The VIC chip views memory differently than does the 6510 chip. VIC sees only 16K at a time and maps the ROM character set into part of this 16K bank at times. These commands allow changes to the normal locations of the screen and character sets. [BANK selects which one of four banks (0 – 3) the VIC chip sees. Normally this is bank 0. [BANK resets the pointer BASIC uses to locate the screen. [VS1K determines which 1K block of the 16 available is used for the text screen. The blocks are numbered 0 – 15. The BASIC screen pointer is reset for this location. [CB2K controls which 2K block of the 8 available is used for the character set. In banks 0 and 2 the ROM set is located at 2K blocks two and three. [CB2K is also used to select which 8K block is used for the bitmap screen, values 0 – 3 select the lower 8K block, and values 4 – 7 select the upper 8K block. These three commands must be used in coordination to smoothly relocate the screen. Caution must be exercised in selecting locations since a system crash will result if the screen overwrites important RAM such as page zero. Banks 2 and 3 must be used with great care. (More on bank 3 usage later.) Program 6 demonstrates relocation to PET standard locations for the screen and BASIC.

Graphics/Text Control

[ECGR [MCGR [BMGR

These commands select extended color, multicolor, or bitmap graphics modes. A value of 0 turns the mode off and a value of 1 turns the mode on. Only multicolor and bitmap work in conjunction with each other to form a combined mode. When extended color and bitmap are both on, the screen will appear blank. This effect might be useful for temporarily hiding the screen.

[MXGR [KMXG [CMXV

These commands set up a simple interrupt routine that allows mixed modes to appear in two sections of the screen. [MXGR will change the contents of one VIC register (reg) or part of its contents (the bits OFF in mask) each time the raster counter register equals one of the two raster select values (rast1 and rast2). The values in val1 or val2 will be stored into the selected VIC register. You must determine the appropriate value for the particular register. For example, [MXGR 33,240, 152,6,252,1 will cause screen lines 51 to 151 to be displayed with background white and lines 152 to 251 with background blue.

The visible portion of the screen extends from raster 51 to raster 251. [KMXG will kill the interrupt and leave the selected register in an unknown state. [CMXV (change mixed-mode values) allows changing val1 and val2 while mixed mode is in force. By setting them equal, a known state will be in effect after [KMXG. The interrupt routines are simple in that normal IRQ still occurs (keyboard scan, clock update, etc.) so that the transition will tend to creep. To keep the change precise, you must disable interrupts from the CIA. This will kill the keyboard, however, so I/O would be limited to joystick ports only.

[SIZE [XYSC

These commands help use the smooth scroll registers of the VIC chip. [SIZE selects 40 or 38 columns for the text display chosen by setting colsel to 1 or 0 (colsel = 1 selects 40 columns) and sets number of lines to 25 or 24 (rowsel = 1 selects 25 lines). [XYSC moves the entire text screen up to seven pixels horizontally or vertically. By setting xpos and ypos to a value in the range 0 – 7, the screen can be stepped a pixel at a time to produce a smooth scroll. When used in conjunction with a machine language scroll routine or the automatic scroll up, text can be scrolled smoothly across or up the entire screen.

[DLCS

[DLCS (download character set) assists in using banks without ROM character set images and in designing custom character sets. You can copy the uppercase graphics set, upper- and lowercase set, or both by setting set equal to 0, 1, or 2 respectively. This is followed by the address of the first location in memory where you wish the ROM set to be positioned. This should be on a 2K boundary unless you wish to change the order of the set. When the address is 53248, the set will be copied into the RAM beneath the ROM set for use in bank 3.

[FBMS [FSCR

The current hi-res screen (determined by the last [CB2K command) can be filled with any byte value with [FBMS (fill bitmap screen). [FBMS 0 would clear the entire 8K screen. [FSCR works in a similar way with the current text screen. The entire screen is filled with a byte value. Since the text screen is used for color control in hi-res mode, [FSCR can be used for hi-res color control.

[PLOT [FLIP [CLPX [MCPL

These commands are used in plotting pixel points in hi-res graphics modes. The first three plot in 320 × 200 resolution two-color mode, the last in 160 × 200 resolution four-color mode. [PLOT sets the selected pixel on, [CLPX turns the pixel off, and [FLIP changes the pixel to the opposite state. [MCPL (multicolor plot) accepts horizontal coordinates in the range 0,159 and plots in one of four colors determined by sel, with sel in the range 0,3. A value of 0 selects background color, 1 selects text screen low-byte color, 2 selects text screen high-byte color, and 3 selects color memory color. Before you execute any of the plotting commands, [CB2K must be used to select the appropriate 8K block and [BMGR 1 must be in force for the plot

FORECASTING

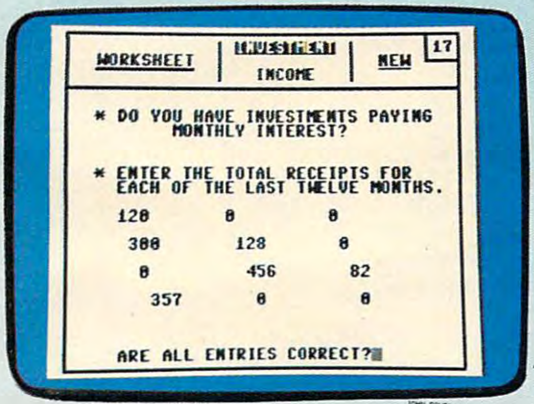
THE *REAL* PROBLEM IN PERSONAL FINANCE MANAGEMENT

Numerous screen graphic displays of detailed first year cash flow projections

- IF YOU HAVE HAD ENOUGH OF 'PIGGY BANK' FINANCE PROGRAMS ...
- IF YOU WANT TO KNOW WHERE YOUR BUDGET IS GOING INSTEAD OF WHERE IT HAS BEEN ...
- IF YOU HAVE SOME DISCRETION AVAILABLE IN ALLOCATING YOUR FUTURE INCOME ...

Then the Xana PERSONAL FINANCE FORECASTER, now available for the Commodore 64*, is specially designed for you.

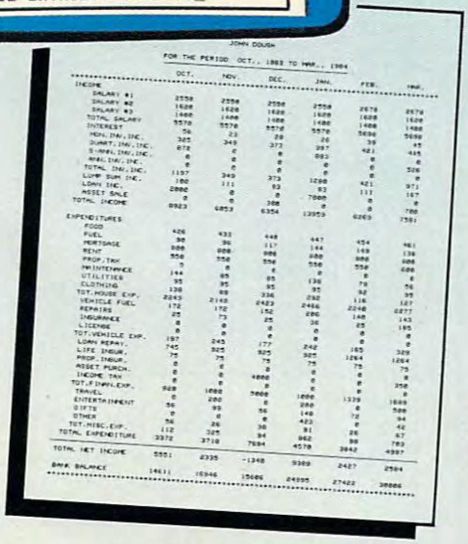
The **FORECASTER** uses **advanced mathematical techniques** to generate a rolling monthly, personalized cashflow projection based upon actual historical information and a limited number of estimates of future expenditures for discretionary items.



Data entry on clearly explained separate item worksheets

SOME UNIQUE FEATURES OF THE FORECASTER INCLUDE:

- *Selection of items from a large variety of income and expenditure categories personalizes your budget.*
- *Immediate revision of each entry and/or each page.*
- *Opportunities to revise any or all data as desired to study the "What if" sensitivity of the budget.*
- *Simple updating from month to month with minimum data entry.*
- *Resistance to typographical errors.*
- *Single keystroke screen dump of any complete page of data.*
- *Printout of input data.*
- *Computer generated forms for organizing input data.*



Tabular printout of complete one to three year projections

master charge

CHARGEX VISA

If your nearest Commodore 64 dealer does not have the PERSONAL FINANCE FORECASTER in stock, contact:

UNITED STATES (213) 410-9884
CANADA (403) 276-6834

XANA DATA SYSTEMS IS A DIVISION OF XANA ENGINEERING LTD.

**Commodore and the Commodore 64 are trademarks of Commodore Computer.*

to be seen. Remember that y coordinates increase as you go down the screen.

[DRAW

[DRAW is used to draw line segments on the hi-res screen. [CB2K and [BMGR must be used in preparation as in plot commands. [DRAW connects the endpoints given in the parameter list. The line is drawn from x1,y1 toward x2,y2.

[HRCS [CHAR [CHRX [CODE

These commands make it easy to put text on the hi-res screen. [HRCS (hi-res character set) stores the address of the character set to be used. It need not be located on a 2K boundary or even be the same set as used on the text screen. The address given is of the first byte of the set. A value of 53248 will select the ROM set (upper/graphics). [CHAR and [CHRX plot an 8 × 8 character to a selected position on the current hi-res screen. The character code (char) to select which character to plot corresponds to the screen POKE codes as listed in Commodore documentation. Example: [CHAR 1,100,100 would plot the letter A with position 100,100 being the upper-left corner of the 8 × 8 character cell. [CHAR plots the cell to the hi-res screen absolutely while [CHRX uses the exclusive OR function to flip the cell pixels. So [CHRX can be used to unplot a previously plotted character. [CODE helps in translating to the screen POKE code used by [CHAR and [CHRX in character selection.

The argument for [CODE must be the name of a defined string variable. Upon execution the ASCII values stored in the string will be converted to screen POKE codes. The RVS ON and RVS OFF control characters can be used within the string to select the upper 128 or lower 128 characters of the set. All other control characters will produce unpredictable results. Once the string is converted using [CODE, use the ASC function and MID$ function to read the codes. The ASC function will give correct results for the 0 character of the set. Be careful when using strings not built to high memory because [CODE will modify the actual string data stored within the BASIC text area.

[HRAM [LOOK [STUF

These commands make use of [BANK 3 possible from BASIC. When bank 3 is selected, the VIC chip uses RAM in the 64 from $C000 to $FFFF and ignores ROM located at the same addresses, including the ROM character set. SuperBASIC allows the location of one text screen ([VS1K block 3 located at $CC00) in bank 3. RAM from $D000 to $FFFF can be used for character sets, sprites, and a hi-res screen. The main problem confronting the bank 3 user is the switching required to read and write to these RAM locations. All plotting commands need to read as well as write to RAM so they can be preceded by [HRAM to accomplish this in bank 3. For example, [HRAMDRAW 1,0,100,100 would draw to the hi-res screen in RAM under the $E000 and $F000 ROMs. [HRAM should be used in this manner with [PLOT, [FLIP, [CLPX, [MCPL, [DRAW, [CHAR, and [CHRX in bank 3. [MXGR should be avoided in bank 3. Using the first 3K of bank 3 will crash SuperBASIC, so make sure the text screen is relocated by [VS1K 3. When the transition to bank 3 is accomplished,

SuperBASIC Commands

Enhanced BASIC Commands

- RESTORE <exp>
- GOTO <exp>
- GOSUB <exp>
- IF <exp> GOTO <exp>
- IF <exp> GOSUB <exp>
- ON <exp> GOTO <exp1>,<exp2>,...
- ON <exp> GOSUB <exp1>,<exp2>,...
- LIST (Shift Key halts list)

New SuperBASIC Commands

Sprite Commands
- [DSPR spr,blk,xexp,yexp,xpos,ypos,multi, sprcolr,mc0,mc1
- [MOVE spr,xpos,ypos
- [KSPR spr
- [ESPR spr
- [BSPP spr,sel

Sound Commands
- [SSND voice,ad,sr,wave,freq,pwidth
- [PLAY 256*wave + voice,freq,pwidth

VIC Color Control
- [BKGD col
- [BKG4 col0,col1,col2,col3
- [EXTC col
- [FCOL col

VIC Memory Mapping
- [BANK sel
- [VS1K sel
- [CB2K sel

Graphics Control
- [ECGR sel
- [MCGR sel
- [BMGR sel
- [MXGR reg,mask,rast1,val1,rast2,val2
- [KMXG
- [CMXV val1,val2
- [SIZE colsel,rowsel
- [XYSC xpos,ypos
- [DLCS set,address
- [FBMS byte
- [FSCR byte
- [PLOT x,y
- [FLIP x,y
- [CLPX x,y
- [MCPL x,y,sel
- [DRAW x1,y1,x2,y2
- [HRCS address
- [CHAR char,x,y
- [CHRX char,x,y
- [CODE str$
- [LOOK address, variable
- [STUF address,byte
- [HRAM <SuperBASIC mnemonic> <parameter list>

THE LAST ONE

The first program you should buy.

The more you use your computer, the more you want it to work for you.

But where do you begin? There are literally thousands of programs. It's time consuming, confusing and frustrating! The answer is to begin with THE LAST ONE™.

THE LAST ONE . . . The program that writes programs!

Now, for the first time, your computer is truly 'personal'. Now, simply and easily, you can create software the way you want it.

From Accounting to the Zodiac, THE LAST ONE puts you keystrokes away from whatever you need from your computer.

THE LAST ONE . . . *See it at your dealer and buy it first!*

Available for Commodore 64™, Commodore 8032™, IBM PC™, Victor 9000™, Apple II™ and IIe™, Radio Shack Model II™ and most CP/M™ systems.

Distributed By
Computer Marketing Services, Inc.
300 W. Marlton Pike, Cherry Hill, NJ 08002 (609) 795-9480

Product of BLUE SKY SOFTWARE

*THE LAST ONE is a registered trademark of D. J. "AI" Systems, Ltd.

The Commodore 64 & CBM 8032, IBM PC, Victor 9000, Apple II & IIe, Radio Shack Model II, and CP/M are registered trademarks of Commodore Business Machines, Inc., International Business Machines Corp., Victor Technologies, Inc., Apple Computers, Inc., The Tandy Corporation, and Digital Research Corp., respectively.

the 1K block at $0400 can be reclaimed for BASIC program storage. [LOOK and [STUF are PEEK and POKE equivalents that can be used with [HRAM to examine and change RAM. [LOOK is different from PEEK in that a defined variable name is used in the parameter list to store the value read from memory. [STUF works the same as POKE and is primarily useful for storing to block $D000 RAM (for example, [HRAMSTUF 53248,255).

Programs 2 – 6 are demonstration programs which should be helpful in seeing the commands used in actual applications.

If you're not up to typing in SuperBASIC yourself, send $3 along with a blank disk (no tapes) and a stamped, self-addressed mailer to:

Martin C. Kees
711 West Henry
Pasco, WA 99301

Program 1: SuperBASIC 64

```
2049 :011,008,000,000,158,050,228
2055 :048,056,048,000,000,000,159
2061 :000,000,000,000,000,000,013
2067 :000,000,000,000,000,000,019
2073 :000,000,000,000,000,000,025
2079 :000,169,039,133,001,169,030
2085 :000,133,020,133,078,169,058
2091 :009,133,021,169,192,133,188
2097 :079,162,012,160,000,177,127
2103 :020,145,078,200,208,249,187
2109 :230,021,230,079,202,208,007
2115 :242,160,008,169,104,032,014
2121 :030,171,169,013,141,119,204
2127 :002,141,120,002,169,002,003
2133 :133,198,169,033,141,001,248
2139 :008,169,020,141,002,008,183
2145 :076,120,008,000,000,000,045
2151 :000,031,147,017,017,048,107
2157 :017,157,082,085,078,019,035
2163 :000,000,000,000,000,169,028
2169 :000,133,020,169,160,133,224
2175 :021,162,032,160,000,177,167
2181 :020,145,020,136,208,249,143
2187 :230,021,202,208,244,162,182
2193 :000,160,003,185,224,160,109
2199 :157,224,160,232,200,224,068
2205 :190,208,244,169,003,141,088
2211 :161,168,169,192,141,162,132
2217 :168,169,074,141,210,166,073
2223 :169,193,141,211,166,141,172
2229 :037,160,169,084,141,036,040
2235 :160,169,219,141,223,160,235
2241 :169,255,141,044,160,169,107
2247 :194,141,045,160,169,038,178
2253 :133,001,169,005,141,143,029
2259 :183,169,076,141,043,169,224
2265 :141,087,169,169,193,141,093
2271 :045,169,141,089,169,169,237
2277 :200,141,088,169,169,227,199
2283 :141,044,169,096,000,000,173
2289 :000,000,000,000,000,000,241
2295 :000,000,000,000,000,000,247
2301 :000,000,000,032,115,000,144
2307 :032,158,173,032,247,183,060
2313 :096,032,139,192,032,000,244
2319 :192,165,020,166,002,157,205
2325 :248,007,032,000,192,165,153
2331 :020,162,029,032,162,192,112
2337 :032,000,192,165,020,162,092
2343 :023,032,162,192,032,097,065
2349 :192,032,000,192,165,020,134
2355 :072,162,028,032,162,192,187
2361 :032,000,192,165,020,166,120
2367 :002,157,039,208,104,240,045
2373 :117,032,000,192,165,020,083
2379 :141,037,208,032,000,192,173
2385 :165,020,141,038,208,169,054
2391 :001,162,021,032,162,192,145
2397 :096,032,139,192,032,000,072
2403 :192,165,021,072,165,020,222
2409 :072,032,000,192,165,002,056
2415 :010,170,232,165,020,157,097
2421 :000,208,202,104,157,000,020
2427 :208,104,162,016,032,162,039
2433 :192,169,000,141,030,208,101
2439 :141,031,208,096,032,000,131
2445 :192,165,020,041,007,133,187
2451 :002,170,169,001,224,000,201
2457 :240,004,010,202,208,252,045
2463 :133,078,096,164,078,201,141
2469 :000,240,006,152,029,000,080
2475 :208,208,006,152,073,255,049
2481 :061,000,208,157,000,208,043
2487 :096,000,007,014,032,019,095
2493 :199,240,150,032,000,192,234
2499 :165,020,041,003,170,189,015
2505 :183,192,133,078,169,212,144
2511 :133,079,032,000,192,165,040
2517 :020,160,005,145,078,032,141
2523 :000,192,165,020,160,006,250
2529 :145,078,032,028,193,165,098
2535 :020,133,002,160,004,145,183
2541 :078,032,037,193,169,015,249
2547 :141,024,212,096,032,000,236
2553 :192,165,020,041,003,170,072
2559 :189,183,192,133,078,169,175
2565 :212,133,079,165,021,133,236
2571 :002,169,000,160,004,145,235
2577 :078,032,037,193,165,002,012
2583 :160,004,145,078,096,169,163
2589 :000,160,004,145,078,076,236
2595 :000,192,032,000,192,165,104
2601 :021,160,001,145,078,165,099
2607 :020,136,145,078,165,002,081
2613 :201,065,208,016,032,000,063
2619 :192,165,021,041,015,160,141
2625 :003,145,078,165,020,136,100
2631 :145,078,096,173,141,002,194
2637 :208,251,076,044,168,076,132
2643 :029,168,240,251,032,003,038
2649 :192,032,019,166,056,165,207
2655 :095,233,001,164,096,176,092
2661 :001,136,133,065,132,066,122
2667 :096,032,000,192,160,000,075
2673 :177,020,133,002,032,115,080
2679 :000,032,040,175,164,002,020
2685 :169,000,032,145,179,166,048
2691 :071,164,072,032,215,187,104
2697 :096,032,000,192,165,020,130
2703 :133,078,165,021,133,079,240
2709 :032,000,192,165,020,160,206
2715 :000,145,078,096,173,014,149
2721 :220,041,254,141,014,220,027
2727 :165,001,041,253,133,001,249
2733 :169,193,072,169,184,072,008
```


SM TEXT means ● Multi-Column Processing ● Sophisticated Mathematics ● Advanced Editing ● Total File Overview ● Form Mode ● Noted Text ● Ergonomic Design ● Ease of Operation and a lot of more features.
SM TEXT is available for commodore-business computers.
MUCH HAS ALREADY BEEN WRITTEN ABOUT WORDPROCESSING. SOME OF IT WITH SM-TEXT - SINCE PROFESSIONAL WRITERS PREFER PROFESSIONAL TOOLS.
MCMLXXXIII
CREATE YOUR LIFE MORE SUCCESSFUL!
CBM
Me too in future . . .
SM-CUDA User Guide
SM-Know-how
SM-TEXT
SM SOFTWARE INC. IS A SUBSIDIARY OF SM SOFTWARE AG, MUNICH, ONE OF EUROPE'S TOP MICRO SOFTWARE HOUSES.
U.S. computer technology, allied with German software engineering, has also produced high quality programming aids such as:
● SM KIT – program de-bug aids
● ESH – ergonomic screen handler
● ISM – index sequential file manager
● LOS – loadable operating system
ASK FOR DETAILS.
SM SOFTWARE INC.
252 Bethlehem Pike
Colmar. PA 18915
SOFTWARE MADE IN MUNICH
Dealer inquiries invited.

```
2739 :032,115,000,076,000,195,085
2745 :165,001,009,002,133,001,240
2751 :173,014,220,009,001,141,237
2757 :014,220,096,165,101,133,158
2763 :254,104,133,002,198,254,124
2769 :208,005,165,002,076,239,136
2775 :167,032,000,192,032,121,247
2781 :000,201,044,240,237,096,015
2787 :032,121,000,201,137,208,158
2793 :003,076,055,169,201,141,110
2799 :240,249,076,050,169,000,255
2805 :000,000,000,000,000,000,245
2811 :000,000,000,000,000,077,072
2817 :079,086,069,093,192,068,076
2823 :083,080,082,101,196,083,120
2829 :083,078,068,191,192,080,193
2835 :076,065,089,246,192,066,241
2841 :075,071,068,056,195,069,047
2847 :088,084,067,065,195,075,093
2853 :083,080,082,074,195,069,108
2859 :083,080,082,084,195,066,121
2865 :083,080,080,092,195,083,150
2871 :084,085,070,137,193,069,181
2877 :067,071,082,125,195,077,166
2883 :067,071,082,150,195,066,186
2889 :077,071,082,175,195,083,244
2895 :073,090,069,187,195,088,013
2901 :089,083,067,211,195,067,029
2907 :065,084,065,250,195,066,048
2913 :065,078,075,053,196,086,138
2919 :083,049,075,113,196,067,174
2925 :066,050,075,151,196,068,203
2931 :076,067,083,172,196,077,018
2937 :088,071,082,065,197,075,187
2943 :077,088,071,180,197,067,039
2949 :077,088,086,200,197,070,083
2955 :067,079,076,217,197,080,087
2961 :076,079,084,130,198,070,014
2967 :076,073,080,122,198,067,255
2973 :076,080,088,138,198,077,046
2979 :067,080,076,148,198,070,034
2985 :083,067,082,197,198,070,098
2991 :066,077,083,232,198,068,131
2997 :082,065,087,097,199,072,015
3003 :082,067,083,060,201,067,235
3009 :072,065,082,134,202,067,047
3015 :072,082,088,142,202,067,084
3021 :079,068,069,150,202,076,081
3027 :079,079,075,107,193,066,042
3033 :075,071,052,105,195,072,019
3039 :082,065,077,158,193,255,029
3045 :255,255,255,255,255,255,223
3051 :255,255,255,255,255,255,229
3057 :255,255,255,255,255,255,235
3063 :255,255,255,255,255,255,241
3069 :255,197,200,162,000,134,177
3075 :002,160,000,177,122,221,173
3081 :000,194,208,026,232,200,101
3087 :192,004,208,243,189,001,084
3093 :194,072,189,000,194,072,230
3099 :165,122,024,105,003,133,067
3105 :122,144,002,230,123,096,238
3111 :165,002,024,105,006,133,218
3117 :002,170,189,000,194,201,033
3123 :255,208,206,076,008,175,211
3129 :032,000,192,165,020,141,095
3135 :033,208,096,032,000,192,112
3141 :165,020,141,032,208,096,219
3147 :032,139,192,169,000,162,001
3153 :021,076,162,192,032,139,191
3159 :192,162,021,076,162,192,124
3165 :032,139,192,032,000,192,168
3171 :165,020,162,027,076,162,199
3177 :192,162,000,134,002,032,115
3183 :000,192,165,020,166,002,144
3189 :157,033,208,232,224,004,207
3195 :208,239,096,032,000,192,122
3201 :165,020,162,017,160,064,205
3207 :032,164,192,165,020,240,180
3213 :239,169,000,152,022,160,125
3219 :016,076,164,192,032,000,115
3225 :192,165,020,162,022,160,106
3231 :016,032,164,192,165,020,236
3237 :240,214,169,000,162,017,199
3243 :160,064,076,164,192,032,091
3249 :000,192,165,020,162,017,221
3255 :160,032,076,164,192,032,071
3261 :000,192,165,020,162,022,238
3267 :160,008,032,164,192,032,015
3273 :000,192,165,020,162,017,245
3279 :160,008,076,164,192,032,071
3285 :000,192,165,020,041,007,126
3291 :133,020,173,022,208,041,048
3297 :248,005,020,141,022,208,101
3303 :032,000,192,165,020,041,169
3309 :007,133,020,173,017,208,027
3315 :041,248,005,020,141,017,203
3321 :208,096,169,032,141,000,127
3327 :002,162,000,142,005,002,056
3333 :134,002,173,141,002,208,153
3339 :251,160,000,189,000,194,037
3345 :153,001,002,232,200,192,029
3351 :004,208,244,169,000,160,040
3357 :002,032,030,171,165,002,175
3363 :024,105,006,133,002,170,219
3369 :189,000,194,201,255,208,064
3375 :215,032,115,000,208,251,100
3381 :096,173,002,221,009,003,045
3387 :141,002,221,032,000,192,135
3393 :165,020,041,003,072,073,183
3399 :003,133,020,173,000,221,109
3405 :041,252,005,020,141,000,024
3411 :221,104,024,106,106,106,238
3417 :133,020,173,136,002,041,082
3423 :063,005,020,141,136,002,206
3429 :096,173,136,002,024,105,125
3435 :003,141,022,192,076,010,039
3441 :192,032,000,192,165,020,202
3447 :041,063,010,010,133,020,140
3453 :173,136,002,041,192,005,162
3459 :020,141,136,002,165,020,103
3465 :010,010,133,020,173,024,251
3471 :208,041,015,005,020,141,061
3477 :024,208,096,173,024,208,114
3483 :041,241,133,002,032,000,092
3489 :192,165,020,041,007,010,084
3495 :005,002,141,024,208,096,131
3501 :173,014,220,041,254,141,248
3507 :014,220,165,001,041,251,103
3513 :133,001,032,000,192,165,196
3519 :020,041,003,162,008,201,114
3525 :002,208,002,162,016,160,235
3531 :208,201,001,208,002,160,215
3537 :216,132,079,160,000,132,160
3543 :078,134,002,032,000,192,141
3549 :166,002,160,000,177,078,036
3555 :145,020,200,208,249,230,255
3561 :021,230,079,202,208,242,191
3567 :165,001,009,004,133,001,040
3573 :173,014,220,009,001,141,035
```

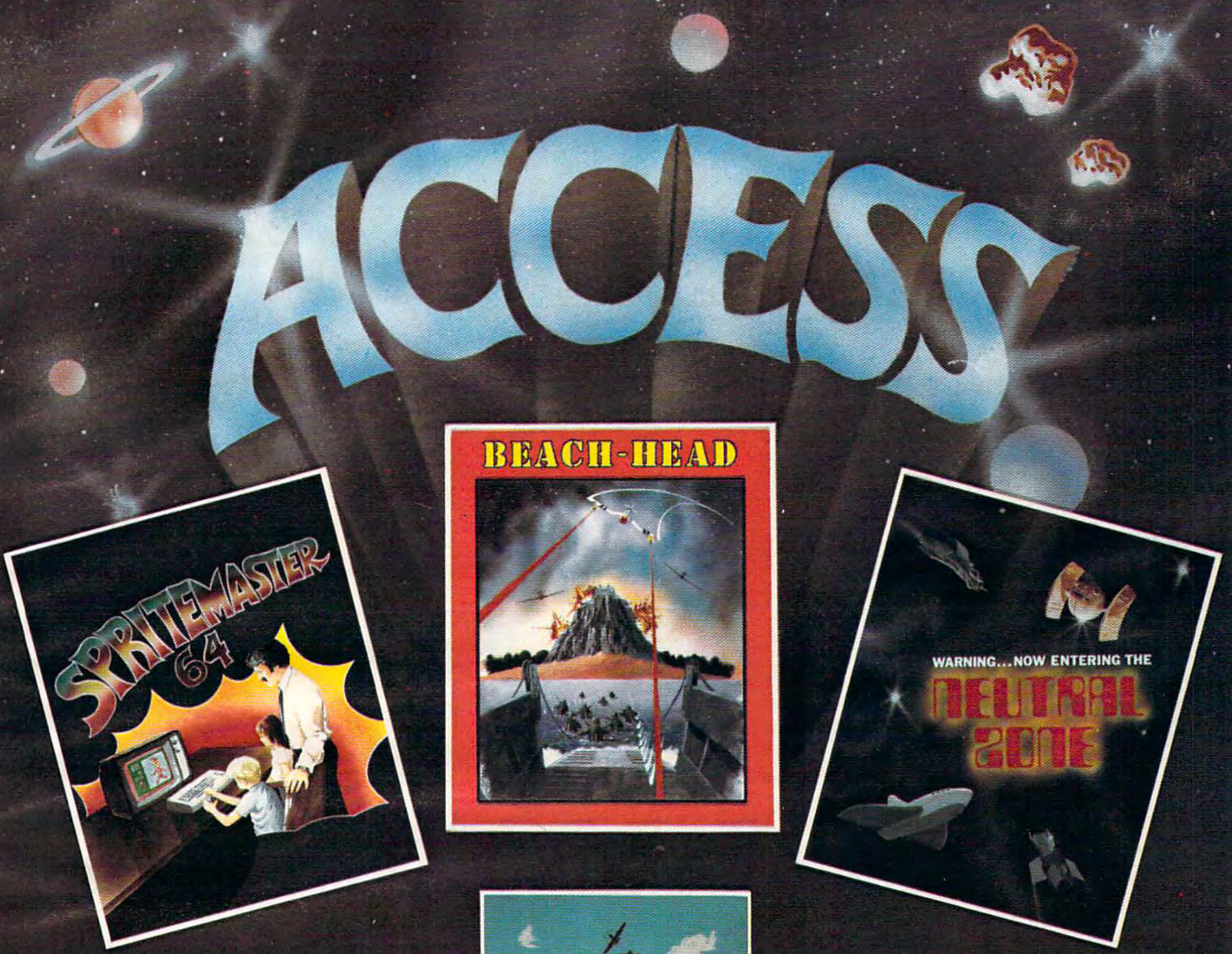

Let the Genesis Genie Loose in Your House!!!
HE WILL WORK MAGIC WITH YOUR COMMODORE 64 OR VIC 20.
G.G. will control switches around your house (or business) using existing wiring. Turns on lights, TVs, radios, coffee makers and many other appliances (will turn them off as well).
VIController $59.95
Just type the words "SPEAK" followed by any other single words, numbers or sentences and the COMvoice speaks those words in four inflection levels.
COMvoice $149.95
An extremely reliable and accurate clock/calendar with battery back-up. It keeps the correct time and date even with the power off!
COMclock $49.95
A battery back-up for your Commodore when the power goes out. You no longer have to lose programs during a power black-out. They keep on running—up to 30 minutes. Charges automatically.
COMpower $99.95
Reads input from the outside world—in combination with the VIController would make a great burglar alarm to turn on the lights or issue a warning.
COMsense $39.95
GENESIS COMPUTER CORPORATION
AVAILABLE WHEREVER COMMODORE PRODUCTS ARE SOLD.
1444 LINDEN STREET, P.O. Box 1143, BETHLEHEM, PA 18018 (215) 861-0850

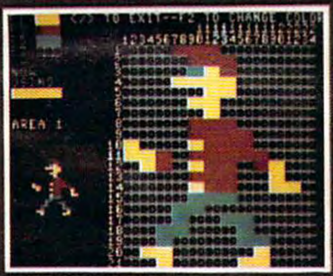
```
3579 :014,220,096,000,000,154,223
3585 :000,006,252,000,006,000,009
3591 :033,240,044,025,208,048,093
3597 :003,076,049,234,173,006,042
3603 :197,073,003,141,006,197,124
3609 :170,172,007,197,185,000,244
3615 :208,045,008,197,029,002,008
3621 :197,153,000,208,173,017,017
3627 :208,041,127,029,001,197,134
3633 :141,017,208,189,000,197,033
3639 :141,018,208,169,001,141,221
3645 :025,208,076,129,234,120,085
3651 :169,240,141,026,208,032,115
3657 :000,192,165,020,141,007,086
3663 :197,032,000,192,165,020,173
3669 :141,008,197,032,000,192,143
3675 :165,020,141,003,197,165,014
3681 :021,041,001,240,002,169,059
3687 :128,141,004,197,032,000,093
3693 :192,165,020,141,002,197,058
3699 :032,000,192,165,020,141,153
3705 :000,197,165,021,041,001,034
3711 :240,002,169,128,141,001,040
3717 :197,032,000,192,165,020,227
3723 :141,005,197,173,017,208,112
3729 :041,127,013,004,197,141,156
3735 :017,208,173,003,197,141,122
3741 :018,208,169,003,141,006,190
3747 :197,169,241,141,026,208,121
3753 :169,009,141,020,003,169,168
3759 :197,141,021,003,088,096,209
3765 :120,169,049,141,020,003,171
3771 :169,234,141,021,003,169,156
3777 :240,141,026,208,088,076,204
3783 :115,000,032,000,192,165,191

3789 :020,141,002,197,032,000,085
3795 :192,165,020,141,005,197,163
3801 :096,032,000,192,165,020,210
3807 :041,015,162,000,157,000,086
3813 :216,157,000,217,157,000,208
3819 :218,157,000,219,232,208,245
3825 :241,141,134,002,096,032,119
3831 :000,192,165,020,041,007,160
3837 :133,002,165,020,041,248,094
3843 :133,251,165,021,133,252,190
3849 :032,000,192,165,020,133,039
3855 :078,041,248,133,020,133,156
3861 :253,169,000,133,254,024,086
3867 :006,253,038,254,006,253,069
3873 :038,254,165,020,101,253,096
3879 :133,253,144,002,230,254,031
3885 :024,006,253,038,254,006,114
3891 :253,038,254,006,253,038,125
3897 :254,165,078,041,007,005,095
3903 :253,133,253,024,165,251,118
3909 :101,253,133,251,165,252,200
3915 :101,254,133,252,173,136,100
3921 :002,041,192,005,252,133,194
3927 :252,173,024,208,041,008,025
3933 :010,010,005,252,133,252,243
3939 :166,002,189,107,198,160,153
3945 :000,096,128,064,032,016,185
3951 :008,004,002,001,192,048,110
3957 :012,003,000,085,170,255,130
3963 :032,246,197,081,251,145,051
3969 :251,096,032,246,197,017,200
3975 :251,145,251,096,032,246,132
3981 :197,073,255,049,251,145,087
3987 :251,096,032,000,192,165,115
3993 :020,041,003,024,105,008,098
3999 :133,002,006,020,038,021,123
4005 :032,255,197,133,002,032,048
4011 :000,192,165,020,041,003,080
4017 :170,189,119,198,037,002,124
4023 :133,020,165,002,073,255,063
4029 :160,000,049,251,005,020,162
4035 :145,251,096,032,000,192,143
4041 :173,136,002,133,252,169,042
4047 :000,133,251,168,162,003,156
4053 :165,020,145,251,200,208,178
4059 :251,230,252,202,208,246,072
4065 :145,251,200,192,232,208,173

4071 :249,096,032,000,192,173,205
4077 :136,002,041,192,133,252,225
4083 :173,024,208,041,008,010,195
4089 :010,005,252,133,252,169,046
4095 :000,133,251,162,032,160,225
4101 :000,165,020,145,251,200,018
4107 :208,251,230,252,202,208,082
4113 :246,096,032,121,000,208,208
4119 :001,096,104,104,076,070,218
4125 :192,169,000,141,176,002,197
4131 :141,178,002,141,179,002,166
4137 :173,167,002,013,168,002,054
4143 :208,002,056,096,162,024,083
4149 :046,176,002,046,177,002,246
4155 :046,178,002,046,179,002,000
4161 :056,173,178,002,237,167,110
4167 :002,168,173,179,002,237,064
4173 :168,002,144,006,140,178,203
4179 :002,141,179,002,202,208,049
4185 :219,046,176,002,046,177,243
4191 :002,024,096,032,000,192,185
4197 :165,020,141,193,002,165,019
4203 :021,141,194,002,032,000,241
4209 :192,165,020,141,197,002,062
4215 :032,000,192,165,020,141,157
4221 :195,002,165,021,141,196,077
4227 :002,032,000,192,165,020,030
4233 :141,198,002,169,000,141,020
4239 :202,002,056,173,198,002,008
4245 :237,197,002,141,199,002,159
4251 :176,014,169,255,141,202,088
4257 :002,077,199,002,141,199,013
4263 :002,238,199,002,169,000,009
4269 :141,203,002,056,173,195,175
4275 :002,237,193,002,141,200,186
4281 :002,173,196,002,237,194,221
4287 :002,141,201,002,176,027,228
4293 :169,255,141,203,002,077,020
4299 :200,002,141,200,002,169,149
4305 :255,077,201,002,141,201,062
4311 :002,238,200,002,208,003,100
4317 :238,201,002,169,000,141,204
4323 :204,002,173,199,002,205,244
4329 :200,002,169,000,237,201,018
4335 :002,176,076,173,199,002,099
4341 :208,005,141,205,002,240,022
4347 :105,141,177,002,173,200,025
4353 :002,141,167,002,173,201,175
4359 :002,141,168,002,169,255,232
4365 :141,205,002,032,030,199,110
4371 :144,003,076,058,201,173,162
4377 :176,002,013,177,002,208,091
4383 :020,169,255,141,176,002,026
4389 :141,177,002,169,000,141,155
4395 :208,002,169,025,141,209,029
4401 :002,208,049,169,000,141,106
4407 :208,002,141,209,002,240,089
4413 :039,169,255,141,204,002,103
```

COMMODORE 64™ SOFTWARE





SPRITEMASTER™ is not just another sprite editor. It's the finest utility available for multicolor sprite animation and game programming. It will have you making full color animated objects in just minutes. People running, birds flying or tanks rolling are a snap with Spritemaster. It will **automatically** append your sprites to other programs. It's easy to use and understand and comes with a full 21 page instruction manual and samples of animated sprites to get you started. (Suggested retail price... $35.95)

GENERAL QUARTERS! BATTLE STATIONS! As chief commander of land and sea forces in the Pacific, your mission is to obtain a quick naval victory, and invade enemy territory with land forces. **BEACH-HEAD™** is a 100% machine language game and offers multi-screen action with high resolution, three dimensional graphics. (Suggested retail price... $34.95)

NEUTRAL ZONE™ takes you to the outer edges of the galaxy, to ALPHA IV, a long range early warning station whose mission is to detect alien intruders from other galaxies.

NEUTRAL ZONE™ is the ultimate in high resolution, fast action, arcade quality games. It is written in 100% machine language and features smooth scrolling of the 360 degree panorama. The realism is unbelievable. (Suggested retail price... $34.95)

SPRITEMASTER™, NEUTRAL ZONE™ AND BEACH-HEAD™ ARE AVAILABLE AT YOUR LOCAL COMMODORE DEALER ON EITHER DISK OR TAPE.

ACCESS SOFTWARE INC

925 EAST 900 SOUTH, SALT LAKE CITY, UTAH 84105
TELEPHONE (801) 532-1134

```
4419 :173,200,002,024,109,201,008
4425 :002,240,171,173,199,002,092
4431 :141,167,002,169,000,141,187
4437 :168,002,173,200,002,141,003
4443 :177,002,169,255,141,205,016
4449 :002,076,016,200,238,200,061
4455 :002,238,199,002,173,193,142
4461 :002,041,007,133,002,173,211
4467 :193,002,041,248,133,251,215
4473 :173,194,002,133,252,173,024
4479 :197,002,032,014,198,017,075
4485 :251,145,251,173,204,002,135
4491 :208,095,173,203,002,240,036
4497 :016,056,173,193,002,233,050
4503 :001,141,193,002,176,013,165
4509 :206,194,002,144,008,238,181
4515 :193,002,208,003,238,194,233
4521 :002,056,173,200,002,233,067
4527 :001,141,200,002,176,003,186
4533 :206,201,002,024,173,200,219
4539 :002,109,201,002,240,120,093
4545 :173,205,002,240,165,024,234
4551 :173,176,002,109,208,002,101
4557 :141,208,002,173,177,002,140
4563 :109,209,002,141,209,002,115
4569 :144,144,173,202,002,240,098
4575 :006,206,197,002,076,107,049
4581 :200,238,197,002,076,107,025
4587 :200,173,202,002,240,006,034
4593 :206,197,002,076,250,200,148
4599 :238,197,002,206,199,002,067
4605 :240,058,173,205,002,240,147
4611 :040,024,173,176,002,109,015
4617 :208,002,141,208,002,173,231
4623 :177,002,109,209,002,141,143
4629 :209,002,144,019,173,203,003
4635 :002,240,017,056,173,193,196
4641 :002,233,001,141,193,002,093
4647 :176,003,206,194,002,076,184
4653 :107,200,238,193,002,208,225
4659 :248,238,194,002,208,243,160
4665 :096,198,122,096,032,000,089
4671 :192,165,020,141,075,201,089
4677 :165,021,141,076,201,096,001
4683 :143,183,000,169,000,141,199
4689 :193,002,141,196,002,032,135
4695 :000,192,165,020,141,197,034
4701 :002,032,000,192,169,056,032
4707 :197,020,169,001,229,021,224
4713 :176,005,169,255,141,193,020
4719 :002,165,020,041,007,133,223
4725 :002,165,020,041,248,133,214
4731 :251,165,021,133,252,032,209
4737 :000,192,169,192,197,020,131
4743 :176,005,169,255,141,196,053
4749 :002,165,020,041,007,141,005
4755 :194,002,141,195,002,165,078
4761 :020,032,014,198,165,251,065
4767 :041,248,133,251,173,197,178
4773 :002,133,020,169,000,133,110
4779 :021,006,020,038,021,006,027
4785 :020,038,021,006,020,038,064
4791 :021,024,173,075,201,101,010
4797 :020,133,020,165,021,109,145
4803 :076,201,133,021,024,165,047
4809 :251,105,008,141,177,002,117
4815 :165,252,105,000,141,178,024
4821 :002,165,021,041,208,201,083
4827 :208,208,007,120,165,001,160
4833 :041,251,133,001,169,000,052
4839 :141,176,002,166,002,240,190
4845 :005,056,106,202,208,251,041
4851 :141,179,002,172,176,002,147
4857 :177,020,166,002,240,004,090
4863 :074,202,208,252,032,077,076
4869 :202,208,238,044,193,002,124
4875 :048,056,056,169,008,229,065
4881 :002,133,002,201,008,240,091
4887 :045,173,177,002,133,251,036
4893 :173,178,002,133,252,169,168
4899 :000,141,176,002,173,194,209
4905 :002,141,195,002,173,179,221
4911 :002,073,255,141,179,002,187
4917 :172,176,002,177,020,166,254
4923 :002,010,202,208,252,032,253
4929 :077,202,208,240,169,004,197
4935 :005,001,133,001,088,096,139
4941 :172,195,002,044,077,201,000
4947 :048,012,133,254,173,179,114
4953 :002,049,251,005,254,076,214
4959 :099,202,081,251,145,251,100
4965 :200,140,195,002,192,008,070
4971 :208,017,160,064,140,195,123
4977 :002,230,252,044,196,002,071
4983 :016,005,169,007,141,176,121
4989 :002,238,176,002,173,176,124
4995 :002,201,008,096,169,000,095
5001 :141,077,201,076,078,201,143
5007 :169,255,141,077,201,076,038
5013 :078,201,032,115,000,032,095
5019 :040,175,234,234,234,234,026
5025 :234,234,165,071,133,020,250
5031 :165,072,133,021,160,000,206
5037 :177,020,240,213,056,165,020
5043 :020,233,002,133,020,176,251
5049 :002,198,021,177,020,197,032
5055 :069,208,196,200,177,020,037
5061 :197,070,208,189,160,003,000
5067 :177,020,133,251,200,177,137
5073 :020,133,252,169,000,133,148
5079 :253,133,002,133,254,160,126
5085 :000,177,071,170,164,002,037
5091 :177,251,201,018,208,007,065
5097 :169,128,133,253,076,009,233
5103 :203,201,146,208,007,169,149
5109 :000,133,253,076,009,203,151
5115 :041,191,016,002,073,192,254
5121 :005,253,164,254,145,251,049
5127 :230,254,230,002,202,208,109
5133 :211,165,254,160,000,145,180
5139 :071,096,000,000,000,000,186
5145 :000,000,000,000,000,000,025
5151 :000,000,072,020,010,000,133
5157 :153,034,147,154,083,085,181
5163 :080,069,082,066,065,083,232
5169 :073,067,032,066,089,032,152
5175 :077,067,083,079,070,084,003
5181 :032,040,067,041,032,049,066
5187 :057,056,051,034,000,101,110
5193 :020,015,000,129,074,178,233
5199 :049,164,053,048,058,161,100
5205 :065,036,058,139,065,036,228
5211 :178,034,034,167,130,058,180
5217 :137,050,048,000,107,020,203
5223 :016,000,130,000,143,020,156
5229 :020,000,153,034,091,067,218
5235 :065,084,065,034,058,144,053
5241 :067,065,084,065,058,144,092
5247 :070,067,079,076,049,052,008
5253 :058,144,066,075,071,068,103
5259 :054,058,162,000,000,000,157
```

Richvale Telecommunications

10610 BAYVIEW (Bayview Plaza)
RICHMOND HILL, ONTARIO, CANADA L4C 3N8
(416) 884-4165

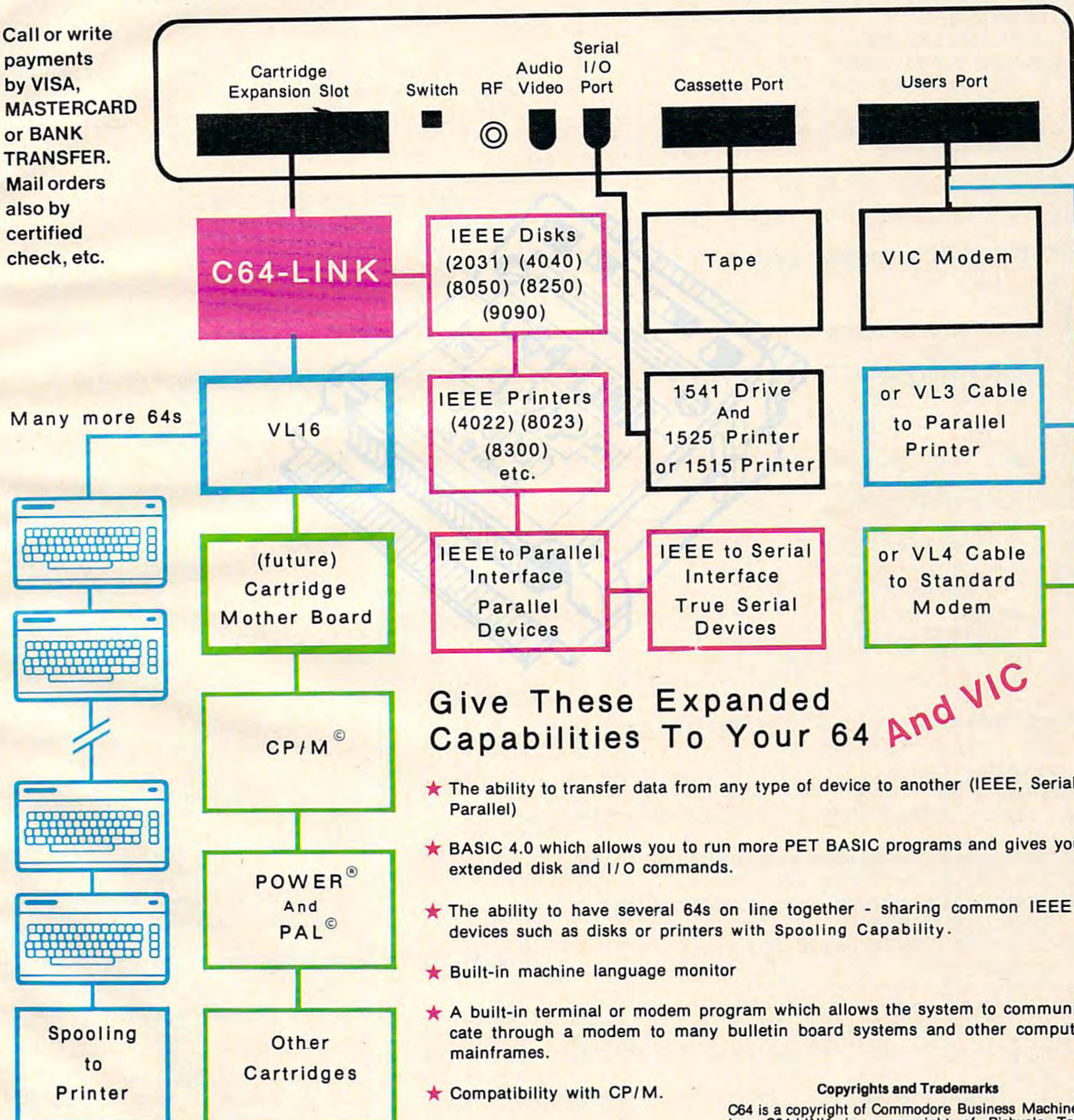
$185.00 Canadian
$149.00 U.S.
PLUS CUSTOMS BROKERAGE, HANDLING AND MAILING CHARGE.

Also available for VIC 20

C64-LINK©
The Smart 64

RTC **RTC**

Call or write payments by VISA, MASTERCARD or BANK TRANSFER. Mail orders also by certified check, etc.



Give These Expanded Capabilities To Your 64 And VIC

- ★ The ability to transfer data from any type of device to another (IEEE, Serial, Parallel)
- ★ BASIC 4.0 which allows you to run more PET BASIC programs and gives you extended disk and I/O commands.
- ★ The ability to have several 64s on line together - sharing common IEEE devices such as disks or printers with Spooling Capability.
- ★ Built-in machine language monitor
- ★ A built-in terminal or modem program which allows the system to communicate through a modem to many bulletin board systems and other computer mainframes.
- ★ Compatibility with CP/M.

Contact your local Commodore Dealer or RTC.

Program 2: Moiré Pattern

```
1 REM MOIRE TITLE PAGE DEMO
5 [EXTCØ
1Ø [CB2K4:[BMGR1:[FBMSØ :[FSCR1
15 FORJ=Ø TO318 STEP2
2Ø [DRAWJ,198,16Ø,1ØØ :NEXT
22 FORJ=Ø TO318 STEP2
23 [DRAWJ,Ø,16Ø,1ØØ :NEXT
24 FORJ=Ø TO198 STEP2
25 [DRAW16Ø,1ØØ,318,J{3 SPACES}:NEXT
26 FORJ=Ø TO198 STEP2
27 [DRAW161,1ØØ,Ø,J{3 SPACES}:NEXT
29 [EXTC4
3Ø M$="SUPERBASIC":[HRCS53248:M$=M$+""
4Ø X=12Ø:Y=8Ø:GOSUB5Ø
45 M$="{RVS}BY MCSOFT":M$=M$+"":X=124:Y=1
   2Ø:GOSUB5Ø
47 [CHRX54,152,89:[CHRX52,16Ø,89
48 FORJ=1TO8ØØ:NEXT
49 [FSCR16:{5 SPACES}GOTO1ØØ
5Ø [CODEM$:FORJ=1TOLEN(M$)
6Ø [CHRXASC(MID$(M$,J,1)),X,Y
7Ø X=X+8:NEXT
8Ø RETURN
1ØØ GETA$:IFA$=""THEN1ØØ
11Ø [BMGRØ:[CB2K2
```

Program 3: Geometric Pattern

```
1 REM STAR DEMO
1Ø PI=2*↑
2Ø INPUT"{CLR}POINTS WANTED (Ø TO END)";P
   W
21 IFPW=ØTHENEND
22 INPUT"SKIP";SK
23 INPUT"RADIUS <1ØØ ";R
3Ø P=PI/PW
5Ø [BMGR1:[CB2K4:[FBMSØ:[FSCR1
6Ø X=16Ø:Y=1ØØ-R:TL=Ø
7Ø FORJ=1TOPW
8Ø TH=TL+SK
9Ø TL=TH:TH=TH*P-(PI/4)
1ØØ X2=COS(TH)*R+16Ø
11Ø Y2=SIN(TH)*R+1ØØ
12Ø [DRAWX,Y,X2,Y2
13Ø X=INT(X2):Y=INT(Y2):NEXT
14Ø GETA$:IFA$=""THEN14Ø
15Ø [BMGRØ:[CB2K2:PRINT"{CLR}":GOTO2Ø
```

Program 4: Joystick-Controlled Sprites

```
1 REM DOODLE
5 GOSUB9ØØ:[DSPR1,13,Ø,Ø,16Ø+16,1ØØ+44,Ø,
  Ø:GOSUB14Ø
1Ø [BANKØ:[CB2K4:[BMGR1:[FBMSØ:[FSCR1:[BS
   PP1,1
2Ø E=1:X=16Ø:Y=1ØØ:C=-1:FORQ=1TO1ØØ:NEXT
3Ø IFPEEK(2Ø3)=6ØTHEN13Ø
31 IFPEEK(2Ø3)=4THENE=-E:IFE>ØTHEN[DSPR1,
   13,Ø,Ø,Ø,Ø,Ø,Ø
32 IFE<ØTHEN[DSPR1,13,Ø,Ø,X+16,Y+44,Ø,12:
   [CLPXX,Y
35 JV=PEEK(5632Ø):FR=JVAND16
4Ø JV=15-(JVAND15)
5Ø IFJV=ØANDFR=16THEN3Ø
6Ø IFJV=1ORJV=5ORJV=9THENY=Y-1:IFY<ØTHENY
   =199
7Ø IFJV=2ORJV=6ORJV=1ØTHENY=Y+1:IFY>199TH
   ENY=Ø
8Ø IFJV>=4ANDJV<=6THENX=X-1:IFX<ØTHENX=31
   9
9Ø IFJV>=8ANDJV<=1ØTHENX=X+1:IFX>319THENX
   =Ø
1ØØ IFFR=ØANDJV=ØTHENC=-C:E=1:FORQ=1TO1ØØ
    :NEXT:IFC>ØTHEN[KSPR1:POKE53288,Ø
1Ø5 IFE<ØTHEN[ESPR1:[MOVE1,X+16,Y+44:[CLP
    XX,Y:GOTO3Ø
11Ø IFC>ØTHEN[PLOTX,Y:GOTO3Ø
12Ø IFC<ØTHEN[ESPR1:[MOVE1,X+16,Y+44:GOTO
    3Ø
13Ø [BANKØ:[BMGRØ:[CB2K2:POKE198,Ø:PRINT"
    {CLR}":[KSPR1:END
14Ø PRINT"{CLR}DOODLE 64"
15Ø PRINT"{DOWN}USE JOYSTICK IN PORT 2"
16Ø PRINT"BUTTON TURNS INK ON/OFF"
165 PRINT"F1 TURNS ERASE MODE ON/OFF"
17Ø PRINT"HIT A KEY TO START"
18Ø PRINT"HIT {RVS}SPACE{OFF} TO STOP"
185 PRINT"THE BLACK + IS YOUR CURSOR WHEN
     INK=OFF"
186 PRINT"THE GREY + IS YOUR CURSOR WHEN
    {SPACE}ERASE=ON":[BKGD1:[FCOLØ
19Ø GETA$:IFA$=""THEN19Ø
2ØØ IFA$=" "THENRETURN
21Ø RETURN
9ØØ X=13*64
91Ø READY:IFY<ØTHENRETURN
92Ø POKEX,Y:X=X+1:GOTO91Ø
1ØØØ DATA1,192,Ø,1,192,Ø,1,192,Ø,1,192,Ø,
     1,192,Ø
1Ø1Ø DATAØ,128,Ø,126,63,Ø,Ø,128,Ø,1,192,Ø
     ,1,192,Ø
1Ø2Ø DATA1,192,Ø,1,192,Ø,1,192,Ø,Ø,Ø,Ø,Ø,
     Ø,Ø
1Ø3Ø DATAØ,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø
1Ø4Ø DATAØ,Ø,Ø,-1
```

Program 5: Sprite Animation

```
1 REM FALLING SHAMROCKS
2 REM HIT A KEY TO STOP PROGRAM
5 [EXTC13:[CB2K4:[BMGR1:[FSCR5:[FBMS171
1Ø X=832:V=53265:R=128
2Ø READA:IFA<ØTHEN35
3Ø POKEX,A:X=X+1:GOTO2Ø
35 FORJ=ØTO7
4Ø [DSPRJ,13,1,1,Ø,Ø,Ø,5+J{2 SPACES}:NEXT
5Ø FORJ=1TO256:FORK=1TO8:[MOVEK-1,J+K*K,J
   *K+K:NEXT:WAITV,R:[FSCRJ/2
55 GETA$:IFA$<>""THEN3ØØ
56 NEXT
6Ø X=PEEK(8192)+1:[FBMSX:GOTO5Ø
1ØØ DATAØ,1Ø2,Ø,Ø,255,Ø,1,255,128,3,255,1
    92
11Ø DATA3,255,192,25,255,152,6Ø,126,6Ø,12
    6,126,126
12Ø DATA255,6Ø,255,255,255,255,127,255,25
    4,255,255,255,255
13Ø DATA24,255,126,24,126,6Ø,24,6Ø,24,24,
    24,Ø,24,Ø,Ø,24,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,-1
3ØØ [CB2K2:[BMGRØ:FORJ=ØTO7:[KSPRJ:NEXT
```

Program 6: Simple PET Emulator

```
1Ø REM ROUTINE TO SET BASIC MEMORY AND SC
   REEN TO PET STANDARD LOCATIONS
2Ø REM SCREEN AT 32768
3Ø REM BASIC 1Ø24 TO 32767
4Ø REM ASSUME IN C-64 STANDARD MAP
5Ø [FSCR Ø:[VS1K Ø :[BANK 2 :PRINT"{CLR}"
6Ø POKE44,4:POKE 45,3:POKE46,4
7Ø POKE55,Ø:POKE56,128
8Ø NEW
```

©

Look at these Features

- Fully screen-oriented
- Horizontal and vertical scrolling
- Terminal mode — never seen before on a wordprocessor
- Supports Commodore disk and cassette handling
- Imbedded commands

Bliztext

Wordprocessor for Commodore 64

BLIZTEXT is a trademark of ELCOMP PUBLISHING, INC.

Commodore-64 and VIC-20 are trademarks of Commodore Business Machines.

Dealer and Distributor inquiries are invited.

BLIZTEXT – SUPER WORDPROCESSOR for the Commodore-64
— ON SALE NOW! —

- Fully screen-oriented, up/down, left and right scrolling – Upper and lower case
- More than 70 commands
- Full I/O compatibility with Commodore peripherals Upper and lower case
- Works with practically every printer on the market, user definable printer control commands
- INCLUDE command allows handling large files on up to 4 diskettes or on cassette.
- Build in terminal software for electronic mail and networking. Telecommunications mode, upload and download, save on disk or cassette.
- Dynamic formatting, Imbedded commands
- Single keystroke for disk directory and error channel
- Program comes on disk or cassette
- Double line spacing, left and right margin justification, centering, page numbering, and practically everything one expects from a good wordprocessor.

AVAILABLE NOW!

Order #4965	$89.00
Manual only (62 pages)	$29.95

MACROFIRE
Editor/Assembler for the Commodore-64
ON SALE NOW
AVAILABLE IMMEDIATELY

One outstanding tool, consisting of 3 powerful elements combined into one efficient program!

1.) Fully screen-oriented Editor (more than 70 commands)
2.) Very fast assembler with macro capability
3.) Machine Language Monitor

Assembly can be started from the editor. Translates in 3 passes. More than 1,000 lables, screen oriented/no line numbers, scrolling, includes disk files.
Practically everything the serious machine language programmer needs everyday!

Manual only	$19.95
Order #4963	$89.00

THE GREAT BOOK OF GAMES, VOL.I,
by Franz Ende
46 programs for the Commodore 64
Introduction to graphics and sound. How to program your own games. Walking pictures, animation, high resolution graphics, programming tips and tricks, hints and useful subroutines for the beginner and advanced programmer. This book is a MUST for every C-64 owner.

Come and get it – It's yours for only		$9.95
Order #182	128 pages	$9.95
Programs from the book on disk.		
Order #4988		$19.95

MORE ON THE SIXTYFOUR, by H.-C. Wagner
How to get the most out of your powerful Commodore 64. Very important subroutines, tricks and hints in machine language for your C-64. How to modify DOS. How to connect a parallel and serial printer. How to design your own terminal program for communication and networking. Dig into I/O for cassette and disk.

Order #183	$9.95
Programs from the book on disk	
Order #4989	$19.95

NEW PRODUCTS
Watch out for our new books, software and add-ons to come soon. ON SALE NOW! -- ORDER TODAY!

How to program in 6502 Machine Language on your C-64, by S. Roberts (Introduction)
Order-#184 $12.95

Commodore-64 Tune-up, Vol. I, by S. Roberts
How to expand and customize your C-64.
Order #185 $12.95

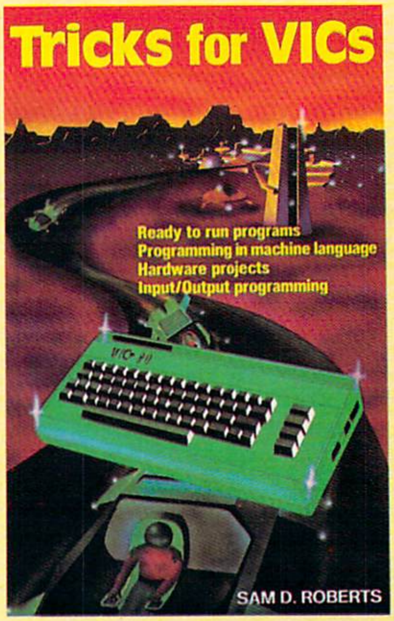
Small Business Programs for the Commodore-64
by S. Roberts
How to make money using your C-64. Mailing list, invoice writing, inventory, simple wordprocessing and much more.
Order # 186 $12.95

Hardware Add-Ons:

Parallel printer interface KIT	Order #4990	$ 19.95
Direct Connect Modem KIT	Order #4991	Ask f.price
Universal Experimenter Board	Order #4970	$ 9.95
Expansion Board, space for four experimenter boards(board only)	Order #4992	$ 29.95

For your VIC-20

Tricks for VICs	$ 9.95
Universal Experimenter board	



HOFACKER

PAYMENT: check, money order, VISA, MASTER CARD, Eurocheck, ACCESS, Interbank
Prepaid orders add $3.50 for shipping (USA)
$5.00 handling for C.O.D.
All orders outside USA: add 15 % shipping, California residents add 6.5 % sales tax.

ELCOMP PUBLISHING, INC
53 Redrock Lane
Pomona, CA 91766
Phone: (714) 623 8314
Telex: 29 81 91

ELCOMP PUBLISHING, INC.
53 Redrock Lane
Pomona, CA 91766
USA
Phone: (714) 623-8314
Telex: 29 81 91

ELCOMP Computer (S) Pte. Ltd.
89 Short Street
Unit 03-07, Golden Wall Auto Centre
SINGAPORE 0718
Phone: 3382623, 3388228
Telex: 56516

Ing.W. Hofacker GmbH
Tegernseerstr. 18
D-8150 Holzkirchen
West-Germany
Phone: 08024 / 73 31
Telex: 52 69 73

MLX

Machine Language Entry Program For Atari And Commodore 64

Charles Brannon, Program Editor

Even the best typists have problems entering machine languare programs as BASIC loaders. Here's the solution.

Have you ever typed in a long machine language program? Chances are you typed in hundreds of DATA statements, numbers, and commas. You're never sure if you've typed them in right. So you go back, proofread, try to run the program, crash, go back and proofread again, correct a few typing errors, run again, crash, recheck your typing — frustrating, isn't it?

Until now, though, that has been the best way to enter machine language into your computer. Unless you happen to own an assembler and are willing to wrangle with machine language on the assembly level, it is much easier to enter a BASIC program that reads the DATA statements and POKEs the numbers into memory.

Some of these *BASIC loaders*, as they are known, use a *checksum* to see if you've typed the numbers correctly. The simplest checksum is just the sum of all the numbers in the DATA statements. If you make an error, your checksum will not match up. Some programmers make the task easier by calculating checksums every ten lines or so, and you can thereby locate your errors more easily.

Almost Foolproof

"MLX" lets you type in long machine language (ML) listings with almost foolproof results. Using MLX, you enter the numbers from a special list that looks similar to BASIC DATA statements. MLX checks your typing on a line-by-line basis. It won't let you enter illegal characters when you should be typing numbers, such as a lowercase L for a 1 or an O for a 0. It won't let you enter numbers greater than 255, which are not permitted in ML DATA statements. It *will* prevent you from entering the wrong numbers on the wrong line. In short, MLX should make proofreading obsolete!

In addition, MLX will generate a ready-to-use tape or disk file. For the 64, you can then use the LOAD command to read the program into the computer, just as you would with any program. Specifically, you enter:

LOAD "program",1,1 (for tape)

or

LOAD "program",8,1 (for disk)

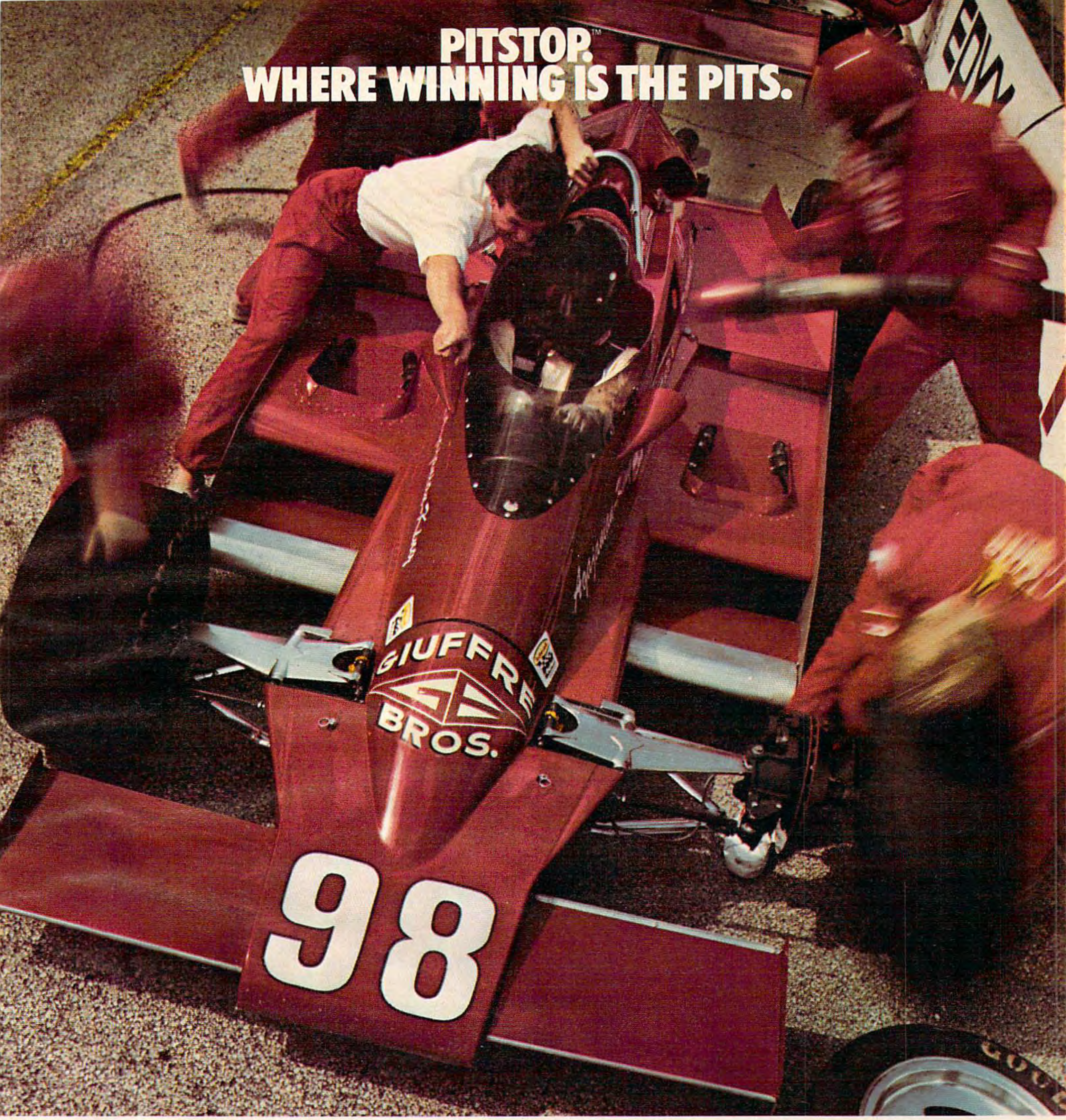
To start the program you need to enter a SYS command that transfers control from BASIC to machine language. The starting SYS will always be given in the article accompanying the machine language program.

For the Atari, MLX will generate a ready-to-use boot tape or boot disk. It also has an option to create binary files for DOS users. A boot disk is like the disks sold with professional games on them. You just insert the disk, remove any cartridges, and turn on your computer. The game will then automatically load.

Boot Tapes

Using a boot tape is almost as simple. Just insert it into your player, rewind, press PLAY. Hold down the START key while turning on your com-

PITSTOP™
WHERE WINNING IS THE PITS.



You'll never make Grand Prix champion just driving in circles.

You've got to stop sometime. The question is when. Right now you're in the lead. But the faster you go, the more gas you consume. And the quicker your tires wear down.

If you do pull into the pits, though, you lose precious seconds. So it's up to you to make sure the pit crew is quick with those tires. And careful with that gas. Otherwise, poof! you're out of the race.

So what'll it be, Mario? Think your tires will hold up for another lap? Or should you play it safe and go get some new ones?

Think it over. Because Pitstop™ is the one and only road race game where winning is more than just driving. It's the pits.

Goggles not included.

One or two players; 6 racecourses, joystick control.

EPYX COMPUTER SOFTWARE

STRATEGY GAMES FOR THE ACTION-GAME PLAYER.

See your retailer for available computer formats.

puter until you hear a beep (like the one you hear with CLOAD). Then press a key on the keyboard and the program will automatically load and run.

Incidentally, the binary file is more useful for utilities than games. Binary files are loaded from the DOS menu (selection L) or automatically if the file is named "AUTORUN.SYS". If you can't stand the thought of putting only one game on each disk (as with boot disks), you can place several binary file machine language games on one disk.

Getting Started

To get started, type in and save MLX (you'll need it for future ML programs published in COMPUTE!). When you're ready to type in the ML program, the program will ask you for several numbers: the starting address and the ending address. In addition, the Atari MLX will request a "Run/Init Address". These vital numbers can be found in the appropriate article accompanying the ML program.

The Atari version will then ask you to press either T for a boot tape, or D for disk. If you press D, you'll be asked if you want to generate a boot disk (press D) or a binary file (press F).

Next you'll see a prompt. The prompt is the current line you are entering from the listing. Each line is six numbers plus a checksum. If you enter any of the six numbers wrong, or enter the checksum wrong, MLX will ring a buzzer and prompt you to reenter the line. If you enter it correctly, a pleasant bell tone will sound and you proceed to the next line.

A Special Editor

You are not using the normal Atari or Commodore 64 screen editor with MLX. For example, it will accept only numbers as input. If you need to make a correction, press ‹DEL/BACK S› (Atari) or ‹INST/DEL› (64). The entire number is deleted. You can press it as many times as necessary back to the start of the line. If you enter three-digit numbers as listed, the computer will automatically print the comma and prepare to accept the next number. If you enter less than three digits (by omitting leading zeros), you can press either the comma, space bar, or RETURN key to advance to the next number. When you get to the checksum value, the Atari MLX will emit a low drone to remind you to be careful. The checksum will automatically appear in inverse video; don't worry, it's highlighted for emphasis.

When testing MLX, we've found that it makes entering long listings extremely easy. With the audio cues provided, you don't even have to look at the screen if you're a touch-typist. We have tested MLX with people lacking any computer background whatsoever. No one has ever managed to enter a listing wrong with it.

Done At Last!

When you finish typing (assuming you type the entire listing in one session) you can then save the completed program on tape or disk. Follow the screen instructions. With a boot disk, the Atari version will offer to format the disk. If you press Y (yes), be sure you have a blank disk in drive one — not your program disk! If you get any errors while saving, you probably have a bad disk, or the disk is full, or you made a typo when entering the actual MLX program. (Remember, it can't check itself!)

Command Control

What if you don't want to enter the whole program in one sitting? MLX lets you enter as much as you want, save that portion, and then reload the file from tape or disk when you want to continue. MLX recognizes these few commands:

S: SAVE
L: LOAD
N: New Address
D: Display

For the Atari, hold down the CTRL key while you type the appropriate key. Hold down SHIFT on the 64 to enter a command key. You will jump out of the line you've been typing, so it's best to perform these commands at a new prompt. Use the SAVE command to save what you've been working on. It will write the tape or disk file as if you've finished, but the tape or disk won't work, of course, until you finish the typing. *Remember what address you stop on.* The next time you run MLX, answer all the prompts as you did before, then insert the disk or tape. When you get to the entry prompt, press CTRL-L (Atari) or SHIFT-L (64) to reload the file into memory. You'll then use the New Address command to resume typing.

New Address And Display

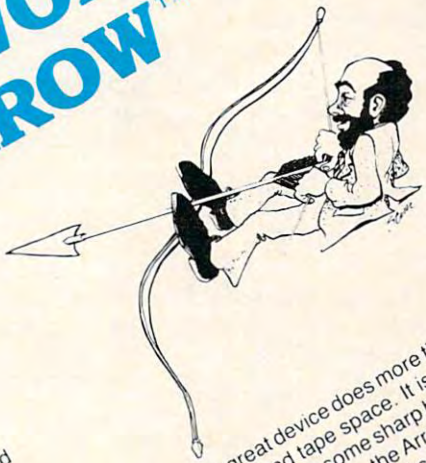
Here's how the New Address command works. After you press SHIFT-N or CTRL-N, enter the address where you previously stopped. The prompt will change, and you can then continue typing. Always enter a New Address that matches up with one of the line numbers in the special listing, or else the checksum won't match up.

You can use the Display command to display a section of your typing. After you press CTRL-D or SHIFT-D, enter two addresses within the line number range of the listing. You can abort the listing by pressing any key.

Tricky Business

The special commands may seem a little confusing at first, but as you work with MLX, they will become easy and valuable. What if you forgot where you stopped typing, for instance? Use the Display

SHOOT THE WORKS with an ARROW™



Have you been spending too much time with your VIC? If it's because you are always waiting for your programs to load, Skyles Electric Works can help.

Enter the Arrow.

This cartridge can load programs off cassette at about eight times the normal speed. It can re-record a program *once* at up to eight times the density on tape. This means that you can get 200 kbytes of programs on a single C-20 tape. In the long run, you'll be able to put 10 tapes into one. The Arrow cartridge will pay for itself in the tape and the vast amount of leading time it saves.

And it's easy to use.

ARROW ...... adds 4 commands: ←S, ←L, ←V, and ←A that are used to SAVE, LOAD, VERIFY AND APPEND at a *high speed* (3600 baud).

ARROW ...... leaves the usual messages that appear on the screen.

ARROW ...... adds a message telling you the length of the program that has been loaded (or saved).

ARROW ...... keeps all of the usual shorthand features of program naming.

ARROW ...... starts automatically when you turn on your VIC-20 or CBM-64.

It gives the length of the program that it has loaded or saved. It can handle hexadecimal/decimal calculations; it even has a machine language monitor in it. Bull's-eye!

And advanced programmers can get an optional machine language assembler in the same cartridge.

Even though it speeds up program loading it does not effect the tape recorder. It's an *extremely* reliable device.

If you want lightning fast cassette operations, shoot for the Arrow.

But this great device does more than cut down on loading time and tape space. It is a fine programmer's aid, too. With some sharp but simple program text editing commands, the Arrow might be the best bargain around in personal software for your VIC. The Arrow is always on target with auto line numbering and line deleting features.

ARROW ...... adds the following programming aids:
- *AUTOLINE* numbering
- *RENUMBERING* of lines
- *DELETE* a line or group of lines
- *SEARCH* with optional replace
- *MOVE* a block of memory
- *COMPARE* two blocks of memory
- *HEXADECIMAL/DECIMAL* calculator converter
- *XIT* two machine language monitor with: Display, Go, Register, Save, Load and Exit commands

Only $49.95 for the VIC or Commodore 64 version. The high-speed cartridge with machine language assemble costs just $89.95. And the easy-to-use Arrow comes with the clear Skyles manual.

AR20 (VIC-20), AR64 (CBM-64) $49.95
ARAS20 (VIC-20), ARAS64 (CBM-64) . 89.95

For CBM-64 and VIC owners only:

This is just 1 of 20 pages of the newest and biggest Skyles catalog, hot off the press.

We know you'll want this page, in its full 7¼ x 9 splendor, and another 19 pages of peripherals, software and books that will make your CBM-64 or VIC computer even nicer to live with. So, if we missed sending you your very own copy within the last few weeks, call us at **(800) 227-9998 (unless you live in California, in which case call (415) 965-1735.**

From Skyles Electric Works, the oldest and largest professional specialists in the business.

Skyles Electric Works
231-E S. Whisman Road
Mountain View, California 94041

command to scan memory from the beginning to the end of the program. When you see a bunch of 170s (64) or zeros (Atari), stop the listing by pressing a key and continue typing where the 170s (or zeros) start. Some programs contain many sections of these zeros or 170s. To avoid typing them, you can use the New Address command to skip over these blocks. Be careful, though; you don't want to skip over anything you *should* type.

Making Copies

You can use the MLX SAVE and LOAD commands to make copies of the completed ML program. Use LOAD to reload the tape or disk, then insert a new tape or disk and use the SAVE command to make a new copy.

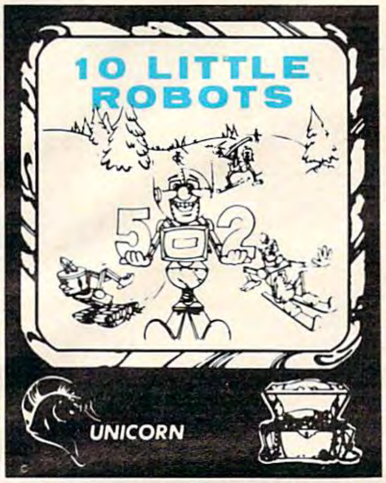
One quirk about tapes made with the 64 MLX SAVE command: When you load them, the message "FOUND program" may appear twice. The tape will load just fine, however.

We hope you will find MLX to be a true labor-saving utility. Since it has been thoroughly tested by entering actual programs, you can count on it as an aid for generating bug-free machine language. And be sure to save MLX; it will be used for future all-machine-language programs in COMPUTE!, COMPUTE!'s Gazette, and COMPUTE! Books.

Program 1: MLX — 64 Version

```
100 PRINT"{CLR}{RED}";CHR$(142);CHR$(8);:
    POKE53281,1:POKE53280,1
101 POKE 788,52:REM DISABLE RUN/STOP
110 PRINT"{RVS}{40 SPACES}";
120 PRINT"{RVS}{15 SPACES}{RIGHT}{OFF}
    E*3£{RVS}{RIGHT} {RIGHT}{2 SPACES}
    E*3{OFF}E*3£{RVS}£{RVS}
    {13 SPACES}";
130 PRINT"{RVS}{15 SPACES}{RIGHT} EG3
    {RIGHT} {2 RIGHT} {OFF}£{RVS}£E*3
    {OFF}E*3{RVS}{13 SPACES}";
140 PRINT"{RVS}{40 SPACES}"
150 V=53248:POKE2040,13:POKE2041,13:FORI=
    832TO894:POKEI,255:NEXT:POKEV+27,3
160 POKEV+21,3:POKEV+39,2:POKEV+40,2:POKE
    V,144:POKEV+1,54:POKEV+2,192:POKEV+3,
    54
170 POKEV+29,3
180 FORI=0TO23:READA:POKE679+I,A:POKEV+39
    ,A:POKEV+40,A:NEXT
185 DATA169,251,166,254,164,255,32,216,25
    5,133,253,96
187 DATA169,0,166,251,164,252,32,213,255,
    133,253,96
190 POKEV+39,7:POKEV+40,7
200 PRINT"{2 DOWN}{PUR}{BLK}{3 SPACES}A F
    AILSAFE MACHINE LANGUAGE EDITOR
    {5 DOWN}"
210 PRINT"E53{2 UP}STARTING ADDRESS?
    {8 SPACES}{9 LEFT}";:INPUTS:F=1-F:C$=
    CHR$(31+119*F)
220 IFS<256OR(S>40960ANDS<49152)ORS>53247
    THENGOSUB3000:GOTO210
225 PRINT:PRINT:PRINT
230 PRINT"E53{2 UP}ENDING ADDRESS?
    {8 SPACES}{9 LEFT}";:INPUTE:F=1-F:C$=
    CHR$(31+119*F)
240 IFE<256OR(E>40960ANDE<49152)ORE>53247
    THENGOSUB3000:GOTO230
250 IFE<STHENPRINTC$;"{RVS}ENDING < START
    {2 SPACES}":GOSUB1000:GOTO 230
260 PRINT:PRINT:PRINT
300 PRINT"{CLR}";CHR$(14):AD=S:POKEV+21,0
310 PRINTRIGHT$("0000"+MID$(STR$(AD),2),5
    );":";:FORJ=1TO6
320 GOSUB570:IFN=-1THENJ=J+N:GOTO320
390 IFN=-211THEN 710
400 IFN=-204THEN 790
410 IFN=-206THENPRINT:INPUT"{DOWN}ENTER N
    EW ADDRESS";ZZ
415 IFN=-206THENIFZZ<SORZZ>ETHENPRINT"
    {RVS}OUT OF RANGE":GOSUB1000:GOTO410
417 IFN=-206THENAD=ZZ:PRINT:GOTO310
420 IF N<>-196 THEN 480
430 PRINT:INPUT"DISPLAY:FROM";F:PRINT,"TO
    ";:INPUTT
440 IFF<SORF>EORT<SORT>ETHENPRINT"AT LEAS
    T";S;"{LEFT}, NOT MORE THAN";E:GOTO43
    0
450 FORI=FTOTSTEP6:PRINT:PRINTRIGHT$("000
    0"+MID$(STR$(I),2),5);":";
451 FORK=0TO5:N=PEEK(I+K):PRINTRIGHT$("00
    "+MID$(STR$(N),2),3);",";
460 GETA$:IFA$>""THENPRINT:PRINT:GOTO310
470 NEXTK:PRINTCHR$(20);:NEXTI:PRINT:PRIN
    T:GOTO310
480 IFN<0 THEN PRINT:GOTO310
490 A(J)=N:NEXTJ
500 CKSUM=AD-INT(AD/256)*256:FORI=1TO6:CK
    SUM=(CKSUM+A(I))AND255:NEXT
510 PRINTCHR$(18);:GOSUB570:PRINTCHR$(20)
515 IFN=CKSUMTHEN530
520 PRINT:PRINT"LINE ENTERED WRONG : RE-E
    NTER":PRINT:GOSUB1000:GOTO310
530 GOSUB2000
540 FORI=1TO6:POKEAD+I-1,A(I):NEXT:POKE54
    272,0:POKE54273,0
550 AD=AD+6:IF AD<E THEN 310
560 GOTO 710
570 N=0:Z=0
580 PRINT"E+3";
581 GETA$:IFA$=""THEN581
585 PRINTCHR$(20);:A=ASC(A$):IFA=13ORA=44
    ORA=32THEN670
590 IFA>128THENN=-A:RETURN
600 IFA<>20 THEN 630
610 GOSUB690:IFI=1ANDT=44THENN=-1:PRINT"
    {LEFT} {LEFT}";:GOTO690
620 GOTO570
630 IFA<48ORA>57THEN580
640 PRINTA$;:N=N*10+A-48
650 IFN>255 THEN A=20:GOSUB1000:GOTO600
660 Z=Z+1:IFZ<3THEN580
670 IFZ=0THENGOSUB1000:GOTO570
680 PRINT",";:RETURN
690 S%=PEEK(209)+256*PEEK(210)+PEEK(211)
691 FORI=1TO3:T=PEEK(S%-I)
695 IFT<>44ANDT<>58THENPOKES%-I,32:NEXT
700 PRINTLEFT$("{3 LEFT}",I-1);:RETURN
710 PRINT"{CLR}{RVS}*** SAVE ***{3 DOWN}"
720 INPUT"{DOWN} FILENAME";F$
730 PRINT:PRINT"{2 DOWN}{RVS}T{OFF}APE OR
    {RVS}D{OFF}ISK: (T/D)"
740 GETA$:IFA$<>"T"ANDA$<>"D"THEN740
750 DV=1-7*(A$="D"):IFDV=8THENF$="0:"+F$
760 OPEN 1,DV,1,F$:POKE252,S/256:POKE251,
```

UNICORN TREASURES MAKE LEARNING A PLEASURE



Each Unicorn educational game teaches as it entertains. All our treasures have been developed and tested at The Computer Learning Center for Children. Written by experts who make them educational and fun, our games feature colorful, high-resolution graphics, multiple difficulty levels, beautiful music, and are completely user-friendly with simple on-screen instructions.

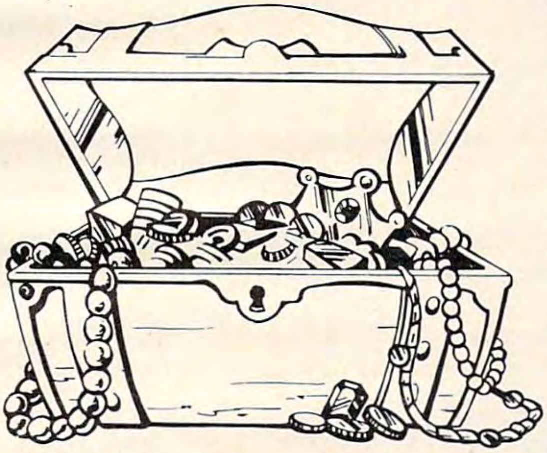
Unicorn's educational games are unique in their flexibility. Parents will be delighted to be able to use them year after year as their child's educational needs change. No need to spend a lot of money on software that children will master in a short time and not use again.

10 LITTLE ROBOTS™— Ages 2-7. The most delightful way to introduce your young learner to the computer. 10 Little Robots has five different games to keep your child's avid attention. There is upper and lower case letter recognition, counting the robots, robot addition, an interactive storybook tale and a unique robot sketch game that will enchant kids in a most creative way. The storybook tale introduces the concept of subtraction and serves as a motivational tool for the beginning reader.

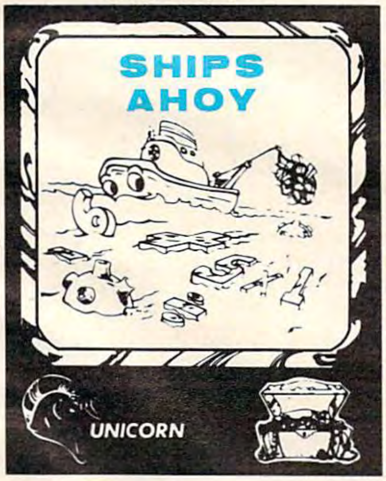
Available for the Apple, Atari, Commodore 64 and IBM computers.

Disk versions only.

IBM version requires color card adapter.

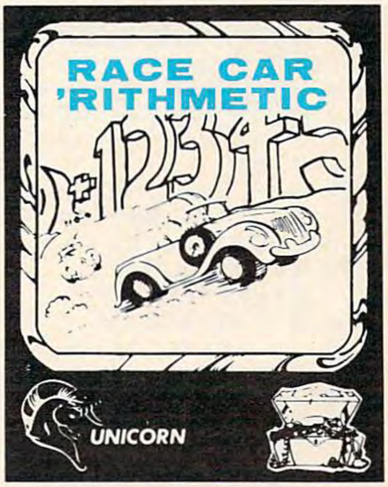


FUNBUNCH™— The most flexible language arts program on the market today. Available on three levels, elementary (grades 1-6), intermediate (junior high school), and college board preparatory (high school). Each level includes over 2000 words and phrases. Within the elementary level the word list can be accessed by grade. The program also allows you to enter your own words and phrases. You can adjust the length of time the words and phrases are displayed on the screen, making Funbunch an excellent tool for *speed reading* as well as *remediation*. There is a built-in printer option which allows you to list the vocabulary for further review. Funbunch also contains a computer doodle drawing game for creative fun. Please specify Funbunch (elementary), Funbunch (intermediate) or Funbunch (college board preparatory) when ordering.



SHIPS AHOY™— Ages 5-13. Outstanding graphics and sound makes this program an entertaining way for children to practice their basic math facts. The object of the game is to sail your ship across the ocean avoiding the treacherous mine hidden beneath the sea. Ships Ahoy allows you to select beginner, intermediate or advanced levels within the four basic math functions. The *flexibility* of this program lies in the option of choosing to be timed or not enabling the academically talented student to practice *speed math*. An equation program and built-in tables enhance the effectiveness of Ships Ahoy. Two unique and different games are included as rewards.

RACE CAR 'RITHMETIC™— Ages 5-Adult. A math game that the entire family can play together. Each person can compete on their own level in a race to the finish line. Children and adults alike will have hour upon hour of educational fun together and can practice the four basic math functions, each choosing their own operation, difficulty level and optional time factor. Race Car 'Rithmetic fully utilizes your computer's superb graphics and sound capabilities. An arcade-style race car game is included as a reward. From 1 to 4 players.



Available at your local computer store. If you can't find them there, you can order directly from Unicorn Software. All programs $39.95. Please enclose $2.00 for shipping and handling. Visa and Mastercard welcomed.

Atari, Commodore 64, IBM PC and Apple are trademarks of Atari, Inc., Commodore Electronics LTD, International Business Machines Corp., and Apple Computer, Inc., respectively.

UNICORN™ SOFTWARE

DEALER INQUIRIES INVITED

UNICORN SOFTWARE™
1775 East Tropicana Avenue
Las Vegas, Nevada 89109
(702) 798-2727 or (702) 798-5990

```
     S-PEEK(252)*256
765 POKE255,E/256:POKE254,E-PEEK(255)*256
770 POKE253,10:SYS 679:CLOSE1:IFPEEK(253)
    >9ORPEEK(253)=0THENPRINT"{DOWN}DONE."
    :END
780 PRINT"{DOWN}ERROR ON SAVE.{2 SPACES}T
    RY AGAIN.":IFDV=1THEN720
781 OPEN15,8,15:INPUT#15,DS,DS$:PRINTDS;D
    S$:CLOSE15:GOTO720
790 PRINT"{CLR}{RVS}*** LOAD ***{2 DOWN}"
800 INPUT"{2 DOWN} FILENAME";F$
810 PRINT:PRINT"{2 DOWN}{RVS}T{OFF}APE OR
     {RVS}D{OFF}ISK: (T/D)"
820 GETA$:IFA$<>"T"ANDA$<>"D"THEN820
830 DV=1-7*(A$="D"):IFDV=8THENF$="0:"+F$
840 OPEN 1,DV,0,F$:POKE252,S/256:POKE251,
    S-PEEK(252)*256
850 POKE253,10:SYS 691:CLOSE1
860 IFPEEK(253)>9 OR PEEK(253)=0 THEN PRI
    NT:PRINT:GOTO310
870 PRINT"{DOWN}ERROR ON LOAD.{2 SPACES}T
    RY AGAIN.{DOWN}":IFDV=1THEN800
880 OPEN15,8,15:INPUT#15,DS,DS$:PRINTDS;D
    S$:CLOSE15:GOTO800
1000 REM BUZZER
1001 POKE54296,15:POKE54277,45:POKE54278,
     165
1002 POKE54276,33:POKE 54273,6:POKE54272,
     5
1003 FORT=1TO200:NEXT:POKE54276,32:POKE54
     273,0:POKE54272,0:RETURN
2000 REM BELL SOUND
2001 POKE54296,15:POKE54277,0:POKE54278,2
     47
2002 POKE 54276,17:POKE54273,40:POKE54272
     ,0
2003 FORT=1TO100:NEXT:POKE54276,16:RETURN
3000 PRINTC$;"{RVS}NOT ZERO PAGE OR ROM":
     GOTO1000
```

Program 2: MLX — Atari Version

```
100 GRAPHICS 0:DL=PEEK(560)+256*PEEK
    (561)+4:POKE DL-1,71:POKE DL+2,6
110 POSITION 8,0:? "MLX":POSITION 23
    ,0:? "failsafe entry":POKE 710,0
    :?
120 ? "Starting Address";:INPUT BEG:
    ? "  Ending Address";:INPUT FIN:
    ? "Run/Init Address";:INPUT STAR
    TADR
130 DIM A(6),BUFFER$(FIN-BEG+127),T$
    (20),F$(20),CIO$(7),SECTOR$(128)
    ,DSKINV$(6)
140 OPEN #1,4,0,"K:":? :? ,"Tape or
    Disk:";
150 BUFFER$=CHR$(0):BUFFER$(FIN-BEG+
    30)=BUFFER$:BUFFER$(2)=BUFFER$:S
    ECTOR$=BUFFER$
160 ADDR=BEG:CIO$="hhh":CIO$(4)=CHR$
    (170):CIO$(5)="LV":CIO$(7)=CHR$(
    228)
170 GET #1,MEDIA:IF MEDIA<>84 AND ME
    DIA<>68 THEN 170
180 ? CHR$(MEDIA):? :IF MEDIA<>ASC("
    T") THEN BUFFER$="":GOTO 250
190 BEG=BEG-24:BUFFER$=CHR$(0):BUFFE
    R$(2)=CHR$((FIN-BEG+127)/128)
200 H=INT(BEG/256):L=BEG-H*256:BUFFE
    R$(3)=CHR$(L):BUFFER$(4)=CHR$(H)
210 PINIT=BEG+8:H=INT(PINIT/256):L=P
    INIT-H*256:BUFFER$(5)=CHR$(L):BU
    FFER$(6)=CHR$(H)
220 FOR I=7 TO 24:READ A:BUFFER$(I)=
    CHR$(A):NEXT I:DATA 24,96,169,60
    ,141,2,211,169,0,133,10,169,0,13
    3,11,76,0,0
230 H=INT(STARTADR/256):L=STARTADR-H
    *256:BUFFER$(15)=CHR$(L):BUFFER$
    (19)=CHR$(H)
240 BUFFER$(23)=CHR$(L):BUFFER$(24)=
    CHR$(H)
250 IF MEDIA<>ASC("D") THEN 360
260 ? :? "Boot Disk or Binary File:"
    ;
270 GET #1,DTYPE:IF DTYPE<>68 AND DT
    YPE<>70 THEN 270
280 ? CHR$(DTYPE):IF DTYPE=70 THEN 3
    60
290 BEG=BEG-30:BUFFER$=CHR$(0):BUFFE
    R$(2)=CHR$((FIN-BEG+127)/128)
300 H=INT(BEG/256):L=BEG-H*256:BUFFE
    R$(3)=CHR$(L):BUFFER$(4)=CHR$(H)
310 PINIT=STARTADR:H=INT(PINIT/256):
    L=PINIT-H*256:BUFFER$(5)=CHR$(L)
    :BUFFER$(6)=CHR$(H)
320 RESTORE 330:FOR I=7 TO 30:READ A
    :BUFFER$(I)=CHR$(A):NEXT I
330 DATA 169,0,141,231,2,133,14,169,
    0,141,232,2,133,15,169,0,133,10,
    169,0,133,11,24,96
340 H=INT(BEG/256):L=BEG-H*256:BUFFE
    R$(8)=CHR$(L):BUFFER$(15)=CHR$(H
    )
350 H=INT(STARTADR/256):L=STARTADR-H
    *256:BUFFER$(22)=CHR$(L):BUFFER$
    (26)=CHR$(H)
360 GRAPHICS 0:POKE 712,10:POKE 710,
    10:POKE 709,2
370 ? ADDR;":";:FOR J=1 TO 6
380 GOSUB 570:IF N=-1 THEN J=J-1:GOT
    O 380
390 IF N=-19 THEN 720
400 IF N=-12 THEN LET READ=1:GOTO 72
    0
410 TRAP 410:IF N=-14 THEN ? :? "New
     Address";:INPUT ADDR:? :GOTO 37
    0
420 TRAP 32767:IF N<>-4 THEN 480
430 TRAP 430:? :? "Display:From";:IN
    PUT F:? ,"To";:INPUT T:TRAP 3276
    7
440 IF F<BEG OR F>FIN OR T<BEG OR T>
    FIN OR T<F THEN ? CHR$(253);"At
    least ";BEG;", Not More Than ";F
    IN:GOTO 430
450 FOR I=F TO T STEP 6:? :? I;":";:
    FOR K=0 TO 5:N=PEEK(ADR(BUFFER$)
    +I+K-BEG):T$="000":T$(4-LEN(STR$
    (N)))=STR$(N)
460 IF PEEK(764)<255 THEN GET #1,A:P
    OP :POP :? :GOTO 370
470 ? T$;",";:NEXT K:? CHR$(126);:NE
    XT I:? :? :GOTO 370
480 IF N<0 THEN ? :GOTO 370
490 A(J)=N:NEXT J
500 CKSUM=ADDR-INT(ADDR/256)*256:FOR
     I=1 TO 6:CKSUM=CKSUM+A(I):CKSUM
    =CKSUM-256*(CKSUM>255):NEXT I
510 RF=128:SOUND 0,200,12,8:GOSUB 57
    0:SOUND 0,0,0,0:RF=0:? CHR$(126)
520 IF N<>CKSUM THEN ? :? "Incorrect
    ";CHR$(253);:? :GOTO 370
530 FOR W=15 TO 0 STEP -1:SOUND 0,50
    ,10,W:NEXT W
540 FOR I=1 TO 6:POKE ADR(BUFFER$)+A
```

YOUR *GIFT GIVING* PROBLEM IS SOLVED!

You can rely on PACE for ONE STOP shopping for all your Micro Computer needs, for gifts, or just for yourself. We have picked the best 2,000 Programs, Books and Accessories—covering all the major brands and put them into one friendly store. And, this is backed up by THOUSANDS of additional items we stock in our central warehouse, ready for overnight shipping to our stores. Magazines? You bet! We carry almost 60 different Micro Magazines on our racks! Plan to visit us soon. Can't visit? Then you can order from our gigantic product list—just write for your FREE copy today!

BIG SAVINGS YEAR 'ROUND!

With PACE, you can save money on all your Micro Computer purchases, each and every day! Just compare and we're sure you'll find we are your BEST source for what you need, and at a price you can afford!

WORD PROCESSING/C-64™

4702-000101	Quick Brn. Fox, C/T	$57.00
5165-000025	Script 64/80 Col. D	$89.00
4100-064207	Easy Script, Disk	$54.95
5841-006401	Totl Text, Tape	$35.95
5841-046401	Totl Text, Disk	$35.95
5841-006402	dTotl Label, Tape	$17.95
5841-046402	Totl Label, Disk	$19.95
5692-000104	Word Pro 3 Plus, D	$99.95
5692-000204	Spellright Plus/64, D	$59.95
5066-000164	Paper Clip, Disk	$100.00

DATABASE/C-64™

5066-000264	Delphi Oracle, D	$120.00
5544-000024	Mirage DataBase, D	$99.00
4538-000401	DataBase Mgr., D	$79.00

UTILITIES/C-64™

4365-004064	Develop 64, Tape	$54.95
4365-404064	Develop 64, Disk	$59.95
4001-000177	Tiny Basic Comp. D	$22.95
4100-064101	Assembler, Disk	$19.95

GAMES/C-64™

5796-003026	Ft. Apocalypse, T	$27.95
5796-004026	Ft. Apocalypse, D	$27.95
5796-003028	Survivor, Tape	$27.95
5796-004028	Survivor, Disk	$27.95
5796-003064	Blue Max, Tape	$27.95
5796-004064	Blue Max, Disk	$27.95
5763-254251	Frogger, Disk	$27.95
5763-254252	Frogger, Tape	$27.95
5763-254201	Crossfire, Disk	$23.95
5763-254202	Crossfire, Tape	$23.95
5763-254803	S. Lightfoot, Cart.	$27.95
4085-002174	Temple Apshai, D	$31.95
4085-002173	Temple Apshai, T	$31.95
4085-005873	Jumpman, Tape	$31.95
4085-005874	Jumpman, Disk	$31.95
5098-000231	Choplifter, Cart.	$35.95
4200-000129	Astroblitz, Cart.	$36.95
4200-000126	Trashman, Cart.	$36.95
4428-000512	Gridrunner, Cart.	$39.95
4770-030002	Critical Mass, D	$31.95
4770-030003	Repton, Disk	$31.95
4770-077010	Type Attack, Cart.	$31.95
5432-245421	Suspended, Disk	$39.95
5432-406408	Planetfall, Disk	$39.95
5432-243444	Witness, Disk	$39.95

BUSINESS/C-64™

5828-000104	Invent. Mgmt.	$69.95
5828-000204	Sales Anal. Mgmt.	$69.95
5828-000304	Accts. Rec./Invoice	$69.95
5828-000404	Accts. Pay/Chkwrite	$69.95
5828-000504	Payroll Mgmt.	$69.95
5828-000604	Cash Flow Mgmt.	$69.95
5828-000704	Gen. Ledger	$69.95
5063-000100	Stock Mgmt., C/D	$29.95
5165-000026	Easy Calc Result, C	$67.15
5165-000028	Calc Result Adv. C	$140.00
4775-000100	Personal Acct., D	$34.95
4775-000101	Personal Acct., T	$29.95
4775-000102	Comp. Mechanic, D	$26.95
4775-000103	Comp. Mechanic, T	$21.95
5190-195925	Home Acct., D	$69.95

5433-000264	Accts. Rec., Disk	$79.95
5433-000364	Accts. Pay., Disk	$79.95
5433-000464	Payroll, Disk	$79.95
5433-000564	Inv. Mgmt., Disk	$79.95
5433-000164	Gen. Ledger, Disk	$79.95

ANY 3 OF ABOVE FOR JUST $229.00!

GAMES/VIC-20™

5763-404203	Crossfire, Tape	$23.95
4770-022010	Type Attack, Cart.	$31.95
5431-005200	Demon Attack, Cart.	$31.95
4085-005263	Ricochet, Tape	$14.95
4085-002103	Temple Apshai, T	$31.95
5842-002304	Sidewinder, Tape	$23.95
5842-002302	Galactic Blitz, T	$19.95
5842-656818	Swarm, Tape	$23.95
5842-002408	Deadly Skies, Cart.	$31.95
4770-022005	Turmoil, Cart.	$31.95
4325-022004	Mutant Herd, Cart.	$31.95
4325-022001	River Rescue, Cart.	$31.95
4325-012002	Jumbo Jet Pilot, Cart.	$31.95
5098-000223	Seafox, Cartridge	$31.95
5098-000224	AE, Cartridge	$31.95
5431-005203	Atlantis, Cart.	$27.95
4428-000307	Shamus, Cart.	$31.95
4115-000372	Paratrooper, Tape	$15.95

PRICES & AVAILABILITY SUBJECT TO CHANGE

BUSINESS/VIC-20™

4200-000136	Home Finance, T	$29.95
4200-000115	Home Inventory, T	$14.95
4200-000109	Car Costs, T	$14.95
4200-000103	Loan Analyzer, T	$14.95
4200-000100	Decision Maker, T	$19.95

GAMES/ATARI®

5796-002008	Ft. Apocalypse, C	$31.95
5796-003008	Ft. Apocalypse, T	$27.95
5796-004008	Ft. Apocalypse, D	$27.95
5796-004016	Necromancer, D	$27.95
5796-003016	Necromancer, T	$27.95
5796-003002	Protector II, T	$27.95
5796-004002	Protector II, D	$27.95
5796-004051	Zeppelin, Disk	$27.95
5796-003051	Zeppelin, Tape	$27.95
5796-003043	Blue Max, Tape	$27.95
5763-152401	Ultima II, Disk	$43.95
5763-154201	Crossfire, Disk	$23.95
5763-154202	Crossfire, Tape	$23.95
5763-154251	Frogger, Disk	$27.95
5763-154252	Frogger, Tape	$27.95
5763-154351	Jawbreaker, Disk	$23.95
5763-154352	Jawbreaker, Tape	$23.95
5763-154801	S. Lightfoot, Disk	$27.95
5763-154803	S. Lightfoot, Cart.	$27.95
4085-002144	Temple Apshai, D	$31.95
4085-002143	Temple Apshai, T	$31.95
5098-000108	Apple Panic, Disk	$23.95
5098-000120	Choplifter, Disk	$27.95
5098-000126	AE, Disk	$27.95
4325-012001	Sub Cmdr., Cart.	$35.95
4325-012005	River Rescue, Cart.	$31.95
5574-392422	Castle Wolfenstein, D	$23.95
5574-197681	Zaxxon, Disk	$31.95
5574-197672	Zaxxon, Tape	$31.95
4070-001906	Miner 2049er, Cart.	$39.95
4770-020008	Type Attack, Disk	$31.95
4770-020010	Critical Mass, D	$31.95
4770-020011	Repton, Disk	$31.95
4770-020014	Wavy Navy, Disk	$31.95
4770-199378	Astrochase, Disk	$37.95
5432-245101	Suspended, Disk	$39.95
5432-245212	Deadline, Disk	$39.95
5432-245309	Starcross, Disk	$31.95
5432-245420	Witness, Disk	$39.95
5432-245424	Planetfall, Disk	$39.95

SPECIAL! BUY ONE, GET ONE FREE!

4325-011010	Jigsaw, Eur. #1, T	$29.95
4325-011006	Humpty Dumpty, T	$29.95
4325-011011	Jigsaw, Eur. #2, T	$29.95
4325-011007	Hickory Dickory, T	$29.95
4325-011014	Comp. 4/Rev., T	$29.95
4325-011008	Jigsaw, Brit. #1, T	$29.95
4325-011013	Figure Fun, T	$29.95
4325-011009	Jigsaw, Brit. #2, T	$29.95

BUSINESS/ATARI®

5098-000081	Bank St. Writer, C	$59.95
5501-028091	Ltr. Pfct. 40/80, D	$139.95
5094-202981	Mil. Stk. Mkt., D	$59.95
5206-001090	Text Wizard, Disk	$79.95
5206-001175	Spell Wizard, D	$49.95
5206-004011	File Mgr., Disk	$79.95

VOLKSMODEM

Lightweight, compact modem. Voice/Data Switch; Full/Half Duplex Switch; 300 Baud; Bell 103 Compatible. Req. 9 Volt Battery (Not Included)

5042-000009 $69.95

INTERFACE CABLES

5042-000021	For Atari	$12.98
5042-000022	For TRS-80, I & III	$12.98
5042-000023	For Commodore	$12.98
5042-000024	For IBM PC	$12.98
5042-000025	For TI 99/4	$12.98

VISIT OUR FIRST STORE—JUST MINUTES WEST OF O'HARE—WE SUPPORT: APPLE®, ATARI®, IBM®, FRANKLIN®, COMMODORE®, RADIO SHACK®, TEXAS INSTRUMENTS®, TIMEX®/SINCLAIR® & More!

PACE MICRO SOFTWARE CENTER
345 East Irving Park Road. Wood Dale, Il 60191
PHONE: (312) 595-3860

APPLE II® · APPLE II®+

5899-119425	Visifile/Visiplot, D	$186.00
5899-580124	Visicalc Bus. Fore., D	$78.00
5899-580014	Visitrend/Plot, D	$223.00
5899-580013	Visiplot, Disk	$156.00
5899-436278	VisiCalc, Disk	$178.00
5899-580016	Visiterm, Disk	$74.40
5899-109385	Visidex, Disk	$186.00
5899-580023	Visischedule, D	$223.00
5777-000101	PFS File, Disk	$100.00
5777-001022	PFS Report, D	$100.00
5777-000403	PFS Graph, D	$100.00
5098-000080	Bank St. Writer, D	$59.95

GAMES/APPLE®

5763-104051	Apple Cider Spider, D	$27.95
5763-104201	Crossfire, Disk	$23.95
5763-104251	Frogger, Disk	$27.95
5763-102401	Ultima II, Disk	$43.95
5763-104801	S. Lightfoot, D	$27.95
4085-002134	Temple Apshai, D	$31.95
4085-005834	Jumpman, Disk	$31.95
5094-202819	Millionaire, D	$47.95
5098-000008	Apple Panic, Disk	$23.95
5098-000020	Choplifter, Disk	$27.95
5098-000022	Seafox, Disk	$23.95
5098-000024	AE, Disk	$27.95
5098-000030	Lode Runner, D	$27.95
5574-392418	Castle Wolfenstein, D	$23.95
5206-197224	Zaxxon, Disk	$31.95
5539-000022	Miner 2049er, D	$31.95
4770-010032	Repton, Disk	$31.95
4770-010033	Type Attack, D	$31.95
4770-010036	Wavy Navy, Disk	$27.95
4770-010037	Critical Mass.	$31.95

DUST COVERS

4108-005600	Vic-20/C-64 Kybrd.	$10.98
4108-005700	Comm. Datasette	$ 4.98
4108-005820	Comm. #1525 Pmtr.	$10.98
4108-005850	Comm. #1541/1540 Disk Drives	$ 9.98
4108-000800	For Atari 400	$11.98
4108-000810	For Atari 410	$ 4.98
4108-000820	For Atari 800	$12.98
4108-000830	For Atari 810	$ 7.98
4108-000845	For Atari 820	$ 8.98
4108-000850	For Atari 822	$ 7.98
4108-000860	For Atari 825	$11.98
4108-000865	For Atari 825 W/Hldr.	$11.98
4108-000600	For Apple III	$15.98
4108-000310	For Apple II	$15.98
4108-000300	For Apple II Kybrd	$ 9.98
4108-000320	For Apple II Single Disk Drive	$ 5.98
4108-000330	For Dbl Disk	$ 9.98

ALSO FOR RADIO SHACK, IBM, TEXAS INSTR. & MORE!

BOOKS/C-64™

4560-000034	Elementary C-64	$14.95
4760-022056	C-64 Prog. Ref. Gde.	$19.95
4105-000020	1st Book of CBM-64	$14.95
4250-000180	More 32 Prog. C-64 (With Diskette)	$29.95
4690-000380	C-64 User's Guide	$14.95
4690-152306	C-64 Computing	$12.95
4690-838136	Sprite Graph./C-64	$15.95
4690-940072	Using C-64 at Home	$10.95
4760-022010	C-64 User's Gde.	$ 9.95
4795-000116	C-64 Basic Hndbk.	$ 9.95
4795-000126	Easy Gde. to C-64	$ 7.95
4800-001640	Graph/Sound Prog.	$14.95

BOOKS/ATARI®

4105-000000	1st Book of Atari	$12.95
4105-000002	Inside Atari DOS	$19.95
4105-000006	2nd Book of Atari	$12.95
4105-000008	1st Bk Atari Graph	$12.95
4105-000015	Atari Basic Srcebk.	$12.95
4198-000022	Computers for Kids	$ 4.95
4198-000034	Creative Atari	$15.95
4250-000084	32 Prog. Atari	$19.95
4250-000172	32 Prog. Atari, (With Diskette)	$34.95
4525-049194	Hands On Basic	$19.95
4525-068579	Learning w/Logo	$14.95

BOOKS/VIC-20™

4105-000007	1st Book Vic-20	$12.95
4105-000013	1st Bk. Vic-20 Games	$12.95
4105-00016	2nd Bk. of Vic-20	$14.95
4250-000059	32 Prog. Vic-20	$19.95
4250-000181	32 Prog. Vic-20, (With Diskette)	$29.95
4410-001057	Vic Graphics	$12.95
4410-001058	Vic Revealed	$12.95
4560-000056	Kids & the Vic	$19.95

BOOKS/Texas Instr.™

4105-000012	Prog. Ref. Gde.	$14.95
4105-000017	1st Bk. of Games	$14.95
4250-000188	32 Basic Prog. (With Diskette)	$34.95
4410-005185	Intro to Basic	$12.95
4525-068580	Learn. W/TI Logo	$14.95
4560-000059	Kids & the TI	$19.95

NEW NEW NEW NEW NEW

"HOW TO" operating manual for the Commodore 1541 Disk Drives. Written by Nancy Wilmont, a professional instructor, in clear, concise, step-by-step format with sample programs. Includes diskette with sample programs. A MUST for Commodore Disk Drive owners!

0001-800001 *Due Early '84* $15.98

COUPON Please Send Me:

QTY	NUMBER	DESCRIPTION	COST	TOTAL
			SHIPPING	$ 2.50
			TOTAL	

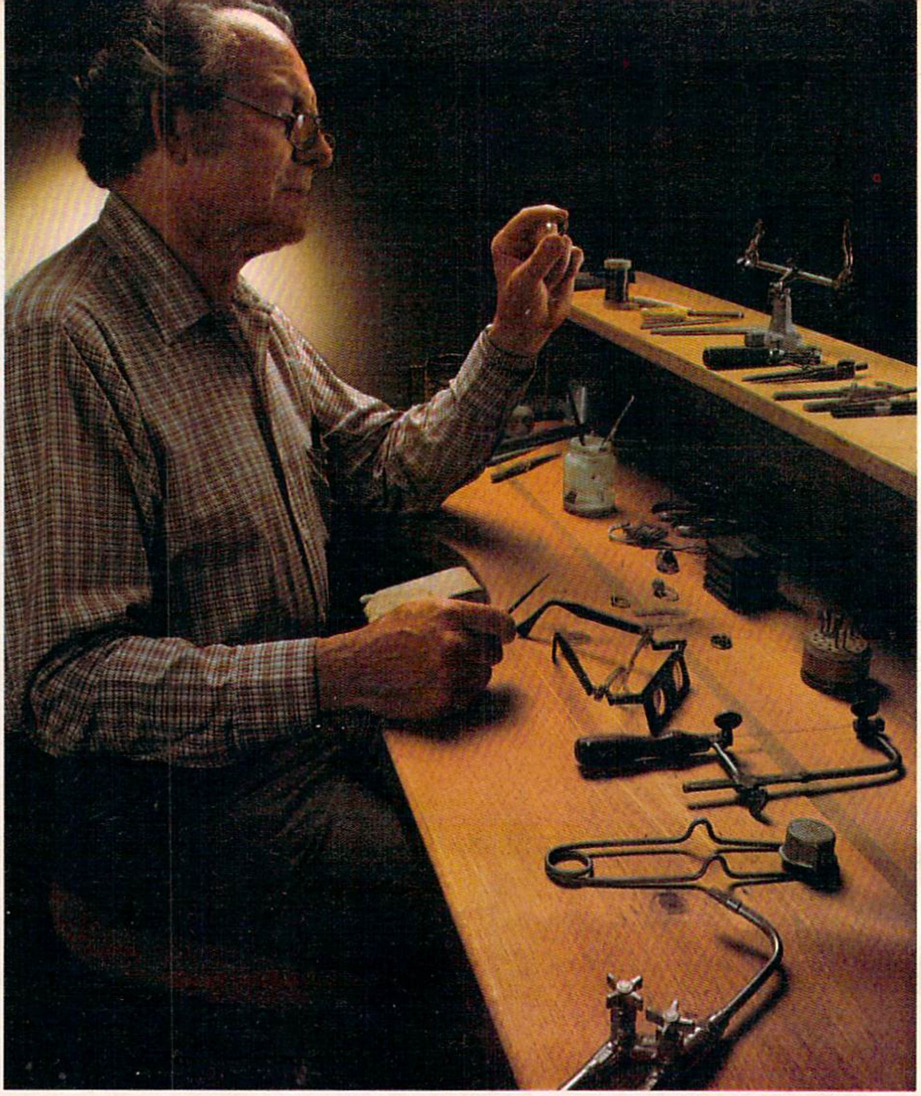
Illinois Residents Please Add 6% Sales Tax.
Foreign Orders. (All outside Continental US). Add 10% Shipping (Minimum $4.00)
Catalogs Shipped Postage Paid

PAYMENT ENCLOSED: ☐CASH ☐CHECK ☐MONEY ORDER
PLEASE CHARGE TO MY: ☐MASTERCARD ☐VISA (Min. Chg. $25)

CARD NUMBER ____________
EXPIRES ____________ INTRBNK # ____________
SHIP TO ____________
STREET ADDRESS ____________ APT ______
CITY ____________ STATE ______ ZIP ______

pace! micro software centers

DEPARTMENT: C-P
Lock Box 328
Bensenville, IL 60106



Craftsmen Need Precision Tools . . . Programmers! Demand Precision Software!

BASIC XL has twice the speed and twice the power of Atari® BASIC. And yet, as befits a fine craftsman's tool, **BASIC XL** is even easier to use and more dependable, while including such outstanding major additions as structured programming, string arrays, programming aids, enhanced graphics, and business capabilities.

Atari BASIC is a good starting point. We should know. We wrote it in 1978. Buy **BASIC XL.** Take advantage of five more years of experience!

So, prepare yourself for some exploration into imaginative programming with **BASIC XL!** Cartridge, excellent tutorial, reference manual . . . $99.

OSS™
Precision Software Tools
1173D S. Saratoga/Sunnyvale Road
San Jose, CA 95129 • (408) 446-3099

Functional, honest, and beautiful describe the simple lines of a crafts man's tools. For the jeweler these tools are an extension of the human hand to better execute complex designs. For you, the programmer, Precision Software Tools keep complications out of your programming while allowing you to produce intricate programs.

See the complete collection of OSS Precision Software Tools!™

MAC/65: The fastest 6502 macro assembler/editor package on cartridge . . . $99.

BUG/65: A powerful debugger. On disk, with OS/A+ . . . $35.

C/65: The first native mode "small c" compiler for Atari and Apple® computers. On disk . . . $80.

ACTION!: The fastest, small computer language ever. A feature-packed cartridge at only . . . $99.

All products on disk include OS/A+ and also require 48 K.

SEE YOUR LOCAL DEALER!

Call or write for informative brochures.

```
    DDR-BEG+I-1,A(I):NEXT I
55Ø ADDR=ADDR+6:IF ADDR<=FIN THEN 37
    Ø
56Ø GOTO 71Ø
57Ø N=Ø:Z=Ø
58Ø GET #1,A:IF A=155 OR A=44 OR A=3
    2 THEN 67Ø
59Ø IF A<32 THEN N=-A:RETURN
6ØØ IF A<>126 THEN 63Ø
61Ø GOSUB 69Ø:IF I=1 AND T=44 THEN N
    =-1:? CHR$(126);:GOTO 69Ø
62Ø GOTO 57Ø
63Ø IF A<48 OR A>57 THEN 58Ø
64Ø ? CHR$(A+RF);:N=N*1Ø+A-48
65Ø IF N>255 THEN ? CHR$(253);:A=126
    :GOTO 6ØØ
66Ø Z=Z+1:IF Z<3 THEN 58Ø
67Ø IF Z=Ø THEN ? CHR$(253);:GOTO 57
    Ø
68Ø ? ",";:RETURN
69Ø POKE 752,1:FOR I=1 TO 3:? CHR$(3
    Ø);:GET #6,T:IF T<>44 AND T<>58
    THEN ? CHR$(A);:NEXT I
7ØØ POKE 752,Ø:? " ";CHR$(126);:RETU
    RN
71Ø GRAPHICS Ø:POKE 71Ø,26:POKE 712,
    26:POKE 7Ø9,2
72Ø IF MEDIA=ASC("T") THEN 89Ø
73Ø REM  DISK
74Ø IF READ THEN ? :? "Load File":?
75Ø IF DTYPE<>ASC("F") THEN 1Ø4Ø
76Ø ? :? "Enter AUTORUN.SYS for auto
    matic use":? :? "Enter filename"
    :INPUT T$
77Ø F$=T$:IF LEN(T$)>2 THEN IF T$(1,
    2)<>"D:" THEN F$="D:":F$(3)=T$
78Ø TRAP 87Ø:CLOSE #2:OPEN #2,8-4*RE
    AD,Ø,F$:? :? "Working..."
79Ø IF READ THEN FOR I=1 TO 6:GET #2
    ,A:NEXT I:GOTO 82Ø
8ØØ PUT #2,255:PUT #2,255
81Ø H=INT(BEG/256):L=BEG-H*256:PUT #
    2,L:PUT #2,H:H=INT(FIN/256):L=FI
    N-H*256:PUT #2,L:PUT #2,H
82Ø GOSUB 97Ø:IF PEEK(195)>1 THEN 87
    Ø
83Ø IF STARTADR=Ø OR READ THEN 85Ø
84Ø PUT #2,224:PUT #2,2:PUT #2,225:P
    UT #2,2:H=INT(STARTADR/256):L=ST
    ARTADR-H*256:PUT #2,L:PUT #2,H
85Ø TRAP 32767:CLOSE #2:? "Finished.
    ":IF READ THEN ? :? :LET READ=Ø:
    GOTO 36Ø
86Ø END
87Ø ? "Error ";PEEK(195);" trying to
     access":? F$:CLOSE #2:? :GOTO 7
    6Ø
88Ø REM  BOOT TAPE
89Ø IF READ THEN ? :? "Read Tape"
9ØØ ? :? :? "Insert, Rewind Tape.":?
     "Press PLAY ";:IF  NOT READ THE
    N ? "& RECORD"
91Ø ? :? "Press RETURN when ready:";
92Ø TRAP 96Ø:CLOSE #2:OPEN #2,8-4*RE
    AD,128,"C:":? :? "Working..."
93Ø GOSUB 97Ø:IF PEEK(195)>1 THEN 96
    Ø
94Ø CLOSE #2:TRAP 32767:? "Finished.
    ":? :? :IF READ THEN LET READ=Ø:
    GOTO 36Ø
95Ø END
96Ø ? :? "Error ";PEEK(195);" when r
    eading/writing boot tape":? :CLO
    SE #2:GOTO 89Ø
97Ø REM CIO Load/Save File#2 opened
     READ=Ø for write, READ=1 for re
    ad
98Ø X=32:REM File#2,$2Ø
99Ø ICCOM=834:ICBADR=836:ICBLEN=84Ø:
    ICSTAT=835
1ØØØ H=INT(ADR(BUFFER$)/256):L=ADR(B
     UFFER$)-H*256:POKE ICBADR+X,L:P
     OKE ICBADR+X+1,H
1Ø1Ø L=FIN-BEG+1:H=INT(L/256):L=L-H*
     256:POKE ICBLEN+X,L:POKE ICBLEN
     +X+1,H
1Ø2Ø POKE ICCOM+X,11-4*READ:A=USR(AD
     R(CIO$),X)
1Ø3Ø POKE 195,PEEK(ICSTAT):RETURN
1Ø4Ø REM  SECTOR I/O
1Ø5Ø IF READ THEN 11ØØ
1Ø6Ø ? :? "Format Disk In Drive 1? (
     Y/N):";
1Ø7Ø GET #1,A:IF A<>78 AND A<>89 THE
     N 1Ø7Ø
1Ø8Ø ? CHR$(A):IF A=78 THEN 11ØØ
1Ø9Ø ? :? "Formatting...":XIO 254,#2
     ,Ø,Ø,"D:":? "Format Complete":?

11ØØ NR=INT((FIN-BEG+127)/128):BUFFE
     R$(FIN-BEG+2)=CHR$(Ø):IF READ T
     HEN ? "Reading...":GOTO 112Ø
111Ø ? "Writing..."
112Ø FOR I=1 TO NR:S=I
113Ø IF READ THEN GOSUB 122Ø:BUFFER$
     (I*128-127)=SECTOR$:GOTO 116Ø
114Ø SECTOR$=BUFFER$(I*128-127)
115Ø GOSUB 122Ø
116Ø IF PEEK(DSTATS)<>1 THEN 12ØØ
117Ø NEXT I
118Ø IF  NOT READ THEN END
119Ø ? :? :LET READ=Ø:GOTO 36Ø
12ØØ ? "Error on disk access.":? "Ma
     y need formatting.":GOTO 1Ø4Ø
121Ø REM
122Ø REM  SECTOR ACCESS SUBROUTINE
123Ø REM Drive ONE
124Ø REM Pass buffer in SECTOR$
125Ø REM sector # in variable S
126Ø REM READ=1 for read,
127Ø REM READ=Ø for write
128Ø BASE=3*256
129Ø DUNIT=BASE+1:DCOMND=BASE+2:DSTA
     TS=BASE+3
13ØØ DBUFLO=BASE+4:DBUFHI=BASE+5
131Ø DBYTLO=BASE+8:DBYTHI=BASE+9
132Ø DAUX1=BASE+1Ø:DAUX2=BASE+11
133Ø REM DIM DSKINV$(4)
134Ø DSKINV$="hLS":DSKINV$(4)=CHR$(2
     28)
135Ø POKE DUNIT,1:A=ADR(SECTOR$):H=I
     NT(A/256):L=A-256*H
136Ø POKE DBUFHI,H
137Ø POKE DBUFLO,L
138Ø POKE DCOMND,87-5*READ
139Ø POKE DAUX2,INT(S/256):POKE DAUX
     1,S-PEEK(DAUX2)*256
14ØØ A=USR(ADR(DSKINV$))
141Ø RETURN
```

©

Jeff and Marilyn Mitchell "designed" their new program themselves. CodeWriter wrote all the computer code. The Mitchells' dream is thriving on fulfilling other people's wishes. Their new home business needs very special information fast: Which fantasies are still open? What's our next completion date? Can we get a list of all fantasies needing out of state travel?

They got it all—with no computer hassle.

And you can too, with CodeWriter. No programming. No 'computerese'. At home or at the office, you create your own programs to handle any information you want—at your fingertips; Payables, receivables, inventory, credit cards, tax details, club or church records—always organized your way.

You work with CodeWriter in plain English. Simply 'draw' any screen layout, add any calculations you'd like done—or help messages you need—and you're done. CodeWriter writes all the BASIC code.

In minutes you've got YOUR OWN PROGRAM on YOUR OWN DISK. You don't need CodeWriter again until you want a new program.

You can begin with **Home FileWriter**™ and expand to more complete business systems with full report and menu design features.

You can get CodeWriter for the Commodore 64®, Atari®, Apple®, IBM PC®, Commodore Business Machine®, Victor 9000®, and Kay Pro II®, computers. Prices range from $69 to $249.

You think this much power can't come this easy? There are thousands of CodeWriter systems in use all over the world—80% are first time computer owners. CodeWriter writes solutions the first time you try!

"This is our first business, our first computer, and our first program—and we really did it ourselves!"

CodeWriter™ A Dynatech Company
Dynatech Microsoftware Inc.

7847 N. Caldwell Ave. Niles, Ill. 60648
Toll-Free 1-800-621-4109 (in Ill. 312-470-0700)

AVAILABLE AT **VideoConcepts**® Your Home Entertainment Store 180 STORES NATIONWIDE

COMPLETE WORD PROCESSING SYSTEM

(Everything you need for word processing — LIST PRICE $1800.00)

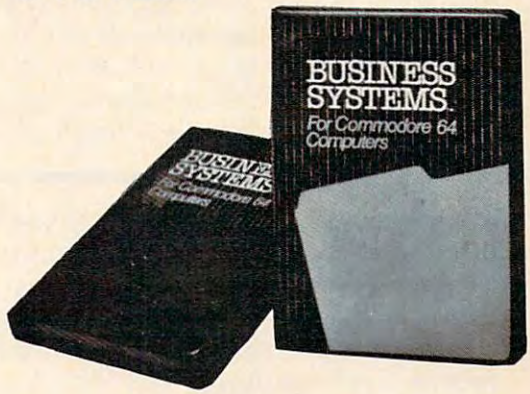
SALE $995.00

COMPLETE SMALL BUSINESS SYSTEM

(Everything you need to computerize your business — LIST PRICE $2200.00)

SALE $1195.00

LOOK AT WHAT YOU GET WITH EACH SYSTEM PACKAGE!!!



- The powerful 84K Commodore 64 Computer! (More features than Apple II)
- 170K Commodore 64 Disk Drive!
- Box of 10 "Loran" Disks!
- Gemini 10X Starmicronics 10" Carriage Deluxe,120CPS, Dot Bit Addressable Tractor-Friction Printer!
- Deluxe Cardco Printer Interface!
- Box of Printer Paper!
- Your choice of 12" Green Screen or Amber Screen Monitor!

The $995 complete word processing system includes: "Script-64 Executive Word Processor Program, 80 columns in color, 20,000 word customizable dictionary, powerful mail merge" — List Price $130)

The $1195 complete small business system includes: "General Ledger, Accounts Payable and Check Writing, Accounts Receivable, Payroll, Inventory, Database Manager" — List Price $595)

15 DAY FREE TRIAL We give you 15 days to try out these SUPER SYSTEM PACKAGES!! If it doesn't meet your expectations, just send it back to us prepaid and we will refund your purchase price!!

90 DAY IMMEDIATE REPLACEMENT WARRANTY If any of the SUPER SYSTEM PACKAGE equipment or programs fail due to faulty workmanship or material we will replace it IMMEDIATELY at no charge!

Add $50.00 for shipping and handling!!

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

WE DO NOT EXPORT TO OTHER COUNTRIES EXCEPT CANADA.

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

PROTECTO
ENTERPRIZES (WE LOVE OUR CUSTOMERS)

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

List And Scroll For The VIC And 64

Tom Forsythe

This utility program — an excellent tool for debugging BASIC programs — separates a BASIC listing into single statements, and sets off FOR-NEXT loops and IF-THEN statements for readability. You can also scroll in either direction to scan the listing.

Are you tired of typing LIST or trying to read BASIC statements that are lumped together on the same line? This machine language program allows listing and scrolling of BASIC statements. It prints each statement on a separate line and provides indents during FOR-NEXT loops and after IF-THEN statements, making your BASIC listing more readable.

For example, a normal screen listing looks like this on a VIC:

```
10 A=10:FORJ=1TO4:FORI
=0TO10:PRINTI;:PRINTA*
B:NEXTI:PRINT"PASS "J"
OK":B=A+B:NEXTJ:IFJ=AT
HENA=B:GOTO5:END
```

With "List And Scroll' it would look like this:

```
10
  A=10:
  FORJ=1TO4:
     FORI=0TO10:
        PRINTI;:
        PRINTA*B:
        NEXTI:
     PRINT"PASS "J"OK":
     B=A+B:
     NEXTJ:
  IFJ=ATHENA=B:
     GOTO5:
     END
```

Simple Operation

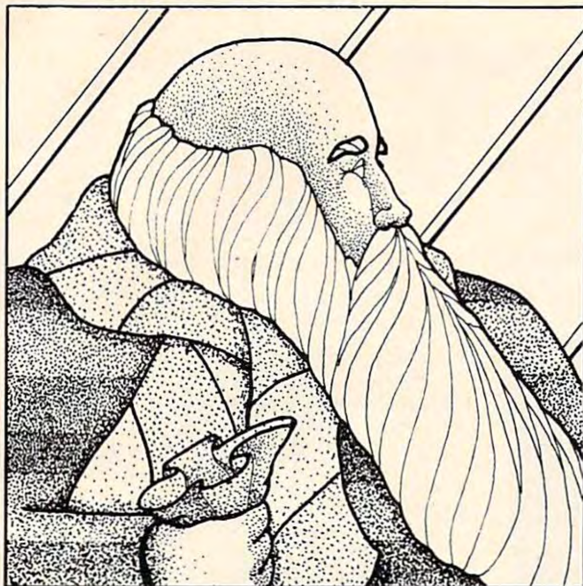
Operation is easier and faster than the normal LIST; just type a period (.) followed by an optional line number. Without the line number, the listing will begin with the first line of your BASIC program. To scroll forward or backward through the listing, use the cursor up or down keys. Pressing the RETURN key or scrolling past either end of the BASIC program will automatically return control to BASIC. You'll know this by the presence of a flashing cursor.

After typing in Program 1 (VIC version) or Program 2 (64 version), be sure to SAVE it to tape or disk. Then you must do one of the following: Type SYS 6769 or type in, SAVE, and RUN Program 3. The first option is fine if the BASIC program you'd like to examine with List And Scroll is not more than 2K (2673 bytes) for the VIC, or 4K (4021 bytes) for the 64. However, you must use Program 3 if your BASIC program exceeds the limits mentioned above.

If you SYS 6769 and your BASIC program is too long, it will write over List And Scroll and render it useless. So, if in doubt, use Program 3. After you type RUN, there will be a short wait and then you'll see a command to SYS to a specified address. Program 3 moves the program to a safe location at the top of memory. VIC users should remove the Super Expander cartridge before using Program 3.

Program 1: List And Scroll (VIC Version)

```
10 I=6768
20 READ A:IF A=256 THEN 40
30 POKE I,A:CK=CK+A:I=I+1:GOTO 20
40 IFCK<>51983THENPRINT"{CLR}ERROR IN DAT
   A STATEMENTS":END
50 END
6768 DATA 1,113,26,173,113,26,133
6776 DATA 55,133,51,173,114,26,133
6784 DATA 56,133,52,234,234,234,169
6792 DATA 76,133,124,173,147,26,133
```



VICTORY SOFTWARE

WOULD LIKE TO WISH OUR CUSTOMERS

H • A • P • P • Y

H O L I D A Y S

AND THANK THEM FOR THEIR PATRONAGE THROUGHOUT THE YEAR.

FEATURING PROGRAMS FOR THE VIC-20 AND THE COMMODORE 64.

BOUNTY HUNTER $19.95

An adventure in the Old West. Journey back with us into the days of Jessie James and Billy the Kid where the only form of justice was a loaded revolver and a hangman's noose. In this full-length text adventure, you play the role of Bounty Hunter, battling against ruthless outlaws, hostile Indians, wild animals and the elements of the wilderness with only your wits and your six gun. Average solving time: 20-30 hours. If you love adventures, this one is a real treat.
Available for **COMMODORE 64** and the **VIC-20** (with 8K or 16K expander). Available on **TAPE** or **DISK**. Played with **JOYSTICK**.

KONGO KONG $19.95

Climb ladders, avoid the barrels the crazy ape is rolling at you, and rescue the damsel. Commodore 64 version features 4 different screens!
Available for **COMMODORE 64** and **VIC-20**. Available on **TAPE** or **DISK**. Played with **JOYSTICK**.

GRAVE ROBBERS $14.95

Introducing the first GRAPHIC ADVENTURE ever available for the VIC-20 or COMMODORE 64! With realistic audio-visual effects, you explore an old deserted graveyard and actually see the perils that lie beyond.
Available for **COMMODORE 64** and **VIC-20**. Available on **TAPE** or **DISK**. Played with **KEYBOARD**.

CHOMPER MAN $19.95

Don't let the bullies catch you as you gobble the goodies! This program has 8 screens and still fits in the standard memory.
Available for **COMMODORE 64** and **VIC-20**. Available on **TAPE** or **DISK**. Played with **JOYSTICK** or **KEYBOARD**.

THE • EARTH • WARRIOR • SERIES

METAMORPHOSIS $19.95

You stumbled into the nest of the Cyglorx and find yourself fighting off robot tanks guarding the Cyglorx eggs. You think you have everything under control and then the eggs start hatching.
Available for **COMMODORE 64** and **VIC-20**. Available on **TAPE** or **DISK**. Played with **JOYSTICK**

CREATOR'S REVENGE $19.95

The creator assembled a massive army of robots and insects to take revenge on the earth. Destroy insects, get treasures, and get the neutron bomb deactivator. Battle robots and destroy the neutron bomb before it annihilates your city. Miss and you must face the mutants. Features 4 different screens.
Available for **COMMODORE 64**. Available on **TAPE** or **DISK**. Played with **JOYSTICK**.

LABYRINTH OF THE CREATOR $19.95

Journey into the most complex and dangerous fortress ever built by the creator. You will encounter deadly robots, skulls, lakes, avalanches, false creators, and a creature who roams 256 rooms relentlessly pursuing you.
Available for **COMMODORE 64**. Available on **TAPE** or **DISK**. Played with **JOYSTICK**.

ILLUSTRATIONS: ELIZABETH HAUCK

Check your LOCAL DEALER or order directly.
ORDERING: We accept personal checks, money orders, VISA, and MasterCard. Charge orders please include number and expiration date.
OVERSEAS ORDER: Please use charge, or have check payable through a U.S. bank.
CANADIAN CUSTOMERS: If you wish to write a check drawn through a Canadian bank, please multiply the total order by 1.25 for proper conversion.
Add $1.50 postage and handling per order. PA residents please add 6% sales tax.

VICTORY SOFTWARE INC.
7 Valley Brook Road
Paoli, Pennsylvania 19301
(215) 296-3787

Fast, Fast Relief From Expensive Software Blahs.

COMMODORE 64

HESWARE

	List Price	But for You!
Coco (CT)	$45	$29
Maze Master (CT)	$35	$23
Turtle Tutor (CT)	$40	$26
Turtle Trainer (CT)	$40	$26
Paintbrush (CT)	$30	$19
Home Manager (D)	$50	$33
Time & Money Manager (D)	$70	$45
Hesman 64 (CT)	$40	$26
Gridrunner (CT)	$40	$26
64 Forth (CT)	$60	$39
Turtle Graphics II (CT)	$60	$40
Multiplan (D)	$100	$79
Omnicalc (D)	$50	$37
Bengi (D&C)	$45	$28
Attack Mutant Camel (CT)	$30	$19

SEGA

Congo Bongo (CT)		
Buck Rogers (CT)	$40	$25
Star Trek (CT)		

SYNAPSE

Protector (D&C)		
Shamus (D&C)		
Sentinel (D&C)		
Morgol (D&C)		
Drelbs (D&C)		
Fort Apocalypse (D&C)		
Pharoah's Curse (D&C)	$35	$22
Protector II (D&C)		
Blue Max (D&C)		
Survivor (D&C)		
Shamus II (D&C)		

INFOCOM

Planet Fall (D)		
Witness (D)		
Enchanter (D)	$50	$33
Infidel (D)		

SPINNAKER

Alphabet Zoo (CT)	$35	$23
Cosmic Life (CT)	$35	$23
Delta Drawing (CT)	$40	$26
Delta Music (CT)	$40	$26
Facemaker (CT)	$35	$23
Fraction Fever (CT)	$35	$23
Kids on Keys (CT)	$35	$23
Kindercomp (CT)	$30	$19
Story Machine (CT)	$40	$26
Up For Grabs (CT)	$40	$26

BRODERBUND

Choplifter (CT)	$45	$29
Seafox (CT)	$40	$26
Serpentine (CT)	$40	$26
David Midnight Magic (D)	$35	$23
Bank St. Writer (D)	$70	$45
Lode Runner (D)	$35	$23
Spare Change (D)	$35	$23
Drol (D)	$35	$23

CREATIVE

Astroblitz (CT)		
Save New York (CT)	$35	$22
Pipes (CT)		
Spitball (CT)		

COMMODORE 64 Cont'd.

EPYX

Jumpman D&C	$40	$25
Temple of Apshai (D&C)	$40	$25
Upper Reaches Apshai (D&C)	$20	$13
Curse of Ra (D&C)	$20	$13
Starfire & Fire One (D&C)	$40	$29
Jumpman Junior (CT)	$40	$29
Lunar Outpost (D&C)	$40	$25
Dragonriders of Pern (D&C)	$40	$25
Gateway to Apshai (C&D)	$40	$25
Pitstop (C&D)	$40	$25
Crush, Crumble, Chomp (C&D)	$30	$19
Fun With Art (CT)	$40	$27
Fun With Music (CT)	$40	$27
Facts (D&C)	$30	$19

"$u¢h-A-Deal! $oftware."
903 S. Rural Rd. #102
Tempe, AZ 85281

TO ORDER FAST, FAST CALL TOLL FREE: 1-800-431-8697

For Information Only Call (602) 968-9128

We sell mail order only!
Dealer Inquiries Invited

COMMODORE 64 Cont'd.

And Incredible Savings On:

Data 20 Word Manager (C)	Just $27
Data 20 Business Manager (D)	$89
Timeworks Data Manager (D&C)	$17
Timeworks Money Manager (D&C)	$17
Timeworks Programmer Kit (D&C)	$17
Continental Home Accountant (D)	$49
Lightning Mastertype (D)	$27
Sirius Wayout (D)	$27
Sirius Gruds in Space (D)	$23
Sublogic Pinball (C)	$20
Acess Neutral Zone (D&C)	$23
Practicalc (D&C)	$35
Programmable Spreadsheet	$55
Datamost Kids & The 64 (book)	$14
Commodore 64 Ref. Guide (book)	$17
Access Beachead (D&C)	$23
Datasoft Pooyan (D&C)	$19
Datasoft Zaxxon (D&C)	$25
Datamost Swashbuckler (D)	$25
Datamost Aztec (D)	$25
Sierra OnLine Threshhold (CT)	$27
Timeworks Programmer Kit II (D&C)	$17

COMMODORE VIC 20

HESWARE

	List Price	But For You!
Robot Panic (CT)	$20	$13
Slime (CT)	$20	$13
Predator (CT)	$20	$13
Coco II (C&D)	$20	$14
Lazer Zone (CT)	$30	$19
Necromancer (CT)	$30	$19
Pharoah's Curse (CT)	$30	$19
Attack Mutant Camel (CT)	$30	$19
Kindercomp (CT)	$40	$26
Facemaker (CT)	$40	$26
Story Machine (CT)	$40	$26

SEGA

Congo Bongo (CT)		
Buck Rogers (CT)	$40	$25
Star Trek (CT)		

BRODERBUND

A.E. (CT)		
Seafox (CT)		
Mastertype (CT)	$40	$24
Lode Runner (CT)		

EPYX

Temple of Apshai (C&D)	$40	$24
Sword of Fargoal (C)	$30	$19

WIZWARE

Electronic Party (C)		
Square Pegs (C)		
Turtle Tracks (C)	$30	$19
Your Vic 20 (C)		

SIERRA ON LINE

Lunar Leeper (CT)		
Crossfire (CT)		
Threshhold (CT)	$30	$18
Cannonball Blitz (CT)		

CREATIVE

Choplifter (CT)	$30	$19
Home Office (C)	$30	$19
Home Office (D)	$35	$22

ORDERING & TERMS: Send cashier check, money order; personal/company checks allow 3 weeks bank clearance. VISA/MasterCard accepted. Provide phone number with order. SHIPPING: Software add $3.00; Hardware add $3.00 or 3% whichever is greater. Arizona residents add 6% sales tax. Returns must have authorization number (call 602-968-9128) for authorization number). All returned merchandise subject to restocking fee and **must** come with **all** original packaging. No returns allowed after 30 days from shipping date. Prices are for cash; VISA and MasterCard add 3%. Prices subject to change without notice. All products subject to availability from manufacturers and/or suppliers.

(C) = Tape (D) = Disk (CT) = Cartridge

$uch A Deal

Others Claim The Lowest Prices We'll Prove It. By Beating Anyone's Price!*

ATARI SOFTWARE

EPYX

	List Price	But For You!
Temple of Apshai (D&C)	$40	$25
Gateway to Apshai (CT)	$40	$27
Upper Reaches Apshai (D&C)	$20	$13
Silicon Warrior (CT)	$40	$27
Pitstop (CT)	$40	$27
Dragonriders of Pern (D&C)	$40	$25
Jumpman Junior (CT)	$40	$27
Fun With Art (CT)	$40	$27

SEGA

	List Price	But For You!
Congo Bongo (CT) / Buck Rogers (CT) / Star Trek (CT)	$40	$25

SPINNAKER

	List Price	But For You!
Facemaker (CT)	$35	$23
Delta Drawing (CT)	$40	$26
Alphabet Zoo (CT)	$35	$23
Story Machine (CT)	$40	$26
Kindercomp (CT)	$30	$19

SYNAPSE

	List Price	But For You!
Dimension X (D&C)	$35	$22
New York City (D&C)	$35	$22
Rainbow Walker (D&C)	$35	$22
Quasimodo (D&C)	$35	$22
Zeppelin (D&C)	$35	$22
Blue Max (D&C)	$35	$22
Drelbs (D&C)	$35	$22
Syn-Calc (D)	$100	$75
Syn-File (D)	$100	$75
Syn-Trend (D)	$100	$75
Syn-Comm (D)	$35	$22
Syn-Graph (D)	$60	$39
Syn-Mail (D)	$50	$33
Syn-Stat (D)	$70	$45
Syn-Stock (D)	$70	$45

DATASOFT

	List Price	But For You!
Pooyan (D&C)	$30	$19
Letter Wizard (CT)	$70	$45
Money Wizard (D)	$70	$45
Zaxxon (D&C)	$40	$25

GENTRY

	List Price	But For You!
Maniac Miner (D)	$20	$12
Sea Bandit (D&C)	$17	$11
Spider Quaker (D&C)	$17	$11
Rosens Brigade (D)	$17	$11

BRODERBUND

	List Price	But For You!
Choplifter (CT)	$45	$28
Arcade Machine (D)	$60	$39
Matchboxes (D&C)	$30	$19
Operation Whirlwind (D)	$40	$26

(C) = Tape (D) = Disk (CT) = Cartridge

INFOCOM

	List Price	But For YOU!
Zork I, II, II each D	$40	$26
Deadline (D)	$50	$33
Witness (D)	$50	$33
Starcross (D)	$40	$26
Suspended (D)	$50	$33
Planet Fall (D)	$50	$33
Enchanter (D)	$50	$33
Infidel (D)	$50	$33

WIZWARE

	List Price	But For YOU!
Square Pairs (C)	$30	$19
Turtle Tracks (D&C)	$30	$19
Microzine (D)	$40	$26

We Take the Byte Out of Apple Software Prices!

(all disks)	But For YOU!
Datasoft Zaxxon	$26
Epyx Temple of Apshai	$26
Epyx Jumpman	$26
MicroFun Miner 2049	$26
Muse Know Apple IIE	$19
Infocom Zork I, II, III ea.	$26
Infocom Witness	$33
Infocom Planet Fall	$33
Infocom Enchanter	$33
Infocom Infidel	$33
Sir Tech Wizardry I	$33
S.T. Wizardry II Knight I	$23
S.T. Wizardry III Legacy	$26
Spinnaker Delta Drawing	$39
Spinnaker Facemaker	$26
Spinnaker Snooper Tps I	$29
Spinnaker Snooper Tps II	$29

More Byte For Your Buck Specials!

Floppiclene Disk Drive Cleaner	Just $22
Innovative Concepts Flip 15	$ 7
Flip 25 (locking)	$18
Flip 50 (locking)	$25
Flip Atari Cart.	$19
Maxell Disks MDI (10)	$23
MD2 (10)	$39
MR Floppy Disk	ea. $2.25

$uch A Deal

CALL TOLL FREE
1-800-431-8697
Orders Only!
For Information,
Release Dates, Availability
Call
602-968-9128

Rock Bottom Prices on Peripherals!

MONITORS

BMC 12" Green	Just $79
BMC 12" HiRes Green	$125
BMC 12" Amber	$89
BMC 12" HiRes Amber	$129
BMC 13" Composite Color	$249

DATA 20

Vic 40-80 Display Manager	$79
C64 Video Pak 80 includes Word Manager Free	$139
Parallel Printer Interface	$45

CARDCO

Vic 3 Slot Motherboard	$26
Vic 6 Slot Motherboard	$65
Vic 5 Slot C64 Motherboard	$45
Printer Utility Software	$17
Numeric Keypad C64	$29
C64 5 Slot	$45
Graphic Printer Interface	$69
Economy Printer Interface	$39

WICO JOYSTICKS

The Boss	$14
Bat Stick	$19
Red Ball	$21
Track Ball	$30

KOALAPAD

Vic, C64, Atari	$79
Apple, IBM	$99

HESWARE

Hescard 5 Slot VIC	$39
HesModem VIC & 64	$49

PRINTERS

Alphacom 40 Column	$119
Alphacom 80 Column includes Vic, C64, Atari cable	$179
Cardco DM1 Two Color Impact	$119

Okidata, Epson, Citoh — call 968-9128

ATARI DISK DRIVES

Rana	329.00
Trak Single Density w/Printer Port	429.00
Trak Double Density	379.00
Trak Double Density w/Printer Port	459.00

MODEMS

Novation, Hayes
Anchor Call 602-968-9128 .... $AVE

*TERMS OF OFFER: If you find a price for any software or peripheral in this issue that is lower than our advertised price, we'll guarantee to beat it! Valid only on products appearing in our ad. Valid only on product in similar in-stock conditions. Valid only on prices appearing in print.

```
6800 DATA 125,173,148,26,133,126,96
6808 DATA 149,26,201,46,208,9,72
6816 DATA 173,122,0,201,0,240,9
6824 DATA 104,201,58,144,1,96,76
6832 DATA 128,0,169,2,141,251,0
6840 DATA 32,115,0,240,14,176,21
6848 DATA 32,107,201,32,209,26,32
6856 DATA 215,202,76,42,197,169,0
6864 DATA 133,20,133,21,24,144,238
6872 DATA 76,8,207,234,234,234,32
6880 DATA 19,198,160,2,177,95,133
6888 DATA 20,200,177,95,133,21,160
6896 DATA 0,177,95,201,0,208,47
6904 DATA 200,177,95,201,0,208,40
6912 DATA 240,69,169,0,197,20,208
6920 DATA 6,197,21,240,59,198,21
6928 DATA 198,20,32,19,198,160,2
6936 DATA 177,95,197,20,208,231,200
6944 DATA 177,95,197,21,208,224,32
6952 DATA 95,229,24,144,201,32,93
6960 DATA 27,32,228,255,201,0,240
6968 DATA 249,201,145,240,204,201,80
6976 DATA 234,234,234,201,13,240,8
6984 DATA 230,20,208,160,230,21,208
6992 DATA 156,96,56,233,127,170,132
7000 DATA 73,160,255,202,240,8,200
7008 DATA 185,158,192,16,250,48,245
7016 DATA 200,185,158,192,48,6,32
7024 DATA 210,255,208,245,96,164,73
7032 DATA 41,127,32,210,255,96,160
7040 DATA 2,32,215,202,230,199,177
7048 DATA 95,170,200,177,95,32,205
7056 DATA 221,198,199,32,215,202,166
7064 DATA 251,32,228,27,169,0,133
7072 DATA 253,160,3,200,177,95,201
7080 DATA 0,240,83,166,253,208,4
7088 DATA 201,128,176,27,32,210,255
7096 DATA 201,34,208,8,72,165,253
7104 DATA 73,1,133,253,104,201,58
7112 DATA 240,38,208,220,234,234,234
7120 DATA 234,234,234,201,130,208,6
7128 DATA 206,251,0,206,251,0,72
7136 DATA 32,54,27,104,201,129,240
7144 DATA 36,201,167,208,191,230,252
7152 DATA 230,252,24,144,184,32,215
7160 DATA 202,169,0,133,253,165,251
7168 DATA 101,252,170,32,228,27,24
7176 DATA 144,166,169,0,133,252,133
7184 DATA 253,96,230,251,230,251,208
7192 DATA 153,224,0,240,7,32,63
7200 DATA 203,202,24,144,245,96,217,256
```

Program 2: List And Scroll (64 Version)

```
10 I=6769
20 READ A:IF A=256 THEN 40
30 POKE I,A:CK=CK+A:I=I+1:GOTO 20
40 IF CK<>51322THENPRINT"{CLR}ERROR IN DA
   TA STATEMENTS":END
6769 DATA 113,26,173,113,26,133,55
6777 DATA 133,51,173,114,26,133,56
6785 DATA 133,52,234,234,234,169,76
6793 DATA 133,124,173,147,26,133,125
6801 DATA 173,148,26,133,126,96,149
6809 DATA 26,201,46,208,9,72,173
6817 DATA 122,0,201,0,240,9,104
6825 DATA 201,58,144,1,96,76,128
6833 DATA 0,169,2,141,251,0,32
6841 DATA 115,0,240,14,176,21,32
6849 DATA 107,169,32,209,26,32,215
6857 DATA 170,76,42,165,169,0,133
6865 DATA 20,133,21,24,144,238,76
6873 DATA 8,175,234,234,234,32,19
6881 DATA 166,160,2,177,95,133,20
6889 DATA 200,177,95,133,21,160,0
6897 DATA 177,95,201,0,208,47,200
6905 DATA 177,95,201,0,208,40,240
6913 DATA 69,169,0,197,20,208,6
6921 DATA 197,21,240,59,198,21,198
6929 DATA 20,32,19,166,160,2,177
6937 DATA 95,197,20,208,231,200,177
6945 DATA 95,197,21,208,224,32,68
6953 DATA 229,24,144,201,32,93,27
6961 DATA 32,228,255,201,0,240,249
6969 DATA 201,145,240,204,201,80,234
6977 DATA 234,234,201,13,240,8,230
6985 DATA 20,208,160,230,21,208,156
6993 DATA 96,56,233,127,170,132,73
7001 DATA 160,255,202,240,8,200,185
7009 DATA 158,160,16,250,48,245,200
7017 DATA 185,158,160,48,6,32,210
7025 DATA 255,208,245,96,164,73,41
7033 DATA 127,32,210,255,96,160,2
7041 DATA 32,215,170,230,199,177,95
7049 DATA 170,200,177,95,32,205,189
7057 DATA 198,199,32,215,170,166,251
7065 DATA 32,228,27,169,0,133,253
7073 DATA 160,3,200,177,95,201,0
7081 DATA 240,83,166,253,208,4,201
7089 DATA 128,176,27,32,210,255,201
7097 DATA 34,208,8,72,165,253,73
7105 DATA 1,133,253,104,201,58,240
7113 DATA 38,208,220,234,234,234,234
7121 DATA 234,234,201,130,208,6,206
7129 DATA 251,0,206,251,0,72,32
7137 DATA 54,27,104,201,129,240,36
7145 DATA 201,167,208,191,230,252,230
7153 DATA 252,24,144,184,32,215,170
7161 DATA 169,0,133,253,165,251,101
7169 DATA 252,170,32,228,27,24,144
7177 DATA 166,169,0,133,252,133,253
7185 DATA 96,230,251,230,251,208,153
7193 DATA 224,0,240,7,32,63,171
7201 DATA 202,24,144,245,96,256
```

Program 3: Relocater (VIC or 64)

```
10 REM MOVE 'EZLIST/SCROLL TO MEMORY TOP.
20 :
30 LB=6769:REM PROGRAM ADDRESS IN LO MEMO
   RY
40 :
50 HB=PEEK(56)*256+PEEK(55)-399:REM PROGRA
   M ADDRESS IN HI MEMORY
60 :
70 REM MOVE BIT BY BIT
80 READA: REM LOC TO CORRECT
100 FORI=0TO382
103 POKEHB+I,PEEK(LB+I)
105 IFA<>LB+IGOTO170
110 V=PEEK(A)+PEEK(A+1)*256:A=V+HB-LB
120 POKEHB+I,A-INT(A/256)*256:I=I+1:POKEH
    B+I,INT(A/256):READA
170 NEXT
180 PRINT"{CLR}TO ENABLE EZ-LISTER
    {3 SPACES}TYPE SYS"HB+2
190 END
195 REM OFFSET VALUES
200 DATA6769,6772,6779,6793,6798,6803
210 DATA6842,6935,7029,7091,7122,0
```

©

SOFTWARE YOU CAN COUNT ON...

PractiCalc 20™‡ and **PractiCalc Plus™‡**: Complete electronic spreadsheets that turn the Commodore VIC-20 into a business computer. ($39.95* TD and $49.95* TD, respectively.)

PractiCalc 64™: The computer spreadsheet for the Commodore 64 with over 20 mathematical functions and the ability to graph, sort, and search for entries. ($49.95* TD)

PS: The Programmable Spreadsheet™: Finally, a computer spreadsheet which can handle the most complicated operations within the structure of a spreadsheet – since *you* can program it with BASIC. Available for the Commodore 64 ($79.95 D) & Apple IIe ($79.95 D).

Rabbit Base™‡: A data-file manager for the Commodore VIC-20 with simple screen instructions for efficient use. ($29.95 T)

Inventory 64™: A smart inventory-tracking system for the Commodore 64 that handles 650 parts. ($39.95 D)

C-64 Analyst: A diagnostic program which tests the Commodore 64 and its peripherals to detect hardware defects. An invaluable tool for C-64 users! ($19.95 D)

† 8K RAM required – ‡16K RAM required
* Price given for tape version. Disk version slightly higher
T Available on tape – D Available on disk
Prices shown are manufacturer's retail prices.

Total Health™: For fitness and health enthusiasts, a program which monitors and encourages proper nutrition. (For the Commodore VIC-20; $24.95 T and C-64 $29.95 D)

AND LEARN FROM...

Math Duel™: A math program for ages 5-12 that combines classroom learning with gameroom fun! Available for the Commodore VIC-20. ($19.95 T)

Sprintyper™: A typing tutorial for the Commodore VIC-20 that encourages speed and accuracy in both the novice and experienced typist. ($19.95 T)

Tiny Tutor™: A pre-schooler program with fun graphics and sound to teach simple math. ($19.95 T)

Composer™: A simple music composition program for the Commodore VIC-20 that teaches musical notation and allows 'melodies' to be saved to tape for later recall. ($19.95 T)

COMPUTER SOFTWARE ASSOCIATES

50 Teed Drive, Randolph, Massachusetts 02368

AND PLAY WITH...

Zeppelin Rescue™: An intelligent rescue game for the Commodore 64 with arcade-like graphics and the greatest challenge for those with persistence & skill. ($24.95 D)

Skramble™: You're lost in enemy territory. But before leaving for home, you can play havoc with their airfields and oil supply. Are you ready for the challenge? (For the Commodore VIC-20; $19.95 T)

Barrel Jumper™†: For the Commodore VIC-20, this game confronts you with a pyramid of steel girders. The present King of the Hill is an angry ape who's hurling barrels at you. Step lively! ($19.95 T)

See your local dealer for CSA programs or order directly by calling toll-free:

1-800-343-1078

For more information about these and many other programs for your home computer, write to CSA.

Programmers with programs to market are encouraged to send copies for review to CSA.

Dealer & distributor inquiries are welcomed by:

Micro Software International Inc
The Silk Mill 44 Oak Street
Newton Upper Falls, Massachusetts 02164

Commodore Files For Beginners
Part 2

Jim Butterfield, Associate Editor

Expanding on his program examples from last month, Associate Editor Jim Butterfield suggests ways to improve and safeguard your files. For disk and tape users.

Creating A File By Program

We can repeat the file creation that we performed last month with direct statements, but this time we'll do it in a more typical way: as part of a program. Here come the statements we have seen before, with a few small enhancements:

```
100 PRINT "FILE CREATION"
110 INPUT "NAME OF FILE";N$
```

When the program runs, we must type in a file name. This might be the same name we used previously (STUDENTS). It's wise to choose a name that hasn't been used before. In fact, with disk it's mandatory: we cannot have two files with exactly the same name on one disk.

Now for the OPEN statement. For disk, we type:

```
120 OPEN 1,8,2,"0:"+N$+",S,W"
```

For tape, we make line 120 read:

```
120 OPEN 1,1,2,N$
```

Now to write the data. Since we're writing a generalized program, it might be wise to ask the user to input the data. As soon as it is received, we'll write it to the file:

```
130 INPUT "NAME";A$
140 INPUT "STUDENT NUMBER";B$
150 INPUT "MARK";M
160 REM PRINT IT
170 PRINT#1,A$;CHR$(13);
180 PRINT#1,B$;CHR$(13);
190 PRINT#1,M;CHR$(13);
```

We could make the program more friendly by asking ARE YOU SURE? in line 155, so that the user could reenter the information if a mistake had occurred.

Now that the record is written, we need to ask if there are any more:

```
200 PRINT
210 INPUT "MORE";X$
220 IF X$="Y" OR X$="YES" GOTO 130
```

When we get beyond this point, the user has signaled that the job is completed. All we need to do is CLOSE the file, and we're finished:

```
230 CLOSE 1
240 PRINT "FILE ";N$;" IS WRITTEN"
```

Trimmings For Disk

If we are using disk, we might add disk error checking. This tells us if we have problems — it's especially important at the time of opening the file. The extra lines for this would be added to the above program:

```
90 OPEN 15,8,15
95 PRINT#15,"I0"
125 INPUT#15,E,E$,E1,E2
126 IF E THEN PRINT E$:STOP
```

Lines 125 and 126 may be repeated after each disk activity, so we could see the same instructions at lines 205 and 206, and again at 235 and 236. You could put these two lines in a subroutine, but they are brief enough to repeat at the appropriate places. Finally, we should CLOSE the command channel with:

```
250 CLOSE 15
```

Always OPEN the command channel at the beginning of a program and CLOSE it at the end. Closing a command channel causes the disk to close any other channels it might have going; it

Merry Christmas!

JUST AS THE DRUIDS COMPILED THE FIRST COMPUTER (Stonehenge) so they also brought the first decorated tree into the home. The winter celebration of these mystics has filled us with the desire to continue their special ways. LISTEN . . .

CASSETTALOG!

DID YOU HEAR IT? Imagine a cassette containing • Descriptions of all our games • Programming hints • Other products • **FREE GAME** • Much more! One side is programmed for the Commodore 64™. The other for the VIC 20™. Send just $2.00 to cover costs and then **DEDUCT $3.00 FROM YOUR FIRST ORDER!** Actually, the very fun of our cassettalog would make a nice Christmas present itself!

GAMES!

Standard VIC 20™ Cassette

Alien Panic	$9.95
Krazy Kong	$9.95
Racefun	$12.95
The Catch	$12.95
Antimatter Splatter	$16.95
Exterminator	$19.95
Music Writer III	$16.95
Kings Ransom	$16.95

Expanded Memory VIC 20™ Cassette

3-D Man (+3K)	$12.95
Space Quest (+8K)	$16.95
Defender on Tri (+3K)	$12.95

Commodore 64™ Diskette

3D-64 Man	$19.95
Exterminator 64	$19.95
Widows Revenge	$19.95
Quiz Pro	$19.95

BONUS!

The first 250 orders over $20 placed from this advertisement will receive a Nufekop dust cover for the 64 or VIC 20 FREE OF CHARGE!

nüfekop

P.O. Box 156, 21255 Hwy. 62, Shady Cove, Oregon 97539 1-800-525-2529

COMMODORE 64

80 COLUMN BOARD

$149.00*

The dream of seeing 80 columns on the screen at one time is now a reality. The Protecto Expansion Board converts your Commodore 64 to 80 columns! PLUS you get a word processor with database mailmerge, an electronic spreadsheet, and a terminal emulator. List $249. SALE $149. Coupon Price $139.00 (Disk Programs add $10.00).

80 COLUMN MONITOR SALE

	List	Sale
9″ Screen - Green Text Display	$139	$ 79
12″ Scrren - Green Text Display (anti reflective screen)	$199	$ 99
12″ Screen - Amber Text Display (anti reflective screen)	$219	$119
14″ Screen - Color Text Display	$299	$249

VIC 20 COMPUTER 40-80 COLUMN BOARD LIST $149.00 SALE $89.00

- ***15 DAY FREE TRIAL***
- ***WE HAVE LOWEST COMMODORE 64 PRICES***
- ***WE HAVE OVER 500 PROGRAMS***
- ***VISA • MASTERCHARGE • COD***
- ***WE LOVE OUR CUSTOMERS***

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

could give you real trouble if performed too early.

Trimmings For Tape

You *could* remove the ;CHR$(13); ending from the PRINT#1 lines if you wish. But it might be best to leave it in place, so that your programs can be converted to disk operation without fuss.

If you have an original small-keyboard PET, you can't write to disk at all and may have trouble with cassette tape (blocks written too closely together). If you're serious about files, you might want to upgrade your machine.

A cassette tape file doesn't need to have a name, but use one anyway.

Reading It Back

It would be nice to bring the file back using direct statements, as we did the first time we wrote the information. However, we can't use INPUT# in direct mode, so we must write a program. Much of it will look familiar. First, we OPEN the file, then ask for the name:

```
100 PRINT "FILE READER"
110 INPUT "FILE NAME";N$
```

For disk, we would write the OPEN statement as:

```
120 OPEN 1,8,2,N$
```

We don't need to specify the drive number as both will be checked. We don't need to specify ,S,R for sequential read because these options will be assumed. It doesn't hurt to specify everything, however.

For tape, we would OPEN with:

```
120 OPEN 1,1,0,N$
```

In fact, if there's only one data file on the tape, or if the one we want is the first, we could write OPEN 1 and everything else would be assumed.

```
130 INPUT#1,A$
140 INPUT#1,B$
150 INPUT#1,M
```

Now that we've input a record, let's print it out:

```
160 PRINT "NAME:{3 SPACES}";A$
170 PRINT "NUMBER: ";B$
180 PRINT "MARK:{3 SPACES}";M
```

Are there any more records? The computer knows; and if we know how, we can ask the computer.

There's a variable in the computer called ST or STATUS. After every file operation — or more exactly, after every input/output operation — variable ST will be set as follows:

ST equals 0: file OK, more to come
ST equals 64: file OK, no more to come
ST other than 0 or 64: file has a problem

For our simple reading program, we can type:

```
190 IF ST=0 GOTO 130
```

Thus, if the file is OK and is not at the end, we'll go back and get another record.

Finally, we CLOSE the file with:

```
200 CLOSE 1
```

RUN the above program, and the information we wrote to file STUDENTS will be recalled and printed out to the screen.

Try Your Hand At These

Our file program is a good working example. You might like to see if you can write some of the following variations:

If you have disk, add error checking. Then try creating errors (bad names) and see what happens.

Modify the program to print only student records for students named JONES.

Modify the program to count the number of students.

Modify the program to calculate an average grade.

We'll look at other aspects of sequential files next time around. ©

80 COLUMN PRINTER SALE—$149.00*

COM-STAR T/F

Tractor Friction Printer

only $219

(Parallel or Serial)

- Lowest price quality tractor friction printer in the U.S.A. • Fast 80 characters per second
- 40, 46, 66, 80, 96, or 132 characters per line spacing • Prints labels, letters, graphs, and tables
- List your programs • Print out data from modem services

*STX-80 COLUMN PRINTER—$149.00

Prints full 80 columns. Super silent operation, 60 CPS, prints Hi-resolution graphics and block graphics, expanded character set, exceptionally clear characters, fantastic print quality, uses inexpensive thermal roll paper!

DELUXE COMSTAR T/F PRINTER—$219.00

The Comstar T/F is an excellent addition to any micro-computer system. (Interfaces are available for Apple, VIC-20, Commodore-64, Pet, Atari 400 and 800, and Hewlett Packard). At only $219 the Comstar gives you print quality and features found only on printers costing twice as much. Compare these features.

- **BI-DIRECTIONAL PRINTING** with a LOGIC SEEKING CARRIAGE CONTROL for higher through-put in actual text printing. 80 characters per second.
- **PRINTING VERSATILITY:** standard 96 ASCII character set plus block graphics and international scripts. An EPROM character generator includes up to 224 characters.
- **INTERFACE FLEXIBILITY:** Centronics is standard. Options include EIA RS232C, 20mA Current Loop.
- **LONG LIFE PRINT HEAD:** 100 million character life expectancy.
- **THREE SELECTABLE LINE SPACINGS:** 6, 8 or 12 lines per inch.
- **THREE SELECTABLE CHARACTER PITCHES:** • 10, 12 or 16.5 characters per inch. 132 columns maximum. Double-width font also is standard for each character pitch.
- **PROGRAMMABLE LINE FEED:** programmable length from 1/144 to 255/144 inches.
- **VERTICAL FORMAT CONTROL:** programmable form length up to 127 lines, useful for short or over-sized preprinted forms.
- **FRICTION AND TRACTOR FEED:** will accept single sheet paper.
- **224 TOTAL CHARACTERS**
- **USES STANDARD SIZE PAPER**

If you want more try —

Premium Quality COMSTAR T/F SUPER-10X PRINTER—$299.00
More Features Than RX-80

For $299 you get all of the features of the Comstar T/F plus 10" carriage 120 cps, 9 x 9 dot matrix with double strike capability for 18 x 18 dot matrix. High resolution bit image (120 x 144 dot matrix), underlining, backspacing, left and right margin settings, true lower descenders, with super and subscripts, and prints standard, Italic, Block Graphics, special characters, plus 2K of user definable characters. For the ultimate in price performance the Comstar T/F Super 10" leads the pack!

Double Immediate Replacement Warranty

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

15 DAY FREE TRIAL

OTHER OPTIONS

Extra Ribbons	$ 5.95
Roll Paper Holder	32.95
Roll Paper	4.95
5000 Labels	19.95
1100 Sheets Fan Fold Paper	13.95

Add $17.50 shipping, handling and insurance. Illinois residents please add 6% tax. Add $40.00 for CANADA, PUERTO RICO, HAWAII, ALASKA orders. WE DO NOT EXPORT TO OTHER COUNTRIES. Enclose cashiers check, money order or personal check. Allow 14 days for delivery, 2 to 7 days for phone orders, 1 day express mail available!! Canada orders must be in U.S. dollars.

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

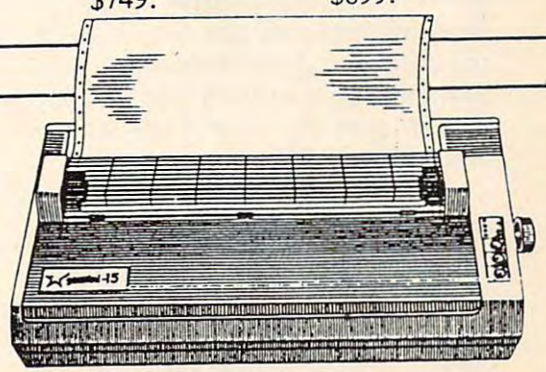
SUPER-10" ABCDEFGHIJKLMNOPQRSTUVWXYZ
ABCDEFGHIJKLMNOPQRSTUVWXYZ 1234567890

SUPER
COM-STAR T/F 15″ PRINTER
SALE $379.00

NOW YOU CAN BUY A TRACTOR-FRICTION 15″ CARRIAGE PRINTER FOR AN INCREDIBLE $379.00

	star COMSTAR 15″	EPSON* MX-100/Type III	OKIDATA* MICROLINE 83A
Warranty This refers to standard warranties. Coverage varies by manufacturer. Complete information is available at your place of purchase	180 day	90 day	90 day
Buffer	2.3 K	1 Line	1 Line
CPS	100	80	120
CPI	10.12.17	10.17	10.17
Over 80 Type Fonts	Yes	No	No
Block Graphics	Yes	No	No
Special Character Sets Available	Yes	No	No
Roll Paper Holder Standard	Yes	No	No
Suggested Retail Price Prices shown are based on data from each manufacturer. Actual retail prices may vary	$599.	$749.	$899.
Special Price	**$379.**		

*Epson and Okidata are registered trademarks of Epson America, Inc. and Okidata Corp., respectively.



Now you can save big on the Com-Star 15″ dot matrix printer from Protecto.

It's the 15½″ carriage printer that combines the most advanced features with a price that's better than ever. And component by component, Com-Star 15″ is so reliable, Protecto can warranty it for up to twice as long as its major competitors.

The popular Com-Star 15″. A great printer at a truly great price. When you add it up, it figures to be a super sale.

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

Add $17.50 for shipping, handling and insurance. Illinois residents please add 6% tax. Add $35.00 for CANADA, PUERTO RICO, HAWAII orders. WE DO NOT EXPORT TO OTHER COUNTRIES.

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

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

Jim Butterfield, Associate Editor

Hopping Around

Transfer of control — jumping and branching — seems to be easy and straightforward to accomplish. In 6502 programming, you can make a decision-based branch, which will take you forward or backward a hundred-odd locations; or an unconditional jump, which will take you anywhere you want to go.

Yet there are a number of techniques that transfer control in unusual ways. Often they may seem like tricks, but they can be useful in achieving programming objectives: speed, flexibility, or compactness. We'll look at some of these techniques here.

The Long Branch

If you want to use a branch to implement a decision, your range is limited to slightly over 120 locations forward or backward. We often want to get around this limitation. It may be argued, by the way, that well-organized programs should never need to branch over any great distance; that your programs should be organized into subroutine modules so that transfers of control will always be short and visible.

For the moment, let's look at an example:

```
2000            LDX  #$20
2002  BIGLOOP   LDA  #$0D
....
....
20C0            DEX
20C1            BNE  BIGLOOP
20C3            ...
```

We have a problem here. We can't branch over the needed range — about 190 bytes. The simple way is to insert a JMP:

```
20C0            DEX
20C1            BEQ  SKIP
20C3            JMP  BIGLOOP
20C6  SKIP      ...
```

Another way is more subtle and must be used with care. It avoids the JMP, and thus makes a routine more easily relocatable. Let's assume that somewhere in our program sequence we have a BNE:

```
2000            LDX  #$20
2002  BIGLOOP   LDA  #$0D
....
....
2065            LDA  $027A
2068            BNE  STEP
```

Now, immediately after the BNE at address 2068, another BNE instruction would never branch. After all, if the Z flag is clear, we will take the previous branch to STEP. And if the Z flag is set, neither branch will be taken. So we might use:

```
2000            LDX  #$20
2002  BIGLOOP   LDA  #$0D
....
....
2065            LDA  $027A
2068            BNE  STEP
206A  LINK      BNE  BIGLOOP
....
....
20C2            DEX
20C3            BNE  LINK
```

As the program executes in the area of 2065, it will never take the branch to BIGLOOP. But when we get down to the bottom, the instruction at 20C3 will (if conditions are right) branch to LINK, and will immediately branch again to BIGLOOP. Each branch is now a shorter hop and easily within range.

Hidden Instructions

Suppose you need a series of PRINT subroutines, one to print a RETURN ($0D), one to print a space ($20), and another to print an exclamation point. You could write three subroutines; or you could write the three Load commands and then branch to a common point; or you could do this:

```
2000  A9 0D      LDA  #$0D    ;return
2002  2C A9 20   BIT  $20A9   ;hidden space
2005  2C A9 3F   BIT  $3FA9   ;hidden question mark
2008  20 D2 FF   JSR  PRINT   ;print it
200B  60         RTS          ;return
```

What happens when we call address 2000? We load the RETURN character, perform two

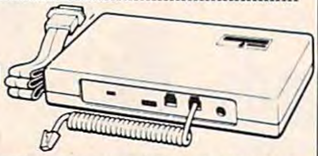
Hewlett Packard

Write or call for prices.

SPECIALS on INTREGATED CIRCUITS

6502	7.45	10/ 6.95	50/ 6.55	100/ 6.15
6502A/6512A	8.40	10/ 7.95	50/ 7.35	100/ 6.90
6520 PIA	5.15	10/ 4.90	50/ 4.45	100/ 4.15
6522 VIA	6.45	10/ 6.10	50/ 5.75	100/ 5.45
6532	7.90	10/ 7.40	50/ 7.00	100/ 6.60
2114-L200		2.45	25/ 2.30	100/ 2.15
2716 EPROM		4.45	5/ 4.20	10/ 3.95
2532 EPROM		5.90	5/ 5.75	10/ 5.45
6116 2K×8 CMOS RAM		6.45	5/ 6.20	10/ 5.95
4116 RAM				8 for 14
4164 RAM				6.50
Zero Insertion Force 24 pin Socket (Scanbe)				2.00

Anchor Automation Signalman Modems



FREE SOURCE MEMBERSHIP WITH SIGNALMAN

All Signalman Modems are Direct Connect, and include cables to connect to your computer and to the telephone. Signalman Modems provide the best price-performance values.

Dealer and OEM inquiries invited

Volksmodem	(79)	59
RS232 or Atari Cable for Volksmodem		9
Mark I RS232	(99)	59
Mark VI for IBM Personal Computer	(279)	179
Mark VII Auto Dial/Auto Answer	(159)	89
Mark XII Smart Model 1200/300	(399)	295

DC HAYES Smartmodem	219
DC Hayes Smartmodem 1200/300	529

PROM QUEEN for VIC	170
Apple Emulator for Commodore 64	Call
STAT Statistics Package for C64	95
Solid Oak 2 Level Stand for C64 or VIC	29
C64/VIC Switch (networking)	125
BACKUP V1.0 tape copier for C64 or VIC	20
CARDBOARD/6 Motherboard - VIC	64
CARDBOARD/5 Motherboard - C64	56
CARD PRINT G Printer Int. with Graphics	72
CARD PRINT B Printer Interface—C64/VIC	40
CARDBOARD/3s Motherboard - VIC	32
CARDCO C64/VIC Calculator Keypad	32
CARDRAM/16 RAM Expansion - VIC	50
Complete CARDCO Line in stock	
CIE and VIE IEEE Interfaces in stock	
MAE Assembler for C64 + CBM	89

APPLE—FRANKLIN ITEMS

KRAFT Apple Joystick	40
Kraft Apple Paddle Pair	30
SPINNAKER Software in stock	
Broderbund Saftware in stock	
16K RAM Card for Apple	59
Multiplan—Microsoft	185
Solid Oak 2 Level Stand for Apple	29
Serial Card for Apple	99
MCP RAM/80 column card for IIe (AP/TXT)	139
Z80 Softcard and CP/M (Microsoft)	235
RANA Elite I with Controller	389
Parallel Printer Interface/Cable	79
Microtek Interfaces in stock	
Apple Dumpling with 16K Buffer	160
Grappler + Interface	129
Kraft Products for Apple in stock	
DC Hayes Micromodem II	299
PFS: File	95
PFS: Report	95
Videx 80 Column Card	209
Hayden Software for Apple 20% OFF	
Apple Blue Book	19

commodore

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

PETSCAN I $245 base price

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

COMPACK/STCP $115

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

VIDEO ENHANCER for Commodore 64 69
Realize video quality equal or better than composite monitor using standard color TV.

SCREENMAKER 80 Column Adapter for C64 145
Provides big screen capability for business applications.

GENESIS Computer Corp

VIController (for C64 as well)	50
combine with BSR modules for home or business control	
COMSENSE Remote Sensing Adapter for C64 or VIC	35
COM VOICE Synthesizer for C64 or VIC	139
includes software for test to speech, pitch, etc.	
COM CLOCK Real Time Clock with battery backup	45

VIC 20 Products and Software in stock

Thorn EMI Software		**UMI Software**	
ABACUS Software		**HES Software**	
16K RAM for VIC	64	Vanilla Pilot	27

VICTORY Software for VIC and C64

Street Sweepers (VIC)	12	Kongo Kong (VIC)	16
Night Rider (VIC)	11	Cosmic Debris (VIC)	12
Annihilator	16	Adventure Pack I	16
Adventure Pack II	16	Metamorphosis	11
Educational Pack I	11	Trek	12
Strategy Pack I	16	Grave Robbers	12

PAPER CLIP Word Processor	
ORACLE Data Base from Batteries Included	
Super BusCard from Batteries Included	179
Commodore 64 Programmers Reference Guide	16
MicroChess for C64—8 levels of play	19
excellent graphics and color	
SPINNAKER Software C64, Apple, IBM, Atari	
Compute!'s First Book of PET/CBM	11
C64 or VIC SWITCH	125
POWER ROM Utilities for PET/CBM	78
WordPro 3+/64 with Spellmaster	85
WordPro 4+ - 8032, disk, printer	295
SPELLMASTER spelling cheoker for WordPro	
VISICALC for PET, ATARI, or Apple	189
PET-TRAX PET to Epson Graphics Software	40
SM-KIT enhanced PET/CBM ROM Utilities	40
Programmers Toolkit - PET ROM Utilities	35
EASY CALC for C64	65
PET Spacemaker II ROM Switch	36
COPYWRITER Word Processor for C64	69
2 Meter PET to IEEE or IEEE to IEEE Cable	40
Dust Cover for PET, CBM, 4040, or 8050	8
CmC Interfaces (ADA1800, ADA1450, SADI in stock)	
Programming the PET/CBM (Compute!) — R. West	20
Compute! First Book of VIC	11
OMNICALC (HES)	79
HES MODEM with Software	65
HES Software and Hardware in stock	
UMI products in stock	

GRIDRUNNER (HES) VIC or C64		29
COCO (HES) Tutorial for C64		39
Public Domain Disks for C64		65
Agressor (HES)		29
HES Turtle Graphics	VIC 29	C64 49
HES Writer	VIC 30	C64 34
Grand Master Chess (UMI) C64 or VIC		27
Renaissance (UMI)	VIC 29	C64 24
Synthesound (HES)		44
HES MON	VIC or C64	29
6502 Professional Developement System (HES)		24
Robot Panic 29	Pirate's Peril 29	Retro Ball 29

DISK SPECIALS

Scotch (3M) 5" ss/dd	10/ 2.20	50/ 2.00	100/ 1.95
Scotch (3M) 5" ds/dd	10/ 3.05	50/ 2.80	100/ 2.75
Scotch (3M) 8" ss/sd	10/ 2.30	50/ 2.10	100/ 2.06
Scotch (3M) 8" ss/dd	10/ 2.85	50/ 2.70	100/ 2.65

We stock VERBATIM DISKS

Write for Dealer and OEM prices.

Sentinal 5" ss/dd	10/ 1.90	50/ 1.85	100/ 1.80
Sentinal 5" ds/dd	10/ 2.55	50/ 2.50	100/ 2.45

We stock Dysan disks

Wabash 5" ss/sd	10/ 1.60	50/ 1.55	100/ 1.45
Wabash 5" ss/dd	10/ 1.90	50/ 1.85	100/ 1.75
Wabash 8" ss/sd	10/ 2.00	50/ 1.95	100/ 1.85

We stock MAXELL DISKS

Write for dealer and OEM prices.

Disk Storage Pages	10 for $5	Hub Rings 50 for $6
Disk Library Cases	8"—3.00	5"—2.25
Head Cleaning Kits	11	

CASSETTE TAPES—AGFA PE-611 PREMIUM

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

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

Zenith ZVM-121 Green Phosphor Monitor	95
Zenith new color and monochrome monitors in stock	
MultiPlan—IBM or Apple	185
Quadboard for IBM available	
Peachtext 5000 Software Package	219
PFS Software for IBM and Apple in stock	
VOTRAX Personal Speech System	280
BMC [illegible]91 Color Monitor	199
BM[illegible]2A 12" Green Monitor	85
Dynax (Brother) DX-15 Daisy Wheel Printer	469
Itoh Prowriter Parallel Printer	379
Panasonic 1090 Printer with Correspondence Mode	279
USI CompuMOD 4 R F Modulator	39
Daisywriter 2000 with 48K buffer	1050
Many printers available (Gemini-Star, Brother, OKI, etc.)	
We Stock AMDEK Monitors	
Amdek DXY-100 Plotter	590
A P Products	15% OFF
Watanabe Intelligent Plotter 990	6-pen 1290
BROOKS 6 Outlet Surge Suppressor/Noise Filter	54
We stock Electrohome Monitors	
Synertek SYM-1 Microcomputer	189

ALL BOOK and SOFTWARE PRICES DISCOUNTED

Panasonic 12" Monitor (20 MHz) with audio	129
Panasonic CT-160 Dual Mode Color Monitor	245

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

ZENITH data systems

HERO 1 Robot (factory assembled)	2145
Z29 Terminal (DEC and ADM compatible)	680
ZT-10 Intel. Terminal with Serial Port	340
Z100 16-bit/8-bit Systems in stock	CALL
We stock entire Zenith line.	

ATARI SPECIALS

WE STOCK ENTIRE LINE—write for prices.
SPINNAKER and Broderbund Software in Stock.

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

A B Computers

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

OAK STAND-C64, VIC, Apple, Atari 29
Beautiful natural solid oak two-level stand. Rests on table above computer. Holds disk drives/cassette deck, as well as your monitor/TV.

KMMM Pascal IV.1 for PET/CBM/C64 $95
Full-featured Pascal for Commodore Computers.

Now suitable for all Advanced Placement Courses.
Includes all features of full Jensen-Wirth Pascal except WITH, SETS, RECORD VARIANTS (plus has STRING extension). Includes machine language Pascal Source Editor (with syntax checking), machine language P-Code Compiler, P-Code to machine language Translator or optimized object code. Run-time package, User Manual, and sample programs.
Requires 32K Please specify configuration.

EARL for PET (disk file based) $65
Editor, Assembler, Relocater, Linker

Generates relocatable object code using MOS Technology mnemonics. Disk file input (can edit files larger than memory).

Paper Clip (Batteries Included) $109
Extremely comprehensive word processor for Commodore and Commodore 64 computers. Has features of WordPro, plus advanced functions like horizontal scroll, column move, column arithmetic, column sort, and comprehensive printer support.

Delphi's ORACLE (Batteries Included) $125
Comprehensive Data Base, Report Writer, Mail Label system allowing large record size (over 8000 characters) with the number of records in a file limited only by disk capacity (7.5 MB on 9090 drive). Fast machine language routines, including full multilevel sorts.

Super BusCard (Batteries Included) $179
- full buffered IEEE488 bus for speed.
- cartridge extension slot.
- parallel printer port.
- DOS "wedge" commands included.
- machine language monitor included.
- room for 24K ROM BASIC 4.0 (optional).

RAM/ROM for PET/CBM
4K or 8K bytes of soft ROM optional battery backup.

Use RAM/ROM as a software development tool to store data or machine code beyond the normal BASIC range, or to load a ROM image to avoid ROM socket conflicts. Possible applications include machine language sort (such as SUPER-SORT), universal wedge, Extramon, etc.

RAM/ROM — 4K $75 RAM/ROM — 8K 90
Battery Backup Option 20

PORTMAKER DUAL RS232 SERIAL PORT $63
Two ports with full bipolar RS232 buffering. Baud rates from 300 to 4800. For PET/CBM, AIM, SYM.

SuperGraphics 2.0
NEW Version with TURTLE GRAPHICS

SuperGraphics, by John Fluharty, provides a 4K machine language extension which adds 35 commands to Commodore BASIC to allow fast and easy plotting and manipulation of graphics and shapes on the PET/CBM video display.

SOUND commands allow you to initiate notes or songs from BASIC, and then play them in the background mode without interfering with your BASIC program.

Additionally, seven new TURTLE commands open up a whole new dimension in graphics.

Specify machine model (and size), ROM type.

SuperGraphics in ROM ($A000 or $9000) $45
Volume discounts available for schools.

FLEX-FILE for PET/CBM Computers — NEW VERSION 2 now for C64

FLEX-FILE is a set of flexible, friendly programs to allow you to set up and maintain a data base. Includes versatile Report Writer and Mail Label routines, and documentation for programmers to use Data Base routines as part of other programs.

RANDOM ACCESS DATA BASE
Record size limit is 256 characters. The number of records per disk is limited only by record size and free space on the disk. File maintenance lets you step forward or backward through a file, add, delete, or change a record, go to a numbered record, or find a record by specified field (or partial field). Field lengths may vary to allow maximum information packing. Both subtotals and sorting may be nested up to 5 fields deep. Any field may be specified as a key. Sequential file input and output, as well as file output in WordPro and PaperMate format is supported. Record size, fields per record, and order of fields may be changed easily.

MAILING LABELS
Typical mail records may be packed 3000 per disk on 8050 (1400 in 4040). Labels may be printed any number wide, and may begin in any column position. There is no limit on the number or order of fields on a label, and complete record selection via type code or field condition is supported.

REPORT WRITER
Flexible printing format, including field placement, decimal justification and rounding. Define any column as a series of math or trig functions performed on other columns, and pass results such as running total from row to row. Totals, nested subtotals, and averages supported. Complete record selection, including field within range, pattern match, and logical functions can be specified.

FLEX-FILE 2 by Michael Riley $110
CBM64, PET/CBM/VIC—32K, Disk. Specify configuration.

SCREEN MAKER (cgrs microtech) $149
80 Column Adapter for Commodore 64

Expand your computer for business applications. Provides 80 column × 24 line display in a 2K video RAM. Linking software provided.

Copy-Writer Word Processor $69
Full-featured professional word processor with over 800 lines of text per memory load on C64. Has features not available in many word processors such as double column printing, built in graphic capability, shorthand notations, and ability to support all printer codes.

SPECIAL COMBINATION PACKAGE $200
Includes SCREEN MAKER AND Copy Writer for C64

PROGRAM YOUR OWN EPROMS $75
Branding Iron EPROM Programmer for PET/CBM software for all ROM versions. Includes all hardware and software to program or copy 2716 and 2532 EPROMs.

DISK I.C.U. $40
Intensive Care Unit by L.C. Cargile
COMPLETE DISK RECOVERY SYSTEM FOR CBM DRIVES

Edit disk blocks with ease; duplicate disks, skipping over bad blocks; un-scratch scratched files; check and correct scrambled files; recover improperly closed files.

Includes complete diagnostic facilities, extensive treatment of relative files, optional output to IEEE488 printer, and comprehensive user manual (an excellent tutorial on disk operation and theory).

Furnished on copy-protected disk with manual.
Backup disk available, $10 additional.

CBM Software
SUBSORT for PET/CBM 35
–excellent general purpose machine language sort routine.
COMAL Package for CBM 25
–includes software on disk, and Comal Handbook
BASIC INTERPRETER for CBM 8096 $95
PEDISK II Systems from cgrs Microtech available.
FILEX IBM 3741/2 Data Exchange Software available.
JINSAM Data Base Management System for CBM.
CASH MANAGEMENT SYSTEM for CBM $45

Petspeed BASIC Compiler	120
Integer BASIC Compiler	120
BPI Accounting Modules	280
UCSD Pascal (without board)	135
Wordcraft 80 or 8096	265

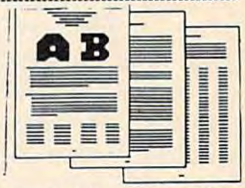
FORTH for PET $50
BY L.C. Cargile and Michael Riley

Features include:
- full FIG FORTH model.
- all FORTH 79 STANDARD extensions.
- structured 6502 Assembler with nested decision macros.
- full screen editing (just as in BASIC)
- auto repeat key.
- sample programs.
- standard size screens (16 lines by 64 characters).
- 150 screens per disk on 4040, 480 screens on 8050.
- ability to read and write BASIC sequential files.
- introductory manual and reference manual.

For Commodore 64, or any 16K/32K PET/CBM with ROM 3 or 4, and CBM disk drive. Please specify configuration when ordering.

Metacompiler for FORTH $30
Simple metacompiler for creating compacted object code which can be run independently (without FORTH system).
Floating Point for FORTH $20

PageMate 60 COMMAND WORD PROCESSOR
by Michael Riley



Paper-Mate is a full-featured word processor for Commodore computers. Page-Mate incorporates 60 commands to give you full screen editing with graphics for all 16K or 32K machines (including 8032), all printers, and disk or tape drives. Many additional features are available (including most capabilities of WordPro 3).

Page-Mate functions with all Commodore machines with at least 16K, with any printer, and either cassette or disk.

To order Page-Mate, please specify machine and ROM type.

Page-Mate (disk or tape) for PET, CBM, VIC, C64 $40

SM-KIT for PET/CBM $40
Enhanced ROM based utilities for BASIC 4. Includes both programming aids and disk handling commands.

STAT for PET/CBM and C64 95
Comprehensive Statistical Analysis Routines

Includes complete disk-based handling routines. Features: normal, T, Chi-Square, F, binomial, Poisson, and exponential distributions; one way ANOVA; two way ANOVA; contingency analysis; linear regression; data transformation; histogram, curve, and scatter plotting; and random sample data generation. Specify machine type and drive when ordering.

MicroChess 64 $19
Machine language version on cassette for C64. Plays at 8 different levels, with excellent board display and graphics.

Commodore 64
Hunter-Killer - Commodore 64 15
authentic naval warfare game (complete with sonar)
SPINNAKER Software in stock
Broderund Software in stock

BASM Compiler and Assembler	85
WordPro 3+/64	75
Vanilla PILOT with Turtle Graphics - also includes sound, Toolkit, joystick support	27
Commodore 64 Programmer Reference Guide	16
EARLY GAMES for Young Children	25
PETSPEED Compiler C64	120
CALC RESULT ADVANCED Spread Sheet Package	135
Adventure (disk)	9
PILOT 64	45
LOGO 64	45
Easy Calc 64	65
MAE Assembler - C64	85
Synthy-64 music and sound synthesizer	26
Tiny BASIC Compiler	17
Assembly Language Tutorial - C64/VIC	27
ScreenGraphics-64 adds BASIC Graphics	22

Abacus Software in stock

All Victory Software in stock. VICTORY SOFTWARE®

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

A B Computers

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

meaningless BIT tests — they set the status flags, but we never test them — and then print RETURN.

But, what happens if we JSR to 2003? That's not an instruction — wait — yes, it is. It's A9 20, which is the same as LDA #$20. So we load the A register with a space character, do one meaningless BIT instruction, and print it. And if we JSR to 2006, we'll load A with $3F, the question mark, and print that.

What's happening here? By inserting the byte 2C ahead of the two extra A9 or LDA commands, we have made them "invisible." We can slide right through them, without needing to jump over them.

The BIT test, $2C, is ideal since it does not affect memory or any registers other than the status register, which we don't need. Some computers have a series of NOP commands of various instruction lengths, which are useful for "hiding" instructions within the address field. Sometimes these instructions have names other than NOP – for example, "Branch Never" or "Rotate 0 Bits" — but you get the idea.

The Invisible Return

Our last example ended with a JSR and RTS. Think about this. We will call a subroutine; it will return to us; and then we will return to the routine that called us. The return addresses are kept on the stack, of course. Suppose we just JMP to the subroutine. When the subroutine is ready to return, it will go directly to the routine that called our program. Thus, with rare exceptions, JSR and RTS are identical to JMP. We've saved a byte and a little time.

Programmers working with limited memory find this kind of tightening up useful, and it often leads to further economies. For example, if there's a routine called DOG and one called CAT; and if DOG ends with JSR CAT:RTS; then the first step is to replace this with JMP CAT. Now, we won't need to jump to CAT if that subroutine immediately follows. Instead of jumping there, we'll just fall into it. Suddenly, two subroutines have become one — with two entry points.

There's another interesting use for this technique. Suppose you've written a subroutine SPC to print a space, and now you want to write a subroutine to print two spaces. You might start with the sequence JSR SPC:JSR SPC:RTS — but a little boiling down will generate the sequence:

```
SPC2   JSR  SPC
SPC    LDA  #$20
       JMP  PRINT
```

It seems odd to see a subroutine that starts out by calling the following instruction as a subroutine. But if you think of the way subroutines work, you'll see that it does a simple job: it executes the subroutine twice.

By the way, some theorists are very strong on the idea that all subroutines should have one entry point and one clearly defined exit. You'll have to decide on your own style. If you have lots of memory and processing time, you might prefer neatness. On the other hand, if you're trying to crowd a lot of programming into a small 2K ROM, you'll take all the economies you can get. ©

NEW

COMMODORE 64

80 COLUMN SCREEN - COLOR PROGRAM SALE $49.00

***Free** excellent P. D. programs!!*
- *Word Processor*
- *Electronic spreadsheet*
- *Data Base*
- *Modem Terminal Program*

Now you can program 80 Columns on the screen at one time! "In color or black and white". Get these excellent P. D. Programs **Free!**
"Word Processor — Spreadsheet — Data Base — Modem Terminal Program"
(Disk Only) List Price $59.00 **Sale $49.00** *Coupon Price $39.00 (Disk Only).

NEW

COMMODORE 64

FANTASTIC!! PROGRAMMERS AID

(Disk Program)

SALE $39.95

This is a must for all Programmers, New and Experienced! 33 New Basic Commands! Renumber, Move Sections, Merge Programs, Rename Variables, Trace and Edit Commands to find out exactly where the mistakes are! Easy to use and understand. Fantastic!!!
List Price $59.95 **Sale $39.95** *Coupon Price $29.95.

Add $3.00 for postage. Add $6.00 for CANADA, PUERTO RICO, HAWAII orders. WE DO NOT EXPORT TO OTHER COUNTRIES.
Enclose Cashiers Check, Money Order or Personal Check. Allow 14 days for delivery, 2 to 7 days for phone orders, 1 day express mail! Canada orders must be in U.S. dollars. We accept Visa and MasterCard. We ship C.O.D.

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

SANYO MONITOR SALE!!

9" Data Monitor

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

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

9" Screen - Green Text Display	**$ 79.00**
12" Screen - Green Text Display (anti-reflective screen)	**$ 99.00**
12" Screen - Amber Text Display (anti-reflective screen)	**$119.00**
14" Screen - Color Monitor (national brand)	**$249.00**

Display Monitors From Sanyo

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

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

SANYO

Official Video Products of the Los Angeles 1984 Olympics

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

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

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

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

PROGRAMMING THE TI

C. Regena

Computer Fun

The best news for TI owners this Christmas season is that Texas Instruments has reduced the price of its peripherals. One complaint about the TI-99/4A has been that the cost of the basic computer was quite reasonable, but if you wanted to add disk drives or a printer, the cost was out of sight. But that's not a valid complaint anymore. The peripheral expansion box with one disk drive, the disk controller card, and the 32K memory expansion card now have a total list price of $550 — I have seen advertisements of prices near $450.

The RS-232 interface card, needed to add a printer or a modem, lists for around $100. Therefore, since the computer itself sells for about $100, you can get a "complete system" for under $700. Although you can use other brands of printers and modems, the TI printer has been reduced to $500, and the TI modem to $100. All of this means that more TI owners will be getting the peripherals and discovering even more ways we can use our computers in our homes.

Computer Choreography

Since December is a festive time of year, the subject of this column is combining music with graphics to create a show I call "computer choreography." Two months ago I wrote about music on the TI-99/4A. This column is a continuation of that topic, with an explanation of one way to add graphics to the music. Remember, there are many ways to program — there's no *one* "correct" way. Your program is "correct" if it works the way you want it to when you run it. If it runs properly, you are successful.

Many programming books tell you to plan your program carefully by sketching a structure chart or writing different sections of coding. High school teachers often have students write out the program by hand before going to the computer. (Actually, often the real reason for this procedure is that the school doesn't have enough computers for the whole class.) If you are using a terminal or a mainframe computer and need to pay for computer time, you do need to plan carefully for efficiency. A home computer allows you to experiment to your heart's content — and even try out your program after every few lines if you wish. Although I usually do sketch out my graphics on graph paper, most of my programming is done by composing right at the console.

Choreography programs require a lot of experimentation, so it is almost better to compose at the console rather than plan each statement in order. Let's get right to an example. I chose a Christmas song that I would like to "play" on the computer. I looked up the music in a songbook then started translating notes. Each CALL SOUND statement contains a duration, a melody note and volume, and two accompaniment notes with their volumes. The duration is expressed in terms of a variable T, which is defined at the beginning of the program.

```
100 REM  SILENT NIGHT
110 T=400
120 CALL SOUND(T*1.5,392,4,330,8,13
    1,10)
130 CALL SOUND(T/2,440,4,349,8,131,
    10)
140 CALL SOUND(T,392,4,330,8,131,9)
150 CALL SOUND(3*T,330,4,262,6,196,
    9)
160 CALL SOUND(T*1.5,392,4,330,8,13
    1,10)
170 CALL SOUND(T/2,440,4,349,8,131,
    10)
180 CALL SOUND(T,392,4,330,8,131,9)
190 CALL SOUND(3*T,330,4,262,6,196,
    9)
200 CALL SOUND(2*T,587,2,349,4,247,
    8)
210 CALL SOUND(T,587,3,349,5,247,9)
220 CALL SOUND(3*T,494,3,294,6,196,
    9)
```

Try different values for T in line 110. For example, try T=600. Then try T=100. By programming the duration in terms of T, you only need to change line 110, not each of the CALL SOUND statements, to increase or decrease the speed of the song. Keep experimenting until you find the tempo you like.

You may write the three notes (frequencies) in the CALL SOUND statement in any order you wish. I usually write the melody note first so I can keep track of the tune. Also, if I later run out of memory I can more easily delete some of the accompaniment notes because I know the melody note is the first frequency.

Each frequency has a corresponding volume. I write the melody notes with a louder volume than the accompaniment notes in order to bring out the melody. Also, many times bass notes sound louder to us naturally, so we need to lower their volume.

By the way, our chart's lowest available note is low A on the bass clef (frequency 110), and you cannot use a frequency number less than 110. However, it is possible to get tones lower than low A. Comparing the tones to an electronic keyboard, Jerry Glaze of Las Vegas, Nevada, has come up with various numbers to get lower tones. He suggests you try this command to hear low G:

```
CALL SOUND(2ØØØ,1475,3Ø,1475,3Ø,147
5,3Ø,-4,1)
```

He specifies three music frequencies of 1475 with a volume of 30, plus the noise parameter of −4 with a volume of 1. Now change each of the 1475 numbers to 1293 and you'll hear low F (one line below the bass clef). Continuing downward, Jerry suggests the following numbers: 1227 – E; 1105 – D; 990 – C. (You may wish to adjust the numbers slightly.)

Adding Graphics To Music

Now let's add graphics. The actual picture I plan on paper first. I sketch out the main picture on graph paper 24 squares by 32 squares to correspond to the 24 rows by 32 columns on the screen. Any pictures that do not fit into the full squares are redrawn on 8 by 8 squares for the high resolution graphics. Then add line 105 CALL CLEAR to clear the screen before running the program. Now we're ready to begin by inserting graphics commands among the present sound commands.

First, you need to define graphics characters for later pictures using CALL CHAR STATEMENTS. This is where you really need to experiment. Try adding the following lines:

```
122 CALL CHAR(128,"Ø1Ø1Ø1Ø3Ø3FF7F1F
    ")
124 CALL CHAR(129,"ØFØF1F3E386Ø4")
```

Be sure those are zeros and not the letter O in the quotes of the character definitions. Now try running the program. It should sound the same as when you ran it without any graphics statements since the TI can play music while it is executing other commands. Depending on how long a note is held, you can define characters between sounds. In this case we were able to define two characters between the first note and the second. You may be able to define more characters, but if you put too many definitions between the sounds, there will be a gap between the notes — so you need to use fewer definitions or commands.

I stayed with just the two definitions between the first two notes. I decided to put the next definition after the third note. Add:

```
145 CALL CHAR(13Ø,"ØØ8Ø8ØCØCØFFFEFC
    ")
```

Line 150 is the sound corresponding to the word "night" in the song "Silent Night," so right after the music is played, I change the screen color to black with

```
152 CALL SCREEN(2)
```

This chord has a rather long duration, so let's define two more characters. Add:

```
154 CALL CHAR(131,"F8FØF8F83CØCØ6")
156 CALL CHAR(132,"ØØØ2Ø4Ø81Ø2Ø4")
```

Next I started drawing a star. In this case the screen is black and characters are naturally black with a transparent background, so any characters placed on the screen won't be seen until the color is changed. I didn't want the star to actually appear until after "holy night." To make the star, add the following statements:

```
162 CALL HCHAR(3,25,128)
164 CALL HCHAR(4,25,129)
166 CALL HCHAR(3,26,13Ø)
168 CALL HCHAR(4,26,131)
```

and after "night" in line 190,

```
192 CALL COLOR(13,16,1)
```

You can use this technique of drawing invisibly by first defining the colors of the character with a CALL COLOR statement to match whatever colors are already on the screen; then placing the characters on the screen with CALL HCHAR and CALL VCHAR; then making the characters visible with another CALL COLOR statement defining the visible colors.

After you add a few more character definitions and some HCHAR commands to draw on the screen, then RESequence the program segment, this is how it will look.

Program 1: "Silent Night"

```
1ØØ REM  SILENT NIGHT
11Ø CALL CLEAR
12Ø T=4ØØ
13Ø CALL SOUND(T*1.5,392,4,33Ø,8,13
    1,1Ø)
```

```
140 CALL CHAR(128,"0101010303FF7F1F
    ")
150 CALL CHAR(129,"0F0F1F3E38604")
160 CALL SOUND(T/2,440,4,349,8,131,
    10)
170 CALL SOUND(T,392,4,330,8,131,9)
180 CALL CHAR(130,"008080C0C0FFFEFC
    ")
190 CALL SOUND(3*T,330,4,262,6,196,
    9)
200 CALL SCREEN(2)
210 CALL CHAR(131,"F8F0F8F83C0C06")
220 CALL CHAR(132,"000204081020 4")
230 CALL SOUND(T*1.5,392,4,330,8,13
    1,10)
240 CALL HCHAR(3,25,128)
250 CALL HCHAR(4,25,129)
260 CALL HCHAR(3,26,130)
270 CALL HCHAR(4,26,131)
280 CALL SOUND(T/2,440,4,349,8,131,
    10)
290 CALL SOUND(T,392,4,330,8,131,9)
300 CALL CHAR(133,"0002020404080 8")
310 CALL SOUND(3*T,330,4,262,6,196,
    9)
320 CALL COLOR(13,16,1)
330 CALL CHAR(134,"00202040408 08")
340 CALL SOUND(2*T,587,2,349,4,247,
    8)
350 CALL HCHAR(4,24,132)
360 CALL HCHAR(5,23,132)
370 CALL HCHAR(6,22,132)
380 CALL SOUND(T,587,3,349,5,247,9)
390 CALL SOUND(3*T,494,3,294,6,196,
    9)
400 CALL HCHAR(5,25,133)
410 CALL HCHAR(6,25,134)
420 CALL HCHAR(7,24,133)
430 CALL HCHAR(8,24,134)
440 GOTO 440
```

The last line, 440 GOTO 440, keeps the picture on the screen until you press CLEAR (SHIFT C on the TI-99/4 or FCTN 4 on the TI-99/4A). I'm going to leave the rest of the song up to you. Since I'm not an artist, I often look at children's picture books or coloring books for picture ideas. For Christmas scenes, you can also try tracing Christmas stencils on graph paper then coloring the squares to plan your shapes. Computer choreography can be a lot of fun, and I know many people who have gotten interested in programming by first designing pictures with music.

A New Year's Present

I promised you a Christmas present, but I've decided to make it a New Year's present instead. I got my first computer for Christmas in 1980, and one of the first programs I wrote was the music for "Auld Lang Syne" with the screen showing 1980 turning into 1981. Each year I change the year and I change the graphics or music slightly. In 1981 I had TI Extended BASIC and made the number 1 out of sprites that moved off the screen to make room for 1982. This year I'm using the natural scrolling of PRINT statements to move 1983 off the screen while bringing in the new year.

I'm also including a TI Extended BASIC version (Program 3). To RUN it, you will need the TI Extended BASIC command module. It includes fireworks and champagne bubbles while the music is playing. In the character definitions, up to four characters may be defined in one command. Trailing REMark statements are allowed with the exclamation point, so the words (or syllables) to the music are written along with the CALL SOUND statements.

If you want to use these programs right at midnight, type RUN then press ENTER at 31 seconds before midnight for the regular TI BASIC program, or 25 seconds before midnight for the TI Extended BASIC program. The year 1984 will be in place exactly for the new year.

Have a happy holiday season!

Program 2: "Auld Lang Syne" (TI BASIC)

```
100 REM  AULD LANG SYNE
110 CALL CLEAR
120 CALL SCREEN(4)
130 CALL CHAR(96,"000101030307070F"
    )
140 CALL CHAR(97,"0F1F1F3F3F7F7FFF"
    )
150 CALL CHAR(98,"FFFFFFFFFFFFFFFF"
    )
160 T=600
170 CALL SOUND(T*1.1,262,5)
180 CALL CHAR(104,"00030F1F3F3F7F7F
    ")
190 CALL CHAR(105,"7F7F3F3F1F0F03")
200 CALL CHAR(106,"00C0F0F8FCFCFEFE
    ")
210 CALL CHAR(107,"FEFEFCFCF8F0C0")
220 CALL SOUND(T*1.5,349,5,262,12,1
    75,15)
230 CALL VCHAR(8,5,98,9)
240 CALL VCHAR(8,4,96)
250 CALL VCHAR(9,4,97)
260 CALL SOUND(T/2,349,5,262,12,196
    ,15)
270 CALL SOUND(T,349,4,262,12,220,1
    5)
280 CALL CHAR(108,"7F3F1F070F1F3F7F
    ")
290 CALL CHAR(109,"FEFCF8C0F0F8FCFE
    ")
300 CALL SOUND(T,440,5,349,12,175,1
    5)
310 CALL HCHAR(8,10,98,3)
320 CALL HCHAR(8,9,104)
330 CALL VCHAR(9,9,98,3)
340 CALL SOUND(T*1.5,392,5,330,12,1
    31,15)
350 CALL HCHAR(12,9,105)
360 CALL HCHAR(12,10,98,3)
370 CALL VCHAR(8,13,106)
380 CALL VCHAR(9,13,98,7)
390 CALL SOUND(T/2,349,5,294,12,131
    ,15)
400 CALL SOUND(T,392,5,330,12,131,1
    5)
410 CALL HCHAR(16,13,107)
420 CALL HCHAR(16,10,98,3)
```

```
430 CALL HCHAR(15,9,98)
440 CALL HCHAR(16,9,105)
450 CALL SOUND(T,440,5,330,12,131,1
    5)
460 CALL HCHAR(8,17,104)
470 CALL HCHAR(8,18,98,3)
480 CALL HCHAR(8,21,106)
490 CALL VCHAR(9,21,98,3)
500 CALL SOUND(T*1.5,349,6,220,12,1
    75,15)
510 CALL VCHAR(9,17,98,3)
520 CALL HCHAR(12,17,108)
530 CALL HCHAR(12,18,98,3)
540 CALL HCHAR(12,21,109)
550 CALL SOUND(T/2,349,6,220,12,175
    ,15)
560 CALL VCHAR(13,17,98,3)
570 CALL SOUND(T,440,4,349,12,175,1
    5)
580 CALL VCHAR(16,17,105)
590 CALL HCHAR(16,18,98,3)
600 CALL HCHAR(16,21,107)
610 CALL SOUND(T,523,3,349,10,175,1
    3)
620 CALL VCHAR(13,21,98,3)
630 CALL HCHAR(9,25,98)
640 CALL HCHAR(8,25,104)
650 CALL SOUND(3*T,587,2,349,8,233,
    10)
660 CALL HCHAR(8,26,98,3)
670 CALL HCHAR(8,29,106)
680 CALL VCHAR(9,29,98,3)
690 CALL HCHAR(12,27,98,2)
700 CALL HCHAR(12,29,109)
710 CALL VCHAR(13,29,98,3)
720 CALL HCHAR(16,29,107)
730 CALL HCHAR(16,26,98,3)
740 CALL HCHAR(16,25,105)
750 CALL HCHAR(15,25,98)
760 CALL SOUND(T,587,2,349,8,233,10
    )
770 CALL SCREEN(8)
780 PRINT " b{3 SPACES}hbbbj
    {3 SPACES}hbbbj{3 SPACES}a"
790 CALL SOUND(T*1.5,523,3,349,10,2
    20,13)
800 PRINT " ab{3 SPACES}b
    {3 SPACES}b{3 SPACES}b
    {3 SPACES}b{3 SPACES}b"
810 CALL SOUND(T/2,440,4,349,12,175
    ,15)
820 PRINT "  b{3 SPACES}b
    {3 SPACES}b{3 SPACES}b
    {3 SPACES}b{3 SPACES}b"
830 CALL SOUND(T,440,6,349,12,175,1
    5)
840 PRINT "  b{3 SPACES}b
    {3 SPACES}b{3 SPACES}b
    {3 SPACES}b{3 SPACES}b  b"
850 CALL SOUND(T,349,6,220,12,175,1
    5)
860 PRINT "  b{3 SPACES}ibbbb
    {3 SPACES}lbbbm{3 SPACES}b  b"
870 CALL SOUND(T*1.5,392,6,330,12,1
    31,15)
880 PRINT "  b{7 SPACES}b
    {3 SPACES}b{3 SPACES}b
    {3 SPACES}bbbbb"
890 CALL SOUND(T/2,349,6,294,12,131
    ,15)
900 PRINT "  b{7 SPACES}b
    {3 SPACES}b{3 SPACES}b
    {6 SPACES}b"
910 CALL SOUND(T,392,6,330,12,131,1
    5)
920 PRINT "  b{3 SPACES}b
    {3 SPACES}b{3 SPACES}b
    {3 SPACES}b{6 SPACES}b"
930 CALL SOUND(T,440,6,330,12,131,1
    5)
940 PRINT "  b{3 SPACES}ibbbk
    {3 SPACES}ibbbk{6 SPACES}b"
950 CALL SOUND(T*1.5,349,6,294,12,1
    47,15)
960 PRINT
970 CALL SOUND(T/2,294,7,220,12,147
    ,15)
980 PRINT
990 CALL SOUND(T,294,7,233,12,117,1
    5)
1000 PRINT
1010 CALL SOUND(T,262,8,233,14,131,
     16)
1020 PRINT
1030 CALL SOUND(3*T,349,8,220,15,17
     5,17)
1040 PRINT :::
1050 CALL SOUND(T,587,5,349,12,175,
     15)
1060 CALL COLOR(9,5,1)
1070 CALL COLOR(10,5,1)
1080 CALL COLOR(2,7,1)
1090 CALL SOUND(T*1.5,523,5,349,12,
     175,15)
1100 FOR I=5 TO 25 STEP 5
1110 CALL HCHAR(6,I,42)
1120 NEXT I
1130 CALL SOUND(T/2,440,6,262,15)
1140 CALL SOUND(T,440,6,349,12,175,
     15)
1150 CALL HCHAR(4,13,42)
1160 CALL HCHAR(4,17,42)
1170 CALL HCHAR(2,11,42)
1180 CALL HCHAR(2,19,42)
1190 CALL SOUND(T,349,6,110,18)
1200 CALL HCHAR(4,8,42)
1210 CALL HCHAR(2,6,42)
1220 CALL HCHAR(4,22,42)
1230 CALL HCHAR(2,24,42)
1240 CALL SOUND(T*1.5,392,6,330,14,
     131,16)
1250 FOR I=5 TO 25 STEP 5
1260 CALL HCHAR(18,I,42)
1270 NEXT I
1280 CALL SOUND(T/2,349,6,294,12,13
     1,17)
1290 CALL SCREEN(8)
1300 CALL SOUND(T,392,7,330,15,131,
     17)
1310 CALL HCHAR(20,13,42)
1320 CALL HCHAR(20,17,42)
1330 CALL HCHAR(22,11,42)
1340 CALL HCHAR(22,19,42)
1350 CALL SOUND(T,587,6,330,14,131,
     16)
1360 CALL HCHAR(20,8,42)
1370 CALL HCHAR(22,6,42)
1380 CALL HCHAR(20,22,42)
1390 CALL HCHAR(22,24,42)
1400 CALL SOUND(T*1.5,262,6,349,14,
```

```
     131,16)
1410 CALL HCHAR(4,3,42)
1420 CALL HCHAR(2,1,42)
1430 CALL HCHAR(4,27,42)
1440 CALL HCHAR(2,29,42)
1450 CALL COLOR(9,7,1)
1460 CALL COLOR(10,7,1)
1470 CALL SOUND(T/2,440,7,131,16)
1480 CALL SOUND(T,440,6,349,14,175,
     16)
1490 CALL HCHAR(20,3,42)
1500 CALL HCHAR(22,1,42)
1510 CALL HCHAR(20,27,42)
1520 CALL HCHAR(22,29,42)
1530 CALL SOUND(T,523,5,220,15)
1540 CALL SOUND(3*T,587,3,349,12,23
     3,14)
1550 CALL COLOR(2,16,1)
1560 CALL SOUND(T,698,2,349,13,233,
     15)
1570 CALL COLOR(2,12,1)
1580 CALL SOUND(T*1.5,523,3,349,12,
     220,14)
1590 CALL COLOR(9,11,1)
1600 CALL COLOR(10,11,1)
1610 CALL SOUND(T/2,440,4,349,13,17
     5,15)
1620 CALL SOUND(T,440,4,349,13,175,
     15)
1630 CALL COLOR(2,5,1)
1640 CALL SOUND(T,349,5,262,13,110,
     15)
1650 CALL COLOR(2,16,1)
1660 CALL SOUND(T*1.5,392,5,330,13,
     131,15)
1670 CALL COLOR(9,14,1)
1680 CALL COLOR(10,14,1)
1690 CALL COLOR(2,7,1)
1700 CALL SOUND(T/2,349,5,294,13,13
     1,15)
1710 CALL COLOR(2,16,1)
1720 CALL SOUND(T,392,5,330,12,131,
     15)
1730 CALL COLOR(2,12,1)
1740 CALL SOUND(T/2,440,5,330,13,13
     9,15)
1750 CALL COLOR(2,16,1)
1760 CALL SOUND(T/2,392,5,330,13,13
     9,15)
1770 CALL COLOR(2,3,1)
1780 CALL SOUND(T*1.5,349,5,294,14,
     147,16)
1790 CALL COLOR(9,16,1)
1800 CALL COLOR(10,16,1)
1810 CALL COLOR(2,16,1)
1820 CALL SOUND(T/2,294,6,220,14,17
     5,16)
1830 CALL COLOR(2,6,1)
1840 CALL SOUND(T,294,7,233,15,117,
     17)
1850 CALL COLOR(2,14,1)
1860 CALL SCREEN(11)
1870 CALL SOUND(T,262,7,165,15,131,
     17)
1880 CALL COLOR(2,12,1)
1890 CALL SOUND(4*T,349,6,220,15,17
     5,17)
1900 CALL SCREEN(8)
1910 CALL COLOR(9,7,1)
1920 CALL COLOR(10,7,1)
1930 CALL COLOR(2,16,1)
1940 CALL COLOR(2,14,1)
1950 CALL COLOR(2,16,1)
1960 CALL COLOR(2,11,1)
1970 CALL COLOR(2,16,1)
1980 CALL COLOR(2,7,1)
1990 CALL COLOR(2,16,1)
2000 CALL COLOR(2,6,1)
2010 GOTO 1930
2020 END
```

Program 3:
"Auld Lang Syne" (TI Extended BASIC)

```
90 REM TI EXTENDED BASIC
100 REM AULD LANG SYNE
110 CALL CLEAR :: CALL SCREEN(4)
120 CALL CHAR(96,"000101030307070F0
    F1F1F3F3F7F7FFFFFFFFFFFFFFFFFFFF
    FFFFFFFFFFFFFFFF")
130 T=600
140 CALL SOUND(T*1.1,262,5)!SHOULD
150 CALL CHAR(104,"00030F1F3F3F7F7F
    7F7F3F3F1F0F030000C0F0F8FCFCFEF
    EFEFEFCFCF8F0C000")
160 CALL SOUND(T*1.5,349,5,262,12,1
    75,15)!AULD
170 CALL VCHAR(8,5,98,9)
180 CALL VCHAR(8,4,96)
190 CALL VCHAR(9,4,97)
200 CALL SOUND(T/2,349,5,262,12,196
    ,15)!AC-
210 CALL SOUND(T,349,4,262,12,220,1
    5)!QUAINT-
220 CALL CHAR(108,"7F3F1F070F1F3F7F
    FEFCF8C0F0F8FCFE")
230 CALL SOUND(T,440,5,349,12,175,1
    5)!ANCE
240 CALL HCHAR(8,10,98,3)
250 CALL HCHAR(8,9,104)
260 CALL VCHAR(9,9,98,3)
270 CALL SOUND(T*1.5,392,5,330,12,1
    31,15)!BE
280 CALL HCHAR(12,9,105)
290 CALL HCHAR(12,10,98,3)
300 CALL VCHAR(8,13,106)
310 CALL VCHAR(9,13,98,7)
320 CALL SOUND(T/2,349,5,294,12,131
    ,15)!FOR-
330 CALL CHAR(94,"10387CD692103844"
    )
340 CALL SOUND(T,392,5,330,12,131,1
    5)!GOT
350 CALL HCHAR(16,13,107)
360 CALL HCHAR(16,10,98,3)
370 CALL HCHAR(15,9,98)
380 CALL HCHAR(16,9,105)
390 CALL SOUND(T,440,5,330,12,131,1
    5)!AND
400 CALL HCHAR(8,17,104)
410 CALL HCHAR(8,18,98,3)
420 CALL HCHAR(8,21,106)
430 CALL VCHAR(9,21,98,3)
440 CALL SOUND(T*1.5,349,6,220,12,1
    75,15)!NEV-
450 CALL VCHAR(9,17,98,3)
460 CALL HCHAR(12,17,108)
470 CALL HCHAR(12,18,98,3)
480 CALL HCHAR(12,21,109)
490 CALL SOUND(T/2,349,6,220,12,175
    ,15)!ER
```

```
500 CALL VCHAR(13,17,98,3)
510 CALL SOUND(T,440,4,349,12,175,1
    5)!BROUGHT
520 CALL VCHAR(16,17,105)
530 CALL HCHAR(16,18,98,3)
540 CALL HCHAR(16,21,107)
550 CALL SOUND(T,523,3,349,10,175,1
    3)!TO
560 CALL VCHAR(13,21,98,3)
570 CALL HCHAR(9,25,98)
580 CALL HCHAR(8,25,104)
590 CALL SOUND(3*T,587,2,349,8,233,
    10)!MIND
600 CALL HCHAR(8,26,98,3)
610 CALL HCHAR(8,29,106)
620 CALL VCHAR(9,29,98,3)
630 CALL HCHAR(12,27,98,2)
640 CALL HCHAR(12,29,109)
650 CALL VCHAR(13,29,98,3)
660 CALL HCHAR(16,29,107)
670 CALL HCHAR(16,26,98,3)
680 CALL HCHAR(16,25,105)
690 CALL HCHAR(15,25,98)
700 CALL SOUND(T,587,2,349,8,233,10
    )!SHOULD
710 CALL SCREEN(8)
720 PRINT " 'b{3 SPACES}hbbbj
    {3 SPACES}hbbbj{3 SPACES}a"
730 CALL SOUND(T*1.5,523,3,349,10,2
    20,13)!AULD
740 PRINT " ab{3 SPACES}b
    {3 SPACES}b{3 SPACES}b
    {3 SPACES}b{3 SPACES}b"
750 CALL CHAR(33,"1010545454545444"
    )
760 CALL SOUND(T/2,440,4,349,12,175
    ,15)!AC-
770 PRINT "  b{3 SPACES}b
    {3 SPACES}b{3 SPACES}b
    {3 SPACES}b{3 SPACES}b"
780 CALL SOUND(T,440,6,349,12,175,1
    5)!QUAINT-
790 PRINT "  b{3 SPACES}b
    {3 SPACES}b{3 SPACES}b
    {3 SPACES}b{3 SPACES}b  b"
800 CALL SOUND(T,349,6,220,12,175,1
    5)!ANCE
810 PRINT "  b{3 SPACES}ibbbb
    {3 SPACES}lbbbm{3 SPACES}b  b"
820 CALL SOUND(T*1.5,392,6,330,12,1
    31,15)!BE
830 PRINT "  b{7 SPACES}b
    {3 SPACES}b{3 SPACES}b
    {3 SPACES}bbbbb"
840 CALL SOUND(T/2,349,6,294,12,131
    ,15)!FOR-
850 PRINT "  b{7 SPACES}b
    {3 SPACES}b{3 SPACES}b
    {6 SPACES}b"
860 CALL SOUND(T,392,6,330,12,131,1
    5)!GOT
870 PRINT "  b{3 SPACES}b
    {3 SPACES}b{3 SPACES}b
    {3 SPACES}b{6 SPACES}b"
880 CALL SOUND(T,440,6,330,12,131,1
    5)!AND
890 PRINT "  b{3 SPACES}ibbbk
    {3 SPACES}ibbbk{6 SPACES}b"
900 CALL SOUND(T*1.5,349,6,294,12,1
    47,15)!DAYS
910 PRINT
920 CALL SOUND(T/2,294,7,220,12,147
    ,15)!OF
930 PRINT
940 CALL SOUND(T,294,7,233,12,117,1
    5)!AULD
950 PRINT
960 CALL SOUND(T,262,8,233,14,131,1
    6)!LANG
970 PRINT
980 CALL SOUND(3*T,349,8,220,15,175
    ,17)!SYNE
990 PRINT : : :
1000 CALL SOUND(T,587,5,349,12,175,
     15)!FOR
1010 CALL COLOR(9,5,1):: CALL COLOR
     (10,5,1)
1020 CALL SOUND(T*1.5,523,5,349,12,
     175,15)!AULD
1030 CALL MAGNIFY(1)
1040 CALL CHAR(120,"92442892284492"
     )
1050 CALL SPRITE(#1,94,13,192,115,-
     9,0)
1060 CALL SPRITE(#28,33,16,198,115,
     -9,0)
1070 CALL SOUND(T/2,440,6,262,15)
1080 CALL SOUND(T,440,6,349,12,175,
     15)!LANG
1090 CALL CHAR(124,"003C424242423C"
     )
1100 CALL SOUND(T,349,6,110,18)
1110 CALL SOUND(T*1.5,392,6,330,14,
     131,16)!SYNE
1120 CALL DELSPRITE(#1,#28)
1130 FOR I=1 TO 10
1140 CALL SPRITE(#I,120,7,90,115)
1150 NEXT I
1160 CALL SOUND(T/2,349,6,294,12,13
     1,17)!MY
1170 CALL SCREEN(8)
1180 CALL MOTION(#1,-10,-10)
1190 CALL MOTION(#2,-10,10)
1200 CALL SOUND(T,392,7,330,15,131,
     17)!DEAR
1210 CALL MOTION(#3,-10,5)
1220 CALL MOTION(#4,-10,-5)
1230 CALL MOTION(#5,-10,0)
1240 CALL MOTION(#6,10,-10)
1250 CALL MOTION(#7,10,10)
1260 CALL SOUND(T,587,6,330,14,131,
     16)!FOR
1270 CALL MOTION(#8,10,-5)
1280 CALL MOTION(#9,10,5)
1290 CALL MOTION(#10,10,0)
1300 CALL SOUND(T*1.5,262,6,349,14,
     131,16)!AULD
1310 CALL COLOR(9,7,1):: CALL COLOR
     (10,7,1)
1320 FOR I=1 TO 5 :: CALL MOTION(#I
     ,0,0):: NEXT I
1330 CALL SOUND(T/2,440,7,131,16)
1340 CALL SOUND(T,440,6,349,14,175,
     16)!LANG
1350 FOR I=6 TO 10 :: CALL MOTION(#
     I,0,0):: NEXT I
1360 CALL SOUND(T,523,5,220,15)
1370 CALL SOUND(3*T,587,3,349,12,23
     3,14)!SYNE
1380 C=16
```

```
1390 CALL SPRITE(#11,42,C,90,115,-1
     0,-16)
1400 CALL SPRITE(#12,42,C,90,115,-1
     0,18)
1410 CALL SPRITE(#13,42,C,90,115,-1
     1,-8)
1420 CALL SPRITE(#14,42,C,90,115,-1
     1,8)
1430 CALL SPRITE(#15,42,C,90,115,-1
     2,0)
1440 CALL SPRITE(#16,42,C,90,115,9,
     -16)
1450 CALL SPRITE(#17,42,C,90,115,9,
     18)
1460 CALL SPRITE(#18,42,C,90,115,13
     ,-9)
1470 CALL SPRITE(#19,42,C,90,115,13
     ,9)
1480 CALL SPRITE(#20,42,C,90,115,15
     ,0)
1490 CALL SOUND(T,698,2,349,13,233,
     15)!WE'LL
1500 FOR I=11 TO 20 :: CALL MOTION(
     #I,0,0):: NEXT I
1510 CALL SOUND(T*1.5,523,3,349,12,
     220,14)!TAKE
1520 CALL COLOR(9,11,1):: CALL COLO
     R(10,11,1)
1530 CALL SOUND(T/2,440,4,349,13,17
     5,15)!A
1540 CALL SOUND(T,440,4,349,13,175,
     15)!CUP
1550 CALL SPRITE(#21,124,5,192,30,-
     12,0)
1560 CALL SPRITE(#22,124,5,192,240,
     -7,0)
1570 CALL SPRITE(#23,124,5,192,64,-
     20,0)
1580 CALL SPRITE(#24,124,5,192,192,
     -24,0)
1590 CALL SOUND(T,349,5,262,13,110,
     15)!O'
1600 CALL SPRITE(#25,124,5,192,103,
     -14,0)
1610 CALL SPRITE(#26,124,5,192,164,
     -30,0)
1620 CALL SPRITE(#27,124,5,192,120,
     -23,0)
1630 CALL SOUND(T*1.5,392,5,330,13,
     131,15)!KIND-
1640 CALL COLOR(9,14,1):: CALL COLO
     R(10,14,1)
1650 CALL SOUND(T/2,349,5,294,13,13
     1,15)!NESS
1660 CALL SOUND(T,392,5,330,12,131,
     15)!YET
1670 CALL SOUND(T/2,440,5,330,13,13
     9,15)!FOR
1680 CALL SOUND(T/2,392,5,330,13,13
     9,15)
1690 CALL SOUND(T*1.5,349,5,294,14,
     147,16)!AULD
1700 CALL COLOR(9,16,1):: CALL COLO
     R(10,16,1)
1710 CALL SOUND(T/2,294,6,220,14,17
     5,16)
1720 CALL SOUND(T,294,7,233,15,117,
     17)!LANG
1730 CALL SCREEN(11)
1740 CALL SOUND(T,262,7,165,15,131,
     17)
1750 CALL SOUND(4*T,349,6,220,15,17
     5,17)!SYNE
1760 CALL SCREEN(8)
1770 CALL COLOR(9,7,1):: CALL COLOR
     (10,7,1)
1780 FOR I=1 TO 20 :: CALL COLOR(#I
     ,16):: NEXT I
1790 FOR I=1 TO 20 :: CALL COLOR(#I
     ,14):: NEXT I
1800 FOR I=1 TO 20 :: CALL COLOR(#I
     ,12):: NEXT I
1810 FOR I=1 TO 20 :: CALL COLOR(#I
     ,7):: NEXT I
1820 GOTO 1780
1830 END
```

©

ART MUSEUM

Floyd Beaston

Both the Commodore VIC and 64 have graphics characters right on the keys. This program lets you take advantage of these graphics by allowing you to SAVE and LOAD screen pictures made using character graphics.

My eight-year-old son loves to "draw" artwork on the screen using combinations of the graphics symbols on the keys. Because the "artworks" vanished forever when we turned off the computer, my son became more and more frustrated.

These programs for the VIC and 64 were written to help with this problem by allowing you to SAVE and LOAD all characters, including graphics symbols, on the screen.

To use the VIC version, first remove any expansion board and then type in Program 1. Then enter this line:

CLR:POKE46,PEEK(46)+4

and SAVE to disk or tape.

Operating The VIC Version

If you wish to draw a picture (to later SAVE), LOAD the program and change line 1 to:

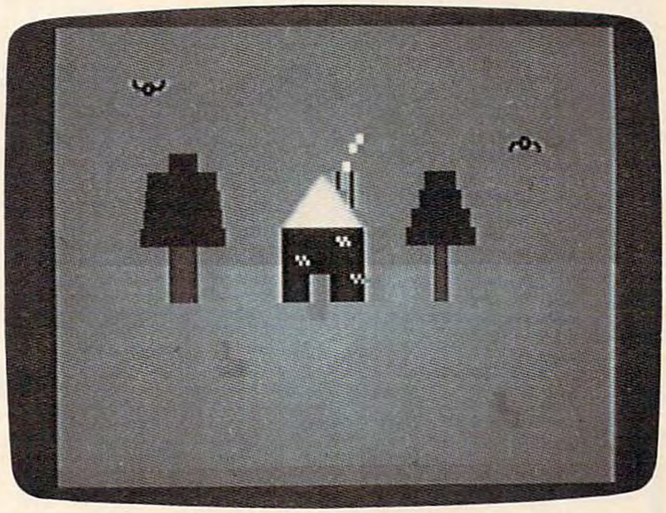
1 REM

Next, clear your screen and begin drawing. When you are finished, change the cursor color to match the background color, then type RUN. (You won't be able to see the command RUN since it will be the same color as the background.) In a few seconds, change the cursor color back to a visible color and then SAVE the program to tape or disk. Your screen will also be saved.

To retrieve your picture, LOAD the program from tape or disk and change line 1 to:

1 GOTO20

This will magically return your picture to the screen.



"Art Museum" can save any screen drawing to tape or disk. 64 version.

Program 1: Art Museum (VIC Version)

```
Ø S=768Ø:C=384ØØ:GOSUB63999
1 GOTO2Ø
1Ø FORJ=ØTO5Ø5:POKEML+J,PEEK(S+J):POKEML+
   5Ø6+J,PEEK(C+J):NEXT:END
2Ø FORJ=ØTO5Ø5:POKES+J,PEEK(ML+J):POKEC+J
   ,PEEK(ML+5Ø6+J):NEXT:PRINT"{HOME}";
21 GOTO21
63999 ML=PEEK(61)+PEEK(62)*256+31:RETURN
```

Program 2: Art Museum (64 Version)

```
1 GOSUB4Ø1Ø:INPUT "{WHT}{CLR}LOAD FILE";W
  $:IFW$="N"THENPRINT"{CLR}":END
```

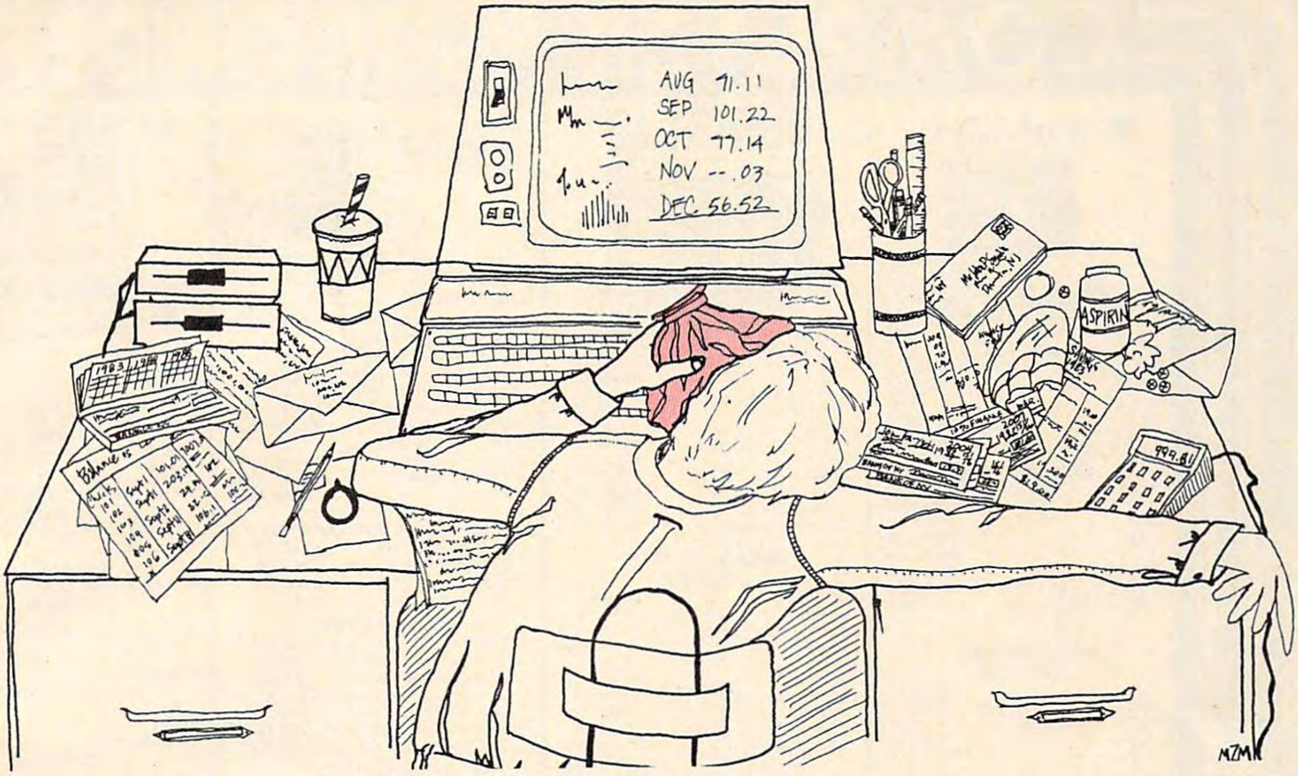
64 Notes

The 64 version of "Art Museum" (Program 2) stores the contents of screen memory at 16384 ($4000) and the contents of color memory at 1750. To use this version, first type in and SAVE the program, then draw your picture on the screen using the cursor control keys and character graphics. When your picture is complete, change the cursor color to the background color and then invisibly type GOTO 10 and press RETURN. Then press S (for SAVE). This saves your screen creation at 16384. After a wait of about 25 seconds, change the cursor color to a visible color and clear the screen.

If you wish to SAVE your screen to tape or disk, type GOSUB 4010:GOTO 1000 and press RETURN. You will then be prompted for filename and storage medium (tape or disk). After responding to these prompts, your screen will be saved to disk or tape. To LOAD a file, type RUN and the program will prompt for filename and storage medium. Once your file is loaded, type GOTO10 and hit any key except S. Your stored file will gradually appear on the screen.

```
2 INPUT"DISK OR TAPE";E$:IFE$="D"THENE=8:
  GOTO1900
3 E=1:GOTO1900
10 POKE55,255:POKE56,63
20 FORT=0TO200:NEXT
30 GETA$:IFA$=""THEN30
35 CO=55296:SC=1024:DR=16384:CR=DR+1024
40 ON((A$="S")+2)GOTO100,200
100 FORT=0TO999:POKEDR+T,PEEK(SC+T)
110 POKECR+T,(PEEK(CO+T)AND15)
120 NEXT:PRINT"{HOME}":END
200 FORT=0TO999:POKESC+T,PEEK(DR+T)
210 POKECO+T,PEEK(CR+T)
220 NEXT:PRINT"{HOME}":END
1000 REM SAVE SCREEN
1010 INPUT"SAVE SCREEN Y OR N";S$
1020 IF S$="N"THEN END
1021 POKE250,0:POKE251,64
1022 POKE252,0:POKE253,96
1030 INPUT"FILE NAME FOR SCREEN";F$
1035 F$="0:"+F$
1036 INPUT"{WHT}DISK OR TAPE";E$:IFE$="D"
     THENE=8:GOTO1040
1037 E=1
1040 OPEN1,E,1,F$:SYS49152:CLOSE1:END
1900 INPUT "FILENAME";L$:L$="0:"+L$
2000 OPEN1,E,0,L$:SYS49162:CLOSE1:END
4010 I=49152
4020 READ A:IF A=256 THEN RETURN
4030 POKE I,A:I=I+1:GOTO 4020
49152 DATA 166,252,164,253,169,250,32
49160 DATA 216,255,96,165,184,166,186
49168 DATA 160,255,32,186,255,169,0
49176 DATA 162,0,160,64,32,213,255
49184 DATA 96,256
```

©



Computerized Finance Can Be Quite A Headache!

Now "COMP-U-CHECK" Spells Relief.

By **HOT DATA**
For COMMODORE 64*
sug. retail $64.95

Features:

- Check writing
- Unlimited number of checking accounts
- Account balancing and reconciliation
- Unlimited number of budget categories
- Easy auto teller transactions
- Unlimited credit account management
- Tax reports in minutes, including checking and credit
- Supports all printers and interfaces
- Suitable for both home and business

Plus:

- Simple (letter format) word processing with automatic addressing (form letters, too)

Single disk drive required

Coming soon:

- Automatic bill paying with our expansion chassis light and appliance controller. Controls lights and appliances while running other programs.

Watch for us
We're **HOT DATA**
Santa Monica, CA

Ask your dealer, or
Call us direct at (213) 393-6405
VISA and MC

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

COMPUTER MAIL ORDER

TeleVideo

TERMINALS

910	$559.00
912	$689.00
920	$739.00
925	$719.00
950	$929.00
970	CALL

COMPUTERS

800A	$1099.00
802	$2699.00
803	$1949.00
802H	$4695.00
806/20	$4999.00
816/40	$9199.00
1602	$3399.00
1603	CALL

MODEMS

HAYES

Smart	$219.00
Smart 1200 (1200 Baud)	$519.00
Chronograph	$199.00
Micromodem 100	$309.00
Micromodem II	$279.00
Micromodem II (with term)	$299.00
Smart Com II	$99.00
Smart 1200B	$469.00

NOVATION

J-Cat	$119.00
Cat	$144.00
D-Cat	$159.00
103 Smart Cat	$189.00
Apple Cat II	$279.00
103/212 Smart Cat	$439.00
212 Apple Cat II	$609.00
Apple Cat II 212 Upgrade	$309.00

ANCHOR

Mark I (RS-232)	$79.00
Mark II (Atari)	$79.00
Mark III (T. I. 99)	$109.00
Mark IV (CBM-PET)	$125.00
Mark V (Osborne)	$95.00
Mark VI (IBM-PC)	$179.00
Mark VII (Auto Ans/Auto Dial)	$119.00
Mark VIII	$269.00
TRS-80 Color Computer	$99.00
9 Volt Power Supply	$9.00

ZENITH

ZT1 Terminal	$369.00

EAGLE

IIE-1	$1369.00
IIE-2	$1649.00
IIE-3	$2399.00
IIE-4	$3199.00
PC-E	$1579.00
PC-1	$2399.00
PC-2	$2799.00
PC-XL	$3599.00
1620	$3599.00
1630	$5499.00
1640	$6499.00
Cyma Software	CALL

MONITORS

AMDEK

300G	$149.00
300A	$159.00
310A	$169.00
Color I	$279.00
Color I plus	$299.00
Color II	$399.00
Color III	$349.00
Color IV	$999.00

USI

Pi 1, 9" G	$99.00
Pi 2, 12" G	$119.00
Pi 3, 12" A	$149.00
Pi 4, 9" A	$139.00
1400 Color	$279.00

ZENITH

ZVM 122A	$109.00
ZVM 123G	$99.00

BMC

12" Green	$85.00
9191 AU 13" Color	$249.00

TAXAN

12 N Green	$129.00
12 A Amber	$139.00

PANASONIC

TR 120 Hi-res. Green	$149.00
CT 160 Dual Mode Color	$279.00

NEC

JB 1260	$119.00
JB 1201	$149.00
JB 1205	$169.00
JC 1215	$299.00
JC 1216	$429.00
JC 1203	$469.00

GORILLA

12" Green	$89.00

IBM

NEC 3550 Printer.... $1799

PERCOM/TANDON

DRIVES

5¼" 320K Floppy	$249.00
5 Meg Hard w/Controller	$1399.00
10 Meg Hard w/Controller	$1699.00
15 Meg Hard w/Controller	$2095.00
20 Meg Hard w/Controller	$2399.00

AMDEK

310A Amber Monitor	$169.00
DXY 100 Plotter	$599.00
Color II	$399.00

AST RESEARCH, INC.

Six Pak Plus...from	$279.00
Combo Plus II...from	$279.00
Mega Plus...from	$309.00
I/O Plus II...from	$169.00

QUADRAM

Quadlink	$549.00
Quadboard...as low as	$309.00
Quad 512 Plus...as low as	$259.00
Quadcolor...as low as	$219.00
Chronograph	$89.00
Parallel Interface Board	$89.00
64K RAM Chips Kit	$79.00

MICRO PRO

Word Star/Mail Merge	$319.00
InfoStar	$299.00
Spell Star	$159.00
CalcStar	$159.00

MICROSTUF

Crosstalk	$129.00

MICROSOFT

Multiplan	$179.00

ASHTON-TATE

D-Base II	$419.00

IUS

EasyWriter II	$219.00
EasySpeller	$119.00
EasyFiler	$239.00

CONTINENTAL SOFTWARE

1st Class Mail/Form Letter	$79.00
The Home Accountant Plus	$99.00

SYNAPSE

File Manager	$99.00

LOTUS

123	$329.00

PFS

	APPLE	IBM
File	$79.00	$89.00
Report	$79.00	$79.00
Graph	$79.00	$89.00
Write	n/a	$89.00

KRAFT

IBM Joystick	$55.00
IBM Paddles	$39.00

PROFESSIONAL SOFTWARE

PC Plus Word Processing	$299.00

SANYO

MBC-555PC	CALL
MBC 1100	$1599.00
MBC 1150	$2099.00
MBC 1200	$1999.00
MBC 1250	$2399.00
FDD 3200-320K Drive	$399.00
FDD 6400-64K Drive	$499.00
PR 5500 Printer	$699.00

PRINTERS

EPSON

MX80 FT, MX100, RX80, FX80, FX100	CALL

OKIDATA

82, 83, 84, 92, 93	CALL

STAR

Delta 10	$559.00
Gemini 10X	$299.00
Gemini P15	$449.00
Serial Board	$75.00

SMITH CORONA

TP-1	$469.00
TP-2	CALL
Tractor Feed	$119.00

C.ITOH

Gorilla	$209.00
Prowriter 8510P	$379.00
Prowriter 1550P	$689.00
Starwriter F10-40P	$1149.00
Printmaster F10-55P	$1569.00
Tractor Feed	$199.00

DAISYWRITER

2000 Letter Quality	$999.00
2500...NEW	CALL
Tractor Feed	$109.00

DIABLO

620	$949.00
630	$1769.00

IDS

Call for ALL Configurations on IDS PRISM PRINTERS.

NEC

8023	$399.00
8025	$729.00
3510	$1449.00
3530	$1499.00
3550	$1799.00
7710/7730	$1999.00

BMC

401 Letter Quality	$699.00
RX-80	$259.00

COMREX

Comwriter II Parallel Printer	$499.00

CABLES & CONNECTIONS

Atari to Parallel	$29.00
Atari to Serial	$29.00
Apple to Parallel	$69.00
Apple to Parallel/Graphics	$99.00
Apple to Serial	$89.00
IBM to Parallel	$35.00
IBM to Serial	$29.00
Parallel to Parallel	$29.00
Serial to Serial	$29.00
Grappler Plus	$129.00
PKASO	$139.00
Atari to Modem Cable	$29.00
CBM 64 to IEEE Board	$79.00
Apple 80-Column Card	$159.00
CBM Pet to Parallel	$99.00
CBM Pet to Serial	$89.00

PAPER SUPPLIES

1 or 2" Address Labels (Tract. Feed)	$9.95
15" Report Paper (Tract Feed)	$24.95
8½" Blnk Wht Paper (Tract. Feed)	$19.95

HEWLETT PACKARD

HP41CV... $209.00

HP 75............ $749.00

HP 41C	$146.00
HP 10C	$52.00
HP 11C	$69.00
HP 12C	$92.00
HP 15C	$92.00
HP 16C	$92.00

For HP41 41CV

HPIL Module	$99.00
HPIL Cassette or Printer	$359.00
Card Reader	$144.00
Extended Functions Module	$64.00
Time Module	$64.00

EPSON COMPUTERS

QX-10	CALL
HX-20	CALL

TI-40 COMPACT COMPUTER

$209

TIMEX SINCLAIR 1000 CALL

16K Memory	$29.95
2040 Printer	$99.95
Vu-Calc	$17.95
Mindware Printer	$99.00

SHARP

PC-1500A $169.00

Pocket Computer

PC-1250A........ $89.00

125 Printer/Micro Cassette	$129.00
150 Printer/Plotter/Cassette	$172.00
152 Cassette Recorder	$62.00
155 8K RAM	$94.00
158 8K RAM Battery	$129.00
161 16K RAM	$139.00

WEST

1-800-648-3311

In NV call (702)588-5654, Dept. 1206
P.O. Box 6689, Stateline, NV 89449
Order Status #: 588-5654

CANADA

1-800-268-4559

In Toronto call (416)828-0866, Dept. 1206
2505 Dunwin Ct., Unit 1B,
Mississauga, Ontario, Canada LSLITI
Order Status #: 828-0866

EAST

1-800-233-8950

In PA call (717)327-9575, Dept. 1206
477 E. Third St. Williamsport, PA 17701
Order Status #: 327-9576

MasterCard

No risk, no deposit on C.O.D. orders. Pre-paid orders receive free shipping within the UPS Continental United States with no waiting period for certified checks or money orders. Add 3% (minimum $5.00) shipping and handling on all C.O.D. and credit card orders. Larger shipments may require additional charges. NV and PA residents add sales tax. All items subject to availability and price change. We stock manufacturer's and third party software for most all computers on the market. Call today for our new catalog.

INSIGHT: Atari

Bill Wilkinson

This month I will discuss extended memory management on the Atari computers. Before I start, though, I would like just to chat for a bit. (If you are waiting for the last part of the series on self-relocatable code, be patient. It's just bigger than I expected it to be, so I've got to massage it a bit more.)

Some Small Talk About Computers

Today I read an interview with Alan Kay in *Technology Illustrated*. As many of you probably know, Alan Kay was perhaps the most instrumental person in the development of the Smalltalk language. (Or is it an operating system? Or is it more properly called simply an "environment"?)

The work he did on Smalltalk while at Xerox caused him to believe that computers were destined to become a household tool, as common as, say, the television set. (Which may seem a mundane belief today, but Kay was saying such things five to ten years ago.) Well, Atari apparently liked Kay's philosophy, vision, and capabilities, and hired him awhile back.

The article I read interested me in two ways. First, it labeled Kay "Atari's Chief of Games." Well, I had been led to believe that he had been brought to Atari to head research and development, presumably to lead Atari into the generation beyond Smalltalk (a logical presumption, since he'd stated that he felt Smalltalk had served its purpose, was obsolete, etc.).

Anyway, with my orientation toward languages and systems, I saw "Chief of Games" as a step downward. Yet the interview made it clear that Kay felt he was in perhaps one of the most challenging positions possible. Hmmm. What has changed? Are games truly the most useful purpose of a computer right now? The marketplace certainly seems to think so. It is food for thought.

The second thing in the article which really got my CPU stirred up was Kay's view of the computer. I had always been under the impression that he believed his real goal in life was to enable everyone not only to use the computer, but to actually command and manipulate it. (I hesitate to say "program it," but then Smalltalk *is* a language.) In the interview, though, Kay stated he was beginning to fear that perhaps the computer was not so much a household tool as it was a fine instrument, like a violin. He strengthened the analogy by noting that very few people can play the violin, just as very few people can properly use a computer.

Well, I for one believe that not only is the analogy inappropriate, but its projection of gloom and pessimism about the future of computers is not justified. Granted, the analogy may hold today. After all, only about 1 percent of the United States population can claim to be able to program at all (or play "Twinkle, Twinkle, Little Star" on the violin). Probably less than .1 percent produce acceptable application programs (or play in a community orchestra or equivalent). Dare we guess that .01 percent are commercial programmers (or make their living playing the violin)? Can it be that only .001 percent can actually write systems and languages (or are the guest soloists of the concert world)?

Actually, these proportions are just order-of-magnitude guesses, but they do seem to support Mr. Kay's analogy. But I say that his analogy has validity mainly because the computer is still such a relatively "rare" instrument. Personally, I prefer a different analogy.

When computers are as much a part of everyday life in this country as automobiles are now (and I firmly believe that they will be), then I think they will be treated much as automobiles are.

Let me sidetrack a little. Here in California, the State has decreed that all high school students shall take a course in "computer literacy." So what happens? Every high school is scrambling to buy one or two computers and begin teaching every kid how to program in BASIC. Great, right? *Nonsense!*

Two Different Classes

First of all, I can't conceive of learning how to use

Looks like a Ferrari. Drives like a Rolls. Parks like a Beetle.

Ask your computer dealer to take the cover off a world-class disk drive.

The all new, 1984 Indus GT.™

The most advanced, most handsome disk drive in the world.

A flick of its power switch can turn an Atari into a Ferrari.

Or an Apple into a Red Hot Apple.

Looks like a Ferrari.

The Indus GT is only 2.65" high. But under its front-loading front end is slimline engineering with a distinctive European-Gran flair.

Touch its LED-lit CommandPost™ function control AccuTouch™ buttons. Marvel at how responsive it makes every Atari or Apple home computer.

Drives like a Rolls.

Nestled into its soundproofed chassis is the quietest and most powerful disk drive power system money can buy. At top speed, it's virtually unhearable. Whisper quiet.

Flat out, the GT will drive your Atari track-to-track 0-39 in less than one second. Increasing data transfer 400%. (Faster than any other drive. And as fast as any Apple disk drive.)

And each GT comes with the exclusive *GT DrivingSystem*™ of software programs.* World-class word processing is a breeze with the *GT Estate WordProcessor*.™ And your dealer will describe the two additional programs that allow GT owners to accelerate their computer driving skills. *Included as standard equipment.

Also, the 1984 Indus GT is covered with the *GT PortaCase*.™ A stylish case that conveniently doubles as a 80-disk storage file.

Parks like a Beetle.

The GT's small, sleek, condensed size makes it easy to park.

And its low price makes it easy to buy. $449 for Atari. $329 for Apple.

So see and test drive the incredible new 1984 Indus GT at your nearest computer dealer soon.

The drive will be well worth it.

INDUS™

The all-new 1984 Indus GT Disk Drive.

The most advanced, most handsome disk drive in the world.

For dealer information, call 1-800-33-INDUS. In California, 1-800-54-INDUS, 213/882-9600.

or program a computer at all if the student/computer ratio is above 3 to 1. More importantly, I think it is senseless to equate "computer literacy" with "learning to program in BASIC." After all, "automobile literacy" consists of learning traffic laws, safe driving techniques, and actually starting to drive a car (it's usually called "Driver Training").

"Automobile expertise," on the other hand, consists of learning what tools do what, the theory and practice of internal combustion engines, and how to maintain and repair an automobile (and this is usually called "Auto Shop"). Does every student take driver training? Yes, or nearly so. Does every student take auto shop? No. Not by a long shot.

So, I believe, it should be with computer literacy. Don't teach everyone how to program. (What would we do with a nation of programmers? The same thing we would do with a nation of auto mechanics?) Instead, teach everyone how to use a computer to do word processing, to balance their budget, to access data bases, and the list could be quite long.

And, yes, keep the computer programming classes. But keep them on the same basis that auto shop classes are offered — as electives, for those interested in learning more than how to "drive" their computers or cars.

Why this confusion of computer literacy and computer expertise among schools and teachers? Partly because the computer industry has promoted the view. (Perhaps fearing that current applications programs are inadequate to a classroom situation?) Partly because of a dismal lack of education and information on the part of the educators. (Pity the poor math or history teacher who is nearing retirement. Suddenly he/she is forced to learn enough about these nasty machines to be able to teach some kids how to use it. Do you wonder that the path of least resistance is most often chosen?) Mostly, I suppose, because BASIC comes built into each machine, while good text processors, spreadsheet programs, etc., cost extra, money which most schools don't have.

So how does this tirade relate to either Alan Kay or you, my patient reader? Well, first of all, I think the analogy of car and computer is a better one than violin and computer. And, perhaps, if computer companies started trying to design mass consumable "cars" instead of trying to ply the public with precision instruments, it is a future that will come true. To be fair, I think that companies such as Atari and Commodore and Apple and others are starting to do so already. But my cynicism leads me to believe that they are driven by the current market, not by the future one.

You're Ahead Of Your Time

Perhaps more importantly, though, I am trying to convey the message that those of you who read this column (and this magazine) are, in some sense, ahead of your time. You are, indeed, the violinists that Alan Kay perceives. Some of you are just learning to play your first notes. Others of you are already tackling the great concertos. But, when the computer revolution really arrives, you will all have the advantage of having already taken at least your first "auto shop" course. So, if you enjoy your computer (and particularly if you enjoy programming), don't give it up easily. And certainly don't give it up now. Someday, others will appreciate your art, however humble or glorious it may be.

Did that sound like a sermon? If so, I apologize. But it's my view of both the present and the future of computers and programming. One last sidelight before we move on: On hearing me espouse the views above, someone once asked me what my position in the hierarchy was, as a person who helped design (as opposed to program) operating systems and first languages for new machines. Actually, that's an easy question: I'm simply a composer. And so, I think, are such people as Alan Kay.

You Can Bank On It

All of the new Atari XL computers (including the 1200XL) will contain 64K bytes of RAM (the 600XL requires an external RAM pack to do so). And all contain 16K bytes of Operating System ROM space. And, further, all (except the 1200XL) include good old Atari 8K BASIC. Let's see here — 64K plus 16K plus 8K — that's over 90,000 bytes of space.

Wait a minute, though. If I plug in a 16K cartridge (such as AtariWriter or ACTION! or BASIC XL), then I could have 104K bytes of RAM and ROM. Wow. That's really nifty, right? Well...

Have you read this column often enough to know that "Well..." means "not really" or "there's more to come"? No? Well...

Not really. To begin with, all Atari computers are built around the same CPU (Central Processing Unit), the 6502. (Which, incidentally, is the same chip used in most Commodore computers and all Apple machines except the Lisa.) However, there is a fundamental restriction involved when using a 6502: There is simply no way to access more than 64K bytes (65,536 bytes) at one time. How, then, can the Atari use 104K bytes? Is someone fibbing to us?

The key here is the phrase "at one time." A juggler may be able to juggle only four things at a time. Does that mean he always juggles the same four objects? Should we presume that the 6502 must always work with the same 64K bytes? Of course not.

In point of fact, the new XL machines allow the 6502 a number of choices about which bytes it will "juggle." How the 6502 makes its choice is

ATARI SOFTWARE FOR THE WHOLE FAMILY

Here are four software packages designed for the different people in your family.

A BASIC COMPILER FOR THE PRO-GRAMMER

ABC (A BASIC Compiler) automatically translates Atari BASIC programs into high-performance integer P-code that runs up to 12 times faster!

Perfect for developing system software and commercial games, ABC accepts most BASIC programs (unless floating point dependent) with little or no modification. Compiled P-code is a self-standing DOS object module that is unLISTable and runs without the BASIC cartridge.

ABC allows expressions in DIM, GOTO, GOSUB, and RESTORE statements, doesn't require you to re-order lines, and fully supports string and sub-string operations.

Give your BASIC programs the look and "feel" of professional products with ABC. 40K Disk ***$69.95***. Manual alone ***$9.95*** (credited toward compiler purchase).

MAKEBOOT lets you create self-booting disk or cassette versions of your ABC compiled software. Reduces overall program load time and saves memory and disk space by eliminating DOS. 40K Disk ***$14.95***.

AN EDUCATIONAL TOY FOR PRE-SCHOOLERS

Monarch is proud to present ***SofToy***, an educational program smart enough to act simple.

Bells ring, balls bounce, owls hoot as SofToy and its colorful interactive display gently introduce children (two years and older) to spatial relations, letters, numbers, even elementary programming! SofToy lets kids become familiar with computers, without arbitrary demands, competition, or intimidation. SofToy grows with children, too. At more difficult levels, the match game is a real challenge for the whole family. 24K Disk ***$29.95***.

TOOLS FOR THE SERIOUS USER

Power Tools I combines four sophisticated text processing tools on one easy-to-use utility disk.

DIFF shows you differences between two ATASCII text files: for example, changes you made in a program or document from one version to the next.

Manually searching for a particular text file can take hours. But now, with *SEARCH*, you specify a search string and a list of files. *SEARCH* examines the files and points out which ones have that string. You'll never lose your Fudge Brownie recipe again!

CHANGE is a powerful search and replace utility that operates on multiple files with one command. For example, you could change character names throughout your novel with a single command, even if each chapter is a separate file.

The special pattern-matching and multiple disk capabilities of *SEARCH* and *CHANGE* are an added plus.

TRANSLIT lets you swap one character set for another (for example, upper case for lower case) throughout a file with one command.

Power Tools I is ideal for professional business and software development text applications. 40K Disk ***$34.95***.

Look for these fine products at your nearest ATARI dealer, or order direct. Include $2.00 for shipping and handling in U.S./Canada. Checks or money orders accepted. MasterCard/Visa add 3%. COD higher. MA residents add 5% tax. Dealer and distributor inquiries invited. ATARI is a trademark of Atari, Inc. ABC, Makeboot, PowerTools, and SofToy are trademarks of Monarch Data Systems, Inc.

MONARCH DATA SYSTEMS

MONARCH DATA SYSTEMS, INC., P.O. Box 207, Cochituate, MA 01778 (617) 877-3457 the subject of this section.

Actually, there is no magic formula or scheme which enables the various choices. In fact, various choices are made by differing means. Generally, the choice is "consciously" made by the program currently in control of the machine. And it makes the choice simply by (usually) storing something in a particular memory location. Confused? Let's digress a little.

Some CPUs (including microcomputers and minis and maxis) treat input/output as a separate domain from general memory. For example, the 8080/Z-80 group of processors allow up to 256 separate input and output ports, which are completely separated from the general RAM/ROM memory (they even have special instructions specifically for reading/writing these I/O ports). On the other hand, many machines (such as the 6800, 68000, and 6502 families, as well as such giants as the PDP-11 series) simply treat input/output ports as part of the general machine memory.

Efficient And Easily Learned

The advantages and disadvantages of each scheme are a subject of hot debate, but I will only present a single aspect of each here: Keeping the I/O ports out of general memory allows a true 64K bytes of RAM when using an 8- or 16-bit microprocessor. Allowing I/O to be treated as part of memory means that any instruction which can access RAM or ROM can also access a port, often resulting in efficient and easy-to-learn coding.

Anyway, note that the 6502 does, indeed, use what is called "memory mapped I/O," and Atari computers do, as a consequence, reserve 2K bytes of memory (addressed from $D000 to $D7FF) which is specifically designed for I/O port addresses. (If losing 2K of your space seems excessive, pity the Apple owner who loses 4K.)

In the case of the XL machines, then, one simply changes the value in an I/O port — which appears to one's program as a memory address — and presto, a different choice of "jugglable" memory is made. But what I/O port to use? Did you notice the fact that Atari 400 and 800 computers have four joystick ports while the XL machines have only two? Guess which ports are now used for memory juggling. Did you need more than one guess?

For the more hardware-oriented of you out there, I will note that all four Atari joystick ports are actually nibble-sized pieces of a 6820 (or 6520) PIA (Peripheral Interface Adapter). The PIA is a very flexible chip; it allows each of its 16 I/O pins to be separately configured to be either an Input line or an Output line. In the case of the 400 and 800, all 16 lines are configured as Input, since they are all used to read the four directional switches of an Atari joystick. In the case of the XL machines, some of them have been changed to Output lines, thus enabling them to act as electronic switches.

On the 1200XL, for example, two of them are used to control the L1 and L2 status LEDs. And (you saw this coming, I presume) two of them choose certain configurations of the computer's memory. (On the other XL machines, still another line is used to control still another possible configuration.)

Since we are discussing memory configuration choices, I might as well confuse the issue a bit more by also mentioning how we at OSS implemented our new SuperCartridges. It is probably no accident that Atari provides the cartridge slot on all machines with a line labeled "CARCTL", an abbreviation for CARtridge ConTroL. Actually, this line is active whenever any memory location from $D500 to $D5FF is accessed. Since no Atari cartridges take advantage of this line, we thought it was time that we did so.

One At A Time

About now, it is past time for a diagram. The figure shows all the possible choices of memory configuration by placing them in memory address order. Note, though, that the 64K addressing restriction of the 6502 applies. Hence, when two or more choices are given for a particular address range in memory, remember that only one such choice may be active at any given time. For each address range where a choice is available, there are two or more *banks* of memory. And choosing one bank over another is called *bank switching* or *bank selection.*

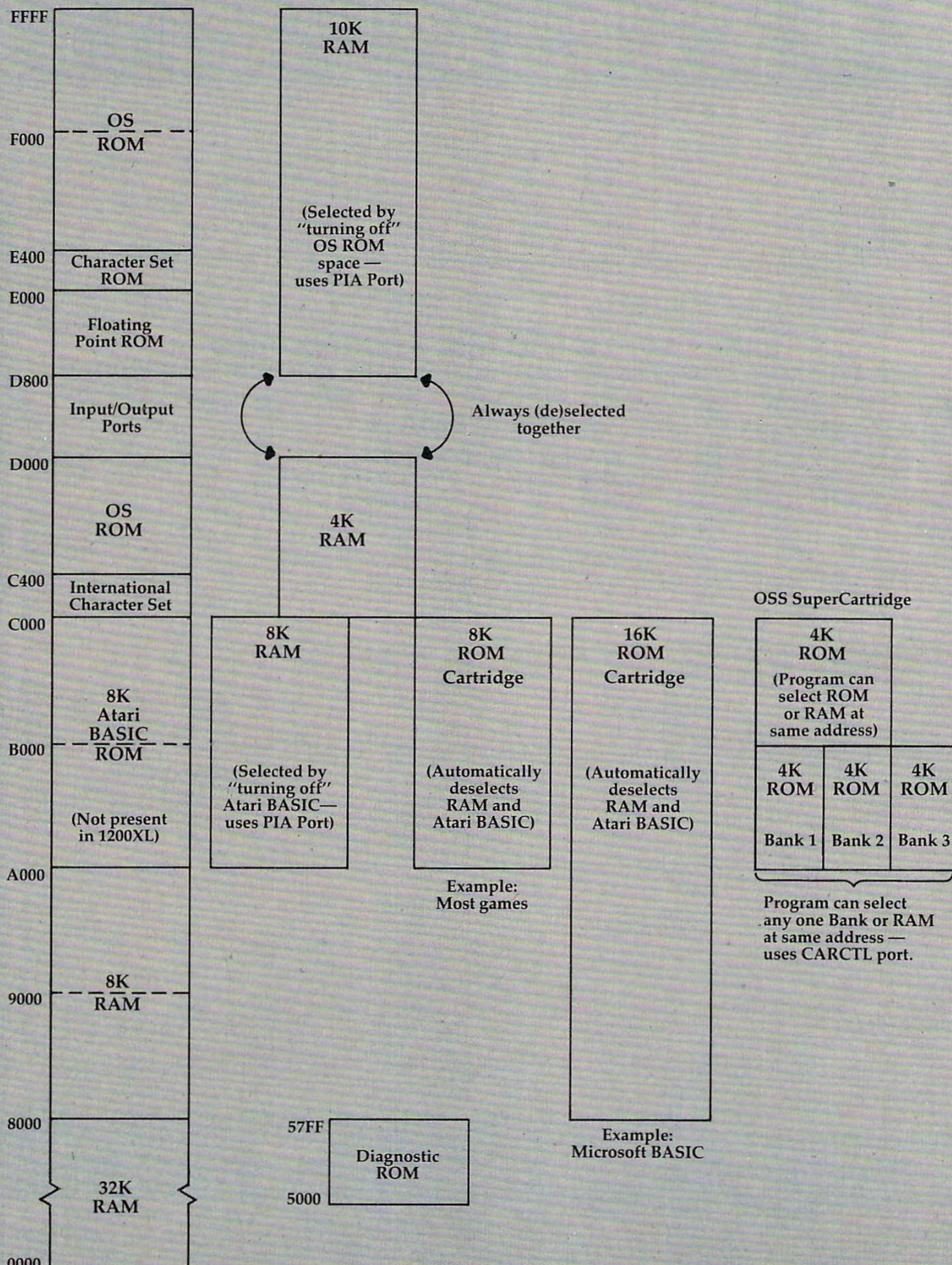
For example, I might choose to use BANK1 of the SuperCartridge while at the same time choosing the RAM BANK of system memory. The important thing to note here is that each set of banks (that is, parallel memory segments), as shown in the figure, is independently bank selectable.

Also, some bank choices are not available at the software level. For example, when you plug in a Microsoft BASIC cartridge, you have 16K bytes of ROM from $8000 to $BFFF. You have no RAM in that address range. You have no choice in the matter. This is, then, hardware bank selection.

The advantage of hardware bank selection is that it is essentially foolproof. If the hardware removes a bank of RAM from your program's "vision," your program can't get into trouble trying to use that bank.

But the advantage of software-selectable banks is, quite simply, that they allow you to expand the capabilities of your machine. If you look at the figure, you can see that a SuperCartridge allows you 16K bytes of programming power while occupying only two 4K byte banks at any given time.

Memory Map Of Atari XL Computers (Showing Parallel Memory Banks At Same Addresses)



And the purpose of this discussion? To show that the XL machines really do have a lot of latent power. How do we make it un-latent? Well....

As I write this article, the number of commercially available programs which allow you to take advantage of the extra 14K bytes of RAM on an XL machine is countable on the fingers of my left foot. Zero. By the time you read this, there will likely be products heading your way that will justify the purchase of an XL machine (or a 64K memory board, such as the one from Mosaic Electronics, for your 800).

Since I am obviously most familiar with DOS XL, let me explain a little of how it works.

When DOS XL boots into an XL computer, it first establishes a set of jump vectors for the various interrupt routines. Why? Because any IRQ, NMI, or SYSTEM RESET will attempt to jump through the vectors which must (by 6502 CPU law) be located at addresses $FFFA through $FFFF. If we deselect the OS ROM bank in order to enable the RAM bank at the same addresses, the contents of these critical addresses are unpredictable. We *must* supply some valid routine addresses or the system will crash.

DOS XL puts most of the DOS code in the RAM bank which is "under" the OS ROMs. It also leaves a piece of itself at the conventional DOS load address of $700 (an area of memory which is not bank selectable). Then, if there is a BASIC cartridge in the machine, it selects the OS ROM bank and jumps to BASIC.

So long as BASIC makes no calls on DOS, all is calm and expected. However, watch what happens when (for example) we try to open a file from BASIC.

1. BASIC sets up an IOCB with a pointer to the filename. Since the filename was specified by the user, the pointer will contain an address somewhere between about $A00 and $9C00. BASIC makes a call to $E456, the CIO entry point.

2. CIO determines that the device requested is actually the disk file manager and uses the "D:" device table to determine the address of the disk's open file routine. It passes control to that routine.

3. Note that the "D:" device table and at least the first part of the file open routine must be in nonselectable RAM (that is, at or near $700). The file open routine is a big one, so it selects the DOS XL RAM (disabling the OS ROM) and jumps to the main part of the code.

4. The main code is able to examine the filename since it is in nonselectable memory, so the file open is performed if possible. The main code exits back to the tail end of the OPEN code, near $700.

5. This tail end then simply reselects the ROM bank and returns to where it was called (somewhere in CIO).

6. When CIO is finished, it returns control to BASIC.

Wasn't that fun? For even more fun, try to trace what happens if interrupts occur during any or all of the above steps.

More Space

But why do we go through all this? Because, even though Atari saw fit to include all this good memory bank selection capability, they provided no software to use it. So why not just forget the bank select and pretend we are running on an Atari 800 or 400? Because the net gain to you, the BASIC or ACTION! or Assembler or whatever user, is about 5,000 bytes of user space. Your programs can be 5K bytes bigger. Your spreadsheets can contain many more cells. You can edit more text.

Of course, some programs (such as VisiCalc) which do not use a standard DOS or which use a heavily protected disk (such as the Microsoft BASIC extensions) will not be able to take advantage of the extra memory. But they, too, can use these techniques to extend their capabilities if the software companies producing them will decide that the XL machines are worth the little extra effort. ©

Computer Outlet

We offer the largest selection of software and hardware for Apple, Atari, Commodore, IBM and Kaypro at 25 to 40% off retail.

1095 East Twain, Las Vegas, NV 89109 • Mon.-Fri. 8AM to 6PM, Sat. 9AM to 5PM

Commodore 64 $219

Avalon Hill Game Company
B-1 Nuclear Bomber (C) .... $ 12
Midway Campaign (C) .... $ 12
North Atlantic Convoy Raider (C) .... $ 12
Nukewar (C) .... $ 12
Planet Miners (C) .... $ 12
Computer Stocks & Bonds (C) .... $ 15
Andromeda Conquest (C) .... $ 14
Computer Football Strategy (C) .... $ 12
Telengard (C) $16

Broderbund
David's Midnight Magic .... $ 23
Choplifter (CT) .... $ 34
Serpentine (CT) .... $ 27
Sea Fox (CT) .... $ 27
Bank Street Writer .... $ 45

Datamost
Roundabout .... $ 20
Mating Zone .... $ 20
Paint Wizard .... $ 33

EPYX/Automated Simulations
Jump Man (D) .... $ 27

Human Engineered Software (HES)
6502 Professional Dev. System (C) .... $ 20
Retro Ball (CRT) .... $ 27
Hesmon (CRT) .... $ 27
Turtle Graphics II (CRT) .... $ 45
Heswriter 64 (CRT) .... $ 35
Gridrunner (CRT) .... $ 23

Infocom
Zork I, II, III .... Ea. $ 27
Deadline (D) .... $ 34
Starcross (D) .... $ 27
Witness .... $ 34

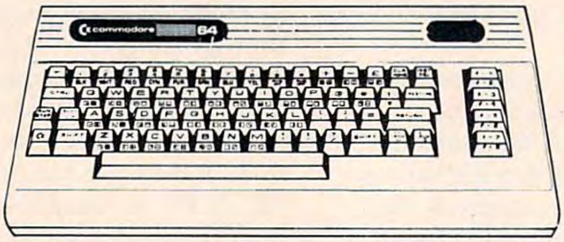
Sierra On-Line
Frogger (D) .... $ 25
Crossfire .... $ 15
Jaw Breaker .... $ 20
Threshold (CRT) .... $ 27
Sammy Light Foot (CRT) .... $ 25

Sirius Software
Blade of Blackpoole (D) .... $ 27
Type Attack (CRT) .... $ 27
Repton (D) .... $ 27
Critical Mass (D) .... $ 27
Snake Byte (D) .... $ 23
Bandits .... $ 27
Squish 'em (CRT) .... $ 23
Final Orbit (CRT) .... $ 23

Spinnaker
Snooper Troops #1 (D) .... $ 30
Facemaker (D) .... $ 23
Kindercomp (D) .... $ 20
Hey Diddle Diddle .... $ 20
In Search of the Most Amazing Thing .... $ 27
Fraction Fever (CRT) .... $ 20
Alphabet Zoo (CRT) .... $ 20
Delta Drawing (CRT) .... $ 20

Synapse Software
Ft. Apocalypse .... (D) $ 23, (C) $ 23
Dreibs .... (D) $ 23, (C) $ 23
Survivor .... (D) $ 23, (C) $ 23
Pharoah's Curse .... (D) $ 23, (C) $ 23

United Microwave Industries
92E-302 Renaissance (C) .... $ 20
92E-331 Motor Mania (C) .... $ 20



1525 Printer $229
1530 Datasette .... $ 64
1541 Disk Drive $249
1600 Modem .... $ 89
1701 Commodore Monitor $289
VIC 1311 Joystick .... $ 8
VIC 1312 Game Paddles .... $ 16

VIC 1210 3K Memory Expander $34
VIC 1110 8K Memory Expander $52
VIC 1111 16K Memory Expander $89
VIC 1011 RS 232 Terminal Interface $43
VIC 1211 Super Expander $59
VT 106A Recreation Pack .... $ 45
VT 107A Home Calculation Pack .... $ 45
VIC 1600 Vicmodem .... $ 89
VM Programmer's Reference Guide .... $ 14
Commodore Programmer's Ref. Guide $ 18

Educational

Books
Kids and the VIC .... $ 18
Programmer's Reference Guide-VIC .... $ 14
Programmer's Reference Guide-64 .... $ 18

Reading & Language Arts
Super Hangman (C) .... $ 14
Simon/Hess (C) .... $ 13
Concentration (C) .... $ 13
Home Babysitter .... $ 23
Word Search .... $ 15
Facemaker .... 64-$ 23 VIC-$ 27
Kindercomp/64 .... $ 20
Snooper Troops I/64 .... $ 27

Mathematics
Sky Math (C) .... $ 12
Space Division .... $ 12
Bingo Speed Math (CT) .... $ 23
Number Crunch (CT) .... $ 23

Number Chaser .... $ 17
Number Gulper .... $ 17

Music
Fun with Music .... $ 27
VIC Music Composer (CT) .... $ 42
HES Synthesound (CT) .... $ 49

Programming Techniques
Intro to Basic Prog. I .... $ 22
Intro to Basic Prog. II .... $ 22
Programmer's Aid Cart .... $ 22
Turtle Graphics/Hess (CT) .... $ 29
Gorteck & The Microchip .... $ 23
Kids on Keys/64 .... $ 23

Social Studies/Science
Visible Solar System .... $ 23
Reagonomics (CT) .... $ 27

Commodore Software
Avenger .... $ 23
Super Alien .... $ 23
Jupiter Lander .... $ 23
Draw Poker .... $ 23
Midnight Drive .... $ 23
Radar Rat Race .... $ 23
Raid on Fort Knox .... $ 23
Sargon II Chess .... $ 29
Cosmic Cruncher .... $ 23
Gorf .... $ 29
Omega Race .... $ 29
Sea Wolf .... $ 23
Adventureland .... $ 29
Pirate Cove .... $ 29
Mission Impossible .... $ 29
The Count .... $ 29
Voodoo Castle .... $ 29
The Sky is Falling .... $ 23
Mole Attack .... $ 23
Bingo Speed Math .... $ 23
Home Babysitter .... $ 23
Visible Solar System .... $ 23
Personal Finance .... $ 29
Quick Brown Fox .... $ 56

Stocking Stuffers
Koala Touch Tablet .... $ 69
Wico "BOSS" Joystick .... $ 15
Elephant Disks s/s .... $ 18
Flip 'n File Diskette Box .... $ 21

Super Holiday Special
Commodore VIC 20 Datasette Recorder Gorteck and the Microchips
$169

Avalon Hill
Tank Arcade .... $ 12
Nuke War .... $ 12

Automated Simulations
Rescue at Rigel (C) .... $ 20
Ricochet (C) .... $ 20
Sword of Fargoal .... $ 20
Temple of Apshai .... $ 27

Broderbund
Martian Raider $15
Multisound Synthesizer $15
Shark Trap $15
Sky Blazer (CT) $27
Sea Fox (CT) $27
A.E. (CT) $27

Creative Software
Black Hole (CRT) .... $ 36
Trashman (CRT) .... $ 36
Astroblitz (CRT) .... $ 36
City Bomber & Minefield (CRT) .... $ 20
Apple Panic (CRT) .... $ 36
Serpentine (CRT) .... $ 36
Choplifter (CRT) .... $ 36
Terraguard (CRT) .... $ 36
Household Finance .... $ 36

HES Software
HES Mon (CT) .... $ 29
HES Writer (CT) .... $ 29
Synthesound Music Synthesizer (CT) .... $ 20
Turtle Graphics (CT) .... $ 27
VIC Forth (CT) .... $ 45
Victrek (C) .... $ 15
Predator (CT) .... $ 27

Sirius
Type Attack $27
Snake Byte $27

Thorn EMI
River Rescue (CT) .... $ 27
Mutant Herd (CT) .... $ 27

Tronix
Galactic Blitz (C) $17
Swarm (C) $20
Sidewinder (C) $20
Gold Fever (CT) $27
Deadly Skies (CT) $27

United Microware
Spiders of Mars (CT) .... $ 34
Meteor Run (CT) .... $ 34
Amok .... (C) $ 15 (CT) $ 27
Alien Blitz .... (C) $ 17 (CT) $ 27
Skymath (C) .... $ 12
Space Division (C) .... $ 12
Super Hangman (C) .... $ 14
The Alien (C) .... $ 17
3D Maze (C) .... $ 12
Kosmic Kamikaze (C) .... $ 17
Sub Chase (C) .... $ 15
Renaissance .... $ 20
Cloud Burst (CT) .... $ 27
Satellites & Meteorites (CT) .... $ 34
Outworld (CT) .... $ 34
Wordcraft .... $ 65

Bitmap Graphics On The 64

Michael Tinglof

High-resolution graphics are achieved by bitmapping. Here's a tutorial and an explanation of what happens in the 64's memory as you bitmap. Also included is a sample program which illustrates the techniques discussed.

High-resolution images of 320 by 200 point (called *pixel*) resolution are possible on the 64. To create these images, the 64's VIC-II video chip uses a technique called bitmapping. Simply defined, this means that every bit in a selected area of memory represents one pixel (the smallest point of light) on the high-resolution screen. Thus, by setting or clearing appropriate bits, a picture can be formed.

You might ask "Why use bitmapped graphics when sprites are available and far more convenient to use?" The answer is simple: Each graphics mode has its own purpose. Several of the main reasons for using bitmapped graphics are to create graphs of formulas or statistics, to create high-resolution color pictures, and to create a detailed background for use with sprites, such as for a game.

Binary Operations

Before the bitmapped mode can be used effectively, it is important to have a basic understanding of binary arithmetic (see the section "Binary And Bitmapping" accompanying this article) and the logical AND and OR commands. Basically, they are used to selectively set and clear one or more bits in a byte. AND and OR cause a bit-by-bit comparison of two bytes to produce a third byte. In the case of AND, if both bits are on (1), the resulting bit is on; and in the case of OR, if *either* bit, or both, is on, the resulting bit, likewise, is on. For example:

```
    10101011        00110001
AND 11011011     OR 10101010
  = 10001011      = 10111011
```

The bits in a byte are usually numbered as follows:

```
7 6 5 4 3 2 1 0
```

AND is used to selectively clear bits, and OR is used to set bits. For example:

Given: 10100101, clear bit 5. To do this, define a byte with bit 5 set (0010000), then take the inverse (properly termed "complement") of the byte by changing all 1's to 0's and vice versa. Finally, AND the calculated byte with the given byte:

```
    10100101  (given)
AND 11011111  (calculated)
    10000101
```

Given: 10011010, set bit 6. To do this, define a byte with bit 6 set. Then OR this byte with the given byte:

```
    10011010  (given)
OR  01000000  (calculated)
    11011010
```

Remember that when BASIC is used, all binary bytes must be converted to decimal first. BASIC's AND or OR instructions will then work as described above.

Setting Up The VIC-II Chip

With an understanding of ANDs and ORs, a high-resolution picture can be created. The first step is to select an area of memory 8,000 bytes in length for the bitmap.

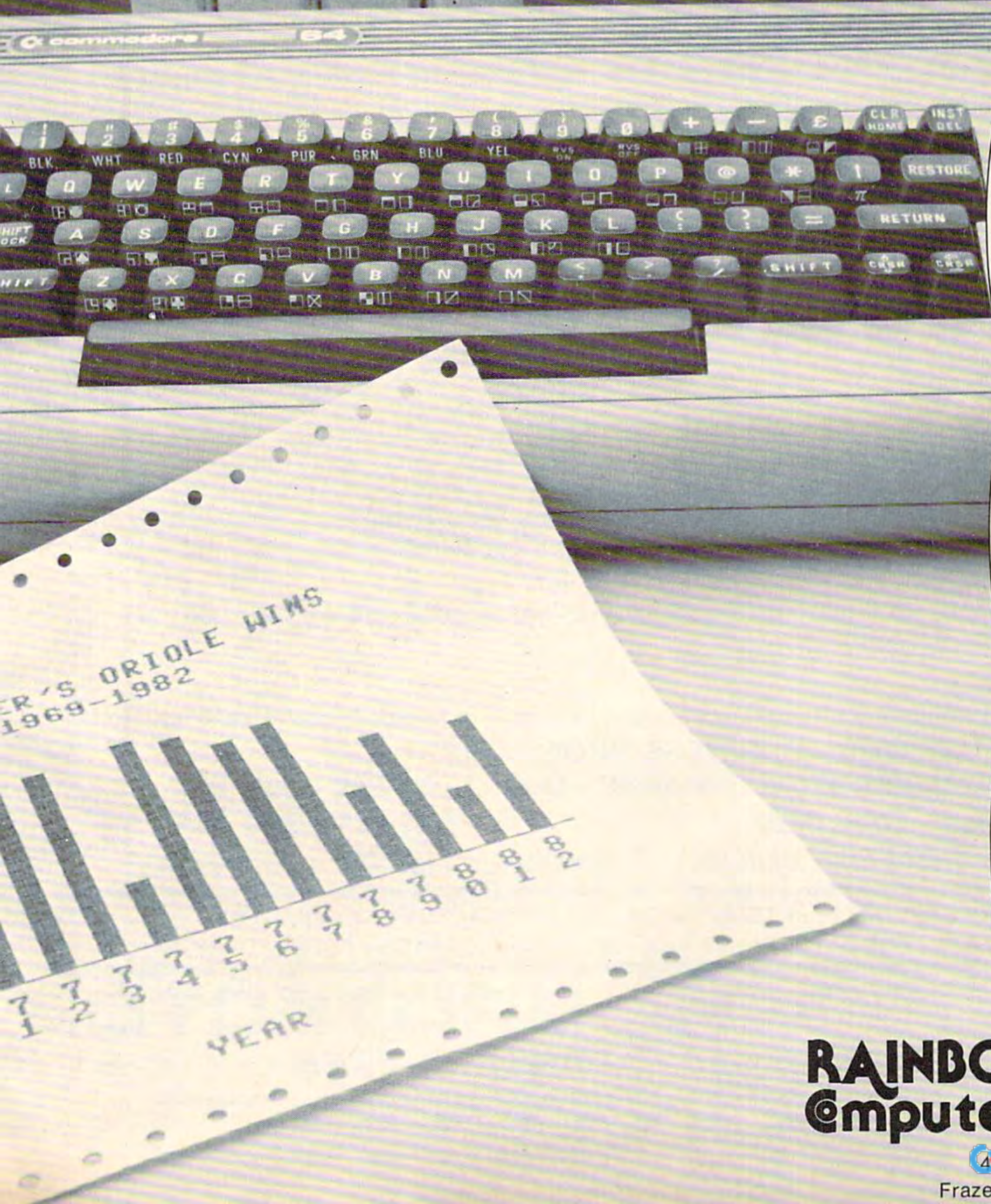
The VIC-II chip accesses only one 16K block of memory at a time. Upon power-up, the VIC-II sees the first 16K from locations 0 to 16383. All video operations, including those for screen mem-

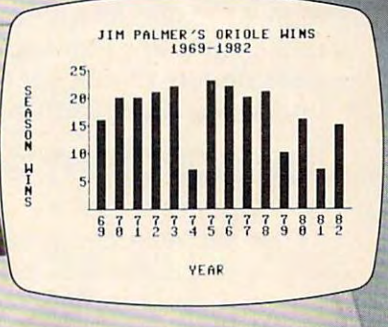
The Most Practical Software — Now Has Graphics

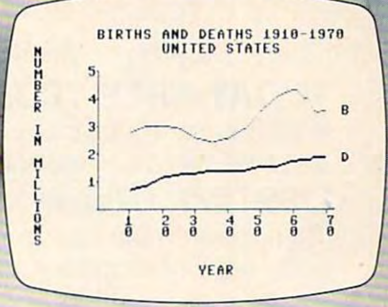
The **Graphics Assistant,** the latest addition to the ASSISTANT SERIES, lets you and your 64 produce charts and graphs in three formats. You can display them on screen or print them out. On screen display is 30 columns by 14 rows — about 60% of the screen. Print-out can be two sizes: a compact 4" x 4" or a full page, 7" x 9", display.

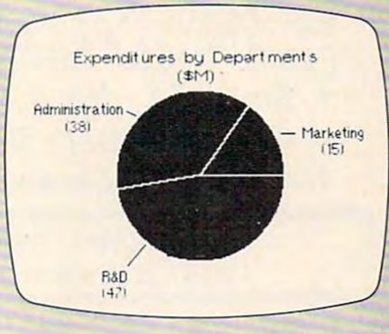
Bar chart format accepts up to 30 bars per chart; line chart allows 200 points per chart; pie chart can be sliced as thin as you desire. Vertical and horizontal labels are clearly displayed. On the pie chart a label with pointer is displayed outside the graph and indicates percentage or raw numeric data, i.e. Rainbow (73) or Graphics (141). You can assign range, limits, and values to create charts. Most importantly, however, you can retrieve data from files created by the **Spreadsheet Assistant.**
The ASSISTANT SERIES is now better than ever! You can now attach graphs to documents created by the **Writer's Assistant.** And produce comparison charts from data that has been calculated and replicated on the **Spreadsheet Assistant.**

The Graphics Assistant **$79.95**









RAINBOW Computer Corporation

490 Lancaster Pike
Frazer, Pennsylvania 19355

Commodore - 64
PROFESSIONAL BUSINESS PROGRAMS

- **INTRODUCTORY OFFER ½ PRICE!**
- **15 DAY FREE TRIAL!**
- **LIFETIME GUARANTEE!**
- **FILE GUARD (TM)**

PROFESSIONAL BUSINESS PROGRAMS! (*Disk*)

You take no risk! We are so sure these professional business programs will meet and exceed your highest expectations, we are willing to allow you to try these programs at our expense! These are designed and produced by **Southern Solutions** who produces professional business programs for **Commodores** most expensive computers. These business program will convert your **Commodore 64** into a Professional Business Machine!!!

INTRODUCTORY OFFER! (Expires 12-25-83)

	List	½ Price
• General Ledger & Cash Flow	$119.00	$59.00
• Accounts Payable Plus Check Writing	$119.00	$59.00
• Accounts Receivable	$119.00	$59.00
• Payroll	$119.00	$59.00
• Inventory	$119.00	$59.00
• Data Base Manager	$119.00	$59.00

15 DAY FREE TRIAL!

We give you 15 days at your business for you to try out these programs! Should they not meet your requirements just send them back prepaid and we'll refund your purchase price!

LIFETIME GUARANTEE!

If a program fails due to faulty workmanship or material anytime you personally own and use the program we will replace it at no charge!

FILE GUARD (TM)

Prevents loss of data and confidential files due to power failure — a Southern Solutions exclusive!

PLUS: THESE PROFESSIONAL BUSINESS PROGRAMS -

- ***Script 64 - No. 1 Executive Word Processor - Disk*** — ***List $99 - Sale $59***
- ***Complete Data Base - Tape-Disk*** — ***List $89 - Sale $59***
- ***Electronic Spread Sheet (like Visicalc) - Tape-Disk*** — ***List $89 - Sale $59***

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

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

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

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

ory and sprite definitions, access the memory in this area. There is no room in this block for an 8K bitmap, however, without conflicting with BASIC. The best solution is to select a different 16K block. (Bits 1 and 0 of address 56576 control where the block is placed in memory.) The combinations of these two bits and the range of addresses they represent are as follows:

decimal	binary	address
0	00	49152–65535
1	01	32768–49151
2	10	16384–32767
3	11	0–16383

Note that each block starts at an even 16K boundary. To select a memory block for the VIC-II chip, use the following command:

```
POKE 56576, Y
```

where Y is one of the decimal values from the above table. The best block to choose when using a bitmap and BASIC is number 2:

```
POKE 56576, 2
```

Within this block, two more areas must be selected: one for the 8K bitmap and one for the 1K screen memory. Address 53272 is used to control these two memory regions. One bit in this byte controls which 8K section in the 16K block is used for the bitmap; four bits control which 1024-byte area is used as the screen memory; and three bits are not used. The bits are arranged in address 53272 as follows:

7 6 5 4	3	2 1 0
screen memory	bit-map	x x x not used

The areas selected must fall on even boundaries — that is, their starting address must be a multiple of their size. For example, if the 16K block selected is from 0 to 16383, the screen memory can fall on 0, 1024, 2048, 4096, and so on. The following table can be used to determine which block should be used for screen memory or the bitmap:

Base plus	screen memory block	bitmap block
0	0	0
1024	1	
2048	2	
3072	3	
4096	4	
5120	5	
6144	6	
7168	7	
8192	8	1
9216	9	
10240	10	
11264	11	
12288	12	
13312	13	
14336	14	
15360	15	

where Base is the first address in the selected 16K block. To set 53272, use the following formula:

```
POKE 53272,   screen memory block * 16 + bitmap
  block * 8
```

If you are using the bitmap and BASIC at the same time, use the following POKE:

```
POKE 53272,120
```

This sets the screen memory block to seven, and the bitmap block to eight. For the 16K block suggested for use with BASIC, this means that screen memory starts at 23552 and the bitmap starts at 24576.

Once the memory pointers have been set, the VIC-II chip must be told to display the bitmap on the screen. Bit 5 of 53265 turns on the bitmap mode, that is, displays bitmap memory. To set this bit, use the following POKE command:

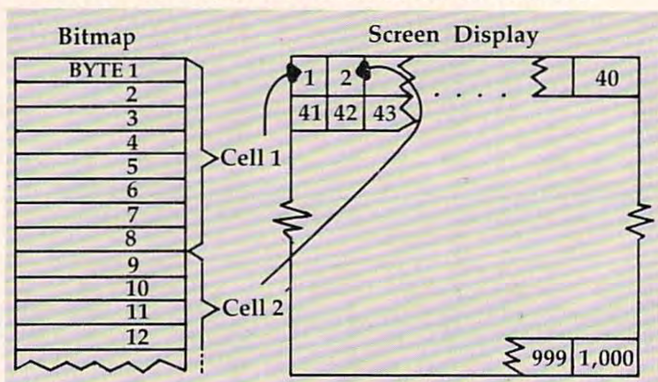
```
POKE 53265, PEEK(53265) OR 2↑5
```

Drawing The Picture

A high-resolution picture can now be created — all you have to do is set and/or clear the appropriate bits in bitmap memory. The problem is determining which bit controls which pixel. This requires an understanding of how the VIC-II chip draws the bitmap on the screen.

The bitmap memory is constructed similar to screen memory in text mode — it is broken into 1000 areas, each eight bytes in size, which we'll call *cells*.

These cells are arranged contiguously in memory — cell 1 follows cell 0, cell 2 follows cell 1, and so on. They are arranged in the bitmap in an order similar to that of screen memory in the text mode, 40 cells per row, 25 rows. The whole process, as described so far, can be illustrated as follows:



Each cell controls an area of 64 pixels arranged in an 8 by 8 matrix. The first byte in the cell controls the top row of pixels in that matrix, the second byte controls the row beneath, and so on down.

The eight bits in each byte control one pixel in that row — the highest valued bit controls the leftmost pixel and so on through the lowest valued bit, which controls the rightmost pixel. Graphically, the process works as follows:

THE CHALLENGE IS YOUR CHOICE

YOU CAN DO THIS

Commodore 64: Getting The Most From It

BY TIM ONOSKO

The Commodore 64 is now yours to master in this new, unique book for beginning users. You'll find this book full of suggestions, hints, and background information! Plus it's:

- Applicable to all versions of the Commodore 64!
- Simply and clearly written for first-time computer users
- The perfect companion for learning the essential skills in BASIC programming, color graphics, sound, word processing, and games.

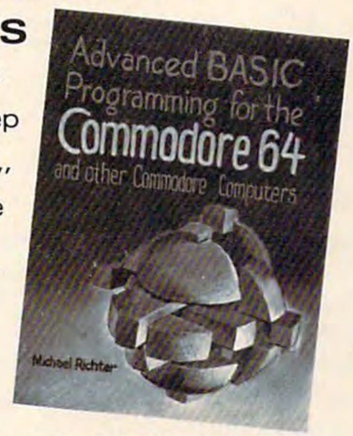
It's a necessary tool for your Commodore 64!

1983/320pp/paper/
ISBN 0-89303-380-4/D3804-4/$14.95

OR THIS

Advanced BASIC Programming for the Commodore 64 and Other Commodore Computers

BY MICHAEL RICHTER



This is the next step for the user who knows the "basics" and wants to move on to more advanced BASIC programming. Here you'll learn:

- How to read, write, and use good programs
- How to gain knowledge through the experience of writing advanced software
- Applications for both personal and professional use
- Numerous examples to enhance each concept

This is the way to maximize the capabilities of the Commodore!

1983/204pp/paper/
ISBN 0-89303-302-2/D3022-3/$14.95

OR BOTH

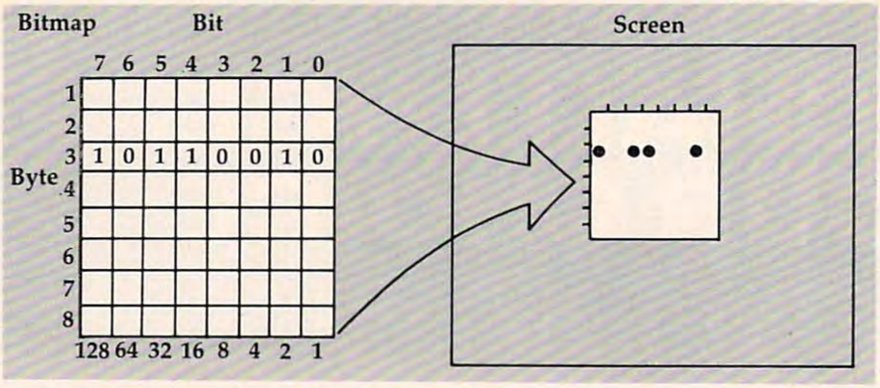
Chances are when you finish **Commodore 64: Getting The Most From It** you'll want to take the next step into **Advanced BASIC for the Commodore 64 and Other Commodore Computers.** One way or another, you're assured a thorough understanding of the powerful Commodore 64.

Available at your local bookstore or computer store. Or call Toll-Free 1-800-638-0220.

BRADY BOOKWARE™

Robert J. Brady Co., Bowie, Maryland 20715

A Prentice-Hall Publishing and Communications Company



Using X and Y coordinates is cumbersome with this system. If this type of plotting is needed, the following equations will determine which bit to set for the X,Y coordinate:

Y1 = INT(Y/8)*8 determines which row of cells
X1 = INT(X/8)*8 determines which cell on the above row
AD = Y1*320 + X1 + Y − Y1 + start of bitmap memory determines address of proper byte
BT = 7 − X1 determines which bit to set
POKE AD, PEEK (AD) OR 2 ↑ BT sets the bit

If you have been following our example setup commands, use a starting address for the bitmap of 24576.

Adding Color

Color is an important part of high-resolution graphics. Each of the 1000 bytes in screen memory controls the color displayed for one cell. Note that screen memory controls the color *only* in bitmap mode — in normal text mode, it contains the characters displayed on the screen. The bytes in screen memory are in the same order as the cells in the bitmap (the color of cell 650 is controlled by byte 650 in screen memory). In each byte, four bits are used to control the color of each bit in the corresponding cell of the bitmap, and four bits are used to control the color of bits equal to zero. These bits are arranged in each byte of screen memory as follows:

7 6 5 4	3 2 1 0
color of bits = 1	color of bits = 0

The colors and their corresponding values are listed on page 159 of the *User's Guide*. Once the values for the desired colors have been found, use the following formula:

(color of bits = 1)*16 + (color of bits = 0)

POKE this value into the appropriate byte of screen memory. Remember that attempting to change the color of one pixel will change the colors of all pixels in that cell of bitmap memory.

Note that screen memory for our working example begins at address 23552.

Recall that this method can be used to create a picture with 320 by 200 pixel resolution. Another bitmap mode is available, however. This second mode allows four colors in each cell rather than two colors as demonstrated above. There is one catch: resolution is reduced to 160 by 200 pixels, and each pixel is twice as wide. The multicolor mode is enabled by turning on bit 4 of location 53270. Use this command to enable multicolor mode:

```
POKE 53270, PEEK(53270) OR 2 ↑ 4
```

Each pixel is now represented by two bits. These two bits have four possible combinations, resulting in four possible colors. To find the color each bit combination represents, several memory locations and/or areas are accessed: screen memory, color memory (this is always from 55296 to 55319), and the background color register at 53281. Color memory is arranged in the same order as screen memory. The following chart shows which bit combinations access which areas of memory:

Bit Combination	color from
00	background register (53281)
01	screen memory (4 bits of greatest value; same as bit equal to one in two-color mode)
10	screen memory (4 bits of least value; as bit equal to 0)
11	color memory

Remember that three of the four colors selected can be different for each cell in the bitmap. The method used to draw the bitmap on the screen in two-color mode is used in the multicolor mode — only now, the bits are grouped together into pairs. The pairs are formed sequentially, so that bit 7 and bit 6 are paired, bit 5 and bit 4 are paired, and so on.

Protecting Your Picture

When using BASIC and the bitmap modes together, BASIC may have a tendency to use the bitmap memory for program and/or variable storage. To prevent this, change addresses 55 and 56, the bytes which point to BASIC's end of memory. Simply change these to point to an address below the lowest address you use. Address 56 is equal to the last address used divided by 256, and address 55 is the remainder. After changing these two bytes, execute a CLR instruction. For example, this instruction insures that BASIC will not use any memory after address 23552:

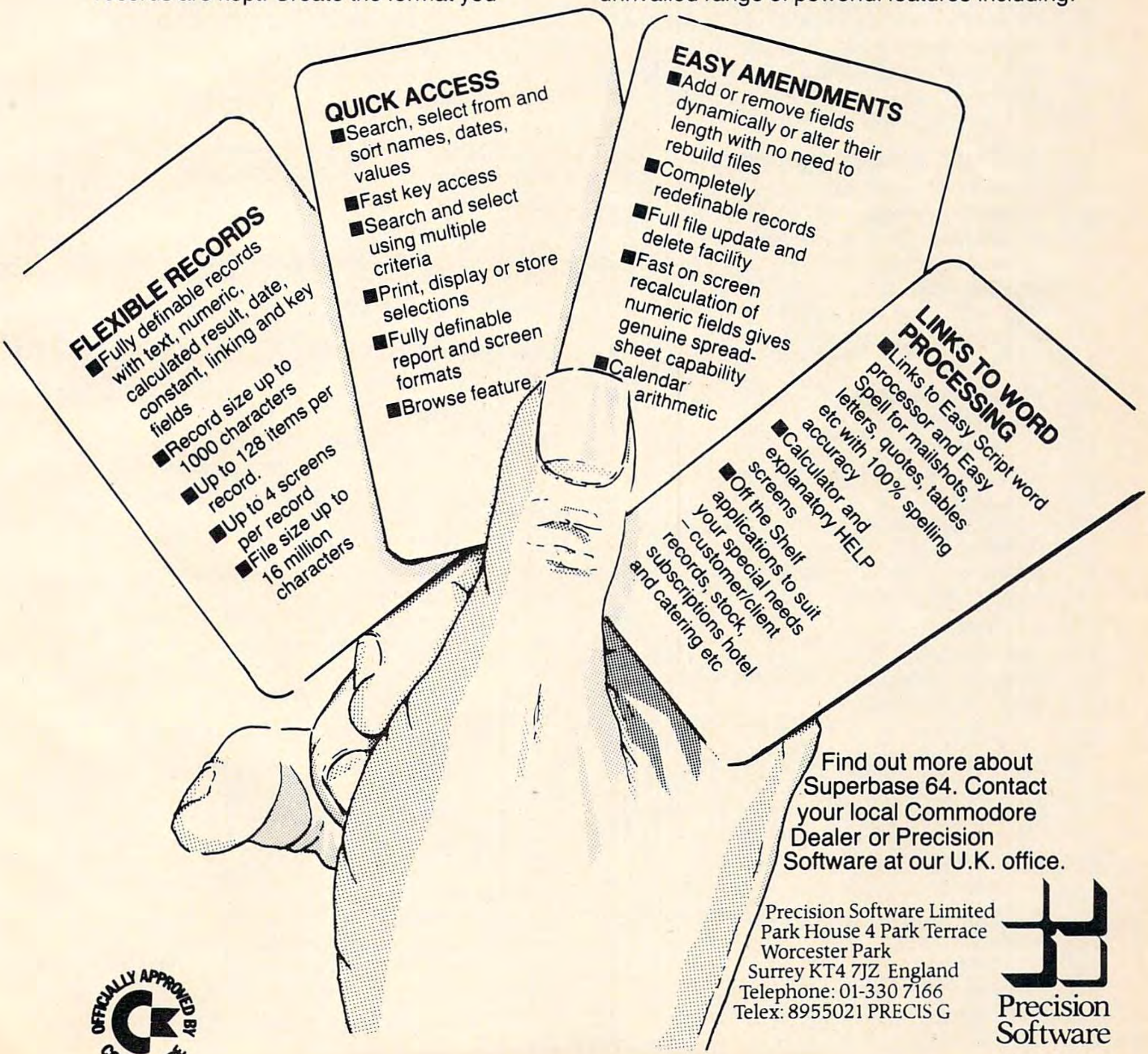
```
POKE 55, 0:POKE 56, 92:CLR
```

To restore your 64 to normal operation, use the following commands:

```
POKE 53265,27:POKE 53270,200:POKE 53272,20:
  POKE 56576,151
```

Strengthen your hand with Superbase 64

The complete information control system for the Commodore 64. Ideal for any home, business or professional environment where records are kept. Create the format you need and enter your records. If the layout or data field sizes are not quite right, correct them and carry on. Superbase gives you an unrivalled range of powerful features including:



Find out more about Superbase 64. Contact your local Commodore Dealer or Precision Software at our U.K. office.

Precision Software Limited
Park House 4 Park Terrace
Worcester Park
Surrey KT4 7JZ England
Telephone: 01-330 7166
Telex: 8955021 PRECIS G

Precision Software

OFFICIALLY APPROVED BY COMMODORE

BINARY AND BITMAPPING

Lance Elko, Assistant Editor

The Commodore 64's high-resolution graphics screen consists of 64,000 (320 by 200) dots or *pixels*. Each one can be turned on or off to let you create your own special graphics. This technique is called *bitmapping*.

At first glance, you might think that if there are 64,000 pixels to control, you'll need to use 64,000 memory cells (bytes) — but this would use more memory than you have available. With bitmapping, one byte controls not one, but *eight* pixels. Since there are eight bits (a bit is the smallest unit of storage in the computer's memory) in one byte, each bit represents one pixel on the hi-res screen. So, only 8,000 (roughly 8K) bytes are needed for bitmapping. Let's see how the computer handles these bits and bytes.

Filaments And Light Bulbs

Computers use the *binary* numbering system rather than the decimal system we're used to. A good way to understand how binary works is to think of a row of light bulbs, each capable of being on or off. The row has eight light bulbs and represents a byte; and each bulb represents one bit. If they are all off:

00000000

we have a value of zero. Now let's turn on the right one:

00000001

This gives us a value of 1. So far, it's not at all tricky.

The next bulb, counting from the right, however, has a somewhat different construction: It has two filaments. If just this bulb is on, it is indicated as:

00000010

but, remember, this bulb has *two* filaments, so the value here is 2. Let's go back and turn on the first bulb, also:

00000011

We now have a value of 3. Two bulbs are on, but three filaments are lit. The next bulb, the third from the right, contains *four* filaments (*twice* the number of the last bulb). So, if this is turned on:

00000100

we have a value of 4. If we turn on the previous bulbs:

00000111

we have 6 (4+2+1) filaments, but only 3 bulbs turned on. The binary value of 00000111, then, equals the decimal value of 6. We can see a pattern emerging here: Each bulb has twice the number of filaments as the one before it:

00000001 = 1	00010000 = 16
00000010 = 2	00100000 = 32
00000100 = 4	01000000 = 64
00001000 = 8	10000000 = 128

Converting Decimal To Binary

On/off combinations of these bulbs will yield any number between 0 and 255 (11111111). Let's pick a number, say 209, and figure out how to represent that number in binary. In other words, if we need exactly 209 filaments lit, which light bulbs should we turn on?

Since we can get 128 of them out of the way, let's do that first:

10000000 (128)

If we add the next available light bulb, with 64 filaments, that will get us up to 192 (128+64):

11000000 (192)

Now, we can't use the next bulb (with 32 filaments) because that would exceed our requirement of 209; so let's check the next one, 16. We can turn this one on because it would get us closer to our goal without going over (192+16=208):

11010000 (208)

We need only one more to make 209, and that's easy because there's only one bulb with one filament, the first one we discussed. Let's turn this one on:

11010001 (209)

and now we have 209 filaments turned on with only 4 light bulbs.

How does all this apply to bitmapping? The VIC-II chip, a microprocessor in the 64 that controls video display, scans an area of memory reserved exclusively for bitmapping. The chip reads each bit in every byte in this area, looking for 1s (on) and 0s (off). When a 1 is noted, the pixel it represents is turned on, and when a 0 is noted, the pixel remains the same as the background color.

Keeping in mind these points about binary numbers, take a look at Michael Tinglof's article to see how to control bits and bytes for effective bitmapping. He also discusses special commands used for manipulating the binary figures we discussed. You might find pages 121 – 28 in the *Commodore 64 Programmer's Reference Guide* helpful as well.

A Graphics Demonstration

If all the computations needed to find the right bit seem complicated to you, and the two-color mode would be satisfactory, use the following utility program. It is written in machine language to increase speed, and can be used through X and Y coordinates. It is accessed from BASIC via the SYS command.

The format of the SYS call is as follows:

SYS (base address of code), command, operand(s)

The commands for the utility are as follows:

- 0 = clear bitmap page (set all bytes to 0)
- 1 = set screen color. Set all bytes in screen memory to the operand. For example, SYS(BS),1,32 sets every byte in screen memory to 32.
- 2 = set point. Set a given point according to its X and Y coordinates. Note that the upper-left corner is (0,0) and the bottom right is (319,199). For example, SYS(BS),2,28,122 sets point (28,122).
- 3 = clear point. The format is the same as above.

This machine language utility is relocatable and can be loaded into memory anywhere simply by changing the pointer in the BASIC loader. Before the utility can be used, however, addresses 680 and 681 must be set. Set address 680 to the start address of the bitmap divided by 256. Likewise, set address 681 to the start address of screen memory divided by 256. If you have set up the bitmap as shown in our working examples, use these POKEs:

POKE 680, 96:POKE 681, 92

To see how the utility and various aspects of bitmapped graphics work, look at the following program, which draws a sine curve on the screen.

Bitmapped graphics are a powerful part of the 64's repertoire. Once mastered, the results can be spectacular. Remember, the best way to learn is by hands-on practice. Once you feel comfortable with the techniques we've covered, try some of these ideas:

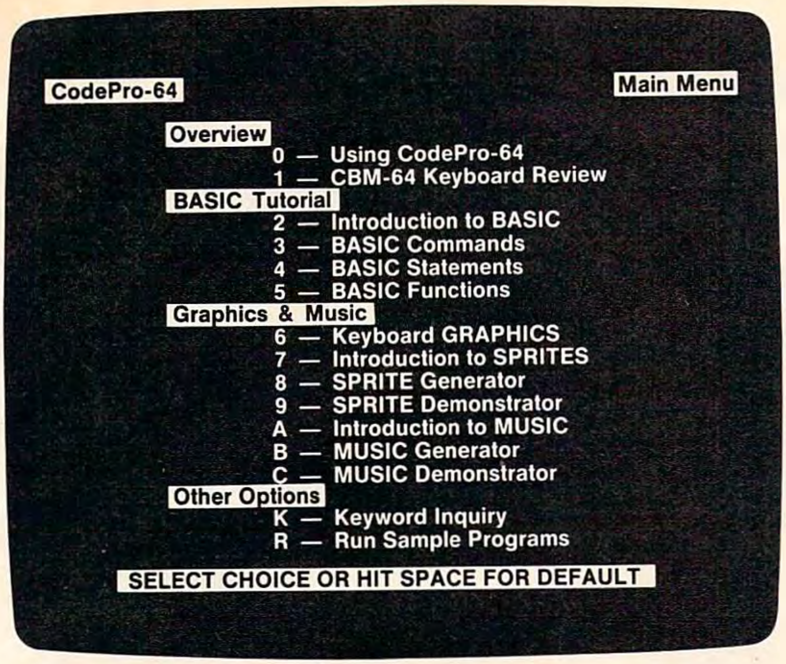
1. Draw the picture into memory, then switch the pointers to it. This makes the graphics appear lightning fast, even from BASIC.

2. Use several bitmaps and switch the pointers between them. Again, this gives the appearance of lightning fast graphics.

3. Use sprites. Since the sprites are totally independent of the background, you can create some fantastic graphics for games.

Sine Curve Graphics

```
3 REM{2 SPACES}COMMAND:
4 REM{5 SPACES}XX SYS (BASE),OPTION,DATA
5 REM{4 SPACES}OPTIONS:
6 REM SYS B, Ø{2 SPACES}-{2 SPACES}CLEAR
  {SPACE}SCREEN
7 REM SYS B, 1, CL - SET COLOR CL
8 REM SYS B, 2, X, Y - SET POINT (X,Y)
9 REM SYS B, 3, X, Y - CLEAR POINT
1Ø AD=32768:REM ** BASE ADDRESS
2Ø READD:IFD=-1THEN5ØØ:REM ** JUMP TO USE
  R ROUTINE
3Ø POKEAD,D:AD=AD+1:GOTO2Ø
1ØØ DATA 32, 115, Ø, 32, 158, 173, 32, 24
    7, 183, 14Ø, 17Ø, 2, 192, Ø
11Ø DATA 24Ø, 6, 192, 1, 24Ø, 32, 2Ø8, 77
    , 173, 168, 2, 133, 252, 24
12Ø DATA 1Ø5, 32, 133, 253, 169, Ø, 133,
    {SPACE}251, 168, 145, 251, 23Ø, 251,
    {SPACE}2Ø8
13Ø DATA 2, 23Ø, 252, 166, 252, 228, 253,
     144, 242, 96, 32, 115, Ø, 32
14Ø DATA 158, 173, 32, 247, 183, 132, 253
    , 173, 169, 2, 56, 233, 1, 133
15Ø DATA 252, 24, 1Ø5, 4, 133, 254, 169,
    {SPACE}8, 133, 251, 16Ø, 247, 165, 25
    3
16Ø DATA 145, 251, 23Ø, 251, 2Ø8, 2, 23Ø,
     252, 166, 252, 228, 254, 144, 242
17Ø DATA 96, 32, 115, Ø, 32, 158, 173, 32
    , 247, 183, 14Ø, 171, 2, 141
18Ø DATA 172, 2, 32, 115, Ø, 32, 158, 173
    , 32, 247, 183, 14Ø, 173, 2
19Ø DATA 152, 41, 248, 133, 253, 141, 18Ø
    , 2, 141, 174, 2, 169, Ø, 133
2ØØ DATA 254, 141, 181, 2, 162, 4, 24, 38
    , 253, 38, 254, 2Ø2, 16, 248
21Ø DATA 162, 2, 24, 46, 18Ø, 2, 46, 181,
     2, 2Ø2, 16, 246, 24, 165
22Ø DATA 253, 1Ø9, 18Ø, 2, 141, 178, 2, 1
    65, 254, 1Ø9, 181, 2, 141, 179
23Ø DATA 2, 173, 171, 2, 41, 248, 141, 17
    6, 2, 173, 172, 2, 141, 177
24Ø DATA 2, 56, 173, 173, 2, 237, 174, 2,
     24, 1Ø9, 176, 2, 133, 251
25Ø DATA 173, 177, 2, 1Ø9, 168, 2, 133, 2
    52, 24, 173, 178, 2, 1Ø1, 251
26Ø DATA 133, 251, 173, 179, 2, 1Ø1, 252,
     133, 252, 56, 173, 171, 2, 237
27Ø DATA 176, 2, 133, 253, 56, 162, 255,
    {SPACE}169, Ø, 1Ø6, 232, 228, 253, 2Ø
    8
28Ø DATA 25Ø, 141, 18Ø, 2, 174, 17Ø, 2, 2
    24, 3, 24Ø, 1Ø, 16Ø, Ø, 177
29Ø DATA 251, 13, 18Ø, 2, 145, 251, 96, 5
    6, 169, 255, 237, 18Ø, 2, 141
3ØØ DATA 18Ø, 2, 16Ø, Ø, 177, 251, 45, 18
    Ø, 2, 145, 251, 96,-1
5ØØ REM ** USER ROUTINE **
5Ø1 REM GRAPHS SINE CURVE
5Ø5 POKE 53265, PEEK(53265)OR2↑5:REM ** S
    ET BIT MAP MODE
51Ø POKE68Ø,96:POKE681,92:REM ** SET POIN
    TERS FOR UTILITY
515 POKE 53272, 12Ø:POKE 56576, 2:REM **
    {SPACE}SET UP VIC II MEMORY
52Ø POKE 55, Ø:POKE 56, 6Ø:CLR:REM ** PRO
    TECTS BIT MAP FROM BASIC PROGRAM
53Ø B=32768:REM ** SET BASE ADDRESS OF UT
    ILITY
54Ø SYS B,Ø: SYS B,1,16:REM ** CLEAR SCRE
    EN AND SET COLOR
55Ø FOR X=Ø TO 6 STEP .Ø5 :Y=SIN(X):REM *
```



NEW! For the Commodore 64™
ANNOUNCING

CodePro-64™

A new concept in interactive visual learning . . .

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

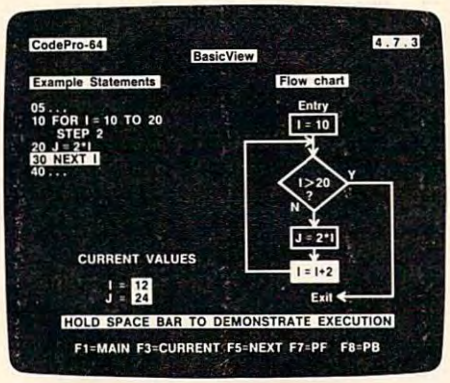
SEE PROGRAM EXECUTION

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

After entering an example you invoke our exclusive **BasicView™** which **shows you how the BASIC program example executes.**

You step through and actually see the execution of sample program statements by simply pressing the space bar. CodePro-64 does the rest.

You see statements with corresponding **flow chart graphics** and variable value displays. You learn by visual examples.



EXTENSIVE TUTORIAL

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

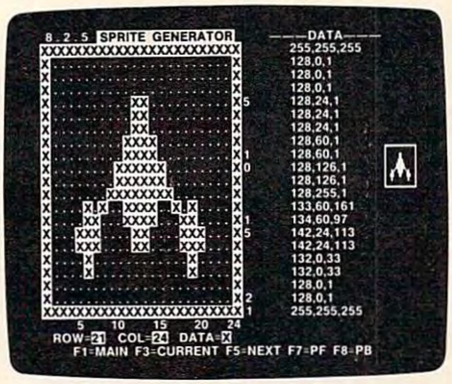
By seeing graphic displays of program segment execution you learn by visual example. You **learn faster and grasp programming concepts easier** with CodePro-64 because you immediately see the results of your input.

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

CodePro-64 **lets you follow your interests** and practice with interactive examples. But you can never get "lost". F1 will always return you to the main menu. Once you have practiced and mastered the BASIC language elements you move on to more advanced concepts. You learn about sprite and music programming.

SPRITE GENERATOR & DEMONSTRATOR

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



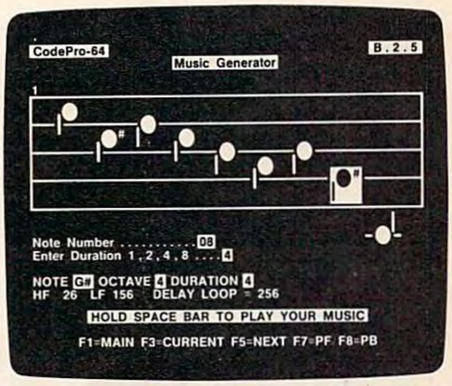
We also help you learn to program with sprites by giving you a **sprite demonstrator** so you can see the effect of changing register values. You can experiment by moving your sprite around in a screen segment, change its color or priority, and see the effects of your changes. You learn by visual examples.

MUSIC GENERATOR & DEMONSTRATOR

To teach you music programming CodePro-64 gives you an interactive music generator and demonstrator. First we help you set all your SID parameters (attack/decay, sustain/release, waveform, etc.). Then you enter notes to play and **we show your tune graphically as it plays,** note by note, on the scale. You learn by seeing and hearing the results of your input.

OUR GUARANTEE

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



Our music demonstrator **lets you experiment** with various combinations of music programming parameters and hear the results. You can **quickly modify any of the SID register values** to hear the effects of the change. For example, you could easily change waveform and attack/decay values while holding all other SID values constant. By seeing your input and hearing the result you quickly learn how to create new musical sounds and special sound effects.

AND MORE . . .

We don't have enough space to tell you everything CodePro-64 offers. You need to see for yourself. BASIC tutorials, graphics, sprites, music, keyboard review, sample programs—the main menu shown above gives you just a summary of the contents of this powerful educational product.

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

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

HOW TO ORDER

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

SMA **SYSTEMS MANAGEMENT ASSOCIATES**
3700 computer Drive, Dept. C
Raleigh, N. C. 27609

Available on **diskette only.** MasterCard/VISA accepted. For faster service on credit card orders call (919) 787-7703.

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

Ad no. 733, Copyright 1983, SMA
Dealer inquiries invited.

```
    * GET VALUE FOR SINE CURVE
560 X1=X*50:Y=Y*50:REM ** ENLARGE GRAPH S
    IZE
570 Y=100-Y:SYS B,2,X1,Y:REM ** GRAPH POI
    NT
580 NEXT X:REM ** GRAPH NEXT
590 GOTO 590
600 REM ** EXIT WITH BREAK/RESTORE
```

©

Atari Screenbyter

Carl Zahrt and Orson Scott Card

Here's a graphics utility that lets you create screen displays in any of the regular pixel graphics modes – and GRAPHICS 6.5 and 7.5 as well. It's simple enough for a child to use. It gives you complete control over color, mode, and display size. And a special Fill Mode lets you quickly draw long lines or fill large areas with color in moments.

Atari home computers have superb graphics. Creating screen displays from BASIC, page flipping, scrolling, redefining characters, continuous memory, and changing from mode to mode to get exactly the effect you want – once you've worked with graphics on the Atari, some other home computers can seem a bit confining.

But that doesn't mean using Atari graphics is *easy*, especially if you want large displays which extend far beyond the edges of the TV screen, or detailed drawings that would take hundreds of PLOT and DRAWTO statements to create from BASIC. Such things take painstaking work on graph paper and many POKEs into screen memory – or a good chunk of your paycheck for software to do it for you.

"Screenbyter" takes the pain out of creating beautiful graphics displays.

- You can work in any of the non-GTIA pixel modes.
- You have access to GRAPHICS 6.5 and 7.5, pixel modes that cannot be used with a simple GRAPHICS statement.
- You can type RUN and start drawing with the joystick – no programming experience is needed.
- You can fill in large areas quickly and easily.
- Since the main action of the program is in machine language, it moves very quickly, but a Slow Mode is provided so you can do detail work, pixel by pixel.
- You can change screen colors with the joystick.
- You aren't always limited by the size of the screen. In GRAPHICS 3 you can create scrolling displays many times larger than the TV screen, and all the modes except 7.5 and 8 allow some scrolling.
- When you save a display to disk, all the parameters – mode, size, and colors – are saved with the screen data, so that you can load them directly into your own programs.

Using Screenbyter

Setup. Screenbyter begins by displaying a directory of all files on the disk with the extender ".PIX". This extender is automatically added to all files created by Screenbyter. If no directory appears, there are no previously saved files on the disk.

"What file should hold your finished screen? (Eight characters)." Respond to this prompt by giving the filename you want your new display to have, when you save it at the end of the editing session. Screenbyter automatically removes everything before a colon or after a period and replaces it with "D1:" and ".PIX", so that you only need to enter the eight-letter filename. If you use illegal characters, Screenbyter will ask you to try again; if you use more than eight characters, only the first eight characters will be used.

If the name you enter is the name of a file already on disk, Screenbyter will remind you of that. To change the name, press RETURN. Or, if you want your new display to overwrite the old file, press any other key to go on.

"Would you like to edit a screen you have already saved? (Y or N)." If you answer *Y*, Screenbyter asks you for the name of the saved file. If the file is not on disk in the form "D1:*filename*.PIX", Screenbyter will tell you and ask you to insert the correct disk or, if you wish, ask you again if you want to edit a previously saved screen.

Once the file is found, Screenbyter reads the first four bytes of the file to get the mode number, the number of bytes per line, and the number of lines in the display as it was saved. Press RETURN if you want to change these parameters. Press any other key to leave them the same.

Changing the parameters can have interesting effects. Remember that four-color modes all read the bytes the same way; if you want to draw your displays in GRAPHICS 3 (ANTIC 8) and then display them in a higher four-color mode, you can. Changing the length of a file either chops off the bottom or adds blank lines at the bottom of the display. Changing the line width, however, will usually result in garbage, since the vertical relationships will all be changed. The option is included, however, because sometimes even "garbage" can be fun.

If you are not editing a previously saved display, or if you are changing the parameters, you get the following series of prompts:

"What Antic mode will you work in?" This prompt is followed by a table that lists the eight ANTIC pixel modes and their graphics mode equivalent. ANTIC 8, for instance, is GRAPHICS 3; ANTIC F (15) is GRAPHICS 8. Two ANTIC modes, C (12) and E (14), have no GRAPHICS equivalent – they are the famous "GRAPHICS 6.5" and "GRAPHICS 7.5." (See Table 1.) Enter the ANTIC mode number: 8, 9, A, B, C, D, E, or F.

"How wide a line? (Minimum *nn* bytes, maximum *nn* bytes)." Depending on the mode you chose, Screenbyter will give you the minimum and maximum number of bytes per line. Remember that in the four-color modes, each byte is four pixels, while in the two-color modes, each byte is eight pixels. The minimum is based on the minimum number of bytes required to fill the screen. The maximum is based on the widest possible line that will allow the display to fit within 4K. If you enter numbers outside the legal range, Screenbyter will select the minimum or maximum, as appropriate.

With ANTIC *E* and *F*, the minimum and maximum are the same – you have no option, so any number you enter will result in the same number of bytes per line. This is because these two modes will not scroll – they both require more than 4K. Scrolling a screen that crosses a 4K boundary requires elaborate arrangements of screen memory that are beyond the scope of this program. Displays created in *E* and *F* will take up 65 sectors on disk; all other displays will take up 33 sectors or fewer.

Table 1: Atari Pixel Modes

ANTIC mode	8	9	A	B	C	D	E	F
Graphics mode	3	4	5	6	—	7	—	8
Colors	4	2	4	2	2	4	4	2
Resolution	24 x 40	48 x 80	48 x 80	96 x 160	192 x 160	96 x 160	192 x 160	192 x 320
Memory, bytes (sectors)	240 (3)	480 (5)	960 (9)	1920 (17)	3840 (33)	3840 (33)	7680 (65)	7680 (65)
Lines/screen	24	48	48	96	192	96	192	192
Bytes/line	10	10	20	20	20	40	40	40
Bits/pixel (Pixels/byte)	2 (4)	1 (8)	2 (4)	1 (8)	1 (8)	2 (4)	2 (8)	1 (4)
Scan lines/ pixel	8	4	4	2	1	2	1	1
Color clocks/ pixel	4	2	2	1	1	1	1	½

Note: ANTIC C and E, the two "hidden" pixel modes, provide the same resolution. All the other pixel modes attempt to create as square a pixel as the TV screen allows – the same number of color clocks wide as scan lines high. *C* and *E*, however, are twice as wide as they are high, making each pixel very short and wide. They come very near the resolution of ANTIC F (GRAPHICS 8). The advantages are that, compared to *F*, *C* uses half the memory and *E* allows four colors.

"How many lines do you want to edit? (Minimum *nn*, maximum *nn*)." The minimum and maximum depend on the mode and the number of bytes per line already selected. Again, if you choose parameters outside the legal range, Screenbyter will select the minimum or maximum. And if you choose the maximum number of bytes per line, only the minimum number of lines per screen will be possible.

When all selections have been made, you are given one last chance to change your mind. All the parameters you chose are displayed on the screen. If they are correct, press START, and the program will go on. If you want to make changes, press OPTION and the program will start over.

Waiting. What's going on while you wait? Screenbyter configures the memory to reserve 10K (40 pages) at the top of memory to hold screen memory (up to 8K), the display list, and the machine language routine that actually puts your drawing on the screen. Screen memory is cleared and the machine language routines are loaded. If you chose to edit a previously saved screen, it is loaded into memory now. All this takes about six seconds. The rest of the time is spent writing the display list. The higher the ANTIC mode, the longer it takes to write the display list – ANTIC F requires about 200 POKEs in BASIC, plus the calculations to find out what numbers to POKE, and it can take as long as 20 seconds.

When Screenbyter is ready for you to edit, there will be a cursor in the upper-left-hand corner.

Moving the cursor. The joystick controls the cursor.

Drawing a line. Hold down the joystick button to draw; let it up to move the cursor without drawing.

Selecting a color. Press 1 or SHIFT-CAPS/LOWR to select Color 1. Press 2 or CONTROL-CAPS/LOWR for Color 2. Press 3 or SHIFT-CONTROL-CAPS/LOWR for Color 3. Press 0 or CAPS/LOWR to select the background color. Drawing in the background color has the effect of erasing.

Color Mode. To change the actual colors that are displayed by Colors 1, 2, or 3, or the background color, press START. You will hear a buzz, and the cursor will no longer respond to the joystick. Instead, moving the joystick will change the colors displayed on the screen. Moving the joystick up or right will change the color from darker to brighter, then jump to the darkest value of the next color. Moving the joystick down or left will change the color from brighter to darker, then jump to the brightest value of the next color.

To change the background color, move the joystick forward or back; to change Color 3, move the joystick left or right. To change Color 2, move the joystick forward or back with the button pressed; to change Color 1, move the joystick left or right with the button pressed.

To return to Cursor Mode, press START again. No other commands will work during Color Mode.

Slow Mode. Press the space bar to enter Slow Mode. A delay loop in the program makes the cursor move much more slowly around the screen, with a click between moves. This mode allows you to create details. To return to Fast Mode, press the space bar again.

Fill Mode. Press the inverse key (Atari logo key) to enter Fill Mode. A low hum will come from the television. In this mode, when you press the joystick button, Screenbyter draws a dot of the selected color at the current cursor location, as usual, but it also searches to the right along the same line. If it finds another dot of the same color before it reaches the end of the line, it will fill in all the area between that dot and the current cursor position with dots of the same color. If no dot of the same color is found, no fill operation is performed.

This allows you to fill large or small areas of the screen with a single color. Simply draw the right-hand edge of the figure first; then enter Fill Mode and draw the left-hand border. It takes some practice to get used to using this function without accidentally erasing parts of your screen, but you may find that this can be the most useful feature of Screenbyter.

To exit Fill Mode, press the inverse key again. The hum will continue as long as you are in Fill Mode, and will stop only when you leave.

Insert a line. Press SHIFT-INSERT to insert a line at the current cursor position. The bottom line of the display will be pushed down and lost.

Delete a line. Press SHIFT-DELETE to delete the current cursor line. A blank line will be added at the bottom of the display.

Clear the screen. Press CONTROL-SHIFT-CLEAR to erase the screen completely. If you haven't already saved the display, it will be lost.

Saving the screen. Press SELECT to save the screen without ending the editing session. The current screen display will be saved as "D1:TEMPFILE.PIX". You can save as often as you like; Screenbyter will simply overwrite any existing TEMPFILE.PIX file.

Ending the editing session. Press OPTION to save the screen and end the editing session. (To exit without saving, press RESET.) The display will be saved as "D1:TEMPFILE.SCR." Then the regular GRAPHICS 0 screen will return and you will be given several prompts:

"Do you want to save the screen as D1:*filename*.PIX? (Y or N)." If you answer *N*, the saved display will be left as TEMPFILE.PIX. If you answer *Y*, Screenbyter will erase any existing file that has the same filename. Then Screenbyter will rename TEMPFILE.PIX with the filename you chose.

"Do you want to quit? (Y or N)." If you answer *Y*, Screenbyter will restore the old top of memory and exit to BASIC. If you answer *N*, you will get another prompt. To return to edit the screen you just left, press OPTION. That display will be reloaded into memory, the display list will be rewritten, and you can start over. To edit an entirely new screen, or to change the name of the save file, press START. In effect, Screenbyter will then start over.

What's Going On Inside The Program?

Like everything else in a computer, your display exists as a series of numbers stored in binary form in memory locations in the computer. The ANTIC chip scans screen memory as it is instructed to do by the display list. But it doesn't read the numbers as numbers. Instead, it reads them as patterns of "on" and "off" bits.

Four-color modes. In the four-color modes, each byte is read as code for four pixels. The eight-bit binary number is treated as four bit-pairs:

```
00 00 00 00
```

Each bit-pair provides the code for one pixel, or rectangle of color on the screen. In GRAPHICS 3, each pixel is the size of a character in GRAPHICS 0. In GRAPHICS 7.5, each pixel is one scan line high and one color clock wide, which gives very good resolution. But all four-color modes read the bit-pairs the same way.

00 means to display the background color (the color code stored at location 712).

01 means to display Color 1 (the color code stored at location 708).

10 means to display Color 2 (the color code stored at location 709).

11 means to display Color 3 (the color code stored at location 710).

This means that the number 216 (binary 11011000) is treated as four pixel color instructions: The first pixel is Color 3, the second pixel is Color 1, the third pixel is Color 2, and the last pixel is the background color.

Two-color modes. The two-color modes treat each bit as a separate pixel instruction, so that each byte controls eight pixels. An "on" bit, or 1, is read as a Color 1 instruction, while an "off" bit, or 0, is read as a background color instruction. In a two-color mode, the number 216 would be treated as eight pixel color instructions: Two "on" pixels, one "off" pixel, two more "on" pixels, and three "off" pixels. (See Table 1 for a listing of all the modes.)

Moving around the screen. Moving the cursor around the screen, then, isn't simply a matter of moving from one byte to the next in screen memory. Screenbyter also has to move from bit to bit or from bit-pair to bit-pair within the bytes. This *can* be done in BASIC by adding or subtracting values, but it is very slow. Machine language, however, has powerful commands that make it easy to move from bit to bit. DRAWTO and PLOT commands do these manipulations for you, but since Screenbyter is circumventing the BASIC graphics commands entirely, there was no practical choice but to execute the main drawing operations in machine language.

To understand what Screenbyter is doing, you need to understand a few machine language commands: EOR, ORA, and AND. The two OR instructions and the AND instruction are not the same as the AND and OR you use in Atari BASIC. In machine language, these are operations on the bits of an eight-bit number, and are often called "bitwise" AND and OR to help keep the difference in mind.

AND, OR, EOR Explained

All three operations compare two numbers, one stored in the accumulator and another somewhere else in memory. The operation results in a third number, which is stored in the accumulator in place of the number that was already there.

• AND, referred to as "bitwise AND," compares the two numbers, bit by bit. Any bit that is on in both numbers *stays* on in the resulting number. All other bits are turned off. In other words, *only* bits that are on in the first number *and* in the second number remain on in the result.

	10010110
AND	11110000
results in	10010000

• ORA, referred to as "bitwise OR," compares the two numbers, but in this case any bit that is on in *either* number stays on in the result:

	10010110
ORA	11110000
results in	11110110

• EOR, referred to as "exclusive OR," compares the two numbers, and any bit that is on in one and *only* one number is left on in the result. Any bit that is on in both numbers or off in both numbers is off in the result:

	10010110
EOR	11110000
results in	01100110

How do these actually work, in practice?

Screenbyter maintains several *masks*. The Color Mask is in page 6, at memory location 1692. This byte is set from BASIC whenever the color is changed, and it is set so that every bit or bit-pair represents a pixel of the selected color. If the background color is selected, the Color Mask is 00000000. If Color 1 is selected, the Color Mask is 01010101. For Color 2, the Color Mask is 10101010, and for Color 3 it is 11111111. With two-color modes, the Color Mask is either 00000000 or 11111111.

The Cursor Mask is kept at location 1696. It is set to represent the current cursor pixel within the cursor byte. The bits in the current pixel are on; all others are off. In four-color modes, if the cursor is in the leftmost pixel of the cursor byte, the Cursor Mask will be set to 11000000; if it is in the rightmost pixel, the mask will be set to 00000011. The two middle pixels are 00110000 and 00001100. In two-color modes, a single "on" bit represents the cursor position.

Whenever you move the cursor left or right or diagonally, the Cursor Mask is shifted left or right, so that at any given moment, the Cursor Mask will mark which bit or bit-pair Screenbyter should change.

If you are drawing, Screenbyter first picks up the value of the current cursor byte and stores it at 1690. Then it picks up the Cursor Mask and EORs it with 11111111 (decimal 255). This reverses the Cursor Mask – any bit that was on is now off, and any bit that was off is now on.

Let's see that in action in a four-color mode, in which the background is black, Color 1 is red, Color 2 is green, and Color 3 is blue. The bit-pairs will be separated in these examples, to make it easier to keep track of the pixels.

Cursor Mask	00 11 00 00	
EOR	11 11 11 11	
results in	11 00 11 11	**(Reverse Cursor Mask)**

Screenbyter then ANDs the Reverse Cursor Mask with the number at 1690, which in effect makes a hole in the cursor position:

Reverse Cursor Mask	11 00 11 11				
AND	01 01 01 11	red	red	red	blue
results in	01 00 01 11	red	—	red	blue

The two bits in the cursor position will *always* be turned off.

Now Screenbyter must prepare the pixel code to go in that hole. Screenbyter picks up the Cursor Mask and ANDs it with the Color Mask. Since all the bits in the Cursor Mask are off except the two bits of the current pixel, the resulting number will have only the bits that represent the current color, and only in the pixel position:

Cursor Mask	00 11 00 00				
AND Color Mask	10 10 10 10	green	green	green	green
results in	00 10 00 00	—	green	—	—

Now we are ready to put the correct pixel code into the hole in the current cursor byte. To do this, we bitwise OR the current pixel we just got with the cursor byte with a hole in it from the operation before. Remember that with ORA, any byte that is on in either or both of the two numbers is on in the result:

correct pixel	00 10 00 00	—	green	—	—
ORA current byte with hole	01 00 01 11	red	—	red	blue
results in	01 10 01 11	red	green	red	blue

The result is then stored in 1690, and later in the program it is put into screen memory.

If you are not drawing (merely moving the cursor) the operation is a little different, but AND, EOR, and ORA perform the same functions.

Machine language is so fast that all this seems to happen instantaneously. In fact, the only reason the cursor doesn't fly around the screen out of control is because Screenbyter keeps leaving the machine language routine, returning to BASIC to check the keyboard for other commands. Even so, the cursor moves so quickly that it has to be slowed down in order to allow you to draw details.

Use of Page 6. The machine language routine at SCROLL uses a field in Page 6 to hold some important variables. The memory locations in Page 6 are explained in Table 2.

Screenbyter Displays In Your Own Programs

Here are two routines you can add to your own programs, which will allow you to load the displays you created with Screenbyter. The first routine, Load and Display List, works with any Screenbyter file. However, it sets up a custom display list with individual LMS instructions, suitable for scrolling. This makes the setup time rather long. So a Simple Load Routine is also included. It will work with any display file that was created using the *minimum* line width and number of lines per screen, except screens created in ANTIC C and E (GRAPHICS 6.5 and 7.5). You cannot use it if you intend to scroll horizontally. However, you *can* use it if you intend to scroll vertically or flip pages, and if your display was created with the minimum line width.

Table 2: Page 6 Locations

1670	WIDE-1. Used to check for the end of the logical line.
1671	Used in fill routine to keep track of right border of fill.
1672	Cursor location: current byte on logical line.
1673	Used by the fill routine to hold the pattern of the rightmost byte of the fill line.
1674-1675	LINE-1. Used to check for last line of display.
1676-1677	Cursor location: current logical line number.
1678	Bytes per screen line-1. Used by the scrolling routine to check for the end of the screen line.
1679	Cursor location: Current byte on screen line.
1680	Lines per screen-1. Used by the scrolling routine to check for the bottom of the screen display.
1681	Cursor location: current screen line number.
1682	Used by the fill routine to hold the pattern of the leftmost byte of the fill line.
1683	A temporary holding location.
1684	Used by the fill routine to hold the real value of the byte currently being tested.
1685	A temporary holding location.
1686-1687	The current screen starting address (the address of the upper-left-hand corner of the screen).
1688-1689	Cursor location: the address of the current cursor byte in screen memory.
1690	The real contents of the current cursor byte.
1691	The reverse (cursor display) contents of the current cursor byte.
1692	Color Mask.
1693	The number of bits per pixel (1 or 2).
1694	Scroll flag (0 = do not scroll).
1695	Fill flag (0 = do not fill).
1696	Cursor Mask.
1697	Joystick value.
1698	Total number of lines per screen. Used in the scroll routine to change the correct number of LMS instructions in the display list.
1699	WIDE. Used in the scroll routine to increment the LMS addresses in the display list.
1700	Fill Test Mask. Used in the fill routine to isolate and test each pixel until a pixel of the selected color is found.
1701	Starting Fill Test Mask. Either 192 (four-color mode) or 128 (two-color mode).
1702-1704	Machine language jump vector: JMP followed by the address of the fill subroutine held in the string FILL$.

Both routines will configure memory to protect the screen display, read the display parameters from whatever display file you choose, and load the file into memory. It uses a load routine very similar to the one used by Fontbyter, so we won't explain them again here.

Notice that in loading displays created in ANTIC E and F (GRAPHICS 7.5 and 8), the screen display must cross a 4K boundary line. The ANTIC chip gets fussy at this point, and ignores anything after a 4K boundary line until the beginning of the line pointed to by the next LMS instruction. Therefore, screen memory must be arranged so that the 4K boundary line comes right at the end of a line; the display list routine will have set the value of SC, the start of screen memory, so that the 4K boundary line will fall right at the end of a line.

Program 1: Load And Display List Routine

```
5 CLR :DIM PPB(7),BPL(7),MXW(7),LPS(
  7),FL$(20):FL$="D1:SHIP.PIX":GOSUB
   4ØØØ
4ØØØ FOR I=Ø TO 7:READ W,N,C,T:PPB(I
     )=W:BPL(I)=N:MXW(I)=C:LPS(I)=T:
     NEXT I
4ØØ5 A=PEEK(1Ø6):TOP=A-36:SP=TOP+4:S
     C=SP*256:DL=256*TOP:POKE 1Ø6,TO
     P:GRAPHICS Ø:PRINT "{CLEAR}"
4Ø1Ø X=16:ICCOM=834:ICBADR=836:ICBLE
     N=84Ø:SCON=PEEK(559):K4=4Ø96
4Ø15 OPEN #1,4,Ø,FL$:GET #1,M:M8=M-8
     :GET #1,WIDE:GET #1,LLO:GET #1,
     LHI:LINE=LLO+256*LHI:SZ=WIDE*LI
     NE
4Ø2Ø FOR I=7Ø8 TO 711:GET #1,N:POKE
     I,N:NEXT I:POKE I,N
4Ø25 SC=SC+((LINE*WIDE)>K4)*(K4-INT(
     K4/WIDE)*WIDE):SH=INT(SC/256):S
     L=SC-256*SH
4Ø3Ø FOR I=Ø TO 2:POKE DL+I,112:NEXT
      I:N=Ø
4Ø35 FOR I=DL+3 TO DL+3*LPS(M8) STEP
      3:C=SC+N*WIDE:POKE I,64+M:T=IN
     T(C/256)
4Ø4Ø POKE I+2,T:POKE I+1,C-256*T:N=N
     +1:NEXT I
4Ø45 POKE I,65:POKE I+1,Ø:POKE I+2,D
     L/256
4Ø5Ø POKE 56Ø,Ø:POKE 561,DL/256
4Ø55 POKE ICBADR+X+1,SH:POKE ICBADR+
     X,SL:POKE ICBLEN+X+1,1+INT(SZ/2
     56):POKE ICBLEN+X,Ø
4Ø6Ø POKE ICCOM+X,7:I=USR(ADR("hhh■L
     V■"),X):CLOSE #1:RETURN
4Ø65 DATA 2,1Ø,17Ø,24,1,1Ø,85,48,2,2
     Ø,85,48,1,2Ø,42,96
4Ø7Ø DATA 1,2Ø,21,192,2,4Ø,42,96,2,4
     Ø,4Ø,192,1,4Ø,4Ø,192
```

Program 2: Simple Load Routine

```
5 CLR :DIM GM(15),FL$(2Ø):FL$="D1:G8
  .PIX":GOSUB 4ØØØ
6 FOR I=Ø TO 3ØØØØ:NEXT I
4ØØØ FOR I=Ø TO 15:READ N:GM(I)=N:NE
     XT I
4ØØ5 A=PEEK(1Ø6):TOP=A-36:SP=TOP+4:S
     C=SP*256:DL=256*TOP:POKE 1Ø6,TO
     P:GRAPHICS Ø:PRINT "{CLEAR}"
4Ø1Ø X=16:ICCOM=834:ICBADR=836:ICBLE
     N=84Ø:SCON=PEEK(559):K4=4Ø96
4Ø15 OPEN #1,4,Ø,FL$:GET #1,M:GET #1
     ,WIDE:GET #1,LLO:GET #1,LHI:LIN
     E=LLO+256*LHI:SZ=WIDE*LINE
4Ø2Ø FOR I=7Ø8 TO 711:GET #1,N:POKE
     I,N:NEXT I:POKE I,N
4Ø25 SC=SC+((LINE*WIDE)>K4)*(K4-INT(
     K4/WIDE)*WIDE):SH=INT(SC/256):S
     L=SC-256*SH
4Ø3Ø GRAPHICS GM(M)+16:IF GM(M)=Ø TH
     EN ? "INVALID MODE":RETURN
4Ø35 DL=PEEK(56Ø)+256*PEEK(561):DL4=
     DL+4:DL5=DL+5:POKE DL4,SL:POKE
     DL5,SH
4Ø55 POKE ICBADR+X+1,SH:POKE ICBADR+
     X,SL:POKE ICBLEN+X+1,1+INT(SZ/2
     56):POKE ICBLEN+X,Ø
4Ø6Ø POKE ICCOM+X,7:I=USR(ADR("hhh■L
     V■"),X):CLOSE #1:RETURN
4Ø65 DATA Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,3,4,5,6,Ø,
     7,Ø,Ø
```

Program: Screenbyter

After the main listing of the BASIC program, you will find several programs to create disk files containing the machine language routines used in Screenbyter. If you prefer, you can easily add these DATA statements to your program and read them that way, or – as we prefer to do – load them into string constants and use them that way, without so many disk accesses. However, typing in strings that have lots of inverse and control characters in them can be tedious and often leads to typing errors, so these DATA statements are necessary in the published version of the program.

If you are also using "Fontbyter" (COMPUTE!, September 1983), you might notice that Screenbyter follows the same structure. That's because Fontbyter was used as the starting point, and changed wherever Screenbyter's needs were different. However, the line insert, line delete, and clear screen machine language routines are *not* identical, so don't try to use the similar Fontbyter routines for Screenbyter – you will hopelessly confuse your Atari if you do, and confused Ataris have unpleasant ways to express their frustration.

Program 3: Screenbyter

```
5 DIM FSAVE$(2Ø),FLOAD$(2Ø),FL$(4Ø),
  FLL$(2Ø),DELETE$(118),EXPAND$(1Ø2)
  ,N$(13),FILL$(23Ø),CLEAR$(26)
1Ø DIM PPB(7),BPL(7),MXW(7),LPS(7),C
   OL(11),CL(3)
15 A=PEEK(1Ø6):TOP=A-4Ø:SP=TOP+8:SC=
   SP*256:DL=256*TOP:SCROLL=DL+6ØØ:P
   OKE 1Ø6,TOP
2Ø X=16:ICCOM=834:ICBADR=836:ICBLEN=
   84Ø:GRAPHICS Ø:SCON=PEEK(559):F=1
```

```
     670:K4=4096:N$="No equivalent"
25 C=707:FOR I=0 TO 7:IF I/2=INT(I/2
   ) THEN C=C+1:IF C=711 THEN C=712
30 COL(I)=C:NEXT I:CL(0)=0:CL(1)=85:
   CL(2)=170:CL(3)=255:FMS=ADR("hhh■
   LV■")
35 RESTORE 770:FOR I=0 TO 7:READ W,N
   ,C,T:PPB(I)=W:BPL(I)=N:MXW(I)=C:L
   PS(I)=T:NEXT I:POKE 16,112:GOTO 3
   15
40 OPEN #1,4,0,FL$:GET #1,MD:GET #1,
   WD:GET #1,LLO:GET #1,LHI:LN=LLO+2
   56*LHI:SZ=WD*LN
45 FOR I=0 TO 6 STEP 2:GET #1,N:POKE
    COL(I),N:NEXT I
50 POKE ICBADR+X+1,SH:POKE ICBADR+X,
   SL:POKE ICBLEN+X+1,1+INT(SZ/256):
   POKE ICBLEN+X,0
55 POKE ICCOM+X,7:I=USR(FMS,X):CLOSE
    #1:RETURN
60 OPEN #1,8,0,"D1:TEMPFILE.PIX":PUT
    #1,M:PUT #1,WIDE:PUT #1,LLO:PUT
   #1,LHI
65 FOR I=0 TO 6 STEP 2:PUT #1,PEEK(C
   OL(I)):NEXT I:POKE PEEK(1688)+256
   *PEEK(1689),PEEK(1690)
70 POKE ICBADR+X+1,SH:POKE ICBADR+X,
   SL:POKE ICBLEN+X+1,1+INT((LINE*WI
   DE)/256):POKE ICBLEN+X,0
75 POKE ICCOM+X,11:I=USR(FMS,X):CLOS
   E #1:RETURN
80 IF ((LINE*WIDE-PIX)<WIDE) THEN RE
   TURN
85 C=USR(ADR(DELETE$)):POKE 1690,PEE
   K(PEEK(1688)+256*PEEK(1689)):POKE
    53279,4:ON SPEED GOSUB 740:RETUR
   N
90 IF ((LINE*WIDE-PIX)<WIDE) THEN RE
   TURN
95 T=SC+WIDE*LINE-WIDE-1:C=INT(T/256
   ):T=T-256*C:POKE 205,T:POKE 206,C
100 POKE (PEEK(1688)+256*PEEK(1689))
    ,PEEK(1690)
105 C=USR(ADR(EXPAND$)):POKE 1690,0:
    POKE 53279,4:ON SPEED GOSUB 740:
    RETURN
110 POKE 1690,PEEK(SC):POKE 1691,121
    :POKE 559,SCON:OPT=8
115 OPT=PEEK(53279):IF OPT=6 THEN GO
    SUB 180:GOTO 115
120 N=PEEK(632):C=USR(SCROLL,N):IF N
    <15 THEN POKE 77,0:IF SPEED THEN
     GOSUB 740:POKE 53279,4
125 IF PEEK(753)=3 THEN GOSUB 140:GO
    TO 115
130 ON OPT=3 GOTO 550:IF OPT=5 THEN
    GOSUB 60:GOTO 115
135 GOTO 115
140 GOSUB 635:ON (C=116)+2*(C=119)+3
    *(C=246) GOTO 80,90,170
145 IF N=60 THEN C=C-59:SHIF=INT(C/6
    4):GOSUB 725
150 IF C=31 OR C=30 OR C=26 OR C=50
    THEN GOSUB 720
155 IF N=33 THEN SPEED=1*(SPEED=0):G
    OSUB 715
160 IF N=39 THEN VERS=255*(VERS=0):P
    OKE 1695,VERS:GOSUB 735
165 RETURN
170 C=USR(ADR(CLEAR$),SP):POKE 1690,
    0:POKE 1691,PEEK(1696):RETURN
175 GOSUB 715:RETURN
180 GOSUB 715
185 DI=PEEK(632):T=PEEK(644):DI=DI+5
    *(DI=7):DI=DI-11:OPT=PEEK(53279)
    :IF OPT=6 THEN 175
190 IF DI<0 OR DI>3 THEN 185
195 DI=4*T+DI:IF DI/2=INT(DI/2) THEN
     POKE COL(DI),PEEK(COL(DI))-2+25
    6*(PEEK(COL(DI))<2):GOTO 185
200 POKE COL(DI),PEEK(COL(DI))+2-256
    *(PEEK(COL(DI))>253):GOTO 185
205 FLL$=FL$:FOR I=1 TO LEN(FL$):N=A
    SC(FL$(I,I)):ON N=58 GOSUB 245:N
    EXT I:FL$=FLL$
210 FLL$=FL$:FOR I=1 TO LEN(FL$):N=A
    SC(FL$(I,I)):ON N=46 GOSUB 250:N
    EXT I:FL$=FLL$
215 IF LEN(FL$)>8 THEN FL$=FL$(1,8)
220 IF LEN(FL$)<1 THEN 265
225 N=ASC(FL$(1,1)):IF N>90 OR N<65
    THEN 260
230 IF LEN(FL$)<2 THEN GOTO 240
235 FOR I=2 TO LEN(FL$):N=ASC(FL$(I,
    I)):ON (N>90 OR N<65) AND (N>57
    OR N<48) GOTO 255:NEXT I
240 FLL$="D1:":FLL$(4)=FL$:N=0:RETURN
245 FLL$=FL$(I+1,LEN(FL$)):RETURN
250 FLL$=FL$(1,I-1):RETURN
255 POP :? "{CLEAR}":? "Illegal char
    acters in ";FL$:GOTO 265
260 ? "{CLEAR}":? FL$;" must start w
    ith a capital":? "letter.":GOTO
    265
265 ? "Let's try that name again.":N
    =1:RETURN
270 TRAP 275:OPEN #1,4,0,FL$:N=0:CLO
    SE #1:RETURN
275 ? :? FL$;" isn't on disk in":? "
    drive 1":? "Insert disk with ";F
    L$;"and":? "press RETURN.":CLOSE
     #1
280 ? "Or to try another file name,
    press anyother key."
285 ON PEEK(753)<>3 GOTO 285:GOSUB 6
    35:ON N=12 GOTO 270:N=1:RETURN
290 TRAP 310:OPEN #1,4,0,FL$:? FL$;"
     is already on disk.":? "Unless
    you change the name, the old"
295 ? "file will be lost.  To change
     the namepress RETURN":? "Or pre
    ss any other key to continue.":C
    LOSE #1
300 ON PEEK(753)<>3 GOTO 300:GOSUB 6
    35:ON N=12 GOTO 305:N=0:RETURN
305 N=1:RETURN
310 CLOSE #1:N=0:RETURN
315 ? "{CLEAR}{12 SPACES}Screenbyter"
    :? :? :?
320 GOSUB 695:? :? "What file should
     hold your finished{3 SPACES}scr
    een? (Eight characters)":POKE 76
    4,255:INPUT FSAVE$
325 FL$=FSAVE$:GOSUB 205:ON N GOTO 3
    20:FSAVE$=FLL$:FSAVE$(LEN(FLL$)+
    1)=".PIX"
330 FL$=FSAVE$:GOSUB 290:ON N GOTO 3
    20
335 FLOAD$="":? :? "Would you like t
    o edit a screen you{3 SPACES}hav
    e already saved? (Y or N) "
340 GOSUB 635:ON N=35 GOTO 390:ON N=
```

```
   43 GOTO 345:GOTO 340
345 ? :? "What is the name of the sa
    ved screen  file? ":POKE 764,255
    :INPUT FLOAD$
350 FL$=FLOAD$:GOSUB 205:ON N=0 GOTO
     355:GOTO 335
355 FLOAD$=FLL$:FLOAD$(LEN(FLL$)+1)=
    ".PIX"
360 FL$=FLOAD$:GOSUB 270:ON N GOTO 3
    35:OPEN #1,4,0,FLOAD$:GET #1,MD:
    GET #1,WD:GET #1,LLO:GET #1,LHI
365 CLOSE #1:FLOAD=1:LN=LLO+256*LHI
370 ? :? FLOAD$;" was saved as:":? "
    Mode ";MD;",":? "with ";LN;" lin
    es":? "of ";WD;" characters per
    line."
375 ? "If you wish to Change these p
    arameterspress RETURN.":? "To le
    ave them unchanged press any
    {5 SPACES}other key."
380 ON PEEK(753)<>3 GOTO 380:GOSUB 6
    35:IF N=12 THEN 395
385 M=MD:M8=M-8:WIDE=WD:LINE=LN:GOTO
     445
390 FLOAD=0
395 ? :? "What Antic mode will you w
    ork in?":? :? "Antic","Graphics"
    :? 8,3:? 9,4:? "A (10)",5:? "B (
    11)",6
400 ? "C (12)",N$:? "D (13)",7:? "E
    (14)",N$:? "F (15)",8:POKE 764,2
    55
405 TRAP 405:OPEN #1,4,0,"K:":GET #1
    ,N:CLOSE #1:ON N<56 OR (N>57 AND
     N<65) OR N>70 GOTO 405
410 M=N-48:M=M-7*(M>9):M8=M-8
415 ? :? "How wide a line?":? "  (Mi
    nimum ";BPL(M8);" bytes":? "
    {3 SPACES}maximum ";MXW(M8);" by
    tes)"
420 POKE 764,255:TRAP 420:INPUT WIDE
    :WIDE=INT(WIDE):GOSUB 640:GOSUB
    745
425 ? :? "How many lines do you want
     to edit?":? "(Minimum ";LPS(M8)
    ;", Maximum ";MXL;")"
430 TRAP 430:INPUT LINE
435 LINE=INT(LINE):ON LINE<=MXL AND
    LINE>=LPS(M8) GOTO 440:LINE=MXL*
    (LINE>MXL)+LPS(M8)*(LINE<LPS(M8)
    )
440 LHI=INT(LINE/256):LLO=LINE-256*L
    HI
445 ? "{CLEAR}":? "You have chosen:"
    :? "Save file--";FSAVE$:? "Load
    file--";FLOAD$
450 ? "Mode ";M:? LINE;" lines of ";
    WIDE;" characters"
455 ? "If this is right, press START
    {9 SPACES}To make changes, press
     OPTION"
460 ON (PEEK(53279)=6)+(2*(PEEK(5327
    9)=3)) GOTO 465,315:GOTO 460
465 ? "{CLEAR}Just a minute while I
    get myself{6 SPACES}together . .
     ."
470 SC=SC+((LINE*WIDE)>K4)*(K4-INT(K
    4/WIDE)*WIDE):SH=INT(SC/256):SL=
    SC-256*SH
475 POKE 1670,WIDE-1:POKE 1674,LLO-1
    +256*(LLO=0):POKE 1675,LHI-(LLO=
    255)
480 POKE 1678,BPL(M8)-1:POKE 1680,LP
    S(M8)-1:POKE 1692,CL(3):POKE 169
    3,PPB(M8):POKE 1698,LPS(M8):POKE
     1699,WIDE
485 GOSUB 755:GOSUB 490:GOSUB 505:GO
    SUB 650:GOSUB 530:ON FLOAD GOSUB
     500:GOTO 110
490 OPEN #1,4,0,"D1:CLEARS.SUB":FOR
    I=1 TO 26:GET #1,N:CLEAR$(I,I)=C
    HR$(N):NEXT I:CLOSE #1
495 C=USR(ADR(CLEAR$),SP):RETURN
500 T=SZ:FL$=FLOAD$:GOSUB 40:SZ=T:RE
    TURN
505 DL4=DL+4:DL5=DL+5:FOR I=0 TO 2:P
    OKE DL+I,112:NEXT I:C=INT(SC/256
    ):N=SC-C*256
510 FOR I=1686 TO 1688 STEP 2:POKE I
    ,N:POKE I+1,C:NEXT I:N=0
515 FOR I=DL+3 TO DL+3*LPS(M8) STEP
    3:C=SC+N*WIDE:POKE I,64+M:T=INT(
    C/256)
520 POKE I+2,T:POKE I+1,C-256*T:N=N+
    1:NEXT I
525 POKE I,65:POKE I+1,0:POKE I+2,DL
    /256:RETURN
530 OPEN #1,4,0,"D:SCROLL.SUB":N=INT
    (SCROLL/256):C=SCROLL-256*N
535 POKE ICBADR+X+1,N:POKE ICBADR+X,
    C:POKE ICBLEN+X+1,3:POKE ICBLEN+
    X,0
540 POKE ICCOM+X,7:I=USR(FMS,X):CLOS
    E #1
545 POKE 560,0:POKE 561,DL/256:CLOSE
     #1:RETURN
550 POKE PEEK(1688)+256*PEEK(1689),P
    EEK(1690):GOSUB 60:GRAPHICS 0:PO
    KE 764,255
555 ? "Screen is saved as D1:TEMPFIL
    E.SCR":? :? "Do you want to save
     the screen as":? FSAVE$;"? (Y o
    r N)"
560 GOSUB 635:ON N<>43 AND N<>35 GOT
    O 560:IF N=43 THEN GOSUB 610:GOT
    O 570
565 FSAVE=0
570 ? :? "Do you want to quit? (Y or
     N)":POKE 764,255
575 GOSUB 635:ON N<>43 AND N<>35 GOT
    O 575:ON N=35 GOTO 580:ON N=43 G
    OTO 605
580 ? :? "To return to edit the same
     screen,{4 SPACES}press OPTION":
    ? :? "To start SCREENBYTER over,
     press START"
585 OPT=PEEK(53279):ON ((OPT=6)+(2*(
    OPT=3))) GOTO 590,595:GOTO 585
590 POKE 106,A:GRAPHICS 0:GOTO 20
595 POKE 106,TOP:FL$="D1:TEMPFILE.PI
    X":IF FSAVE=1 THEN FL$=FSAVE$
600 GOSUB 755:GOSUB 40:GOSUB 505:POK
    E 560,0:POKE 561,DL/256:GOTO 110
605 POKE 106,A:POKE 764,255:GRAPHICS
     0:END
610 FSAVE=1:TRAP 615:OPEN #2,4,0,FSA
    VE$:CLOSE #2:XIO 36,#2,0,0,FSAVE
    $:XIO 33,#2,0,0,FSAVE$:GOTO 620
615 CLOSE #2
620 FL$="D1:TEMPFILE.PIX,":FLL$=FSAV
    E$(4,LEN(FSAVE$)):FL$(17)=FLL$
625 XIO 32,#1,0,0,FL$:RETURN
```

```
630 ON PEEK(753)<>3 GOTO 63Ø:RETURN
635 C=PEEK(764):N=C-64*INT(C/64):RET
    URN
640 IF WIDE>=BPL(M8) AND WIDE<=MXW(M
    8) THEN RETURN
645 WIDE=MXW(M8)*(WIDE>MXW(M8))+BPL(
    M8)*(WIDE<BPL(M8)):RETURN
650 OPEN #1,4,Ø,"D:DELETES.SUB":FOR
    I=1 TO 118:GET #1,N:DELETE$(I,I)
    =CHR$(N):NEXT I:CLOSE #1
665 OPEN #1,4,Ø,"D:EXPANDS.SUB":FOR
    I=1 TO 1Ø2:GET #1,N:EXPAND$(I,I)
    =CHR$(N):NEXT I:CLOSE #1
680 OPEN #1,4,Ø,"D:FILL.SUB":FOR I=1
     TO 23Ø:GET #1,N:FILL$(I,I)=CHR$
    (N):NEXT I
690 CLOSE #1:C=ADR(FILL$):N=INT(C/25
    6):C=C-N*256:POKE 17Ø2,76:POKE 1
    7Ø3,C:POKE 17Ø4,N:RETURN
695 TRAP 71Ø:XIO 36,#1,Ø,Ø,"D1:*.PIX
    "
7ØØ ? :? "Currently saved screen fil
    es:"
7Ø5 FL$="D1:*.PIX":OPEN #1,6,Ø,FL$:F
    OR I=Ø TO 5Ø:INPUT #1,FLL$:? FLL
    $:NEXT I
71Ø CLOSE #1:RETURN
715 FOR I=Ø TO 1Ø:POKE 53279,4:NEXT
    I:RETURN
72Ø SHIF=(C=31)+2*(C=3Ø)+3*(C=26)
725 POKE 53279,4:POKE 1692,CL(SHIF):
    IF PPB(M8)=1 AND SHIF>Ø THEN SHI
    F=3:POKE 1692,CL(SHIF)
73Ø RETURN
735 N=(VERS=255):SOUND Ø,2ØØ*N,14*N,
    4*N:RETURN
74Ø FOR I=Ø TO 1Ø:NEXT I:RETURN
745 IF BPL(M8)=MXW(M8) THEN MXL=LPS(
    M8):RETURN
75Ø MXL=INT(K4/WIDE):RETURN
755 FOR I=1677 TO 1681 STEP 2:POKE I
    ,Ø:NEXT I:FOR I=1686 TO 1688 STE
    P 2:POKE I,SL:POKE I+1,SH:NEXT I
76Ø N=128+64*(PPB(M8)=2):POKE 1696,N
    :POKE 17Ø1,N
765 POKE 1672,Ø:POKE 1676,Ø:VERS=Ø:G
    OSUB 735:POKE 1695,VERS:RETURN
77Ø DATA 2,1Ø,17Ø,24,1,1Ø,85,48,2,2Ø
    ,85,48,1,2Ø,42,96
775 DATA 1,2Ø,21,192,2,4Ø,42,96,2,4Ø
    ,4Ø,192,1,4Ø,4Ø,192
```

Program 4: Insert Line Routine

```
9ØØ OPEN #1,8,Ø,"D1:EXPANDS.SUB"
91Ø FOR I=1 TO 1Ø2:READ N:PUT #1,N:N
    EXT I:CLOSE #1:? I:END
1ØØØ DATA 1Ø4,56,165,2Ø5,237,163,6,1
     33
1ØØ8 DATA 2Ø3,165,2Ø6,233,Ø,133,2Ø4,
     56
1Ø16 DATA 173,138,6,237,14Ø,6,133,2Ø
     7
1Ø24 DATA 173,139,6,237,141,6,133,2Ø
     8
1Ø32 DATA 165,2Ø8,24Ø,5,162,255,24,1
     44
1Ø4Ø DATA 2,166,2Ø7,172,163,6,177,2Ø
     3
1Ø48 DATA 145,2Ø5,136,2Ø8,249,2Ø2,24
     Ø,31
1Ø56 DATA 56,165,2Ø5,237,163,6,133,2
     Ø5
1Ø64 DATA 165,2Ø6,233,Ø,133,2Ø6,56,1
     65
1Ø72 DATA 2Ø3,237,163,6,133,2Ø3,165,
     2Ø4
1Ø8Ø DATA 233,Ø,133,2Ø4,24,144,212,1
     65
1Ø88 DATA 2Ø8,2Ø8,2Ø6,172,163,6,169,
     Ø
1Ø96 DATA 145,2Ø3,136,2Ø8,251,96
```

Program 5: Delete Line Routine

```
9ØØ OPEN #1,8,Ø,"D1:DELETES.SUB"
91Ø FOR I=1 TO 118:READ N:PUT #1,N:N
    EXT I:CLOSE #1:? I:END
1ØØØ DATA 1Ø4,56,173,152,6,237,136,6
1ØØ8 DATA 133,2Ø3,173,153,6,233,Ø,13
     3
1Ø16 DATA 2Ø4,24,165,2Ø3,1Ø9,163,6,1
     33
1Ø24 DATA 2Ø5,165,2Ø4,1Ø5,Ø,133,2Ø6,
     56
1Ø32 DATA 173,138,6,237,14Ø,6,133,2Ø
     7
1Ø4Ø DATA 173,139,6,237,141,6,133,2Ø
     8
1Ø48 DATA 165,2Ø8,24Ø,5,162,255,24,1
     44
1Ø56 DATA 2,166,2Ø7,172,163,6,177,2Ø
     5
1Ø64 DATA 145,2Ø3,136,2Ø8,249,2Ø2,24
     Ø,31
1Ø72 DATA 24,165,2Ø5,1Ø9,163,6,133,2
     Ø5
1Ø8Ø DATA 165,2Ø6,1Ø5,Ø,133,2Ø6,24,1
     65
1Ø88 DATA 2Ø3,1Ø9,163,6,133,2Ø3,165,
     2Ø4
1Ø96 DATA 1Ø5,Ø,133,2Ø4,24,144,212,1
     65
11Ø4 DATA 2Ø8,2Ø8,2Ø6,172,163,6,169,
     Ø
1112 DATA 145,2Ø5,136,2Ø8,251,96
```

Program 6: Cursor Movement Routine

```
9ØØ OPEN #1,8,Ø,"D1:SCROLL.SUB"
91Ø FOR I=1 TO 65Ø:READ N:PUT #1,N:N
    EXT I:CLOSE #1:? I:END
1ØØØ DATA 1Ø4,1Ø4,1Ø4,141,161,6,173,
     152
1ØØ8 DATA 6,133,2Ø7,173,153,6,133,2Ø
     8
1Ø16 DATA 16Ø,Ø,14Ø,158,6,173,154,6
1Ø24 DATA 145,2Ø7,173,161,6,41,8,24Ø
1Ø32 DATA 92,173,161,6,41,4,2Ø8,71
1Ø4Ø DATA 172,157,6,173,16Ø,6,42,176
1Ø48 DATA 8,136,2Ø8,25Ø,141,16Ø,6,24
     Ø
1Ø56 DATA 54,42,136,2Ø8,252,141,148,
     6
1Ø64 DATA 173,136,6,2Ø8,2,24Ø,4Ø,173
1Ø72 DATA 148,6,141,16Ø,6,56,173,136
1Ø8Ø DATA 6,233,1,141,136,6,56,173
1Ø88 DATA 152,6,233,1,141,152,6,173
1Ø96 DATA 153,6,233,Ø,141,153,6,173
11Ø4 DATA 143,6,24Ø,6,2Ø6,143,6,24
1112 DATA 144,99,173,158,6,9,8,141
112Ø DATA 158,6,24,144,88,172,157,6
```

```
1128 DATA 173,160,6,106,176,8,136,20
     8
1136 DATA 250,141,160,6,240,71,106,1
     36
1144 DATA 208,252,141,148,6,173,136,
     6
1152 DATA 205,134,6,208,2,240,54,173
1160 DATA 148,6,141,160,6,24,173,136
1168 DATA 6,105,1,141,136,6,24,173
1176 DATA 152,6,105,1,141,152,6,173
1184 DATA 153,6,105,0,141,153,6,173
1192 DATA 143,6,205,142,6,240,6,238
1200 DATA 143,6,24,144,8,173,158,6
1208 DATA 9,4,141,158,6,173,161,6
1216 DATA 41,1,240,83,173,161,6,41
1224 DATA 2,208,62,173,140,6,205,138
1232 DATA 6,208,8,173,141,6,205,139
1240 DATA 6,240,124,24,173,140,6,105
1248 DATA 1,141,140,6,173,141,6,105
1256 DATA 0,141,141,6,24,173,152,6
1264 DATA 109,163,6,141,152,6,173,15
     3
1272 DATA 6,105,0,141,153,6,173,145
1280 DATA 6,205,144,6,240,6,238,145
1288 DATA 6,24,144,75,173,158,6,9
1296 DATA 1,141,158,6,24,144,64,173
1304 DATA 140,6,208,5,173,141,6,240
1312 DATA 54,56,173,140,6,233,1,141
1320 DATA 140,6,173,141,6,233,0,141
1328 DATA 141,6,56,173,152,6,237,163
1336 DATA 6,141,152,6,173,153,6,233
1344 DATA 0,141,153,6,173,145,6,240
1352 DATA 6,206,145,6,24,144,8,173
1360 DATA 158,6,9,2,141,158,6,173
1368 DATA 152,6,133,207,173,153,6,13
     3
1376 DATA 208,173,132,2,240,36,160,0
1384 DATA 177,207,141,154,6,73,255,4
     5
1392 DATA 160,6,141,155,6,173,160,6
1400 DATA 73,255,45,154,6,13,155,6
1408 DATA 141,155,6,173,158,6,240,40
1416 DATA 208,41,160,0,177,207,141,1
     55
1424 DATA 6,173,156,6,45,160,6,141
1432 DATA 161,6,173,160,6,73,255,45
1440 DATA 155,6,141,155,6,13,161,6
1448 DATA 141,154,6,173,158,6,208,3
1456 DATA 24,144,98,41,8,240,17,56
1464 DATA 173,150,6,233,1,141,150,6
1472 DATA 173,151,6,233,0,141,151,6
1480 DATA 173,158,6,41,4,240,17,24
1488 DATA 173,150,6,105,1,141,150,6
1496 DATA 173,151,6,105,0,141,151,6
1504 DATA 173,158,6,41,1,240,18,24
1512 DATA 173,150,6,109,163,6,141,15
     0
1520 DATA 6,173,151,6,105,0,141,151
1528 DATA 6,173,158,6,41,2,240,24
1536 DATA 56,173,150,6,237,163,6,141
1544 DATA 150,6,173,151,6,233,0,141
1552 DATA 151,6,24,144,3,24,144,67
1560 DATA 173,150,6,133,203,173,151,
     6
1568 DATA 133,204,24,173,48,2,105,4
1576 DATA 133,205,173,49,2,133,206,1
     74
1584 DATA 162,6,160,0,165,203,145,20
     5
1592 DATA 200,165,204,145,205,24,165
     ,205
1600 DATA 105,3,133,205,165,206,105,
     0
1608 DATA 133,206,24,165,203,109,163
     ,6
1616 DATA 133,203,165,204,105,0,133,
     204
1624 DATA 202,208,215,173,155,6,160,
     0
1632 DATA 145,207,173,159,6,201,255,
     208
1640 DATA 8,173,132,2,208,3,32,166
1648 DATA 6,96
```

Program 7: Clear Screen Routine

```
900 OPEN #1,8,0,"D1:CLEARS.SUB"
910 FOR I=1 TO 26:READ N:PUT #1,N:NE
    XT I:CLOSE #1:? I:END
1000 DATA 104,104,104,133,208,162,32
     ,169
1008 DATA 0,133,207,160,255,145,207,
     136
1016 DATA 208,251,145,207,230,208,20
     2,208
1024 DATA 238,96
```

Program 8: Fill Subroutine

```
900 OPEN #1,8,0,"D1:FILL.SUB"
910 FOR I=1 TO 230:READ N:PUT #1,N:N
    EXT I:CLOSE #1:? I:END
1000 DATA 173,136,6,141,135,6,173,15
     4
1008 DATA 6,141,146,6,165,207,133,20
     3
1016 DATA 165,208,133,204,162,0,173,
     160
1024 DATA 6,141,148,6,172,157,6,78
1032 DATA 148,6,176,52,136,208,248,1
     73
1040 DATA 146,6,45,148,6,141,149,6
1048 DATA 173,156,6,45,148,6,205,149
1056 DATA 6,240,20,141,149,6,173,148
1064 DATA 6,73,255,45,146,6,13,149
1072 DATA 6,141,146,6,24,144,205,173
1080 DATA 146,6,129,207,141,154,6,96
1088 DATA 173,135,6,205,134,6,240,24
     7
1096 DATA 238,135,6,24,165,203,105,1
1104 DATA 133,203,165,204,105,0,133,
     204
1112 DATA 161,203,141,148,6,173,165,
     6
1120 DATA 141,164,6,173,164,6,45,148
1128 DATA 6,141,149,6,173,164,6,45
1136 DATA 156,6,205,149,6,240,13,172
1144 DATA 157,6,78,164,6,176,193,136
1152 DATA 208,248,240,223,172,157,6,
     14
1160 DATA 164,6,176,29,136,208,248,1
     73
1168 DATA 164,6,45,156,6,141,149,6
1176 DATA 173,164,6,73,255,45,148,6
1184 DATA 13,149,6,141,148,6,24,144
1192 DATA 219,162,0,173,148,6,129,20
     3
1200 DATA 173,146,6,129,207,141,154,
     6
1208 DATA 56,173,135,6,237,136,6,240
1216 DATA 12,168,136,240,8,173,156,6
1224 DATA 145,207,136,208,251,96
```

©

Disk Explorer For Commodore

Robert W. Baker

If you've ever been curious about the 1541's memory, this program gives you an inside view of the unit's ROMs. It allows you to display both a disassembly of the 1541's machine language instructions and a hex dump of the drive's RAM and ROM addresses.

"Disk Explorer," a program written for the 64 but suitable for other Commodore users, is designed to let you look around inside the VIC-1541 disk controller. You can directly display a disassembly of the machine language instructions in the disk unit's ROMs. Alternately, you can display a hexadecimal dump of any area of the disk controller 6502 microprocessor's address space, including peripheral chips, RAM, or ROM. With some knowledge about assembly language and a little about hardware, this program provides an easy method of exploring the disk controller.

A Variety Of Choices

When the program starts, there's a short delay while a data array is built for the disassembler (lines 110 – 130). Then you're prompted for the starting address of where you'd like to start looking. The desired address can be entered as a decimal number, or a hexadecimal number preceded by a dollar sign. Program lines 160-240 validate the digits of the address and convert a hex address to a decimal value. An invalid address is discarded and you're prompted again for the starting address.

The program normally displays the data on the screen, but you can select printed output as shown in lines 250 – 270. You'll notice the OPEN statement in line 270 opens either device 3 or 4 depending on whether a printed output is desired. Device 3 is the display screen, and device 4 is the printer. This provides a simple switch between devices for all following PRINT#4 statements without having separate routines for display and printed data. You can still force output to the display screen by using the simple PRINT statement.

The last prompt is for the data display type: either a hexadecimal dump or an instruction disassembly. If a hex dump is selected, then eight bytes of data are displayed, in hex, per screen line. Each line also includes the hex address and the ASCII translation of the data displayed. The ASCII translation is simply the displayable character for each byte shown, with nondisplayable characters converted to periods.

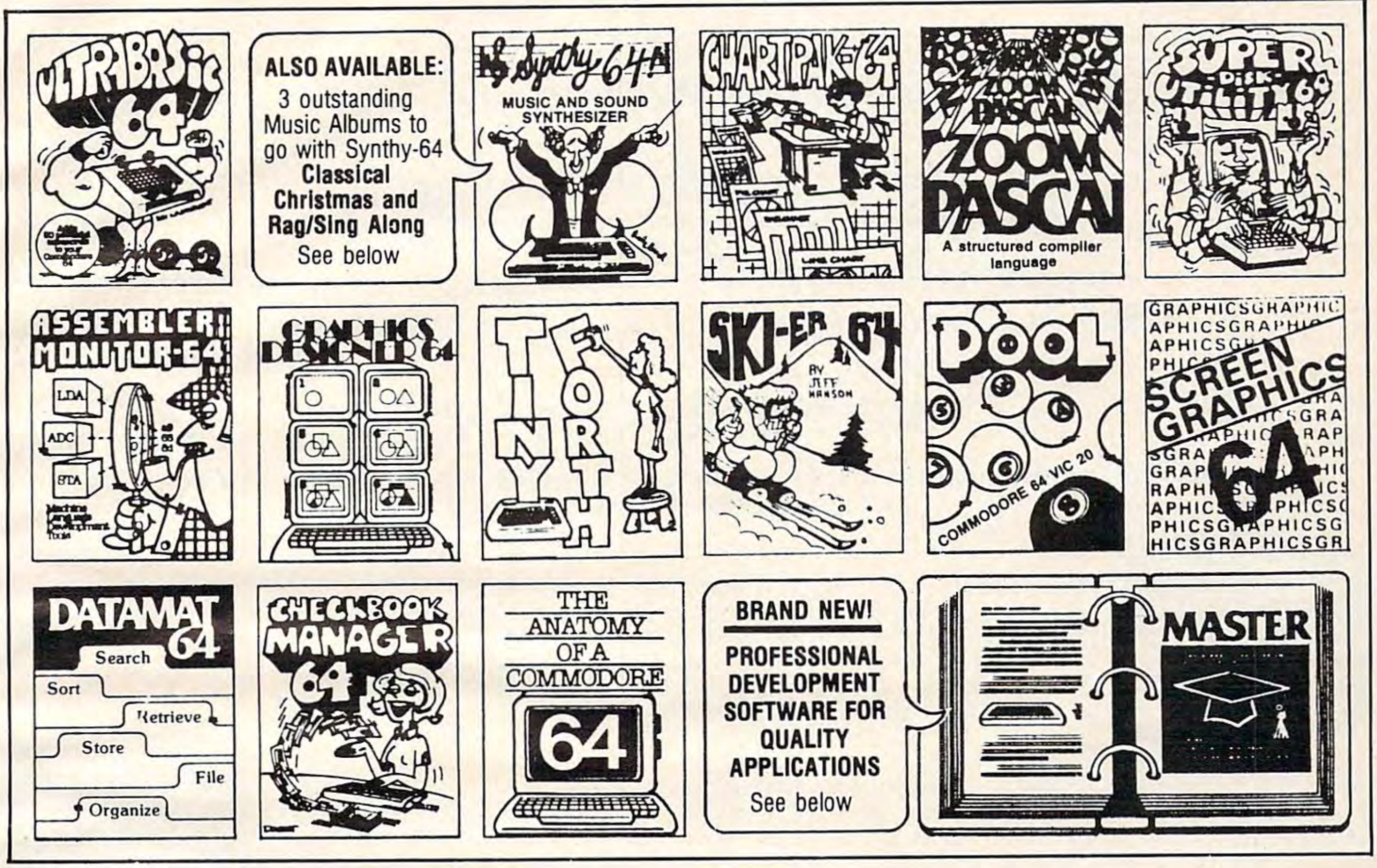
An instruction disassembly shows one 6502 instruction per line using the standard mnemonics. Each line indicates the address of the instruction in both decimal and hex, along with the hex opcode for the instruction displayed. To make things a little easier to read, branch instructions indicate the hex address to which the instruction would branch rather than an offset from the current location.

Three Choices

When displaying data on the screen, the program will pause after 16 lines of hex data or 20 disassembled instructions. A prompt message will ask whether you want to: continue displaying data with the next sequential location; restart the display with a new address and/or format; or stop the program and return to BASIC.

When data is being printed, pressing any key

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



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

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

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

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

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

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

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

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

SKIER-64...This arcade-quality game adds hours of action and excitement to your Commodore-64. **TAPE $14.95 DISK $17.95**

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

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

DATAMAT-64...Simple powerful data base management with search, sort, report capability at low price. **DISK $32.95**

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

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

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

✱ DEALER INQUIRIES INVITED

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

DISTRIBUTORS

Great Britain: ADAMSOFT, 18 Norwich Ave., Rochdale, Lancs.

Great Britain CCI Software, 167 Great Portland St., London WI, 01-636-6354

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

Canada West: L.S.I. Distributors Ltd., 810 W Broadway #163, Vancouver, BC V5Z 4C9, 604/733-0211

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

Sweden: TIAL TRADING, PO 516, 34300 Almhult, 476-12304

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

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

AVAILABLE AT COMPUTER STORES, OR WRITE:

Abacus Software

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

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

FOR QUICK SERVICE PHONE 616-241-5510

EPYX
TOP 10 GAME SALE
* BUY TWO — GET ONE FREE *

SPECIAL FREE GAME
Buy Two Epyx Games From Protecto And Get A Free Game

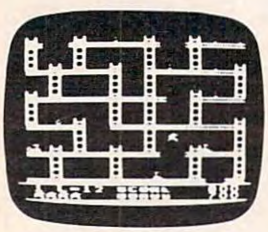
The Thinking Man's Paradise

(Disk/Cassette)

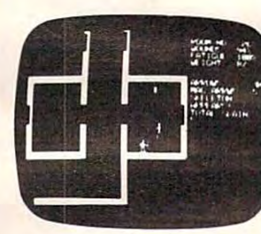
- Award Winning Games
- Fantastic Graphics
- Skill (not luck) Needed
- Already in top 10 charts

COMMODORE - 64 / VIC-20

Buy Any Two Epyx Games From Protecto And Send The Proof Of Purchase Seals To Epyx And Epyx Will Send You A Free Game.



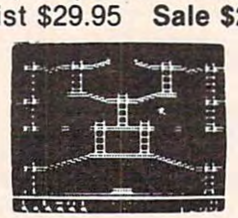
JUMPMAN
If you like Donkey Kong, you'll love Jumpman. Over 30 different screens with 8 speeds and 5 skill levels make this the fastest action game in the country. You must leap girders, climb ropes, and scale ladders to reach and diffuse bombs while avoiding robots, birds, bullets, explosives, crumbling walls, vanishing escape routes, and many other obstacles. (Truly a fantastic game!)
List $39.95 **Sale $27.95** (DISK ONLY)



TEMPLE OF APSHAI (computer game of the year)
This is the standard by which other adventure games are judged. Full color graphics portray the temple and all its contents - magic, monsters, doomed cities and damsels in distress. Do battle in real time with over 20 monsters, expansion modules will keep your adventure alive in the future.
List $39.95 **Sale $29.95**

SWORD OF FARGOAL
Search for the wondrous sword in the depths of an ever changing dungeon. Make yourself invisible, teleport to a new location, drink a healing potion or use enchanted treasures, but watch out for traps and hideous creatures who will try to stop you. (Fantastic dungeon adventure)
List $29.95 **Sale $21.95**

CRUSH CRUMBLE & CHOMP (Computer game of the year nominee)
Choose one of six monsters or create your own, and use your monster to destroy one of four unsuspecting cities. The cities aren't totally defenseless, they call on police, national guard, and even a mad scientist, complete with helicopter, to save humanity from the relentless threat.
List $29.95 **Sale $21.95**



JUMPMAN JUNIOR
The devilish ALIENATORS are back! And they have overrun the Jupiter Command Substation. In this cartridge format sequel to the best-selling Jumpman, players must leap through 12 all new screens featuring electrocution traps, moving walls, hellstones, and dangers much too bizarre to be believed. How many screens can you master? Twelve different screens, 8 speeds.
List $39.95 **Sale $27.95**

We have all Epyx games

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