051 in Indiana) or write: TENEX Computer Express, P.O. Box 6578, South Bend, Indiana 46660. (K1A)

MISCELLANEOUS

IBM PCjr REPORT: THE NATIONAL NEWS-LETTER PCjr-specific articles, reviews, Public Domain from across the nation. $18/yr. PCjr CLUB, POB 95067, Schaumburg, IL 60195

The Bard's Tale Hint book, maps, magic items, uses, for Apple or C64 for $12.50 Ck or MO. Send to Don Dannelley, 2914 Pennsylvania, Wichita Falls, TX 76309

COMPUTE! Classified is a low-cost way to tell over 350,000 microcomputer owners about your product or service.

Rates: $25 per line, minimum of four lines. Any or all of the first line set in capital letters at no charge. Add $15 per line for boldface words, or $50 for the entire ad set in boldface (any number of lines.) Inquire about display rates.

Terms: Prepayment is required. Check, money order, American Express, Visa, or MasterCard is accepted. Make checks payable to COMPUTE! Publications.

Form: Ads are subject to publisher's approval and must be either typed or legibly printed. One line equals 40 letters and spaces between words. Please underline words to be set in boldface.

General Information: Advertisers using post office box numbers in their ads must supply permanent address and telephone numbers. Ad will appear in next available issue after receipt.

Closing: 20th of the third month preceding cover date (e.g., June issue closes March 20th). Send order and remittance to: Harry Blair, Classified Manager, COMPUTE!, P.O. Box 5406, Greensboro, NC 27403. To place an ad by phone, call Harry Blair at (919) 275-9809.

Notice: COMPUTE! Publications cannot be responsible for offers or claims of advertisers, but will attempt to screen out misleading or questionable copy.

Advertisers Index

GET UP TO 200 FUN-FILLED PROGRAMS EACH YEAR— WHEN YOU SUBSCRIBE NOW TO COMPUTE!

Yes, I want to subscribe to COMPUTE! and receive up to 20 all-new programs in each issue—up to 200 fun-filled programs each year.

☐ 1 Year/**$24.** (You save 33% off the newsstand price.)

Name _____ Apt. _____

Address _____

City, State, Zip _____

☐ Payment Enclosed ☐ Bill Me Later ☐ Check Here If Renewal

PLEASE INDICATE THE COMPUTER YOU USE: APPLE ☐ 01
ATARI ☐ 02 COMMODORE ☐ 03 IBM ☐ 05 OTHER ☐ _____

For Foreign & Canadian Subscribers, please add $6 (U.S.) per year postage.
Offer Expires 12/31/86.

S

I1676

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 7478 DES MOINES, IOWA

POSTAGE WILL BE PAID BY ADDRESSEE

COMPUTE!

PO BOX 10954
DES MOINES, IOWA 50347

COMPUTE!'s Disk

SPECIAL INTRODUCTORY OFFER

YES! I want to save time and money. Please enter my quarterly subscription to the following COMPUTE! Disk:

☐ Commodore ☐ Apple ☐ Atari ☐ IBM

☐ Save 33% off the single issue price. 1 year subscription, $39.95

☐ Save even more! 2 year subscription, $69.95

All *Disk* orders must be prepaid.

☐ Payment enclosed (check or money order)
☐ Charge ☐ MasterCard ☐ Visa

Acct. No. _____ Exp. Date _____

Signature _____

Name _____

Address _____

City _____ State _____ Zip _____

(Outside U.S. and Canada, add $9.00 per year for shipping and handling.)

Save 33%

S

JT825

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 7551 DES MOINES, IA

POSTAGE WILL BE PAID BY ADDRESSEE

COMPUTE!'s DISK
P.O. BOX 10036
DES MOINES, IA 50347

WE TOP APPLE AND COMMODORE BEAUTIFULLY

Thomson offers a whole new world of graphics capabilities for your Commodore or Apple IIC or IIE computers. And Thomson offers you more than the competition.

Thomson monitors offer these advantages:

- Compatible with IBM,® Apple,® Commodore,® Atari™ and others
- The choice of monochrome or color monitors with TV-grade to high resolution text and graphics
- Monochrome text-editing mode for color monitors
- Alternate use as cable or VCR monitor
- Broad range of the features you need at the prices you want

Because of its versatility and compatibility, you can still use your Thomson monitor if you switch computers; you'll never outgrow the capabilities of a Thomson monitor.

Who is Thomson? Thomson is a six billion dollar multi-national company. Unlike our competition, we design and manufacture our own monitors; so the quality you get is constant and superb.

Your Apple or Commodore computer is a great body. Choose a Thomson monitor, and give it a beautiful face.

For the name of the Thomson dealer nearest you, call 1-800-325-0464. In California call 1-213-568-1002. (Monday–Friday, 9 a.m. to 5:00 p.m. PST)

THOMSON

A SIGHT FOR SORE EYES.™

5731 W. Slauson Avenue, Suite 111, Culver City, CA 90230

From Origin comes the long-awaited sequel to the award-winning Ultima™ III

Ultima IV

Quest of the Avatar

Available on Apple®

A state-of-the-art fantasy role-playing game of unprecedented magnitude by Lord British™.

Prepare yourself for a grand adventure: Ultima™ IV, sixteen times larger than Ultima III, is a milestone in computer gaming—one that challenges your physical and mental skills while testing the true fabric of your character.

Enter Britannia, kingdom of Lord British. Journey through terrain of infinite proportions, conversing with characters on hundreds of topics. Unravel the mysteries of a superior magic system. At each turn beware of daemons, dragons and long-dead wizards haunting the most tranquil of places. Encounters with parties of mixed enemy types test your strategic abilities. Shrewd use of terrain can lead to victory against seemingly impossible odds.

Survive this multi-quest fantasy, then begin the final conflict, your quest of the Avatar. The ultimate challenge—the self—awaits....

ORIGIN SYSTEMS INC. 340 HARVEY ROAD, MANCHESTER, NH 03103 (603) 644-3360

ULTIMA™ III sends you on an incredible fantasy role-playing journey through monster-plagued Sosaria in search of the elusive Exodus.

MOEBIUS™ takes you through the elemental planes of a colorful Oriental world of fantasy and adventure in search of the Orb of Celestial Harmony.

AUTODUEL™ is a futuristic, fast-paced strategy role-playing game where the right of way goes to the biggest guns.

OGRE™ is a strategy game fought on the nuclear battlefield of tomorrow as an inhuman juggernaut Cybertank battles conventional forces.

Ultima and Lord British are trademarks of Richard Garriott/Moebius is a trademark of Greg Malone/AutoDuel and Ogre are trademarks of Steve Jackson/Apple is a trademark of Apple Computer Inc/Previous Ultimas are not needed to enjoy Ultima IV

Authors wanted. Call us today.