Telecommuting: Dawn Of The Electronic Cottage

COMPUTE!'s GAZETTE

$2.50
December 1983
Issue 6 Vol. 1, No. 6
63380 $3.25 in Canada

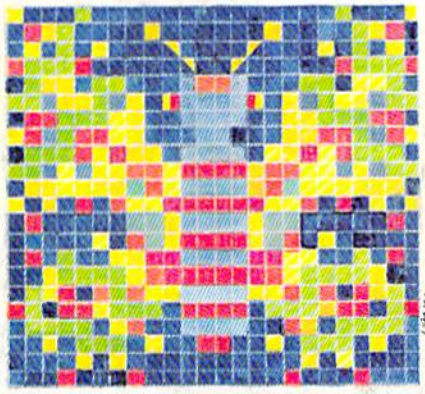
For Owners And Users Of **Commodore VIC-20**™ And **64**™ Personal Computers

SPIKE

Arcade-Action Game For Commodore 64

Written entirely in machine language, Spike is an outstanding arcade-style game with stunning high-resolution graphics — one of the best games we've ever published. Only the skillful can evade the random power spikes and escape the Grid.

Sprites Made Easy For Commodore 64



A simple program to add sprite commands to Commodore BASIC. Examples show how you can animate shapes on the screen in your own programs with a minimum of tricky PEEKs and POKEs.

Educational Games: A Kid's View

A teenager speaks out on what youngsters like to see in educational computer games — and he includes his own game for the VIC-20 and Commodore 64 to show exactly what he means.

A SURVIVAL GUIDE FOR BEGINNERS

Lost in the woods without a compass? Here's a complete guide to finding help through user groups, computer classes, books and magazines, and your fellow computerists.

Also In This Issue

0 74470 63380 12

Budget Planner

Charles B. Silbergleith

This home budget program allows you to keep track of various household expenses and calculate totals quickly and easily. The same program works on either a Commodore 64 or VIC-20 (at least 8K memory expansion required).

In the dark days prior to automation, I would plan my budget by writing all my month's expenses on a sheet of paper, adding items, and adjusting amounts as I received a bill.

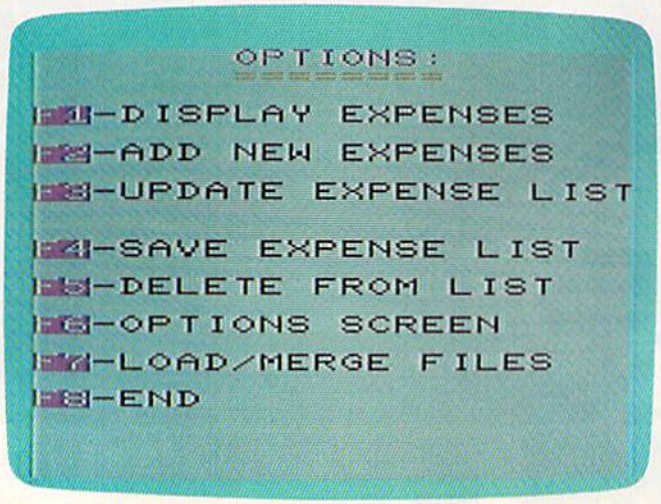
This process worked very well except for the number of revisions necessary for revolving credit accounts such as credit cards. Every time one of the item amounts changed, the grand total changed and needed to be recalculated. That was messy.

I decided to write a program which allowed me to make a list of my monthly expenses, to change amounts, and which provided a grand total of all items. I also wanted the program to save this list to tape and recall it.

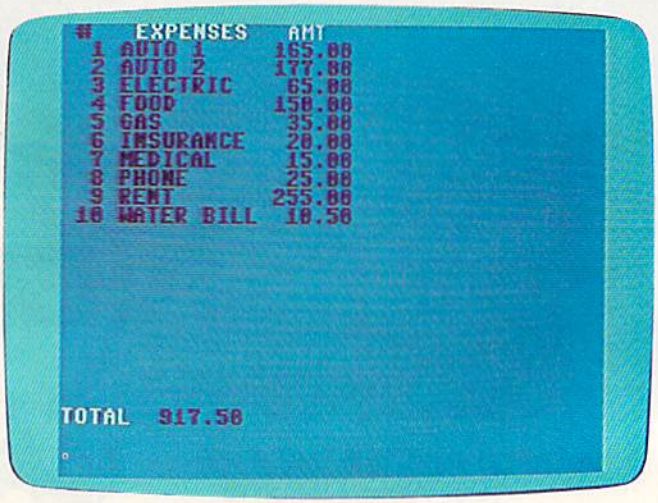
What was produced was a program that allowed me to maintain a list of expense items, add new items, change amounts, delete items, and it would quickly sort and sum all the amounts. This was useful in seeing whether new expenses could be incurred (could I really afford that new disk drive or not?), or whether bill consolidation would help.

Program Operation

First here are some basic characteristics of the program before I discuss how to use it. The list allows



The main menu in "Budget Planner" (VIC version).



A typical expense list made with "Budget Planner" (64 version).

for entries of ten characters (maximum) per item and amounts of up to 9999.99. The list will be sorted, a total calculated over all item amounts, and the options menu displayed at the end of an add, update, or delete modification to the list. The sort is done by item name. You will be repeatedly prompted for the next add, update, or delete to the list until you type *END to one of the prompts for input. In fact, any function will terminate whenever you respond with an *END to a prompt.

Since the program was written for a VIC-20 (and converted for the Commodore 64 also), it uses the special function keys f1 through f8. Described below are the functions:

f1 **Display Expense List.** This function displays the list and a total of all item amounts at the bottom of the screen. Pressing f1 will display the next 20 items, and the cursor up and down keys scroll the list vertically. All function keys are available.

f2 **Add New Expense To The List.** This allows you to add a new item to the list. The program will not check for duplicates. However, it's simple enough to change or delete an item if you mistakenly duplicate one. Names are up to ten characters, and amounts should not be larger than + or − 9999.99. These restrictions are used to prevent the screen display from overlapping, wrapping around, or otherwise messing up on the 22-column VIC. Type *END to return to the menu screen.

f3 **Expense List Update.** The screen lists a number next to each item. This number is the item's index. Use this number for the ITEM # prompt. The item will be displayed and a new name or amount may be entered replacing the old data. Pressing the RETURN key without data when prompted for an ITEM NAME or AMT will leave the current data intact. Again, type *END to return to the menu.

f4 **Save The List On Tape.** The program asks for a FILE NAME. This should be any name that follows normal Commodore file naming conventions. This is the filename SAVEd on tape. Remember it.

f5 **Delete Items From The List.** The START AT and END AT prompts allow a block of items to be deleted by putting the starting and ending index numbers in the appropriate places. Leaving out the ending index will delete only the starting index number's item. Type *END when prompted for the starting index number to return to the main menu.

f6 **Display The Option Menu.** Function keys and their associated functions are displayed. See program lines 6030-6100 for details.

f7 **Load Or Merge A List.** A previously saved list can be loaded into memory or a list on tape can be merged with a list in memory. For the merge, an item on tape is compared to the items in memory, and if the item names match, their amounts are averaged together and replace the previous amount. If the item doesn't match, the item is added to the list.

f8 **End Of Program.** This function allows you to first save the list before actually ending the program—handy if you've forgotten to save the list before.

Technical Notes

The program is written using the modular concept of structured programming. This means that the program is written in order to isolate its various tasks. Common routines are separate from the routines that use them and are accessed by GOSUB statements.

The main routine (lines 200-299) calls various subfunctions at the user's request. A request to display the list (f1) calls a subroutine at lines 1000-1999; update (f3) calls lines 3000-3999, etc. Notice that each function key corresponds to a range of 1000 line numbers—f1 is lines 1000-1999; f2 is lines 2000-2999; f3 is lines 3000-3999, etc. This makes it easier to remember where things are in the program.

In addition, two utilities are included as separate modules for use by any function. These are the bubble sort, lines 500-599, and an accumulator, lines 300-399.

GOTO statements are kept to a minimum and are used only for branching within subroutines. While certain advocates of structured programming insist on GOTO-less code, I find it sometimes more cumbersome to eliminate all of them than to use a few. Again, the word to remember is *few*.

One last note. The variable SZ (line 20) controls the number of items that can be listed. Naturally, the more items on the list, the more memory is required. Since the computer will consume more memory as needed when the program runs, it is possible to make this variable too large and run out of room while working with the program. As an exercise, I suggest you add a function which will display the amount of memory left. Use the ? key to invoke it. I think you'll find it fairly easy to do given the way the program is organized.

See program listing on page 220.

```
1012 PRINT" {RED}BUT, SOME {PUR}PURPLE
     {RED} ONES MIGHT TURN BACK" :rem 148
1014 PRINT"{DOWN}TO {BLK}BLACK{RED}!"
                                 :rem 117
1015 PRINT"{2 DOWN} {BLU}EACH NUMBER YOU
     {SPACE}ENTER WILL CHANGE THE"
                                 :rem 237
1017 PRINT"{DOWN}COLORS IN ITS OWN WAY."
                                  :rem 23
1025 PRINT"{2 DOWN} {GRN}TRY TO CHANGE AL
     L THE BLOCKS TO {PUR}PURPLE{GRN}
                                 :rem 172
1030 PRINT"{DOWN}IN AS FEW TRIES AS YOU C
     AN.":GOSUB2000:GOTO6        :rem 240
2000 REM GET KEYPRESS            :rem 252
2001 PRINT"{HOME}{23 DOWN}"TAB(14)"{RVS}
     {BLU}TOUCH A KEY{OFF}";     :rem 170
2002 POKE198,0                   :rem 241
2005 GETA$:IFA$=""THEN2005       :rem 177
2010 RETURN                      :rem 163
3000 PRINT"{7 DOWN}"TAB(15)"{WHT}{A}{C
     }{R}{C}{R}{C}{S}"           :rem 237
3010 PRINTTAB(15)"B B B B"       :rem 43
3020 PRINTTAB(15)"{Q}{C+C+C}{W}"
                                 :rem 129
3030 PRINTTAB(15)"B B B B"       :rem 45
3040 PRINTTAB(15)"{Q}{C+C+C}{W}"
                                 :rem 131
3050 PRINTTAB(15)"B B B B"       :rem 47
3060 PRINTTAB(15)"{Z}{C}{E}{C}{E}{C}{X}
     "                           :rem 61
3070 RETURN                      :rem 170
```

Budget Planner

(Article on page 108.)

BEFORE TYPING...

Before typing in programs, please refer to "How To Type COMPUTE!'s Gazette Programs," "A Beginner's Guide To Typing In Programs," and "The Automatic Proofreader" that appear before the Program Listings.

```
10 REM DEF VARIABLES             :rem 173
20 SZ=100:I=-19                   :rem 52
30 R$=CHR$(13):TA=0                :rem 8
40 DIM A$(SZ),AE(SZ)              :rem 81
50 DEFFNRN(X)=INT(X*100+.5)/100  :rem 235
200 REM MAINROUTINE              :rem 193
210 GOSUB6000                    :rem 217
220 Z$="":GETZ$:IFZ$=""THENGOTO220
                                 :rem 239
230 IFZ$=CHR$(133)THENI=I+20:GOSUB1000
                                 :rem 206
235 IFZ$=CHR$(134)THENGOSUB3000   :rem 64
240 IFZ$=CHR$(135)THENGOSUB5000   :rem 63
245 IFZ$=CHR$(136)THENGOSUB7000   :rem 71
250 IFZ$=CHR$(137)THENGOSUB2000   :rem 63
255 IFZ$=CHR$(138)THENGOSUB4000   :rem 71
260 IFZ$=CHR$(139)THENGOSUB6000   :rem 70
265 IFZ$=CHR$(140)THENGOSUB8000   :rem 69
270 IFZ$=CHR$(17)THENI=I-1:GOSUB1000
                                 :rem 116
275 IFZ$=CHR$(145)THENI=I+1:GOSUB1000
                                 :rem 169
299 GOTO220                      :rem 113
300 REM ACCUM TOTALS             :rem 183
310 TA=0                         :rem 150
320 FOR J=1TOMX                  :rem 124
330 TA=TA+AE(J)                   :rem 73
340 NEXTJ                         :rem 32
399 RETURN                       :rem 133
400 REM LOAD FILES                :rem 11
410 INPUT"FILE NAME";F$           :rem 79
420 IFF$="*END"THENGOSUB6000:RETURN
                                 :rem 160
450 OPEN1,1,0,F$                  :rem 75
455 PRINT"{RVS}{GRN}FOUND{OFF}{BLK}";F$
                                 :rem 226
460 INPUT#1,MX                    :rem 79
470 FORJ=1TOMX                   :rem 130
480 INPUT#1,Y,A$(J),AE(J)        :rem 126
490 NEXTJ                         :rem 38
495 CLOSE1                        :rem 73
499 RETURN                       :rem 134
500 REM SORT BY NAME             :rem 125
505 IFMX=1THENGOTO599             :rem 75
510 PRINT"{2 DOWN}{5 RIGHT}{RVS}SORTING
    {OFF}"                       :rem 228
520 FORJ=1TOMX-1                 :rem 220
530 FORK=J+1TOMX                 :rem 245
540 IFA$(K)>A$(J)THENGOTO590     :rem 109
550 SM$=A$(K):SM=AE(K)           :rem 213
560 A$(K)=A$(J):AE(K)=AE(J)      :rem 147
570 A$(J)=SM$:AE(J)=SM           :rem 213
590 NEXTK                         :rem 40
595 NEXTJ                         :rem 44
599 RETURN                       :rem 135
1000 REM DISPLAY                 :rem 187
1010 IF(I<1)OR(I>MX)THENI=1       :rem 92
1020 PRINT"{CLR} #"TAB(5)"{CYN}EXPENSES
     {BLK}"TAB(16)"{PUR}AMT{BLK}":rem 102
1030 FORJ=ITOI+19                :rem 252
1040 IFJ>MXTHENPRINT" ":GOTO1080 :rem 189
1050 PR$=STR$(AE(J)+.001):PR$=MID$(PR$,2,
     (LEN(PR$)-2))               :rem 196
1060 IFAE(J)=0THENPR$="0.00"      :rem 24
1065 J$=MID$(STR$(J),2)          :rem 220
1070 PRINTTAB(3-LEN(J$))J$;TAB(4)A$(J)TAB
     (21-LEN(PR$))PR$            :rem 244
1080 NEXTJ                        :rem 82
1090 TA$=STR$(TA+.001)           :rem 173
1100 TA$=LEFT$(TA$,LEN(TA$)-1)   :rem 132
1110 IFTA=0THENTA$="0.00"        :rem 123
1120 PRINT"{CYN}TOTAL {BLK}"TA$    :rem 1
1999 RETURN                      :rem 188
2000 REM ADD NEW                  :rem 89
2010 R=MX+1:N$="":E1$=""         :rem 213
2020 PRINT"{CLR}{3 RIGHT}ADD NEW EXPENSES
     "                           :rem 157
2030 PRINT"{DOWN}{12 RIGHT}ITEM #";R
                                 :rem 226
2040 INPUT"{DOWN}ITEM NAME ";N$  :rem 168
2050 IFN$="*END"THENGOTO2999     :rem 143
2055 IFLEN(N$)>10THENN$=LEFT$(N$,10)
                                  :rem 25
2060 A$(R)=N$                    :rem 127
2070 INPUT"{DOWN}ITEM AMT{2 SPACES}";E1$
                                 :rem 148
2080 IFE1$="*END"THENGOTO2999    :rem 186
2085 IFVAL(E1$)=0THENAE(R)=0:GOTO2100
                                 :rem 148
2090 AE(R)=FNRN(VAL(E1$))        :rem 132
2095 IFAE(R)>9999.99THENAE(R)=9999.99
                                  :rem 99
2100 MX=MX+1                     :rem 166
2110 GOTO2010                    :rem 192
2200 MX=MX+1                     :rem 167
```

```
2999 GOSUB500:GOSUB300:GOSUB6000:RETURN
                                    :rem 217
3000 REM UPDATE                     :rem 106
3010 PRINT "{CLR}{BLU}EXPENSE ";"{RVS}UPD
     ATE{OFF}{BLK}"                 :rem 213
3020 INPUT"{DOWN}ITEM # ";P1$       :rem 220
3025 IFP1$="*END"THENGOTO3999       :rem 198
3026 IF(VAL(P1$)=0)OR(VAL(P1$)<1)THENPRIN
     T"{2 DOWN}{4 RIGHT}{PUR}{RVS}INPUT E
     RROR{OFF}{BLK}":GOTO3020       :rem 225
3027 P=INT(VAL(P1$))                :rem 110
3030 N$="":E1$=""                    :rem 14
3040 IFP>SZTHENPRINT"MAX EXCEEDED":P=SZ:M
     X=P                            :rem 142
3050 IFP>MXTHENMX=P                 :rem 235
3060 PR$=STR$(AE(P)+.001):PR$=MID$(PR$,2,
     (LEN(PR$)-2))                  :rem 205
3065 IFAE(P)=0THENPR$="0.00"         :rem 37
3070 PRINTP;TAB(4)A$(P)TAB(21-LEN(PR$))PR
     $                              :rem 184
3080 INPUT"{DOWN}ITEM NAME";N$      :rem 173
3090 IFN$="*END"THENGOTO3999        :rem 149
3100 IFN$<>""THENA$(P)=N$           :rem 103
3105 IFLEN(A$(P))>10THENA$(P)=LEFT$(A$(P)
     ,10)                           :rem 210
3110 INPUT"AMT ";E1$                 :rem 80
3120 IFE1$="*END"THENGOTO3999       :rem 183
3125 IFE1$=""GOTO3010               :rem 114
3130 IF(VAL(E1$)=0)AND(E1$<>"0")THENPRINT
     "{2 DOWN}{3 RIGHT}{RVS}{PUR}INPUT ER
     ROR{OFF}{BLK}":GOTO3110        :rem 41
3135 IFVAL(E1$)=0THENAE(P)=0:GOTO3800
                                    :rem 151
3140 AE(P)=FNRN(VAL(E1$))           :rem 127
3150 IFAE(P)>9999.99THENAE(P)=9999.99
                                    :rem 88
3800 GOTO3010                       :rem 200
3999 GOSUB500:GOSUB300:GOSUB6000:RETURN
                                    :rem 218
4000 REM SAVE FILE                  :rem 247
4010 PRINT"{CLR}{3 RIGHT}SAVE EXPENSE LIS
     T"                               :rem 3
4020 INPUT"{2 DOWN}FILE NAME";F$    :rem 162
4030 IFF$="*END"THENGOSUB6000:RETURN
                                    :rem 209
4050 OPEN1,1,1,F$                   :rem 124
4060 PRINT#1,MX                     :rem 124
4070 FORJ=1TOMX                     :rem 178
4080 PRINT#1,J;R$;A$(J)R$;AE(J);R$
                                    :rem 146
4090 NEXTJ                           :rem 86
4100 CLOSE1                         :rem 108
4999 GOSUB6000:RETURN                :rem 63

5000 REM DELETE                      :rem 92
5005 DT=0:TM=0                       :rem 23
5010 PRINT"{CLR}{8 RIGHT}DELETE"    :rem 197
5020 S1$=""                         :rem 240
5030 INPUT"{2 DOWN}START AT";S1$    :rem 196
5040 IFS1$="*END"THENGOTO5900       :rem 184
5050 DS=INT(VAL(S1$))               :rem 182
5060 S1$=""                         :rem 244
5070 IFDS=0THENPRINT"{DOWN}{6 RIGHT}{RVS}
     {PUR}INPUT ERROR{OFF}{BLK}":GOTO5020
                                    :rem 66
5080 S1$=""                         :rem 246
5090 INPUT"{2 DOWN}END AT";S1$       :rem 19
5100 IFS1$="*END"THENGOTO5900       :rem 181
5110 IFS1$=""ORS1$="0"THENDE=0:GOTO5200
                                    :rem 216
5120 DE=INT(VAL(S1$))               :rem 166
5125 IFDE>MXTHENDE=MX               :rem 98
5130 IFDE=>DSTHENGOTO5200           :rem 34
5135 PRINT"{2 DOWN}{2 RIGHT}{RVS}{PUR}0 O
     R NUMBER GREATER"              :rem 77
5140 PRINT"{2 DOWN}{2 RIGHT}THAN{OFF}
     {RED}";DE;"{RVS}{PUR}REQUIRED"
                                    :rem 34
5150 GOTO5080                       :rem 209
5200 IFDE=0THENDE=DS                :rem 216
5205 TM=DE-DS+1                      :rem 83
5207 DT=DT+TM                         :rem 7
5210 FORJ=DSTODE                    :rem 249
5220 A$(J)="[9 B]":AE(J)=0         :rem 201
5230 NEXTJ                           :rem 83
5240 GOTO5010                       :rem 202
5900 GOSUB500                       :rem 227
5910 MX=MX-DT                        :rem 27
5999 GOSUB300:GOSUB6000:RETURN      :rem 141
6000 REM OPTIONS MENU                :rem 11
6010 PRINT"{CLR}{7 RIGHT}{PUR}OPTIONS:
     {BLK}"                         :rem 136
6020 PRINT"{7 RIGHT}{YEL}========{BLK}"
                                    :rem 122
6030 PRINT"{DOWN}{RVS}{PUR}F1{OFF}{BLK}-D
     ISPLAY EXPENSES"               :rem 160
6040 PRINT"{DOWN}{RVS}{PUR}F2{OFF}{BLK}-A
     DD NEW EXPENSES"                :rem 63
6050 PRINT"{DOWN}{RVS}{PUR}F3{OFF}{BLK}-U
     PDATE EXPENSE LIST"            :rem 58
6060 PRINT"{DOWN}{RVS}{PUR}F4{OFF}{BLK}-S
     AVE EXPENSE LIST"              :rem 168
6070 PRINT"{DOWN}{RVS}{PUR}F5{OFF}{BLK}-D
     ELETE FROM LIST"               :rem 74
6080 PRINT"{DOWN}{RVS}{PUR}F6{OFF}{BLK}-O
     PTIONS SCREEN"                 :rem 21

6090 PRINT"{DOWN}{RVS}{PUR}F7{OFF}{BLK}-L
     OAD/MERGE FILES"               :rem 93
6100 PRINT"{DOWN}{RVS}{PUR}F8{OFF}{BLK}-E
     ND"                            :rem 251
6999 RETURN                         :rem 193
7000 REM LOAD/MERGE                 :rem 106
7010 PRINT"{CLR}{6 RIGHT}LOAD/MERGE"
                                    :rem 153
7020 PRINT"{DOWN}{5 RIGHT}EXPENSE FILES"
                                    :rem 199
7030 INPUT"LOAD OR MERGE (L/M)";AN$
                                    :rem 214
7040 IFAN$="L"THENMX=0:GOSUB400:GOTO7999
                                    :rem 190
7050 IFAN$="*END"THENGOSUB6000:RETURN
                                     :rem 31
7060 IFAN$<>"M"GOTO7030              :rem 29
7070 PRINT"{DOWN}{4 RIGHT}MERGE"    :rem 148
7080 INPUT"{DOWN}FILE NAME";F$      :rem 154
7090 IFF$="*END"THENGOSUB6000:RETURN
                                    :rem 218
7120 OPEN1,1,0,F$                   :rem 124
7130 INPUT#1,T1                      :rem 96
7140 FORT2=1TOT1                    :rem 207
7150 INPUT#1,Y,T3$,T4               :rem 193
7160 FORJ=1TOMX                     :rem 181
7170 IFA$(J)=T3$THENAE(J)=INT(((AE(J)+T4)
     /2)*100)/100:T3$=""            :rem 199
7180 NEXTJ                           :rem 89
7190 IFT3$<>""THENMX=MX+1:A$(MX)=T3$:AE(M
     X)=T4                          :rem 211
7200 NEXT                             :rem 8
7210 CLOSE1                         :rem 113
7999 GOSUB500:GOSUB300:GOSUB6000:RETURN
                                    :rem 222
```

```
8000 REM END OF JOB                    :rem 243
8010 PRINT"{CLR}{4 RIGHT}END OF PROGRAM
     {2 DOWN}"                         :rem 71
8020 PRINT"WOULD YOU LIKE TO SAVE (Y/N)":
     INPUT AN$                         :rem 190
8030 IFAN$="*END"THENGOSUB6000:RETURN
                                       :rem 30
8040 IFAN$="N"THENGOTO8060             :rem 19
8050 GOSUB4000                         :rem 17
8060 PRINT"{CLR}THANK YOU"             :rem 165
8070 PRINT"{13 RIGHT}END"              :rem 240
8080 END                               :rem 167
```

Machine Language For Beginners

(Article on page 154.)

Program 1: VIC Version

```
12288  LDY   # 0
12290  LDA   # 6
12292  STA   37888 ,Y
12295  STA   38144 ,Y
12298  INY
12299  BNE         12292
12301  LDY   # 0
12303  LDA   # 224
12305  STA   4096 ,Y
12308  STA   4580 ,Y
12311  INY
12312  CPY   # 22
12314  BNE         12305
12316  RTS
```

Program 2: 64 Version

```
49152  LDY   # 0
49154  LDA   # 8
49156  STA   55296 ,Y
49159  STA   55552 ,Y
49162  STA   55808 ,Y
49165  STA   56064 ,Y
49168  INY
49169  BNE         49156
49171  LDY   # 0
49173  LDA   # 224
49175  STA   1024 ,Y
49178  STA   1984 ,Y
49181  INY
49182  CPY   # 40
49184  BNE         49175
49186  RTS
```

Program 3: Assembler Convenience

```
245 IFMN$="XX"THENPRINT"TO ADDRESS":INPUT
    DA:SA=DA:GOTO230
```

Program 4: VIC Loader

Remember to POKE 56,48

```
800 FOR ADRES=12288TO12316:READ DATTA:POK
    E ADRES,DATTA:NEXT ADRES
864 DATA 160, 0, 169, 6, 153, 0
870 DATA 148, 153, 0, 149, 200, 208
876 DATA 247, 160, 0, 169, 224, 153
882 DATA 0, 16, 153, 228, 17, 200
888 DATA 192, 22, 208, 245, 96
```

Program 5: 64 Loader

```
800 FOR ADRES=49152TO49186:READ DATTA:POK
    E ADRES,DATTA:NEXT ADRES
864 DATA 160, 0, 169, 8, 153, 0
870 DATA 216, 153, 0, 217, 153, 0
876 DATA 218, 153, 0, 219, 200, 208
882 DATA 241, 160, 0, 169, 224, 153
888 DATA 0, 4, 153, 192, 7, 200
894 DATA 192, 40, 208, 245, 96
```

Disk File Manager

(Article on page 130.)

BEFORE TYPING...

Before typing in programs, please refer to "How To Type COMPUTE!'s Gazette Programs," "A Beginner's Guide To Typing In Programs," and "The Automatic Proofreader" that appear before the Program Listings.

Disk Manager For VIC And 64

```
3 POKE 49152,10:IF PEEK(49152)<>10 THEN C
  0=1:GOTO6                            :rem 204
5 C0=2                                 :rem 23
6 DIM DOS%(65)                         :rem 215
7 FR=FRE(0):IF FR<0 THEN FR=FR+65536
                                       :rem 7
8 S=(FR-400)/2:M2=INT(S/256)+1         :rem 128
9 DIM TEMP%(S)                         :rem 18
10 PRINT"{CLR}";
15 PRINT"{2 SPACES}******************"
                                       :rem 43
20 PRINT"{2 SPACES}*{2 SPACES}DISK MANAGE
   R{2 SPACES}*"                       :rem 173
25 PRINT"{2 SPACES}******************"
                                       :rem 44
50 PRINT:PRINT"1.DISK DIRECTORY"       :rem 60
60 PRINT"2.FORMAT NEW DISK"            :rem 117
70 PRINT"3.INITIALIZE DISK"            :rem 182
80 PRINT"4.COPY FILE ON SAME DISK"
                                       :rem 228
85 PRINT"5.COPY FILE ON NEW (FORMATTED) D
   ISK"                                :rem 165
88 PRINT"6.COPY BOTH DOS WEDGE PROGRAMS"
                                       :rem 202
90 PRINT"7.RENAME FILE"                :rem 119
100 PRINT"8.ERASE FILE(S)"             :rem 252
110 PRINT"9.VALIDATE FILES"            :rem 135
120 PRINT"10.WRITE DISK MANAGER"       :rem 164
130 PRINT"11.ERROR STATUS"             :rem 99
140 PRINT"12.EXIT TO BASIC":PRINT:rem 253
170 INPUT"CHOICE{4 SPACES}{4 LEFT}";CHOIC
    E                                  :rem 113
180 IF (CHOICE<1)OR(CHOICE>12)THEN PRINT"
    {UP}";:GOTO 170                    :rem 166
200 ON CHOICE GOSUB 250,300,350,400,800,1
    200,450,500,550,600,650,700        :rem 127
210 GOTO 10                            :rem 45
250 REM *** DISPLAY DIRECTORY ***      :rem 66
251 PRINT"{CLR}"                       :rem 252
252 OPEN 1,8,0,"$"                     :rem 80
253 GET #1,A$,B$                       :rem 241
254 GET#1,A$,B$                        :rem 242
256 GET #1,A$,B$                       :rem 244
258 C=0:IF A$<>""THEN C= ASC(A$)       :rem 119
260 IF B$<>""THEN C=C+ ASC(B$)*256
                                       :rem 189
```