A NOTE FROM TONY McGOVERN

We have received the following note from Tony thanking the Group for our donation for his Funnelweb program. The article on page 2 of this newsletter speaks to a problem rampent in the T.I. community. Maybe group collections, like this one for Funnewweb, should be done by other groups and for other programmers. Our group sent a check for $184.00 to the McGOVERNS.

Tony's letter follows:

215 Grinsell St.
Kotara, NSW 2289 Australis
April 4, 1991

Dear Jim,

Thanks to all the MUNCHkins (my 18 year old daughter Eileen is in the habit of using that term on occasion) for their vote of appreciation.

I have been working recently on a full re-write of the 80-col Editor, and found the need for better error handling in the Script-Loader found some bugs in Low-load and in FW itself (in error path only). So this is a late march version update which is enclosed.

Since disks are rediculously expensive here (with box they are 80 cents apiece currently), there is not much point in sending disks to everyone, but what I will do is write out a bunch of signed disk labels to be given to all involved.

The Editor work will continue. Now is about the time I was intending to get an Amiga, but some extraneous matters seem to be putting that off for a while.

Tell Hugh that the real reason my letters are handwritten is that if I sit down at the TI, I can't resist the temptation to program - there is always another bug waiting to be tracked down, so they get written at times when I'm not near the machine. At the moment this is up at Hawks Nest for Easter holidays at the beach (50 miles North of Newcastle at the northers tip of Point Stephens. The disk will get made up some time next week.

Best regards to all,

Tony McG.

I hope to have these new disks ready for the contributors by the May meeting, Jim.

TI Users are "Cheapskates"
By: Andy Frueh (and Jim Peterson)

Before writing this article, I thought I had better ask Jim Peterson if I could quote him here. He said, "I have never tried to cover my belief that TI users are cheapskates." O.K. He thinks a bit stronger about this than I do, but not much stronger.

Mr. Peterson has TONS of great software, a good deal of it written by him but also has a huge collection of Public Domain programs. Some Fairware is there as well, but only if the author agrees to put it there. These disks are only $1.50 and are FULL. His own software ranges. He dropped prices on his Tips disks and Nuts Bolts. He would love to write education software, but no one buys software for education anymore, it seems.

As for myself, I know that TI users are "cheapskates." I myself try to not use Fairware because it is hard for someone with a job like mine to send off $20 or $15. Because I can't easily pay for it, I try to RARELY use it. I HAVE sent donations before, and I'm saving to send something for Funnelweb. However, how many of you bought a Fairware item at a Fair or out of a magazine or catalog, and didn't send the authors a thing? Not even a note saying "I liked it, but can't pay now," or "It was good, but needed this changed..." Where are these people? Is writing a note of appreciation so hard? Is $0.25 too much to ask for good, hard work?

Unfortunately, we already have some of our best people pulling out. WE a a community of TI users are making some of our strongest supporters against us Jim Peterson has released EVERY piece of his copyrighted software EXCEPT the Nuts Bolts disks in the public domain. I had a letter written to me by another Fairware author saying, "I hope you're making money, I'm not." When Jim Peterson announced his "suicide" sale, and slashed the cost on his own copyrighted material to $0.25, I can understand why people would flock to buy it. You go to Fairs LOOKING for bargains. But with Fairware it's different. You have to send the authors a donation if you like it. That's right, you HAV to. Unless you don't want any more software. This computer is GOOD. Better than a LOT of others I have seen. We have ways to make it portable, RAMdisks with mega memory, a new MIDI interface on the way, and even an 80 column card. But what if you have no software to use this equipment. What if there was no BOOT or MENU. No Funnelweb or DM-1000. Do you know that if the FAIRWARE authors hadn't written DM-1000, we'd all probably be using Disk Manager 2 or even DM 1?!? THINK ABOUT IT, folks.

I myself am not an excellant programmer. I do XB music and utilities and that's it. But I do it, and that's the point. I love my computer, and want t give it support. But what good is a support beam if it has no foundation to hold it up? Jack Sughrue says he sees a "new age" coming for the TI. I'd lik to feel the same way. But it's up to the USERS to support the PROGRAMMERS, or else this new age is going to be titled "R.I.P." After my updates to Home Filer and the new FileUtilities, Phantom of the Opera II, and maybe Les Miserables (another, more upbeat sounding musical) are finished, so am I. I personally feel cheated by several people, so I see no reason to offer any mor software support.

[This article/item comes from the March 1991 issue of BITS, BYTES PIXELS (Charles Good, editor), the newsletter of the Lima OH 99/4A User Group, P.O. Box 647, Venedocia, OH 45894. It may be used by other user groups as long as proper credit is given.]

PUZZLE-12

by WESLEY R. RICHARDSON
NORTHCOAST 99ER'S, CLEVELAND,OH

PUZZLE-12 is an Extended BASIC puzzle which uses joystick number 1 and works best with a color monitor or TV. The objective is to fit twelve pieces of various shapes into a rectangle at the center of the screen. The pieces are not allowed to overlap each other, however pegs on one piece may fit into holes on another. There are only two unique solutions, and 16 symmetry related solutions.

To use a piece, place the cursor on the piece using joystick #1, and press the fire button. The piece will be colored black for rotation. Each of the pieces can be rotated or flipped into eight orientations with the joystick.

Pressing the fire button again will color the piece white, and it can be moved with the joystick. When it is at the desired location, pressing the fire button will place the piece, if it is at an allowable position. The thick portion of the piece must be at the position indicated by the arrows, and the piece must not overlap another piece for the position to be acceptable.

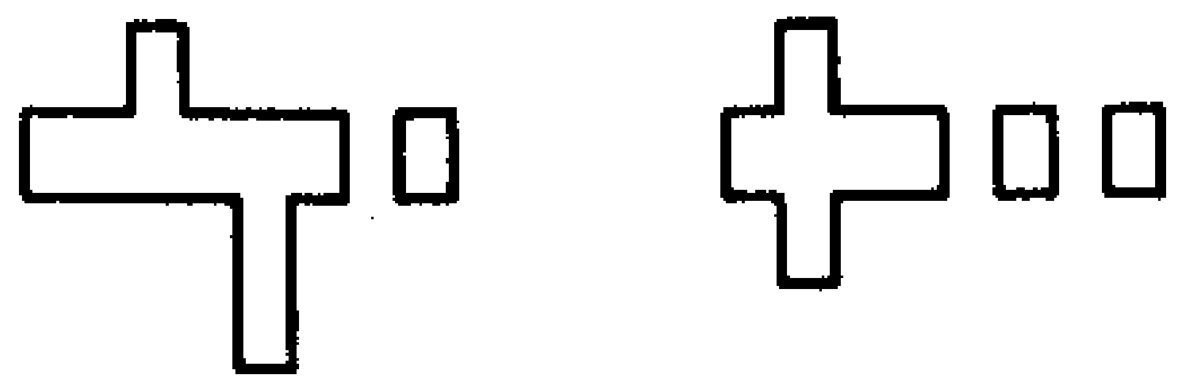
There are no time constraints for working the puzzle, so speed is not required. Pieces on the arrow line may also be removed from the rectangle and another piece tried in its place.

Pressing fire while the cursor is in the RESTART OR QUIT box, will allow you to restart the game or quit. The restart places all of the pieces in their starting boxes. The color of the pieces can be changed by pressing fire when the cursor is in the CHANGE COLOR box. There is no difference in piece shape as a result of changing colors, but some colors will be easier to see than others on the screen.

The program takes almost three minutes to initialize values and progress of the loading is shown on the two instruction screens. Restart during the game does not require the initialization sequence delay.

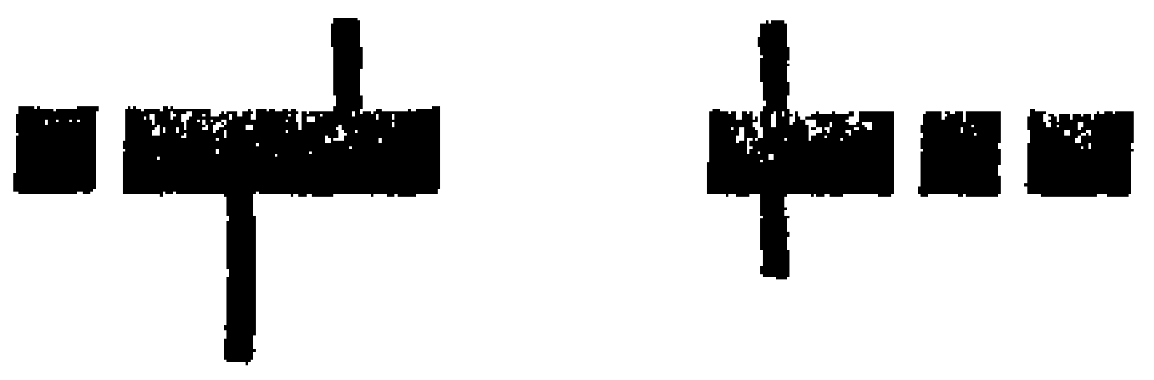
To receive the program on disk, send a SSSD or DSSD disk with some programs on it to 18140 Rolling Brook, Bainbridge, OH 44022-4860.

```
100 REM PUZZLE-12
110 REM BY WESLEY R. RICHARDSON, FEB 1
    991
120 REM NORTHCOAST 99ERS, CLEVELAND, O
    H
130 REM TI-99/4A EXTENDED BASIC
140 REM VARIABLES A(24),B$(12,8),C(36)
    ,C$(12,8),D(12),D$,E$,I,J,K,T,V,W,
    X,Y
150 DATA 2,3,4,5,6,7,8,9,10,11,13,14,1
    5
160 DATA "PLACE ALL 12 PIECES IN THE",
    "MIDDLE BOX.  PRESS FIRE ON","JOYS
    TICK #1 TO CHANGE COLOR"
170 DATA "TO BLACK TO ROTATE THE PIECE
    ","TO ANY OF 8 ORIENTATIONS","OR W
    HITE TO MOVE THE PIECE."
180 DATA "PRESSING FIRE AGAIN WILL","P
    LACE THE PIECE, IF IT IS","AT AN A
    LLOWABLE POSITION."
190 DATA "","  ALPHA LOCK MUST BE UP!"
200 DATA C0C0F3C000003F00,C0C0F3C00000
    3F00,C0C0FF0C0000F300,C0C0F3000000
    FF00
210 DATA C0C0FF000000F300,00C0F3C00000
    3F00,00C0F3C00000F300,00C0F3000000
    F3C0
220 DATA 00C0FF0000003300,00C0F3000000
    3F00,00C0FF0000003300,0000F3000000
    3300
230 DATA "THERE ARE ONLY TWO UNIQUE","
    SOLUTIONS, AND 16 SYMMETRY","RELAT
    ED SOLUTIONS. PRESSING"
240 DATA "THE FIRE BUTTON WHEN THE","C
    URSOR IS IN THE QUIT BOX","WILL AL
    LOW YOU TO RESTART"
250 DATA "OR END THE GAME. HAVE FUN,",
    "IT CAN BE DONE.","","","  ALPHA L
    OCK MUST BE UP!"
260 DIM A(24),B$(12,8),C(36),C$(12,8),
    D(12)
270 CALL DELSPRITE(ALL):: CALL MAGNIFY
    (4):: CALL SCREEN(12):: GOTO 300 :
    : D$,E$,I,J,K,T,V,W,X,Y
280 CALL CHAR :: CALL CLEAR :: CALL CO
    INC :: CALL COLOR :: CALL HCHAR ::
     CALL JOYST :: CALL KEY :: CALL LO
    CATE
290 CALL PATTERN :: CALL SPRITE :: CAL
    L VCHAR :: !@P-
300 W=0 :: GOSUB 1650 :: READ C(0):: F
    OR I=1 TO 12 :: READ C(I):: C(I+12
    )=5 :: C(I+24)=7 :: NEXT I
310 CALL CHAR(36,"103070FFFF703010",37
    ,"080C0EFFFF0E0C08")
320 CALL CHAR(40,"01070911113F11110907
    010000000000000C0201010F8101020C000
    0000000000")
330 CALL CHAR(60,RPT$("0",14)&"FF",61,
    RPT$("FF00",4),62,"FF")
340 GOSUB 1660 :: FOR I=1 TO 12 :: REA
    D B$(I,1):: NEXT I
350 FOR I=1 TO 12
360 FOR J=0 TO 4 STEP 4
370 FOR K=2 TO 4
380 D$=B$(I,J+K-1)
390 B$(I,J+K)=SEG$(D$,10,1)&SEG$(D$,1,
    1)&SEG$(D$,12,1)&SEG$(D$,3,1)&SEG$
    (D$,14,1)&SEG$(D$,5,1)&SEG$(D$,16,
    1)&SEG$(D$,7,1)
400 B$(I,J+K)=B$(I,J+K)&SEG$(D$,2,1)&S
    EG$(D$,9,1)&SEG$(D$,4,1)&SEG$(D$,1
    1,1)&SEG$(D$,6,1)&SEG$(D$,13,1)&SE
    G$(D$,8,1)&SEG$(D$,15,1)
410 NEXT K
420 D$=B$(I,1)
430 B$(I,5)=SEG$(D$,7,1)&SEG$(D$,16,1)
    &SEG$(D$,5,1)&SEG$(D$,14,1)&SEG$(D
    $,3,1)&SEG$(D$,12,1)&SEG$(D$,1,1)&
    SEG$(D$,10,1)
440 B$(I,5)=B$(I,5)&SEG$(D$,15,1)&SEG$
    (D$,8,1)&SEG$(D$,13,1)&SEG$(D$,6,1
    )&SEG$(D$,11,1)&SEG$(D$,4,1)&SEG$(
    D$,9,1)&SEG$(D$,2,1)
450 NEXT J :: CALL HCHAR(22,14+I,58)::
     NEXT I
460 GOSUB 1650 :: GOSUB 1660
470 FOR I=1 TO 12
480 FOR J=1 TO 8
490 C$(I,J)="" :: D$=B$(I,J)
500 FOR K=1 TO 16 STEP 2
510 C$(I,J)=C$(I,J)&RPT$(SEG$(D$,K,2),
    4)
520 NEXT K
530 NEXT J :: CALL HCHAR(22,14+I,58)
540 NEXT I :: CALL COLOR(1,9,1)
550 REM RESTART POINT
560 T=13 :: W=18 :: GOSUB 1650
570 CALL VCHAR(1,14,61,14):: CALL VCHA
    R(1,19,61,14):: CALL HCHAR(1,15,61
    ,4)
580 CALL HCHAR(14,10,61,14):: CALL HCH
    AR(18,10,61,14):: CALL HCHAR(24,10
    ,61,14)
590 CALL VCHAR(1,4,61,24):: CALL VCHAR
    (1,9,61,24):: CALL VCHAR(1,24,61,2
    4):: CALL VCHAR(1,29,61,24)
600 FOR I=0 TO 5 :: CALL HCHAR(1+4*I,5
    ,62,4):: CALL HCHAR(1+4*I,25,62,4)
    :: NEXT I
610 CALL HCHAR(24,5,60,4):: CALL HCHAR
    (24,25,60,4)
620 Y=97 :: X=113 :: CALL SPRITE(#1,40
    ,C(0),Y,X)
```



```
630 RANDOMIZE :: FOR I=1 TO 12 :: D(I)
    =1+INT(8*RND)
640 A(I)=I :: A(I+12)=0 :: CALL CHAR(9
    2+4*I,C$(I,D(I)))
650 CALL SPRITE(#I+1,92+4*I,C(I+V),1+3
    2*(I-1)+192*(I>6),33-160*(I>6))
660 NEXT I
670 REM MAIN LOOP, CURSOR
680 GOSUB 1680
690 DISPLAY AT(15,14)SIZE(1):"=" :: DI
    SPLAY AT(16,8)SIZE(14):"CHANGE=RES
    TART" :: DISPLAY AT(17,8)SIZE(14):
    "COLORS=OR QUIT"
700 CALL VCHAR(T-12,13,37):: CALL VCHA
    R(T-12,20,36):: IF T=25 THEN 720
710 CALL VCHAR(T-11,13,32):: CALL VCHA
    R(T-11,20,32)
720 CALL LOCATE(#1,Y,X)
730 CALL KEY(1,K,J):: IF J THEN 760
740 CALL JOYST(1,K,J):: IF (K=0)*(J=0)
    THEN 730
750 K=2*K :: J=-2*J :: X=MIN(MAX(25,X+
    K),201):: Y=MIN(MAX(1,Y+J),169)::
    GOTO 720
760 IF K<>18 THEN 730
770 IF (Y<105)+(Y>121)+(X<65)+(X>161)T
    HEN 890
780 IF (X>64)*(X<105)THEN 860
790 IF (X<121)+(X>161)THEN 890
800 GOSUB 1680 :: CALL LOCATE(#1,200,2
    00):: DISPLAY AT(15,8)SIZE(13):"C
    TO CONTINUE"
810 DISPLAY AT(16,8)SIZE(12):"R TO RES
    TART" :: DISPLAY AT(17,8)SIZE(9):"
    Q TO QUIT"
820 GOSUB 1630 :: IF (I=82)+(I=114)THE
    N 550
830 IF (I=67)+(I=99)THEN 680
840 IF (I=81)+(I=113)THEN CALL CLEAR :
    : STOP
850 GOTO 820
860 REM CHANGE COLOR
870 V=V+12 :: IF V>24 THEN V=0
880 FOR I=1 TO 12 :: CALL COLOR(#I+1,C
    (I+V)):: NEXT I :: GOTO 730
890 REM CHECK FOR ON PIECE
900 J=1 :: IF X>184 THEN X=193 :: J=7
    :: GOTO 920
910 IF X>41 THEN 930 ELSE X=33
920 Y=1+32*INT((Y+8)/32):: J=J+INT((Y+
    7)/32):: GOTO 1000
930 IF (Y>89)+(X<105)+(X>121)THEN 730
940 X=113 :: J=12+(Y+7)/8
```

```
950 IF J<>(T-1)THEN 730
960 IF A(J)=0 THEN 730
970 Y=Y+8*(D(A(J))<5):: IF T=25 THEN 9
    90
980 CALL VCHAR(T-11,13,32):: CALL VCHA
    R(T-11,20,32)
990 T=MAX(13,T-1)
1000 IF A(J)=0 THEN 730 ELSE I=A(J):: A
     (J)=0
1010 REM PIECE COLOR BLACK FOR ROTATE
1020 CALL LOCATE(#1,200,200):: CALL COL
     OR(#I+1,2)
1030 GOSUB 1680 :: DISPLAY AT(16,12)SIZ
     E(8):"PIECE ";CHR$(64+I):: DISPLAY
      AT(17,11)SIZE(8):"ROTATE";D(I)
1040 CALL KEY(1,K,J):: IF J THEN 1150
1050 CALL JOYST(1,K,J):: IF (K=0)*(J=0)
     THEN 1040
1060 K=0.25*K :: D(I)=D(I)+J
1070 IF D(I)<1 THEN D(I)=D(I)+8
1080 IF D(I)>8 THEN D(I)=D(I)-8
1090 J=0 :: IF D(I)>4 THEN J=4
1100 D(I)=D(I)+K
1110 IF D(I)<(1+J)THEN D(I)=4+J
1120 IF D(I)>(4+J)THEN D(I)=1+J
1130 CALL CHAR(92+4*I,C$(I,D(I))):: CAL
     L PATTERN(#I+1,92+4*I)
1140 GOTO 1020
1150 IF K<>18 THEN 1040
1160 REM PIECE COLOR WHITE TO MOVE
1170 GOSUB 1680 :: DISPLAY AT(16,12)SIZ
     E(8):"PIECE ";CHR$(64+I):: DISPLAY
      AT(17,13)SIZE(4):"MOVE"
1180 CALL VCHAR(T-12,13,32):: CALL VCHA
     R(T-12,20,32)
1190 CALL VCHAR(T-11,13,37):: CALL VCHA
     R(T-11,20,36):: CALL COLOR(#I+1,16
     )
1200 CALL KEY(1,K,J):: IF J THEN 1240
1210 CALL JOYST(1,K,J):: IF (K=0)*(J=0)
     THEN 1200
1220 K=2*K :: J=-2*J :: X=MIN(MAX(33,X+
     K),193):: Y=MIN(MAX(1,Y+J),161)
1230 CALL LOCATE(#I+1,Y,X):: GOTO 1200
1240 IF K<>18 THEN 1200
1250 REM POSITION OK?
1260 IF (Y>89)+(X<>113)THEN 1360
1270 CALL COINC(ALL,J):: IF J THEN 1200
1280 E$=B$(I,D(I))
1290 IF (Y=1)*((SEG$(E$,1,2)<>"00")+(SE
     G$(E$,9,2)<>"00"))THEN 1200
1300 IF (Y=81)*((SEG$(E$,7,2)<>"00")+(S
     EG$(E$,15,2)<>"00"))THEN 1200
1310 IF (Y=89)*((SEG$(E$,5,2)<>"00")+(S
     EG$(E$,13,2)<>"00"))THEN 1200
1320 J=12+(Y+7)/8 :: J=J-(D(I)<5)
1330 IF J<>T THEN 1200
```



```
1340 IF A(J)>0 THEN 1200
1350 CALL VCHAR(T-11,13,32):: CALL VCHA
     R(T-11,20,32):: T=MIN(25,T+1):: GO
     TO 1420
1360 IF X<185 THEN 1380
1370 X=193 :: J=7 :: GOTO 1400
1380 IF X>41 THEN 1200
1390 X=33 :: J=1
1400 J=J+INT((Y+8)/32)
1410 Y=1+32*INT((Y+8)/32)
1420 IF A(J)>0 THEN 1200 ELSE A(J)=I
1430 CALL LOCATE(#I+1,Y,X)
1440 CALL COLOR(#I+1,C(I+V))
1450 FOR J=1 TO 3
1460 IF A(J)>0 THEN I=J :: GOTO 1490
1470 IF A(J+6)>0 THEN I=J+6 :: GOTO 149
     0
1480 NEXT J :: GOTO 1550
1490 FOR K=6 TO J+1 STEP -1
1500 IF A(K)=0 THEN 1530
1510 IF A(K+6)=0 THEN K=K+6 :: GOTO 153
     0
1520 NEXT K :: GOTO 1550
1530 A(K)=A(I):: A(I)=0 :: I=A(K)
1540 CALL LOCATE(#I+1,1+32*(K-1)+192*(K
     >6),33-160*(K>6))
1550 REM CHECK FOR WIN
1560 FOR J=24 TO 13 STEP -1
1570 IF A(J)=0 THEN 680
1580 NEXT J :: CALL SCREEN(16)
1590 DISPLAY AT(16,9)SIZE(11):"YOU DID
     IT!" :: DISPLAY AT(17,8)SIZE(13):"
     PRESS ANY KEY"
1600 GOSUB 1630 :: CALL SCREEN(12):: GO
     TO 680
1610 STOP
1620 REM SUBROUTINES
1630 CALL KEY(0,I,K):: IF K THEN RETURN
1640 CALL KEY(1,I,K):: IF K THEN RETURN
      ELSE 1630
1650 CALL CLEAR :: DISPLAY AT(W+2,10):"
     PUZZLE-12" :: DISPLAY AT(W+4,10):"
     WESLEY R." :: DISPLAY AT(W+5,10):"
     RICHARDSON" :: RETURN
1660 FOR I=7 TO 17 :: READ E$ :: DISPLA
     Y AT(1+I,1):E$ :: NEXT I :: DISPLA
     Y AT(22,5):"LOADING"
1670 CALL HCHAR(21,15,60,12):: CALL HCH
     AR(23,15,62,12):: RETURN
1680 FOR J=15 TO 17 :: DISPLAY AT(J,8)S
     IZE(14):"" :: NEXT J :: RETURN
1690 !@P+
1700 END
```

PUTTING IT ALL TOGETHER No. 9

by Jim Peterson

The hard part of learning to program is not in learning what the various commands do - it is learning how to put them together to do what you want them to do! Key in this little program and run it to see what it does, then study the explanation of how it does it.

```
1 !STRAIGHT-LINE CALCULATOR
   TINYGRAM by Jim Peterson
   Accepts input such as
   6+6-11*2+3/4
2 T,F=0 :: C$="+-*/" :: ACCE
PT AT(12,1)ERASE ALL VALIDAT
E(NUMERIC,C$):F$ :: L=LEN(F$
):: FOR J=1 TO L :: X$=SEG$(
F$,J,1):: P=POS(C$,X$,1):: I
F P=0 THEN 5
3 IF F=0 THEN T=VAL(SEG$(F$,
1,J-1)):: F=1 :: A=J+1 :: P2
=P :: GOTO 5
4 V=VAL(SEG$(F$,A,J-A)):: A=
J+1 :: GOSUB 7 :: P2=P
5 NEXT J :: V=VAL(SEG$(F$,A,
255)):: GOSUB 7 :: DISPLAY A
T(12,L+1):"=";STR$(T)
6 DISPLAY AT(24,1):"PRESS AN
Y KEY" :: CALL KEY(0,K,S)::
IF S=0 THEN 6 ELSE 2
7 IF P2=1 THEN T=T+V ELSE IF
 P2=2 THEN T=T-V ELSE IF P=3
 THEN T=T*V ELSE T=T/V
8 RETURN
```

The calculations are done from left to right, not in the mathematical hierarchy of multiplication and division first.

The variables T and F are reset to 0 because program execution returns here. A string of math symbols is placed in C$. The calculation is accepted into F$, using ERASE ALL to clear the screen; the VALIDATE will accept only numeric characters (numerals and decimal point) and the symbols assigned to C$. L measures the length of the string. The J loop examines the characters in the string, from the first to the last, extracting one character at a time into X$. POS checks whether that character is the 1st, 2nd, 3rd or 4th character of the C$ "+-*/" and places that value in P, or a 0 if it does not match any of them. In this case, X$ was a numeric character so execution jumps to NEXT J to continue the loop.

Otherwise, the first math symbol in the string has been found. F (a flag variable) still equals 0 so VAL converts the part of F$ from the first character up to the math symbol into its numeric form, in T. The flag F is set to 1 so that line 3 will be skipped over from now on. The position of the first character after the math symbol (the beginning of the next number) is saved in A and the value of P (the position of the math symbol in C$) is saved in P2. The loop continues, finding the digits of the next number, until another math symbol is found. F does not equal 0 so execution jumps to line 4. The segment of F$ starting from the position saved in A, to J-A (the character preceding the current math symbol) is converted to numeric by VAL and placed in V. The position to start looking for the next number is again saved in A. The GOSUB jumps to line 7. Depending on the position in C$ ("+-*/"), saved in P2, of the previously found math symbol, the value of this second number, saved in V, is aded to, subtracted from, multiplied by or divided into the previous number saved in T, and the new value is saved in T. Execution then RETURNs to the last statement in line 4, to save the value of P (the location in C$ of the current, not yet used, math symbol) in P2, and the loop continues.

When the loop is completed, in line 5, the value of the final numeric characters is determined, the GOSUB again uses the value saved in P2 to determine the final calculation, and the result is printed out. Since the original input was in row 12, column 1, and the length of the input was saved in L, L+1 places the "=" directly after it, and converting the value T into a string by using STR$ causes it to print directly thereafter without an intervening space.

If S (status) in the CALL KEY is 0, it means that no key was pressed, so the line is repeated; otherwise, execution goes back for another input.

FUNNELWEB Vn4.31 TUTORIAL (Part 1)
DISK REVIEW (1.1) by Jerry Keisler

**** GENERAL ****

DISK REVIEW has the newest changes in FW so I will start with it. To improve the usefulness of this tutorial I will show the screens and explain the active parts of each screen. This will take several months, hence the ()'s in the title. (DR) = DISK REVIEW. The upper left is a general discriptor of the screen. The upper right is my filename for the TI-Artist picture.

**** FIRST SCREEN ****

This is the first screen seen when DR is selected.
<ESCAPE> (esc) is FCTN/9 or CTRL/C. This allows backing out of one or more screens or windows.
<BREAK> is FCTN/4. This terminates most processes.
(1-9) Reads the indicated DISK DIRECTORY from several locations in DR and produces the DskDir screen. (esc) or (E/X) keys will return to a valid directory display if one is present.
(0) zero will cycle screen colors on several screens.
(D) DISK UTILITIES are activated. The D-Util screen will be shown in a later tutorial. CTRL/A or FCTN/6 will produce D-Util unless a valid directory is present.
(F) produces the current FUNNELWEB Central Manu. The FnlWeb screen will be shown in a later tutorial.
(c=) CTRL/= returns to FUNNELWEB from several screens.

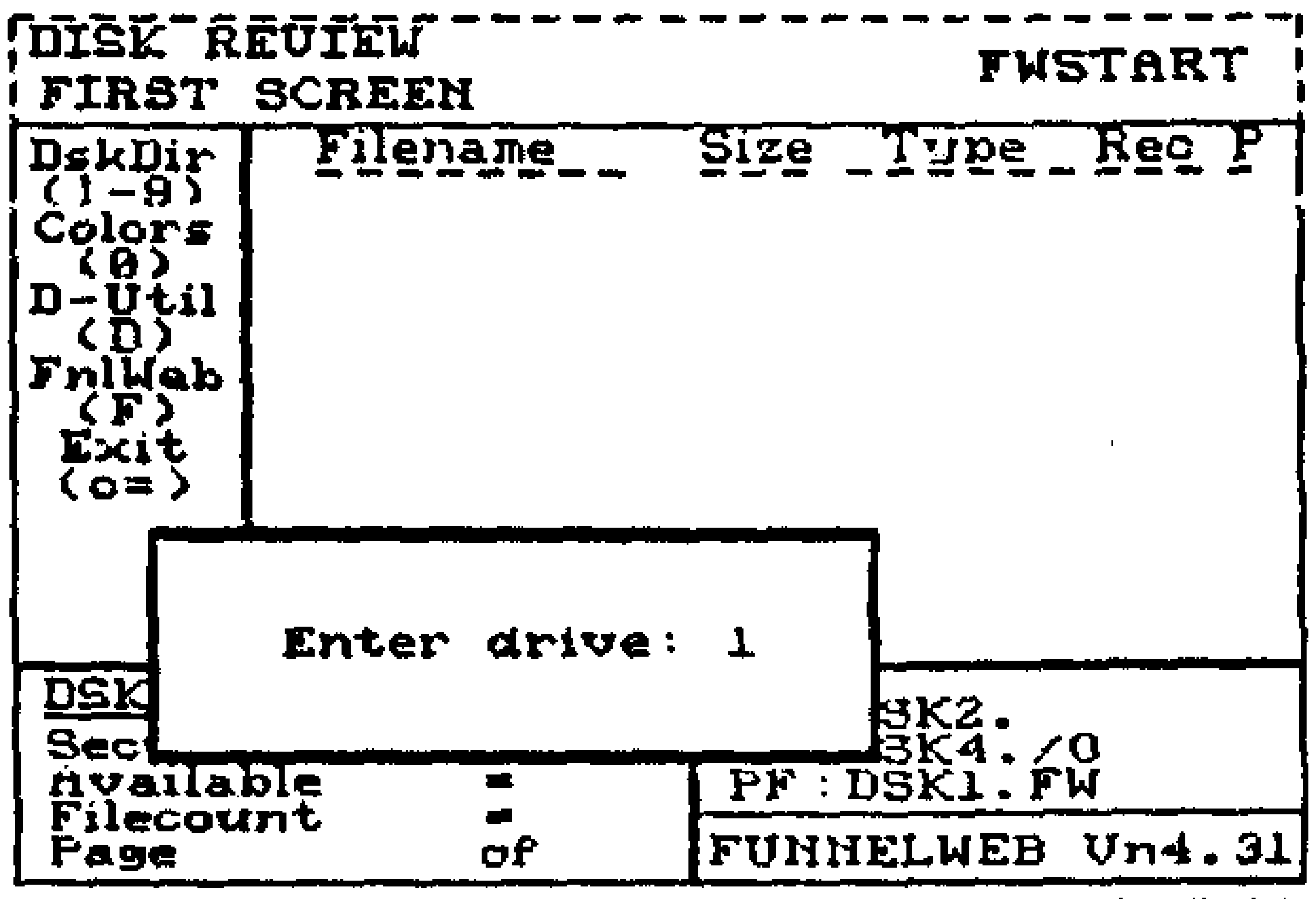
**** DISK DIRECTORY (1-9) SCREEN ****

FCTN/8 re-reads directory.
<enter> returns to initial screen.
(esc) CTRL/C or FCTN/9. ^FP^(E/X) and FCTN (E/X) move the cursor up and down the directory one line at a time.
(B/N) and CTRL (E/X) Pages thru the directory.
(space-bar) marks file. It may appear in WF OF PF block if appropriate.
WF current workfile ie, TI-Writer LF/SF, DV80, DF80, etc.
OF current object file DF80.
PF current assembly program file.
<O> Oldfile restores old filename before DR was used to WF.

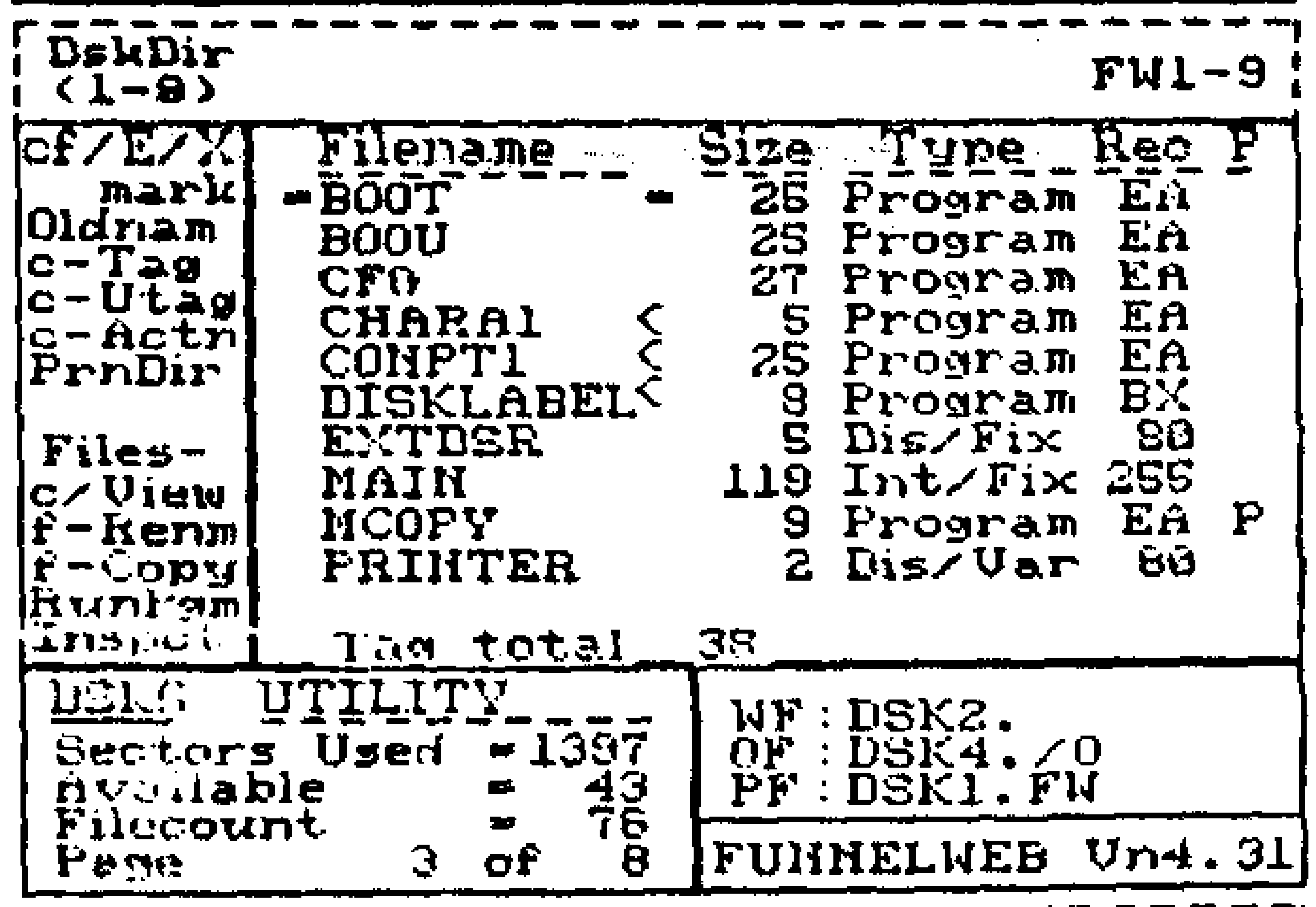
-----TAG OPERATIONS---------
<T> tags the file under the cursor with <.
CTRL/T tags all files with <.
<U> untags the file under the cursor.
CTRL/U untags all files.
Tag total is the total size of all tagged files.
CTRL/C c-Action works on all tagged files as follows:

** See TAG c-Action windows **
<C> copies all tagged files to up to 8 different drives. <BREAK> FCTN/4 is checked after each file is completely copied. If you FCTN/C before a file is completely copied, it should be deleted.
<U/P> unprotects or protects all tagged files.
<D> deletes tagged unprotected files. Each file is presented for verification. FCTN/6 is required to delete each file.
<R> rename all tagged files.
<U/P D R> Dir is reread after these actions are completed or after (esc) for verification.
You may have to (esc) a tag operation 3 times then return to the tag function.
----end TAG operations------

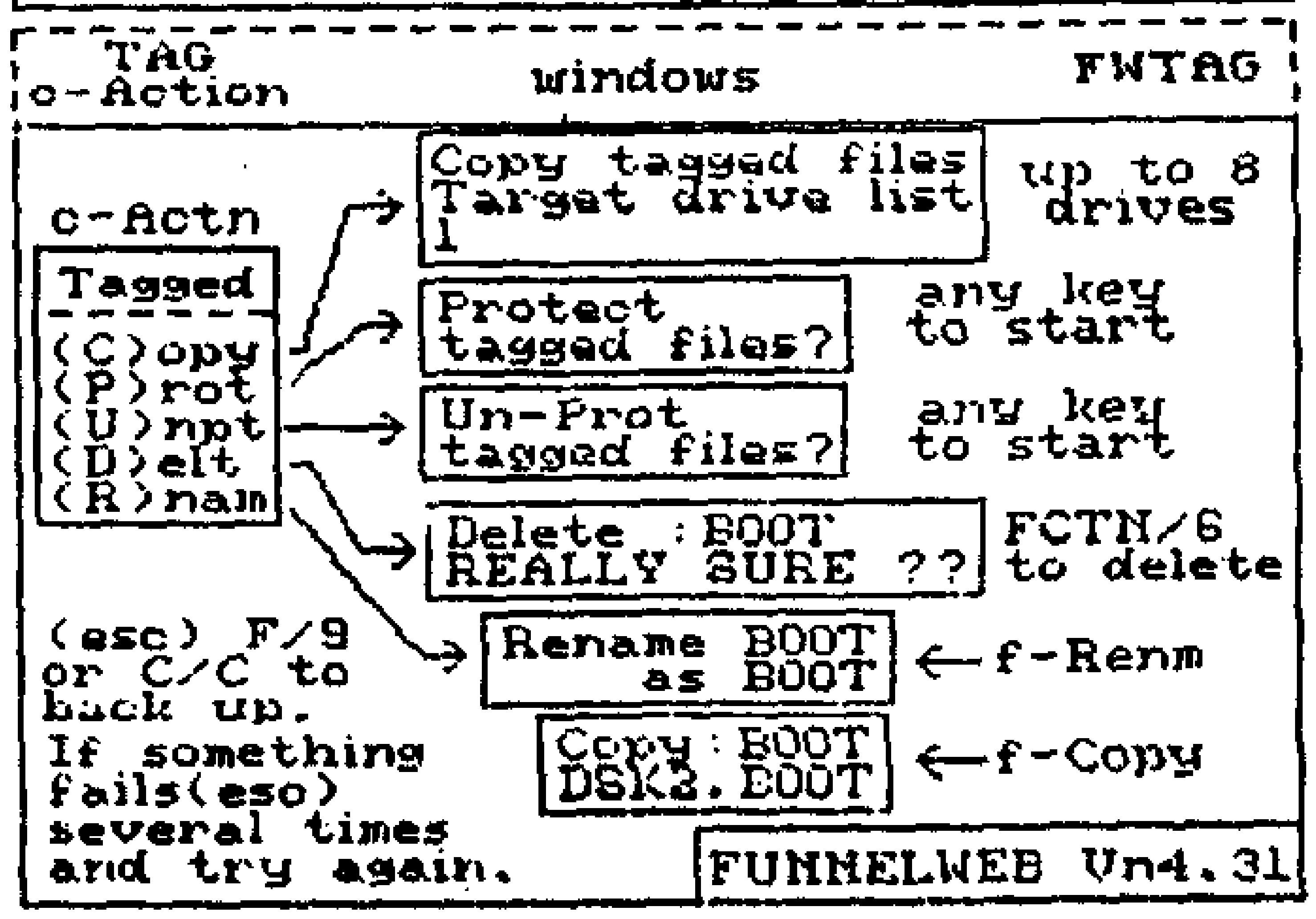
<P> print directory, prints to device shown in TI-Writer PF in the APPEND mode.
(V) views any type file to screen one screen at a time.
CTRL/V views any type file to screen in continious scroll.
VIEW will be covered in a later tutorial in greater detail.
FCTN/R renames file under cursor. The directory will be reread to verify changes.
FCTN/C copies file under cursor to any drive 1-9, using same or different name. Will prompt for disk swapping if same drive and name. Copy buffer is 46 sectors.
<R> run program sends marked file (space-bar) to appropriate loader.
Basic/XB should load and run automatically. Most assembly program files should load under 2 GPL.
Text files gives you a warning. If you continue, it is treated as a script file to load up to 15 E/A object files. Script file will be covered in a later tutorial.
<I> inspect and edit sectors will be covered in a later tutorial.



```
DISK REVIEW                              FWSTART
FIRST SCREEN
DskDir   Filename__    Size _Type_ Rec P
(1-9)
Colors
 (0)
D-Util
 (D)
FnlWeb
 (F)
Exit
(c=)
           Enter drive: 1
DSK                              SK2.
Sec                              SK4./O
Available          =          PF : DSK1.FW
Filecount          =
Page               of         FUNNELWEB Vn4.31
```



```
DskDir
(1-9)                                      FW1-9
cf/E/X   Filename__    Size _Type_  Rec P
  mark  -BOOT       -    25 Program  EA
Oldnam   BOOU            25 Program  EA
c-Tag    CFO             27 Program  EA
c-Utag   CHARA1     <     5 Program  EA
c-Actn   CONPT1     <    25 Program  EA
PrnDir   DISKLABEL  <     9 Program  BX
         EXTDSR           5 Dis/Fix  80
 Files-  MAIN           119 Int/Fix 255
c/View   MCOPY            9 Program  EA  P
f-Renm   PRINTER          2 Dis/Var  80
f-Copy
Runprm
Inspct    Tag total   38
DSKG  UTILITY____            WF : DSK2.
Sectors Used  = 1397         OF : DSK4./O
Available     =   43         PF : DSK1.FW
Filecount     =   76
Page        3  of   8        FUNNELWEB Vn4.31
```



```
TAG
c-Action            windows                FWTAG

                  Copy tagged files    up to 8
c-Actn            Target drive list     drives
                  1
Tagged
------            Protect               any key
(C)opy            tagged files?         to start
(P)rot
(U)npt            Un-Prot               any key
(D)elt            tagged files?         to start
(R)nam
                  Delete  : BOOT        FCTN/6
                  REALLY SURE ??        to delete

(esc) F/9          Rename BOOT       ←f-Renm
or C/C to              as BOOT
back up.
If something       Copy : BOOT       ←f-Copy
fails(esc)         DSK3.BOOT
several times
and try again.                   FUNNELWEB Vn4.31
```

From: J.U.G. 3-91

MAKE YOUR OWN FLASH CARDS

by Tony Falco

Last summer a friend came to me with a programming problem. He wanted a program to display arithmetic flash cards, with any numbers, and problems in text book format. It was not as easy a task as I thought it might be, but the string commands in BASIC coupled with T.I. Extended BASIC's DISPLAY AT and ACCEPT AT commands did the job.

The user picks one of three operations. Then he picks his own numbers. Entering <Q> for the first number will end the program. The program works best if the child and parent work at the computer together.

In a future article, I will show how to adapt the program so the computer generates the problems.

```
10 CALL CHAR(104,"FF80808080
8080FFFF010101010101FF")
20 DISPLAY AT(12,6)ERASE ALL
:"PICK ONE-->+-x  +"
30 ACCEPT AT(12,22)SIZE(-1)V
ALIDATE("+-X")BEEP:OP$
40 CALL CLEAR :: CALL FLASH
:: CALL CHAR(95,"0000FFFF")
50 DISPLAY AT(3,4)SIZE(-6)BE
EP:"RIGHT:" :: DISPLAY AT(3,
15)SIZE(-6):"WRONG:"
60 DISPLAY AT(5,10)SIZE(-6):
"SCORE:"
70 FOR J=12 TO 15 :: DISPLAY
 AT(J,1)SIZE(-25):" " :: NEX
T J
80 DISPLAY AT(13,7)SIZE(-1)B
EEP:OP$
90 ACCEPT AT(12,9)VALIDATE(D
IGIT,"Q")SIZE(-4):A$
100 IF A$="Q" THEN 280 ELSE
ACCEPT AT(13,9)VALIDATE(DIGI
T)SIZE(-10):B
110 A=VAL(A$)
120 C=-(A+B)*(OP$="+")-(A*B)
*(OP$="X")-(A-B)*(OP$="-")
130 B$=STR$(B):: C$=STR$(C):
: M=MAX(LEN(A$),LEN(B$)):: N
=MAX(M,LEN(C$))
140 DISPLAY AT(13,1)SIZE(-5)
:" " :: DISPLAY AT(13,7-M+LE
N(A$))SIZE(-10):OP$
150 DISPLAY AT(13,9+LEN(A$)-
LEN(B$))SIZE(-10):B$
160 DISPLAY AT(14,8+LEN(A$)-
N)SIZE(-10):RPT$(CHR$(95),N+
2)
170 ACCEPT AT(15,9+LEN(A$)-L
EN(C$))SIZE(-LEN(C$))VALIDAT
E(DIGIT,"-")BEEP:D
180 IF D=C THEN R=R+1 :: CAL
L SAY("#THAT IS RIGHT"):: CA
LL DELAY(200):: GOTO 250
190 DISPLAY AT(12,19)SIZE(-L
EN(A$)):A$
200 DISPLAY AT(13,17-M+LEN(A
$))SIZE(-1):OP$
210 DISPLAY AT(13,19+LEN(A$)
-LEN(B$))SIZE(-LEN(B$)):B$
220 DISPLAY AT(14,18+LEN(A$)
-N)SIZE(-9):RPT$(CHR$(95),N+
2)
230 DISPLAY AT(15,19+LEN(A$)
-LEN(C$))SIZE(-LEN(C$)):C$ :
: W=W+1
240 CALL SAY("#THAT IS INCOR
RECT"):: CALL DELAY(700)
250 S=INT(100*R/(W+R)+.5)
260 DISPLAY AT(3,10)SIZE(3):
R :: DISPLAY AT(3,21)SIZE(-3
):W
270 DISPLAY AT(5,17)SIZE(-4)
:STR$(S)&"%" :: GOTO 70
280 CALL SAY(STR$(R)):: CALL
 SAY("CORRECT AND"):: CALL S
AY(STR$(W))
290 CALL SAY("NOT CORRECT"):
: CALL CLEAR :: END
300 SUB DELAY(X):: FOR D=1 T
O X :: NEXT D :: SUBEND
310 SUB FLASH :: CALL SCREEN
(12):: FOR Z=1 TO 8 :: CALL
COLOR(Z,2,15):: NEXT Z
320 CALL COLOR(9,2,2,10,12,1
2):: CALL HCHAR(1,1,104,768)
330 FOR Z=2 TO 6 :: CALL HCH
AR(Z,5,32,22):: NEXT Z
340 FOR Z=10 TO 18 :: CALL H
CHAR(Z,3,32,27):: NEXT Z
350 CALL HCHAR(7,6,96,22)::
CALL VCHAR(3,27,96,4)
360 CALL HCHAR(19,4,96,27)::
 CALL VCHAR(11,30,96,8):: SU
BEND
```

NEXT MEETING TUESDAY MAY 14, 1991. SPRING IS HERE!!!!!!!

MUNCH OFFICERS AND NUMBERS (all in 508 area unless noted)

President	W.C. Wyman	865-9683
Vice President	Bruce Willard	852/3250
Secretary	Jim Cox	
Treasurer	Jim Cox	869-2704
Acting Editor	Jim Cox	
Adv.Prog. Chair	Dan Rogers	248-5502
Library	Al/Lisa Cecchini	
Disk Librarian	Lou Holmes 617	965/3584
Tape Librarian	Walter Nowak 413	436/7675
NEW-AGE/99	Jack Sughrue	476/7630

MUNCH DUES

NEW MEMBERSHIP	$25.00
RENEWAL MEMBERSHIP	$15.00
NEWSLETTER ONLY SUBSCRIPTION	$12.00

APRIL MEETING. There were 12 members at the meeting. There was a little mix-up as there was another meeting scheduled for the hall, but everything was worked out. Corson showed a video of the BCS fair. Don Mason demoed a new game, Backensteine, and also his Horizion Ram Disk. A discussion was held on what to d with the new hardware bought at the fair.

MAY MEETING. I, Jim, am happy to anounce my return this month. I started a new job at the Milford National Bank which will free up my Tuesday evenings. I am not sure exactly what we will have this month but I expect Jack to have something to demo and other surprises. I hope to have the new Funnelweb's for the contributors, see Page 1.

RAFFLE. Every month we have a raffle to help defer the rental cost of our meetin hall. A typical raffle will have game and utility programs, T-Shirts, books, bumper stickers, blank discs and all sorts of odds and ends for the T.I.

LIBRARY NOTICE. Please return any items borrowed from our library. If you can not come to a meeting or give these items to someone who will be at the meeting.

REPRINTS. Reprints are permitted as long as credit is given to M.U.N.C.H.

ARTICLES. I am always looking for articles for this newsletter, anything which interests you will probably interest other members of the TI community, so please share your ideas and opinions with all of us.

DISK LIBRARY. The disk library will be at the meetings from now on. We have copies of all disks in the library and they are available to members for just $1.50 each.

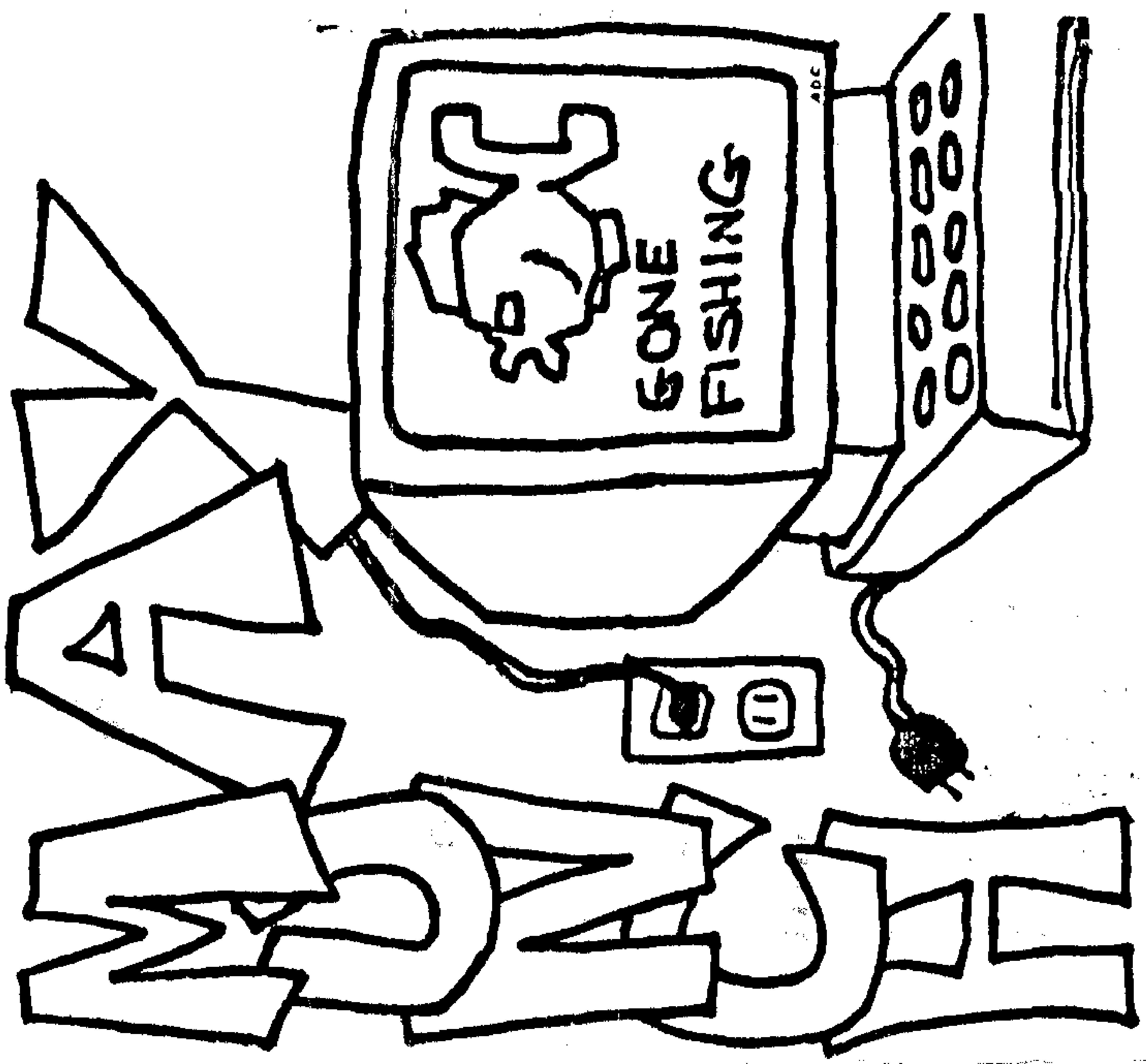
FOR SALE. The group has a TI Count Business Software package available for sale. If interested contact Jim Cox at the above numer or the club address.

DISK OF THE MONTH. I am sorry I could not get the Disks done for last month, but there was a family emergency which took up a lot of my time. This month we will have the game disk planned for last month.

THANK YOU. I want to thank everyone who helped out while I couldn't attend the metings. An especially big Thank You to Walt Nowak for all of his help!!!

Mass Users of the Ninety-nine and Computer Hobbyists

MAY 1991 Monthly Newsletter Version 10.05

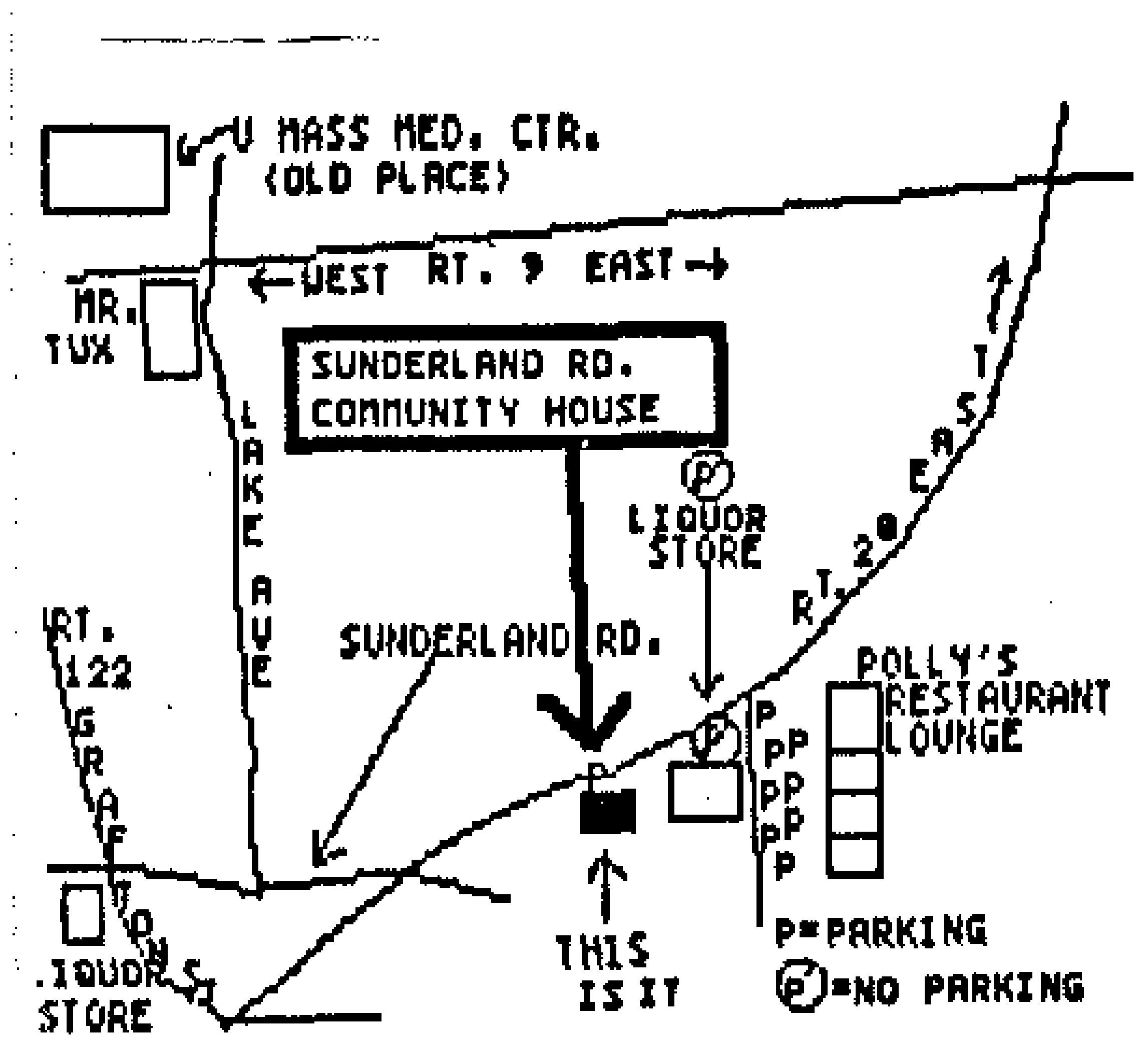


M.U.N.C.H.
560 LINCOLN ST.
P.O. BOX 7193
WORCESTER, MA. 01605-7193

WORCESTER, MA 016 PM 8 MAY 1991

FIRST CLASS

Next Meeting MAY 14TH.



POSTMASTER: Forwarding and Address Correction Requested.