```
34054 :165,141,102,003,169,160,234
34060 :141,078,138,032,068,166,123
34066 :169,012,141,064,003,169,064
34072 :011,141,060,003,169,010,162
34078 :141,061,003,169,013,141,046
34084 :062,003,169,014,141,063,232
34090 :003,032,117,137,032,091,198
34096 :137,024,162,255,160,127,145
34102 :032,153,255,032,191,227,176
34108 :120,169,019,141,004,003,004
34114 :169,136,141,005,003,169,177
34120 :201,141,006,003,169,135,215
34126 :141,007,003,169,163,141,190
34132 :008,003,169,135,141,009,037
34138 :003,169,228,141,024,003,146
34144 :169,133,141,025,003,169,224
34150 :001,141,022,003,169,134,060
34156 :141,023,003,169,074,141,147
34162 :143,002,169,153,141,144,098
34168 :002,088,160,000,185,138,181
34174 :133,032,210,255,200,192,124
34180 :090,208,245,076,116,164,007
34186 :147,153,018,032,032,032,040
34192 :051,045,068,073,077,069,015
34198 :078,083,073,079,078,032,061
34204 :080,073,067,084,085,082,115
34210 :069,032,065,078,065,076,035
34216 :089,083,073,083,032,032,048
34222 :032,032,032,032,032,032,110
34228 :032,146,154,153,018,032,203
34234 :032,032,032,032,032,032,122
34240 :032,032,032,066,089,032,219
34246 :067,072,082,073,083,084,147
34252 :073,065,078,032,077,046,063
34258 :032,080,065,078,078,069,100
34264 :084,079,078,032,032,032,041
34270 :032,032,032,146,154,013,119
34276 :072,138,072,152,072,169,135
34282 :127,141,013,221,172,013,153
34288 :221,016,003,076,114,254,156
34294 :032,188,246,032,225,255,200
34300 :240,003,076,114,254,165,080
34306 :055,133,001,032,163,253,127
34312 :032,024,229,032,018,133,220
34318 :108,002,160,230,122,208,076
34324 :002,230,123,096,032,017,008
34330 :134,032,138,173,032,191,214
34336 :177,032,121,000,164,100,114
34342 :165,101,096,162,000,165,215
34348 :100,016,015,162,001,056,138
34354 :169,000,229,101,133,101,015
34360 :169,000,229,100,133,100,019
34366 :096,032,027,134,032,041,168
34372 :134,142,092,003,165,100,192
34378 :141,091,003,165,101,141,204
34384 :090,003,032,024,134,032,139
34390 :041,134,142,095,003,165,154
34396 :100,141,094,003,165,101,184
34402 :141,093,003,032,024,134,013
34408 :032,041,134,142,098,003,042
34414 :165,100,141,097,003,165,013
34420 :101,141,096,003,076,000,021
34426 :144,032,063,134,165,038,186
34432 :133,251,133,025,165,040,107
34438 :133,252,133,026,032,024,222
34444 :134,165,101,141,065,003,237
34450 :076,108,138,032,063,134,185
34456 :165,038,133,251,165,040,176
34462 :133,252,032,017,134,032,246
34468 :063,134,165,038,133,253,182
34474 :165,040,133,254,032,024,050
34480 :134,141,065,003,076,242,069
34486 :138,032,027,134,141,099,241
34492 :003,032,024,134,141,100,110
34498 :003,032,024,134,141,101,117
34504 :003,032,024,134,141,102,124
34510 :003,096,032,027,134,208,194
34516 :003,076,091,137,201,001,209
34522 :208,011,169,224,141,078,025
34528 :138,032,117,137,076,039,251
34534 :137,169,160,141,078,138,029
34540 :032,117,137,076,065,137,032
34546 :032,027,134,201,001,208,077
34552 :011,032,039,137,169,160,028
34558 :141,078,138,076,193,137,249
34564 :032,065,137,169,224,141,004
34570 :078,138,076,189,137,032,148
34576 :027,134,141,064,003,032,161
34582 :024,134,141,060,003,032,160
34588 :024,134,141,061,003,032,167
34594 :024,134,141,062,003,032,174
34600 :024,134,141,063,003,076,225
34606 :117,137,032,027,134,141,122
34612 :084,003,140,085,003,032,143
34618 :024,134,141,086,003,140,074
34624 :087,003,032,024,134,141,229
34630 :088,003,140,089,003,076,213
34636 :101,141,032,027,134,140,139
34642 :156,003,141,155,003,032,060
34648 :024,134,140,158,003,141,176
34654 :157,003,032,024,134,140,072
34660 :160,003,141,159,003,032,086
34666 :024,134,140,162,003,141,198
34672 :161,003,032,024,134,140,094
34678 :085,003,141,084,003,032,210
34684 :024,134,140,087,003,141,141
34690 :086,003,032,024,134,140,037
34696 :089,003,141,088,003,032,236
34702 :024,134,141,065,003,076,073
34708 :251,145,032,027,134,201,170
34714 :001,208,003,076,189,137,000
34720 :076,193,137,032,115,000,201
34726 :201,204,144,025,201,217,134
34732 :176,021,032,180,135,076,024
34738 :174,167,233,203,010,168,109
34744 :185,014,137,072,185,013,022
34750 :137,072,076,115,000,032,110
34756 :121,000,076,231,167,016,039
34762 :066,201,255,240,062,036,038
34768 :015,048,058,170,132,073,192
34774 :201,204,176,010,160,160,101
34780 :132,035,160,158,132,034,103
34786 :208,011,233,076,170,160,060
34792 :136,132,035,160,199,132,002
34798 :034,160,000,010,240,016,186
34804 :202,016,012,230,034,208,178
34810 :002,230,035,177,034,016,232
34816 :246,048,241,200,177,034,178
34822 :048,008,032,071,171,208,032
34828 :246,076,243,166,076,239,034
34834 :166,166,122,160,004,132,000
34840 :015,189,000,002,016,007,253
34846 :201,255,240,062,232,208,204
34852 :244,201,032,240,055,133,173
34858 :008,201,034,240,086,036,135
34864 :015,112,045,201,063,208,180
34870 :004,169,153,208,037,201,058
34876 :048,144,004,201,060,144,149
34882 :029,132,113,160,000,132,120
34888 :011,136,134,122,202,200,109
34894 :232,189,000,002,056,249,038
34900 :158,160,240,245,201,128,192
34906 :208,048,005,011,164,113,127
34912 :232,200,153,251,001,185,094
34918 :251,001,240,089,056,233,204
34924 :058,240,004,201,073,208,124
34930 :002,133,015,056,233,085,126
34936 :208,159,133,008,189,000,049
34942 :002,240,223,197,008,240,012
34948 :219,200,153,251,001,232,164
34954 :208,240,166,122,230,011,091
34960 :200,185,157,160,016,250,088
34966 :185,158,160,208,180,160,177
34972 :255,202,200,232,189,000,210
34978 :002,056,249,199,136,240,020
34984 :245,201,128,208,002,240,168
34990 :173,166,122,230,011,200,052
34996 :185,198,136,016,250,185,126
35002 :199,136,208,226,189,000,120
35008 :002,016,155,076,009,166,104
35014 :032,068,080,076,079,212,233
35020 :068,076,073,078,197,068,252
35026 :068,082,065,215,080,065,017
35032 :082,065,205,083,067,082,032
35038 :069,069,206,065,078,073,014
35044 :205,067,079,076,079,210,176
35050 :065,078,071,076,197,068,021
35056 :067,073,082,067,076,197,034
35062 :068,067,076,069,065,210,033
35068 :070,083,069,212,070,073,061
35074 :071,085,082,197,070,067,062
35080 :076,069,065,210,000,122,038
35086 :134,148,134,162,134,182,140
35092 :134,207,134,241,134,014,116
35098 :135,047,135,077,135,149,192
35104 :135,078,151,057,152,007,100
35110 :153,173,000,221,041,248,106
35116 :009,004,141,000,221,169,076
35122 :057,141,024,208,169,059,196
35128 :141,017,208,169,024,141,244
35134 :022,208,096,173,000,221,014
35140 :041,248,009,005,141,000,000
35146 :221,169,009,141,024,208,078
35152 :169,059,141,017,208,169,075
35158 :024,141,022,208,096,173,238
35164 :000,221,041,248,009,007,106
35170 :141,000,221,169,021,141,023
35176 :024,208,169,027,141,017,178
35182 :208,169,008,141,022,208,098
35188 :096,173,064,003,141,032,113
35194 :208,173,060,003,141,033,228
35200 :208,173,061,003,010,010,081
35206 :010,010,013,062,003,160,136
35212 :250,153,255,203,153,249,123
35218 :204,153,243,205,153,237,061
35224 :206,153,255,127,153,249,015
35230 :128,153,243,129,153,237,177
35236 :130,136,208,229,173,063,079
35242 :003,160,250,153,255,215,182
35248 :153,249,216,153,243,217,127
35254 :153,237,218,136,208,241,095
35260 :096,169,224,208,002,169,032
35266 :160,133,088,169,224,133,077
35272 :089,169,000,133,087,168,078
35278 :145,087,200,208,251,230.047
35284 :088,230,089,208,245,096,144
35290 :165,026,074,074,074,133,252
35296 :087,010,010,024,101,087,031
35302 :133,087,074,074,133,088,051
35308 :165,087,010,010,010,010,016
35314 :010,010,133,087,169,007,146
35320 :037,026,024,101,087,133,144
35326 :087,096,165,025,074,074,007
35332 :133,089,074,074,074,074,010
35338 :074,133,090,165,089,010,059
35344 :010,010,133,089,169,003,174
35350 :037,025,170,169,000,133,044
35356 :091,173,065,003,240,024,112
35362 :106,144,005,189,104,138,208
35368 :133,091,173,065,003,106,099
35374 :106,144,009,024,189,104,110
35380 :138,010,101,091,133,091,104
35386 :096,152,072,120,169,048,203
35392 :133,001,024,165,089,101,065
35398 :087,133,093,165,090,101,227
35404 :088,105,224,133,094,160,112
35410 :000,177,093,061,100,138,139
35416 :005,091,145,093,169,055,134
35422 :133,001,088,104,168,096,172
35428 :063,207,243,252,064,016,177
35434 :004,001,032,218,137,032,018
35440 :000,138,076,059,138,198,209
35446 :028,056,165,101,229,099,028
35452 :133,101,165,028,229,100,112
35458 :133,028,032,000,138,164,113
35464 :026,200,136,132,026,032,176
35470 :218,137,032,059,138,196,154
35476 :028,208,243,096,032,000,243
35482 :138,164,026,136,200,132,182
35488 :026,032,218,137,032,059,152
35494 :138,196,028,208,243,096,051
35500 :230,027,024,165,101,101,052
35506 :099,133,101,165,027,101,036
35512 :100,133,027,032,218,137,063
35518 :164,025,136,200,132,025,104
35524 :032,000,138,032,059,138,083
35530 :196,027,208,243,096,198,146
35536 :027,056,165,101,229,099,117
35542 :133,101,165,027,229,100,201
35548 :133,027,032,218,137,164,163
35554 :025,200,136,132,025,032,008
35560 :000,138,032,059,138,196,027
35566 :027,208,243,096,165,251,204
35572 :133,025,165,252,133,026,210
35578 :197,254,144,012,032,039,160
35584 :139,165,253,133,251,165,082
35590 :254,133,252,096,165,253,135
35596 :133,251,165,025,133,253,204
35602 :165,251,133,025,165,254,243
35608 :133,252,165,026,133,254,219
35614 :165,252,133,026,056,032,182
35620 :039,139,096,229,254,133,158
35626 :102,165,251,197,253,144,130
35632 :060,229,253,197,102,144,009
35638 :027,133,098,165,102,133,200
35644 :097,032,174,139,165,251,150
35650 :133,027,230,027,032,207,210
35656 :138,198,026,165,027,197,055
35662 :253,208,245,096,133,097,086
35668 :165,102,133,098,032,174,020
35674 :139,165,252,133,028,230,013
35680 :028,032,117,138,198,025,122
35686 :165,028,197,254,208,245,175
```

```
35692 :Ø96,Ø56,165,253,229,251,134
35698 :197,1Ø2,144,Ø29,133,Ø98,Ø49
357Ø4 :165,1Ø2,133,Ø97,Ø32,174,Ø55
3571Ø :139,23Ø,Ø99,165,251,133,119
35716 :Ø27,198,Ø27,Ø32,172,138,214
35722 :198,Ø26,165,Ø27,197,253,236
35728 :2Ø8,245,Ø96,133,Ø97,165,Ø64
35734 :1Ø2,133,Ø98,Ø32,174,139,Ø6Ø
3574Ø :165,252,133,Ø28,23Ø,Ø28,224
35746 :Ø32,117,138,23Ø,Ø25,165,1Ø1
35752 :Ø28,197,254,2Ø8,245,Ø96,172
35758 :23Ø,Ø97,23Ø,Ø98,162,ØØ8,231
35764 :165,Ø98,133,1ØØ,169,ØØØ,Ø77
3577Ø :ØØ6,1ØØ,Ø42,197,Ø97,144,ØØ4
35776 :ØØ4,229,Ø97,23Ø,1ØØ,2Ø2,Ø3Ø
35782 :2Ø8,242,133,Ø98,162,ØØ8,Ø25
35788 :169,ØØØ,133,Ø99,165,Ø98,1ØØ
35794 :ØØ6,Ø99,Ø42,197,Ø97,144,Ø27
358ØØ :ØØ4,229,Ø97,23Ø,Ø99,2Ø2,Ø53
358Ø6 :2Ø8,242,198,1ØØ,169,ØØØ,115
35812 :133,1Ø1,Ø96,ØØØ,ØØØ,ØØ1,Ø47
35818 :Ø3Ø,ØØ2,Ø6Ø,ØØ3,Ø89,ØØ4,166
35824 :119,ØØ5,148,ØØ6,177,ØØ7,19Ø
3583Ø :2Ø5,ØØ8,232,Ø1Ø,ØØ3,Ø11,2Ø3
35836 :Ø29,Ø12,Ø54,Ø13,Ø78,Ø14,196
35842 :1Ø2,Ø15,124,Ø16,144,Ø17,164
35848 :164,Ø18,182,Ø19,199,Ø2Ø,Ø98
35854 :214,Ø21,228,Ø22,24Ø,Ø23,25Ø
3586Ø :25Ø,Ø25,ØØ2,Ø26,ØØ8,Ø27,1Ø2
35866 :Ø12,Ø28,Ø14,Ø29,Ø14,Ø3Ø,153
35872 :Ø12,Ø31,ØØ7,Ø32,ØØØ,Ø32,146
35878 :246,Ø33,234,Ø34,219,Ø35,Ø71
35884 :2Ø2,Ø36,181,Ø37,158,Ø38,184
3589Ø :132,Ø39,1Ø3,Ø4Ø,Ø71,Ø41,22Ø
35896 :Ø35,Ø41,253,Ø42,211,Ø43,169
359Ø2 :166,Ø44,117,Ø45,Ø65,Ø46,Ø33
359Ø8 :Ø1Ø,Ø46,2Ø6,Ø47,144,Ø48,Ø57
35914 :212,Ø49,ØØ7,Ø49,189,Ø5Ø,118
3592Ø :111,Ø51,Ø29,Ø51,199,Ø52,Ø61
35926 :1Ø9,Ø53,Ø15,Ø53,173,Ø54,Ø31
35932 :Ø7Ø,Ø54,22Ø,Ø55,1Ø9,Ø55,143
35938 :25Ø,Ø56,13Ø,Ø57,ØØ6,Ø57,142
35944 :134,Ø58,ØØ1,Ø58,12Ø,Ø58,Ø21
3595Ø :234,Ø59,Ø87,Ø59,192,Ø6Ø,Ø33
35956 :Ø36,Ø6Ø,131,Ø6Ø,222,Ø61,174
35962 :Ø52,Ø61,133,Ø61,21Ø,Ø62,189
35968 :Ø25,Ø62,Ø92,Ø62,154,Ø62,Ø73
35974 :211,Ø63,ØØ7,Ø63,Ø54,Ø63,Ø83
3598Ø :Ø97,Ø63,134,Ø63,166,Ø63,214
35986 :194,Ø63,216,Ø63,234,Ø63,211
35992 :246,Ø63,254,Ø64,ØØØ,239,25Ø
35998 :132,238,231,246,254,255,234
36ØØ4 :246,231,132,143,149,142,183
36Ø1Ø :148,15Ø,142,134,145,242,1Ø7
36Ø16 :146,134,149,132,231,151,Ø95
36Ø22 :134,145,145,138,15Ø,144,Ø14
36Ø28 :145,133,1Ø6,132,1Ø7,169,212
36Ø34 :ØØØ,157,Ø68,ØØ3,157,Ø71,138
36Ø4Ø :ØØ3,152,Ø16,Ø13,Ø24,169,Ø65
36Ø46 :1Ø4,1Ø1,1Ø6,133,1Ø6,144,132
36Ø52 :ØØ2,23Ø,1Ø7,23Ø,1Ø7,Ø56,176
36Ø58 :165,1Ø6,233,18Ø,168,165,211
36Ø64 :1Ø7,233,ØØØ,144,ØØ5,132,Ø77
36Ø7Ø :1Ø6,254,Ø68,ØØ3,165,1Ø6,164
36Ø76 :2Ø1,Ø9Ø,144,Ø14,189,Ø68,174
36Ø82 :ØØ3,2Ø8,ØØ3,254,Ø71,ØØ3,Ø16
36Ø88 :169,18Ø,229,1Ø6,176,ØØ8,Ø92
36Ø94 :189,Ø68,ØØ3,157,Ø71,ØØ3,233
361ØØ :165,1Ø6,Ø1Ø,133,1Ø6,168,18Ø
361Ø6 :185,231,139,157,Ø67,ØØ3,Ø24
36112 :185,232,139,157,Ø66,ØØ3,Ø3Ø
36118 :Ø56,169,18Ø,229,1Ø6,168,162
36124 :185,231,139,157,Ø7Ø,ØØ3,Ø45
3613Ø :185,232,139,157,Ø69,ØØ3,Ø51
36136 :Ø96,133,Ø89,132,Ø9Ø,169,237
36142 :ØØØ,133,Ø91,133,Ø92,162,145
36148 :Ø17,Ø24,1Ø2,Ø92,1Ø2,Ø91,224
36154 :1Ø2,Ø9Ø,1Ø2,Ø89,144,Ø13,Ø86
3616Ø :Ø24,165,Ø87,1Ø1,Ø91,133,153
36166 :Ø91,165,Ø88,1Ø1,Ø92,133,228
36172 :Ø92,2Ø2,2Ø8,23Ø,Ø96,Ø32,168
36178 :Ø41,141,ØØ6,Ø9Ø,Ø38,Ø91,233
36184 :Ø38,Ø92,ØØ6,Ø9Ø,Ø38,Ø91,187
3619Ø :Ø38,Ø92,165,Ø91,164,Ø92,224
36196 :Ø96,173,Ø84,ØØ3,172,Ø85,2Ø1
362Ø2 :ØØ3,162,ØØØ,Ø32,189,14Ø,12Ø
362Ø8 :173,Ø86,ØØ3,172,Ø87,ØØ3,124
36214 :162,ØØ6,Ø32,189,14Ø,173,Ø52
3622Ø :Ø88,ØØ3,172,Ø89,ØØ3,162,129
36226 :Ø12,Ø32,189,14Ø,173,Ø83,247
36232 :ØØ3,Ø77,Ø68,ØØ3,141,1Ø5,Ø21
36238 :ØØ3,173,Ø81,ØØ3,133,Ø87,11Ø
36244 :173,Ø82,ØØ3,133,Ø88,173,Ø32
3625Ø :Ø66,ØØ3,172,Ø67,ØØ3,Ø32,241
36256 :Ø81,141,141,1Ø3,ØØ3,14Ø,ØØ1
36262 :1Ø4,ØØ3,173,Ø83,ØØ3,Ø77,Ø97
36268 :Ø71,ØØ3,141,1Ø8,ØØ3,173,159
36274 :Ø69,ØØ3,172,Ø7Ø,ØØ3,Ø32,Ø15
3628Ø :Ø81,141,141,1Ø6,ØØ3,14Ø,Ø28
36286 :1Ø7,ØØ3,173,Ø83,ØØ3,Ø77,124
36292 :Ø77,ØØ3,Ø73,ØØ1,141,12Ø,Ø99
36298 :ØØ3,173,Ø75,ØØ3,172,Ø76,192
363Ø4 :ØØ3,Ø32,Ø81,141,141,118,212
3631Ø :ØØ3,14Ø,119,ØØ3,173,Ø74,214
36316 :ØØ3,Ø77,1Ø8,ØØ3,141,114,154
36322 :ØØ3,173,Ø72,ØØ3,133,Ø87,185
36328 :173,Ø73,ØØ3,133,Ø88,173,1Ø7
36334 :1Ø6,ØØ3,172,1Ø7,ØØ3,Ø32,149
3634Ø :Ø81,141,174,114,ØØ3,2Ø8,197
36346 :ØØ8,141,112,ØØ3,14Ø,113,255
36352 :ØØ3,24Ø,Ø15,Ø56,169,ØØØ,227
36358 :229,Ø91,141,112,ØØ3,169,239
36364 :ØØØ,229,Ø92,141,113,ØØ3,Ø78
3637Ø :173,Ø74,ØØ3,Ø77,1Ø5,ØØ3,197
36376 :Ø73,ØØ1,141,117,ØØ3,173,Ø2Ø
36382 :1Ø3,ØØ3,172,1Ø4,ØØ3,Ø32,191
36388 :Ø81,141,174,117,ØØ3,2Ø8,248
36394 :ØØ8,141,115,ØØ3,14Ø,116,Ø53
364ØØ :ØØ3,24Ø,Ø15,Ø56,169,ØØØ,Ø19
364Ø6 :229,Ø91,141,115,ØØ3,169,Ø34
36412 :ØØØ,229,Ø92,141,116,ØØ3,129
36418 :174,1Ø5,ØØ3,24Ø,Ø17,Ø56,149
36424 :169,ØØØ,237,1Ø3,ØØ3,141,213
3643Ø :1Ø3,ØØ3,169,ØØØ,237,1Ø4,182
36436 :ØØ3,141,1Ø4,ØØ3,174,1Ø8,1Ø5
36442 :ØØ3,24Ø,Ø17,Ø56,169,ØØØ,Ø63
36448 :237,1Ø6,ØØ3,141,1Ø6,ØØ3,18Ø
36454 :169,ØØØ,237,1Ø7,ØØ3,141,247
3646Ø :1Ø7,ØØ3,173,Ø8Ø,ØØ3,Ø77,Ø39
36466 :Ø68,ØØ3,Ø73,ØØ1,141,114,ØØ2
36472 :ØØ3,173,Ø78,ØØ3,133,Ø87,Ø85
36478 :173,Ø79,ØØ3,133,Ø88,173,ØØ7
36484 :Ø66,ØØ3,172,Ø67,ØØ3,Ø32,219
3649Ø :Ø81,141,174,114,ØØ3,2Ø8,Ø91
36496 :Ø21,Ø24,1Ø9,112,ØØ3,141,Ø42
365Ø2 :112,ØØ3,152,1Ø9,113,ØØ3,13Ø
365Ø8 :141,113,ØØ3,Ø16,Ø46,238,2Ø1
36514 :114,ØØ3,Ø16,Ø24,Ø56,173,Ø36
3652Ø :112,ØØ3,229,Ø91,141,112,Ø88
36526 :ØØ3,173,113,ØØ3,229,Ø92,Ø19
36532 :141,113,ØØ3,Ø48,ØØ5,2Ø6,184
36538 :114,ØØ3,Ø16,Ø17,Ø56,169,Ø49
36544 :ØØØ,237,112,ØØ3,141,112,Ø29
3655Ø :ØØ3,169,ØØØ,237,113,ØØ3,211
36556 :141,113,ØØ3,173,Ø8Ø,ØØ3,2Ø5
36562 :Ø77,Ø71,ØØ3,Ø73,ØØ1,141,Ø64
36568 :117,ØØ3,173,Ø69,ØØ3,172,241
36574 :Ø7Ø,ØØ3,Ø32,Ø81,141,174,211
3658Ø :117,ØØ3,2Ø8,Ø21,Ø24,1Ø9,198
36586 :115,ØØ3,141,115,ØØ3,152,251
36592 :1Ø9,116,ØØ3,141,116,ØØ3,216
36598 :Ø16,Ø46,238,117,ØØ3,Ø16,17Ø
366Ø4 :Ø24,Ø56,173,115,ØØ3,229,Ø84
3661Ø :Ø91,141,115,ØØ3,173,116,129
36616 :ØØ3,229,Ø92,141,116,ØØ3,Ø8Ø
36622 :Ø48,ØØ5,2Ø6,117,ØØ3,Ø16,153
36628 :Ø17,Ø56,169,ØØØ,237,115,1Ø2
36634 :ØØ3,141,115,ØØ3,169,ØØØ,2Ø1
3664Ø :237,116,ØØ3,141,116,ØØ3,136
36646 :173,Ø8Ø,ØØ3,Ø77,Ø77,ØØ3,195
36652 :Ø73,ØØ1,141,111,ØØ3,173,Ø34
36658 :Ø75,ØØ3,172,Ø76,ØØ3,Ø32,155
36664 :Ø81,141,141,1Ø9,ØØ3,14Ø,159
3667Ø :11Ø,ØØ3,173,Ø72,ØØ3,172,Ø83
36676 :Ø73,ØØ3,Ø32,Ø81,141,133,Ø19
36682 :Ø87,132,Ø88,173,Ø8Ø,ØØ3,125
36688 :Ø77,Ø74,ØØ3,Ø77,Ø71,ØØ3,129
36694 :141,1Ø5,ØØ3,173,Ø69,ØØ3,Ø68
367ØØ :172,Ø7Ø,ØØ3,Ø32,Ø81,141,Ø79
367Ø6 :174,1Ø5,ØØ3,2Ø8,Ø21,Ø24,121
36712 :1Ø9,1Ø3,ØØ3,141,1Ø3,ØØ3,Ø54
36718 :152,1Ø9,1Ø4,ØØ3,141,1Ø4,211
36724 :ØØ3,Ø16,Ø46,238,1Ø5,ØØ3,Ø15
3673Ø :Ø16,Ø24,Ø56,173,1Ø3,ØØ3,241
36736 :229,Ø91,141,1Ø3,ØØ3,173,1ØØ
36742 :1Ø4,ØØ3,229,Ø92,141,1Ø4,Ø39
36748 :ØØ3,Ø48,ØØ5,2Ø6,1Ø5,ØØ3,254
36754 :Ø16,Ø17,Ø56,169,ØØØ,237,129
3676Ø :1Ø3,ØØ3,141,1Ø3,ØØ3,169,162
36766 :ØØØ,237,1Ø4,ØØ3,141,1Ø4,235
36772 :ØØ3,173,Ø8Ø,ØØ3,Ø77,Ø74,Ø62
36778 :ØØ3,Ø77,Ø68,ØØ3,Ø73,ØØ1,139
36784 :141,1Ø8,ØØ3,173,Ø66,ØØ3,158
3679Ø :172,Ø67,ØØ3,Ø32,Ø81,141,166
36796 :174,1Ø8,ØØ3,2Ø8,Ø21,Ø24,214
368Ø2 :1Ø9,1Ø6,ØØ3,141,1Ø6,ØØ3,15Ø
368Ø8 :152,1Ø9,1Ø7,ØØ3,141,1Ø7,Ø51
36814 :ØØ3,Ø16,Ø46,238,1Ø8,ØØ3,1Ø8
3682Ø :Ø16,Ø24,Ø56,173,1Ø6,ØØ3,Ø78
36826 :229,Ø91,141,1Ø6,ØØ3,173,193
36832 :1Ø7,ØØ3,229,Ø92,141,1Ø7,135
36838 :ØØ3,Ø48,ØØ5,2Ø6,1Ø8,ØØ3,Ø91
36844 :Ø16,Ø17,Ø56,169,ØØØ,237,219
3685Ø :1Ø6,ØØ3,141,1Ø6,ØØ3,169,ØØ2
36856 :ØØØ,237,1Ø7,ØØ3,141,1Ø7,Ø75
36862 :ØØ3,Ø96,173,Ø92,ØØ3,Ø77,186
36868 :1Ø5,ØØ3,133,1Ø5,173,1Ø3,114
36874 :ØØ3,133,Ø87,173,1Ø4,ØØ3,ØØ1
3688Ø :133,Ø88,173,Ø9Ø,ØØ3,172,163
36886 :Ø91,ØØ3,Ø32,Ø41,141,165,239
36892 :Ø91,164,Ø92,166,1Ø5,2Ø8,Ø86
36898 :ØØ6,133,Ø38,132,Ø39,24Ø,11Ø
369Ø4 :Ø13,Ø56,169,ØØØ,229,Ø91,Ø86
3691Ø :133,Ø38,169,ØØØ,229,Ø92,195
36916 :133,Ø39,173,Ø95,ØØ3,Ø77,Ø6Ø
36922 :1Ø8,ØØ3,133,1Ø5,173,1Ø6,174
36928 :ØØ3,133,Ø87,173,1Ø7,ØØ3,Ø58
36934 :133,Ø88,173,Ø93,ØØ3,172,22Ø
3694Ø :Ø94,ØØ3,Ø32,Ø41,141,165,Ø4Ø
36946 :Ø91,164,Ø92,166,1Ø5,2Ø8,14Ø
36952 :Ø13,Ø24,1Ø1,Ø38,133,Ø38,179
36958 :152,1Ø1,Ø39,133,Ø39,Ø76,122
36964 :115,144,Ø56,165,Ø38,229,Ø79
3697Ø :Ø91,133,Ø38,165,Ø39,229,Ø33
36976 :Ø92,133,Ø39,173,Ø98,ØØ3,138
36982 :Ø77,111,ØØ3,133,1Ø5,173,2Ø8
36988 :1Ø9,ØØ3,133,Ø87,173,11Ø,227
36994 :ØØ3,133,Ø88,173,Ø96,ØØ3,114
37ØØØ :172,Ø97,ØØ3,Ø32,Ø41,141,11Ø
37ØØ6 :165,Ø91,164,Ø92,166,1Ø5,157
37Ø12 :2Ø8,Ø16,Ø24,1Ø1,Ø38,133,156
37Ø18 :Ø38,152,1Ø1,Ø39,133,Ø39,144
37Ø24 :Ø16,Ø36,23Ø,1Ø5,Ø16,Ø19,Ø7Ø
37Ø3Ø :Ø56,165,Ø38,229,Ø91,133,11Ø
37Ø36 :Ø38,165,Ø39,229,Ø92,133,1ØØ
37Ø42 :Ø39,Ø48,ØØ4,198,1Ø5,Ø16,Ø76
37Ø48 :Ø13,Ø56,169,ØØØ,229,Ø38,177
37Ø54 :133,Ø38,169,ØØØ,229,Ø39,Ø3Ø
37Ø6Ø :133,Ø39,165,Ø38,133,Ø87,Ø23
37Ø66 :165,Ø39,133,Ø88,173,1Ø1,133
37Ø72 :ØØ3,16Ø,ØØØ,Ø32,Ø41,141,Ø73
37Ø78 :165,Ø9Ø,133,Ø87,165,Ø91,177
37Ø84 :133,Ø88,173,1Ø2,ØØ3,16Ø,111
37Ø9Ø :ØØØ,Ø32,Ø41,141,165,1Ø5,198
37Ø96 :2Ø8,Ø11,Ø24,173,Ø99,ØØ3,238
371Ø2 :1Ø1,Ø9Ø,133,Ø38,Ø76,253,161
371Ø8 :144,Ø56,173,Ø99,ØØ3,229,18Ø
37114 :Ø9Ø,133,Ø38,173,Ø92,ØØ3,Ø11
3712Ø :Ø77,114,ØØ3,133,1Ø5,173,Ø93
37126 :112,ØØ3,133,Ø87,173,113,115
37132 :ØØ3,133,Ø88,173,Ø9Ø,ØØ3,246
37138 :172,Ø91,ØØ3,Ø32,Ø41,141,242
37144 :165,Ø91,164,Ø92,166,1Ø5,Ø39
3715Ø :2Ø8,ØØ6,133,Ø4Ø,132,Ø41,Ø78
37156 :24Ø,Ø13,Ø56,169,ØØØ,229,231
37162 :Ø91,133,Ø4Ø,169,ØØØ,229,192
37168 :Ø92,133,Ø41,173,Ø95,ØØ3,Ø73
37174 :Ø77,117,ØØ3,133,1Ø5,173,15Ø
3718Ø :115,ØØ3,133,Ø87,173,116,175
37186 :ØØ3,133,Ø88,173,Ø93,ØØ3,Ø47
37192 :172,Ø94,ØØ3,Ø32,Ø41,141,Ø43
37198 :165,Ø91,164,Ø92,166,1Ø5,Ø93
372Ø4 :2Ø8,Ø13,Ø24,1Ø1,Ø4Ø,133,Ø91
3721Ø :Ø4Ø,152,1Ø1,Ø41,133,Ø41,Ø86
37216 :Ø76,112,145,Ø56,165,Ø4Ø,178
37222 :229,Ø91,133,Ø4Ø,165,Ø41,Ø33
37228 :229,Ø92,133,Ø41,173,Ø98,1Ø6
37234 :ØØ3,Ø77,12Ø,ØØ3,133,1Ø5,Ø43
3724Ø :173,118,ØØ3,133,Ø87,173,Ø39
37246 :119,ØØ3,133,Ø88,173,Ø96,226
37252 :ØØ3,172,Ø97,ØØ3,Ø32,Ø41,224
37258 :141,165,Ø91,164,Ø92,166,189
37264 :1Ø5,2Ø8,Ø16,Ø24,1Ø1,Ø4Ø,126
3727Ø :133,Ø4Ø,152,1Ø1,Ø41,133,238
37276 :Ø41,Ø16,Ø36,23Ø,1Ø5,Ø16,Ø88
37282 :Ø19,Ø56,165,Ø4Ø,229,Ø91,25Ø
37288 :133,Ø4Ø,165,Ø41,229,Ø92,1ØØ
37294 :133,Ø41,Ø48,ØØ4,198,1Ø5,191
373ØØ :Ø16,Ø13,Ø56,169,ØØØ,229,151
373Ø6 :Ø4Ø,133,Ø4Ø,169,ØØØ,229,Ø29
37312 :Ø41,133,Ø41,165,Ø4Ø,133,233
37318 :Ø87,165,Ø41,133,Ø88,173,117
37324 :1Ø1,ØØ3,16Ø,ØØØ,Ø32,Ø41,Ø29
```

```
37330 :141,165,105,208,011,024,096
37336 :173,100,003,101,090,133,048
37342 :040,076,234,145,056,173,178
37348 :100,003,229,090,133,040,055
37354 :169,159,197,038,176,002,207
37360 :133,038,169,199,197,040,248
37366 :176,002,133,040,096,173,098
37372 :084,003,172,085,003,162,249
37378 :000,032,189,140,173,086,110
37384 :003,172,087,003,162,006,185
37390 :032,189,140,173,088,003,127
37396 :172,089,003,162,012,032,234
37402 :189,140,173,077,003,077,173
37408 :080,003,141,126,003,173,046
37414 :075,003,133,087,173,076,073
37420 :003,133,088,173,078,003,010
37426 :172,079,003,032,081,141,046
37432 :141,124,003,140,125,003,080
37438 :173,077,003,077,068,003,207
37444 :141,129,003,173,066,003,071
37450 :172,067,003,032,081,141,058
37456 :141,127,003,140,128,003,110
37462 :173,077,003,077,083,003,246
37468 :141,138,003,173,081,003,119
37474 :172,082,003,032,081,141,097
37480 :141,136,003,140,137,003,152
37486 :173,074,003,073,001,141,063
37492 :132,003,173,072,003,141,128
37498 :130,003,173,073,003,141,133
37504 :131,003,173,071,003,077,074
37510 :080,003,073,001,141,135,055
37516 :003,173,069,003,133,087,096
37522 :173,070,003,133,088,173,018
37528 :078,003,172,079,003,032,007
37534 :081,141,141,133,003,140,029
37540 :134,003,173,071,003,077,113
37546 :083,003,141,123,003,173,184
37552 :081,003,172,082,003,032,037
37558 :081,141,174,123,003,208,144
37564 :008,141,121,003,140,122,211
37570 :003,240,015,056,169,000,165
37576 :229,091,141,121,003,169,186
37582 :000,229,092,141,122,003,025
37588 :173,068,003,077,074,003,098
37594 :077,080,003,141,123,003,133
37600 :173,066,003,133,087,173,091
37606 :067,003,133,088,173,072,254
37612 :003,172,073,003,032,081,088
37618 :141,133,087,132,088,173,228
37624 :078,003,172,079,003,032,103
37630 :081,141,174,123,003,208,216
37636 :021,024,109,121,003,141,167
37642 :121,003,152,109,122,003,008
37648 :141,122,003,016,046,238,070
37654 :123,003,016,024,056,173,161
37660 :121,003,229,091,141,121,222
37666 :003,173,122,003,229,092,144
37672 :141,122,003,048,005,206,053
37678 :123,003,016,017,056,169,174
37684 :000,237,121,003,141,121,163
37690 :003,169,000,237,122,003,080
37696 :141,122,003,169,000,133,120
37702 :106,032,002,148,173,139,158
37708 :003,141,147,003,173,140,171
37714 :003,141,148,003,173,141,179
37720 :003,141,149,003,173,142,187
37726 :003,141,150,003,173,143,195
37732 :003,141,151,003,173,144,203
37738 :003,141,152,003,173,145,211
37744 :003,141,153,003,173,146,219
37750 :003,141,154,003,024,165,096
37756 :106,105,010,133,106,032,104
37762 :002,148,173,147,003,133,224
37768 :251,173,148,003,133,252,072
37774 :173,139,003,141,147,003,236
37780 :133,253,173,140,003,141,223
37786 :148,003,133,254,032,242,198
37792 :138,173,149,003,133,251,239
37798 :173,150,003,133,252,173,026
37804 :141,003,141,149,003,133,230
37810 :253,173,142,003,141,150,016
37816 :003,133,254,032,242,138,218
37822 :173,151,003,133,251,173,050
37828 :152,003,133,252,173,143,028
37834 :003,141,151,003,133,253,118
37840 :173,144,003,141,152,003,056
37846 :133,254,032,242,138,173,162
37852 :153,003,133,251,173,154,063
37858 :003,133,252,173,145,003,167
37864 :141,153,003,133,253,173,064
37870 :146,003,141,154,003,133,050
37876 :254,032,242,138,165,106,157
37882 :201,090,240,003,076,122,214
37888 :147,096,056,169,090,229,019
37894 :106,032,066,151,173,123,145
37900 :003,141,068,003,173,121,009
37906 :003,172,122,003,032,081,175
37912 :141,141,066,003,140,067,070
37918 :003,173,129,003,141,074,041
37924 :003,173,127,003,172,128,130
37930 :003,032,081,141,141,072,000
37936 :003,140,073,003,173,135,063
37942 :003,141,080,003,173,133,075
37948 :003,172,134,003,032,081,229
37954 :141,141,078,003,140,079,136
37960 :003,165,106,032,066,151,083
37966 :173,126,003,141,071,003,083
37972 :173,124,003,172,125,003,172
37978 :032,081,141,141,069,003,045
37984 :140,070,003,173,132,003,105
37990 :141,077,003,173,130,003,117
37996 :172,131,003,032,081,141,156
38002 :141,075,003,140,076,003,040
38008 :173,138,003,141,083,003,149
38014 :173,136,003,172,137,003,238
38020 :032,081,141,141,081,003,099
38026 :140,082,003,032,010,149,042
38032 :165,038,141,139,003,165,027
38038 :040,141,140,003,173,071,206
38044 :003,073,001,141,071,003,192
38050 :173,077,003,073,001,141,118
38056 :077,003,173,083,003,073,068
38062 :001,141,083,003,032,010,188
38068 :149,165,038,141,141,003,049
38074 :165,040,141,142,003,173,082
38080 :068,003,073,001,141,068,034
38086 :003,173,074,003,073,001,013
38092 :141,074,003,173,080,003,166
38098 :073,001,141,080,003,032,028
38104 :010,149,165,038,141,145,096
38110 :003,165,040,141,146,003,208
38116 :173,071,003,073,001,141,178
38122 :071,003,173,077,003,073,122
38128 :001,141,077,003,173,083,206
38134 :003,073,001,141,083,003,038
38140 :032,010,149,165,038,141,019
38146 :143,003,165,040,141,144,126
38152 :003,096,173,066,003,141,234
38158 :090,003,173,067,003,141,235
38164 :091,003,173,068,003,240,086
38170 :017,056,169,000,237,090,083
38176 :003,141,090,003,169,000,182
38182 :237,091,003,141,091,003,092
38188 :173,071,003,141,092,003,015
38194 :208,026,024,173,069,003,041
38200 :109,090,003,141,090,003,236
38206 :173,070,003,109,091,003,255
38212 :141,091,003,016,048,238,093
38218 :092,003,016,026,056,173,184
38224 :090,003,237,069,003,141,111
38230 :090,003,173,091,003,237,171
38236 :070,003,141,091,003,048,192
38242 :005,206,092,003,016,017,181
38248 :056,169,000,237,090,003,147
38254 :141,090,003,169,000,237,238
38260 :091,003,141,091,003,173,106
38266 :072,003,141,093,003,173,095
38272 :073,003,141,094,003,173,103
38278 :074,003,240,017,056,169,181
38284 :000,237,093,003,141,093,195
38290 :003,169,000,237,094,003,140
38296 :141,094,003,173,077,003,131
38302 :141,095,003,208,026,024,143
38308 :173,075,003,109,093,003,108
38314 :141,093,003,173,076,003,147
38320 :109,094,003,141,094,003,108
38326 :016,048,238,095,003,016,086
38332 :026,056,173,093,003,237,008
38338 :075,003,141,093,003,173,170
38344 :094,003,237,076,003,141,242
38350 :094,003,048,005,206,095,145
38356 :003,016,017,056,169,000,217
38362 :237,093,003,141,093,003,020
38368 :169,000,237,094,003,141,100
38374 :094,003,173,078,003,141,210
38380 :096,003,173,079,003,141,219
38386 :097,003,173,080,003,240,070
38392 :017,056,169,000,237,096,055
38398 :003,141,096,003,169,000,154
38404 :237,097,003,141,097,003,070
38410 :173,083,003,141,098,003,255
38416 :208,026,024,173,081,003,019
38422 :109,096,003,141,096,003,214
38428 :173,082,003,109,097,003,239
38434 :141,097,003,016,048,238,065
38440 :098,003,016,026,056,173,156
38446 :096,003,237,081,003,141,095
38452 :096,003,173,097,003,237,149
38458 :082,003,141,097,003,048,176
38464 :005,206,098,003,016,017,153
38470 :056,169,000,237,096,003,119
38476 :141,096,003,169,000,237,210
38482 :097,003,141,097,003,173,084
38488 :161,003,133,087,173,162,039
38494 :003,133,088,173,090,003,072
38500 :172,091,003,032,081,141,108
38506 :173,092,003,240,013,056,171
38512 :169,000,229,091,133,091,057
38518 :169,000,229,092,133,092,065
38524 :024,165,091,109,155,003,159
38530 :141,090,003,165,092,109,218
38536 :156,003,141,091,003,169,187
38542 :000,141,092,003,173,091,130
38548 :003,016,020,238,092,003,008
38554 :056,169,000,237,090,003,197
38560 :141,090,003,169,000,237,032
38566 :091,003,141,091,003,173,156
38572 :093,003,172,094,003,032,057
38578 :081,141,173,095,003,240,143
38584 :013,056,169,000,229,091,230
38590 :133,091,169,000,229,092,136
38596 :133,092,024,165,091,109,042
38602 :157,003,141,093,003,165,252
38608 :092,109,158,003,141,094,037
38614 :003,169,000,141,095,003,113
38620 :173,094,003,016,020,238,252
38626 :095,003,056,169,000,237,018
38632 :093,003,141,093,003,169,222
38638 :000,237,094,003,141,094,039
38644 :003,173,096,003,172,097,020
38650 :003,032,081,141,173,098,010
38656 :003,240,013,056,169,000,225
38662 :229,091,133,091,169,000,207
38668 :229,092,133,092,024,165,235
38674 :091,109,159,003,141,096,105
38680 :003,165,092,109,160,003,044
38686 :141,097,003,169,000,141,069
38692 :098,003,173,097,003,016,170
38698 :020,238,098,003,056,169,114
38704 :000,237,096,003,141,096,109
38710 :003,169,000,237,097,003,051
38716 :141,097,003,076,000,144,009
38722 :010,168,185,231,139,133,164
38728 :088,185,232,139,133,087,168
38734 :096,032,027,134,201,001,057
38740 :208,047,173,164,003,240,151
38746 :003,076,053,164,238,165,021
38752 :003,169,119,205,165,003,248
38758 :208,003,238,164,003,173,123
38764 :170,003,172,171,003,032,147
38770 :231,151,173,170,003,105,179
38776 :011,141,170,003,173,171,021
38782 :003,105,000,141,171,003,037
38788 :096,201,002,208,047,173,091
38794 :166,003,240,003,076,053,167
38800 :164,238,167,003,169,080,197
38806 :205,167,003,208,003,238,206
38812 :166,003,173,172,003,172,077
38818 :173,003,032,231,151,173,157
38824 :172,003,105,011,141,172,004
38830 :003,173,173,003,105,000,119
38836 :141,173,003,096,173,168,166
38842 :003,240,003,076,053,164,213
38848 :238,169,003,169,080,205,032
38854 :169,003,208,003,238,168,219
38860 :003,173,174,003,172,175,136
38866 :003,032,231,151,173,174,206
38872 :003,105,011,141,174,003,141
38878 :173,175,003,105,000,141,051
38884 :175,003,096,133,029,132,028
38890 :030,032,024,134,032,041,015
38896 :134,160,002,138,145,029,080
38902 :136,165,100,145,029,136,189
38908 :165,101,145,029,032,024,236
38914 :134,032,041,134,160,005,252
38920 :138,145,029,136,165,100,209
38926 :145,029,136,165,101,145,223
38932 :029,032,024,134,032,041,056
38938 :134,160,008,138,145,029,128
38944 :136,165,100,145,029,136,231
```

```
38950 :165,101,145,029,032,024,022
38956 :134,160,009,145,029,032,041
38962 :024,134,160,010,145,029,040
38968 :024,096,032,027,134,201,058
38974 :001,208,025,173,165,003,125
38980 :208,003,076,053,164,173,233
38986 :165,003,141,176,003,169,219
38992 :000,133,029,169,192,133,224
38998 :030,076,141,152,201,002,176
39004 :208,025,173,167,003,208,108
39010 :003,076,053,164,173,167,222
39016 :003,141,176,003,169,029,113
39022 :133,029,169,197,133,030,033
39028 :076,141,152,173,169,003,062
39034 :208,003,076,053,164,173,031
39040 :169,003,141,176,003,169,021
39046 :141,133,029,169,200,133,171
39052 :030,160,000,177,029,141,165
39058 :090,003,200,177,029,141,018
39064 :091,003,200,177,029,141,025
39070 :092,003,200,177,029,141,032
39076 :093,003,200,177,029,141,039
39082 :094,003,200,177,029,141,046
39088 :095,003,200,177,029,141,053
39094 :096,003,200,177,029,141,060
39100 :097,003,200,177,029,141,067
39106 :098,003,200,177,029,141,074
39112 :065,003,200,177,029,208,114
39118 :021,032,000,144,165,038,094
39124 :133,251,133,025,165,040,191
39130 :133,252,133,026,032,108,134
39136 :138,076,242,152,032,000,096
39142 :144,165,038,133,253,165,104
39148 :040,133,254,032,242,138,051
39154 :206,176,003,240,016,024,139
39160 :165,029,105,011,133,029,208
39166 :165,030,105,000,133,030,205
39172 :076,141,152,096,032,027,016
39178 :134,201,001,208,017,169,228
39184 :000,141,164,003,141,165,118
39190 :003,141,170,003,169,192,188
39196 :141,171,003,096,201,002,130
39202 :208,019,169,000,141,166,225
39208 :003,141,167,003,169,029,040
39214 :141,172,003,169,197,141,101
39220 :173,003,096,169,000,141,122
39226 :168,003,141,169,003,169,199
39232 :141,141,174,003,169,200,124
39238 :141,175,003,096,173,003,149
39244 :220,240,027,165,198,208,110
39250 :020,164,190,177,195,240,044
39256 :009,141,119,002,230,198,019
39262 :230,190,208,005,169,000,128
39268 :141,003,220,076,066,235,073
39274 :173,141,002,208,035,165,062
39280 :203,201,004,208,007,169,136
39286 :183,160,153,076,149,153,224
39292 :201,005,208,007,169,193,139
39298 :160,153,076,149,153,201,254
39304 :006,208,007,169,203,160,121
39310 :153,076,149,153,076,072,053
39316 :235,133,195,132,196,165,180
39322 :198,208,023,032,072,235,154
39328 :165,198,240,016,198,198,151
39334 :169,016,141,003,220,169,116
39340 :000,141,001,220,169,000,191
39346 :133,190,076,066,235,083,193
39352 :067,082,069,069,078,032,069
39358 :048,013,000,083,067,082,227
39364 :069,069,078,032,049,013,250
39370 :000,083,067,082,069,069,060
39376 :078,032,050,013,000,013,138
```

Program 2: Complex Animation Demo

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

```
10 REM SET FIGURE 1          :rem 36
20 FCLEAR 1                  :rem 64
30 FOR I=1 TO 33 : READ X,Y,Z,
   CO,A                      :rem 238
40 FSET 1,X*500,Y*500,Z*500,CO
   ,A                       :rem 190
50 NEXT I                   :rem 237
60 REM SET FIGURE 2          :rem 42
70 FCLEAR 2 :RESTORE        :rem 164
80 FOR I=1 TO 33 : READ X,Y,Z,
   CO,A                     :rem 243
90 FSET 2, Y*500, X*500,-Z*500
   ,4-CO,A                   :rem 82
100 NEXT I                   :rem 25
110 REM{3 SPACES}ANIMATION OF
    {SPACE}THE CROSS         :rem 22
120 A=1:B=2:C=3              :rem 26
130 PARAM 80,100,25,165
                            :rem 133
140 COLOR 12,11,10,13,14
                            :rem 179
150 FORJ=10TO50STEP10        :rem 10
160 FORI=10TO80STEP10        :rem 13
170 ANGLEI,J,I{4 SPACES}:ANIM1
    :FIGURE1:FIGURE2         :rem 30
180 ANGLEI+5,J,I+5:ANIM2:FIGUR
    E1:FIGURE2              :rem 224
190 NEXTI,J                 :rem 152
200 FORI=1TO2000:NEXT        :rem 14
210 SCREENØ:REM NORMAL:rem 106
220 END                     :rem 107
230 REM CO-ORDINATES OF THE CR
    OSS                     :rem 225
240 DATA -1,-1,-1,1,Ø,-1,-4,-1
    ,1,1,-1,-4,-3,1,1 :rem 147
250 DATA -1,4,-3,1,1,-1,4,-1,1
    ,1,1,4,-1,1,1,1,4 :rem 161
260 DATA -3,1,1,1,-4,-3,1,1,1,
    -4,-1,1,1,1,-1,-1,1
                            :rem 248
270 DATA 1,-1,-4,-1,1,Ø,1,-4,-
    1,1,1,-1,-4,-3,1,Ø:rem 200
280 DATA 1,-4,-3,1,1,-1,4,-3,1
    ,Ø,1,4,-3,1,1,1,1 :rem 164
290 DATA -1,1,Ø,1,4,-1,1,1,-1,
    1,-1,1,Ø,-1,4,-1,1:rem 200
300 DATA 1,-1,-1,-1,2,Ø,1,-1,-
    1,2,1,1,-1,-2,2,1 :rem 143
310 DATA -1,-1,-2,2,1,-1,1,-2,
    2,1,-1,1,-1,2,1,1,1
                            :rem 239
320 DATA -1,2,1,1,1,-2,2,1,1,-
    1,-2,2,1,1,1,-2,2,Ø
                            :rem 244
330 DATA -1,1,-2,2,1,-1,-1,-1,
    2,Ø,-1,-1,-2,2,1    :rem 99
```

Program 3: Observation Angles Demo

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

```
10 PARAM 80,100,30,165 :rem 78
20 COLOR 12,11,10,13,14
                       :rem 128
30 FORJ=5TO355STEP10   :rem 227
40 FORI=ØTO340STEP20   :rem 217
50 ANGLE J,I,I{6 SPACES}:ANIM2
   :GOSUB100           :rem 220
60 ANGLE J,I+10,I+10:ANIM1:GOS
   UB100               :rem 244
70 NEXTI,J             :rem 101
80 FORI=1TO500:NEXT    :rem 183
90 SCREENØ:END         :rem 106
100 DLINE-500,Ø,Ø,2000,Ø,Ø,1
                       :rem 122
110 DLINEØ,-500,Ø,Ø,2000,Ø,2
                       :rem 124
120 DLINEØ,Ø,-500,Ø,Ø,2000,3
                       :rem 126
130 RETURN             :rem 116
```

©

IBM Graphics On A Monochrome Monitor

Thomas G. Hanlin III

Though advanced IBM graphics require a color/graphics adapter, you can create simple graphics and even animation on a monochrome system as well. Here's a short program to show how it's done.

IBM PC computers can generate stunning graphics, but advanced BASIC graphics features are available only on PCjrs or PCs with a color/graphics adapter. However, with the right programming methods, your monochrome system can produce graphics, too. Granted, they are fairly low resolution—and no amount of programming skill can make your monochrome monitor display more than one color—but they are graphics nonetheless. You may find them handy for utilitarian purposes (for example, adding interest to bar graph displays), or you may enjoy making simple graphic screens, animated figures, or games. Once you master the basic technique, more and more applications will come to mind.

Character Graphics

When an IBM PC boots up, it checks to see if the system contains a color/graphics adapter and configures itself accordingly. If a color/graphics adapter is present, you may use advanced BASIC graphics commands like PUT and GET. If not, those commands cause an error. However, even a monochrome system has the ability to display a large set of special characters. IBM graphics characters have ASCII values of 128 to 255 and include a number of different shapes useful in creating boxes, borders, and so on.

The characters we're interested in are those which consist of a solid block. All computer graphics are produced by turning *pixels* (picture elements) on or off to light up different parts of the screen. The smaller the size of the pixel dots, the more detailed the image. Although the IBM character set doesn't include any pixel-sized characters—each character is composed of several pixels—it does include some we can use like giant pixels.

Giant Pixels

For example, CHR$(219) is a solid block character, the inverse of CHR$(32), the blank space. Using these two characters together provides a graphics screen with 80 × 25 resolution. To turn on a "dot" within this coarse screen, print the solid block at the desired spot. To turn off a dot, print a space. The BASIC function SCREEN(Y,X) tells you whether a given location contains a dot or an empty space. Though you're limited to simple, quite blocky shapes, this system is fast and simple to use. However, it's possible to do much better.

Besides the block and space characters which light up or blank out an entire screen location, there are some which light up only part of a screen position. For instance, CHR$(220) is solid on the bottom half and blank on the top. The reverse is true of CHR$(223). By using these characters, we can double our resolution to 80 × 50 pixels. This complicates matters a bit, since we want to use only half a screen position, and BASIC lets you print only to an entire screen position. Here's a point-plotting routine that handles the tricky details for you:

```
10000 GR.Y=Y\2+1:GR.SC=SCREEN
      (GR.Y,X+1):GR.OFFSET=(Y
       MOD 2)*3:IF Z=Ø THEN 1
      ØØ2Ø ELSE IF GR.SC=32 T
      HEN GR.SC=223-GR.OFFSET
       ELSE IF GR.SC+GR.OFFSE
      T<>223 THEN GR.SC=219
10010 LOCATE GR.Y,X+1:PRINT C
      HR$(GR.SC);:RETURN
10020 IF GR.SC+GR.OFFSET=223
      THEN GR.SC=32 ELSE IF G
      R.SC<>32 THEN GR.SC=22Ø
      +GR.OFFSET
10030 GOTO 1ØØ1Ø
10040 GR.Y=Y\2+1:S9=SCREEN(GR
      .Y,X+1):Z=(GR.SC=219 OR
       GR.SC+(Y MOD 2)*3=223)
      :RETURN
```

To plot a point with this routine, set the variable X to the desired horizontal coordinate (0–79) and the variable Y to the vertical coordinate (0–49). Now you've set the screen location for the giant pixel. To turn it on, set the variable Z to 1. Set Z to 0 to turn the pixel off. Then call the subroutine with GOSUB 10000. Line 10040 is a separate routine that tells you whether a given location is lit up or blank. To test any point on the screen, set the variables X and Y to the appropriate coordinates; then GOSUB 10040. The variable Z equals −1 if that point is lit or 0 if it's blank.

An Animated Snake

Though this system emulates a simple graphics screen, keep in mind that you are still printing characters. Thus, there are four screen locations that cause everything to scroll upward if you plot a point there: locations (79,46), (79,47), (79,48), and (79,49). To avoid scrolling your display, either do not use these particular locations or restrict your screen to 79 × 50 pixels (use horizontal locations 0–78). Note that you can mix text and graphics freely, but putting graphics on top of text causes some surprising results. The following program demonstrates how to animate a simple figure. Add these lines to the point-plotting routine and save the program. Make sure the numeric keypad is in numeric mode before you run it.

```
1Ø KEY OFF:CLS:DEFINT A-Z:Y=Ø
   :Z=1:FOR X=Ø TO 24:SNAKE$=
   SNAKE$+CHR$(X)+CHR$(Y):GOS
   UB 1ØØØØ:NEXT:DX=1:DY=Ø:X=
   X-1
2Ø I$=INKEY$:IF I$<>"" THEN D
   X=SGN(INSTR("369",I$)-INST
   R("147",I$)):DY=SGN(INSTR(
   "123",I$)-INSTR("789",I$))
   :IF I$=" " THEN CLS:END
3Ø X=ASC(RIGHT$(SNAKE$,2))+DX
   :Y=ASC(RIGHT$(SNAKE$,1))+D
   Y:IF X>78 THEN X=Ø ELSE IF
    X<Ø THEN X=78
4Ø IF Y>49 THEN Y=Ø ELSE IF Y
   <Ø THEN Y=49
5Ø Z=1:GOSUB 1ØØØØ:SNAKE$=SNA
   KE$+CHR$(X)+CHR$(Y):X=ASC(
   LEFT$(SNAKE$,1)):Y=ASC(MID
   $(SNAKE$,2,1)):Z=Ø:GOSUB 1
   ØØØØ:SNAKE$=MID$(SNAKE$,3)
   :GOTO 2Ø
```

Control the direction of the wandering animated snake by using the numeric keypad. Press the space bar to end the program. To improve its speed, the point-plotting routine is as short as possible. However, if you don't require fast drawing, you might want to add other features. Perhaps you'd like to color or shade the points to introduce different degrees of brightness (of course, since each two-pixel pair corresponds to a single character, there's a limit to this technique). You might add range checking to check for valid coordinates before you plot a point. And you could also modify the routine to place graphics on top of text correctly. ©

WORLD TRADE

Make us an offer we can't refuse!

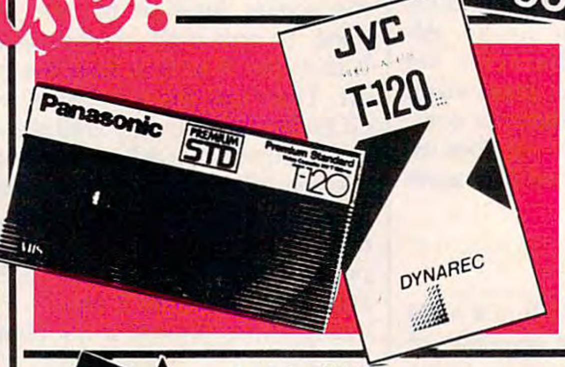
NEW FOR
commodore
ENHANCER 2000 DISK DRIVE

$159⁹⁵

Super high speed slim sign construction commodore compatable direct drive motor.

SUPER SPECIAL!

AUDO TAPES
TDK-SA90
MAXELL XL2
YOUR CHOICE
1⁹⁹ EA.
Min. purchase 10



VIDEO TAPES
TDK T120
SCOTCH T120
JVC T120
PANASONIC T120
YOUR CHOICE
4³⁹ EA.
Min. purchase 10

SUPER SPECIAL!

MODEMS
ANCHOR HIGH SPEED
1200-BAUD
VOLKSMODEM-12

$179⁹⁵
(COMMODORE COMPATABLE)

NASHUA
9⁹⁵

ELEPHANT
11⁹⁵

SCOTCH
12⁹⁵

MIX OR MATCH 10 BOXES & GET 60 DISK HOLDER FREE!

WE'LL SHOW YOU OUR ACTUAL COST OF MERCHANDISE YOU MAKE US A REASONABLE OFFER... YOU'LL GET THE BUY OF A LIFETIME!

You Set the Price

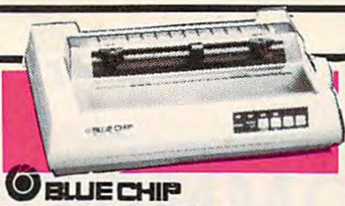
5 SUPER SIZE STORES	PATCHOGUE 437 Sunrise Hwy. 516-475-0560	BAYSIDE 191-28 Northern Blvd. 718-539-9555	W. BABYLON 358 Sunrise Hwy. 516-587-1010	NEW HYDE PARK 2075 Jericho Tpke. 516-437-3020	SYOSSET 27 Jackson Ave. 516-921-5454	VISA C.O.D. MasterCard

commodore 128

PERSONAL COMPUTER

128K Expandable to 512K. 40 and/or 80 column full color display, Professional typewriter keyboard with full numeric keypad. Built-in basic. Full CP/M Plus.

OUR COST **$268** OUR PRICE $319



BLUE CHIP

M120/NLQ PRINTER

Near letter quality button Tractor/friction, 120cps/25cps, NLQ.

OUR COST **$179** OUR PRICE $199

ATARI

520ST COMPUTER SYSTEM

CALL

Includes 512K keyboard, Monitor, Disk Drive & Mouse.

BLUE CHIP

DISK DRIVE

OUR COST **$134** OUR PRICE $169

100% Commodore compatible. Reads & writes programs faster than Commodore disk drive. Enhanced technology enables the BCD/5.25 to be more compact, use less power & gives off less heat.

Color 13" Monitor 1 Year Parts & Labor

TEKNIKA COMPOSITE MONITOR

OUR COST **$142** OUR PRICE $169

ATARI

130XE COMPUTER AND1050 DISK DRIVE

$299.95

COMPUTER WHOLESALERS

WORLD TRADE

MAIL ORDERS TO: WORLD TRADE 2075 JERICHO TURNPIKE, NEW HYDE PARK, NEW YORK 11040

SPECIAL SOFTWARE FOR THE 128 TIME WORKS

	Our Cost	Our Price
Data Manager	36.93	69.95
Swiftcalc with sideways	36.93	69.95
Word Writer with Spell checker	36.93	69.95

CBS SOFTWARE FOR COMMODORE 64

Grand prize $10,000 cash for tuition. 1st prize $1,000 educational software 2nd prize $100 each for school supplies Plus...with proof of purchase you will get $10 rebate on CBS software's Easy Key programs.

	Our Cost	Our Price
Dr. Seuss Fix Up The Mix-Up Puzzle	17.00	29.95
Many ways to say I Love You	17.00	29.95
Astro-grover	20.00	34.95
Big Bird's Funhouse	20.00	34.95
Ernie's Magic Shapes	14.00	24.95
Weather Tamers	11.00	19.95
Forcast	28.00	49.95
Dinosaur dig	23.00	39.95
Coast to Coast	23.00	39.95
Mastering the SAT	47.94	77.95
Success with Math Series	15.00	24.95
Success with Algebra Series	20.00	34.95
Muppet Learning Keys	52.00	89.95

WORD PROCESSING

	Our Cost	Our Price
Fleet system 2 w/70,000 Word spellpak	50.00	89.95
Front master	12.00	24.95

MODEMS

	Our Cost	Our Price
Video 7	40.00	79.95
Miley Mo (1 hour free compuserve time)	55.00	89.95
West Ridge	54.00	89.95

ACCESS SOFTWARE

	Our Cost	Our Price
Neutral Zone	20.00	34.95
Mach 5	21.00	34.95
Beach-Head II	24.00	34.95

TIMEWORKS SOFTWARE

	Our Cost	Our Price
Data Manager II	29.97	49.95
Evelyn Wood Dynamic Reader	29.97	49.95
Work Writer with Spellcheck	29.97	49.95
Swifcalc w/Sideways	29.97	49.95
Cave of the Word Wizards	20.97	34.95

EPYX SOFTWARE

	Our Cost	Our Price
Hot Wheels	17.97	29.95
Summer Games II	19.00	39.95
Fast Load	19.00	29.95
Barbie	17.97	29.95
Breakdance	19.00	29.95
Baseball	19.00	34.95
Pitstop II	19.00	34.95
9 to 5 Typing	23.97	34.95

INTERFACES

		Our Price
Xetec Super Graphix	50.00	89.95
8K Buffer 10 printing modes		
Super Graphix Jr.	30.00	59.95
Tymac Connection		
G-Wiz	45.00	69.95

DISKETTES

FREE EXTRA DISK INSIDE
Featuring Success with Math, and Success with Algebra by CBS. (For Commodore Only!)

Elephant (box of 10) DD/DS **10.50** 11.95

	Our Cost	Our Price
Nashau DD/DS (box of 10)	9.00	12.95
Nashua diskettes 3½" DD/SS (box of 10)	8.00	14.95
Nashua (box of 10) DD/SS	22.50	34.95
Scotch (3M) (box of 10) DD/SS	10.99	14.95
Scotch Head Cleaner Kit	8.70	14.95

NEW! NIBBLE NOTCH

Turns your 5½" single Disk into A 2 Sided Disk

Our Cost **$5** Our Price $14.95

DISK FILES

	Our Cost	Our Price
10 Disk Holder	4.70	7.00
60 Disk Holder (stackable)	8.00	14.95

MONITORS

	Our Cost	Our Price
Commodore 1702	168.00	189.00
Sharp 13" Color Compsite	148	169.00
Sharp green 12"	50.00	79.00
Sharp amber 12"	55.00	89.00
Teknika RGB/Composite 13"	225	275

JOYSTICKS

	Our Cost	Our Price
Wico Boss	9.00	19.95
Wico Bat	11.00	15.95
Wico Threeway	18.00	27.95
Gim 1000-left/right firing button	7.00	14.95

PRINTERS

	Our Cost	Our Price
Blue Chip M150/10 Tractor/Friction 130pcs	180.00	249.00
Blue Chip M150/15 StarSG10 Tractor/Friction 130cps.	195.00	259.00
Blue Chip 120/NLQ	175.00	239.00
Blue Chip 120/NLQ	180.00	249.00
SG10-C w/built in Commodore Interface	210.00	269.00
Okimate 10 color	180.00	249.00
Legende 808	148.00	179.95
Legende 880	205.00	239.95
Legende 1080	227.00	269.95
Legende 1380	272.00	300.95
Legende 1385	342.00	379.95
Cardco LQ3 letter quality		

PAPER & RIBBON

Delivery on paper included in price

paper size 9½ x 11"

1000 x 3 boxes Microperf	35.00	70.00
2700 microperf	28.00	62.00
Ribbon bluechip M120/M150	7.00	14.95
Ribbon SG10/SG15/ Okidate 82A	2.00	4.95

SURGE PROTECTORS

	Our Cost	Our Price
6 Outlet with lighted switch	17.00	29.95
Surge Plus	22.00	34.95

OSS SOFTWARE

	Our Cost	Our Price
Basic XL	47.00	79.00
Basic XE	57.00	99.00
Mac 65	57.00	99.00
Action	57.00	99.00
Dos XL w/Bug 65	24.00	39.00
Bug 65	21.00	34.00
Writers Tool	57.00	99.00
Action Tool Kit	24.00	34.95
Basic XL Tool Kit	24.00	34.95
Mac 65 Tool Kit	24.00	34.95

BATTERIES INCLUDED SOFTWARE

	Our Cost	Our Price
Paper Clip	50.00	79.00
B. Graph	41.00	69.00
Home Pak	29.00	45.00

BRODERBUND SOFTWARE

	Our Cost	Our Price
Print Shop	25.00	39.95
Print Shop Lib I	14.00	24.95
Print Shop Lib II	14.00	24.95

TK SOFTWARE

	Our Cost	Our Price
Monday Morning Manager	21.00	29.95
Bowling Version 2	21.00	29.95

PROFESSIONAL SOFTWARE

Fleet System II Work Processor with built in spell check	50.00	89.95

NEW! MINDSCAPE INC. PRESENTS "THE HALLEY PROJECT: A MISSION IN OUR SOLAR SYSTEM" ...... CALL
FOR USE WITH BOTH ATARI & COMMODORE

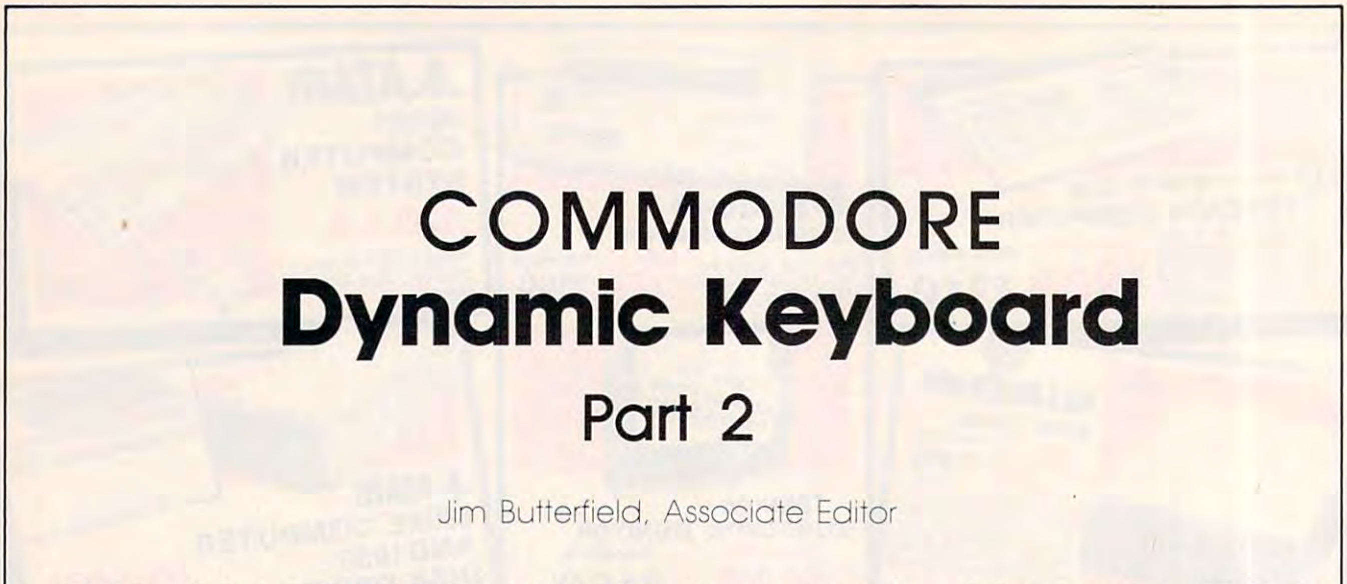
TO ORDER **CALL** OUTSIDE N.Y. **1-800-722-4002** ORDER LINE ONLY

NEW! BIG FIVE SOFTWARE PRESENTS "BOUNTY BOB STRIKES BACK!" ...... CALL
FOR USE WITH BOTH ATARI & COMMODORE

IN NEW YORK 1-718-895-2388
WHEN IN THE NEW YORK AREA VISIT OUR 5 BIG L STORES.
• West Babylon • New Hyde Park • Syosset • Patchogue • Bayside

Check, Money Order, MC or VISA accepted/Call for shipping and handling information / NYS residents add applicable sales tax / Prices and availability are subject to change without notice / Air factory fresh merchandise with manufacturer's warranty. Dealers welcome. Call for recent price reductions and new rebate information. No returns without prior authorization.
FOR INFORMATION & CUSTOMER SERVICE CALL 1-718-895-2388.

VISA MasterCard C.O.D.



COMMODORE
Dynamic Keyboard
Part 2

Jim Butterfield, Associate Editor

Part 1 discussed the fundamentals of dynamic keyboard programming, which in effect allows the computer to "type on its own keyboard." Now let's look at some important applications for this technique.

As we saw in Part 1, dynamic keyboard programming uses a two-step method to let a program give itself direct-mode commands. Step 1 is to print the command at a specific location on the screen. Step 2 is to put a RETURN character in the computer's keyboard buffer, then stop the program with the cursor flashing over the screen command. The RETURN character makes the computer execute the command just as if you'd pressed RETURN.

It's worth mentioning that you may print more than one command on a screen line. Just as in a program line, separate the multiple direct-mode commands with colons. You can use more than one screen line of direct-mode commands as well. However, you must be careful to put the commands in exactly the right place, and make sure the cursor flashes directly over the line to be executed when the program stops.

Here are some applications for the dynamic keyboard technique:

- Allow a user to enter a formula that the program will use;
- Allow a program to load another program;
- Allow a program to modify itself (tricky);
- Run test programs to determine, for instance, how the computer responds to certain direct commands and calculations.

Keyboard Buffer Locations

The following table shows the location of the keyboard buffer counter and the start of the keyboard buffer on most Commodore computers:

	Counter	Buffer
VIC-20, Commodore 64	198	631
Commodore 16, Plus/4	239	1319
PET/CBM (4.0 & Upgrade BASIC)	158	623
PET (Original ROM)	525	527
B128 (Model 700)	209	939

Usually your program must POKE a value of 1 into the counter and a value of 13 (the character code for RETURN) into the buffer. That tells the computer there's one RETURN character in the buffer waiting to be processed. If there's more than one line of direct-mode commands on the screen to be performed, you need a higher count and more characters. On the B128, it's wise to execute a BANK 15 command before the POKEs.

Entering A Formula

Let's write a brief program that allows a student to enter a formula and then generates a table of values based on the formula. More complex versions of the program might solve an equation or draw a graph, but we'll keep the example simple. In practice, it would be wise for your program to check for valid syntax before evaluating the formula. Again, for the sake of brevity, we'll do only the dynamic keyboard portion.

This program is for VIC-20 and Commodore 64 only. If you have another Commodore model, use the table above to change the POKE addresses in line 140. Also, don't forget the colon that appears just before the GOTO statement in line 130.

```
100 PRINT"{CLR}{DOWN}FORMULA E
    VALUATION.":PRINT"INPUT A
    {SPACE}FORMULA"      :rem 52
110 PRINT"BASED ON VARIABLE X"
    :PRINT"SUCH AS:":PRINT"
    {DOWN}{2 SPACES}Y= X*7-SQR
    (X)":PRINT            :rem 7
120 PRINT"YOUR FORMULA:":INPUT
    "{DOWN}{2 SPACES}Y=";F$:PR
    INTCHR$(147):PRINT:PRINT
                        :rem 160
130 PRINT"Y=";F$;":GOTO150":DI
    MV(10):FORX=1TO10:PRINTCHR
    $(19)               :rem 178
140 POKE 198,1:POKE631,13:END
                        :rem 103
150 V(X)=Y:NEXT X:FOR X=1 TO 1
    0:PRINT X,V(X):NEXT X
                          :rem 2
```

Notice how this program does a task which would be difficult or impossible without using the dynamic keyboard technique.

Loading Another Program

If you put a LOAD command in a program, the new program doesn't load in the usual way. Instead, it's *chained* to the old program. The new program retains the variables and arrays (if certain rules are observed), and the effect is that of two successive programs working continuously on a single job. That's not always what is wanted. Especially with menu programs or bootstraps (program-loading programs), your goal may be simply to start the new program without preserving variables or data from the old one. That's what happens when you perform LOAD as a direct command. With the dynamic keyboard technique, we can simulate this from within a program.

Let's write a simple dynamic keyboard loading sequence. Again, the program is given for VIC-20 and Commodore 64 only. For other Commodore models, use the table above to change the POKE addresses in line 120.

```
100 PRINT"{CLR}{DOWN}PROGRAM L
    OADING":PRINT"PROGRAM
    {2 DOWN}":PRINT"PROGRAM NA
    ME":INPUTP$
110 PRINT"{CLR}":PRINT:PRINT:P
    RINT"LOAD";CHR$(34);P$;CHR
    $(34);",8":PRINT:PRINT
120 PRINT:PRINT:PRINT"RUN":PRI
    NTCHR$(19):POKE198,2:POKE6
    31,13:POKE632,13
```

Note that there are two separate command lines: one for LOAD and one for RUN. Of course, it's important to position the lines correctly, but that's not hard to work out when you set up the program. You see everything happening on the screen, and, if you've placed your command a line too high or low, the problem is easy to spot. (For the VIC, you must limit the length of the filename you enter to seven or fewer characters. Otherwise, an unrelated bug built into the VIC's INPUT statement causes the program to fail.)

Tricks And Advanced Points

On computers with color capabilities, you can hide your dynamic keyboard tricks if you wish. If you print the direct-mode commands in the same character color as the screen background, they won't be visible to you, but the computer can still see and execute them. Your program can even change colors as it runs so that some parts of the commands are visible and some are not.

Occasionally, you'll want to use the dynamic keyboard technique to change a program as it runs. That's tricky, since any time you add or change a program line, the values of all variables are lost. It's hard to run a program when its variables disappear, but it can be done if handled carefully. The critical variables can be reentered using the dynamic keyboard technique, using lines such as X=7:L=120: GOTO 580. Another, somewhat more cumbersome method is to POKE the value of each variable into spare memory and PEEK the value later when needed.

Why would a program need to change itself? The most usual situation involves converting an ASCII program listing into tokenized BASIC format. It's common to list programs in ASCII (untokenized) form when translating from one computer to another. This is especially true when you transfer programs over the phone line with a modem. As each line of the ASCII listing arrives, it must be entered as if it were being typed, to store it in tokenized format. While it's possible to do the whole job by hand (by printing each line on the screen and pressing RETURN), the dynamic keyboard technique lets the computer do this busywork for you.

Next month, in Part 3, we'll cover the use of the dynamic keyboard technique for self-modifying programs in more detail. ©

Simple Assembling With IBM DEBUG

Tim Victor, Editorial Programmer

You don't need to buy an expensive assembler to write short machine language programs on an IBM PC or PCjr—a copy of PC-DOS already contains the basic tools you require. This article, which assumes some familiarity with hexadecimal numbers and machine language theory, shows how to make the most of the DEBUG utility when you're ready to tackle 8088 machine language.

Tucked away on the *DOS Supplemental Programs* disk that came with your copy of PC-DOS is a file called DEBUG. DEBUG is a simple but powerful development tool for exploring your computer and writing short machine language (ML) programs. It includes a *miniassembler,* which converts assembly language instructions into machine language directly in memory, and a *disassembler,* which allows you to reverse this process and examine ML programs already in memory. DEBUG also has trace and breakpoint functions for testing ML programs, utilities for loading and saving programs on disk, and several other valuable features. Using these tools, we'll show how to write a small ML program.

To get started with DEBUG, boot up DOS from your master disk. When the DOS prompt A> appears, insert the *DOS Supplemental Programs* disk into drive A:, type DEBUG, and press ENTER. DEBUG loads and runs, replacing the DOS prompt with its own prompt, a hyphen (-). You can return to DOS at any time by putting your master disk back in the drive, typing Q for Quit, and pressing ENTER.

Since you should preserve your *DOS Supplemental Programs* disk as an archival backup, let's ask DEBUG to copy itself onto another disk. You could use the DOS COPY command, but using DEBUG is a good way to learn how to load and save machine language program files.

Cloning DEBUG

DEBUG has three commands for disk operations: L (Load), W (Write), and N (Name). N creates a data structure called a *file control block* (FCB) that DOS uses for all disk operations, including DEBUG's Load and Write. The FCB contains the name of a file, along with information such as size and file organization. To learn more about the FCB, consult Appendix E of the *DOS 2.00 Manual,* or Chapter 6 of the *DOS 2.10 Technical Reference Manual.*

The first step in backing up DEBUG is to load another copy of it into memory. Type N DEBUG.COM and press ENTER. (You need to include the .COM extension because DEBUG doesn't make any assumptions about the file type.) DEBUG responds with another hyphen. Next, type L and press ENTER. The disk drive whirs, and then another hyphen appears. You've loaded a second copy of DEBUG.

Remove the *Supplemental Programs* disk. Replace it with a formatted disk that you'll be using for ML programs. Type W and press ENTER. The drive comes on again, and then DEBUG displays the message "Writing 2E80 bytes" and the hyphen prompt. You now have a copy of DEBUG.COM on your ML disk.

A Sample Program

Let's try assembling a program with DEBUG. Start by typing A 100 to start assembling at address 100H. (IBM programmers generally denote hexadecimal numbers by appending an H to the number. All input and output with DEBUG is expressed in hexadecimal.) DEBUG responds with *xxxx*:0100, where *xxxx* is a four-digit hexadecimal number. This number is the current value of the *code segment register.* It's of minor importance right now and will be discussed in detail later.

Now type in the following program. DEBUG displays the memory address of each instruction for you. All you need to enter are the instructions.

```
MOV AH,09
MOV DX,109
INT 21
INT 20
DB "HELLO THERE$"
```

Press ENTER to leave the assembler. This program is the ML equivalent of everyone's first BASIC program:

```
10 PRINT "HELLO THERE"
```

The ML version looks quite a bit longer, but it would be even more involved if it weren't for the INT 21H instruction, which calls a DOS function routine (Print String) by executing a software INTerrupt. Before calling this routine, the program takes two preparatory actions. The first instruction loads the AH register (an internal 8088 register) with the value 9. In 8088 machine language, instructions with two operands like MOV AH,09 operate from right to left—just as A=9 in

BASIC *moves* the value 9 into the variable A. You specify the destination operand first, then the source operand. This might seem a little backwards, but it's a common convention and you'll soon adjust to it.

AH is the high (most significant) byte of AX, the 16-bit (two-byte) accumulator register of the 8088. When a program calls Interrupt 21H, the value in AH indicates the function you're asking DOS to perform. Function 9, Print String, displays a string on the screen, starting with the character at the address contained in the DX register and ending with the character $. The second instruction moves the address 109H into the DX register. The last instruction, INT 20H, ends the program by returning control to the program that called it—in this case, DEBUG.

Finally, we create the string we want to print using DB, a pseudo-opcode (*pseudo-op*). When the assembler sees a pseudo-op such as DB, it performs a function instead of generating code. This particular pseudo-op tells the assembler to store bytes of data in memory, beginning at the current location. The data can be either a list of hexadecimal numbers between 00 and FF, separated by spaces or commas, or a quoted string, as shown above. If the data is a string, the ASCII code for each character is entered in memory. The dollar sign at the end of the string is very important. Without this *delimiter*, the Print String function will keep printing whatever bytes it happens to find in memory following the message. It might be a long time before it comes across a $ and stops.

8088 Memory Addressing

Now that the program is in memory, we can use the disassembler to examine it. Type U for Unassemble, and DEBUG displays several rows of text on the screen (the number of rows differs between 40- and 80-column displays). Notice that the disassembled code is aligned in four columns. The first column shows the address of each instruction as two four-digit hexadecimal numbers separated by a colon, just as was displayed when you entered the program. The first four-digit number is the current value of the code segment register mentioned before, and the second is the value of the *instruction pointer*. To understand why two registers are needed to point to a single memory location requires some understanding of the 8088's addressing scheme.

The 8088 microprocessor can access up to one megabyte (1024K) of memory using 20-bit addresses. However, for compatibility with older Intel processors, the 8088 has only a 16-bit instruction pointer. Because a 16-bit (four hexadecimal digit) register can only have values between 0 and 65,535, another register, the code segment register, is needed to address the entire 1,048,576 bytes allowed by the 8088. The code segment register is also a 16-bit register, but instead of addressing individual bytes, it points to blocks of 16 bytes, called *paragraphs*. Any five-digit hexadecimal address that ends in a zero is the beginning of a paragraph. For example, the byte of memory at 5D320H is at the beginning of the paragraph addressed by a segment register containing 5D32H.

The code segment register points to the first paragraph of a 64K block of memory called the *code segment* (CS). There are three other segments, the *data segment* (DS), *stack segment* (SS), and *extra segment* (ES), plus a register that points to the beginning of each. In simple programs, however, all the segment registers usually have the same value as CS. To find the next byte of code to be fetched, the value in the instruction pointer is added to the address of the beginning of the code segment. The physical address of this byte can be found with this formula:

Physical Address = IP+(CS*16)

The effect of organizing memory this way is that a programmer doesn't have to know where the program will be loaded. When DOS loads a .COM program, it starts the code segment at the beginning of any available paragraph in memory. The program is loaded at an offset of 100H bytes above the start of the segment and the instruction pointer is set to 100H. The four segment registers, CS, DS, SS, and ES, all point to the start of the code segment.

The second instruction of the example program moves an address, 109H, into DX. This address is an offset into the current data segment. The string to be printed is located at an offset of 109H only if the data segment register is equal to the code segment register and the program starts at offset 100H. In practice, the CS register is rarely changed except by DOS and needs little or no attention in most programs.

Displaying Binary Code

The second column of the disassembled listing on the screen contains four- or six-digit hexadecimal numbers. These are the contents of the memory locations, the binary code which the 8088 can execute. Notice that the first MOV instruction is one byte shorter than the second. The first instruction only loads half of a 16-bit register (AH is the upper half of AX), so the data occupies one byte, but the second MOV loads all of DX, which takes two bytes of data (a *word*).

The third column shows the *mnemonics*—symbolic names for each opcode instruction. The fourth column displays the operands. This program consists of four opcodes: two MOV instructions followed by two INT instructions. Notice that the DB pseudo-op doesn't show up in a disassembly. Instead of displaying your characters, DEBUG tries to convert the string into assembler mnemonics, and therefore prints several meaningless instructions. DEBUG is frequently fooled this way because program instructions and data are both stored as binary bytes. DEBUG has no way of knowing where the program ends and the data begins.

If you type another U, DEBUG continues to disassemble and display the next 16 or 32 bytes in memory (depending on your screen width). Since the program is only 21 bytes long, DEBUG starts displaying part of itself, still in memory from when you copied it. Type U 100 to disassemble from the beginning of your program again. DEBUG's U command also accepts both starting and ending addresses if you separate them with a space.

It's a good idea to save your program on disk before running it.

If the program causes something unexpected, like an infinite loop or a complete system crash, it's nice to have a copy saved. Then you can load it and search for the error without typing the program again from scratch.

As before, you need to tell DEBUG the name of your file. Type N HELLO.COM. Now there's one more thing to consider: How many bytes of memory should DEBUG write to disk? When we used the W command to copy DEBUG, it wrote the same number of bytes that it had loaded, but now we're saving a new program which has never been loaded. When DEBUG loads a file, it stores the size of the file in the CX register and the four least significant bits of the BX register. The same registers are used when DEBUG writes a file. So if your program is less than 65,536 bytes long (most are), the BX register should be set to zero.

To examine and change CX, type R CX. DEBUG prints the contents of CX (probably 2E80H, left over from copying DEBUG), then prints a colon at the beginning of the next line. You can press ENTER to leave the value unchanged, or type a new value. Since the new program is 21 bytes long, type 15 (the hexadecimal equivalent of 21) and press ENTER. Now type W to write the program to disk. DEBUG responds with the message "Writing 0015 bytes," then returns the prompt.

Running And Debugging

Now that your program is safe on disk, run it by typing G and pressing ENTER. The screen should display HELLO THERE. Then DEBUG prints "Program completed normally" followed by its usual prompt. If your program completed but didn't print correctly, disassemble starting from 100H and check that all instructions are correct. If your program locked up the computer, reboot, restart DEBUG, and thank yourself for saving the program. Reload the program with N and L, then disassemble it to see what it looks like. If you don't know what's wrong, one technique is to try setting a *breakpoint*. This halts the program at a predetermined point so you can check the contents of the registers.

For instance, to make the program stop before the INT 20H instruction, you can set one or more breakpoints. To set a breakpoint, type G followed by the addresses of one or more instructions in your program. If you set more than one breakpoint, separate the addresses with spaces. The program begins executing, but stops when the instruction pointer equals the address of a breakpoint. DEBUG displays the contents of all registers and flags and disassembles the instruction at the breakpoint (the instruction pointed to by the instruction pointer, the next instruction to be executed). Type G to restart the program at the instruction that the instruction pointer references.

If you stopped your program with a breakpoint but want to restart it from the beginning, type G =100. DEBUG sets the instruction pointer to 100H (or whatever address you specify) before starting. You can also set both the starting address and one or more breakpoints. Just include the breakpoint addresses on the same command line, separating them from the starting address and each other with spaces.

Keep this in mind: Before DEBUG executes a G command, it saves the values of all the registers, including the instruction pointer. If the program runs normally, and completes by executing INT 20H, DEBUG restores all the registers. This is great if your program runs all the way from beginning to end. You just type G and your program runs again. If, however, your program has just completed after being restarted from a breakpoint, the instruction pointer now points to the location where the breakpoint was set. Typing G starts it from the breakpoint again. To run the program from the beginning, type G =100.

Learning More About DEBUG

You've now used DEBUG to load and store program files, to assemble and disassemble a new machine language program, and to execute a program. Some other useful commands we don't have room to cover are D (Dump), which displays the contents of a block of memory as hexadecimal numbers and ASCII characters; E (Enter), to examine and change the contents of individual memory locations; and T (Trace), which executes an ML program one instruction at a time, displaying all registers and flags between instructions.

As you learn more about 8088 machine language, you'll find DEBUG a big help in testing your programs. Though you might use a separate assembler when your programs get larger, DEBUG remains useful for testing and modifying the assembled programs. If you want to know more, there is a complete description of each DEBUG command in Chapter 12 of the *DOS 2.00 Manual* and Chapter 8 of the *DOS 2.10 Manual.* Information on the DOS functions and interrupts can be found in Appendix D of the *DOS 2.00 Manual* and Chapter 5 of the *DOS 2.10 Technical Reference Manual.* To learn more about machine language programming on the IBM PC and PCjr, see COMPUTE!'s *Beginner's Guide to Machine Language on the IBM PC & PCjr.* ©

Save-With-Replace:
Debugged At Last
Part 2

P.A. Slaymaker

Last month, Part 1 proved that a long-suspected bug in the Commodore Save-with-Replace command really exists. Using a program that demonstrated the bug on a 1541 disk drive, the article showed how disks can be scrambled when files are scratched and rewritten with Save-with-Replace. The article also offered a brief explanation of the bug and how to avoid it. This month, Part 2 examines the Save-with-Replace bug in greater detail for technically advanced readers. The author is the president of Quantum Software, which produces the Peek a Byte disk utility for the Commodore 64.

What actually causes the Save-with-Replace bug? When and how does it occur and is there a fix for it? We have performed extensive testing to determine exactly how the bug happens. As explained last month, we've determined that the bug is avoidable if the drive number (drive 0) is specified in *all* disk commands. If you don't always specify drive 0, the bug occasionally bites. That's significant information in itself—but we wanted to know *why.*

DOS Thievery

First, we should note that although the SAVE@ command deletes a disk file and saves a replacement in a single operation, it works differently than if you issued separate SCRATCH and SAVE commands. SAVE@ calls entirely different DOS routines—the SCRATCH and SAVE are executed as part of a continuous procedure, and the SAVE@ command therefore requires that more drive buffers be available.

DOS V2.6 has five internal buffers, numbered 0 to 4. These buffers start at memory pages $300, $400, $500, $600, and $700, respectively. Normally an image of the disk's BAM (block availability map) is stored in the page at $700, an image of the directory sector in use is stored at $600, and the other three buffers are available for file use. As long as a buffer is active, it cannot be used for anything else. If DOS has assigned an internal channel to the BAM at $700, then trying to open a direct channel to buffer 4 (from BASIC: OPEN 2,8,2,"#4") will produce a 70,NO CHANNEL,00,00 error.

Similarly, DOS assigns channels and buffers to the directory sector and file sectors which are being read or written. Normally DOS assigns two read or two write channels and uses only three of the five buffers. The SAVE@ command, however, requires all five buffers—two read, two write, and the BAM. If DOS can't find a free buffer, it tries to steal an assigned but inactive buffer. This thievery causes the SAVE@ command to occasionally fail—for reasons which will be discussed shortly.

Why does omitting the drive number in disk commands cause DOS to steal a buffer? When a file is opened or loaded via the OPEN routine ($D7B4), DOS searches the internal directory to look for the specified filename (DOS routine names and addresses in this article conform to those listed in *Inside Commodore DOS*, Datamost, 1984). ONEDRV ($C312) determines whether a drive was specified. OPTSCH ($C3CA) assigns a default or specified drive for each file in the command, and also calls AUTOI ($C63D). AUTOI reads the BAM of the disk in the specified drive, and also tries to initialize drive 1 if no drive was specified. Usually buffer 3 ($600) is allocated for the phantom drive 1 BAM, and a B1 SEEK command is issued to the disk controller. This results in an internal DRIVE NOT READY error in the disk controller. The error is trapped by AUTOI but not reported outside the disk drive. This leaves buffer 3 allocated but inactive. FFST ($C49D) then reads the directory and tries to find the file.

The reason this inactive buffer assignment is important is that the SAVE@ command requires all five buffers, but only four are now available. Whenever DOS needs to allocate a buffer, it calls GETBUF ($D28E). If one is not free, GETBUF tries to steal an inactive one by calling STLBUF ($D339). If the drive number is always specified and no direct access buffers are allocated, STLBUF is never called. We verified this by modifying GETBUF after copying DOS onto an EPROM (Eraseable-Programmable Read Only Memory). If a channel can't be stolen, then a NO CHANNEL error occurs. But if STLBUF is called, the SAVE@ bug sometimes occurs.

Stealing The Wrong Buffer

STLBUF can be called several times during a SAVE@ command. The

COMPUTER MAIL ORDER

THE CMO ADVANTAGE

- ✓ THE BEST PRICES!
- ✓ Next day shipping on all in stock items.
- ✓ Free easy access order inquiry.
- ✓ Orders from outside Pennsylvania and Nevada save state sales tax.
- ✓ Free technicial support with our factory trained technical staff.
- ✓ There is no limit and no deposit on C.O.D. orders.
- ✓ There's no extra charge for using your credit card. Your card is not charged until we ship.
- ✓ No waiting period for cashiers checks.
- ✓ We accept purchase orders from qualified corporations. Subject to approval.
- ✓ Educational discounts available to qualified institutions.
- ✓ FREE CATALOG MEMBERSHIP.

ORDER LINE

1-800-233-8950
In PA 1-800-242-4215

CUSTOMER SERVICE & TECH SUPPORT
1-717-327-1450
Dept. A211

MAILING ADDRESS

Computer Mail Order
Dept. A211
477 East Third Street
Williamsport, PA 17701

MEMBER DIRECT MARKETING ASSOCIATION

CREDIT CARDS

VISA MasterCard AMERICAN EXPRESS DINERS CLUB

SHIPPING

Add 3%, minimum $5.00 shipping and handling on all orders. Larger shipments may require additional charges.

All items subject to availability and price change.

Returned shipments may be subject to a restocking fee.

CANADIAN ORDERS

1-800-268-3974
Ontario/Quebec

1-800-268-4559
Other Provinces

1-416-828-0866
In Toronto

TELEX: 06-218960

2505 Dunwin Drive,
Mississauga, Ontario
Canada L5L1T1

All prices shown are for U.S.A. orders.
Call The Canadian Office for Canadian prices.

HOME COMPUTERS

ATARI

130XE (128K)	**CALL**
520ST (512K)	**CALL**

800XL 64K	CALL
850 Interface	$119.00
1010 Recorder	$49.99
1050 Disk Drive	$169.00
1025 Dot Matrix Printer	$199.99
1027 Letter Quality Printer	$269.99
1030 Direct Connect Modem	$59.99

8036 Atari Writer	$29.99
Star Raiders	$4.99
Missile Command	$4.99
Defender	$4.99
Galaxian	$4.99
Dig Dug	$4.99
Donkey Kong	$4.99
Joust	$4.99
Eastern Front	$4.99

While Supply Lasts!

BOARDS FOR ATARI

Axlon 32K	$39.99
Axlon 48K (400)	$69.99
Axlon 128K	$269.99
Bit 3 Full View 80	$169.00

APPLE

APPLE IIe	CALL
APPLE IIc	CALL
MacINTOSH	CALL
IIc LCD Display	CALL

Macintosh Software

Lotus Jazz	CALL
Microsoft Excel	CALL
Microsoft Business Pak	$375.00
Living Videotext ThinkTank 512	$159.00
Manhatten Ready, Set, Go	$79.99
Creighton Development Mac Spell	$69.99
Monogram Dollars & Sense	$99.99
Peachtree Back to Basics - GL	$109.00
PFS File & Report (New Version)	$129.00
Silicon Beach Airborn	$25.99

commodore

C128 Computer	**$299.00**
C1571 (Disk Drive for C128)	**$NEW**
C1902 (RGB 13'' Monitor for C128)	**$NEW**
C1670 (Modem for C128)	**$NEW**

Commodore Plus 4	$199.00
CBM 64	$149.00
C1541 Disk Drive	$199.00
C1530 Datasette	$39.99
M-801 Dot Matrix Printer	$169.00
MCS 803 Dot Matrix	$179.00
C1702 Color Monitor	$189.00
C1660 Auto Modem	$59.99
DPS 1101 Daisy Printer	$339.00

Professional Software

Fleet System II w/Spell✓	$49.99
Trivia Fever	$29.99
Word Pro 4 Plus/5 Plus each	$239.00
Info Pro	$179.00
Administrator	$399.00

BRODERBUND

The Print Shop	$29.99

pfs:

File (64)	$59.99
Report (64)	$49.99

BATTERIES INCLUDED

PaperClip w/Spell Pack	$79.99
The Consultant DBMS	$69.99
Bus Card II	$139.00
80 Col Display	$129.00

MODEMS

ANCHOR

Volksmodem	$59.99
Volksmodem 300/1200	$189.99
Mark XII (1200 Baud)	$259.00
Signalman Express	$299.00
Lightning 2400 Baud	$399.00

Hayes

Smartmodem 300	$139.00
Smartmodem 1200	$389.00
Smartmodem 1200B	$359.00
Smartmodem 2400	$629.00
Micromodem IIe	$149.00
Smart Com II	$89.99
Chronograph	$199.00
Transet 1000	$309.00

AST

Reach 1200 Baud Half Card	$399.00

MICROBITS

MPP-1000E AD/AA (Atari)	$79.99
MPP-1064 AD/AA (C-64)	$69.99

Novation

Smart Cat Plus	$319.00
J-Cat	$99.99
Novation 2400	CALL
Apple Cat II	$229.00
212 Apple Cat II	$379.00
Apple Cat 212 Upgrade	$229.00
Macmodem	$319.00

QUADRAM

Quadmodem II 300/1200	$339.00
300/1200/2400	$499.00

TELELEARNING

C64 300 Baud......(Closeout)	$39.99

GRAPHICS

Koala

IBM	$89.99
Apple/Franklin	$79.99

Polaroid

Palette	$1299.00

PORTABLE COMPUTERS

HEWLETT PACKARD

41CV	$189.99
41CX	$249.99
HP 71B	$419.99
HP 11C	$62.99
HP 12C/15C/16C	$89.99
HP 75D	$949.00
HPIL Module	$98.99
HPIL Cassette or Printer	$359.99
Card Reader	$143.99
Extended Function Module	$63.99
Time Module	$63.99

We stock the full line of HP calculator products

NEC

PC-8401	CALL
PC-8201 Portable Computer	$289.00
PC-8231 Disk Drive	$599.00
PC-8221A Thermal Printers	$149.00
PC-8281A Data Recorder	$99.99
PC-8201-06 8K RAM Chips	$105.00

SHARP

PC-1350	$159.99
PC-1261	$159.99
PC-1260	$109.99
PC-1500A	$165.99
PC-1250A	$88.99
CE-125 Printer/Cassette	$128.99
CE-150 Color Printer Cassette	$171.99
CE-161 16K RAM	$134.99

DISKETTES

maxell

3½'' SS/DD	$39.99
3½'' DS/DD	$54.99
5¼'' MD-1 w/Hardcase	$17.99
5¼'' MD-2 w/Hardcase	$23.99
5¼'' MD-2-HD for AT	$44.99

Verbatim.

5¼'' SS/DD	$21.99
5¼'' DS/DD	$29.99
Disk Analyzer	$24.99

Dennison

Elephant 5¼'' SS/SD	$13.99
Elephant 5¼'' SS/DD	$15.99
Elephant 5¼'' DS/DD	$16.99
Elephant Premium DS/DD	$23.99

IBM

5¼'' DS/DD Floppy Disks (Box of 10)	$25.99

DISK HOLDERS

INNOVATIVE CONCEPTS

Flip-in-File 10	$3.99
Flip-in-File 50	$17.99
Flip-in-File 50 w/lock	$24.99
Flip-in-File (400/800 ROM)	$11.99

AMARAY

50 Disk Tub 5¼''	$9.99
30 Disk Tub 3½''	$8.99

DRIVES

HARD

ALLOY

PC Stor	CALL

IDEAssociates

5 meg Removable/Internal	$1399.00
10 meg Fixed/Internal	$1249.00
15 meg 5 Removable/10 Fixed	$2149.00
25 meg 5 Removable/20 Fixed	$2499.00

I·MEGA

10 meg Bernoulli Box	$1999.00
20 meg Bernoulli Box	$2599.00
5 meg "MacNoulli"	$1499.00

TALLGRASS TECHNOLOGIES

COMMITTED TO MEMORY

12, 25, 35, 50, 80 meg (PC)	from $1499.00

FLOPPY

INDUS

Atari GT	$219.00
C-64 GT	$229.00

µ-SCI

A1.5 Apple	$179.00
A2 Apple	$179.00

MSD

SD1 C-64 Single	$219.00
SD2 C-64 Dual	$469.00

Tandon

320K 5¼'' (PC)	$119.00

TEAC

320K 5¼''	$109.00

CALL TOLL-FREE

COMPUTER MAIL ORDER

MONITORS

AMDEK

300 Green	$129.00
300 Amber	$139.00
310 Amber IBM-Plug	$169.00
Color 300 Composite	$179.00
Color 500 Composite/RGB	$389.00
Color 600 Hi-Res (640x240)	$399.00
Color 700 Hi-Res (720x240)	$499.00
Color 710 Long Phosphor	$579.00
Color 722 IBM Enhanced	$579.00

NAP

12" Amber/Green Composite	$99.99
12" Amber/Green TTL	(ea.) $119.00

NEC

JB 1260 Green	$59.99
JB 1201/1205	(ea.) $99.99
JB 1270 Green	$139.00
JB 1275 Amber	$149.00
JB 1280 G TTL/1285 A TTL	$149.00
JC 1410 RGB	$669.00

PRINCETON

MAX-12E Amber	$189.00
HX-9 9" RGB	$469.00
HX-9E Enhanced	$519.00
HX-12 12" RGB	$469.00
HX-12E Enhanced	$559.00
SR-12 Hi-Res	$599.00
SR-12 P Enhanced	$649.00

TAXAN

115 12" Green Mono	$119.00
116 12" Amber Mono	$119.00
121 Green TTL	$139.00
122 Amber TTL	$149.00
420 Hi-Res RGB (IBM)	$429.00
440 Ultra Hi-Res RGB	$559.00

QUADRAM

8400 Quadchrome I	$479.00
8410 Quachrome II	$429.00
8420 Amberchrome	$179.00
8500 Quad Screen	$1499.00

ZENITH

ZVM 1220/1230	(ea.) $99.99
ZVM 1240 IBM Amber	$149.00
ZVM 130 Color	$269.00
ZVM 131 Color	$249.00
ZVM 133 RGB/Color	$429.00
ZVM 135 RGB/Color	$459.00
ZVM 136 RGB/Color	$599.00

INTERFACES

AST

Multi I/O (Apple II)	$189.00

PRACTICAL PERIPHERALS

Graphcard	$79.99
Seriall Card	$99.99
Microbuffer II +	$169.00
Microbuffer 32K	$189.00

QUADRAM

Microfazer	from $139.00
Efazer (Epson)	from $79.99

Orange Micro

Grappler CD (C64)	$99.99
Grappler + (Apple)	$89.99
Grappler 16K + (Apple)	$159.00

DIGITAL DEVICES

Ape Face (Atari)	$49.99
U-Print A (Atari)	$54.99
U-A16/Buffer (Atari)	$74.99
U-Call Interface (Atari)	$39.99
U-Print C (C64)	$49.99
P-16 Print Buffer	$74.99
U-Print II Apple IIc	$89.99

TYMAC

C64 Epson	$59.99
C64 Okidata	$59.99
C64 Star	$57.99

PRINTERS

Canon

A40	CALL
LBP-8A1 Laser	CALL

CITIZEN

MSP-10 (80 col.)	$279.00
MSP-15 (132 col.)	$399.00
MSP-20 (80 col.)	$349.00
MSP-25 (132 col.)	$509.00

C.ITOH

Prowriter 7500	$219.00
Prowriter 8510P	$299.00
Prowriter 8510 NLQ	$329.00
Prowriter 1550P	$449.00
Starwriter 10-30	$459.00

corona

Lazer LP-300	$2799.00

DIABLO

D25 Daisywheel	$549.00
630-109 Daisywheel	$1749.00
D80IF Daisywheel	CALL

daisywriter

2000	$749.00

EPSON

LX-80, LX-90, FX-85, FX-185 JX-80, DX-10, DX-20, SQ 2000 HS-80, Homewriter 10	CALL

JUKI

6000 Letter Quality	CALL
6100 Letter Quality	CALL
6300 Letter Quality	CALL

NEC

8027 Transportable	$299.00
2000 Series	$699.00
3000 Series	$1099.00
8000 Series	$1449.00
ELF 360	$449.00
Pinwriter 560	$999.00

OKIDATA

84, 182, 192, 193, 2410	CALL
Okimate 10 (Specify C64/Atari)	$189.00
Okimate 20 (IBM)	CALL

OLYMPIA

Needlepoint Dot Matrix	$289.00
Compact RO	$339.00
Compact 2	$369.00

Panasonic

KX1091	$259.00
KX1092	$389.00
KX1093	$479.00

QUADRAM

Quadjet	$399.00
Quad Laser	CALL

SILVER-REED

500 Letter Quality	$279.00
550 Letter Quality	$419.00
770 Letter Quality	$759.00

star

SG-10C (C64 Interface)	$NEW
SB/SD/SG/SR Series	CALL
Powertype Letter Quality	CALL

Texas Instruments

T1850	$529.00
T1855	$799.00
T1865	$1049.00

TOSHIBA

1340 (80 column)	$589.00
P351 (132 column)	$1149.00

PC COMPATIBLES

IBM PC SYSTEMS

Configured to your specification.
Call for Best Price!
IBM-PC, IBM-PC II, IBM-XT, IBM-AT

SOFTWARE FOR IBM

ALPHA

Electric Desk IBM-PC	$199.00

ASHTON-TATE

Framework	$369.00
dBase III	$369.00

BORLAND INTERNATIONAL

Turbo Pascal 3.0	$49.99
Sidekick	$39.99

CENTRAL POINT

Copy II PC-Backup	$29.99

DECISION RESOURCES

Chartmaster	$259.00
Signmaster	$179.00

FOX & GELLER

Quickcode III	$169.00

FUNK SOFTWARE

Sideways	$39.99

Harvard Software Inc.

Harvard Project Manager	$209.00
Total Project Manager	$269.00

Human Edge™

Communication Edge	$99.99
Management Edge	$119.00
Negotiation Edge	$139.00
Sales Edge	$119.00

LIFETREE

Volkswriter Duluxe	$159.00

LIVING VIDEOTEXT

Think Tank	$109.00

Lotus

Symphony	$439.00
1-2-3	$309.00

MECA SOFTWARE

Managing Your Money	$109.00

MICROSTUF

Crosstalk XVI	$89.99
Microstuf Remote	$89.99

MICRORIM

R:Base 4000	$259.00
R:Base 5000	$399.00
Clout 2.0	$129.00

MicroPro

WordStar 2000	$249.00
WordStar 2000 +	$309.00
WordStar Professional	$299.00

MICROSOFT.

Word	$239.00
Mouse	$139.00
Flight Simulator	$39.99
MultiPlan	$129.00

MultiMate

Multi Mate Word Proc	$249.00

NOUMENON

Intuit	$69.99

NORTON

Norton Utilities 3.0	$59.99

Peachtree Software

Peachtext 5000	$169.00
PeachPack (GL/AP/AR)	$219.00

pfs:
IBM

Access	$79.99
Write/Graph/File/Plan	(ea).$79.99
Report	$74.99
Proof	$59.99

Professional Software

Wordplus-PC w/Boss	$249.00

ROSESOFT

Prokey	$89.99

THE SOFTWARE GROUP

Enable	$339.00

SATELLITE SYSTEMS

Word Perfect 4.0	$219.00

SORCIM/IUS

Accounting AP/AR/GL/INV/OE	(ea.) $295.00
SuperCalc III	$195.00
EasyWriter II System	$195.00
Super Project	$195.00

SPI

Open Access	$379.00

THOUGHTWARE

Trigger	$289.00
Sell, Sell, Sell Training	$299.00
Application	$179.00

AT&T

Safari (7300)	CALL
6300	CALL

corona

PPC400 Dual Portable	$1289.00
PPCXT 10 meg Portable	$1989.00
PC40022 Dual Desktop	$1389.00
PC400-HD2 10 meg	$1989.00

ITT

ITT X-TRA 256K, 2 Drive System	CALL
256K,10 meg Hard Drive System	CALL
XP5, 20 meg	CALL

SANYO

MBC 550-2 Single Drive	$699.00
MBC 555-2 Dual Drive	$969.00
MBC 511 10 meg	$1799.00
MBC 675 Portable	CALL
MBC 880 Desktop	CALL

ZENITH

PC-158-21 Single Desktop	CALL
PC-158-52 Dual Desktop	CALL
PC-158-53 10 meg Desktop	CALL
PC-161-21 Single Portable	CALL
PC-161-52 Dual Portable	CALL
Z-200 (AT)	CALL
171 (Portable)	CALL
138 (Transportable)	CALL
148 (DeskTop)	CALL

(NEW)

MULTIFUNCTION CARDS

AST

Six Pack Plus	$239.00
Mega Plus II	$269.00
I/O Plus II	$139.00
Advantage-AT	$399.00
Graph Pak	$599.00
Monograph Plus	$399.00
Preview Mono	$299.00
PC Net Cards	$379.00
5251/11 On-line	$799.00
5251/12 Remote	$579.00
3780 Emulation Card	$639.00
BSC Bisync	$489.00

dca

IRMA 3270	$879.00
IRMA Print	$999.00

EVEREX

Color Card (Graphics Edge)	$299.00
Magic Card II	CALL
Edge Card	$299.00

HERCULES

Graphics	$299.00
Color	$159.00

IDEAssociates

IDEAmax - ZPR, 64K, C, S, P.	$229.00
IDEAmini - YPR, C, S, P	$189.00
IDEAminimax - MPR 128K	$229.00
IDEAshare Software	$219.00
IDEA 5251	$699.00

MYLEX

The Chairman	$479.00

PARADISE

Modular Graphics Card	$279.00
Multi Display Card	$229.00
Five Pack C, S	$159.00

PERSYST

Bob Board	$389.00

TECMAR

Captain - 64	$199.00
Captain Jr. 128K	$279.00
Graphics Master	$469.00

QUADRAM

Quadport-AT	$119.00
Quadmeg - AT (128K)	$349.00
The Gold Quadboard	$449.00
The Silver Quadboard	$239.00
Expanded Quadboard	$219.00
Quad 512 +	$229.00
Liberty	$309.00
QuadSprint	$499.00
QuadLink	$399.00
Quadcolor 1	$199.00
QuadJr. Expansion Chassis	$469.00
Expansion Chassis Memory	$199.00
Chronagraph	$79.99
Parallel Interface Board	$64.99

1-800-233-8950

result is that the BAM and directory sectors can be reassigned to different buffers during a single SAVE@. We have found the BAM and directory sectors in every drive buffer after different SAVE@ commands. We have found copies of the current directory sector in two different buffers, one an old sector and one properly updated, but the wrong one had been written to the disk. Somehow, the pointers to the BAM and directory sectors are not properly accounted for. Which buffer is stolen by STLBUL depends on prior buffer usage and the values stored in LRUTBL,Y ($FA,Y), the least recently used table. It appears that STLBUF updates all pointers except LRUTBL,Y. This means that multiple calls to STLBUF may steal the wrong buffer—in this case the wrong buffer to steal is the BAM!

The BAM is stored in the drive in one of the buffers. STLBUF should not steal the drive 0 BAM, but should instead take back the unused buffer incorrectly assigned to drive 1. It never steals the drive 1 BAM, buffer 3 at $600, because STLBUF cannot take a buffer which encountered a drive error. Remember that an internal DRIVE NOT READY error did occur, because there is no drive 1!

To test this, we copied into EPROM an altered version of DOS with STLBUF modified to allow stealing a buffer with this error. This allowed the phantom drive 1 BAM buffer to be freed, and the SAVE@ bug did not strike during tests with this modified DOS.

If this buffer-stealing occurs, why does SAVE@ work most of the time? We must dig deeper into DOS to answer this question. When a file is opened and blocks (or sectors) are written to a disk, the BAM is *not* directly updated in the drive memory. Instead, a BAM image for each of two tracks is stored at BAM ($2A1–$2B0). Each time a new block is allocated by WUSED ($EF90), it is recorded in the BAM image. When a new track is tested for free sectors, DOS checks if it has a BAM image for it. If not, it calls SWAP ($F05B), which first updates the BAM with the BAM image from the next-to-last track, copies the new track's BAM map into the BAM image, and then zeros that track in the BAM. This all works perfectly—most of the time.

After the last file sector is written to the disk, the BAM still has not been written to the disk. In fact, the BAM in the drive is wrong because it has not yet been updated from the BAM images. When a file is closed, the disk directory is closed, CLSDIR ($DBA5), by reading in the file's directory sector, testing for a replace file type, and then rewriting it to the disk. MAPOUT ($EEF4) is called to read the BAM off the disk, if necessary, and to then update it from the BAM images by calling PUTBAM ($F0A5). The updated BAM is then written back to the disk.

During a SAVE@ command, DOS performs an additional step after reading the directory sector. The file type is designated as replace, so DELFIL ($C87D) is called to delete the original version of the file from the BAM. It reads in the BAM if necessary when freeing the first sector, FRETS ($EF5F), and then proceeds to trace through the file and delete sectors in the BAM images. The BAM is then written to the disk.

Bungled BAM

Normally this procedure works correctly. But havoc results if the BAM buffer is stolen while the file is being closed. This can happen during a SAVE@ command because DELFIL requires two additional buffers. The BAM can be stolen at different points during the procedure, depending on which buffers were previously used—which, in turn, depends on the number of sectors in the file and the tracks on which it is stored.

After the BAM is stolen, it is read back in when needed and updated from the BAM images. Only *two* tracks can be updated, however, since there are only two images. If more than two tracks have been accessed by SAVE@, the BAM may *not* be correctly updated. A track could be updated correctly, left unchanged, or fully allocated, depending on when the BAM was stolen.

If extra sectors are allocated, the BAM is incorrect, but no permanent harm is done. A validate command will cure the problem. If sectors are not allocated, then a new file will be saved on top of the old file's sectors. In the example program listed in Part 1, a fourth SAVE@ command would result in the file being written on top of the old file's first four sectors, and then the whole new file would be scratched—a tragic result, indeed.

Based on these findings, we recommend that you avoid the SAVE@ command when direct access channels to the drive are open or if you don't always specify the drive number in disk commands. You should also avoid SAVE@ when using programs or cartridges intended to speed up access on the 1541 disk drive. These programs often reserve internal drive buffers and may cause problems even if the drive number is specified. If you're using the DOS Wedge, we recommend issuing a >UI or >UJ command before each SAVE@ command to be sure all the buffer pointers are reset. Many word processors also allow you to send these commands to the drive. Otherwise, the drive should be turned off and then on before using SAVE@. (On the SX-64, press the drive reset button.)

During our studies we found several other minor bugs in DOS V2.6, including the subroutine which puts the value 2 at the drive memory location $197. This bug does no harm since it affects a normally unused section of drive memory. However, we have found it can affect DOS routines downloaded into the drive. There may be other bugs or quirks which we have not found, so the Commodore DOS controversy may never be fully closed.

In Part 1 of this article, there was a minor error in the example to illustrate the problems caused by not specifying a drive number (using the DOS Wedge program). The article stated that giving the Wedge command >$TEST results in a blinking disk error light if the file TEST does not exist on the disk. Actually, >$TEST does not cause the error light to blink unless it is used twice in succession. The first >$TEST correctly prints a blank directory of drive 0, but leaves the 1541 looking for the nonexistent drive 1 so that the second >$TEST results in the DRIVE NOT READY error described last month. ©

Atari REMover

Jeff Stefanski

This short BASIC utility automatically removes REM statements from programs. It runs on the Atari 400/800, XL, and XE series computers.

Many programmers use REMark statements to document how their programs work—a good programming practice. Once the program is finished and debugged, however, the REMs can be deleted to save memory and slightly increase execution speed (although it's a good idea to save a version with the REMs in case you have to make modifications later). Scanning through a program and deleting REM statements one by one has always been a tedious job. But it's easy with "Atari REMover."

This short routine automatically removes the REMs from BASIC programs, leaving everything else intact. Type in Atari REMover as listed below, then save the program by LISTing it to disk or cassette. *You must store the program with LIST, rather than SAVE.* (Example: LIST "C:" for cassette or LIST"D:*filename.ext*" for disk.) Since Atari REMover deletes itself from memory after running, be sure to save a copy before using it for the first time.

REMover is easy to use. First load the program from which you want to delete the REMs. Then append REMover to the end of the first program by ENTERing it from disk or cassette. (Example: ENTER "C:" for cassette or ENTER"D:*filename.ext*" for disk.) Type GOTO 32000 and press RETURN to activate REMover. The routine looks through your program and deletes each line that contains nothing but a REM statement. If a multistatement line ends with REM, the REM portion is cut off and the line is reentered.

It may take a while for REMover to delete all the REMs in a large program, so be patient. After the job is done, REMover deletes itself.

Note that REMover uses line numbers above 32000. If your program uses the same line numbers, renumber it before using this routine. If your program contains a GOTO or GOSUB to a REM line (poor programming practice in any case), change the line reference yourself after using REMover.

Atari REMover

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

```
00 32000 CLR :GRAPHICS 0:STM
         TAB=PEEK(136)+PEEK(
         137)*256:POKE 82,2:
         POKE 83,39:DIM L$(1
         14)
CL 32001 LINE=PEEK(STMTAB)+P
         EEK(STMTAB+1)*256
AK 32002 IF LINE=32000 THEN
         32015
FA 32003 PRINT CHR$(125):POS
         ITION 2,6:LIST LINE
HH 32004 LOCATE 3+LEN(STR$(L
         INE)),7,A:LOCATE 4+
         LEN(STR$(LINE)),7,B
         :LOCATE 5+LEN(STR$(
         LINE)),7,C
CA 32005 IF A=82 AND B=69 AN
         D C=77 THEN 32009
OP 32006 L=1:FOR X=7 TO 9:FO
         R Y=2 TO 39:LOCATE
         Y,X,M:L$(L)=CHR$(M)
         :L=L+1:NEXT Y:NEXT
         X
GI 32007 FOR X=1 TO 110:IF L
         $(X,X+3)=":REM" THE
         N 32012
ND 32008 NEXT X:STMTAB=STMTA
         B+PEEK(STMTAB+2):GO
         TO 32001
KH 32009 PRINT CHR$(125):POS
         ITION 2,6:PRINT LIN
         E
NC 32010 POSITION 0,0:POKE 8
         42,13:POSITION 2,7:
         PRINT "CONT":POSITI
         ON 2,4:STOP
LM 32011 POKE 842,12:GOTO 32
         001
IM 32012 PRINT CHR$(125):POS
         ITION 2,6:PRINT L$(
         1,X-1):PRINT "CONT"
FC 32013 POSITION 0,0:POKE 8
         42,13:POSITION 2,4:
         STOP
JF 32014 POKE 842,12:STMTAB=
         STMTAB+PEEK(STMTAB+
         2):GOTO 32001
BI 32015 PRINT CHR$(125):POS
         ITION 2,6:FOR X=320
         00 TO 32016:PRINT X
         :NEXT X:PRINT "PRIN
         T CHR$(125):POKE 84
         2,12:END"
BI 32016 POKE 842,13:POSITIO
         N 2,2:STOP           ©
```

Plus/Term
For Commodore 1660 Modem

Mark Wood

By adding a few lines to COMPUTE!*'s popular "Plus/Term" program, you can use it with a Commodore 64 and the Commodore 1660 direct-connect modem, dialing and hanging up under program control.*

"Plus/Term," published in COMPUTE!, February 1985 (and in *Telecomputing on the Commodore 64*, COMPUTE! Books), is an excellent terminal program, offering an 80-column display mode (with "Screen-80," COMPUTE!'s GAZETTE, September 1984) and many other desirable features. However, since my Commodore 1660 is a direct-connect modem which doesn't allow manual dialing, I had no way to use Plus/Term. My solution was to add auto-dialing and hang-up routines to the program.

To include these new features in Plus/Term, you'll first need to type in the original program. Then type in the additional lines listed below. Once you're finished, resave the program (perhaps with a different name to distinguish it from the original Plus/Term).

Plus/Term now offers two additional options: You can dial a number from within the program (press D) or hang up the line whenever you want (H). After selecting Dial, type in the number you want, then choose between rotary and tone dialing, depending on which service you have on your phone system. You may add spaces or dashes between the numbers if you like, but they're not necessary. If you press RETURN without entering a number, or enter a string that contains no numbers, Plus/Term simply returns you to terminal mode. Rotary dialing is simulated by rapidly disconnecting and reconnecting the line the correct number of times for each number. Tone dialing signals are generated with the 64's SID chip.

Plus/Term Modifications

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

```
10 DATA152,85,74,6Ø,117,77,168
   ,44,152,85,168,44,161,94,16
   8,44,117,77,85,49   :rem 156
20 DATA152,85,85,49,161,94,85,
   49,117,77,15Ø,54,152,85,15Ø
   ,54,161,94,15Ø,54   :rem 14Ø
271 PRINTA$"H. HANG UP";
                        :rem 68
272 PRINTA$"D. DIAL"; :rem 152
281 IFM1$="H"THENPOKE56577,(PE
    EK(56577)OR32)      :rem 8
282 IFM1$="D"THEN2Ø2Ø :rem 133
685 DIMDIG(15):FORWXE=ØTO9:FOR
    COL=1TO4:READSID(WXE,COL):
    NEXT:NEXT           :rem 33
2Ø2Ø ZX$="":AZ$="":INPUT"{CLR}
     {6 DOWN}{2 RIGHT}NUMBER T
     O DIAL";AZ$:IFLEN(AZ$)=ØT
     HEN176Ø           :rem 12Ø
2Ø3Ø FORJ=1TOLEN(AZ$):G$=MID$(
     AZ$,J,1):IFG$=>"Ø"ANDG$<=
     "9"THENZX$=ZX$+G$:rem 19Ø
2Ø4Ø NEXT:IFLEN(ZX$)=ØTHEN176Ø
                        :rem 62
2Ø5Ø PRINT"{DOWN}{2 SPACES}
     {RVS}R{OFF}OTARY OR {RVS}
     T{OFF}ONE?"       :rem 232
2Ø6Ø GETTY$:IFTY$<>"R"ANDTY$<>
     "T"THEN2Ø6Ø       :rem 2Ø8
2Ø7Ø FORWXE=1TOLEN(ZX$):DIG(WX
     E)=VAL(MID$(ZX$,WXE,1)):N
     EXT               :rem 154
2Ø8Ø POKE56579,(PEEK(56579)OR3
     2):POKE56577,(PEEK(56577)
     AND223):FORJ=1TO6ØØ:NEXT
                       :rem 127
2Ø9Ø PRINT"{3 DOWN}{5 RIGHT}DI
     ALING...":IFTY$="R"THEN21
     6Ø                 :rem 71
21ØØ POKE54296,15:POKE54276,16
     :POKE54283,16:POKE54277,Ø
     :POKE54284,Ø       :rem 46
211Ø POKE54278,24Ø:POKE54285,2
     4Ø:POKE54295,Ø:FORWXE=1TO
     LEN(ZX$)          :rem 236
212Ø POKE54272,SID(DIG(WXE),1)
     :POKE54273,SID(DIG(WXE),2
     )                  :rem 13
213Ø POKE54279,SID(DIG(WXE),3)
     :POKE5428Ø,SID(DIG(WXE),4
     )                  :rem 23
214Ø POKE54276,17:POKE54283,17
     :FORDEL=1TO75:NEXT
                       :rem 129
215Ø POKE54276,16:POKE54283,16
     :FORDEL=1TO75:NEXT:NEXT:P
     OKE54296,Ø:GOTO176Ø:rem 9
216Ø FORWXE=1TOLEN(ZX$):VA=DIG
     (WXE):IFVA=ØTHENVA=1Ø
                        :rem 19
217Ø FORCL=1TOVA:POKE56577,(PE
     EK(56577)OR32):FORDEL=1TO
     26:NEXT            :rem 78
218Ø POKE56579,(PEEK(56579)OR3
     2):POKE56577,(PEEK(56577)
     AND223):NEXT      :rem 11Ø
219Ø FORDEL=1TO25Ø:NEXT:NEXT:G
     OTO176Ø            :rem 99
```

©

Atari Animation With P/M Graphics

Part 3

Robert Powell

Animation with Atari player/missile graphics involves a number of programming techniques. Parts 1 and 2 in this series showed how to design a player/missile object, display it on the screen, control its color, and animate it horizontally. This month's article shows one method of vertical animation without resorting to machine language.

As we saw in Part 2, horizontal animation with player/missile graphics is quite simple: A single POKE into the horizontal position register moves the P/M strip to any place on the screen. Last month, Program 2 demonstrated how easy it is to move a player horizontally with a joystick.

Vertical animation, however, is not so simple. There is no such thing as a vertical position register which corresponds to the horizontal position register. Since P/M objects are strips of memory taller than the screen, a vertical register wouldn't make sense—you couldn't see the strip moving, anyway. Instead, to achieve vertical animation your program must move the P/M shape you've designed through the strip of player memory.

Program 1 below is a slightly modified version of last month's program which defined player 0 as a happy face. It shows how a shape can be moved through P/M memory with a FOR-NEXT loop in BASIC. Plug a joystick into port 1 to control the player's vertical movement. As you'll see, vertical animation in BASIC is disappointingly slow. BASIC just isn't fast enough to move the player shape through memory without a rippling "inchworm" effect.

There are two solutions to this problem. One is to write a machine language subroutine for vertical animation. Over the past five years, COMPUTE! has published several such routines which require no knowledge of machine language—you just drop the routine into your BASIC program and call it with a USR statement. The back issues are now out of print, but these and various other routines for vertical motion are discussed in several books (*COMPUTE!'s First Book of Atari Graphics; Mapping the Atari; COMPUTE!'s First Book of Atari Games;* and *COMPUTE!'s Second Book of Atari Graphics*).

Another solution which avoids machine language yet is comparable in speed takes advantage of BASIC's fast string-manipulation routines. We'll cover this method here.

A Few Strings Attached

The string-animation technique depends on making the computer think that a BASIC string is located in the P/M memory area, rather than in the usual memory area where the computer stores strings. Therefore, when you redefine the string, P/M memory changes—and the P/M object changes along with it. You can use this technique to rapidly change the shape of a player, move it vertically, or erase it off the screen.

Program 2 shows how to fool the computer into thinking a long string is located in the P/M memory area. A full explanation is beyond the scope of this article; however, even if you don't understand this technique, you can use it in your own programs by copying lines 10–100. This module adjusts itself for single- or double-line P/M resolution when you change the statement in line 20. Set MODE=1 for single-line resolution, or MODE=2 for double-line resolution.

When you run this program, several things become apparent. First, it eliminates the usual delay

Player/Missile Addresses Using String Animation

	Double-Line Resolution	Single-Line Resolution
Missiles 0–3	1–128	1–256
Player 0	128–256	256–512
Player 1	256–384	512–768
Player 2	384–512	768–1024
Player 3	512–640	1024–1280

caused by using a FOR-NEXT loop to clear out P/M memory with zeros. Instead, the three statements in line 100 clear out P/M memory instantly. This trick works by setting A$ to zeros after lines 10–90 fool the computer into thinking that A$ coincides with P/M memory.

Second, the program does not define the player shape by POKEing into P/M memory, as does Program 1 and last month's programs. Instead, the bytes which form the player shape in line 120 are read into a string (B$) in line 110. This is the key to the string-animation technique. Since the computer thinks that A$ overlays P/M memory, the statement in line 130 copies the player shape in B$ into the middle of the player 0 memory area. This places the shape at midscreen.

With a statement like A$(Y,Y+LEN(B$))=B$, you can instantly change the player's vertical position. For an example of vertical animation, replace lines 130 and 140 in Program 2 with the following lines:

```
130 FOR Y=256 TO 512
140 A$(Y,Y+LEN(B$))=B$
150 NEXT Y
160 GOTO 130
```

It's a convincing demonstration that fast vertical motion can be easily achieved in BASIC using strings.

Self-Erasing Players

If you look closely at the player shape bytes in line 120, you'll notice that a pair of zeros precedes and follows the series of numbers. Ordinarily, it doesn't make sense to see zeros in player shape data, because zeros show up blank on the screen. But these zeros have a special purpose. As the player shape moves through P/M memory, it would leave a trail of itself on the screen unless you erased it after every movement. Although it would be easy to erase the player shape by filling B$ with zeros (using a formula like the one in line 100), this extra step would slow down the animation by a fraction of second. By tacking a zero onto each end of the player data, the shape erases itself as it moves.

In this case, two zeros surround the player data. This allows even faster vertical motion by moving the player shape *two* steps at a time. To see this in action, add the above changes to Program 2 with this alteration to line 130:

```
130 FOR Y=256 TO 512 STEP
    2
```

Now change STEP 2 to STEP 15. As you can see, you can have as many shapes displayed in the vertical band as will fit.

Another important advantage of string-animation is that you can store several different shapes in different strings (such as B$, C$, D$, and so on). You can instantly flip between the shapes simply by reassigning A$, as in A$(Y,Y+LEN(D$))=D$.

What About Diagonals?

Once you learn how to move P/M objects horizontally and vertically, it's easy to animate them diagonally as well. Just combine a horizontal step with each vertical step, interweaving them to achieve a diagonal path.

For an example, start with Program 2 and add these changes:

```
130 FOR Y=256 TO 511
140 A$(Y,Y+LEN(B$))=B$:PO
    KE 53248,Y-256
150 NEXT Y
160 GOTO 130
```

If you experiment with these programs, you should be able to take it from here. All these examples use player 0, but the other players and missiles can be used in a similar manner. Just calculate the vertical screen position by figuring where A$ overlaps the appropriate player/missile area, then position the player shape data at that point in A$. (Refer to the accompanying table for a guide.)

Trying drawing a background screen with PLOT and DRAWTO, then move your players above or beneath it. Also, although P/M graphics are commonly used for games, try using these techniques to add interest and variety to your text programs as well. You can turn players or missiles into thin vertical lines to delineate data columns, or change them into cursors that change color to signal for input. This three-part series merely covers the basics—there's a lot more to Atari P/M graphics, such as priority registers and collision registers. The possibilities are endless.

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

Program 1: Vertical Movement With FOR-NEXT

```
MF 1Ø  POKE 1Ø6,PEEK(1Ø6)-8
NF 2Ø  POKE 54279,PEEK(1Ø6)
HD 3Ø  GRAPHICS Ø:SETCOLOR 2,
       Ø,Ø
CN 4Ø  PMBASE=PEEK(1Ø6)*256
ML 5Ø  POKE 559,62
PM 6Ø  POKE 53277,3
DP 7Ø  POKE 7Ø4,68:POKE 7Ø5,1
       98:POKE 7Ø6,168:POKE 7
       Ø7,148
PA 8Ø  POKE 53248,16Ø:POKE 53
       249,17Ø:POKE 5325Ø,18Ø
       :POKE 53251,19Ø
NB 9Ø  FOR X=PMBASE+1Ø24 TO P
       MBASE+2Ø48:POKE X,Ø:NE
       XT X
LB 95  VERTICAL=PMBASE+1152
KD 1ØØ RESTORE :FOR X=1 TO 1
       3
OP 11Ø READ A
AM 12Ø POKE VERTICAL+X,A
CL 13Ø NEXT X
LP 14Ø DATA Ø,24,6Ø,126,9Ø,2
       19,255,219,195,1Ø2,6Ø
       ,24,Ø
CF 15Ø S=STICK(Ø)
JK 16Ø IF S=13 THEN VERTICAL
       =VERTICAL+1:GOTO 1ØØ
JO 17Ø IF S=14 THEN VERTICAL
       =VERTICAL-1:GOTO 1ØØ
GD 18Ø GOTO 1ØØ
```

Program 2: Vertical Movement With Strings

```
GL 1Ø  DIM A$(1),B$(15)
OA 2Ø  MODE=2:REM MODE=2 FOR
       DOUBLE-RES, MODE=1 FOR
        SINGLE-RES
FH 3Ø  PMPAGE=PEEK(1Ø6)-4*MOD
       E:POKE 1Ø6,PMPAGE:POKE
        54279,PMPAGE
CE 4Ø  GRAPHICS Ø:SETCOLOR 2,
       Ø,Ø:PMBASE=256*PMPAGE
EH 5Ø  POKE 559,3Ø+16*MODE:PO
       KE 53277,3:POKE 53248,
       16Ø:POKE 7Ø4,68
HH 6Ø  VTAB=PEEK(134)+256*PEE
       K(135):ATAB=PEEK(14Ø)+
       256*PEEK(141)
CD 7Ø  OFFSET=PMBASE+384*MODE
       -ATAB:REM FIND DISTANC
       E FROM ATAB START TO P
       LAYER ZERO START
FB 8Ø  HI=INT(OFFSET/256):LO=
       OFFSET-256*HI:L=64Ø*MO
       DE:HL=INT(L/256):LL=L-
       256*HL
AN 9Ø  POKE VTAB+2,LO:POKE VT
       AB+3,HI:POKE VTAB+6,LL
       :POKE VTAB+7,HL
JD 1ØØ A$(1)=CHR$(Ø):A$(64Ø*
       MODE)=CHR$(Ø):A$(2)=A
       $
BB 11Ø FOR I=1 TO 15:READ A:
       B$(I,I)=CHR$(A):NEXT
       I
HF 12Ø DATA Ø,Ø,24,6Ø,126,9Ø
       ,219,255,219,195,1Ø2,
       6Ø,24,Ø,Ø
KM 13Ø A$(19Ø*MODE,19Ø*MODE+
       LEN(B$))=B$
GD 14Ø GOTO 14Ø              ©
```

Amiga's *Amazing* Graphics

Charles Brannon, Program Editor

Commodore's Amiga presents programmers with more graphics features than ever before—both an exciting prospect and a bewildering abundance. This overview covers the fundamentals of the Amiga's graphics capabilities and shows how they differ from those on previous personal computers.

Graphics make the Amiga special. Although the Amiga's other features—such as its stereo sound, high-speed 68000 microprocessor, built-in 880K disk drive, and multitasking operating system—are certainly noteworthy, it's the graphics that first catch your eye. The 4,096 color variations allow nearly seamless transition between colors; the 640 × 400 high-resolution bitmap mode is close to broadcast TV quality; and the custom chips permit fast, complex animation. The Amiga is a machine for the artist in all of us.

Making the most of these features requires programmers to master some new techniques, however. There are some important differences between the way the Amiga handles graphics and the methods used on previous personal computers. Of course, there are many similarities, too.

Mixing A Rainbow

Understanding any computer's graphics is easier if you know some background about video displays. In any monitor or TV, video images are electronically painted by electron guns on the inside of the CRT (*cathode ray tube*, commonly known as the picture tube). From our point of view when looking at the screen, three electron beams sweep left to right, top to bottom, across the inside of the CRT. The CRT is coated with special phosphors that glow either red, green, or blue when hit by the stream of particles from the electron guns. Each phosphor dot can glow bright, dim, dark, or anywhere in between.

Once painted on the screen, the video image quickly fades away, so the electron beams repeat the cycle to draw a new frame 60 times per second. This *refresh rate* is more than fast enough to fool our eyes into seeing motion when the video images are changing each frame, as they are with TV shows and animated computer graphics.

Unlike most computers, the Amiga does not limit you to a fixed set of colors. Instead, you mix three primary colors—red, green, and blue—to create your own custom colors. Each primary color has 16 luminance, or brightness, levels, from 0 (no color) to 15 (very bright). This makes up to 4,096 combinations possible (16 * 16 * 16 = 4,096).

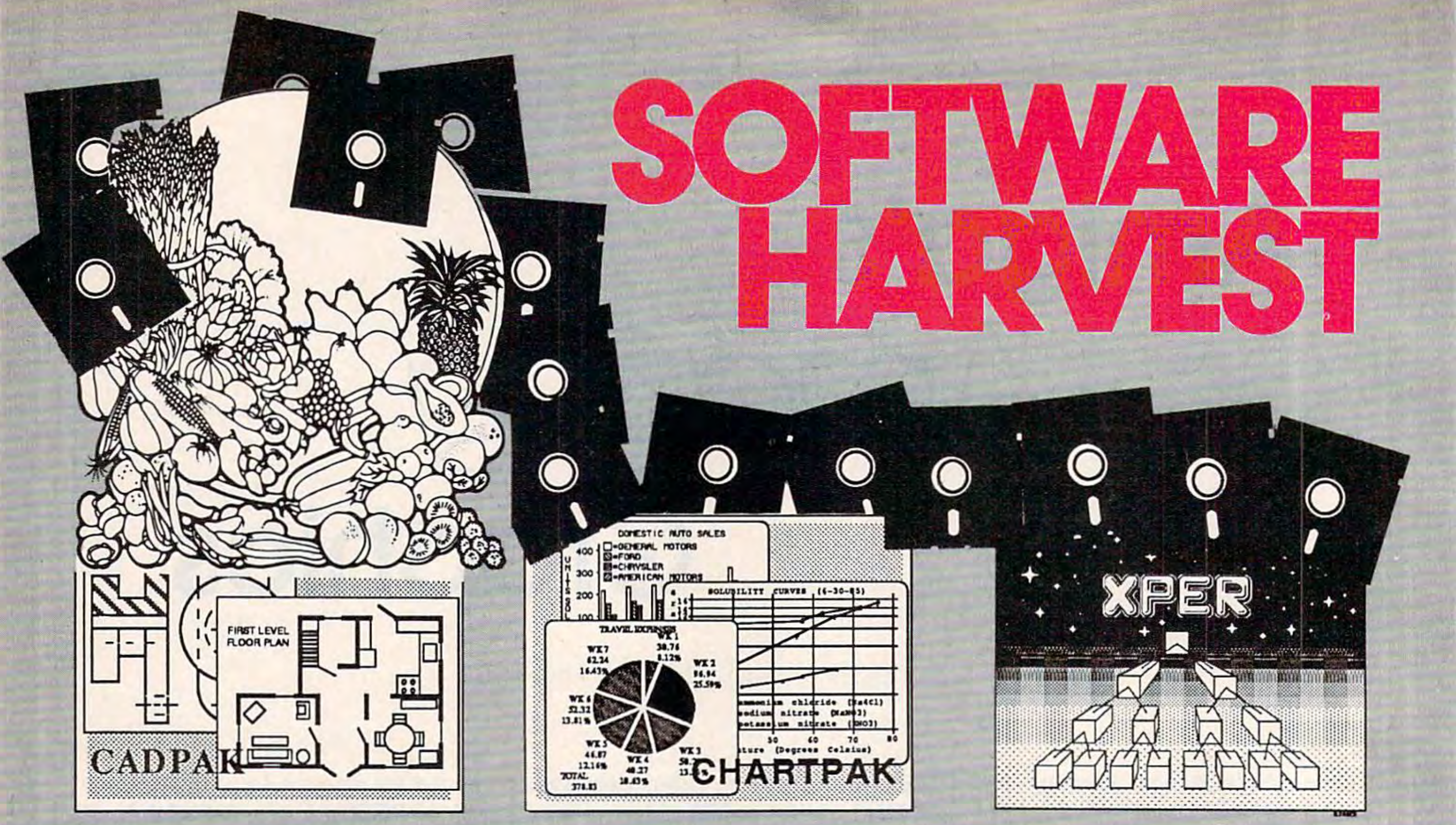
To display all these colors, the Amiga requires a special monitor called *analog RGB* (red-green-blue). The Amiga also works with another type of RGB monitor, called *digital RGB*, but these devices can display only 16 colors. (Digital RGB monitors are the type used with IBM computers and the Commodore 128.) If you plug the Amiga into a regular TV via its built-in RF modulator, 3,616 colors can be displayed.

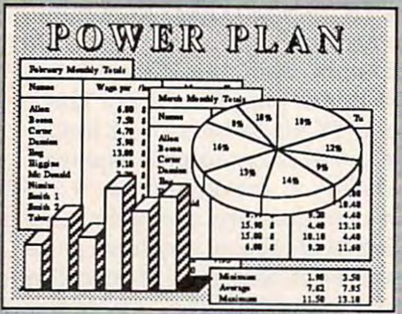
With a maximum of 4,096 color combinations available using analog RGB, almost any hue can be closely approximated. Most colors can be separated into red, green, and blue components. Because video images appear as transmitted, not reflected light, the red-blue-yellow primary color mixing you may have learned does not always apply. For example, red, blue, and green combine to form white, not dark brown. To get brown you'd need to combine red and green to get a greenish red color which appears to be brown. Turning up the brightness of green and red gives yellow. Combine dark red and dark blue to get violet. Bright red and dark blue yields a pastel shade of purple.

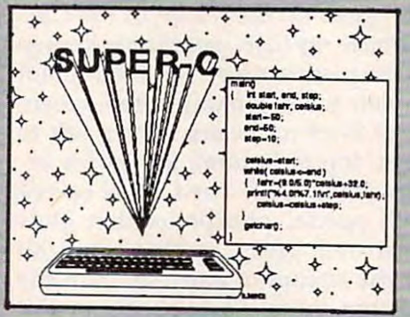
Color Indirection

In most of the Amiga's graphics modes, you cannot display all 4,096 colors simultaneously. Instead, you're limited to a palette of 16 or 32 colors, depending on the mode. However, you can choose which of the 4,096 colors will be available in the palette.

SOFTWARE HARVEST

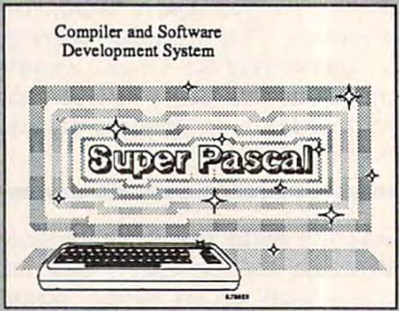






```
main()
{   int start, end, step;
    double fahr, celsius;
    start=-50;
    end=60;
    step=10;

    celsius=start;
    while( celsius<=end )
    {   fahr=(9.0/5.0)*celsius+32.0;
    printf("%4.0f%7.1f\n",celsius,fahr);
        celsius=celsius+step;
    }
    getchar();
}
```



Cadpak Enhanced Version - Graphics design and drawing package. Create detailed designs with dimensioning, scaling, text, rotating, more. Use with or without lightpen. Hardcopy. $39.95

PowerPlan - *"one of the best programs ever written for the C-64"* Commodore Magazine, Nov 1985. Complete spreadsheet with builtin graphics. Cell formatting, protection, windowing, HELP. $39.95

Chartpak - Make professional quality pie, bar and line charts/graphs directly from your data. Instantly draw or redraw charts to your specifications. Statistics capabilities. Hardcopy. $39.95

Super C - Complete **C** language development system. Enhanced full-screen editor, compiler, linker, library and extensive handbook. Used by thousands in schools and industry. $79.95

Xper - First expert system for C-128 and C-64. Ordinary data bases play back facts, but XPER's *knowledge base* help you identify, diagnosis and select - in short helps make decisions. $59.95

Super Pascal - Complete Pascal language development system. Full-screen editor, compiler, integrated assembler, graphics toolkit, fast DOS (3X faster than 1541) and handbook. $59.95

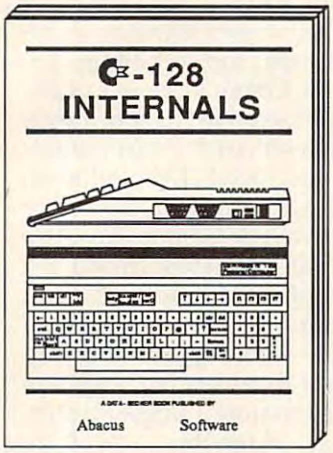
Call **now**, for the name of your nearest dealer. Or order directly from ABACUS with your Mastercard, VISA or AMEX card. Add $4.00 for postage and handling. Foreign orders add $10.00 per product. Other software and books are also available. Call or write for **free** catalog. Dealer inquiries welcome - over 1200 dealers nationwide.

Abacus Software

P.O. Box 7211 Grand Rapids, MI 49510
Tel. 616/241-5510 Telex 709-101

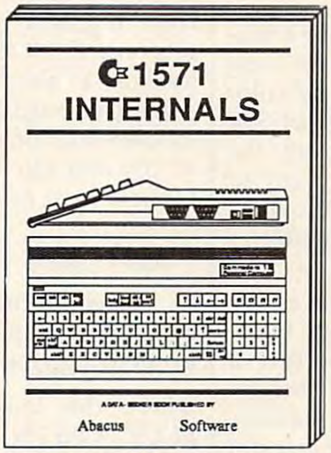
AUTHORATATIVE, COMPREHENSIVE, DEFINITIVE

booksbooksbooks

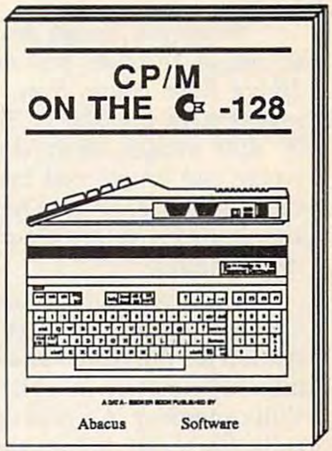


With ROM listings
Avail. Nov. $19.95

For the programmer
Avail. Nov. $19.95

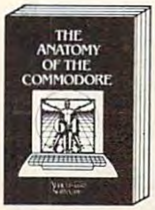


With ROM listings
Avail. Dec. $19.95

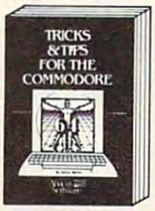


Especially for the '128
Avail. Dec. $19.95

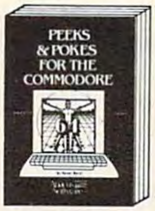
...and more books.



For two years a best seller. C-64 internals w/ROM listings. $19.95



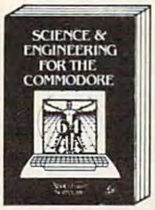
Favorite among programmers. 75,000+ sold worldwide. $19.95



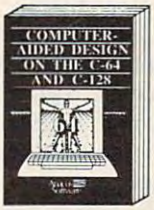
Quickhitting, easy-to-use routines for every C-64 owner. $14.95

All time best seller. Revised & expanded. ROM listings. $19.95

Brand new! Complete maintenance and repair procedures. $19.95

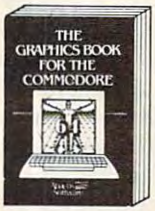


Intro to computers and world of science. Real examples. $19.95

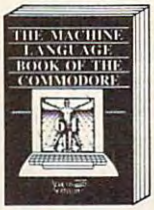


CAD techniques using C-128/C-64. Many program examples. $19.95

Learn to design and write your own compiler. With examples. $19.95

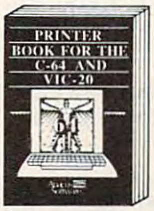


Most in depth treatment available. Dozens of techniques. $19.95

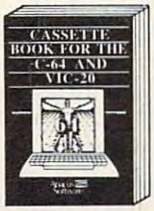


Intro to machine language geared to C-64. Assembler incl. $14.95

Techniques never covered before interrupts, controllers, etc. $14.95



All about using printers and '64. Graphics, text, interfaces. $19.95



A must for cassette owner. Hi speed cassette system. $19.95

Write your own adventures. Learn strategy, motivation. $14.95

Dozens of interesting pro-ects for your '64. Easy to read. $12.95

OTHER BOOKS COMING SOON!

OPTIONAL DISKETTES are also available for each of our book titles. Each diskette contains the programs found in the book to save you the time of typing them in at the keyboard. Price of each diskette is $14.95.

SAVE! Thru Nov 30th, receive the optional diskette for 1/2 price when ordered at the same time as any book. Only $7.50

Call **now**, for the name of your nearest dealer. Or order directly from ABACUS with your MC, VISA or AMEX card. Add $4.00 for postage and handling. Foreign orders add $6.00 per book. Other software and books are also available. Call or write for **free** catalog. Dealer inquiries welcome - over 1200 dealers nationwide. Call **616 / 241-5510**

Abacus Software

P.O. Box 7211 Grand Rapids, MI 49510
Phone 616/241-5510 Telex 709-101

The colors in the Amiga palette are determined by 32 memory registers. Each register is 12 bits wide (1½ bytes), the number of bits needed to hold a number from 0 to 4,095.

When the Amiga draws a video image, the dots that make it up derive their colors from the color registers. In the 320 × 200 or 320 × 400 modes, each dot on the screen can be colored from any of the 32 color registers. Therefore, 32 simultaneous colors are possible in these modes.

Some computers, such as the Commodore 64, store color information for the screen in a section of memory known as *color memory.* Color memory is a grid of memory cells. Each cell defines the color of an 8 × 8 pixel zone. The number in color memory is a number representing 1 of 16 fixed colors. Other computers, such as the Atari machines, store color information by another method known as *color indirection.* The Amiga uses the latter technique. The number representing the dot on the screen does not encode the color directly, but instead selects a particular color register. The dot gets its color indirectly through the color register. If you change the color register, everything drawn with that register instantly changes.

Color indirection is extremely powerful. Unique glowing effects are possible by cycling between all the colors at high speed. You don't have to redraw the entire screen, which takes a lot of time. You can merely change a color register to instantly modify the appearance of everything drawn in that color. A single memory change affects an entire screen, which makes possible some high-speed effects even in a relatively slow language like BASIC.

For example, if you draw a series of concentric circles, each circle deriving its color from a different color register, you could create a 3-D tunnel illusion by changing the color registers in sequence. You could fill all color registers with the same color, then change one color register at a time to create the illusion of growing circles. Objects can be made invisible by changing their color to the same color as the screen background, then made to appear instantly by giving them a contrasting color.

A Nybble Of Color

Color information is stored in the color registers by flipping certain bits on and off. Each 12-bit register assigns 4 bits for each primary color. (A group of 4 bits is called a *nybble*—half of a byte.) In the 320 × 200 and 320 × 400 modes, here is the format of the color registers:

11 10 9 8	7 6 5 4	3 2 1 0	*bit number*
blue	green	red	*primary color*

A handy formula for setting a color register in this mode is:

```
BLUE*256+GREEN*16+RED
```

where the luminance values of BLUE, GREEN, and RED range from 0–15.

The color registers for the hi-res 640 × 400 mode with an analog RGB monitor are a little trickier:

11 10 9	8 7 6	5 4	3	2 1 0	*bit number*
blue	green	rm rl	rh	b g r	*color*

As you can see, the color bits have been scattered all over the 12-bit range. Bits 9–11 define 3 bits of data for blue (range of 0–7); bits 6–8 define 3 bits of data for green (range 0–7). Bits 4 and 5 are the low and medium bits of data defining red, and bit 3 is the high bit of red data. (You would think the red bits would be arranged high-medium-low instead of high-low-medium, but the Amiga engineers must have had some reason for this strange order.) Bits 0–2 are the enable bits for the red, green, and blue electron guns.

The formula for setting a color register in this mode is also more complicated:

```
BLUE*512+GREEN*64+INT(RED/2)*
  16+(RED AND 4)*2+BEN*4+GEN
  *2+REN
```

This formula assumes RED, GREEN, and BLUE range from 0–7; REN, GEN, and BEN (the RGB enable bits) are either 0 or 1; that INT takes the integer result of its argument (as in BASIC); and that AND performs a bitwise AND.

A Binary Tower

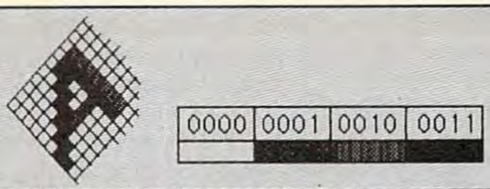
Each screen dot, or *pixel,* derives its color from one of these color registers. How are these dots laid out in memory? For a 32-color mode, each pixel is represented by a five-bit binary quantity ($2^5=32$). However, a five-bit quantity does not pack into a byte very well. Therefore, the Amiga maps its screen memory in a different way from most computers.

Traditionally, computers have laid out their screen memory serially, left to right, top to bottom. For instance, the Commodore 64's multicolor graphics mode fits four pixels into a byte, with each bit pair representing one pixel (00=color 0, 01=color 1, 10=color 2, 10=color 3). That's why the Commodore 64's 160 × 200 multicolor mode requires 8K of screen memory. With this memory scheme, to get more colors you would have to group more pixels together. But with five bits needed to store a single pixel on the Amiga, three bits would be wasted in every byte. If the Amiga used a serial scheme to store its display, it would take 64K to hold a 320 × 200 screen with 32 colors.

This problem was solved by grouping the bits a different way. Instead of using horizontally adjacent bits within the same byte to select a color register, the Amiga overlays bytes and reads the bits vertically. For example, all bits in bit position 7 from each of five overlaid bytes form a five-bit quantity. It's as if each pixel were a five-bit tower rising above the screen map. If you cross-section the vertical bytes making up the screen, you get five layers of bits called *bit planes.*

Each bit plane permits one bit of color definition. The simplest screen has only one bit plane, with one bit per pixel. This arrangement permits only on/off possibilities for each pixel. To get a broader range, you need to add another bit plane. That way, the bit on the primary bit plane and the bit in the corresponding position in the second bit plane permit two bits, or four possibilities of color definition. The accompanying figure shows how the Amiga uses bit planes for color selection, and the table gives a summary of the Amiga's screen modes.

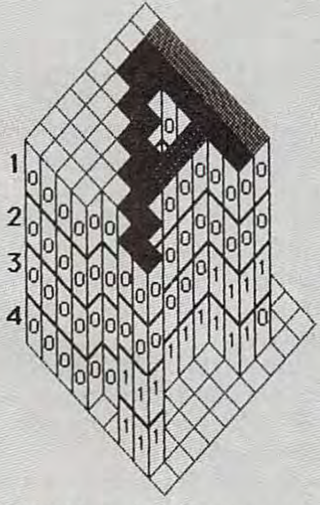
Incidentally, the Amiga has no true text modes like those found on earlier computers. Text is drawn as graphics objects, usually in 640 × 200 graphics, the default mode used by the Amiga's operating system, Intuition.

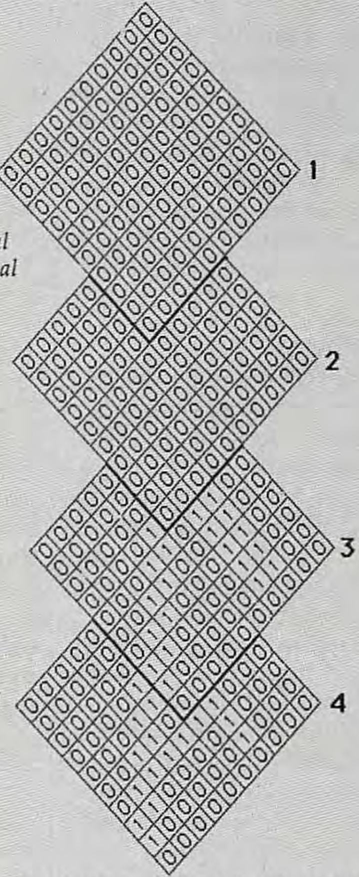


0000	0000	0000	0000	0000	0000	0000	0000	0000	000
0000	0000	0000	0000	0000	0011	0011	0010	0000	000
0000	0000	0000	0000	0011	0011	0011	0010	0000	000
0000	0000	0000	0011	0011	0000	0011	0010	0000	000
0000	0000	0011	0011	0000	0000	0011	0010	0000	00
0000	0011	0011	0001	0001	0001	0011	0010	0000	0
0011	0011	0000	0000	0000	0000	0011	0010	0000	
0011	0000	0000	0000	0000	0000	0011	0010	0000	
0000	0000	0000	0000	0000	0000	0000	0000	00	

This shows how screen memory is mapped serially. Every pixel is represented by a four-bit binary number. We can pack two pixels per eight-bit byte. This scheme is not efficient for packing pixel sizes of three, five, or seven bits.

Below, the pixels are represented as "binary towers." The planes on the right show the three-dimensional figure separated into two-dimensional bit-planes.





A more flexible and efficient way of representing multiple-bit objects (pixels) is to layer the bits "vertically." Each bit-plane is equivalent to one high-resolution screen. A pixel is represented by a single bit position, hence eight pixels per byte. To permit a pixel to represent more than just on or off, additional bit fields are layered. All bits in a corresponding bit position together define a four-bit value.

Amiga Screen Modes

Mode	Pixels	Bit Planes	Memory	Onscreen Colors
Normal res, noninterlaced	320 × 200	1 to 5	8K–40K	2 to 32
Normal res, interlaced	320 × 400	1 to 5	16K–80K	2 to 32
Hi res, noninterlaced	640 × 200	1 to 4	16K–64K	2 to 16
Hi res, interlaced	640 × 400	1 to 4	32K–128K	2 to 16
Hold and modify	320 × 200	5 or 6	48K	256 to 4,096
Hold and modify	320 × 400	5 or 6	96K	256 to 4,096

Bit Planes Save Memory

Although the bit planes are stacked together, don't think of them as multiple, transparent screens. The pile of bit planes creates only one screen. Bit planes are merely a way to make more efficient use of memory. If you don't need 32 colors, use fewer bit planes. Since each bit plane uses 8K in 320 × 200 resolution, memory usage can range from 8K (one bit plane, 2 colors) to 40K (five bit planes, 32 colors). Memory usage for the 320 × 400 mode can be up to 80K; for 640 × 200, 64K; and 640 × 400 uses as much as 128K. Again, these are only if all allowed bit planes are used. The mininum memory requirement for 640 × 200 is 16K, for a two-color mode.

The Amiga permits up to six bit planes, though only five are used at a time in 320 × 200 or 320 × 400 resolution. The hi-res 640 × 200 and 640 × 400 modes use only four bit planes, for a total of 16 colors. Although six bit planes are available, the video circuitry can't fetch more than five bits per pixel in 320 × 200 mode, or it would lag behind the speeding video beam. The video beam that refreshes the picture can't wait, so the video circuitry must keep up with the beam.

That's why 640 × 200 and 640 × 400 modes are limited to four bit planes, or 16 onscreen colors. The video circuitry just can't fetch memory fast enough to change twice as many pixels per line. In fact, adjacent pixels cannot always have different colors in the 640 modes. The adjacent values may point to different color registers, but it's not possible to fully change the hardware color output between even and odd pixels in the 640 mode. A dark green pixel to the right of a bright white one may appear gray ("dark white"). Only the luminance information can be effectively changed before the beam has left that pixel position.

Interlaced Modes

Most personal computers up to now have been limited to a vertical resolution of 200 scan lines (a *scan line* is the thin horizontal line painted on the CRT by the electron guns as they scan from left to right). However, the Amiga can make use of *interlacing* to double the number of scan lines. A TV or monitor displays more than 400 visible scan lines, but normally uses only every other scan line of a screen, filling in the odd lines with data from the even lines. In interlace mode, alternate screen refreshes are shifted up or down by one scan line, permitting full vertical resolution.

Interlacing on the Amiga works like this: In 1/60 second, the first 320 × 200 or 640 × 200 screen is scanned. Before the phosphors fade, a second 320 × 200 or 640 × 200 screen is scanned, shifted down one line to interweave it with the previous screen. The first screen displays even lines, and the second displays the odd lines. The result is doubled resolution—400 scan lines instead of 200.

Because the total picture takes twice as long to display, the phosphors in the even lines begin to fade as the odd lines are drawn. Therefore, some flickering and jittering in the 640 × 400 mode is visible. The only way to avoid this would be to redesign the monitor to refresh its screen at a faster rate than 1/60 second (a technique used by the Macintosh and Atari ST monochrome monitors).

The Amiga's video chip is smart enough to handle interlacing with ease. The bit planes are laid out in memory as if there were just one continuous 320 × 400 or 640 × 400 screen. You tell the video chip how far to skip ahead in memory to display the next line. By choosing an offset twice the normal line width, you can make the chip skip the odd lines of data for the first scan, then display the next screen from the odd lines, skipping the even lines. This greatly simplifies screen layout. The operating system actually takes care of these details, so you needn't even know how 320 × 400 or 640 × 400 are supported.

Dual Playfield Mode

In the normal-res modes, you can set up and overlay two independent graphics screens. A portion of one screen can be transparent to show the underlying screen. You can specify which screen has priority over the other. Each screen can be manipulated independently, even resized and moved over or under the other.

The overlay screens can use up to three bit planes each, since there are six available. However, you can use fewer bit planes if you want to save memory at the expense of color selection. Three bit planes permit only eight colors per screen, but each screen has its own color palette. And the palettes can contain any of the 4,096 hues, of course.

Dual playfields raise some exciting possibilities. In a game, one screen could show your cockpit window or starship control panel, with the windshield or viewport simply a transparent hole. The secondary screen could show your view of the sky or of the depths of space, visible through the transparent part of the primary screen.

Intuition uses this feature to let you slide down the top screen to see another behind it. For business applications, you could have two spreadsheets or documents running on the two screens simultaneously. Each screen can have its own windows, too.

Fine Scrolling

Scrolling is a technique that lets an *actual screen* pan across a much larger *virtual screen.* The actual screen is what you see on the monitor; the virtual screen includes the portions which won't fit on the monitor but can be scrolled into view. Scrolling lets you work with a very large document, spreadsheet, or page of graphics, and also makes for exciting computer games (such as *Defender* or *Eastern Front*).

Some computers are limited to *coarse scrolling*—the actual screen can be scrolled over the virtual screen only in character-sized increments. *Fine scrolling* is a more difficult technique that scrolls the actual screen pixel by pixel.

Fine scrolling is easy on the Amiga. The start of the screen map is found in two memory registers which are bit plane pointers. To scroll the screen up, just change the registers to point one line higher in memory. To scroll down, you subtract the line width from the bit plane pointers, displaying from the previously off-screen line of data.

For horizontal scrolling, a single register lets you shift the screen by up to 16 positions. You must fetch an extra data word per line to provide the pixels that should appear as the screen is scrolled. After you've scrolled 16 times, the program must perform a coarse scroll by repointing the bit plane pointer to the next word of memory. The whole display appears to move, but you're really just displaying a different section of memory.

4,096 Colors At Once

A special video mode lets you display more than just 32 colors at a time. *Hold and modify* mode can display 4,096 colors simultaneously in the normal-res modes.

It's a difficult mode to use, though. Each pixel is defined as a modification of the color of the previous pixel. You can hold this value, and modify a portion of it (hence *hold and modify*). Instead of bit plane data defining a color register, the bits from bit planes 5 and 6 determine which portion of the previous color output should be modified, and the bits from bit planes 0–4 are substituted in the selected portion of the color output.

You can define an entire screen of color, even the background color, just by modifying a single color register in this mode. You could start with bright white, then set the blue bits to zero to select yellow. From yellow you can decrease the red level to get green. You could then turn off the green bits to get black, which can in turn be modified to get bright blue. You can modify only the R, G, or B portion, or start over with data from a new color register.

Since color in the hold and modify mode is dependent on previous values, changing one pixel could change the colors of all following ones. It's a difficult mode to use for dynamic displays, so it is best suited for static pictures that need 4,096 colors.

There's much more to the Amiga's graphics than we can cover in this article. We haven't even begun to discuss blitter animation, sprites, the copper (video coprocessor), and options like external video mixing, video digitizing, and framegrabbing. It will probably be a while before programmers learn to take advantage of all these features. In the meantime, we'll have a lot to look forward to. ©

A Better Way To POKE On The Commodore 64

Matthew MacKenzie

This "program-writing program" for the Commodore 64 can speed up any BASIC routine that uses POKE to fill large areas of memory. Using clever programming techniques, it writes a new routine that employs fast PRINT statements in place of POKEs.

BASIC programs often require that you fill a certain memory area with data. The data may be a machine language routine, sprite shape definitions, a high-resolution graphics screen, or whatever. In most cases, the job is done with POKE statements: The program READs values from DATA statements and POKEs each value into the computer's memory. Unfortunately, POKE is one of BASIC's slower statements. In fact, it's so slow that some programs display countdown timers during memory-filling operations to tell you how much longer you'll have to wait.

PRINT, on the other hand, is very fast. Though it's intended for a different purpose than POKE, PRINT also puts new values into certain memory locations. After all, the screen is just another memory area in the computer: It consists of 1000 locations numbered from 1024 to 2023, with 1024 at the upper-left corner. When you PRINT the letter *A* in the upper-left corner of the screen, you're storing a new value in memory location 1024. Because of this similarity, it's possible to store values in memory with PRINT instead of POKE.

However, PRINT's memory-changing ability has certain limitations. Usually, you can print only in the 1000-character screen memory area. And after you've changed character 999, the screen begins to scroll. The top line of the screen disappears, and everything below that line moves up. Finally, POKE and PRINT use different codes to represent characters, requiring conversion from Commodore ASCII (for PRINT) to screen code values (for POKE).

POKEing With PRINT

"Print Poker" solves all those problems. If you already have a routine that uses POKEs to fill memory, Print Poker can write a new, faster routine that does the same job with PRINTs. You don't have to understand how the special PRINT technique works to use Print Poker; it automatically creates new BASIC program lines containing everything you need. Type`in the program listed below, and then save it. Because this program does some unusual things, be sure to read the following instructions *before* you try to run it.

As an example, say you've written a routine that puts eight sets of sprite shape data in memory locations 12288–12798. Your routine works fine, but POKEing the data into those 511 locations causes a noticeable delay. The Print Poker program can write a new BASIC routine that uses PRINT to do the same job more quickly. Before you run it, however, *you must run your own routine* to put the sprite data in memory. (Print Poker works only when the needed data is already in the proper memory area.)

Once that's done, enter NEW. Now you can load and run Print Poker. The program first asks you for beginning and ending addresses. In this case, you want the special PRINT statements to fill locations 12288–12798, so you enter 12288 for the beginning address and 12798 for the ending. This tells the program which memory area to look at when creating the special PRINT statements.

Next, you're asked to enter a starting line number. This is the first line number of the routine Print Poker is about to write for you. Use whatever line number is appropriate for your routine. Since Print Poker itself uses line numbers from 60000 to 60380, use numbers considerably below 60000 to prevent a conflict. The program also asks you for a line increment. Since you won't have any reason to edit the new routine after it's made, line increments of two or five are fine.

Finally, you're asked whether Print Poker should delete itself when it's finished. If you're creating only one new routine, press Y to answer yes. Press N for no if you're creating two or more sets of special PRINT statements in a single session. (Use RUN 60000 to run Print Poker a second time. You should always delete Print Poker the last time it's used.)

One Program Writes Another

After you answer all the prompts, Print Poker goes to work, using the *dynamic keyboard* technique to write each line of your new routine. The program itself is writing another program. First, it puts a line number and the needed characters on the screen. Then it stores the line in BASIC memory just as if you'd moved the cursor to that line and pressed RETURN. When large memory areas are involved, this may take a couple of minutes. After the program stops and the blinking cursor reappears, your new routine is complete, ready to be saved and incorporated into a program.

The special PRINT statements look quite strange, of course. Because POKE can take any value from 0 to 255, the equivalent PRINT statement is usually a collection of graphics characters, including some nonprinting character values like CHR$(2). Such lines are difficult if not impossible to edit. Thus, it's best to use Print Poker only when your data is in final form (after you've finished making changes in the sprite shapes and so on). If you must make a change, you'll find it much easier to change and rerun the POKE version of your routine, and then run Print Poker again.

It took some creative programming to overcome PRINT's limitations. The value stored in location 648 tells the 64 where in memory PRINT should put its data. By carefully manipulating the value in 648, you can divert PRINT's output to any memory location in the computer and defeat screen scrolling as well. When Print Poker has finished its work, it sets everything back to normal with POKE 648, 4. Note that this technique does not work correctly in the highest 256 bytes of memory used by BASIC (locations 40740–40959). Use the conventional POKE method to put data in those locations. In addition, if you intend to put data at the top of BASIC user space, with Print Poker or without it, remember to move down the top-of-BASIC and top-of-string storage pointers to protect your data.

Print Poker

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

```
60000 POKE53281,14:PRINT"{CLR}
      {DOWN}{BLK}{9 SPACES}P R
       I N T{3 SPACES}P O K E
      {SPACE}R"             :rem 6
60010 INPUT"{5 DOWN}STARTING A
      DDRESS";S            :rem 32
60020 INPUT"{2 DOWN}ENDING ADD
      RESS";E              :rem 41
60030 INPUT"{2 DOWN}STARTING L
      INE NUMBER";L   :rem 211
60040 INPUT"{2 DOWN}LINE INCRE
      MENT";I              :rem 65
60050 INPUT"{2 DOWN}SELF- DEST
      RUCT (Y/N)";D$  :rem 128
60060 V1=INT(S/256):V2=S-256*V
      1:IFLEFT$(D$,1)="Y"THEND
      =1                  :rem 209
60070 PRINT"{CLR}"L"
      {11 SPACES}P1=PE(648):PO
      648,"V1":A$=CH(34)+CH(34
      )+CH(20)"            :rem 31
60080 L=L+I:PRINT:PRINTL"
      {11 SPACES}?"CHR$(34)"
      {HOME}"CHR$(34)";";:IFV2
      =0THEN60100          :rem 11
60090 PRINT":FORX=1TO"V2":?"CH
      R$(34)"{RIGHT}"CHR$(34)"
      ;:NE":PRINT:RL=60110:GOT
      O60370              :rem 133
60100 PRINT:PRINT:RL=60110:GOT
      O60370              :rem 203
60110 PRINT"{CLR}"L"PRINT"CHR$
      (34);:PS=1024+POS(X):FOR
      X=1TO12:PRINT"{6 SPACES}
      ";:NEXT             :rem 192
60120 PRINT:PRINT:FORX=0TO3:A$
      (X)=MID$(CHR$(34)+"{A}$"
      +CHR$(34),X+1,1):NEXT
                          :rem 196
60130 S1=PEEK(S):IFS=ETHEN6024
      0                    :rem 77
60140 S1=PEEK(S):IFS1AND128AND
      R=0THENPOKEPS,146:PS=PS+
      1:R=1:GOTO60160  :rem 29
60150 IF(S1AND127)=S1ANDR=1THE
      NPOKEPS,210:PS=PS+1:R=0
                          :rem 243
60160 IFS1=34ORS1=162THENFORX=
      0TO3:POKEPS+X,ASC(A$(X))
      :NEXT:PS=PS+3:GOTO60180
                           :rem 87
60170 POKEPS,(S1AND127):rem 62
60180 IF(S+1)/256=INT((S+1)/25
      6)THENS2=INT(S/256)+1:GO
      TO60220             :rem 180
60190 IFPS>1090THEN60210
                           :rem 98
60200 PS=PS+1:S=S+1:GOTO60130
                          :rem 189
60210 POKEPS+1,34:POKEPS+2,59:
      RL=60110:S=S+1:GOTO60370
                          :rem 182
60220 POKEPS+1,34:POKEPS+2,59:
      L=L+I:PRINT:PRINT:PRINTL
      "PO648,"S2":?:?:?";
                          :rem 130
60230 PRINTCHR$(34)"{HOME}"CHR
      $(34)";":RL=60110:PRINT:
      S=S+1:GOTO 60370:rem 144
60240 POKEPS+1,34:POKEPS+2,59:
      PRINTL+I"{11 SPACES}";
                          :rem 198
60250 PRINT"PO648,P1:?"CHR$(34
      )"{HOME}"CHR$(34):RL=-1:
      IFDTHENRL=60280    :rem 8
60260 GOTO60370           :rem 55
60270 RL=60300:PRINT"{CLR}6000
      0":PRINT"60010":PRINT"60
      020":PRINT"60030"
                          :rem 112
60280 PRINT"60040":PRINT"60050
      ":PRINT"60060":PRINT"600
      70{DOWN}":GOTO60380
                          :rem 101
60290 RL=60320:PRINT"{CLR}6008
      0":PRINT"60090":PRINT"60
      100":PRINT"60110"
                          :rem 130
60300 PRINT"60120":PRINT"60130
      ":PRINT"60140":PRINT"601
      60{DOWN}":GOTO60380
                           :rem 91
60310 RL=60340:PRINT"{CLR}6017
      0":PRINT"60180":PRINT"60
      135":PRINT"60200"
                          :rem 133
60320 PRINT"60210":PRINT"60220
      ":PRINT"60230":PRINT"602
      40{DOWN}":GOTO60380
                           :rem 92
60330 RL=60360:PRINT"{CLR}6025
      0":PRINT"60260":PRINT"60
      270":PRINT"60280"
                          :rem 143
60340 PRINT"60290":PRINT"60300
      ":PRINT"60310":PRINT"603
      20{DOWN}":GOTO60380
                           :rem 99
60350 RL=-1:PRINT"{CLR}60330":
      PRINT"60340":PRINT"60350
      ":PRINT"60360"  :rem 236
60360 PRINT"60370":PRINT"60380
      ":PRINT"60390":PRINT"601
      05{DOWN}"            :rem 1
60370 PRINT"S="S"{LEFT}:E="E"
      {LEFT}:L="L+I"{LEFT}:I="
      I"{LEFT}:D="D"{LEFT}:R="
      R;                  :rem 251
60380 PRINT"{LEFT}:GOTO"RL"
      {DOWN}":POKE631,19:FORX=
      632TO640:POKEX,13:NEXT:P
      OKE198,10            :rem 94
```

©

Adding TIME$ To Atari

Kenneth S. Szajda

Here's a useful routine that adds a missing feature to Atari BASIC: TIME$. Now your programs can have realtime clocks and timed loops without PEEKs or POKEs. Requires only 8K RAM for cassette or 16K RAM for disk (with DOS 2.0, 2.5, and 3.0).

Atari BASIC is a very versatile and useful language. However, like all computer languages, it is not perfect. One useful feature that Atari BASIC lacks is the TIME$ function. For beginners, TIME$ provides a method to accurately time loops; for advanced programmers, TIME$ is a useful tool for avoiding the system timers, saving a lot of extraneous coding.

Timing from BASIC usually comes in two forms: TIME$ and TI (or some other appropriate numeric variable). Both supply the same information, but TIME$ represents the time as "HH:MM:SS" (or "HHMMSS") and TI represents the time in *jiffies* (1/60 second). In general, TIME$ is more useful because it is already formatted and ready for printing. Program 1 adds the TIME$ function to Atari BASIC, giving you easy access to the time without ever touching the system timers.

A Few Rules

Since we're patching this function into Atari BASIC, there are a few rules to follow for it to work properly. First, TIME$ must be DIMensioned like any other string variable, and it *must* be the first variable of any kind to appear in your program. To DIMension it, always use DIM TIME$(8). To activate the routine, use A=USR(1536) after it has been loaded into memory. (The routine automatically changes the dimension to eight and the length to eight no matter what you specify in the DIM statement, but it is best to use the correct value to avoid slight inaccuracies rippling throughout BASIC as a result.)

Second, TIME$ must always be the first variable BASIC sees when you LOAD or ENTER a program. If a new program is typed in with TIME$ as the first variable, no problems will result. However, adding TIME$ to a program already on tape or disk is a little tricky, but not very difficult. First, load your program into the computer (using LOAD or ENTER). When the READY prompt appears, add DIM TIME$ (8): A=USR(1536) as the very first executable (non-REM or DATA) line in your program. Then add any features using TIME$ to your program. (This step can be done at any time as long as the first line is already in the program.) Then store the program on tape or disk with the LIST command.

Using SAVE instead of LIST disables TIME$ when the program is loaded, because TIME$ will not be the first variable in BASIC's *variable name table* (which holds the name of every variable in your program). If you want to store the program with SAVE rather than LIST, just type NEW (to clear the variable name table) and reENTER the program after you have LISTed it to tape or disk. This rewrites the variable name table in the order that the variables appear in the program: TIME$ should be first.

Once TIME$ is first in the variable name table, subsequent SAVEs will not change its position, and everything will work smoothly.

The last rule, and probably the most important, is to avoid the CLR command. When CLR is executed while TIME$ is activated, the time will be lost and TIME$ will show "junk." Then you'll have to either type NEW and repeat the steps above or simply rerun the program (since RUN clears all variables and cleans things up).

An Autoboot Routine

The A=USR(1536) statement after the DIM statement actually sets TIME$ to "00:00:00" and activates the routine. The time will be expressed in military format, from 00:00:00 to 23:59:59. Then it returns to 00:00:00 and begins to count upward again.

The TIME$ program was written to be used as an autoboot file—AUTORUN.SYS with disk and a boot tape with cassette. For a disk-based system, run Program 1 and specify disk at the prompt. Remember, however, that this will erase an existing AUTORUN.SYS file on the disk, unless the AUTORUN.SYS file is locked, in which case the SAVE will fail.

For cassette-based systems, run Program 1 and specify cassette. The loader program will modify the TIME$ routine to make a boot tape.

In either case, loading the program on power up is simple. For disk, just boot up with the disk containing the AUTORUN.SYS file.

The routine loads into memory and can be activated with the DIM TIME$ (8):A=USR(1536) sequence. For cassette, boot the program by holding down the START button while turning on the computer. Press RETURN when the buzzer sounds, and the program automatically loads into memory. Again, activate the routine with the DIM TIME$(8): A=USR(1536) sequence.

You can set the time by using normal string manipulations. For example, to set TIME$ to 11 a.m., just type TIME$="11:00:00". If you press SYSTEM RESET, the time will *not* be stopped. Once started, the only way to stop the time is to use NEW. As long as TIME$ remains first in the variable name table, the counter will continue to update the time. No other special commands or techniques are required.

To see TIME$ at work, try running Program 2 after TIME$ is activated. The program asks for the current time and the time to sound an alarm (both in HH:MM:SS format). The computer displays the time until the alarm time arrives and then sounds five bell characters. Program 2 shows how TIME$ can be used like any other string variable.

How TIME$ Ticks

The TIME$ routine takes advantage of the Atari's timers. Atari computers contain many timers, but the ones most often used are the five two-byte timers at memory addresses 536–545 ($218 to $221 hex). Each timer is set up in the usual 6502 least significant byte (LSB), most significant byte (MSB) order.

Unlike most of the other system timers, however, these timers count down to zero instead of counting up from zero. During each vertical blank period, each timer is decremented. Since a vertical blank occurs every 1/60 second and the highest timer value possible is 65535, the timers can time a maximum of 18 minutes, 12¼ seconds each. When a timer counts down to zero, one of two things happens: Either a flag is set or a JSR (machine language Jump to SubRoutine) is executed. For timers 3, 4, and 5, a flag is set: CDTMF3 (address 554, $22A hex) for timer 3, CDTMF4 (556, $22C hex) for timer 4, and CDTMF5 (558, $22E hex) for timer 5. When timer 1 or 2 counts down to 0, a JSR is executed through CDTMA1 (550 and 551, $226 and $227 hex) for timer 1, and CDTMA2 (552 and 553, $228 and $229 hex) for timer 2.

TIME$ uses timer 2. Timer 3, 4, or 5 would require another routine to monitor the appropriate flag. Timers 2 through 5 are stopped during critical vertical blanks, which occur during input/output with peripherals. Timer 1 seems ideal, since it's the only timer that isn't affected by the critical vertical blanks. However, the serial input/output handler (SIO) uses it as a device timeout timer (to provide us with the ever-famous ERROR-138). So the TIME$ routine is forced to use timer 2.

Furthermore, to create an interrupt every second, the routine stores a value of 60 into the timer (remember, the timer is decremented every 1/60 second), and the interrupt service routine resets the timer to 60 at each interrupt. The only drawback is that input/output with peripherals temporarily stops the timer, and TIME$ will be slightly behind the true time (but can be easily changed as shown above).

Startup Routines

The initial call to the startup routine checks to see if TIME$ is the first variable and, if so, sets up the vector for the interrupt update routine, intercepts the SYSTEM RESET initialization vector (to keep the interrupt routine going after a SYSTEM RESET), determines the address of TIME$, sets its length and dimension to eight, initializes TIME$ to 00:00:00, and returns to BASIC. The interrupt service routine again checks to make sure TIME$ is the first variable (just in case a NEW was executed), determines the address of TIME$ again (since its address may change as a result of additions and corrections to the program in memory), sets the length and dimension to eight (just in case), and resets the timer value to 60 to cause another interrupt one second later.

Of course, during this entire process, TIME$ is updated to reflect the change in time. If a NEW has been executed since the last interrupt, the value of TIME$ is *not* updated and the timer is *not* reset to 60. In other words, if a NEW occurs, the routine effectively dismantles itself. The routine must be restarted with another DIM TIME$(8):A=USR (1536) sequence.

The program traps NEW but not CLR because it is difficult to tell when a CLR has been executed—CLR does not cause any actions within BASIC's tables that could not be caused by some other command or routine. Since NEW effectively blocks the variable name table, it is relatively simple to check for.

Please refer to "COMPUTE!'s Guide to Typing In Programs" before entering these listings.

Program 1: TIME$ BASIC Loader

```
BH 1Ø  DIM TIME$(8)
IL 2Ø  GRAPHICS Ø:POKE 752,1:
       ? "PROCESSING..."
IB 3Ø  CHECKSUM=Ø:RESTORE
EG 4Ø  FOR X=1536 TO 1758:REA
       D A:CHECKSUM=CHECKSUM+
       A:POKE X,A:NEXT X
DL 5Ø  IF CHECKSUM<>25Ø68 THE
       N ? :? "{2 BELL}***ERR
       OR IN DATA STATEMENTS*
       **":END
JI 6Ø  OPEN #2,4,Ø,"K":? :? "
       Cassette or Disk ?";:G
       ET #2,A
ON 7Ø  IF A<>67 AND A<>68 THE
       N CLOSE #2:GOTO 6Ø
KI 8Ø  IF A=67 THEN 16Ø
LL 9Ø  ? :? "Type Y to create
        AUTORUN.SYS":? "NOTE:
        Existing AUTORUN.SYS
       will be deleted"
CO 1ØØ  GET #2,A:IF A<>89 THE
        N END
GN 11Ø  TRAP 12Ø:XIO 33,#1,Ø,
        Ø,"D:AUTORUN.SYS"
BM 12Ø  TRAP 65535:OPEN #1,8,
        Ø,"D:AUTORUN.SYS":PUT
         #1,255:PUT #1,255:PU
        T #1,Ø
HF 13Ø  PUT #1,6:PUT #1,222:P
        UT #1,6
KO 14Ø  FOR X=1536 TO 1758:PU
        T #1,PEEK(X):NEXT X:C
        LOSE #1
CM 15Ø  ? :? "AUTORUN.SYS is
        now on disk":POKE 752
        ,Ø:NEW
EM 16Ø  ? :? "Position tape,
        press REC and PLAY":?
         "and press RETURN...
        ";:GET #2,A
KN 17Ø  POKE 1528,Ø:POKE 1529
        ,2:POKE 153Ø,248:POKE
         1531,5:POKE 1532,255
        :POKE 1533,5
PJ 18Ø  POKE 1534,24:POKE 153
        5,96
HH 19Ø  POKE 16Ø1,2:POKE 16Ø5
        ,3:POKE 1741,234:POKE
        1742,234:POKE 1743,2
        34
GI 200  POKE 764,0:OPEN #1,8,
        128,"C:":POKE 764,255
GA 210  POKE 850,11:POKE 852,
        248:POKE 853,5:POKE 8
        56,231:POKE 857,0
ME 220  A=USR(ADR("hhh█LVE"),
        16)
AH 230  CLOSE #1:? :? "Boot f
        ile is now on tape":P
        OKE 752,0:NEW
LP 240  DATA 104,169,0,133,17
        8,160
FP 250  DATA 4,177,130,217,21
        8,6
JF 260  DATA 208,56,136,16,24
        6,32
HA 270  DATA 156,6,165,178,20
        8,9
KD 280  DATA 160,8,169,48,145
        ,176
MO 290  DATA 136,16,251,169,5
        8,160
FJ 300  DATA 2,145,176,160,5,
        145
FM 310  DATA 176,169,71,141,4
        0,2
CO 320  DATA 169,6,141,41,2,1
        69
OJ 330  DATA 2,162,0,160,60,3
        2
NA 340  DATA 92,228,169,198,1
        33,12
GF 350  DATA 169,6,133,13,96,
        160
GB 360  DATA 4,177,130,217,21
        8,6
MM 370  DATA 208,66,136,16,24
        6,216
GM 380  DATA 32,156,6,169,58,
        160
GC 390  DATA 2,145,176,160,5,
        145
JJ 400  DATA 176,160,7,177,17
        6,24
LH 410  DATA 105,1,145,176,21
        7,210
GI 420  DATA 6,144,25,169,48,
        145
PO 430  DATA 176,136,177,176,
        24,105
FK 440  DATA 1,145,176,217,21
        0,6
KE 450  DATA 144,8,169,48,145
        ,176
JJ 460  DATA 136,136,16,219,1
        69.2
CH 470  DATA 162,0,160,60,32,
        92
DI 480  DATA 228,96,169,2,162
        ,0
DF 490  DATA 160,0,32,92,228,
        96
MA 500  DATA 160,0,169,129,14
        5,134
BB 510  DATA 160,4,169,8,145,
        134
BN 520  DATA 200,200,145,134,
        136,169
KO 530  DATA 0,145,134,200,20
        0,145
IN 540  DATA 134,160,2,177,13
        4,24
BN 550  DATA 101,140,133,176,
        200,177
PB 560  DATA 134,101,141,133,
        177,96
NC 570  DATA 169,255,133,178,
        32,17
MA 580  DATA 6,32,64,21,24,96
AP 590  DATA 51,52,48,54,58,4
        8
BF 600  DATA 54,58,84,73,77,6
        9
EM 610  DATA 164
```

Program 2: Sample TIME$ Program

```
AD 10  DIM TIME$(8),A$(8):A=U
       SR(1536)
OD 20  GRAPHICS 0
OO 30  ? "Enter current time
       ";:INPUT TIME$
JA 40  ? :? "Enter time to so
       und alarm ";:INPUT A$
OH 50  GRAPHICS 1
EL 60  POSITION 5,9:? #6;TIME
       $:POKE 708,INT(PEEK(53
       770)/16)*16+8
ON 70  IF TIME$=A$ THEN POSIT
       ION 5,9:? #6;TIME$:? "
       {5 BELL}{CLEAR}It's ti
       me...":POKE 708,40:END
AH 80  GOTO 60                ©
```

Apple Program Protector

Boris Troyanovsky

Do you have an Applesoft BASIC program you want to protect from prying eyes? With this technique, you can keep other people from listing your programs—while still giving them the freedom to make copies. For all Apple II-series computers with DOS 3.3.

"Apple Program Protector" is an easy to use utility that keeps other people from listing your BASIC and machine language programs. It works by moving the disk catalog on the protected disk to another track, and by preventing users from breaking out of the program by pressing CTRL-C or RESET. The only requirement is that your program must run itself when the user boots the disk.

It's easy enough to ensure that the program runs automatically when the disk is booted. DOS 3.3 always loads and runs a BASIC program whenever a disk is booted if the program is named HELLO. Only a single BASIC program can be started in this way. If you wish to have more than one BASIC program on the protected disk, you must make HELLO a menu program that allows you to select the desired program from the disk. If the program you wish to protect is written in machine language, you can use HELLO to start it. For example, if you want to protect a program named MLGAME, your HELLO program might be simply:

```
10 PRINT CHR$(4);"BRUN MLGAME"
```

The next step is defend against CTRL-RESET and CTRL-C. In each BASIC program on the disk you wish to protect, add these two lines:

```
0 POKE 1011,0:ONERR GOTO 63999
63999 RESUME
```

The POKE in line 0 defends against the CTRL-RESET key (or just RESET on some Apples). If the CTRL-RESET key is pressed, any Applesoft BASIC program in memory is erased and the computer reboots.

Since the CTRL-C interrupt code (which is used to stop program execution) is considered an error by Applesoft BASIC, the ONERR GOTO statement in line 0 transfers program control to line 63999 when CTRL-C is encountered. If your program uses ONERR to test for other conditions, include the statement IF PEEK(222)=255 THEN 63999 in your error testing. This ensures that CTRL-C is still trapped.

Machine language programs can be protected in this way, too. Just add these lines to your source code:

```
LDA #$00
STA $03F3
```

Like the BASIC lines above, these instructions erase the program from memory and reboot the computer if CTRL-RESET is pressed.

Preparing A Protected Disk

Begin by preparing a disk containing the program or programs you wish to protect. If the disk is to contain only one program, load it into memory, insert a new disk, type INIT HELLO, and hit RETURN. If you wish to have several programs on the disk, the HELLO program must be a menu program and the others can simply be saved on disk in the usual manner. Remember to add to each program the lines mentioned above to protect against CTRL-RESET and CTRL-C. *Make sure all programs are fully tested and debugged before you run Program Protector.* To be safe, you should always keep backup copies of the programs on an unprotected disk.

Next, type in Program 1 below and save a copy on a separate disk. Use the filename PROTECTOR. With the built-in machine language monitor, enter the data from Program 2. (If you are unsure about using the monitor, consult your user's manual.) Save the machine language onto the same disk with Program 1 using the command BSAVE IOB, A$0300, L$40.

The Program Protector disk is now ready to use. To protect one of your disks against intrusion by outsiders, follow these steps:

1. Insert the Program Protector disk into the drive (the drive should be addressed as slot 6, drive 1).

2. Type BLOAD IOB.

3. Type RUN PROTECTOR.

4. Program Protector is now loaded into memory. You should see the prompt DESTINATION TRACK: on the screen. Remove the disk with Program Protector and insert the disk that you'd like to protect into drive 1. Type the number of the track to which you want to move the catalog and press RETURN. The number must be greater than or equal to 3 (DOS occupies tracks 0–2), and less than or equal to 34, since there are only 35 tracks (numbered 0–34) on the disk. Also, the number can't be 17, because that's where the catalog is already.

5. The disk drive whirs a bit, then the Applesoft] prompt reappears. The disk is now protected. If you type CATALOG, you'll be shown an empty directory. If you try to load a program from the protected disk or save an additional program to it, you'll get nothing but a DISK FULL error message. However, if you now boot the protected disk, the HELLO program loads and runs normally, except that CTRL-C no longer stops the program and CTRL-RESET only reboots the system.

There is a way to regain access to the programs on the protected disk. Boot a normal disk, then enter POKE 44033,*n* (substitute for *n* the number of the track to which the catalog was moved). You can now display the catalog and load and save programs. You can also use this technique if the program you're protecting needs to access another program on a different disk. To let the program know where the catalog of the new disk is, POKE 44033 with the catalog track of the disk you'd like to access.

Don't try to relocate the catalog on a disk more than once. The results are unpredictable.

Program 1: Apple Program Protector

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

```
7B 10 HIMEM: 8191
E8 2Ø HOME : HTAB 11: INVERSE :
      PRINT "PROGRAM PROTECTOR":
       NORMAL : VTAB 1Ø: INPUT "
      DESTINATION TRACK: ";DT
ED 3Ø IF DT < 3 OR DT > 34 OR DT
       = 17 THEN PRINT "ILLEGAL
      TRACK NUMBER": FOR A = Ø T
      O 6ØØ: NEXT A: RUN
9E 4Ø REM **MOVE TRACKS**
31 5Ø SS = Ø:SE = 15:TR = 17:BU
      = 8192:OP = 1: GOSUB 38Ø:
      REM READ CATALOG TRACK
51 6Ø GOSUB 18Ø
AA 7Ø D1 = PEEK (DT * 4 + 56 + 8
      192):D2 = PEEK (DT * 4 + 5
      7 + 8192):O1 = PEEK (17 *
      4 + 56 + 8192):O2 = PEEK (
      17 * 4 + 57 + 8192)
DB 8Ø POKE 17 * 4 + 56 + 8192,D1
      : POKE 17 * 4 + 57 + 8192,
      D2: POKE DT * 4 + 56 + 819
      2,O1: POKE DT * 4 + 57 + 8
      192,O2
EA 9Ø SS = Ø:SE = 15:TR = DT:BU
       = 16384:OP = 1: GOSUB 38Ø:
       REM READ NEW TRACK
64 1ØØ REM **ALTPOINT ROUTINE**
C9 11Ø TV = 8192
56 12Ø FOR TX = Ø TO 15: POKE TV
        + 1,DT:TV = TV + 256: NE
       XT TX
CD 13Ø SS = Ø:SE = 15:TR = DT:BU
        = 8192:OP = 2: GOSUB 38Ø
       : REM WRITE CATALOG TRACK
5D 14Ø SS = Ø:SE = 15:TR = 17:BU
        = 16384:OP = 2: GOSUB 38
       Ø: REM WRITE NEW TRACK
F8 15Ø SS = 11:SE = 11:TR = 1:BU
        = 8192:OP = 1: GOSUB 38Ø
       : POKE 8193,DT:SS = 11:SE
        = 11:TR = 1:BU = 8192:OP
        = 2: GOSUB 38Ø: REM CHAN
       GE DOS
4B 16Ø GOSUB 23Ø
97 17Ø END
Ø4 18Ø REM **ALTER CATALOG'S T/S
        POINTERS**
BE 19Ø BF = 8192 + 256
DD 2ØØ FOR PR = 11 TO 221 STEP 3
       5: IF PEEK (BF + PR) = DT
        THEN POKE BF + PR,17
38 21Ø NEXT PR:BF = BF + 256: IF
        BF < > 12288 THEN GOTO 2
       ØØ
16 22Ø RETURN
99 23Ø REM **CHANGE T/S LISTS**
B5 24Ø BF = 8192 + 256
CC 25Ø FOR PR = 11 TO 221 STEP 3
       5: IF PEEK (BF + PR) < >
       Ø THEN GOSUB 29Ø
41 26Ø NEXT PR
Ø6 27Ø BF = BF + 256: IF BF < >
       12288 THEN GOTO 25Ø
22 28Ø RETURN
A1 29Ø REM **DIRTY WORK**
ED 3ØØ LT = PEEK (BF + PR):LS =
       PEEK (BF + PR + 1)
A8 31Ø SS = LS:SE = LS:TR = LT:B
       U = 16384:OP = 1: GOSUB 3
       8Ø:BU = BU - 256
9C 32Ø FOR CT = 12 TO 254 STEP 2
       : IF PEEK (BU + CT) = DT
       THEN POKE BU + CT,17
B7 33Ø NEXT CT
B9 34Ø IF PEEK (BU + 1) = DT THE
       N POKE BF + 1,17
3Ø 35Ø OP = 2:SS = LS:SE = LS:TR
        = LT:BU = 16384: GOSUB 3
       8Ø:BU = BU - 256
53 36Ø IF PEEK (BU + 1) < > Ø TH
       EN LT = PEEK (BU + 1):LS
       = PEEK (BU + 2): GOTO 31Ø
21 37Ø RETURN
7E 38Ø REM **DISK ACCESS**
91 39Ø FOR SA = SS TO SE
FØ 4ØØ POKE 788,TR: POKE 789,SA:
        POKE 796,OP
CØ 41Ø HB = INT (BU / 256):LB =
       BU - (HB * 256)
67 42Ø POKE 792,LB: POKE 793,HB
2Ø 43Ø CALL 768:BU = BU + 256: N
       EXT SA
1C 44Ø RETURN
```

Program 2: IOB Routine

Enter this listing with the machine language monitor.

```
Ø3ØØ- A9 Ø3 AØ 1Ø 2Ø D9 Ø3 6Ø
Ø3Ø8- ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
Ø31Ø- Ø1 6Ø Ø1 ØØ 11 ØF 3Ø Ø3
Ø318- ØØ 2Ø ØØ ØØ Ø1 ØØ FE 6Ø
Ø32Ø- Ø1 ØØ ØØ ØØ ØØ ØØ ØØ ØØ
Ø328- ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
Ø33Ø- ØØ Ø1 EF D8 ØØ ØØ ØØ ØØ
Ø338- ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
```

©

The Beginners Page

Tom R Halfhill, Editor

Variable Accuracy

There are a few more points about integer variables that we didn't cover last month—including some important exceptions to general rules.

Note that in some versions of BASIC, such as Commodore BASIC, you aren't allowed to use integer variables as counters in FOR-NEXT loops. In other words, a statement such as FOR X%=1 to 10:NEXT X% would cause an error. However, integer variables *can* be used as counters in IBM BASIC. Just remember that because integer variables are restricted to a minimum value of −32,768 and a maximum of 32,767, you'll have to make sure your loops don't exceed those limits.

In some BASICs, there's also one exception to the rule about denoting all integer variables with the % symbol. In IBM BASIC, for instance, you can insert a DEFINT (*define integer*) statement near the beginning of the program to define a whole group of variable names as integer variables by default. The statement DEFINT A–M declares that all variable names beginning with the letters A through M are automatically integer variables. Since integer variables execute faster and consume less memory on the IBM than regular variables, a DEFINT statement can improve a program's performance. (Remember from last month that integer variables don't save memory and actually run slower on Commodore and Apple computers.)

Incidentally, another way to speed up your programs is to replace frequently used constants with variables. In most Microsoft BASICs, variables execute faster than constants (the reverse is true of Atari BASIC). By predefining the most commonly used numbers (usually 0–10) as variables, statements such as Y=Y+1 can be changed to Y=Y+C1. And if you're programming on an IBM, you can define them as integer variables and pick up even more speed. Try this technique in your next program and see if it adds a little zip.

Improving Precision

Another type of variable is provided in some BASICs to improve mathematical accuracy. Called *double-precision* variables, they can help avoid the small rounding errors that sometimes accumulate and cause strange results. (Rounding errors are slight discrepancies that can crop up when the computer converts our everyday decimal numbers to its internal system of binary numbers, performs some arithmetic, and then converts the answer back into decimal again.)

Double-precision variables are available in IBM BASIC and some versions of TRS-80 BASIC, but not in Commodore BASIC, Applesoft, Atari BASIC, or TI BASIC. Even if your BASIC doesn't have double-precision variables, however, they're worth learning about. As personal computers grow more powerful, you're sure to encounter such features sooner or later.

Here's an example which demonstrates a common type of rounding error—in this case, on the IBM PC/PCjr. This program initializes the variable Y to 100, then subtracts the value .05 ten times using a FOR-NEXT loop. The final value of Y is printed after the loop is finished.

```
10 Y=100
20 FOR X=1 TO 10
30 Y=Y-.05
40 NEXT X
50 PRINT Y
```

The answer, of course, should be 99.5. Instead, here's the program's answer:

```
99.49997
```

If you swap the statements in lines 40 and 50 so the program prints the current value of Y after each pass through the loop, you'll see that the rounding error starts with the second calculation and keeps increasing until the final result is off by .00003. That's not a huge discrepancy—but still, if this were some kind of banking program that was subtracting a nickel-a-day service charge from customer accounts over a period of time, someone might be cheated out of a penny now and then. (It's happened to me, by the way.)

Fortunately, you can program the computer to deliver a better answer. To convert Y from a regular (single-precision) variable into a double-precision variable, add the # symbol to every occurrence of Y in lines 10, 30, and 50. The modified program yields this result:

```
99.49999999254942
```

Well, even computers aren't perfect. This time the rounding error starts with the first calculation, although the errors are smaller and the final answer is off by only .0000001192093. For even greater accuracy, we can define the value of .05 as a *double-precision constant* by adding the # symbol to .05 in line 30. Here's the result:

```
99.49999999999999
```

This is even more accurate; now the computer is off by only .00000000000001. Furthermore, by switching lines 40 and 50 to see the results of each calculation through the loop, you'll notice that all the intermediate answers are exactly correct. That still leaves us with an infinitesimal error in the final answer, however. If you're a stickler for absolute accuracy, there are ways to get around these runaway fractions—but that's a topic for a future column. ©

Computers and Society

David D. Thornburg, Associate Editor

Personal Computers And Personal Freedom

Last July I spent a day in Colonial Williamsburg, Virginia, catching a glimpse of life as it was in the mid-1700s. The publishing exhibit particularly caught my attention. One could see paper being made by hand—a process that required skill and strength on the part of the craftsman, and which took a long time. In the print shop one could see the pages of a book being printed from hand-set type—another expensive and time-consuming process. The bindery exhibit showed how the printed pages were folded into signatures and stitched together by hand before being bound in leather.

It was clear from this exhibit that access to books was limited to the wealthy. The cost of spreading the printed word was quite high, and yet this period gave us a rich collection of people who had much of importance to say—Patrick Henry and Thomas Paine, to name just two.

As I thought about our advances in communication technology since that period—typewriters, copiers, computers, and so on—it became clear that the reduced cost of communication was one of the main reasons that literacy could spread to the public at large. The printed word has spread like wildfire, carrying messages into homes that would have been bookless in the 1700s.

The freedom to communicate is one of our most treasured freedoms. There are nations on this planet where individual ownership of copiers and computers is forbidden. It is easy to see why—it's important for a totalitarian government to control the flow and distribution of information. Otherwise, individuals could create, publish, and distribute their own ideas without the censorship of the state.

Computer Publishing

Prior to the widespread sale of personal computers, we had restrictions of our own that limited the widespread dissemination of ideas.

Before an opinion can be expressed in printed form, the author must either convince a publisher that it is worth expressing, or must elect to publish it alone. Even if a publisher accepts a work, it will reach an audience only if stores decide to stock it.

Suppose you've written something you think others might like to read—a collection of poetry, for example, or a political treatise. You may find that traditional publishers are not interested in your material because your market is too specialized. Or, you may find that they are interested, but that if you wait the four to six months (or longer) that it takes for your words to be printed, your material will have lost its currency and impact.

In this case, you may elect to publish the material yourself.

Prior to the personal computer, you might be restricted to running copies of your material at the local print shop. Depending on the size of your document, you may find that it costs several dollars per copy to have it printed.

But, in an era where personal computers are increasingly commonplace, there's another way of publishing your ideas—especially if what you have to say is of particular value to others who own computers. You can publish your ideas on a disk! Disks are inexpensive, reusable, and can be duplicated as needed. Publishing your material on disk lets you fix mistakes quickly without having to wait for a new printing. Your material might consist of text files that can be read with a word processor, or you can write your own program that lets people read or print your files as they choose.

However, along with the freedom to publish your own materials in the privacy of your home using nothing more than your personal computer there comes a responsibility. It is correctly said that the pen is mightier than the sword. As your own publisher, you can say anything you wish, but you must always keep in mind that the printed (or displayed) word is very powerful. Think your ideas through carefully before publishing them.

Talking Books

I recently used this publication technique for my book *In Search of the One-Minute Megatrends—Surviving the Bad Times in Silicon Valley* (Innovision Press, $12.95). While the information in this book is of potential interest to a broader audience, I initially made it available on a Macintosh disk, since that's the computer with which I do most of my writing. As I was creating the book, it occurred to me that this method of publishing had much greater flexibility than the printed page. For example, readers could change the typeface and size if they desired.

I also included a set of files on the disk that lets the book read itself aloud to the user with the Smooth-Talker speech synthesizer from First Byte. This not only provides another alternative for reading the book, but also makes the material available to those with impaired vision.

While this book is not available in stores, I have been able to sell it quite well through direct mail by placing inexpensive advertisements in regional computer-interest newspapers. The success of this venture convinces me that anyone with a message of interest to computer owners can be an author and a publisher as well.

If only the founding fathers could see us now! ©

Telecomputing Today

Arlan R. Levitan

Faster Than A Speeding Byte

Last month I mentioned Fastlink, a new 10,000 bits per second modem from Digital Communications Associates. Not only is it five times faster than the latest "high-speed" 2400 bps modems, it even works over regular telephone lines. Until now, anything close to 10,000 bps required you to lease special data-grade lines from the phone company. But not the DCA Fastlink. Although the Fastlink's $2,000 price tag is a little rich for most casual users' blood, there has been a fair amount of incredulous reaction like "how the heck can they do 10,000 bits per second?" from readers of this column (and even from my editor).

While the subject is a bit technical, I've distilled an explanation that will either satisfy your curiosity or teach you never to ask me about this sort of thing again. To get started, let's review our old friend, the ordinary 300 bps modem.

Modems exchange information over phone lines by transmitting and receiving audio tones. A 300 bps modem transmits over two channels, one for each direction. Each second of time is divided into 300 slices, and each slice is called a *baud*. A 300 bps modem packs one bit into each baud (1200 and 2400 bps modems both operate at 600 baud and pack 2 and 4 bits into each baud, respectively). One channel transmits signals in the audio range of 1070–1270 hertz, and the other at 2025–2225 hertz. That means each channel has a fairly wide bandwidth (200 hertz), and they're separated by a guard band of no signal (755 hertz wide) that makes it easy for the modem circuitry to differentiate between the two channels.

DCA's Fastlink uses a very low 7.3 baud rate, so it can drastically narrow the channel bandwiths and guard bands. The Fastlink also uses the entire 0–4000 hertz audio spectrum of normal phone lines. When two Fastlink modems link up, they attempt to establish a maximum of 512 separate channels, each 7.8 hertz apart. They analyze each channel to determine which ones are noise-free enough to handle transmission techniques that pack 4 or 6 bits into each time slice, or baud.

Then the Fastlink transmits data by using a hybrid parallel/-serial system (300 bps modems send data in a serial stream of bits—one bit after another with one bit per baud). The bits carried by all channels in use during one baud are considered a single packet of information. Outgoing data bits are assigned to channels as they're prepared for sending (with either 4 or 6 bits per channel), beginning with the channels at the lowest frequencies. Once the packet is assembled, it's sent across the active channels. So the data is sent in parallel within the packets, and the packets themselves are sent serially.

Blistering Speed

Using the Fastlink method, the maximum theoretical throughput is 512 channels × 6 bits per baud × 7.3 baud per second, or more than 20,000 bps. Given the quality of most voice-grade lines, that limit is very theoretical. Most channels operate at only 4 bits per baud, and throughput is further limited by the overhead of error detection and correction, which is automatically handled by Fastlink. All these factors reduce the Fastlink's actual throughput to a blistering 10,000 bps on local phone lines. A Fastlink modem operating on lines provided by the most popular long-distance carrier should work at about 8,000 bps. On the lines provided by other common carriers, the Fastlink averages about 7,000 bps.

The Fastlink monitors the quality of the phone line during the linkup, shutting down channels that become marginally acceptable or opening up channels if quality improves. DCA refers to the process as DAMQAM, or *Dynamically Adaptive Multicarrier Quadrature Amplitude Modulation* (say it five times fast). To handle all this data manipulation and line monitoring, the Fastlink is actually a full-fledged, highly specialized computer with a megabit of memory and two central processing units—a Motorola 68008 working in tandem with a Texas Instruments 320.

There are some fine points to keep in mind while daydreaming about cruising along at 10,000 bps. The Fastlink dynamically assigns channels to incoming or outgoing data based on the volume going back and forth. If there is an equal amount of data moving in both directions, the Fastlink channels would be equally divided between incoming and outgoing data, resulting in an effective maximum speed of only 5,000 bps for each data stream.

In practice, the data flow is usually quite lopsided, with ratios of 99 to 1 more common than 50:50. So the bulk of data flow on a Fastlink is assigned the lion's share of channels, resulting in throughput that is very close to the 10,000 bps ideal.

Fastlink modems currently come in two flavors. An internal version for the IBM PC and compatibles goes for $1,995 and includes a special version of Microstuf's *Crosstalk* program adapted for the Fastlink. An outboard RS-232 Fastlink is priced at $2,395. Both modems are also capable of communicating at plain old 300 or 1200 bps with non-Fastlink modems. If you're still curious, you can get even more information by contacting DCA at 1000 Alderman Drive, Alpharetta, GA 30201. ©

The World Inside the Computer

Fred D'Ignazio, Associate Editor

The Case Of The Phantom Programmers

Earlier this year I wrote about one of my high school assistants—Howard Boggess, my "Computer Handyman." This time I'd like to introduce you to another one of my assistants—Huntef Baker, my "Phantom Programmer."

Like Howard, Hunter came to me from David James's computer science class at Patrick Henry High School, here in Roanoke, Virginia. When Hunter arrived at my house on the first day, I took him and his mother to the dark, hot attic where Howard had rescued several broken-down computers (see "The World Inside The Computer," COMPUTE!, January 1985). "This is your first task," I said, with a sweep of my arm. "If you can clean this attic, then I know you can do anything."

Hunter is a quiet, mild-mannered person. He simply nodded when I told him to clean the attic. But this was no ordinary attic. And I worried about him every day when he trudged up the attic stairs.

I shouldn't have worried. Sending Hunter into the attic was like sending Cinderella into her stepmother's kitchen, or Hercules into the Augean stables. In a month, Hunter had the attic better organized than the rest of the house. He had everything filed away in labeled filing cabinets and had built a computer database so we could instantly know where to look for our long underwear, computer manuals, extension cords, extra paper, Christmas tree lights, winter gloves and mittens, and RS-232 cables.

Then Hunter moved downstairs. When he first confronted the downstairs office, computer software was piled to the ceiling and computer cables and circuit cards spilled out the door into the middle of the living room. But, for Hunter, after facing the horrors of the attic, this awful mess was no more than a tasty dessert. In only a couple of weeks everything was cataloged, labeled, and filed. The mess had vanished, and Hunter was hard at work at one of the computers.

Computer Trivia

One day I walked into the room, looked over Hunter's shoulder at a BASIC program on the display screen, and asked him what he was doing. He explained that he and his friend Amy Powell were doing a computer project for National History Day. They planned to create a history trivia game on the IBM computer, and Hunter asked if he and Amy could start coming over to our house after hours to work on the program. "Of course," I said, since I was sure he was only talking about a couple of evenings and maybe a weekend or two.

Ha! After watching Hunter clean the attic and the office, I should have been wiser. Hunter doesn't do anything halfway, and this project was no exception. For the next month, he and Amy came over almost every night after dinner, and most Saturdays and Sundays. They rarely left until the wee hours of the morning.

One night I was awakened around 2 a.m. by strange clicking noises. Alarmed, I tiptoed to the bedroom closet and grabbed the machete my parents had bought me in the Dominican Republic. (The machete was duller than a letter opener, and it had a parrot inscribed on its side, but it looks deadly, especially when I wave it threateningly above my head.)

I made my way cautiously down the stairs. I noticed a light was switched on in the downstairs office. I guessed that a thief must be inside stealing one of my beloved computers!

Leaping down the remaining stairs, I burst into the office, screaming and waving the machete.

It took a moment for my eyes to adjust to the bright lights in the room. When they did, I noticed Hunter and Amy seated at two IBM computers, working on their History Day program. "We're sorry we're here so late," said Hunter politely.

"Tomorrow's the competition," explained Amy. ©

INSIGHT: Atari

Bill Wilkinson

Deactivating BASIC

My coworkers and I have received many requests from owners of the Atari 600XL, 800XL, and 130XE for a simple way to turn off the BASIC built into those computers. Of course, the method recommended by Atari is to hold down the OPTION button when you boot the system. If you forget to do this when booting a program that doesn't require BASIC, the ROM-based BASIC occupies address space that costs you more than 8,000 bytes of RAM. There are other reasons for turning off BASIC as well. For instance, you might like to turn it off temporarily to gain extra memory while duplicating a few files or disks. These jobs take less time and fewer disk swaps if the computer can use the 8K of memory vacated by disabling BASIC. And avoiding a reboot or two can save time, too.

Our solution is a pair of short machine language programs that let you turn BASIC on and off from DOS. (Note that they can't turn off a BASIC *cartridge*—or any other cartridge, for that matter—so they serve no purpose on the Atari 400, 800, and 1200XL computers.) Atari manuals suggest that turning off the built-in BASIC is as simple as changing one bit in the XL/XE memory control location (which used to control joystick ports 3 and 4 in the 400 and 800). That may be true if you're writing a machine language program that takes over complete control of the computer, but in many cases it doesn't work.

First, whenever you press the RESET button, the operating system restores the built-in BASIC to the state in which you booted it. Second, if you're using ordinary graphics mode screens (without a custom display list, etc.), the screen handler doesn't use the memory freed by removing BASIC. It thinks you're still using a 40K machine. Going the other way—turning on BASIC after booting without it—can be even messier. If you suddenly enable BASIC without doing something about the screen, you'll find yourself staring at garbage as BASIC blithely wipes out the display list, screen memory, and perhaps more. Fortunately, all of these problems can be solved by following these few steps:

1. Turn the built-in BASIC off or on.
2. Tell the operating system you did so.
3. Change the master top-of-RAM pointer.
4. Close channel 0, the screen editor.
5. Reopen channel 0.

We can tell the operating system we changed the state of BASIC via the flag in memory location 1016 ($3F8). The master top-of-RAM pointer is RAMTOP at location 106 ($6A). Channel 0 is closed and reopened to force the screen driver to use the highest available memory. Don't worry if that sounds a bit arcane. The program listed here automatically creates two machine language programs that do all the work for you. Be sure to save a copy before you run it.

```
GD 100 DIM NAME$(20)
NM 110 LINE=800:GOSUB 210
NO 120 LINE=900:GOSUB 210
GL 130 END
OE 210 CHECK=0:RESTORE LINE
GH 220 FOR CNT=1 TO 57:READ
        BYTE
EL 230 CHECK=CHECK+BYTE:NEXT
        CNT
EI 240 READ TEST:IF CHECK<>T
       EST THEN STOP
OJ 250 READ NAME$:OPEN #1,8,
       0,NAME$
OE 260 RESTORE LINE
GM 270 FOR CNT=1 TO 57:READ
       BYTE
KF 280 PUT #1,BYTE:NEXT CNT
GF 290 CLOSE #1
HD 300 RETURN
IN 810 DATA 255,255,0,4,44,4
       ,173,1,211,9,2,141,1
DL 830 DATA 211,169,1,141,24
       8,3,169,12,32,24,4
OG 840 DATA 169,192,133,106,
       169,3,141,66,3,169,42
ID 860 DATA 141,68,3,169,4,1
       41,69,3,162,0,76,86
OK 870 DATA 228,69,58,0,226,
       2,227,2,0,4
HM 880 DATA 5045,D:BASICOFF.
       COM
MF 910 DATA 255,255,0,4,44,4
       ,173,1,211,41,253,141
JI 930 DATA 1,211,169,0,141,
       248,3,169,12,32,24,4
OC 940 DATA 169,160,133,106,
       169,3,141,66,3,169,42
ID 950 DATA 141,68,3,169,4,1
       41,69,3,162,0,76,86
OL 970 DATA 228,69,58,0,226,
       2,227,2,0,4
EG 980 DATA 5295,D:BASICON.C
       OM
```

The program writes two binary files to disk on drive 1, naming them BASICON.COM and BASICOFF.COM. The first turns BASIC on and the second turns it off. To use either of them from DOS, simply choose the L (load binary file) option and enter the filename when prompted. (OS/A+ and DOS XL users need only type BASICON or BASICOFF in response to the D1: prompt.)

The next time you need to duplicate a disk or large file, load BASICOFF.COM first, copy the disk or file, then load BASICON.COM to reactivate BASIC. You'll save time, especially on a single-drive system. If you're writing machine language programs, call BASICOFF as a subroutine when you start your program. ©

Donald B. Trivette

A Promise Of Things To Come

When I saw the advertisement for the Key Tronic KB 5152V, I knew it was a product designed with me in mind. Who hasn't dreamed of using a typewriter that will type every word you speak—or better yet, a computer that can understand spoken commands? The KB 5152V speech-recognition keyboard for the IBM PC, manufactured by Key Tronic of Spokane, Washington, seemed to hold just that promise. While waiting for a demonstration unit to arrive, I had visions of a new, laid-back life. Since my hands would no longer be needed for typing, I could dictate prose while holding a beverage and munching pretzels. Nor would I be restricted to a sitting position. This very column—in the interest of evaluating the product, of course—would be written from my bed.

The new keyboard arrived and plugged right into the socket vacated by the original IBM keyboard. It's an enhanced keyboard with separate numeric keypad and LED indicator lights on the Caps Lock and Num Lock keys. And, of course, there's one other enhancement: A telephone operator's headset that plugs into the back of the keyboard. Without bothering to read the manual, I spoke: "Now type this." Nothing happened, nor had I really thought it would.

The first step to using the keyboard is to teach it a vocabulary. Key Tronic supplies a menu-driven BASIC program that creates a standard ASCII text file—the vocabulary. For example, the vocabulary entry for the color BLUE might appear as BLUE;BLUE. The word to the left of the semicolon is the prompt—the word you speak; the word to the right of the semicolon is what is sent to the PC, just as though it had been typed on the keyboard. It's called the response. Thus, saying "blue" types *BLUE*. But that doesn't have to be true. You can teach the keyboard that blue is red and red is white, and it won't be the wiser.

The response characters can be more than one word and may contain characters in braces to represent keys, such as Enter, Backspace, and the special function keys. You can also define responses by keyboard scan codes or ASCII codes, so every key and key combination is accessible.

Once the written vocabulary is defined, the keyboard must be taught to recognize each word—or, more accurately, how the user pronounces each word. This is accomplished with a training session using the same BASIC program. As the computer displays each word from the vocabulary, you pronounce it at least three times. Of course, the keyboard doesn't know whether the pronunciation is correct—it can't even distinguish English from Greek or Chinese. It merely associates your pronunciation with the vocabulary word.

Voiceprinting

How does this work? The keyboard converts the sound into a pattern of zeros and ones called a *voiceprint*. As you speak, the keyboard tries to match what it hears with a previously recorded voiceprint stored in its memory. If it finds a match, the keyboard sends the appropriate word to the PC, just as though the word had been typed.

Voiceprints are stored on disk so you won't have to retrain the keyboard each morning, and the keyboard lets you mix spoken and typed input.

Following this procedure, I trained my keyboard for six words and said, "Now type this." The screen remained blank.

The manual advises, "Your voiceprints in the morning are slightly different from your voiceprints in the afternoon. Therefore you can train a vocabulary in the morning, then in the afternoon update it a few passes to build some variation into the voiceprints." With that in mind, I built variation into my vocabulary, and tried again: "Now type this." The keyboard responded by typing *this* on the computer screen.

On the subject of recognition, the manual continues, "First-time users of speech recognition products usually have poor recognition for the first few days. After working with the equipment, the ability to achieve good recognition improves dramatically. The reason for this improvement is learning to relax." When I relaxed and spoke more slowly (and stopped eating pretzels and drinking beer), the keyboard performed beautifully: *type this now type blue now type this blue*. But there's only so much that can be written with six or even 160 words. And 160 words is the vocabulary limit.

Of course, the keyboard has more serious uses than accommodating a lazy writer. As a relatively inexpensive ($995) speech-recognition product for the IBM PC, it has both industrial and personal applications. Voice recognition can be a big help to the physically handicapped. One of Boeing Computer Systems' sharpest programmers is a quadraplegic who uses a workstation built around an IBM PC-XT and a Key Tronic speech-recognition keyboard. He writes programs in BASIC and Pascal and develops spreadsheets using *Lotus 1-2-3*.

Voice recognition for the IBM PC is not advanced enough that I could comfortably write this column from a horizontal position—but surely the Key Tronic keyboard is a promise of things to come. ©

C. Regena

The States Of The Union

I have previously written programs to identify the states and their capitals in certain regions ("Southern States," COMPUTE!, August 1984; "Western States" and "New England States," *Programmer's Reference Guide to the TI-99/4A*, COMPUTE! Books). Now, after numerous requests, I'll demonstrate a program for the middle Atlantic states.

The central portion of the United States can be outlined on a map as I did with the Southern States and Western States programs. However, some of the middle Atlantic states such as Delaware, Maryland, and New Jersey are too small to outline adequately on the TI-99/4A. The color-blinking method used in the New England States program won't work, either, because more color sets are needed than are available. Therefore, I've used a slightly different method for the region which includes Delaware, Maryland, New Jersey, New York, Pennsylvania, Virginia, and West Virginia. The states are drawn in high-resolution graphics with some color sets used for more than one state.

When a question mark appears on a state, the student types in the state's name. If the response is correct, the program asks for the state capital.

Be careful typing the DATA statements—make sure the commas are placed correctly. Lines 290–380 contain definitions for graphics characters. Lines 820–840 contain sets of row, column, and character numbers for drawing the graphics.

If you wish to save typing effort, you can obtain a copy of this program by sending a blank cassette or disk, a stamped, self-addressed mailer, and $3 to:

C. Regena
P.O. Box 1502
Cedar City, UT 84720

Please be sure to specify the title ("Middle Atlantic States") and that you need the TI version.

Middle Atlantic States

```
100 REM  NORTH ATLANTIC
110 CALL CLEAR
120 PRINT "  ************
    ***********":"  *";TA
    B(25);"*"
130 PRINT "  * IDENTIFY T
    HE STATES *":"  *";TA
    B(25);"*"
140 PRINT "  ************
    ***********"
150 PRINT ::"{3 SPACES}NO
    RTH ATLANTIC STATES"
160 PRINT :::"TYPE THE NA
    ME OF THE STATE  THEN
     PRESS <ENTER>."
170 PRINT :"IF THE STATE
    IS CORRECT,
    {4 SPACES}TYPE THE CA
    PITAL CITY."
180 PRINT :"NAMES MUST BE
     SPELLED{7 SPACES}COR
    RECTLY TO BE ACCEPTED
    ."
190 PRINT :"PRESS FCTN 8
    REDO TO ERASE."
200 CALL CHAR(33,"3F3F3F3
    F3F3F3F3F")
210 CALL CHAR(35,"FFFFFFF
    FFFFFFFFF")
220 CALL CHAR(43,"F8F8F8F
    8F0F0F0F")
230 CALL CHAR(45,"F0F0F0F
    8F8FCFCFE")
240 CALL CHAR(47,"FFFFFFF
    FFFFFFFFF")
250 FOR C=96 TO 159
260 READ C$
270 CALL CHAR(C,C$)
280 NEXT C
290 DATA FFFFFEFEFCFCFCF8
    ,F0F0F0F0F0F8FEFF,FFF
    FFEFCFCFCF8F8,3F3F3F3
    F3F3F3F3F,0
300 DATA 0,,,00000103073F
    FFFF,0303070767 6FFFFF
310 DATA 00000101030 30F0F
    ,0F0F0F0F0F0703,C0C
    0808080 80E0E0,F8F8F8F
    0F0E0C08,FFFFFFFEFCFC
    F8F
320 DATA F0E0602,80F8FEFF
    FFFFFFFF,F7E0C0C0808,
    F0E0C08,FFFFFFFFFFF2F
    0C
330 DATA FFFFFFFFFCFCF8F8
    ,F0F0F0F0C,FFFFFFFFFF
    F8,FF3F3F3F3F3F1F,030
    303030303 0303
340 DATA FFFFFFFCF8F0C,FF
    FF,,,,,,FFFFFFFFFF3F1
    F1F,0F070301,FF7FFFFF
    FF3F1F0F,FFFEFE,,,,,F
    FFFFFFFFFFFFFFF
350 DATA DFDF9F9F9F9F9F9F
    ,9F9F9F9F9F8FCFC7,,,,
    ,,FFFFFFFFFFFFFFFF,00
    0000000000 30F3F,000000
    01071FFFFF
360 DATA 0000009 8FCFFFFFF
    ,F8FCFEFFFFFFFFFF,FFF
    FFFFFEFEC7E,C0E0E0F0
    C00080C,00000080C0E0F
    0F
370 DATA FFFFFE7E1E0E,FCF
    CFCFCFCFCFCFC,0103070
    F1F3F7FFF,0103071F3FF
    FFFFF,00000000000 0FFF
    F
380 DATA 0000000000 03F3F1F
    ,1F0F0F0F1F3FFF,010
    30307 0F0F0F0F
390 CALL COLOR(1,7,1)
400 CALL COLOR(9,7,12)
410 CALL COLOR(10,12,1)
420 CALL COLOR(11,12,3)
430 CALL COLOR(12,14,12)
440 CALL COLOR(13,14,3)
450 CALL COLOR(14,14,1)
460 CALL COLOR(15,3,1)
470 CALL COLOR(16,3,1)
480 RESTORE 490
490 DATA NEW YORK,ALBANY,
    7,25,PENNSYLVANIA,HAR
    RISBURG,10,21,NEW JER
    SEY,TRENTON,11,26
500 DATA WEST VIRGINIA,CH
    ARLESTON,15,17,MARYLA
    ND,ANNAPOLIS,13,23
510 DATA DELAWARE,DOVER,1
    4,25,VIRGINIA,RICHMON
    D,17,21
520 FOR L=1 TO 7
530 READ S$(L),CAP$(L),X(
    L),Y(L)
540 NEXT L
550 PRINT :"PRESS <ENTER>
     TO START.";
560 CALL KEY(0,K,S)
570 IF K<>13 THEN 560
580 CALL CLEAR
590 CALL SCREEN(8)
600 CALL COLOR(2,16,1)
610 PRINT TAB(16);"!#####
    ##"
620 PRINT TAB(16);"!#####
    #`p"
630 PRINT TAB(16);"!#####
    #adl"
```

```
640 PRINT TAB(16);"c#####
    #bdm"
650 PRINT TAB(15);"hdxyz
    {3 SPACES}+n"
660 PRINT TAB(14);"iddddq
    {3 SPACES}-o"
670 PRINT TAB(13);"jdddsr
    {4 SPACES}/"
680 PRINT TAB(13);"kddt"
690 PRINT TAB(14);"wvu"::
    :::::
700 CALL VCHAR(3,27,153,3
    )
710 CALL VCHAR(6,27,144,4
    )
720 CALL VCHAR(3,26,144,7
    )
730 CALL VCHAR(4,25,144,5
    )
740 CALL VCHAR(5,24,144,4
    )
750 CALL HCHAR(7,21,144,3
    )
760 CALL HCHAR(8,20,144,4
    )
770 RESTORE 820
780 FOR I=1 TO 28
790 READ R,C,G
800 CALL HCHAR(R,C,G)
810 NEXT I
820 DATA 3,25,154,4,24,15
    4,6,23,155,6,22,156,6
    ,21,156,6,20,157,7,20
    ,158,8,19,159,10,27,1
    52,10,28,146
830 DATA 10,29,147,13,22,
    128,13,23,136,13,24,1
    36,14,22,129,14,23,13
    6,14,24,137,15,23,130
    ,15,24,138
840 DATA 16,25,131,16,24,
    151,16,23,144,17,24,1
    50,17,23,149,18,23,14
    8,18,24,147,18,14,146
    ,17,15,145
850 CALL VCHAR(15,22,144,
    4)
860 CALL VCHAR(15,21,144,
    4)
870 CALL VCHAR(16,20,144,
    3)
880 CALL VCHAR(16,19,144,
    3)
890 CALL HCHAR(18,15,144,
    4)
900 FOR C=1 TO 7
910 T=0
920 RANDOMIZE
930 R=INT(7*RND)+1
940 IF S$(R)="" THEN 930
950 CALL GCHAR(X(R),Y(R),
    G)
960 CALL HCHAR(20,1,100,1
    60)
970 FOR L=1 TO 7
980 CALL HCHAR(21,2+L,ASC
    (SEG$("STATE ?",L,1))
    )
990 CALL HCHAR(X(R),Y(R),
    32)
1000 CALL HCHAR(X(R),Y(R)
    ,63)
1010 NEXT L
1020 CALL HCHAR(21,11,100
    ,15)
1030 S1$=""
1040 CALL SOUND(150,1397,
    2)
```

```
1050 FOR L=1 TO 15
1060 CALL KEY(0,K,S)
1070 IF S<1 THEN 1060
1080 IF K=13 THEN 1130
1090 IF K=6 THEN 1020
1100 CALL HCHAR(21,10+L,K
     )
1110 S1$=S1$&CHR$(K)
1120 NEXT L
1130 CALL SOUND(100,880,2
     )
1140 IF S$(R)=S1$ THEN 12
     60
1150 CALL SOUND(100,330,2
     )
1160 CALL SOUND(100,262,2
     )
1170 T=T+1
1180 IF T<2 THEN 1020
1190 CALL HCHAR(21,11,100
     ,15)
1200 FOR L=1 TO LEN(S$(R)
     )
1210 CALL HCHAR(21,10+L,A
     SC(SEG$(S$(R),L,1)))
1220 NEXT L
1230 GOSUB 1600
1240 C=C-1
1250 GOTO 1570
1260 GOSUB 1670
1270 FOR L=1 TO 9
1280 CALL HCHAR(23,2+L,AS
     C(SEG$("CAPITAL ?",L
     ,1)))
1290 NEXT L
1300 T=0
1310 CALL HCHAR(23,13,100
     ,15)
1320 S1$=""
1330 CALL SOUND(150,1397,
     2)
1340 FOR L=1 TO 15
1350 CALL KEY(0,K,S)
1360 IF S<1 THEN 1350
1370 IF K=13 THEN 1430
1380 IF K=6 THEN 1310
1390 CALL HCHAR(23,12+L,K
     )
1400 S1$=S1$&CHR$(K)
1410 NEXT L
1420 CALL SOUND(100,880,2
     )
1430 IF CAP$(R)=S1$ THEN
     1550
1440 CALL SOUND(100,330,2
     )
1450 CALL SOUND(100,262,2
     )
1460 T=T+1
1470 IF T<2 THEN 1310
1480 CALL HCHAR(23,12,100
     ,15)
1490 FOR L=1 TO LEN(CAP$(
     R))
1500 CALL HCHAR(23,12+L,A
     SC(SEG$(CAP$(R),L,1)
     ))
1510 NEXT L
1520 GOSUB 1600
1530 C=C-1
1540 GOTO 1570
1550 GOSUB 1670
1560 S$(R)=""
1570 CALL HCHAR(X(R),Y(R)
     ,G)
1580 NEXT C
1590 GOTO 1720
1600 FOR L=1 TO 11
```

```
1610 CALL HCHAR(24,20+L,A
     SC(SEG$("PRESS ENTER
     ",L,1)))
1620 NEXT L
1630 CALL KEY(0,K,S)
1640 IF K<>13 THEN 1630
1650 CALL HCHAR(24,21,100
     ,11)
1660 RETURN
1670 CALL SOUND(100,262,2
     )
1680 CALL SOUND(100,330,2
     )
1690 CALL SOUND(100,392,2
     )
1700 CALL SOUND(200,523,2
     )
1710 RETURN
1720 CALL CLEAR
1730 PRINT "TRY AGAIN?  Y
      OR N"
1740 CALL KEY(0,K,S)
1750 IF K=89 THEN 480
1760 IF K<>78 THEN 1740
1770 CALL CLEAR
1780 END                ©
```

COMPUTE!'s Guide To Typing In Programs

Before typing in any program, you should familiarize yourself with your computer. Learn how to use the keyboard to type in and correct BASIC programs. Read your manuals to understand how to save and load BASIC programs to and from your disk drive or cassette unit. Computers are precise—take special care to type the program *exactly* as listed, including any necessary punctuation and symbols, except for special characters as noted below. To help you with this task, we have implemented a special listing convention as well as a program to help check your typing—the "Automatic Proofreader." Please read the following notes before typing in any programs from COMPUTE!. They can save you a lot of time and trouble.

Commodore, Apple, and Atari programs can contain some hard-to-read (and hard-to-type) special characters, so we have developed a listing system that indicates the function of these control characters. (There are no special control characters in our IBM or TI-99/4A listings.) You will find Commodore and Atari special characters within curly braces; *do not type the braces*. For example, {CLEAR} or {CLR} instructs you to insert the symbol which clears the screen on the Atari or Commodore machines. For Commodore, Apple, and Atari, a symbol by itself within curly braces is usually a control key or graphics key. If you see {A}, hold down the CTRL key and press A. This will produce a reverse video character on the Commodore (in quote mode), a graphics character on the Atari, and an invisible control character on the Apple. Commodore computers also have a special control key labeled with the Commodore logo. Graphics characters entered with the Commodore logo key are enclosed in a special bracket that looks like this: [<A>]. In this case, you would hold down the Commodore logo key as you type A. Our Commodore listings are in uppercase, so shifted symbols are underlined. A graphics heart symbol (SHIFT-S) would be listed as S. One exception is {SHIFT-SPACE}. When you see this, hold down SHIFT and press the space bar. If a number precedes a symbol, such as {5 RIGHT}, {6 S}, or [<8 Q>], you would enter five cursor rights, six shifted S's, or eight Commodore-Q's. On the Atari, inverse characters (printed in white on black) should be entered after pressing the inverse video key.

Since spacing is sometimes important, any more than two spaces will be listed. For example, {6 SPACES} means to press the space bar six times. Our listings never leave a space at the end of a line, instead moving it to the next printed line as {SPACE}. For your convenience, we have prepared this quick-reference chart for the Commodore and Atari special characters:

Atari 400/800/XL/XE

When you see	Type	See	
{CLEAR}	ESC SHIFT <	↖	Clear Screen
{UP}	ESC CTRL -	↑	Cursor Up
{DOWN}	ESC CTRL =	↓	Cursor Down
{LEFT}	ESC CTRL +	←	Cursor Left
{RIGHT}	ESC CTRL *	→	Cursor Right
{BACK S}	ESC DELETE	◀	Backspace
{DELETE}	ESC CTRL DELETE	◧	Delete character
{INSERT}	ESC CTRL INSERT	◨	Insert character
{DEL LINE}	ESC SHIFT DELETE	⊟	Delete line
{INS LINE}	ESC SHIFT INSERT	⊞	Insert line
{TAB}	ESC TAB	▶	TAB key
{CLR TAB}	ESC CTRL TAB	◧	Clear tab
{SET TAB}	ESC SHIFT TAB	◨	Set tab stop
{BELL}	ESC CTRL 2	◳	Ring buzzer
{ESC}	ESC ESC	⇤	ESCape key

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

When You Read:	Press:		See:	When You Read:	Press:		See:
{CLR}	SHIFT	CLR/HOME	♥	[< 1 >]	COMMODORE	1	✚
{HOME}		CLR/HOME	S	[< 2 >]	COMMODORE	2	◤
{UP}	SHIFT	↑ CRSR ↓	●	[< 3 >]	COMMODORE	3	☒
{DOWN}		↑ CRSR ↓	Q	[< 4 >]	COMMODORE	4	▢
{LEFT}	SHIFT	← CRSR →	▐	[< 5 >]	COMMODORE	5	⊡
{RIGHT}		← CRSR →	]	[< 6 >]	COMMODORE	6	▌
{RVS}	CTRL	9	R	[< 7 >]	COMMODORE	7	◆
{OFF}	CTRL	0	▬	[< 8 >]	COMMODORE	8	▚
{BLK}	CTRL	1	█	{ F1 }		f1	▬
{WHT}	CTRL	2	E	{ F2 }	SHIFT	f1	◥
{RED}	CTRL	3	£	{ F3 }		f3	▬
{CYN}	CTRL	4	◣	{ F4 }	SHIFT	f3	◣
{PUR}	CTRL	5	▒	{ F5 }		f5	▌
{GRN}	CTRL	6	↑	{ F6 }	SHIFT	f5	◢
{BLU}	CTRL	7	←	{ F7 }		f7	▐
{YEL}	CTRL	8	π	{ F8 }	SHIFT	f7	█
				←	←		▒

The Automatic Proofreader

We have developed a series of simple, yet effective programs that can help check your typing. Type in the appropriate Proofreader program listed below, then save it for future use. On the VIC, 64, or Atari, run the Proofreader to activate it, then enter NEW to erase the BASIC loader (the Proofreader remains active, hidden in memory, as a machine language program). Pressing RUN/STOP–RESTORE or SYSTEM RESET deactivates the Proofreader. You can use SYS 886 to reactivate the VIC/64 Proofreader, or PRINT USR(1536) to reenable the Atari Proofreader. On the Apple, the Proofreader automatically erases the BASIC portion of itself after you activate it by typing RUN, leaving only the machine language portion in memory. It works with either DOS 3.3 or ProDOS. Disable the Apple Proofreader by pressing CTRL-RESET before running another BASIC program. The IBM Proofreader is a BASIC program that simulates the IBM BASIC line editor, letting you enter, edit, list, save, and load programs that you type. Type RUN to activate.

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

All you need to do is compare the value provided by the Proofreader with the checksum printed in the program listing in the magazine. In Commodore listings, the checksum is a number from 0 to 255. It is set off from the rest of the line with *rem*. This prevents a syntax error if the checksum is typed in, but the REM statements and checksums need *not* be typed in. It is just there for your information.

In Atari, Apple, and IBM listings, the checksum is given to the left of each line number. Just type in the program one line at a time (without the printed checksum) and compare the checksum generated by the Proofreader to the checksum in the listing. If they match, go on to the next line. If not, check your typing: You've made a mistake. On the Commodore, Atari, and Apple Proofreaders, spaces are not counted as part of the checksum, so be sure you type the right number of spaces between quote marks. The Commodore and Atari Proofreaders do not check to see that you've typed the characters in the right order, so if characters are transposed, the checksum still matches the listing. Because of the checksum method used, do not type abbreviations, such as ? for PRINT. The IBM Proofreader is the pickiest of all; it *will* detect errors in spacing and transposition. Be sure to leave Caps Lock on, except when typing lowercase characters.

IBM Proofreader Commands

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

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

Special Proofreader Notes For Commodore Cassette Users

The Proofreader resides in a section of memory called the cassette buffer, which is used during tape LOADs and SAVEs. Therefore, be sure to press RUN/STOP–RESTORE to get the Proofreader out of the way before saving or loading a program. If you want to use the Proofreader with tape, run the Proofreader, then enter these two lines *exactly* as shown, pressing RETURN after each one:

```
A$="PROOFREADER.T":B$="{10
  SPACES}":FOR X=1 TO 4:A$=A$
  +B$:NEXT
FOR X=886 TO 1018:A$=A$+CHR$
(PEEK(X)):NEXT:OPEN 1,1,1,A$:
  CLOSE1
```

Then insert a blank tape and press RECORD and PLAY to save a special version of the Proofreader. Anytime you need to reload the Proofreader after it has been erased—for example, after you reload a paritally completed program—just rewind the tape, type OPEN1:CLOSE1, then press PLAY. You'll see the message FOUND PROOFREADER.T, but not the familiar LOADING message. Don't worry; the Proofreader is in memory. When READY comes back, enter SYS 886.

Program 1: VIC/64 Proofreader

By Charles Brannon, Program Editor

```
10 PRINT"{CLR}PLEASE WAIT...":
   FORI=886TO1Ø18:READA:CK=CK+
   A:POKEI,A:NEXT
2Ø IF CK<>17539 THEN PRINT"
   {DOWN}YOU MADE AN ERROR":PR
   INT"IN DATA STATEMENTS.":EN
   D
3Ø SYS886:PRINT"{CLR}{2 DOWN}P
   ROOFREADER ACTIVATED.":NEW
4Ø DATA 173,Ø36,ØØ3,2Ø1,15Ø,2Ø
   8,ØØ1,Ø96,141,151,ØØ3,173
5Ø DATA Ø37,ØØ3,141,152,ØØ3,16
   9,15Ø,141,Ø36,ØØ3,169,ØØ3
6Ø DATA 141,Ø37,ØØ3,169,ØØØ,13
   3,254,Ø96,Ø32,Ø87,241,133
7Ø DATA 251,134,252,132,253,ØØ
   8,2Ø1,Ø13,24Ø,Ø17,2Ø1,Ø32
8Ø DATA 24Ø,ØØ5,Ø24,1Ø1,254,13
   3,254,165,251,166,252,164
9Ø DATA 253,Ø4Ø,Ø96,169,Ø13,Ø3
   2,21Ø,255,165,214,141,251
1ØØ DATA ØØ3,2Ø6,251,ØØ3,169,Ø
    ØØ,133,216,169,Ø19,Ø32,21Ø
11Ø DATA 255,169,Ø18,Ø32,21Ø,2
    55,169,58,Ø32,21Ø,255,166
12Ø DATA 254,169,ØØØ,133,254,1
    72,151,ØØ3,192,Ø87,2Ø8,ØØ6
13Ø DATA Ø32,2Ø5,189,Ø76,235,Ø
    Ø3,Ø32,2Ø5,221,169,Ø32,Ø32
14Ø DATA 21Ø,255,Ø32,21Ø,255,1
    73,251,ØØ3,133,214,Ø76,173
15Ø DATA ØØ3
```

Program 2: Atari Proofreader

By Charles Brannon, Program Editor

```
1ØØ GRAPHICS Ø
11Ø FOR I=1536 TO 17ØØ:RE
    AD A:POKE I,A:CK=CK+A
    :NEXT I
12Ø IF CK<>19Ø72 THEN ? "
    Error in DATA Stateme
    nts.  Check Typing.":
    END
13Ø A=USR(1536)
14Ø ? :? "Automatic Proof
    reader Now Activated.
    "
15Ø END
16Ø DATA 1Ø4,16Ø,Ø,185,26
    ,3,2Ø1,69,24Ø,7
17Ø DATA 2ØØ,2ØØ,192,34,2
    Ø8,243,96,2ØØ,169,74
18Ø DATA 153,26,3,2ØØ,169
    ,6,153,26,3,162
19Ø DATA Ø,189,Ø,228,157,
    74,6,232,224,16
2ØØ DATA 2Ø8,245,169,93,1
    41,78,6,169,6,141
21Ø DATA 79,6,24,173,4,22
    8,1Ø5,1,141,95
```

```
220 DATA 6,173,5,228,105,
    Ø,141,96,6,169
230 DATA Ø,133,2Ø3,96,247
    ,238,125,241,93,6
24Ø DATA 244,241,115,241,
    124,241,76,2Ø5,238
25Ø DATA Ø,Ø,Ø,Ø,Ø,32,62,
    246,8,2Ø1
26Ø DATA 155,24Ø,13,2Ø1,3
    2,24Ø,7,72,24,1Ø1
27Ø DATA 2Ø3,133,2Ø3,1Ø4,
    4Ø,96,72,152,72,138
28Ø DATA 72,16Ø,Ø,169,128
    ,145,88,2ØØ,192,4Ø
29Ø DATA 2Ø8,249,165,2Ø3,
    74,74,74,74,24,1Ø5
3ØØ DATA 161,16Ø,3,145,88
    ,165,2Ø3,41,15,24
31Ø DATA 1Ø5,161,2ØØ,145,
    88,169,Ø,133,2Ø3,1Ø4
32Ø DATA 17Ø,1Ø4,168,1Ø4,
    4Ø,96
```

Program 3: IBM Proofreader

By Charles Brannon, Program Editor

```
1Ø 'Automatic Proofreader Ver
   sion 2.ØØ (Lines 27Ø,51Ø,5
   15,517,62Ø,63Ø changed fro
   m V1.Ø)
1ØØ DIM L$(5ØØ),LNUM(5ØØ):COL
    OR Ø,7,7:KEY OFF:CLS:MAX=
    Ø:LNUM(Ø)=65536!
11Ø ON ERROR GOTO 12Ø:KEY 15,
    CHR$(4)+CHR$(7Ø):ON KEY(1
    5) GOSUB 64Ø:KEY (15) ON:
    GOTO 13Ø
12Ø RESUME 13Ø
13Ø DEF SEG=&H4Ø:W=PEEK(&H4A)
14Ø ON ERROR GOTO 65Ø:PRINT:P
    RINT"Proofreader Ready."
15Ø LINE INPUT L$:Y=CSRLIN-IN
    T(LEN(L$)/W)-1:LOCATE Y,1
16Ø DEF SEG=Ø:POKE 1Ø5Ø,3Ø:PO
    KE 1Ø52,34:POKE 1Ø54,Ø:PO
    KE 1Ø55,79:POKE 1Ø56,13:P
    OKE 1Ø57,28:LINE INPUT L$
    :DEF SEG:IF L$="" THEN 15
    Ø
17Ø IF LEFT$(L$,1)=" " THEN L
    $=MID$(L$,2):GOTO 17Ø
18Ø IF VAL(LEFT$(L$,2))=Ø AND
     MID$(L$,3,1)=" " THEN L$
    =MID$(L$,4)
19Ø LNUM=VAL(L$):TEXT$=MID$(L
    $,LEN(STR$(LNUM))+1)
2ØØ IF ASC(L$)>57 THEN 26Ø 'n
    o line number, therefore
    command
21Ø IF TEXT$="" THEN GOSUB 54
    Ø:IF LNUM=LNUM(P) THEN GO
    SUB 56Ø:GOTO 15Ø ELSE 15Ø
22Ø CKSUM=Ø:FOR I=1 TO LEN(L$
    ):CKSUM=(CKSUM+ASC(MID$(L
    $,I))*I) AND 255:NEXT:LOC
    ATE Y,1:PRINT CHR$(65+CKS
    UM/16)+CHR$(65+(CKSUM AND
     15))+" "+L$
23Ø GOSUB 54Ø:IF LNUM(P)=LNUM
     THEN L$(P)=TEXT$:GOTO 15
    Ø 'replace line
24Ø GOSUB 58Ø:GOTO 15Ø 'inser
    t the line
26Ø TEXT$="":FOR I=1 TO LEN(L
    $):A=ASC(MID$(L$,I)):TEXT
    $=TEXT$+CHR$(A+32*(A>96 A
    ND A<123)):NEXT
27Ø DELIMITER=INSTR(TEXT$," "
    ):COMMAND$=TEXT$:ARG$="":
    IF DELIMITER THEN COMMAND
    $=LEFT$(TEXT$,DELIMITER-1
    ):ARG$=MID$(TEXT$,DELIMIT
    ER+1) ELSE DELIMITER=INST
    R(TEXT$,CHR$(34)):IF DELI
    MITER THEN COMMAND$=LEFT$
    (TEXT$,DELIMITER-1):ARG$=
    MID$(TEXT$,DELIMITER)
28Ø IF COMMAND$<>"LIST" THEN
    41Ø
29Ø OPEN "scrn:" FOR OUTPUT A
    S #1
3ØØ IF ARG$="" THEN FIRST=Ø:P
    =MAX-1:GOTO 34Ø
31Ø DELIMITER=INSTR(ARG$,"-")
    :IF DELIMITER=Ø THEN LNUM
    =VAL(ARG$):GOSUB 54Ø:FIRS
    T=P:GOTO 34Ø
32Ø FIRST=VAL(LEFT$(ARG$,DELI
    MITER)):LAST=VAL(MID$(ARG
    $,DELIMITER+1))
33Ø LNUM=FIRST:GOSUB 54Ø:FIRS
    T=P:LNUM=LAST:GOSUB 54Ø:I
    F P=Ø THEN P=MAX-1
34Ø FOR X=FIRST TO P:N$=MID$(
    STR$(LNUM(X)),2)+" "
35Ø IF CKFLAG=Ø THEN A$="":GO
    TO 37Ø
36Ø CKSUM=Ø:A$=N$+L$(X):FOR I
    =1 TO LEN(A$):CKSUM=(CKSU
    M+ASC(MID$(A$,I))*I) AND
    255:NEXT:A$=CHR$(65+CKSUM
    /16)+CHR$(65+(CKSUM AND 1
    5))+" "
37Ø PRINT #1,A$+N$+L$(X)
38Ø IF INKEY$<>"" THEN X=P
39Ø NEXT :CLOSE #1:CKFLAG=Ø
4ØØ GOTO 13Ø
41Ø IF COMMAND$="LLIST" THEN
    OPEN "lpt1:" FOR OUTPUT A
    S #1:GOTO 3ØØ
42Ø IF COMMAND$="CHECK" THEN
    CKFLAG=1:GOTO 29Ø
43Ø IF COMMAND$<>"SAVE" THEN
    45Ø
44Ø GOSUB 6ØØ:OPEN ARG$ FOR O
    UTPUT AS #1:ARG$="":GOTO
    3ØØ
45Ø IF COMMAND$<>"LOAD" THEN
    49Ø
46Ø GOSUB 6ØØ:OPEN ARG$ FOR I
    NPUT AS #1:MAX=Ø:P=Ø
47Ø WHILE NOT EOF(1):LINE INP
    UT #1,L$:LNUM(P)=VAL(L$):
    L$(P)=MID$(L$,LEN(STR$(VA
    L(L$)))+1):P=P+1:WEND
48Ø MAX=P:CLOSE #1:GOTO 13Ø
49Ø IF COMMAND$="NEW" THEN IN
    PUT "Erase program - Are
    you sure";L$:IF LEFT$(L$,
    1)="y" OR LEFT$(L$,1)="Y"
     THEN MAX=Ø:GOTO 13Ø:ELSE
     13Ø
5ØØ IF COMMAND$="BASIC" THEN
    COLOR 7,Ø,Ø:ON ERROR GOTO
     Ø:CLS:END
51Ø IF COMMAND$<>"FILES" THEN
     52Ø
515 IF ARG$="" THEN ARG$="A:"
     ELSE SEL=1:GOSUB 6ØØ
517 FILES ARG$:GOTO 13Ø
52Ø PRINT"Syntax error":GOTO
    13Ø
54Ø P=Ø:WHILE LNUM>LNUM(P) AN
    D P<MAX:P=P+1:WEND:RETURN
56Ø MAX=MAX-1:FOR X=P TO MAX:
    LNUM(X)=LNUM(X+1):L$(X)=L
    $(X+1):NEXT:RETURN
58Ø MAX=MAX+1:FOR X=MAX TO P+
    1 STEP -1:LNUM(X)=LNUM(X-
    1):L$(X)=L$(X-1):NEXT:L$(
    P)=TEXT$:LNUM(P)=LNUM:RET
    URN
6ØØ IF LEFT$(ARG$,1)<>CHR$(34
    ) THEN 52Ø ELSE ARG$=MID$
    (ARG$,2)
61Ø IF RIGHT$(ARG$,1)=CHR$(34
    ) THEN ARG$=LEFT$(ARG$,LE
    N(ARG$)-1)
62Ø IF SEL=Ø AND INSTR(ARG$,"
    .")=Ø THEN ARG$=ARG$+".BA
    S"
63Ø SEL=Ø:RETURN
64Ø CLOSE #1:CKFLAG=Ø:PRINT"S
    topped.":RETURN 15Ø
65Ø PRINT "Error #";ERR:RESUM
    E 15Ø
```

Program 4: Apple Proofreader

By Tim Victor, Editorial Programmer

```
1Ø C = Ø: FOR I = 768 TO 768 +
   68: READ A:C = C + A: POKE I
   ,A: NEXT
2Ø IF C < > 7258 THEN PRINT "ER
   ROR IN PROOFREADER DATA STAT
   EMENTS": END
3Ø IF PEEK (19Ø * 256) < > 76 T
   HEN POKE 56,Ø: POKE 57,3: CA
   LL 1ØØ2: GOTO 5Ø
4Ø PRINT CHR$ (4);"IN#A$3ØØ"
5Ø POKE 34,Ø: HOME : POKE 34,1:
    VTAB 2: PRINT "PROOFREADER
   INSTALLED"
6Ø NEW
1ØØ DATA 216,32,27,253,2Ø1,141
11Ø DATA 2Ø8,6Ø,138,72,169,Ø
12Ø DATA 72,189,255,1,2Ø1,16Ø
13Ø DATA 24Ø,8,1Ø4,1Ø,125,255
14Ø DATA 1,1Ø5,Ø,72,2Ø2,2Ø8
15Ø DATA 238,1Ø4,17Ø,41,15,9
16Ø DATA 48,2Ø1,58,144,2,233
17Ø DATA 57,141,1,4,138,74
18Ø DATA 74,74,74,41,15,9
19Ø DATA 48,2Ø1,58,144,2,233
2ØØ DATA 57,141,Ø,4,1Ø4,17Ø
21Ø DATA 169,141,96
```

©

MLX Machine Language Entry Program For Commodore 64

Charles Brannon, Program Editor

MLX is a labor-saving utility that allows almost fail-safe entry of machine language programs published in COMPUTE!. *You need to know nothing about machine language to use MLX—it was designed for everyone. At least 8K expansion memory is required.*

MLX is a new way to enter long machine language (ML) programs with a minimum of fuss. MLX lets you enter the numbers from a special list that looks similar to BASIC DATA statements. It checks your typing on a line-by-line basis. It won't let you enter illegal characters when you should be typing numbers. It won't let you enter numbers greater than 255 (forbidden in ML). It won't let you enter the wrong numbers on the wrong line. In addition, MLX creates a ready-to-use tape or disk file.

Using MLX

Type in and save the appropriate version of MLX (you'll want to use it in the future). When you're ready to type in an ML program, run MLX. MLX for the 64 asks you for two numbers: the starting address and the ending address. These numbers are given in the article accompanying the ML program.

When you run MLX, you'll see a prompt corresponding to the starting address. The prompt is the current line you are entering from the listing. It increases by six each time you enter a line. That's because each line has seven numbers—six actual data numbers plus a *checksum number*. The checksum verifies that you typed the previous six numbers correctly. If you enter any of the six numbers wrong, or enter the checksum wrong, the computer rings a buzzer and prompts you to reenter the line. If you enter it correctly, a bell tone sounds and you continue to the next line.

MLX accepts only numbers as input. If you make a typing error, press the INST/DEL key; the entire number is deleted. You can press it as many times as necessary back to the start of the line. If you enter three-digit numbers as listed, the computer automatically prints the comma and goes on to accept the next number. If you enter less than three digits, you can press either the space bar or RETURN key to advance to the next number. The checksum automatically appears in inverse video for emphasis.

To simplify your typing, MLX redefines part of the keyboard as a numeric keypad (lines 581–584):

```
    U  I  O                7  8  9
 H  J  K  L  become  0     4  5  6
    M  ,  .                1  2  3
```

64 MLX Commands

When you finish typing an ML listing (assuming you type it all in one session), you can then save the completed program on tape or disk. Follow the screen instructions. If you get any errors while saving, you probably have a bad disk, or the disk is full, or you've made a typo when entering the MLX program itself.

You don't have to enter the whole ML program in one sitting. MLX lets you enter as much as you want, save it, and then reload the file from tape or disk later. MLX recognizes these commands:

SHIFT-S: Save
SHIFT-L: Load
SHIFT-N: New Address
SHIFT-D: Display

When you enter a command, MLX jumps out of the line you've been typing, so we recommend you do it at a new prompt. Use the Save command to save what you've been working on. It will save on tape or disk, as if you've finished, but the tape or disk won't work, of course, until you finish the typing. Remember what address you stop at. The next time you run MLX, answer all the prompts as you did before, then insert the disk or tape. When you get to the entry prompt, press SHIFT-L to reload the partly completed file into memory. Then use the New Address command to resume typing.

To use the New Address command, press SHIFT-N and enter the address where you previously stopped. The prompt will change, and you can then continue typing. Always enter a New Address that matches up with one of the line numbers in the special listing, or else the checksum won't work. The Display command lets you display a section of your typing. After you press SHIFT-D, enter two addresses within the line number range of the listing. You can abort the listing by pressing any key.

64 MLX: Machine Language Entry

```
10 REM LINES CHANGED FROM MLX
   {SPACE}VERSION 2.00 ARE 750
   ,765,770 AND 860       :rem 50
20 REM LINE CHANGED FROM MLX V
   ERSION 2.01 IS 300 :rem 147
100 PRINT"{CLR}E63";CHR$(142);
    CHR$(8);:POKE53281,1:POKE5
    3280,1                :rem 67
101 POKE 788,52:REM DISABLE RU
    N/STOP               :rem 119
110 PRINT"{RVS}{39 SPACES}";
                         :rem 176
120 PRINT"{RVS}{14 SPACES}
    {RIGHT}{OFF}E*3£{RVS}
    {RIGHT} {RIGHT}{2 SPACES}
    E*3{OFF}E*3£{RVS}£{RVS}
    {14 SPACES}";        :rem 250
130 PRINT"{RVS}{14 SPACES}
    {RIGHT} EG3{RIGHT}
    {2 RIGHT} {OFF}£{RVS}£
    E*3{OFF}E*3{RVS}
    {14 SPACES}";         :rem 35
140 PRINT"{RVS}{41 SPACES}"
                         :rem 120
200 PRINT"{2 DOWN}{PUR}{BLK} M
    ACHINE LANGUAGE EDITOR VER
    SION 2.02{5 DOWN}":rem 238
210 PRINT"E53{2 UP}STARTING AD
    DRESS?{8 SPACES}{9 LEFT}";
                         :rem 143
215 INPUTS:F=1-F:C$=CHR$(31+11
    9*F)                 :rem 166
220 IFS<256OR(S>40960ANDS<4915
    2)ORS>53247THENGOSUB3000:G
    OTO210               :rem 235
225 PRINT:PRINT:PRINT :rem 180
230 PRINT"E53{2 UP}ENDING ADDR
    ESS?{8 SPACES}{9 LEFT}";:I
    NPUTE:F=1-F:C$=CHR$(31+119
    *F)                   :rem 20
240 IFE<256OR(E>40960ANDE<4915
    2)ORE>53247THENGOSUB3000:G
    OTO230               :rem 183
250 IFE<STHENPRINTC$;"{RVS}END
    ING < START{2 SPACES}":GOS
    UB1000:GOTO 230     :rem 176
260 PRINT:PRINT:PRINT :rem 179
300 PRINT"{CLR}";CHR$(14):AD=S
                          :rem 56
310 A=1:PRINTRIGHT$("0000"+MID
    $(STR$(AD),2),5);":";
                          :rem 33
315 FORJ=ATO6             :rem 33
320 GOSUB570:IFN=-1THENJ=J+N:G
    OTO320               :rem 228
390 IFN=-211THEN 710      :rem 62
400 IFN=-204THEN 790      :rem 64
410 IFN=-206THENPRINT:INPUT"
    {DOWN}ENTER NEW ADDRESS";Z
    Z                     :rem 44
415 IFN=-206THENIFZZ<SORZZ>ETH
    ENPRINT"{RVS}OUT OF RANGE"
    :GOSUB1000:GOTO410:rem 225
417 IFN=-206THENAD=ZZ:PRINT:GO
    TO310                :rem 238
420 IF N<>-196 THEN 480
                         :rem 133
430 PRINT:INPUT"DISPLAY:FROM";
    F:PRINT,"TO";:INPUTT
                         :rem 234
440 IFF<SORF>EORT<SORT>ETHENPR
    INT"AT LEAST";S;"{LEFT}, N
    OT MORE THAN";E:GOTO430
                         :rem 159
450 FORI=FTOTSTEP6:PRINT:PRINT
    RIGHT$("0000"+MID$(STR$(I)
    ,2),5);":";           :rem 30
451 FORK=0TO5:N=PEEK(I+K):PRIN
    TRIGHT$("00"+MID$(STR$(N),
    2),3);",";            :rem 66
```

```
460 GETA$:IFA$>""THENPRINT:PRI
    NT:GOTO310              :rem 25
470 NEXTK:PRINTCHR$(20);:NEXTI
    :PRINT:PRINT:GOTO310
                            :rem 50
480 IFN<0 THEN PRINT:GOTO310
                           :rem 168
490 A(J)=N:NEXTJ           :rem 199
500 CKSUM=AD-INT(AD/256)*256:F
    ORI=1TO6:CKSUM=(CKSUM+A(I)
    )AND255:NEXT           :rem 200
510 PRINTCHR$(18);:GOSUB570:PR
    INTCHR$(146);           :rem 94
511 IFN=-1THENA=6:GOTO315
                           :rem 254
515 PRINTCHR$(20):IFN=CKSUMTHE
    N530                   :rem 122
520 PRINT:PRINT"LINE ENTERED W
    RONG : RE-ENTER":PRINT:GOS
    UB1000:GOTO310         :rem 176
530 GOSUB2000              :rem 218
540 FORI=1TO6:POKEAD+I-1,A(I):
    NEXT:POKE54272,0:POKE54273
    ,0                     :rem 227
550 AD=AD+6:IF AD<E THEN 310
                           :rem 212
560 GOTO 710               :rem 108
570 N=0:Z=0                 :rem 88
580 PRINT"{£}";             :rem 81
581 GETA$:IFA$=""THEN581
                            :rem 95
582 AV=-(A$="M")-2*(A$=",")-3*
    (A$=".")-4*(A$="J")-5*(A$=
    "K")-6*(A$="L")         :rem 41
583 AV=AV-7*(A$="U")-8*(A$="I"
    )-9*(A$="O"):IFA$="H"THENA
    $="0"                  :rem 134
584 IFAV>0THENA$=CHR$(48+AV)
                           :rem 134
585 PRINTCHR$(20);:A=ASC(A$):I
    FA=13ORA=44ORA=32THEN670
                           :rem 229
590 IFA>128THENN=-A:RETURN
                           :rem 137
600 IFA<>20 THEN 630        :rem 10
610 GOSUB690:IFI=1ANDT=44THENN
    =-1:PRINT"{OFF}{LEFT}
    {LEFT}";:GOTO690        :rem 62
620 GOTO570                :rem 109
630 IFA<48ORA>57THEN580
                           :rem 105
640 PRINTA$;:N=N*10+A-48
                           :rem 106
650 IFN>255 THEN A=20:GOSUB100
    0:GOTO600              :rem 229
660 Z=Z+1:IFZ<3THEN580  :rem 71
670 IFZ=0THENGOSUB1000:GOTO570
                           :rem 114
680 PRINT",";:RETURN   :rem 240
690 S%=PEEK(209)+256*PEEK(210)
    +PEEK(211)             :rem 149
691 FORI=1TO3:T=PEEK(S%-I)
                            :rem 67
695 IFT<>44ANDT<>58THENPOKES%-
    I,32:NEXT              :rem 205
700 PRINTLEFT$("{3 LEFT}",I-1)
    ;:RETURN                 :rem 7
710 PRINT"{CLR}{RVS}*** SAVE *
    **{3 DOWN}"            :rem 236
715 PRINT"{2 DOWN}(PRESS {RVS}
    RETURN{OFF} ALONE TO CANCE
    L SAVE){DOWN}"         :rem 106
720 F$="":INPUT"{DOWN} FILENAM
    E";F$:IFF$=""THENPRINT:PRI
    NT:GOTO310              :rem 71
730 PRINT:PRINT"{2 DOWN}{RVS}T
    {OFF}APE OR {RVS}D{OFF}ISK
    : (T/D)"               :rem 228
740 GETA$:IFA$<>"T"ANDA$<>"D"T
    HEN740                  :rem 36
750 DV=1-7*(A$="D"):IFDV=8THEN
    F$="0:"+F$:OPEN15,8,15,"S"
    +F$:CLOSE15            :rem 212
760 T$=F$:ZK=PEEK(53)+256*PEEK
    (54)-LEN(T$):POKE782,ZK/25
    6                        :rem 3
762 POKE781,ZK-PEEK(782)*256:P
    OKE780,LEN(T$):SYS65469
                           :rem 109
763 POKE780,1:POKE781,DV:POKE7
    82,1:SYS65466           :rem 69
765 K=S:POKE254,K/256:POKE253,
    K-PEEK(254)*256:POKE780,25
    3                       :rem 17
766 K=E+1:POKE782,K/256:POKE78
    1,K-PEEK(782)*256:SYS65496
                           :rem 235
770 IF(PEEK(783)AND1)OR(191AND
    ST)THEN780             :rem 111
775 PRINT"{DOWN}DONE.{DOWN}":G
    OTO310                 :rem 113
780 PRINT"{DOWN}ERROR ON SAVE.
    {2 SPACES}TRY AGAIN.":IFDV
    =1THEN720              :rem 171
781 OPEN15,8,15:INPUT#15,E1$,E
    2$:PRINTE1$;E2$:CLOSE15:GO
    TO720                  :rem 103
790 PRINT"{CLR}{RVS}*** LOAD *
    **{2 DOWN}"            :rem 212
795 PRINT"{2 DOWN}(PRESS {RVS}
    RETURN{OFF} ALONE TO CANCE
    L LOAD)"                :rem 82
800 F$="":INPUT"{2 DOWN} FILEN
    AME";F$:IFF$=""THENPRINT:G
    OTO310                 :rem 144
810 PRINT:PRINT"{2 DOWN}{RVS}T
    {OFF}APE OR {RVS}D{OFF}ISK
    : (T/D)"               :rem 227
820 GETA$:IFA$<>"T"ANDA$<>"D"T
    HEN820                  :rem 34
830 DV=1-7*(A$="D"):IFDV=8THEN
    F$="0:"+F$             :rem 157
840 T$=F$:ZK=PEEK(53)+256*PEEK
    (54)-LEN(T$):POKE782,ZK/25
    6                        :rem 2
841 POKE781,ZK-PEEK(782)*256:P
    OKE780,LEN(T$):SYS65469
                           :rem 107
845 POKE780,1:POKE781,DV:POKE7
    82,1:SYS65466           :rem 70
850 POKE780,0:SYS65493 :rem 11
860 IF(PEEK(783)AND1)OR(191AND
    ST)THEN870             :rem 111
865 PRINT"{DOWN}DONE.":GOTO310
                            :rem 96
870 PRINT"{DOWN}ERROR ON LOAD.
    {2 SPACES}TRY AGAIN.{DOWN}
    ":IFDV=1THEN800        :rem 172
880 OPEN15,8,15:INPUT#15,E1$,E
    2$:PRINTE1$;E2$:CLOSE15:GO
    TO800                  :rem 102
1000 REM BUZZER            :rem 135
1001 POKE54296,15:POKE54277,45
     :POKE54278,165        :rem 207
1002 POKE54276,33:POKE 54273,6
     :POKE54272,5           :rem 42
1003 FORT=1TO200:NEXT:POKE5427
     6,32:POKE54273,0:POKE5427
     2,0:RETURN            :rem 202
2000 REM BELL SOUND         :rem 78
2001 POKE54296,15:POKE54277,0:
     POKE54278,247         :rem 152
2002 POKE 54276,17:POKE54273,4
     0:POKE54272,0          :rem 86
2003 FORT=1TO100:NEXT:POKE5427
     6,16:RETURN            :rem 57
3000 PRINTC$;"{RVS}NOT ZERO PA
     GE OR ROM":GOTO1000
                            :rem 89
```

©

To receive additional information from advertisers in this issue, use the handy reader service cards in the back of the magazine.

CAPUTE!

Modifications or Corrections To Previous Articles

Commodore 64 Disk Commander

This program from the September issue (p. 80) has a bug in the DOPEN command. *Do not* use DOPEN until you correct the problem. If you disassemble the relocated "Disk Commander" code, you'll notice the instruction CMP $062D,X at location $A797 in the DOPEN routine. The instruction should be CMP $026D,X. This portion of the routine is intended to assign a unique secondary address to each opened file, but the bug causes all files to be opened with the same secondary address. If you never DOPEN more than one file at a time, there's no problem. However, multiple DOPENs lead to improperly closed files, which are denoted in the disk directory by an asterisk next to the filename. If you see any of these so-called poison files on your disk, you should remove them with the VALIDATE command (OPEN 1,8,15,"V0:" :CLOSE 1), not the SCRATCH command.

To fix "Disk Commander," first enter POKE 44,20:POKE 5120,0:NEW to reconfigure memory, then load and run "MLX". Use the MLX Load command (SHIFT-L) to load your existing version of Disk Commander. Next, use the New Address command (SHIFT-N) to move to line 3591, then enter the following new data:

```
3591 :Ø48,ØØ7,221,1Ø9,ØØ2,24Ø,122
```

Now press SHIFT-S to call the MLX Save feature and save a copy of the corrected program.

Our thanks to reader Franz Paulsen for uncovering this bug.

Atari Animation With P/M Graphics, Part 2

Part of line 90 is missing in the program example in the first column on page 102 of this article from the October issue. It should read as follows:

```
NB 9Ø FOR X=PMBASE+1Ø24 TO PMBASE+2Ø48:
      POKE X,Ø:NEXT X
```

The Last Warrior

A number of readers have had difficulties with line 480 in the IBM version of this game program from the September issue (p. 54). The first character within quotes in that line is the lowercase letter l, *not* the numeral 1. The two characters do have a similar appearance in the listing, but 164 is not a reasonable parameter for a PLAY statement, while l64 is. ©

News & Products

Word Processor For IBM

Professional Software has introduced *Write 'n Spell*, a $149 word processor which contains an integrated 90,000-word dictionary that checks and corrects spelling. In addition to standard features found in most sophisticated word processors, *Write 'n Spell* also contains built-in mail-merge, graphing, and simultaneous typing-while-printing capabilities. The program will merge with *Lotus 1-2-3*, *pfs:File*, *Multiplan*, and many other popular application programs. The word processor also has pull-down HELP and OPTION windows.

Write 'n Spell is available for the IBM PC, PCjr, AT, and compatible computers.

Professional Software, Inc., 51 Fremont Street, Needham, MA 02194.

Circle Reader Service Number 220.

World War II Air War

Strategic Studies Group (SSG), creators of *Reach for the Stars* and *Carriers at War* strategy games, has released *Europe Ablaze: The Air War Over England and Germany 1939-1945*. This historical simulation contains three major scenarios, selected from the various phases of the air war, and also a game design kit that lets you create your own scenarios. Major bombing missions are planned twice each day, and players are required to select targets, plot course and speed, determine H-hour, and allocate squadrons. Fighter aircraft patrol and intercept in response to ground and radar sightings.

Europe Ablaze is available for Apple II-series computers (with 64K RAM) and for the Commodore 64, at a suggested retail price of $50.

Strategic Studies Group, 1747 Orleans Court, Walnut Creek, CA 94598.

Circle Reader Service Number 221.

Apple II Spreadsheet

Mouse Calc, a mouse-controlled spreadsheet for the Apple IIc and 128K Apple IIe, has been announced by International Solutions, Inc. The program includes integrated graphics, mouse-operated editing and selecting techniques, pull-down menus, and color display. *Mouse Calc* is the first in a series of application programs from International Solutions.

Users can perform 24 of the most commonly used arithmetic, logical, search, and other spreadsheet functions with *Mouse Calc*. The program provides rounding and logical functions such as AND/OR and TRUE/FALSE. *Mouse Calc* can merge two or more files, and it can read files created with *VisiCalc*, *AppleWorks*, and other programs using the DIF format.

The program requires a mouse controller, such as the AppleMouse II, and a second disk drive is recommended. Suggested retail is $149.95, and includes a 90-day warranty.

International Solutions, Inc., 910 West Maude Avenue, Sunnyvale, CA 94086.

Circle Reader Service Number 222.

Educational, Entertainment Programs

Among the software titles recently introduced by CBS Software are several educational and entertainment programs. Included are *The Body in Focus*, a self-paced color-graphics human anatomy program for the Apple II+, IIe, IIc, Commodore 64, and IBM PC ($39.95 each); *Success with Math*, a series of math tutorials for ages 6 through 18 for the Apple II+, IIe, IIc, Commodore 64, and IBM PC/PCjr ($24.95 each); *Success with Algebra*, a similar series covering algebra for grades 7–12 for the Apple II series, Commodore 64, and IBM ($34.95); and *Quink*, a game of pattern recognition and knowledge for ages ten and older, for the Apple II series, 64, and IBM PC/PCjr ($34.95).

CBS Software, One Fawcett Place, Greenwich, CT 06836.

Circle Reader Service Number 223.

64 Bulletin Board

Bozart Co. has introduced two telecommunications packages for the Commodore 64: *Bozboard*, a full-featured bulletin board program, and *Bozterm*, an all-purpose terminal program. *Bozboard* is set to run with one or two 1541 disk drives or the MSD SD-2 dual drive. It is compatible with the Commodore 1650 Automodem and also with the Westridge 6420, TeleLearning, Mitey Mo, and HES II modems. The system requires a printer.

With *Bozboard* ($40), you have a choice of eight subboards, public messages, electronic mail, uploading, downloading, and a magazine feature which allows the system operator (sysop) to publish a color/graphics electronic magazine on the BBS. The program transfers files using the standard XMODEM protocol and its own Bozart protocol. The Bozart protocol is capable of transferring high-resolution graphics and allows the bulletin board user to view the graphics screen as it is downloaded.

Bozterm ($20) offers the user the option of automatic dialing or manual dialing. Seven of the function keys can be defined to automatically transmit any 80-character message, read the disk directory, upload buffer contents, capture incoming data to the buffer, print it to the screen or a printer, or save it to disk as an edited or unedited file.

Bozart Co., 7818 Summerfield Road, Summerfield, NC 27358.

Circle Reader Service Number 224.

Atari Interface

Integrated Computer Equipment Company (ICECO) has introduced the ICEPIC (ICE's Parallel Interface Converter), a printer interface for Atari computers which also includes graphics software drivers.

The ICEPIC converts parallel-interface (Centronics-compatible) printers to a joystick interface (joystick port 2 or 4) which can be used by Atari 400/800 and XL computers with no hardware modifications. The ICEPIC requires no 850 Interface Module, no cable, and no external power supply. The software supports any printer in text mode and provides graphic functions for Epson or Okidata 92/93 graphics compatible printers. The ICEPIC works with most Atari programs, such as most BASIC programs, *AtariWriter*, *Letter Perfect*, *B/GRAPH*, Koala *Micro Illustrator*, and *AtariArtist*. Several utility programs are included with the software, including a diagnostic checkout program, a warm reboot program, and a *MicroPainter* file display program.

Suggested retail is $49.95 for the interface, software, and manual. There is a 30-day money back guarantee, a 90-day replacement warranty, and a lifetime $19 repair/replacement policy.

Integrated Computer Equipment Company, 8507 Natural Bridge Road, St. Louis, MO 63121.

Circle Reader Service Number 225.

64, Apple Karate

Data East has converted its arcade action game, *Karate Champ*, to a new computer version for the Commodore 64 and Apple II series. The $29.95 game features two-player and player-versus-computer modes as you guide your karate fighter through successive matches. Using the joystick, you can make your fighter lunge, kick, spin, somersault, reverse-punch, and block.

Data East USA, Inc., 470 Gianni Street, Santa Clara, CA 95054.

Circle Reader Service Number 226.

Pascal For 64 & 128

A complete Pascal development system for the Commodore 64 and 128 has been released by Abacus Software. *Super Pascal* includes an extensive compiler, a source file editor, an integrated assembler, and a comprehensive utility package for file and disk management.

Also included are a handbook with more than 200 pages and a systems disk. Suggested retail price is $59.95.

Abacus Software, 2201 Kalamazoo SE, P.O. Box 7211, Grand Rapids, MI 49510.

Circle Reader Service Number 227.

New From Epson

Epson has developed several new printers for home users. Among these are the DX-10 ($399), a daisywheel printer which prints at ten characters per second (cps); and the DX-20 ($499), a daisywheel with a 1K byte print buffer, 20 characters-per-second (cps) print speed, and a Diablo All Purpose Interface (RS-232C, IEEE-488, and parallel).

Also new from Epson is the Spectrum LX-90, a dot-matrix printer with draft and near-letter-quality (NLQ) modes. It comes with a printer interface cartridge that makes it ready for use with the IBM PC, PCjr, or Apple IIc. Draft copy is printed at 100 cps; NLQ at 16 cps. The Spectrum LX-90 retails for $389.

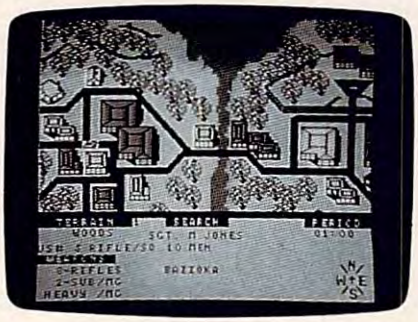
Comrex, a division of Epson, has released the CR-128 intelligent printer buffer. Features include 128K buffer memory and built-in serial-to-serial, serial-to-parallel, parallel-to-parallel, and parallel-to-serial interfaces. Suggested retail is $299. Another new Comrex product is the CR-700 series of bidirectional A-B switch boxes, which simplify connections to the computer and eliminate the need to swap cables when changing peripherals. The switch boxes retail for $39.95.

Epson America, 2780 Lomita Blvd., Torrance, CA 90505.

Circle Reader Service Number 228.

World War II Combat Game

Under Fire, from Avalon Hill, combines the depth of a war game simulation with the colorful graphics of an arcade game. Authentic armies, weapons, and maps from World War II add to the game's realism. Different maps and scenarios are included on disk; players can also create their own.



A sample screen from Avalon Hill's Under Fire *strategy game.*

Under Fire is available for the Apple II series. A joystick is optional for the Apple IIc and IIe, but required for the II+. Suggested retail price is $59.95.

Avalon Hill Game Company, 4517 Hartford Rd., Baltimore, MD 21214.

Circle Reader Service Number 229.

TI Disk Organizer

TI programs can be organized on a single disk with *Disk Data Base* from Asgard Software. The program lets you sort and print a catalog by either disk name or filename, to print it out unsorted, or to selectively print out all entries that contain a certain string. The catalog can also be broken up into blocks of 250 entries for easy management.

Data files can be converted from the Master Disk File to the *Disk Data Base* format. Also featured are numerous prompts and an online dictionary of terms. *Disk Data Base* requires Extended BASIC, a 32K memory expansion unit, and a disk drive and controller. A printer and second disk drive are recommended. Price: $15.

Asgard Software, P.O. Box 10306, Rockville, MD 20850.

Circle Reader Service Number 230.

Bulletin Board Directory

A new directory of computer bulletin boards, called *Plumbline*, is now available from the publishers of *Plumb*, a newsletter about personal telecommunications. The directory lists over 1,000 bulletin boards available to the public. Each entry includes a brief description of the bulletin board, the type of computer it runs on, and its primary area of interest.

Plumbline is included with a subscription to *Plumb*, $26.50; or can be purchased separately for $8.

Plumb, P.O. Box 300, Harrods Creek, KY 40027

Circle Reader Service Number 231.

Pascal Tutorial For Apple

Wiley Software's new *Visible Pascal* uses graphics, word processing, and music to teach the Pascal programming language on Apple II computers. Programs are displayed while they're being created, at a speed controlled by the programmer. The system has more than 80 error messages for pointing out mistakes. Users can create "productions," with animated characters and a soundtrack. The package also includes 56 sample programs that are ready to run.

No prior computing knowledge is needed. *Visible Pascal* runs on Apple II-series computers with at least 64K RAM. A joystick is recommended.

Wiley Professional Software, 605 Third Ave., New York, NY 10158.

Circle Reader Service Number 232.

Boolean Games

Sunburst has introduced *High Wire Logic*, a game for teaching Boolean logic to youngsters in grades 5 through 12. Two sets of colored shapes appear on the screen: one on a high wire and another set that falls to the net below. Using the logical functions AND, OR, AND-AND, OR-OR, and EXCLUSIVE OR, students earn points by writing rules to fit the shapes on the high wire but not the shapes in the net.

High Wire Logic is available for Apple II computers with at least 48K RAM; retail price is $59.

Sunburst Communications, Inc., 39 Washington Ave., Pleasantville, NY 10570.

Circle Reader Service Number 233.

Life/Time Manager

A new program from Psychometric

Software provides assistance in identifying goals and organizing time. Developed by a psychiatrist, *Life/Time Manager* is based on psychological and time management principles. It consists of three sections: Goals, Activities, and Schedules. Included are a prioritized daily To Do List and a weekly schedule analysis.

The program runs on the IBM PC, PCjr, or AT, with at least 128K RAM; or on the Apple II+, IIc, or IIe. Suggested price is $49.95.

Psychometric Software, Inc., 2050 S. Patrick Dr., Indian Harbour Beach, FL 32937.

Circle Reader Service Number 234.

Nutrition And The Apple

The Center for Science in the Public Interest, a nonprofit consumer group, has developed *Nutrition Express.* This game teaches the basic concepts of nutrition and diet through a series of questions and clues. Action takes place in the land of FodaFoda, where the student answers questions correctly in order to earn currency for the grocery store and to invite friends from FodaFoda back home. The game is geared toward youngsters aged nine and up.

Nutrition Express comes with a user's guide, teaching suggestions, and a "Nutrition Scoreboard" wall chart. For the Apple II series; price is $39.95.

Center for Science in the Public Interest, 1501 Sixteenth St., NW, Washington, DC 20036.

Circle Reader Service Number 235.

Titling Videos

A new program from Videoware can put titles, custom messages, colored screens, and leaders onto videotapes. *Video Title Editor* offers a menu of more than 20 different displays, including some for weddings, birthdays, and video mail. Also included are displays for Presented By, Starring, and Credits.

The program requires a videocassette recorder and either an Apple II, Atari, Commodore 64, VIC-20, or IBM PC/PCjr. Price is $29.95.

Videoware, 19777 W. 12 Mile Rd., Suite 180, Southfield, MI 48076.

Circle Reader Service Number 236.

New IBM Telecommunications Utility

Mastercom, a new release from The Software Store, is a full-featured smart terminal and file transfer utility for the IBM PC and PCjr. It turns the computer into a terminal on a time-sharing system, captures data onto a disk and/or printer from almost any computer, and can send files to almost any type of computer. *Mastercom* supports most communication protocols including Christensen XMODEM, xon/xoff, line at a time, and no protocol. Other features of *Mastercom* include auto dial, auto answer, batch file transfer, and host mode unattended operation.

For the IBM PC, PCjr, and most compatibles. Suggested retail price is $49.

The Software Store, 706 Chippewa Square, Marquette, MI 49855.

Circle Reader Service Number 237.

Multi-Color Printing Package

A black-and-white printer can now make up to 80 full-color prints using an Apple computer and *Prince,* a new program from Baudville. The program's library of fonts can be used to make color T-shirt transfers, banners, letterheads, and labels. *Prince* can also capture any standard or double hi-res picture for editing and printing.

Four color ribbons are included for the following printers: Imagewriter, DMP, C. Itoh 8510/Prowriter, 8510 SC, NEC 8023, and Epson MX-80, RX-80, and FX-80. *Prince* sells for $69.95.

Baudville, 1001 Medical Park Dr. SE, Grand Rapids, MI 49506.

Circle Reader Service Number 238.

Electronic Trivia

Mentor Learning Systems has introduced *Ultimate Trivia,* a game featuring 4,000 facts and 200 color graphics. The facts are divided into nine categories: Music, Cinema, Geography, Sports, General Information, People, Art, History, and Television. The graphics are revealed piece by piece as each category is answered correctly. *Ultimate Trivia* can be played individually or in teams.

The program sells for $49.95 and runs on all Apple computers (with at least 64K RAM) and the IBM PC/PCjr.

Mentory Learning Systems, Inc., 1825 De La Cruz Blvd., Santa Clara, CA 95050.

Circle Reader Service Number 239.

New Infocom Adventures

Infocom has added two new products to its interactive fiction line. In *A Mind Forever Voyaging,* you play the role of a computer that has been raised as a human being up to the age of 20. You must enter a simulation of the future to see whether a plan proposed by current government and industry leaders will be beneficial for the country. Due to Infocom's new development system and an expanded 1,700-word vocabulary, the emphasis in this game is less on solving puzzles and more on revealing the story's details. (Requires at least 128K RAM; Apple II series, IBM PC/PCjr/XT/AT, Atari ST, Amiga, and Macintosh; $44.95.)

Spellbreaker completes Infocom's *Enchanter* trilogy of adventures. When a world based on sorcery finds its magic failing, you, as the leader of the Circle of Enchanters, must find and destroy the cause of this failure. (Apple II series, Amiga, IBM PC/PCjr/XT/AT, Macintosh, and MS-DOS compatibles, $49.95; Atari 400/800, XL/XE, ST, Commodore 64/128, $44.95.)

Infocom, Inc., 125 Cambridge Park Drive, Cambridge, MA 02140.

Circle Reader Service Number 240.

Parallel Printer Converters

Two new serial-to-parallel printer converters have been released by Practical Peripherals. The *Switchport IIc* was designed especially for the Apple IIc and allows the computer to be interfaced with a parallel printer. The *Switchport 232* transforms serial data into parallel, allowing any RS-232 computer to be interfaced with a Centronics parallel printer.

Both units come with a five-year limited warranty and retail for $109.

Practical Peripherals, 31245 LaBaya Drive, Westlake Village, CA 91362.

Circle Reader Service Number 241.

Productivity, Education, Entertainment Software

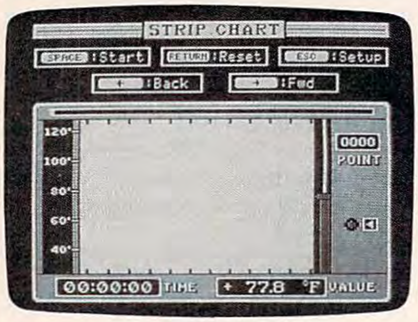
Brøderbund Software has introduced a hardware/software combination that turns your home computer into a science lab. The *Science Toolkit Master Module* includes a temperature-sensing probe, a light-sensing probe, and a special interface that connects them to an Apple II via the joystick port. Using the software's thermometer, light meter, timer, and strip chart, you can perform a wide variety of scientific experiments. (At least 64K RAM required; Apple IIe/IIc, II+ with joystick port adapter; $59.95.)

Two new packages have been added to the *Bank Street* series of productivity software. *Bank Street Mailer* is a combination letter-writing/mailing list program. *Bank Street Filer* is a database manager/report-generating program. There are two versions of each program: a 64K version for the Apple II+ and IIe offers a 40-column screen display, and a 128K version for the Apple IIc and 128K IIe offers a 40- or 80-column display and includes an onscreen calculator. They are compatible with the *Bank Street Writer* word processor. All retail for $69.95 each.

Captain Goodnight and the Islands of Fear is an arcade game that plays like an adventure movie. In your role as

Captain Goodnight, you must pilot helicopters, airplanes, tanks, trucks, and a submarine in your attempt to save the world from destruction. (Apple II series with at least 48K RAM; $34.95.)

Brøderbund Software, 17 Paul Drive, San Rafael, CA 94903.



A strip chart from Brøderbund's Science Toolkit.

Circle Reader Service Number 242.

Computerized Diet Plan From Bantam

The Complete Scarsdale Medical Diet, based on the bestselling book by the same name, is now available in a software package from Bantam Electronic Publishing. Based on sound nutritional principles, the program offers healthy, controlled weight loss. *Diet* features a meal-planning calendar, shopping list, expandable food directory, meal planner and analysis, and comparison charts.

Available for the Apple II series and IBM PC/PCjr, the program retails for $39.95.

Bantam Electronic Publishing, 666-5th Ave., New York, NY 10103.

Circle Reader Service Number 243.

Koalapad+ For Apple II

Koala Technologies has announced an enhanced version of the Koalapad, called *Koalapad+*, for the Apple IIc and IIe. The new version offers enhanced product styling, a gridded tablet surface, and additional graphics software. The software, *Graphics Exhibitor,* lets users edit images they have created.

Suggested retail price for the *Koalapad+* is $125.

Koala Technologies, 2065 Junction Ave., San Jose, CA 95131.

Circle Reader Service Number 244.

Productivity Software For Commodore 64

Datamost has announced the *KWIK* line of home productivity software for the Commodore 64. Each package includes *KWIK-LOAD!* (a Datamost fast-loading program) and retails for $19.95.

The series includes *KWIK-WRITE!,* a word processor; *KWIK-SPELL!,* a spelling checker; *KWIK-FILE!,* a database manager; *KWIK-CALC!,* a spreadsheet program; *KWIK-PAINT!,* a graphics editor; *KWIK-CHECK!,* a checkbook balancing and maintenance program; *KWIK-PAD!,* a desk secretary program; and *KWIK-PHONE,* a communications program.

Datamost, 19821 Nordhoff Street, Northridge, CA 91324.

Circle Reader Service Number 245.

Foreign Language Vocabulary Programs

Gessler Educational Software has produced three foreign language versions of its bestselling vocabulary program *Word Attack!. Bataille De Mots* (French), *Batalla De Palabras* (Spanish), and *Wortgefecht* (German) are available for the Apple II series, IBM PC/PCjr, and Commodore 64 for $49.95.

Word Attack!, as well as its foreign language versions, teaches vocabulary words and grammar with word displays, quizzes, sentence completion, and an arcade game.

Gessler Educational Software, 900 Broadway, New York, NY 10003.

Circle Reader Service Number 246.

Inexpensive Accounting Software

DAC-Easy Accounting, from DAC Software, is a seven-in-one accounting package offered at a special introductory price of $49.95. Its seven individual modules—general ledger, accounts receivable, accounts payable, billing, purchase order, inventory, and forecasting—are integrated, allowing automatic posting between modules.

The system also has spreadsheet capability, letting the user experiment with "what-if" scenarios without entering actual data. It is compatible with the IBM PC and PCjr.

DAC Software, Inc., 5580 Peterson, Suite 130, Dallas, TX 75240.

Circle Reader Service Number 247.

MIDI Editor

RolandCorp has released *MUSE (MIDI Users Sequencer/Editor)* for the Apple II series and Commodore 64. The program features eight independent tracks for recording and overdubbing sequences, track merging capability, track muting, looping by song or track length, and selectable time signatures. The editing functions can be used to insert, delete, move, copy, and rearrange measures of any track so that a composition can be changed after it has been recorded.

MUSE is compatible with any MIDI instrument and can be synchronized with drum machines, other sequencers, and multitrack tape decks. An interface is required. Suggested retail price is $150 for each version.

RolandCorp US, 7200 Dominion Circle, Los Angeles, CA 90040.

Circle Reader Service Number 248.

New PCjr Drive

A second disk drive can now be added to the IBM PCjr without adding extra circuitry or another power supply. The Junior Drive II System, from PC Enterprises, includes a 360K double-sided double-density 5-1/4 inch floppy disk drive with power supply, an adapter module, a software patch, a two-drive signal cable, and an instruction manual. The system is compatible with existing external modems, parallel printer ports, and memory expansions.

The Junior Drive II System lists for $395. For those who wish to connect their own IBM-compatible drive, the adapter module and software patch are available separately.

PC Enterprises, P.O. Box 292, Belmar, NJ 07719.

Circle Reader Service Number 249.

The Smoking Decision

A new program from Sunburst was created to alert students to the dangers of cigarette smoking. It begins by presenting facts about health risks related to smoking, and then explores issues such as peer pressure. Throughout the program, students are confronted with a series of incremental decisions, leading to a final decision whether to smoke.

The Smoking Decision is suitable for youngsters in grades 6 through 12. It runs on any Apple II computer with at least 48K memory. The $59 retail price includes a backup disk and teacher's guide.

Sunburst Communications, Inc. 39 Washington Ave., Pleasantville, NY 10570.

Circle Reader Service Number 250.

Arcade And Adventure Games For Commodore 64

Artworx has released two new games for the Commodore 64 and 128. *Falcon Patrol II* puts the user in the pilot's seat of a Falcon fighter, fully equipped with air-to-ground and air-to-surface missiles. The object of the game is to ward off the enemy's helicopter attack squadrons. Its 16 levels of play are enhanced by 3-D graphics and sound effects.

In *Sorcery,* you are the last of the great sorcerers, given new strength and powerful spells. You must use them to regain your conquered homeland and restore its previous quality of life. *Sorcery* resembles an arcade game, but plays much like an adventure game.

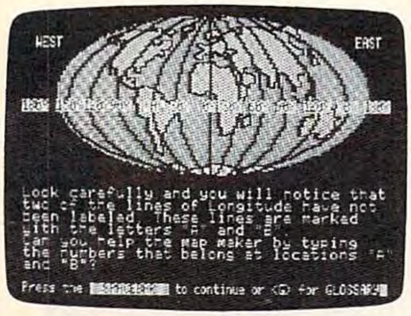
Both games retail for $19.95.

Artworx Software Company, Inc., 150 N. Main St., Fairport, NY 14450.

Circle Reader Service Number 251.

Educational Software For The Classroom

Focus Media, Inc. publishes an extensive line of classroom programs for a variety of computers. In *Za-Zoom, The Geography Genie,* students take the role of explorer as they try to determine where they are by examining the culture around them. The two programs in this package, *Travels with Za-Zoom: The World* and *Travels with Za-Zoom: The U.S.* retail for $129; either program can be purchased separately for $79. (Apple II series, Commodore 64, IBM PC/PCjr.)



```
Look carefully and you will notice that
two of the lines of Longitude have not
been labeled. These lines are marked
with the letters "A" and "B".
Can you help the map maker by typing
the numbers that belong at locations "A"
and "B"?
Press the [illegible] to continue or <G> for GLOSSARY
```

Students learn about such concepts as latitude and longitude with The Language of Maps.

Students can go back in time with *The Time Tunnel: America Series Package.* During each journey, students must use clues to gather facts and guess the identities of historical figures. The package contains six programs: *Early America* (2), *A Nation Emerges* (2), and *The Presidents* (2). Suggested retail price for the complete package is $179; each series can be purchased individually for $79. (Apple II series, Commodore 64, IBM PC/PCjr.)

The Language of Maps is a series of six programs that helps students learn about maps and map terminology. Topics covered include oceans and continents; land areas and water bodies; highlands and lowlands; and finding places on maps. The Instant Computerized Glossary explains unfamiliar terms. *The Surface of the Earth* and *Location and Distance* retail as a package for $159; individually, each costs $79. (Apple II series.)

A Teacher's Lesson Planner and free backup disks are included with all packages.

Focus Media, Inc., 839 Stewart Ave., Garden City, NY 11530.

Circle Reader Service Number 252.

presenting . . . CAPTURE™

A NEW WAY TO UNLOCK THE POWER OF YOUR C64 OR C128*

- **CAPTURE** IS A CARTRIDGE THAT PLUGS INTO YOUR COMPUTER'S EXPANSION PORT.
- **CAPTURE** DOES NOTHING — UNTIL YOU PRESS ITS **CAPTURE** BUTTON. UNTIL THEN, A RUNNING PROGRAM CANNOT DETECT ITS PRESENCE.
- **CAPTURE** THEN TAKES CONTROL — NO IFS, ANDS OR BUTS — AND PRESENTS A MENU.
- **CAPTURE** WILL NEATLY SAVE EVERYTHING IN YOUR COMPUTER TO YOUR 1541 OR COMPATIBLE DISK DRIVE — ALL 64K OF RAM, CPU, VIC AND SID CHIP REGISTERS — EVERYTHING. IN EASY TO VIEW CHUNKS.
- **CAPTURE** WILL, IF YOU WANT, PRE-CONFIGURE YOUR COMPUTER'S RAM SO THAT ONLY MEMORY ALTERED BY YOUR PROGRAM NEED BE SAVED.
- **CAPTURE** WILL WRITE A BOOT ON YOUR DISK SO YOU CAN RELOAD AND BEGIN EXECUTION RIGHT WHERE YOU LEFT OFF.
- **CAPTURE** DOES ALL THIS AT A COST OF JUST **$39.95**

BUT HERE'S THE BEST PART

- **CAPTURE** WILL MAKE AN AUTO-START CARTRIDGE OF YOUR PROGRAM. IT'S EASY! JUST FOLLOW THE DIRECTIONS ON THE SCREEN. NOW PLUG IN YOUR CARTRIDGE AND TURN ON YOUR COMPUTER. IN LESS THAN TWO SECONDS YOUR PROGRAM BEGINS AGAIN AT PRECISELY THE POINT WHERE YOU **CAPTURE**'D IT. MAGIC!

  BESIDES **CAPTURE,** YOU NEED A *promenade* C1 AND A SUPPLY OF CPR3 CARTRIDGE KITS.

ORDERING INFORMATION

• **CAPTURE** CARTRIDGE — COMPLETE WITH INSTRUCTIONS	$ 39.95
• *promenade* C1 — EPROM PROGRAMMER WITH DISK SOFTWARE	99.50
• CPR3 CARTRIDGE KIT — PC BOARD, CASE AND 3 EPROMS	29.95
• DR—EPROM ERASER, TWO AT A TIME, 3 TO 10 MINUTES	34.95
• STARTER SET — **CAPTURE,** *promenade* C1 AND 1 CPR3	149.95
• DELUXE SET — **CAPTURE,** *promenade* C1, DR AND 2 CPR's	199.95

SHIPPING AND HANDLING — USA: UPS SURFACE $3.00
BLUE LABEL $5.00 NEXT DAY AIR $13.00
CANADA: AIR MAIL $7.00 OTHER FOREIGN AIR $12.00

CALIFORNIA RESIDENTS ADD APPLICABLE SALES TAX
COD ORDERS, USA ONLY, ADD $3.00

C64 AND C128 TM COMMODORE ELECTRONICS, LTD. *WHEN OPERATING IN 64 MODE

TO ORDER: TOLL FREE	800-421-7731	TECHNICAL SUPPORT AND	408-287-0259
FROM CALIFORNIA	800-421-7748	FROM OUTSIDE THE US:	408-287-0264

JASON-RANHEIM
580 PARROT STREET
SAN JOSE, CA USA 95112

VISA MasterCard AMERICAN EXPRESS

DISK WORLD! is proud to introduce the lowest-priced, LIFETIME-WARRANTY diskettes ever!

And they're BRAND NAME PRODUCT to boot!

5.25″SSDD→.69 ea. 5.25″ DSDD→.79 ea.

5.25″ DSDD-HD→$2.25 ea.

3.50″ SSDD→$2.09 ea. 3.50″ DSDD→$2.55 ea.

Based on multiples of 100 each.
Boxed in 10's with heavy-duty cardboard sleeves, user ID labels, reinforced hubs (where appropriate) and write-protect tabs.

Introducing Wabash Pinnacle Series Diskettes.

Two years ago, if you'd told me I'd be writing this ad, I would have laughed.

At that time, Wabash diskettes were synonymous with "s—t".

Just saying that quality control was poor would be charitable.

So much was wrong that DISK WORLD wouldn't sell them.

That was yesterday.

Kearney-National Inc., a $202-million division of a much larger company, came into Wabash.

Out went the old management, the old methods, the old production techniques...and in went a lot of new people, ideas, production lines and some really imaginative thinking.

The end result.

Today, I'm proud to offer you the Wabash Pinnacle Series of diskettes at the prices shown.

This isn't evolution in diskette manufacturing: it's revolution.

Here's what you get.

Wabash Pinnacle diskettes are

...certified 100% Error Free

...are coverd by a LIFETIME WARRANTY

...meet or exceed all industry specifications (by quite some distance)

...and are simply the best value in diskettes available today.

The torture test.

Considering Wabash's earlier dubious reputation, I wasn't exactly a true believer when their Director of Marketing came into my office with samples.

So I took a box at random, selected a disk, bent the thing every which way and slipped it into my IBM-PC.

It formatted. It booted. It stored and retrieved data.

That wasn't enough.

I gave samples of the diskettes to Curt Rostenbach and, in turn, to Tom Streit, both hackers of long experience and members of the Waukegan (Illinois) Apple Users Group.

Tom really went at it.

He took a quartz-halogen lamp, aimed it at the diskette until it started to smoke (and melt)...and then formatted, booted the diskette and stored and retrieved data!

The same terribly (and intentionally) mutilated diskette ran on an ITT, Corona and IBM.

Curt was nicer.

He simply bent the diskette every which way...and it still formatted, booted and ran on his Apple.

The best buy I've ever seen.

DISK WORLD!, Inc. sells more flexible magnetic media by mail-order than anyone else in the world.

I, as President of the corporation, won't tolerate a product with a failure rate of more than 1/1000th of 1 percent.

I also don't like companies who try to milk a "quality" or "premium" image for a higher price like Dysan and Verbatim did...until they failed.

As President of DISK WORLD!, Inc., my motto is simple: "the best diskette for the least amount of money."

Wabash is it.

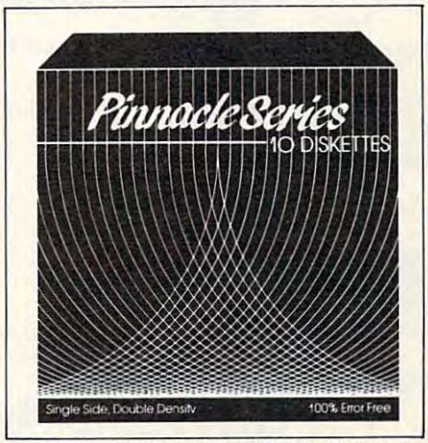
Right now, there is no better value than the Wabash Pinnacle Series of diskettes.

Granted, you have to buy a hundred at a time, but so what? Split the order with friends, relatives, co-workers or even your worst enemies.

The key thing is to get the most diskette for the money.

And this is it.

(Incidentally, as a corporation, we put our money where our mouth is. Our first order for Wabash Pinnacle Diskettes was 1.5-million units.)



That's an awful lot of faith and confidence.

But, then again, I have the diskette that Tom Streit literally melted...and kept on running.

The truth about $1.00 or less diskettes.

More and more ads are popping up offering diskettes for $1.00 or less.

By the same token, more and more people who were selling used cars a few months ago are now selling diskettes by mail.

We did a little survey of current ads for diskettes advertised for a dollar or less and did some analysis of the market and here's what we found as it applies to 5.25″ DSDD diskettes "supposedly" selling for a dollar or less.

VENDOR:	ADVERTISED LOW PRICE:	ACTUAL PRICE PER 100:	ACTUAL MFGR.:
Unitech	.89 ea.	.92 ea.	Unspecified.
Datatech	.99 ea.	.99 ea.	Unspecified.
Computer Club	.95 ea.	.98 ea.	Unspecified.
	.99 ea.	1.02 ea.	Unspecified.
Communications & Electronics	.49 ea.	.80 ea.	Unspecified.
Precision Data	.89 ea.	.93 ea.	Unspecified.
Diskette Connec.	.93 ea.	.93 ea.	Unspecified.
Comp Soft Serv.	.77 ea.	.77 ea. + shpg.	Unspecified.
Computer/Computer	.99 ea.	.99 ea.	Unspecified.
DISK WORLD	.89 ea.	.92 ea.	Wabash Datatech

The real truth about $1.00 or less diskettes.

It costs all diskette manufacturers about the same to produce a diskette. Some may charge more because they want to project a "premium quality" image, ala the late, lamented Dysan who bought their basic media from 3M.

Some charge less because they sell a sub-standard product...and we're not foolish enough to name names here.

But here's the truth about the $1.00 or less diskette market.

It falls into four categories:

1. The DISK WORLD's of the universe who simply are so big that they can buy first quality product in massive quantities and choose to pass on the savings to you. (Precision Data and Diskette Connection on **BRAND NAME** products also fall into this category.)

2. The people who buy "cosmos"...stuff from major manufacturers that usually hits quality control standards, but is cosmetically blemished and thus can't be packaged and sold under the manufacturer's own name.

3. "Duplicator Quality". Uncertified media, usually below manufacturer's own standards and frequently below ANSI and IBM standards. Sold on an "as-is" basis with the understanding that the manufacturer's name will never be divulged. Usually about a 20% reject rate...as compared to DISK WORLD's standard of less than 1/1000th of 1% reject/return rate. Next to garbage, this is the source of most diskettes advertised at a dollar or less.

They may work...and then again they may not. (Frankly, the odds at the Blackjack table in Las Vegas are more in your favor.)

4. Garbage. Stuff that shouldn't be sold at all. But some manufacturers are hurting for cash, so they sell it anyway. (After all, they want to meet their payroll. Look what happens when you don't: you become a Dysan or Verbatim. Lots of history, but no money.) More and more garbage is being dumped into the market as manufacturers become pressed for cash and are motivated into selling anything and everything they can manufacture. (Read the article in FORBES about Verbatim and its "Bonus" brand.)

Finally, the Taiwanese counterfeiters are moving into the act. Perfect duplicates of the packaging of major manufacturers with one exception: the quality isn't there.

The Critical Factor.

Only DISK WORLD!, Inc. offers fully brand-identified, LIFETIME-WARRANTY product for less than a dollar.

Every one else offering 5.25″ product for less than a buck doesn't tell you who makes it.

We do.

And that ought to tell you a lot right there.

Ordering & Shipping Instructions

SHIPPING: Wabash Pinnacle Diskettes are sold in multiples of 100 only. Shipping charges are $3.00 per 100, regardless of type or size.

PAYMENT: VISA, MASTERCARD and PREPAID orders accepted. Corporations rated 3A2 or better and government and quasi-government open accounts are accepted on a NET 15 basis.

C.O.D. orders are subject to a $5.00 special handling charge. (Sorry for the increase, but too many people have been refusing C.O.D. orders or using bad checks. It's a classic example of a few "bad eggs" making life more expensive for everyone else.)

APO, FPO, AK, HI & PR ORDERS: Include shipping as shown **and** an additional 5% of the total amount of the order to cover PAL and insurance.

No other non-continental U.S. orders are accepted.

TAXES: Illinois residents only, add 7%.

MINIMUM ORDER: $35.00

All orders subject to acceptance.
Not responsible for typographical errors.

ORDERS ONLY:
1-800-621-6827
(In Illinois: 1-312-256-7140)

INQUIRIES & INFORMATION
1-312-256-7140

FOR FASTEST SERVICE, USE MCI MAIL:
Just address **"DISKORDER"**
(24-hour shipping on any item in stock if you order via MCI MAIL.)

DISK WORLD!, Inc.

629 Green Bay Road
Wilmette, Illinois 60091

MONITORS DISK DRIVES

$149

FULL
COLOR
COMMODORE 1702

$159

1541
COMMODORE

VIDEO

COLOR
VIDEO CAMERA
& RECORDER

$698

$198 PORTABLE
VIDEO RECORDER

1-800-328-8322 EXT 581

T & D ELECTRONICS
10517 UPTON CIRCLE
BLOOMINGTON. MN 55431

PROFESSIONAL
HANDICAPPING SYSTEMS

• PRESENTED BY PROFESSOR JONES •

MARKET YOUR SOFTWARE INEXPENSIVELY

Professor Jones will be reviewing 3rd party software for our newest catalogue

If you currently market or plan to market PROFESSIONAL software dedicated to handicapping, stock market analysis, sports, accounting, or business application, Prof. Jones is now reviewing software for the Spring '86 catalogue. ALL software will be reviewed and developers will either receive a letter of review and/or a multilevel contract proposal.

(Sorry, we will review software but it cannot be returned)

Call (208) 342-6939 or Mail to:

Professor Jones
1940 West State Street
Boise, Idaho 83702

$50 for your old interface

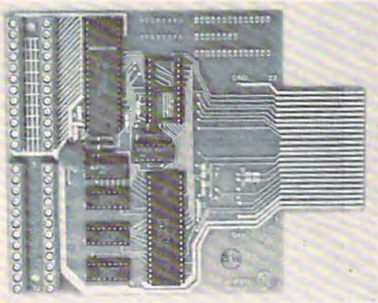
NEW! Universal Parallel Graphics Interface

- Built-in self-test with status report
- Optional RAM printer buffer
- Provides virtually total emulation of Commodore printers for compatability with popular software
- ASCII conversion, total test, Emulate & tranparent mode
- Fully intelligent interface that plugs into standard Commodore printer socket
- Exclusive graphic key-match function
- Switch selectable Commodore graphics mode for Epson, StarMicronics, C. Itoh, Prowriter, Okidata, Seikosha, Banana, BMC, Panasonic, Mannesman Talley & others

Micrografix MW-350 .... **$129**
.... **$79.00 with any trade in**

Universal Input/Output Board for C-64 & C-128

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



CBM-64 used MW-611 .... **$225.00**

Dealer Inquiries invited

VISA MasterCard

Micro World Computers, Inc. (303) 987-9531
3333 W. Wadsworth Blvd. #C105
Lakewood, CO 80227

COMPUTE! Subscriber Services

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

COMPUTE! Magazine
P.O. Box 10955
Des Moines, IA 50950

or call the Toll Free number listed below.

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

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

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

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

COMPUTE!
800-334-0868
In NC 919-275-9809

What the world really needs is a 69 cent Double Sided, Double Density Diskette with a LIFETIME WARRANTY!

And DISK WORLD! has it.

Introducing Super Star Diskettes: the high quality diskette with the lowest price and the best LIFETIME WARRANTY!

In the course of selling more than a million diskettes every month, we've learned something: higher prices don't necessarily mean higher quality.

In fact, we've found that a good diskette manufacturer simply manufactures a good diskette...no matter what they charge for it. (By way of example, consider that none of the brands that we carry has a return rate of greater than 1/1,000th of 1 percent!)

In other words, when people buy a more expensive diskette, they aren't necessarily buying higher quality.

The extra money might be going toward flashier advertising, snazzier packaging or simply higher profits.

But the extra money in a higher price isn't buying better quality.

All of the good manufacturers put out a good diskette. Period.

How to cut diskette prices ...without cutting quality.

Now this discovery posed a dilemma: how to cut the price of diskettes without lowering the quality.

There are about 85 companies claiming to be "diskette" manufacturers.

Trouble is, most of them aren't manufacturers.

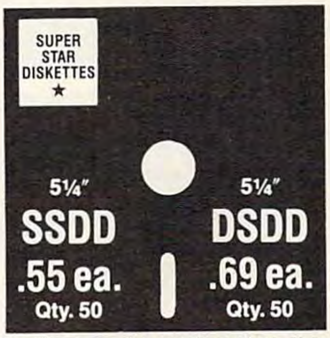
Rather they are fabricators or marketers, taking other company's components, possibly doing one or more steps of the processing themselves and pasting their labels on the finished product.

The new Eastman Kodak diskettes, for example, are one of these. So are IBM 5¼" diskettes. Same for DYSAN, Polaroid and many, many other familiar diskette brand names. Each of these diskettes is manufactured in whole or in part by another company!

So, we decided to act just like the big guys. That's how we would cut diskette prices...without lowering the quality.

We would go out and find smaller companies to manufacture a diskette to our specifications...specifications which are higher than most...and simply create our own "name brand" diskette.

Name brand diskettes that offered high quality at low prices.



Super Star diskettes are sold in multiples of 50 only. Diskettes are shipped with white Tyvec sleeves, reinforced hubs, user ID labels and write-protect tabs.

Boy, did we get lucky. Our Super Star Diskettes are the same ones you've been using for years...without knowing it.

In our search for the low priced, high quality diskette of our dreams, we found something even more interesting.

We found that there are several manufacturers who don't give a hoot about the consumer market for their diskettes. They don't spend millions of dollars in advertising trying to get you, the computer user, to use their diskettes.

Instead, they concentrate their efforts on turning out the highest quality diskettes they can...because they sell them to the software publishers, computer manufacturers and other folks who (in turn) put their name on them...and sell them for much higher prices to you!

After all, when a software publisher or computer manufacturer or diskette marketer puts their name on a diskette, they want it to work time after time, everytime. (Especially software publishers who have the nasty habit of copy-protecting their originals!)

Super Star Diskettes. You already know how good they are. Now you can buy them...cheap.

Well, that's the story.

Super Star diskettes don't roll off the boat from Pago-Pago or emerge from a basement plant just east of Nowhere.

Super Star diskettes have been around for years...and you've used them for years as copy-protected software originals, unprotected originals. Sometimes, depending on which computer you own, the system master may have been on a Super Star diskette. And maybe more than once, you've bought a box or two or more of Super Star diskettes without knowing it. They just had some "big" company's name on them.

Super Star Diskettes are good. So good that a lot of major software publishers, computer manufacturers and other diskette marketers buy them in the tens or hundreds of thousands.

We buy them in the millions.

And than we sell them to you.

Cheap.

When every little bit counts, it's Super Star Diskettes.

You've used them a hundred times...under different names.

Now, you can buy the real McCoy, the same diskette that major software publishers, computer manufacturers and diskette marketers buy...and call their own.

We simply charge less.

Super Special!

Order 50 Super Star Diskettes and we'll be happy to sell you an Amaray Media-Mate 50 for only $8.75, shipping included...a lot less than the suggested retail price of $15.95.

Regular DISK WORLD! price: $9.69 ea. + $2.00 Shpng.

The Super Star LIFETIME WARRANTY!

Super Star Diskettes are unconditionally warranted against defects in original material and workmanship so long as owned by the original purchaser. Returns are simple: just send the defective diskettes with proof of purchase, postage-paid by you with a short explanation of the problem, and we'll send you the replacements. (Incidentally, coffee stained diskettes and diskettes with staples driven through them don't qualify as "defective".)

DISKETTE STORAGE CASES

DISK CADDIES

The original flip-up holder for 10 5¼" diskettes. Beige or Grey only.

$1.65 ea. + .20 Shpng.

DISKETTE 70 STORAGE

Dust-free storage for 70 5¼" diskettes. Six dividers included. An excellent value.

$9.95 ea. +$3.00 Shpng.

HOURS:
Human: 8AM-6PM Central Time, Monday through Friday
Answering Machine: 6PM-8AM, All Times
MCI MAIL: 24 hours a day.

HOW TO ORDER:

ORDERS ONLY:
1-800-621-6827
(In Illinois: 1-312-256-7140)
INQUIRIES:
1-312-256-7140

FOR FASTEST SERVICE, USE NO-COST MCI MAIL: Our address is DISKORDER. It's a FREE MCI MAIL letter. No charge to you. (Situation permitting, we'll ship these orders in 24 hours or less.)

SHIPPING: 5¼" & 3½" DISKETTES—Add $3.00 per each 100 or fewer diskettes. **OTHER ITEMS:** Add shipping charges as shown in addition to other shipping charges. **PAYMENT:** VISA, MASTERCARD and Prepaid orders accepted. **COD ORDERS:** Add additional $5.00 special handling charge. **APO, FPO, AK, HI & PR ORDERS:** Include shipping charges as shown and additional 5% of total order amount to cover PAL and insurance. We ship only to United States addresses, except for those listed above. **TAXES:** Illinois residents, add 7% sales tax.

MINIMUM ORDER: $35.00.

WE WILL MEET OR BEAT ANY NATIONALLY ADVERTISED PRICE ON THE SAME PRODUCTS AND QUANTITIES SUBJECT TO THE SAME TERMS AND CONDITIONS.

DISK WORLD!, INC.

629 Green Bay Road
Wilmette, Illinois 60091

Incredible value!

Nashua™ Diskettes

LIFETIME WARRANTY!

85¢ ea. 5¼" SSDD Qty. 50

97¢ ea. 5¼" DSDD Qty. 50

These are poly-bagged diskettes packaged with Tyvek sleeves, reinforced hubs, user identification labels and write-protect tabs. NASHUA Corporation is a half-billion dollar corporation and a recognized leader in magnetic media.

SOFT SECTOR ONLY! Sold in multiples of 50 only!

FOR ORDERS ONLY:
1-800-621-6827
(In Illinois: 1-312-256-7140)

INFORMATION & INQUIRIES:
1-312-256-7140

HOURS: 8AM-5PM Central Time, Monday-Friday

WE WILL BEAT ANY NATIONALLY ADVERTISED PRICE ON THE SAME PRODUCTS AND QUANTITIES!

DISK WORLD!, Inc.
629 Green Bay Road • Wilmette, Illinois 60091

DISK WORLD! Authorized Distributor NASHUA MAGNETIC MEDIA

ATHANA DISKETTES

The great unknown!

79¢ ea. ← 5¼" SSDD Qty. 50

85¢ ea. 5¼" DSDD → Qty. 50

You've used these diskettes hundreds of times…as copy-protected originals on some of the most popular software packages. They're packed in poly-bags of 25 with Tyvek sleeves, reinforced hubs, user identification labels and write-protect tabs.

LIFETIME WARRANTY!

SOFT SECTOR ONLY! Sold in multiples of 50 only.

FOR ORDERS ONLY:
1-800-621-6827
(In Illinois: 1-312-256-7140)

INFORMATION & INQUIRIES:
1-312-256-7140

HOURS: 8AM-5PM Central Time, Monday-Friday

WE WILL BEAT ANY NATIONALLY ADVERTISED PRICE ON THE SAME PRODUCTS AND QUANTITIES!

DISK WORLD!, Inc.
629 Green Bay Road • Wilmette, Illinois 60091

DISK WORLD! Authorized Distributor ATHANA MAGNETIC MEDIA

DISKETTE STORAGE CASES

AMARAY MEDIA-MATE 50: A REVOLUTION IN DISKETTE STORAGE

Every once in a while, someone takes the simple and makes it elegant! This unit holds 50 5¼" diskettes, has grooves for easy stacking, inside nipples to keep diskettes from slipping and severa' other features. We like it! **$9.69** ea. + $2.00 Shpng.

DISKETTE 70 STORAGE: STILL A GREAT BUY.

Dust-free storage for 70 5¼" diskettes. Six dividers included. An excellent value. **$9.95** + $3.00 Shpng.

DISK CADDIES

The original flip-up holder for 10 5¼" diskettes. Beige or grey only. **$1.65** ea. + 20¢ Shpng.

FOR ORDERS ONLY:
1-800-621-6827
(In Illinois: 1-312-256-7140)

INFORMATION & INQUIRIES:
1-312-256-7140

HOURS: 8AM-5PM Central Time, Monday-Friday

WE WILL BEAT ANY NATIONALLY ADVERTISED PRICE ON THE SAME PRODUCTS AND QUANTITIES!

DISK WORLD!, Inc.
629 Green Bay Road • Wilmette, Illinois 60091

DISK WORLD! The value leader in Computer supplies And accessories.

FANTASTIC LOW PRICES ON

BASF QUALIMETRIC DISKETTES!

LIFETIME WARRANTY!

Buy in bulk and save. 150 to the carton with envelopes, write-protect tabs and user ID labels. Boxed product is the same, except in cardboard boxes of 10.

	Qty. 50	Qty. 150
5.25" SSDD	.83 ea.	.80 ea.
5.25" DSDD	.94 ea.	.92 ea.
5.25" DSDD-HD	2.13 ea.	N/A
5.25" SSDD-96TPI	.94 ea.	N/A
5.25" DSDD-96TPI	1.06 ea.	N/A
3.50 SSDD-135TPI	1.84 ea.	1.68 ea.
3.50 DSDD-135TPI	2.40 ea.	2.28 ea.

NOTE: 3.50" diskettes in Quantity 50 are packed in plastic library cases. That's why they seem to be a better buy. But there are only 5 diskettes to a case…so the bulk diskettes are really a better deal, unless you like expensive little library cases.

FOR ORDERS ONLY:
1-800-621-6827
(In Illinois: 1-312-256-7140)

INFORMATION & INQUIRIES:
1-312-256-7140

HOURS: 8AM-5PM Central Time, Monday-Friday

WE WILL BEAT ANY NATIONALLY ADVERTISED PRICE ON THE SAME PRODUCTS AND QUANTITIES!

DISK WORLD! Authorized Reseller Information Processing Media BASF

DISK WORLD!

Ordering & Shipping Instructions

Shipping: 5¼" & 3.5" DISKETTES—Add $3.00 per each 100 or fewer diskettes. **Other Items:** Add shipping charges as shown in addition to other shipping charges. **Payment:** VISA and MASTERCARD accepted. **COD Orders:** Add additional $5.00Special Handling charge. **APO, FPO, AK, HI & PR Orders:** Include shipping charges as shown **and** additional 5% of total order amount to cover PAL and insurance. **Taxes:** Illinois residents only, add 7% sales tax.

Prices subject to change without notice.
This ad supercedes all other ads.
Not responsible for typographical errors.
MINIMUM TOTAL ORDER: $35.00

FOR ORDERS ONLY:
1-800-621-6827
(In Illinois: 1-312-256-7140)

INFORMATION & INQUIRIES:
1-312-256-7140

HOURS: 8AM-5PM Central Time
Monday-Friday

WE WILL BEAT ANY NATIONALLY ADVERTISED PRICE ON THE SAME PRODUCTS AND QUANTITIES!

DISK WORLD!, Inc.
629 Green Bay Road • Wilmette, Illinois 60091

DISK WORLD!

PRINTER RIBBONS:

at extraordinary prices!

Brand new ribbons, manufactured to Original Equipment Manufacturer's specifications, in housings. (Not re-inked or spools only.)

LIFETIME WARRANTY!

Epson MX-70/80	**$2.70** ea.	+ 25¢ Shpng.
Epson MX-100	**$4.08** ea.	+ 25¢ Shpng.
Okidata Micro83	**$1.37** ea.	+ 25¢ Shpng.
Okidata Micro84	**$2.98** ea.	+ 25¢ Shpng.

FOR ORDERS ONLY:
1-800-621-6827
(In Illinois: 1-312-256-7140)

INFORMATION & INQUIRIES:
1-312-256-7140

HOURS: 8AM-5PM Central Time, Monday-Friday

WE WILL BEAT ANY NATIONALLY ADVERTISED PRICE ON THE SAME PRODUCTS AND QUANTITIES!

DISK WORLD!, Inc.
629 Green Bay Road • Wilmette, Illinois 60091

DISK WORLD!

Now, the lowest prices ever on

3M Scotch® DISKETTES

LIFETIME WARRANTY!

The best deal on 3M diskettes you've ever seen!

FREE!

FLIP 'N FILE 15 with every order for 5.25" SSDD and DSDD.

$1.36 ea. …. 5.25" SSDD with FREE Flip 'n File 15
5.25" DSDD with FREE Flip 'n File 15…**$1.67 ea.**

MINIMUM ORDER: 50 Diskettes
(Orders less than 50 are $1.39 and $1.70, respectively.)

100% LIFETIME WARRANTY!

This is a Super Special Promotion. It was supposed to end May 31, 1985.

But we decided to buy more than 1,000,000 3M diskettes packed in the FREE Flip 'n File 15…and give you the benefits of this terrific value.

One word of warning: this offer is limited only to supplies on hand. Once this inventory is gone, that's it. The prices stay the same…but there's no FREE Flip 'n File.

The last time we ran an offer like this, everything sold out in about six weeks.

So don't wait. Order now.

Other 3M diskettes

	Qty. 20	Qty. 50
5.25" SSDD-96TPI	$2.07 ea.	$2.04 ea.
5.25" DSDD-96TPI	$2.51 ea.	$2.48 ea.
5.25" DSDD-HD FOR AT	$3.18 ea.	$3.14 ea.
3.50"SSDD-135TPI (Mac)	$2.29 ea.	$2.26 ea.

DATA CARTRIDGES

100% certified 3M data cartridges

DC-1000	$12.62 ea.
DC-300XLP	$19.38 ea.
DC-600A	$20.61 ea.

Sold in cases of 10 only.
Add $5.00 shipping per 10.

FIRST TOUCH KEYBOARD STRIP

The best way we've ever seen to avoid static danger. Adheres to the front ledge of your keyboard. Overpriced at $19.95, so we cut the price to $10.95 + $1.50 Shipping.

A VERY SPECIAL OFFER
3M POST-IT NOTES & DISPENSER

Suggested Retail: $20.95. Includes dispenser, 12 pads of 3" x 5" Post-It notes and Pilot Ball Point Pen. Order it with 50 3M diskettes or 10 data cartridges and it's only $9.75 + $3.00 Shipping.

DISK WORLD! Authorized Distributor Information Processing Products 3M

MUST LIQUIDATE At Far Below Dealer Cost!

TOTAL Personal Computer System

Factory Reconditioned with Factory Warranty!

Carries easily as a suitcase! Plugs into 115V outlet!

GREAT GIFT IDEA
FOR STUDENTS!

Sorry, we're not permitted to PRINT the famous brand-name. BUT, we CAN "tell all" if you call us TOLL FREE: 1-800-328-0609!

THE COMPUTER

Snap-on computer keyboard! 64K RAM, 20K ROM. Full-size typewriter keyboard. Upper and lower case letters, numerals, symbols, reverse characters. 2 cursor control keys, 4 function keys, programmable to 8. Music synthesizer with 3 independent voices, each with 9 octave range. Input/output ports accommodate . . . user, serial, ROM cartridge, joysticks, external monitor, phone modem.

Built-in disk drive! Intelligent high speed unit with 5¼" floppy disk recorder. 170K formatted data storage; 35 tracks. 16K ROM. Uses single sided, single density disk. Serial interface. Second serial port to chain second drive or printer.

Built-in color monitor! Displays 40 columns x 25 lines of text on 5" screen. High resolution. 320 x 200 pixels. 16 background, character colors.

Built-in ROM cartridge port! Insert ROM program cartridge. Multitude of subjects available in stores across the nation!

Original List Price **$995.00**

Liquidation Priced At Only **$388**

Item H-725-63631-00 Ship, handling: $20.00

THE PRINTER

Print method: Bi-directional impact dot matrix.

Character matrix: 6 x 7 dot matrix.

Characters: Upper and lower case letters, numerals and symbols. All PET graphic characters.

Graphics: 7 vertical dots — maximum 480 columns. Dot addressable.

Character codes: CBM ASCII code.

Print speed: 60 characters per second.

Maximum columns: 80 columns.

Character spacing: 10 characters per inch.

Line feed spacing: 6 lines per inch in character mode or 8 lines per inch selectable. 9 lines per inch in graphics mode.

Line feed speed: 5 lines per second in character mode. 7.5 lines per second in graphics mode.

Paper feed: Friction feed.

Paper width: 4.5" to 8.5" width.

Multiple copies: Original plus maximum of two copies.

Dimensions: 13"W x 8"D x 3¼"H. Wt.: 6½ lbs. **Power:** 120V AC, 60 Hz.

Original List Price: $200.00

Liquidation Priced At **$118**

Item H-725-63831-00 Ship, handling: $7.00

THE SOFTWARE

"Easy Script" One of the most powerful word processors at any price! Cut re-typing, create documents from standard paragraphs, do personalized letters, see and change a document before it is printed. Instruction manual has extensive training section that simplifies use . . . even for someone who has never used a computer or word processor before!

"The Manager" A sophisticated database manager for business or home use. Business uses: accounts payable/receivable, inventory, appointments, task manager. Home uses: mailing lists, home inventory, recipes, collection organizer, investment tracking, checkbook balancing. School uses: research article index, gradebook.

Original List Price: $73.98

Liquidation Price **$24**

Item H-725-64011-03 Ship, handling: $3.00

BUY INDIVIDUAL UNITS OR GET THIS ULTRA-FAMOUS SYSTEM AT ONE LOW PACKAGE PRICE!

TOTAL Personal Computer System available at FAR BELOW dealer cost!

Original List Price **$1,293.00**

TOTAL PACKAGE PRICE **$488**

Item H-725-64011-02 Ship, handling: $24.00

Compatible with above Computer System (Not included in package price.)

JOYSTICKS (Set of 2)

Mfr. List: $59.90 pr.

Liquidation Price . . . . **$18 pr.**

Item H-725-63622-00 S/H: $6.00 pr.

64 MODEM

Mfr. List: $124.95

Liquidation Price **$44**

Item H-725-63646-00 S/H: $4.00

VISA MasterCard

Credit card members can order by phone, 24 hours a day, 7 days a week.

Toll-Free: 1-800-328-0609

TeleCheck **Your check is welcome! No delays when you pay by check!**

C.O.M.B. CO.®
Authorized Liquidator
14605 28TH AVENUE NORTH
MINNEAPOLIS, MINNESOTA 55441-3397

SEND ME THE ITEMS I HAVE LISTED BELOW

Price subject to change after 60 days. Sales outside continental U.S. are subject to special conditions. Please call or write to inquire.

Item No.	#	Item	Price	S/H
		TOTAL		

C.O.M.B. CO.® Item H-725
14605 28th Ave. N./Minneapolis, MN 55441-3397

Send the items indicated at left. (Minnesota residents add 6% sales tax. Please allow 3-4 weeks delivery. Sorry, no C.O.D.)

□ My check or money order is enclosed. (No delays in processing orders paid by check, thanks to TeleCheck.)

Charge: □ MasterCard® □ VISA

Acct. No. ______________ Exp. ______

PLEASE PRINT CLEARLY

Name ______________

Address ______________

City ______________

State ______________ ZIP ______

Phone ______________

Sign here ______________

WHITE HOUSE COMPUTER

P.O. Box 4025, Williamsport, PA 17701

"Where Prices Are Born, Not Raised"

TOLL FREE 1-800-351-3442

PA CALL 1-717-322-7700

— PA Residents FREE Shipping —

MONDAY - FRIDAY 9 AM - 6 PM
VISA & MC ACCEPTED 4%

POLICY: No deposit on COD orders. Free freight on all prepaid cash orders over $300 in the continental USA. APO & FPO add $5.00 per hundred. For priority mail add $8.00 per hundred. PA residents add 6% sales tax. Defective products must have Prior RA number. Schools net 15.

PRINTERS

EPSON	
RX 80	209.00
RX 100	369.00
JX 80	479.00
FX 85	345.00
FX 185	499.00
LQ 1500 PAR	979.00
LQ 1500 SER	1039.00
LX 80	222.00
Homewriter	209.00
Comrex 220 Atari	199.00
Comrex 220 Comm.	199.00
LX 90	245.00
SQ 2000	1525.00
DX 10 Daisywheel	235.00
DX 20 Daisywheel	319.00
HS 80 Letterjet	359.00
ATARI	
XTM 201	99.95
XTC 201	109.95
XDM 121	209.95
XMM 801	169.95
STC 504	139.95
STD 121	219.95
SMM 801	279.95
CITIZEN	
MSP 10	305.00
MSP 15	450.00
MSP 20	450.00
MSP 25	575.00

OKIDATA	
Okimate 10	179.95
182	219.95
84	640.95
192	**349.95**
193	525.95
COMMODORE	
MPS 801	165.95
802	199.95
803	165.95
DPS 1101	295.95
STAR MICRONICS	
SG-10	**214.00**
SG-15	379.00
SD-10	345.00
SD-15	450.00
SR-10	485.00
SR-15	585.00
Powertype	307.00
LEGEND	
1380	**269.00**
1385	**305.00**
1080	229.00
880	199.00
PANASONIC	
1090	187.00
1091	CALL
1092	385.00
1093	425.00
3151	455.00

INTERFACES

850 (Atari)	108.95
UPrint/port	49.95
UPrint/16k Buffer	**69.95**
UPrint/64k Buffer	89.95
Cardco G	39.95
G-Wiz	**48.95**
Apple Dumpling GX	**59.95**

PRINTER PAPER

2500 Sheets	
Lazor Edge	**24.95**
1000 Shts Lazor	14.95
500 Shts Lazor	9.95
Color Paper Assorted Pastels	
2500 Shts Lazor	42.95
1000 Shts Lazor	23.95
500 Shts Lazor	14.95

SOFTWARE

Print Shop	28.95
Graphics Library I	17.50
Graphics Library II	17.50

VIDEO CASSETTE RECORDER
$279.00

DISK DRIVES

ATARI	
Indus GT	**205.95**
1050	164.95
Happy 1050	335.95
Happy Enhancer	160.95
MSD Dual	459.00
C-64	
Indus GT	**229.00**
1541	195.00
1571	249.95
1572	375.95

MODEMS

C-1650	54.95
C-1670 300/1200	185.95
MPP 1064	54.95
Tele Learning	39.95
Hayes 300	149.95
Hayes 1200	385.95
Mitey Mo	59.95
Westridge	59.95
Compuserve	19.95
Micro Stuffer	95.95
MPP 1000E	**69.95**

DISKETTES

SKC	
SS/DD	10.95
DS/DD	14.95
BONUS	
SS/DD	9.50
DS/DD	13.50
MAXELL	
MD 1	15.95
MD 2	20.95

MONITORS

ZENITH	
122 A	82.95
123 G	75.95
AMDEK	
300 G	119.00
300 A	129.00
Color 300	**229.00**
Color 500	339.00
Color 600	399.00
Color 700	469.00
Color 710	539.00
310A	145.00
ATARI	
XC 141	199.95
SM 124	135.95
SC 1224	335.95
TEKNIKA	
MJ-10	185.95
MJ-22 RGB	**269.95**
SAKATA	
SC-100	179.00
COMMODORE	
C-1702	185.95
C-1902 RGB	259.95
C-1901	129.95

COMPUTERS

COMMODORE	
C-64	139.95
C-128	275.95
ATARI	
800 XL	89.95
130 XE	139.95
520 ST Monochrome & More	CALL
520 ST Color RGB & More	CALL

IT'S LIKE

FREE DISKETTES

Your 5¼" single side disks are usable on the other side. You paid for one side, why not use the other... **IT'S FREE!**

Nibble Notch will **open** your **new** disk. It's easy... won't harm existing data. Try it!

nibble notch I	nibble notch II
For Apple, Franklin, Commodore & Atari (w/Atari Drives); square notch.	For all other computers; square notch & index hole.
only 14^{95}* PLUS P&H	only 21^{90}* PLUS P&H

DISK OPTIMIZER II©

Apple II Series Software
Pro DOS • DOS 3.3 • Pascal
Examines your **new** disk, locks out bad sectors and **certifies it 100% ERROR-FREE** in 30 seconds or less! Also checks drive speed... and more!

SUPER SAVER PACKAGE
Nibble Notch I and Disk Optimizer Combo (Optimizer alone reg. $24.95)
$29.95* FOR BOTH!

QUALITY DISKETTES low as 99¢

*add $2 ($5 frgn) for P & H. Fl. Res. add 5% Sales Tax

Toll Free 1-800-642-2536
FL 1-305-748-3770
OR SEND CHECK OR MONEY ORDER TO:

nibble notch® computer products

VISA MasterCard

4211 NW 75th TERRACE, • DEPT. 662 LAUDERHILL, FL 33319

PROFESSIONAL HANDICAPPING SYSTEMS

• PRESENTED BY PROFESSOR JONES •

GLD. Thoroughbred "Gold" Edition™
A "Full" featured thoroughbred analysis designed for the professional and the serious novice. **$159.95** complete

EGLD. Enhanced "Gold" Edition™
"Gold" Edition with complete Master Bettor™ system integrated onto the same disk. This powerful program will transfer all horses and scores to the bet analysis with a "single keystroke." (Master Bettor™ included) **$199.95** complete

GLTD. Limited "Gold"™
Enables **Professional Handicappers** to assign specific values to the racing variables "they" feel are important. Create program weight based on a particular track and fine tune it for **maximum** win percentage. This program is designed for "ease of use". The user needs **no programming experience.**
(contains Integrated Bettor™) **$299.95** complete

GD. Gold Dog Analysis™
The "ONLY" professional greyhound analysis available that evaluates ALL variables. **$149.95** w/integrated Bettor **$199.95**
Limited Version **$299.95**

MHH. Master Harness Handicapper™
Professional software designed to provide a thorough analysis of all trotter and pacer races in North America and Canada. **$159.95** complete
$199.95 w/Integrated Bettor Limited **$299.95**

Professor Pix Football™
Complete STATISTICAL ANALYSIS on Data Base allowing "Designated" previous games to be evaluated. Statistical Series **$99.95**
w/Win-Loss Power Ratings **$149.95**

NBA. Basketball™ NBA **$99.95** w/college **$129.95**
w/power ratings **$149.95**

LOT. Lottery Analysis™
Statistical comparison program designed to detect subtle patterns in winning numbers and digits. Lottery (3-4 digit) **$79.95**
w/Lotto (Max. 99 Digit) **$99.95**

PC-3 Portable Computer (4k) with choice of Thoroughbred, Greyhound or Trotter™ **$249.95** (Includes portable computer and program.)

M-100 Portable (24k) w/choice of Thoroughbred, Greyhound, or Trotter. **$649.95** (Includes portable computer and program.)

BROCHURE AVAILABLE

IBM™
APPLE™
TRS-80™
CPM™
COMMODORE™

Prof. Jones
1940 W. State St.
Boise, ID 83702

VISA MasterCard

48 HR. FREE SHIPPING

☎ CALL 208-342-6939

TERMS: FREE SHIPPING ALL SOFTWARE. Add $6.00 hardware / $6.00 C.O.D. / UPS Blue $6.00 / Out of Country $9.00 / ID Residents 4% / 3 weeks personal checks / Cash price only, add 2% Visa, MC / Prices subject to change.

$699 Computer Breakthrough

IBM compatible computer with 128K memory and two disk drives is more than a bargain.

The Visual Commuter sells for only $699. The IBM monochrome monitor is optional.

A *It carries like a briefcase.*

B *With LCD lid open.*

It's a new concept. And if my hunches are correct, the new Visual Commuter will take years for others to copy.

By Joseph Sugarman, President

First, it is a tremendous bargain. At $699 nothing even comes near. Secondly, it's powerful. You get 128K memory that's expandable to 512K. And finally, it's totally IBM compatible. It will even run the Flight Simulator program and Lotus Symphony™ But there's more.

The unit can be used as a stand-alone office computer as shown above or it can be packed up and used as a portable—but without all the weight. There's an optional 16-line x 80 character LCD display that pops up to replace the heavy CRT monitor. Unlike the smaller portable computer keyboards, the Commuter keyboard is a full-size replica of the IBM with its ten function keys and numeric key pad. And the Commuter uses 5¼" disks so you have full access to all the popular IBM software.

It's lighter (only 18 lbs), flatter (only 3½" thick) and carries like a briefcase. In short, the Visual Commuter is a combination of all the good features of a portable computer (size, weight, portability), all the good features of a traditional desktop computer (full-size keyboard, 5¼" disks, full power) and none of the disadvantages of either.

USE IBM PROGRAMS

Even if you have another computer but miss some of the IBM programs, for only $699 and an IBM compatible monitor, you've got a complete MS/DOS system.

It was also made modular so you can select just those components that you need for your particular application. For example, you may not need the 16 line x 80 column LCD display which adds 2 lbs to the unit's weight (a blank lid comes with the unit). Or you may not need a monitor because your other computer may already have one. But you may want more power—256K or 512K—so you order just what you need.

For all you technical people, listen to these specs. There's a 16-bit 8088 processor, 128K memory with parity, parallel printer port, serial ASYNC RS 232C port, Din connector RF modulator or composite video output for TV and composite video input monitors, RGB/direct drive output for high resolution monochrome or color monitors, IBM compatible color graphic support, support logic for 80x25 or 40x25 character display and LCD display, connector to IBM expansion unit, disk controller supporting two 5¼" disk drives, ANSI standard ROM-based terminal emulation, and ROM-based extended diagnostics. The dual disk drives are double sided-double density (360 Kbytes). The Commuter runs at the same clock speed as the IBM PC (4.77 MHz) but because of its new design, it runs between 8 to 10 percent faster.

ATTRACTIVE CASE

There's an attractive carrying case made by American Tourister that holds your software, your power cord, your documents and even our optional 1200 baud modem. The compatible Maxwell modem lets you communicate with other data banks. Made by the world's largest modem manufacturer, Racal-Vadic, it is normally a $500 value but our price is only $249 which includes a complete communications software package. There's also a toll free, online warranty service and a customer hotline to answer any of your technical questions.

You may have recently heard of Visual Technology Incorporated. They are innovators in the design and manufacture of smart alphanumeric terminals and some of the finest graphic terminals in the country.

The Visual Commuter was scheduled to sell for over $2500 with the LCD display. And even at that price, when compared to the IBM system, it was a good value. But JS&A and Visual (in a joint venture with SGD Holding Corp.) saw the opportunity of having just one customer. Together, by selling directly to you, we've eliminated the distributors, dealers and all the sales, administration and advertising costs and have passed the savings on to you. But there are a few catches.

JUST A FEW THOUGH

Once we install the memory, you'll have to send the unit back to us to add more memory. So we ask that you estimate, in advance, the maximum power that you'll require for your needs. 128K memory is plenty for most applications but if you want to run Lotus Symphony, you'll need all 512K. Secondly, we ask that you act quickly. Although we have most of the product in stock right now, there's always the chance that we'll run out.

The Visual Commuter measures only 3½ x 15½ x 18" wide and comes complete with power cord (it only operates on standard AC current), the operating system (Micro-Soft's MS/DOS ver. 2.1) complete with basic and utilities, two beautifully written manuals, lid (without LCD display) and a limited 90-day warranty. There are service centers throughout the United States set up to service the unit in addition to the service-by-mail facility at Visual's home office near Boston.

I urge you to give the Visual Commuter a test. Order one from JS&A and use it for 30 days without risk. Plug in your IBM monitor and load any of the IBM software you currently have. See how the large keyboard matches the IBM perfectly and how its handle makes a perfect hand rest while typing or a comfortable handle for carrying the unit. See how convenient the unit is to take home or bring with you on a trip with its fold open LCD monitor. If you don't feel that the Visual Commuter is more than you expected, pack it up and ship it back within 30 days for a prompt and courteous refund including the $25 postage charge. You can't lose.

PERSONALLY USED

I have personally used the Visual Commuter. I have taken it with me on trips, set it up as a stand alone by plugging in my IBM monitor. I have run everything from Symphony™ to Wordstar®—from 1-2-3® to the Flight Simulator program. I strongly recommend the system.

To order, send your check or money order to the address below listing the items and order numbers (shown in parentheses) or order by credit card using our toll free number below.

Thanks to the latest in technology and a direct-to-consumer marketing program, this system can be yours at a tremendous value. Order your Visual Commuter at no obligation, today.

Visual Commuter with 2 Disk Drives and 128K Memory (6105)	**$699**
16 x 80 LCD Alphanumeric Display (6106)	**299**
IBM 12" Green Monitor (6107)	**229**
Zenith 12" Amber Monitor (6114)	**99**
Expansion to 256K (6110)	**199**
Expansion to 512K (6111)	**399**
Maxwell 1200 Baud Modem (6112)	**249**
Carrying Case (6115)	**69**

Add $25 postage and handling per system.
IBM is a Trademark of International Business Machines Corporation.

JS&A PRODUCTS THAT THINK

Dept. PT One JS&A Plaza
Northbrook, Il 60062
QUESTIONS ONLY: (312) 564-7000
CALL TOLL FREE 800 228-5000

IL residents add 7% sales tax. © JS&A Group, Inc. 1985

Classified

SOFTWARE

ATARIWARE FROM THE LIBRARY OF JAX ATARI - Computer Enthusiasts! Over 70 disks to choose from. $7.95 ea. Catalog with SASE. JACE, 1187 Dunbar Ct., Orange Park, FL 32073

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

ATARI ST USERS - Life organizer & entertainment jackpot. Big software package. Write: MC, 94 Macalester Bay, Winnipeg, Manitoba R3T 2X5 Canada

BASICally SIMPLE, a fast, easy method to learn programming in basic on your Commodore 64. Clear, concise instruction in the use of all commands. No gobblede-gook. Send $20 for this tutorial and reference disk to:
Free Spirit Software, Inc., 5836 S. Mozart, Chicago, IL 60629

TI-99/4A SST Expanded Basic Compiler $49.00. Converts Basic to machine language. Requires E/A or MM, Disk, Memory Expansion. Compiler and Graphics Utility $59.00. SST SOFTWARE, P.O. Box 26, Cedarburg, WI 53012

HOW GOOD A COACH ARE YOU? CALL THE plays against Atari or a friend. Football text game features varied team strength, weather, overtime, 2 pt. conversions, team stats, standings + more! Easy to play. For Atari 800 + Basic. $25 disk. James Franklin, 1600 NE 135 St., Apt. 803, N. Miami, FL 33181

Free Educational Software Catalog
158 pp, color pictures, 1000+ programs for Com., Apple, IBM, Atari. Send name & address to: Interstate Software, P.O. Box 8952, Boise, ID 83707, (208) 342-3347

ATARI MICROSOFT Basic I/II: Try ACEY DUCEY, a great casino card game for 1–4 players. ACEY DUCEY is designed as a tutorial, utilizing Microsoft Basic's best features. Full documentation explains every detail. 400/800/XL/XE disk or tape + docum. Send $8.50. CALTECH, Box 136, Kenilworth, NJ 07033-0136

ORGANIZE YOUR GENEALOGY with BRANCHES on your C-64 & 1541 drive. Send $14.95 to Inventure Soft., 3353 South Main, Suite 101, Salt Lake City, UT 84115

TI-99/4A QUALITY SOFTWARE for Business, Home and Entertainment ** BONUS Software Offer! ** Send for FREE Catalog to MICRO-BIZ HAWAII, Box 1108, Pearl City, HI 96782

AT BAT STATS 64 - Baseball stats for C64. Keeps 22 offensive stats, holds 162 games and 40 players. Prints 7 reports. On disk $32.95 postpaid. THE 43RD ST. SOFTWARE SHOP, 720 43rd St., Ashland, KY 41101. Ideal for coach or player! (606) 324-7403.

GOMUKO — Compiled basic program 15 by 15 board, Unmove, load/save game user changable logic for 64. Send $14.95 to: W. L. Schneider, 6027 S. High, Suite 410, Oklahoma City, OK 73149

IBM PCjr. OWNERS: We have the software you want. We also have kits for easy 2nd drive addition. Free catalog. OOWL SOFTWARE, 1435 Burnley Sq. N., Columbus, OH 43229

TEACHERS: ELECTRONIC GRADEBOOK for Atari. Powerful, simple to use. Record keeping, grade averaging. Screen display and variety of pre-formatted printed reports. Clear instructions. Requires 48K and one disk drive. $29.95. GRADESCAN, 1722 Golden Ct., Crofton, MD 21114

TI-99/4A Software/Hardware bargains. Hard-to-find items. Huge selection. Fast service. Free catalog. D.E.C., Box 690, Hicksville, NY 11801

REAL ESTATE SOFTWARE for C64, Apple II, IBM-PC. Includes Mortgage Payments, Amortization, Depreciation, Cash Flows and Taxable Income Effects. All for $35 pp. J.A.I., Box 234, East Texas, PA 18046. (215) 398-0434

APPLE II SERIES, COMM 64, TI 99/4A, IBM/PC Discount Entertainment Software. Specify which free catalog. New titles. The Adventure Store, Roscoe, IL 61073-0320

TOR FIGHTER: You are entrusted with the daring adventure of defending earth, $12. C64, C16, +4 (disk), TI (cass.). Check or money order to: Computex, Box 7245, Ponce, PR 00732

TIM, The Investment Manager and COMEX Gold and Silver Management, plus Fun-Anl Analyze Stock Program for 64. All three $19.95 or write for free details to:
W. L. Schneider, 6027 S. High, Suite 410, Oklahoma City, OK 73149

LOTTO PICKER. Go for Million Dollar Jackpots! Picks all USA Lotto games +! PRO FOOTBALL ANALYST. Beat the points consistently and easily! They pay for themselves! Each $35. IBM-PC/TI99 ver. RIDGE, 170 Broadway, #210-C NYC, NY 10038. 718-833-6335. Free Catalog.

HARDWARE

Trade in your used Commodore or Atari on a brand new C-128 or Atari ST. This offer may not be available through retail outlets. Brochure $2.00 and SASE. NEW WEST TECHNOLOGY, 4B Monroe Pkw., Box 200, Ste. 134, Lake Oswego, OR 97034

HARDWARE & SOFTWARE 30% BELOW RETAIL. Apple, Atari, C64, IBM-PC, TI-99. Over 1000 titles. Hard to find items. Send $1.00 for catalog. Specify computer. Multi-Video, P.O. Box 246, East Amherst, NY 14051

MISCELLANEOUS

64 AUTHOR'S CLUB - We get you published. Send for free details or send $25.00 (a 50% savings) to: AUTHOR'S CLUB, 6027 S. High, Suite 410, Oklahoma City, OK 73149

1986 CALENDAR. The history of computers 1592–1985, especially micros. $6 ch or m.o. Spiral, 3 holes, 8½ × 11. HEATSTROKE SOFTWARE, P.O. Box 62171, Tucson, AZ 85734-6171

HELP IS ON THE WAY!
Just call 1-800-334-0868 to get your free copy of the latest COMPUTE! Books Catalog! If you need help in getting information on all of the latest COMPUTE! book titles available plus all COMPUTE! backlist titles, call us today!

NATIONWIDE BBS DIRECTORY. 250 exciting network numbers. Computer to computer. Coast-to-coast and Canada. Send $3 to: Compuquest, Box 267, Fairfax, VA 22030-0267.

RIBBONS for ANY PRINTER at LOW PRICES!!
DELTA MICRONICS
BOX 10933, ERIE, PA 16514
(814) 455-5667

*** MR. SOFTWARE CO. ALL POPULAR TITLES ***
* Pritners, Monitors, Drives, VISA, MC *
heavy discounts - Send $1.00 for catalog
11-9 Exton Complex, Somers Point, NJ 08244

FREE! USE YOUR MODEM! Call our innovative electronic shopping center, FANTASY PLAZA. VISA and MASTERCARD accepted. 300 BAUD. You've never seen anything like it! Use your Modem NOW! (818) 840-8066

EARN MONEY, PART OR FULL TIME, AT HOME with your computer. 50 page manual with forms. Money back guarantee, $9.95, JV Tech, P.O. Box 563, Ludington, MI 49431

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

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

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

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

General Information: Advertisers using post office box numbers in their ads must supply permanent address and telephone numbers. Orders will not be acknowledged. Ad will appear in next available issue after receipt.

Closing: 10th of the third month preceding cover date (e.g., June issue closes March 10th). Send order and remittance to: Harry Blair, Classified Manager, COMPUTE!, P.O. Box 5406, Greensboro, NC 27403. To place an ad by phone, call Harry Blair at (919) 275-9809.

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

maxell DISKS

LIFETIME WARRANTY

TIRED OF WAITING
FOR SERVICE AND PRICE?
9 out of 10 SURVEYED
DISK BUYERS PREFERRED

NORTH HILLS
#1 IN SERVICE AND PRICE
1-800-328-3472

Formatted and hard sectored disks
in stock-Dealer inquiries invited.
COD, VISA, MASTERCARD
All orders shipped within 24 hrs.

NH

NORTH HILLS CORP.
INTERNATIONAL
3564 Rolling View Dr.
White Bear Lake, MN. 55110
MN. call collect—612-770-0485

Maxell Floppy Disks

The Mini-Disks
with maximum quality.

Dealer inquiries invited. C.O.D's accepted.
Call FREE (800) 235-4137.

PACIFIC EXCHANGES
100 Foothill Blvd., San Luis
San Luis Obispo, CA 93401.
In Cal. call (800) 592-5935 or
(805)543-1037.

pd

STATE-OF-THE-ART
MAGNETIC MEDIA

5¼" DISKETTES

- With Hub Rings
- With Write Protect Tabs
- With Static-Free, Dust-Free Envelopes
- With User ID Labels
- In Factory Sealed Poly Packs

69¢ Single Side Double Density	**79¢** Double Side Double Density

100% ERROR FREE - LIFETIME WARRANTY
MEET OR EXCEED APPLICABLE ANSI, ECMA, ISO
STANDARDS

Minimum quantity: 50 diskettes. Discount for 300 or more diskettes. **Shipping and Handling:** $4.00 per 100 diskettes. Reduced shipping charge for larger quantities. C.O.D. add $4.00. Cash or certified check. MI residents add 4% sales tax. Prices subject to change without notice.

C.O.D.

Precision Data Products
P.O. Box 8367, Grand Rapids, MI 49508
(616) 452-3457 • Michigan 1-800-632-2468
Outside Michigan 1-800-258-0028

3M Diskettes
Lifetime Warranty

TIRED OF WAITING
FOR SERVICE AND PRICE?
9 out of 10 SURVEYED
DISK BUYERS PREFERRED

NORTH HILLS
#1 IN SERVICE AND PRICE
1-800-328-3472

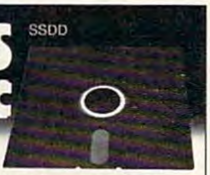
Formatted and hard sectored disks
in stock-Dealer inquiries invited.
COD, VISA, MASTERCARD
All orders shipped within 24 hrs.

NH

NORTH HILLS CORP.
INTERNATIONAL
3564 Rolling View Dr.
White Bear Lake, MN. 55110
MN. call collect—612-770-0485

DISKS 75¢

SSDD



QUALITY MEDIA **LIFETIME REPLACE** **GUARANTEE** **HUB RINGS, TYV. EPS.** **WRITE PROTECTS**	Because we buy in huge volume to supply software manufacturers, our prices can't be beat.

		50+	250+	500+	1000+
5.25" SSDD	1.00	.90	.85	.80	.75
5.25" DSDD	1.15	1.05	1.00	.95	.85
PC FORMATTED	1.45	1.35	1.25	1.15	1.10
3.5" 1D (For Mac)	2.70	2.60	2.45	CALL	CALL
3.5" 2D (For HP)	3.50	3.25	3.15	CALL	CALL

BLACKSHIP
COMPUTER SUPPLY

P.O. Box 883362 • San Francisco, CA 94188
Inquiries + No. Cal. orders 415-550-0512
So. Cal. orders 800-235-6647 x31
USA orders 800-431-6249

Add $3.00 shipping and handling
(CA residents add 6.5% sales tax) VISA/MC/COD

COMPUCAT

Catalog of Computers and Supplies
Our Prices are **WHOLESALE** +10%

Call, Write or
send an E-Letter for our
Free Catalog!

We support the complete **ATARI** and **COMMODORE** product lines. Ask for our free price list.

(408) 353-1836

(Ask about our Free Software.)

Instant shipping (or as fast as we can). *Mastercard & Visa accepted* (no extra charge). Shipping & handling add 6%. California customers add 6.5% sales tax. Order by *phone* (Mon. - Fri. 10 am - 5 pm PST). Order by *modem* (daily 6 pm-9am) from our online TeleCatalog. Prices subject to change without notice.

COMPUCAT
24500 Glenwood Hwy., Los Gatos, CA 95030

J&R MUSIC WORLD

23 PARK ROW, NEW YORK, N.Y. 10038

ORDER TOLL-FREE **800-221-8180** IN NEW YORK, ALASKA & CANADA **CALL:** (212) 693-0396

Shop at home and save 10% to 50% and more . . . on quality merchandise —fully guaranteed

maxell.
GOLD STANDARD FLOPPY DISKS

Avoid floppy disk trial and error. Start with Maxell Performance you can depend on, with a lifetime warranty

MAXELL MD1D Single sided/double density 10-pack	$19.95
MAXELL MD2D Double sided/double density 10-pack	$24.95

commodore 128
NEW
128K PERSONAL COMPUTER

•Three separate processort •Runs virtually all Commodore 64 software •New easy-to-use keyboard design

Call For Price

'TOP 10' COMPUTER HARDWARE

ATARI **520ST COLOR COMPUTER SYSTEM**

•512K RAM
•500K Mini-floppy drive
•RGB Color Monitor
•MAC-LIKE Graphics Menus Windows

$849.95

COMMODORE 64 COMPUTER *Great graphics, 64K RAM*	Call
COMMODORE 1541 DISK DRIVE *Adds fast storage/retrival*	$179.95
ATARI 800XL COMPUTER *64K RAM, graphics, sound*	$89.95
ATARI 1050 DSIK DRIVE *Dual density, DOS included*	$169.95
IBM PC PERSONAL COMPUTER *256K RAM, Dual 360K drives*	$1799.95
APPLE IIe PRO SYSTEM *•128K, dual drives, monitor*	$1369.95
COMMODORE 1702 COLOR MONITOR *Composite input*	$189.95
ZENITH ZVM-123 AMBER MONITOR *12" screen, 40/80 column*	$99.95
XTRON IA-12X RGB MONITOR *IBM compatible, 12" screen*	$109.95

'TOP 10' COMPUTER PRINTERS

STAR MICRONICS SG-10 DOT MATRIX COMPUTER PRINTER

•Built-in buffer
•Near letter quality mode •Up to 120 cps

$249.95

CANON PW1080A IMPACT MATRIX *FX compatible, 160 cps*	$279.95
AXIOM ELITE-5 LETTER QUALITY *C64 interface, daisy wheel*	$189.95
JUKI 6100 LETTER QUALITY *Triple pitch, daisy wheel, parallel*	$369.95
Commodore MPS803 DOT MATRIX *C64 serial interface, 60 cps*	$139.95
ATARI 1027 LETTER QUALITY *Atari compatible, 12 pitch*	$239.95
TOSHIBA P1340 DOT MATRIX *Letter qual., draft, graphics*	$529.95
OKIDATA 192P DOT MATRIX *160 cps, 80-column, bi-directional*	$349.95
EPSON SPECTRUM LX-80 MATRIX *80-column, 100 cps, NEW MODEL*	$219.95
OKIDATA OKIMATE-10 4-COLOR *C64 interface included*	$179.95
EPSON FX-185 DOT MATRIX *Friction tractor, 136 columns, graphics*	$499.95

'TOP 10' APPLE SOFTWARE

GATO Spectrum Holobyte *Submarine simulation (SHP GATO)*	$24.95
FLIGHT SIMULATOR II Sub Logic *Flight simulation package (SLP FLIGHTII)*	$29.95
PFS: FILE Software Pub. *Data Manager (SOP PFS-FILE)*	$74.95
BANK STREET WRITER Broderbund *Word Processor for IIC (BRP BANKIIC)*	$39.95
COMPUTER BASEBALL Micro League *Great simulation w/stats (MLP BASEBALL)*	$24.95
PRINTSHOP Broderbund *Create printing fonts (BRP PRINTSHOP)*	$34.95
MUSIC CONSTRUCTION SET Elect.Arts *Write your own music (EAP MUSIC-D)*	$24.95
TYPING TUTOR III Simon & Schuster *Learn touch typing (KSP TYPINGIII)*	$29.95
F-15 STRIKE EAGLE Microprose *Air battle simulation (MAP F15)*	$24.95
KARATEKA Broderbund *Martial arts action (BRP KARATEKA)*	$24.95

'TOP 10' COMMODORE 64 SOFTWARE

FASTLOAD Epyx *Disk utility (AU6 FASTLOAD)*	$24.95
SUPER HUEY Cosmi *Helicopter simulation (CO6 SUPERHUEY)*	$14.95
BASEBALL Micro League *Real players & Stats (MLE BASEBALL)*	$24.95
PRINTSHOP Broderbund *Create print fonts (BR6 PRINTSHOP-D)*	$29.95
GRAPHICS LIBRARY Broderbund *100 fonts f/printshop (BR6 GRAPHICS-D)*	$14.95
SUMMER GAMES II Epyx *Olympic events plus (AU6 SUMMERII-D)*	$24.95
INDIANA JONES Mindscape *The lost kingdom (MD6 INDIANA-D)*	$19.95
PAPERCLIP Batteries Included *With spellcheck (BI6 PAPERCLIP)*	$79.95
MACH 5 Access *Disk drive utility (AS6 MACH5)*	$24.95
FLIGHT SIMULATOR II Sub Logic *Flight simulation (SL6 FLIGHT-II)*	$34.95

'TOP 10' ATARI SOFTWARE

ATARIWRITER Atari *ROM Word processor (ATA RX8036)*	$34.95
SYNFILE PLUS Synapse *Database manager merge w/ SYN series (SYN SYNFILE-D)*	$29.95
F-15 STRIKE EAGLE Microprose *Airborn adventure (MPA F15-D)*	$24.95
RESCUE ON FRACTELUS Epyx *Graphic adventure (AUS FRACTELUS)*	$24.95
SARGON II CHESS Hayden *Play friends or computer (HAA SARGONII-D)*	$15.95
BALL BLAZER Epyx *High-speed, two player (AUS BALLBLAZE)*	$24.95
PAPERCLIP Batteries Included *Word processing pack (BIA PAPERCLIP-D)*	$44.95
GHOSTBUSTERS Activision *Movie hit arcade game (ACS GHOSTBUST-D)*	$19.95
HOMEPAK Batteries Included *Word processor, filer, T.C. integrated (BIA HOMEPAK)*	$39.95
COMPUTER BASEBALL Micro League *Graphics, players, stats (MLA BASEBALL-D)*	$24.95

'TOP 10' IBM COMPATIBLE SOFTWARE

LOTUS 1-2-3 Lotus Development *The integrated package (LOI 123)*	$279.95
FLIGHT SIMULATOR Microsoft *Simulation, great graphics (MSI FLIGHT)*	$29.95
TURBO PASCAL Borland *Great Pascal compiler (BOI TURBOPASCAL)*	$39.95
TYPING TUTOR III Simon & Schuster *Learn touch typing (KSI TYPINGIII)*	$29.95
HITCHHIKERS GUIDE Infocom *Text adventure game (INI HITCH)*	$29.95
KINGS QUEST Sierra *Fantasy adventure (ONI KINGS)*	$24.95
dBASE III Ashton-Tate *Advanced database system (ASI DBASEIII)*	$379.95
SYMPHONY Lotus Development *1-2-3 and more, windows (LOI SYMPHONY)*	$399.95
FRAMEWORK Ashton-Tate *Integrated business package (ASI FRAMEWORK)*	$349.95
MATH BLASTER Davidson Assoc. *Math tutorial (DVI MATHBLASTER)*	$29.95

'TOP 10' MACINTOSH SOFTWARE

JAZZ from **Lotus** *Development*

•New powerful versatile integrated software for Macintosh **$379.95**

FACTFINDER Forethought *Organizer (FTM FACTFINDER)*	$119.95
ART GRABBER BODYSHOP Hayden *Creative graphics (HOM BODYSHOP)*	$34.95
ENSEMBLE Hayden *Integrated package (HOM ENSEMBLE)*	$169.95
FILEVISION Telos *Creative database mgt. (TEL FILEVISION)*	$129.95
HITCHHIKER'S GUIDE Infocom *Strategy adventure (INM HITCHIKERS)*	$29.95
THINK TANK Living Videotext *Idea processor (LVM THINKTANK)*	$84.95
MUSIC WORKS Hayden *Music construction (HOM MUSICWORKS)*	$49.95
TYPING TUTOR III Simon & Schuster *Touch typing (KSM TYPINGIII)*	$34.95
MAC VISION Koala Technologies *Digitizer (KUM MACVISION)*	$199.95

'TOP 10' COMPUTER ACCESSORIES

discwasher®
SPIKEMASTER SURGE SUPRESSOR

•4 outlet multimode multistage protection •Rocker on/off illuminated switch **$49.95**

SURGE SUPRESSOR MidlandRoss *6 outlet, circuit breaker, extension cord*	$24.95
DISK DRIVE CLEANING KIT Head *5.25" drives, reuseable*	$14.95
COMPUTER PAPER *Perforated 1000 sheets, 8½"x11"*	$14.95
PARALLEL PRINTER INTERFACE Cardco G-WIZ *For C64 with graphics*	$59.95
DISKETTE STORAGE Ring King *Smoked black plastic, holds 50 5.25"*	$14.95
80-COLUMN PRINTER STAND Ring King *Universal smoked plastic*	$29.95
SM2400 SMART MODEM Hayes *1200/2400 baud, external*	$629.95
V1660 MODEM Commodore *Auto dial w/software*	$59.95
SMALL COMPUTER TABLE Bush *Walnut finish*	$79.95

Remember this is only a sampling of our stock of computers, software, peripherals and accessories. If what you need is not listed please call or write for prices and availability

ALL SOFTWARE DISK-BASED UNLESS OTHERWISE NOTED

NOTE: WE CARRY A FULL LINE OF AUDIO, VIDEO, AND COMPUTERS IN EVERY MAJOR BRAND. THIS LISTING IS JUST A SMALL SAMPLE IN OUR $10,000,000 INVENTORY. PLEASE CALL US TOLL FREE OR WRITE TO US FOR PRICES ON ITEMS NOT LISTED IN THIS AD.

ALL PRODUCTS CARRY U.S. WARRANTIES
ALL DEFECTIVE PRODUCTS EXCHANGED WITHIN 30 DAYS

HOW TO ORDER BY MAIL: FOR PROMPT AND COURTEOUS SHIPMENT, SEND MONEY ORDER, CERTIFIED CHECK, CASHIER'S CHECK, MASTERCARD/VISA (include card number, interbank No. expiration date and signature.) **DO NOT SEND CASH.** PERSONAL AND BUSINESS CHECKS MUST CLEAR OUR BANK BEFORE PROCESSING. **$25 MINIMUM ORDER.** Shipping, Handling & Insurance Charge is 5% of Total Order with a **$3.95** minimum. WE SHIP TO CONTINENTAL U.S., ALASKA, HAWAII, PUERTO RICO AND CANADA ONLY. (Canadian Orders Add 15% Shipping, with a $9.95 minimum charge.) For shipments by air, please double these charges. **SORRY, NO C.O.D.'s.** NEW YORK RESIDENTS PLEASE ADD SALES TAX. **ALL MERCHANDISE SHIPPED BRAND NEW, FACTORY FRESH, AND 100% GUARANTEED. WE ARE NOT RESPONSIBLE FOR ANY TYPOGRAPHICAL ERRORS.**

CUT OUT AND MAIL

ALL NEW FOR 1985

J&R MUSIC WORLD

4 FREE CATALOGS

- AUDIO/VIDEO/ELECTRONICS CATALOG
- COMPUTER CATALOG
- RECORD, CASSETTE AND CD CATALOG
- VIDEO MOVIE CATALOG

IF YOU HAVE NOT ALREADY RECEIVED OUR LATEST CATALOGS PLEASE FILL OUT COUPON AND MAIL OR CALL TOLL FREE 1-800-426-6027 AND WE'LL SEND THEM OUT IMMEDIATELY

PLEASE CHECK WHICH CATALOGS YOU WANT

□ AUDIO VIDEO □ COMPUTER □ RECORD, CASSETTE □ VIDEO MOVIE

23 PARK ROW, DEPT. C11, NYC, NY 10038

NAME: ______________________

ADDRESS: ______________________

CITY: ____________ STATE: ______ ZIP: ______

PLEASE SEND $1.00 FOR SHIPPING AND HANDLING

CUT OUT AND MAIL

Advertisers Index

Reader Service Number/Advertiser		Page
102	Abacus Software	120
103	Abacus Software	121
104	Activision	11
105	American TV	144
106	Artificial Intelligence Research Group	143
	Atari Corp.	9
107	Basix	60
108	Basix	64
	Batteries Included	41
109	Blackship Computer Supply	159
110	Bowater Computer Forms, Inc.	21
111	Cardco, Inc.	IBC
	CBS Software	86
	C.O.M.B. Co.	149
	C.O.M.B. Co.	155
	Commodore	BC
112	Compucat	159
113	CompuServe	24–25
	ComputAbility	93
114	Computer Centers of America	42–43
115	Computer Direct	83–85
116	Computer Mail Order	112–113
117	Davidson & Associates	91
	Disk World!, Inc.	151
	Disk World!, Inc.	153
	Disk World!, Inc.	154
118	Electronic Arts	IFC–1
119	Electronic Arts	88
120	Elek-Tek, Inc.	97
121	EPYX	33
122	EPYX	35
123	EPYX	37
	Frontrunner Computer Industries	144
124	Inmac	87
125	J&R Music World	159
126	Jason-Ranheim	107
127	Jason-Ranheim	150
	JS&A	157
	Lyco Computer	50–51
128	MegaSoft, Ltd.	49
129	Micro Prose Simulation Software	53
	Micro World Computers, Inc.	152
130	Mimic Systems, Inc.	29

Reader Service Number/Advertiser		Page
131	Mindscape, Inc.	12–13
132	Mindscape, Inc.	15
133	Nibble Notch Computer Products	156
	North Hills Corp.	159
	North Hills Corp.	159
	NRI Schools	89
134	Okidata	2–3
135	Pacific Exchanges	159
136	Professor Jones	152
137	Professor Jones	156
138	Protecto	82
139	Precision Data Products	159
140	Quinsept, Inc.	143
141	Scarborough Systems, Inc.	7
142	The Source	81
	Spinnaker	38–39
	Spinnaker	40
143	Strategic Simulations, Inc.	63
144	Timeworks, Inc.	4
145	subLOGIC Corporation	45
146	T & D Electronics	152
147	Starpoint Software	47
148	USA★Flex	107
149	White House Computer	156
	The Wholesale Outlet	110
150	World Trade Wholesale Outlet	104–105

Classified Ads	158
COMPUTE! Back Issues	94–95
COMPUTE! Books' New Fall Releases	61
COMPUTE!'s Commodore 64 Books Special Offer	57
COMPUTE!'s Telecomputing Books	23
COMPUTE! Subscription	17
MacOffice & Second Book of Apple	55
Machine Language for Beginners & Second Book of Machine Language	59

This holiday season put your fellow PC users on line with

COMPUTE!

The magazine for Home, Education and Recreational Computing.

SPECIAL HOLIDAY GIFT SAVINGS

For each gift you give, save 25% off the regular $24 subscription rate. You pay only $18 per gift subscription.

A SEASONAL CARD WILL ANNOUNCE EACH GIFT

YOUR NAME ____________________

ADDRESS ____________ APT # ______

CITY/STATE/ZIP ____________________

A SPECIAL OFFER FOR YOU

I HAVE GIVEN AT LEAST 1 GIFT PLEASE
□ ENTER OR □ EXTEND MY SUBSCRIPTION AT THE SAME LOW RATE
□ PAYMENT ENCLOSED
□ PLEASE BILL ME

1. TO: ____________________

ADDRESS ____________ APT # ______

CITY/STATE/ZIP ____________________

GIFT CARD TO READ ____________________

2. TO: ____________________

ADDRESS ____________ APT # ______

CITY/STATE/ZIP ____________________

GIFT CARD TO READ ____________________

S

PRICES GOOD IN USA ONLY. FOR CANADA AND FOREIGN PLEASE ADD $6 PER SUBSCRIPTION M1020

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 7478 DES MOINES, IOWA

POSTAGE WILL BE PAID BY ADDRESSEE

COMPUTE!
PO BOX 10954
DES MOINES, IOWA 50347

COMPUTE!'s FREE Reader Information Service

Use these cards to request FREE information about the products advertised in this issue. Clearly print or type your full name and address. Only one card should be used per person. Circle the numbers that correspond to the key number appearing in the advertisers index.

Send in the card and the advertisers will receive your inquiry. Although every effort is made to insure that only advertisers wishing to provide product information have reader service numbers, COMPUTE! cannot be responsible if advertisers do not provide literature to readers.

Please use these cards *only* for subscribing or for requesting product information. Editorial and customer service inquiries should be addressed to: COMPUTE!, P.O. Box 5406, Greensboro, NC 27403. Check the expiration date on the card to insure proper handling.

Use these cards and this address only for COMPUTE!'s Reader Information Service. Do not send with payment in any form.

COMPUTE!

101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117
118	119	120	121	122	123	124	125	126	127	128	129	130	131	132	133	134
135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150	151
152	153	154	155	156	157	158	159	160	161	162	163	164	165	166	167	168
169	170	171	172	173	174	175	176	177	178	179	180	181	182	183	184	185
186	187	188	189	190	191	192	193	194	195	196	197	198	199	200	201	202
203	204	205	206	207	208	209	210	211	212	213	214	215	216	217	218	219
220	221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	236
237	238	239	240	241	242	243	244	245	246	247	248	249	250	251	252	253

Circle 101 for a one year new U.S. subscription to COMPUTE!: you will be billed for $24.

Please let us know. Do you

own:		plan to buy:
□ 270	Apple ______	□ 271
□ 272	Atari ______	□ 273
□ 274	Commodore ______	□ 275
□ 276	IBM ______	□ 277
□ 278	TI-99/4A ______	□ 279
□ 280	Other ______ (specify model)	□ 281

Please print or type name and address.
Limit one card per person.

Name ______

Address ______

City ______

State/Province ______ Zip ______

Country ______

Please include zip code. Expiration date 12/31/85 C01185

SUBSCRIBE TO COMPUTE!

For Fastest Service,
Call Our **Toll-Free**
US Order Line
800-334-0868
In NC call 919-275-9809

My Computer Is:

01 □ Apple 02 □ Atari 03 □ Commodore 64
04 □ VIC-20 05 □ IBM 06 □ TI-99/4A
99 □ Other______ □ Don't yet have one.

□ $24.00 One Year US Subscription
□ $45.00 Two Year US Subscription

(Readers outside of the US, please see our foreign readers subscription card or inquire for rates).

Name ______

Address ______

City ______ State ______ Zip ______

□ Payment Enclosed □ Bill me
Charge my: □ VISA □ MasterCard □ American Express
Account No. ______ Expires ____ / ____

Your subscription will begin with the next available issue. Please allow 4-6 weeks for delivery of first issue. Subscription prices subject to change at any time.

The COMPUTE! subscriber list is made available to carefully screened organizations with a product or service which may be of interest to our readers. If you prefer not to receive such mailings, please check this box □.

J1098

Place
Stamp
Here

COMPUTE! Reader Service
P.O. Box 2141
Radnor, PA 19089

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 7478 DES MOINES, IOWA

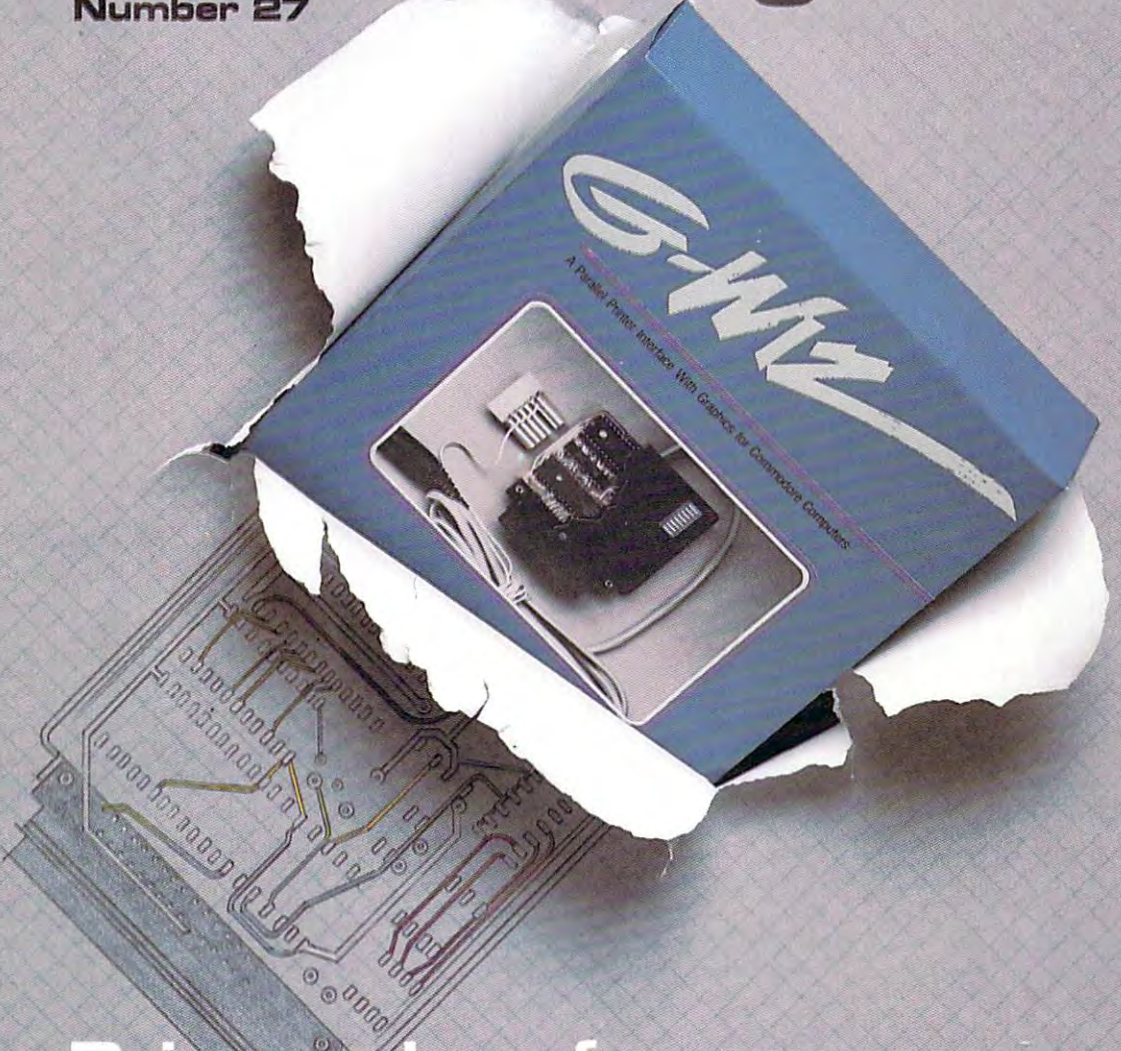
POSTAGE WILL BE PAID BY ADDRESSEE

COMPUTE!
PO BOX 10954
DES MOINES, IOWA 50347

Technical

Breakthrough

Number 27


G-Wiz
A Parallel Printer Interface With Graphics for Commodore Computers

Price and performance

no other printer interface can touch!

Only CARDCO would dare improve on its own best seller (the +G has far out-sold any other printer interface, and has set the industry standard by which competitors are judged). The G-Wiz is even better — and costs 30% less.* Our 27th major innovation in Commodore accessories has all the +G's features, and more...

Built-In Buffer for More Speed

The G-Wiz buffer dumps high-resolution screens up to 18 times faster than competitive interfaces without buffers. No more waiting! A complex 50-minute printout now takes just 2.5 minutes with the G-Wiz.

Exclusive! Aspect Ratio Selection

Only the G-Wiz matches graphics characters to standard characters on Okidata, Epson, Star Micronics, Prowriter, Smith Corona, C-ITOH, Gorilla Banana, and many other dot matrix printers. Now you can perfectly align high resolution graphics characters within text blocks, or in columns.

CARDCO excellence triumphs again! The G-Wiz is the "best bang for the buck" on the printer interface market today — and it's backed by CARDCO'S exclusive lifetime warranty! G-Wiz: another distant target for the competition to shoot at.

CARDCO, Inc. 300 S. Topeka/Wichita, KS 67202

* Actual price may vary according to dealer and region.

The Wizards from the Land of Oz Have Done It Again!

HOW TO EVOLVE TO A HIGHER INTELLIGENCE.

THE COMMODORE 128.

The first step is buying the Commodore 128™ Personal Computer. The smartest computer available for the price. It's like getting three computers for less than one usually costs. You can run CP/M® business software, the new programs written for the 128, and over 3,000 Commodore 64® programs. You start out with more software than most machines give you after years on the market.

THE COMMODORE 128 WORKS FASTER.

To run all that software and run it faster, you'll want the 1571 Disk Drive. You can't find a faster drive at the price. It transfers nearly 1,000 words a second (5200 cps), so you can load most programs instantly.

THE COMMODORE 128 GETS SMARTER.

Now try improving your memory. Plug in our 1750 RAM Expansion Module and your 128 moves up to a powerful 512K. That's enough to handle just about anything you can dish out, from complicated business forecasting to giant data bases.

THE COMMODORE 128 LEARNS TO COMMUNICATE.

There's no real intelligence without the ability to communicate. So you'll want our 1670 Modem/1200. It puts you in touch with a new world of shopping, banking, communications and information over your telephone line. And it operates at a lightning-fast 1200 baud to save on your phone bill.

THE COMMODORE 128 LEARNS TO WRITE.

Looking good in print could be your next move with the MPS 1000 Printer. It's a new dot matrix printer designed to make the most of the 128's speed and high-resolution graphics. The MPS turns out about 1200 words a minute (100 cps) of draft-quality printing, or gives you near-letter-quality at about 240 words a minute (20 cps).

THE COMMODORE 128 IMPROVES YOUR VISION.

Brains aren't enough without good looks, so improve your vision with Commodore's new 1902 RGB Color Monitor. The high-resolution screen gives you a sharper image and better color than your standard TV, so you can really appreciate the 128's great graphics.

All these evolutionary steps ahead won't set you back when it comes to paying for them. Additions to your Commodore 128 are available at a store near you and are as affordable as the 128 itself. We think that's a smart way to help you build a computer system.

COMMODORE 128 PERSONAL COMPUTER
A Higher Intelligence