99

HOCUS

Milwaukee APR 86

Home Computer
Users Spotlight
a monthly publication of the
Milwaukee Area 99/4 Users Group

Dear Home Computer Magazine Subscriber:

On behalf of Home Computing Journal, I would like to welcome you as a former subscriber to Home Computer Magazine. Home Computer Magazine (HCM) ceased publication with the Volume 5, Number 6 issue. We will be fulfilling the balance of your HCM subscription with Home Computing Journal. You will receive a number of issues equivalent to the value remaining on your original magazine and/or media subscription(s).

Home Computing Journal is a quarterly publication that is delivered to your door with a computer disk of valuable software. Many of the best features of the old Home Computer Magazine are now being incorporated into the Journal.

AUTORUN PART II

by N. A. Molander
Suncoast Beeper Newsletter

DELETING AUTORUN

To eliminate auto-execution delete the AUTORUN line from the object code file and resave the file. To execute this saved program will now require the Program Name, the symbol in the REF/DEF record with the same memory address. Autorun can be restored by re-entering the record in the object code exactly at the same line number and with the same format that was deleted.

These changes can be made with the EDITOR of the E/A program. As the procedure being described here only applies to UNCOMPRESSED object code files, be sure the format complies. The screen should be filled with ALPHANUMERICS, if there are only two spaces between object code tags (B's ect) and most are blank except for recognizable text, it is a COMPRESSED object code program and cannot be altered. Scroll to the last lines of the program to see the line sequence desribed above and proceed with the changes. DISKO, TI-WRITER, and DISKFIXER can also be used in the same way.

ADDING AUTORUN

Adding Autorun to an object code program is similar to replacing a deleted one except that the Entry Point desired must be chosen and the checksum calculated. The checksum will be ignored if the TAG is changed from a 7 to an 8. (E/A pg 241).

The most likely choice is the address of the Program Name used to run the program. In the above example by selecting 50006TEST as the entry address then line 3 would be created and inserted into the object code file as shown. By ignoring the checksum record 2000680000F could be entered as the record of line 3.

Of the 14 Oject Code Tags used by the TMS9900, 5 relate to Absolute addresses and 5 to Relocatable addresses, (pg 240 E/A). Programs with absolute addresses use the AORG directive and are placed in a specific memory location. The only effect on the above example would be that TAGS 2 5 become 1 6.

The simplified procedure presented here will work in many cases, however, complications can arise.

◆

APRIL 1986

Milwaukee Area User Group
4122 Glenway Wauwatosa WI 53222

President......	James Schroeder	264-4735
Vice-president.	Donald Walden	529-2173
Treasurer......	Phil Norton	462-8954
Secretary......	George Kasica	321-7558
Librarians.....	E.J. VonDerEhe	549-0597
.....	Fred Pabian	327-3618
Newsletter.....	Gene Hitz	535-0133
.....	Jerry Trinkl	327-0170
Forth Info.....	Gene Hitz	
Assembly SIG...	Jerry Trinkl	

Meetings:

Second Saturday each month
Wauwatosa S&L 7500 W. State Street
1:00 til 4:00

First Monday each month
National S&L National Ave & Moorland
7:00 til 10:00

Dues:

Individuals	- $10
Families	- $15

NOW THAT'S

USER FRIENDLY!!

THE NEW ENGLAND FAYUH

Here is a quick report on the New England Fayuh, as seen by one of the organizers (me):

The new computer-on-a-card was there and working in the morning! It was a wire wrap version that crashed by the afternoon MYARC presentation in the auditorium. A lot of people saw it, and it certainly wasn't vaporware. The first 20 PC boards have been on order for about two weeks. If they had been delivered on time, one of them would have been at New England, not the wire wrap version. They should be ready real soon now (or even sooner). For the wire wrap version to survive the trip at all was perhaps fortunate. They aren't made to travel! Mass production by June, if all goes well. The card will include four ports - one for the IBM type keyboard, one for a mouse, one for joysticks and one for the monitor. The card will not support your color TV as the 4A does. If you want color you need a composite or RGB monitor. RGB will be necessary if you are to take full advantage of the high-res graphics or 80-column text capability and still have color. The present TI monitor doesn't have the resolution needed for 80 columns! As far as XBII is concerned, version 2.1 was not quite ready yet. The MIN/MAX is debugged and the integer math is working but the user generated DEFs and SUBs are still giving some trouble. Lou Phillips made it quite clear that purchase of version 2.0 now entitles you to 2.1 without additional charge (except mailing?) If one person in an area gets the update on disk, it can be passed on to others, since the disk alone is worthless without the EPROM on the 129/512K card. There is no good reason to wait to buy the present version, unless you just don't want to be bothered with an incomplete version for now.

I bought it this week to become familiar with the new functions, even though I knew it would not run most of my own software yet. I expect it will be readv within the month. Lou also clarified some comments made at TICOFF. The finished computer will have a new operating system with perhaps some GEM similarities. (Paul Charlton says GEM-like displays aren't an operating system, but an I/O device.) XBII will be version 3.0 with new graphic modes, not the currently expected 2.1, and not even be called "Extended BASIC" any more but given a new name. I expect not all will be done for the initial run if it's in June, but both BASIC and the O.S. will be on disk anyway and can be upgraded as improved.

Barry Traver gave a superb inspirational talk about the TIfamily community and how it pulls together to support us all. Jim Horn talked about the role of telecommunications in the TI world and in general and then he called JZ out of the audience for a few words - ostensibly about Compuserve. Well, JZ showed us all he is a very fine public speaker and an astute analyst of sociological trends. Remember that, future faire organizers! We all knew JZ could talk on a keyboard but he is even better up in front of an audience! Paul Charlton, as usual, enthralled the technically sophisticated. His audiences are small because his level is beyond a lot of us, but those that stay really listen. Chris Bobbit spoke of exciting new things under investigation by ASGARD, particularly Richard Roseen. Their work could lead to an alternative type of computer - bare bones in execution but very fast and very capable. Prototypes will be developed but there is no commitment to market it unless it looks like there really is a demand for it. All-in-all, the New England Fayuh seemed to run very well. It certainly turned a good profit for all the clubs participating and the dealers seemed to do quite well too. The headcount was about 600 people with about half the cars in the parking lot from out of state. It was done without any paid advertising but every other source we could think, was exploited and the word reached the people it was intended. Peter Hoddie's phone was ringing every few minutes the last three days before the Fayuh from people who had heard and wanted directions.

TIPS FROM THE TIGERCUB

#29

Copyright 1985

TIGERCUB SOFTWARE
156 Collingwood Ave.
Columbus, OH 43213

Distributed by Tigercub Software to TI-99/4A Users Groups for promotional purposes and in exchange for their newsletters. May be reprinted by non-profit users groups, with credit to Tigercub Software.

Over 13Ø original programs in Basic and Extended Basic, available on casette or disk, only $3.ØØ each plus $1.5Ø per order for PPM. Entertainment, education, programmer's utilities.
Descriptive catalog $1.ØØ, deductable from your first order.
Tips from The Tigercub, a full disk containing the complete contents of this newsletter Nos. 1 through 14, 5Ø original programs and files, just $15 postpaid.
Tips from the Tigercub Vol. 2, another diskfull, complete contents of Nos. 15 through 24, over 6Ø files and programs, also just $15 postpaid. Or, both for $27 postpaid.
Nuts & Bolts (No. 1), a full disk of 1ØØ Extended Basic utility subprograms in merge format, ready to merge into your own programs. Plus the Tigercub Menuloader, a tutorial on using subprograms, and 5 pages of documentation with an example of the use of each subprogram. All for just $19.95 postpaid.
Nuts & Bolts No. 2, another full disk of 1Ø8 utility subprograms in merge format, all new and fully compatible with the last, and with 1Ø pages of documentation and examples. Also $19.95 postpaid, or both Nuts Bolts disks for $37 postpaid.
Tigercub Full Disk Collections, just $12 postpaid! Each of these contains either 5 or 6 of my regular $3 catalog programs, and the remaining disk space has been filled with some of the best public domain programs of the same category. I am NOT selling public domain programs - my own programs on these disks are greatly discounted from their usual price, and the public domain is a FREE bonus!
TIGERCUB'S BEST
PROGRAMMING TUTOR
PROGRAMMER'S UTILITIES
BRAIN GAMES
BRAIN TEASERS
BRAIN BUSTERS!
MANEUVERING GAMES
ACTION GAMES
REFLEX AND CONCENTRATION
TWO-PLAYER GAMES
KID'S GAMES
MORE GAMES
WORD GAMES
ELEMENTARY MATH
MIDDLE/HIGH SCHOOL MATH
VOCABULARY AND READING
MUSICAL EDUCATION
KALEIDOSCOPES AND DISPLAYS
For descriptions of these send a dollar for my catalog!
The offer made last month is still good until 1 January - a 1Ø% rebate directly to the user group if one of their members mentions the user group when ordering from me. So far, I've had only 1Ø responses - and I suspect that 8 or 9 of those didn't even know about the offer!

I goofed again. In the I/O ERROR routine in Tips #28, the ON ERROR STOP will do no good in the place where I put it. It should be placed after the file is opened in line 1ØØ so that it will become the current error trap if the file is opened correctly.

And the CALL KEY example in Tips #28 will look better if R=14. A couple of very knowledgeable programmers have written to tell me that I was wrong, and the manual is right, about CALL KEY status -1. They say that -1 simply means that the same key is being pressed as was pressed during the last keyscan, and that it could have been released and repressed in the interim. This may be, but try this routine and see if you can release and repress a key without getting a status code Ø (no key pressed) and status code 1 (different key pressed) before another status code -1.

```
1ØØ CALL KEY(Ø,K,S):: PRINT
K,S :: GOTO 1ØØ
```

George Steffen has responded to the challenge in the last Tips, by publishing in the LA 99ers Topics a remarkably compact routine to translate the internal format string representation of numeric data back into numbers. The following lines will update the Menu Loader accordingly.

```
1ØØ !by A. Kludge/M. Gordon/
T. Boisseau/J. Peterson/G. S
teffen/etc.Version #8, 11/85
14Ø @,@@,A,A$,B,C,D$,E,F,FLA
G,I,J,K,KD,KK,M,M$,N$,NN,P,P
$,PG$(),PP,PP$,Q$,S,ST,T$(),
TT,VT,V(,),W$,X,X$,Y,K2,S2
81Ø F=1 :: E=ASC(SEG$(M$,1,1
)):: M=ASC(SEG$(M$,2,1)):: I
F E=Ø AND M=Ø THEN GOTO 817
ELSE IF E>128 AND M>128 THEN
 F=-1 :: E=255-E :: M=256-M
815 FOR I=1 TO 6 :: M=M+(ASC
(SEG$(M$,I+2,1)))/1ØØ^I :: N
EXT I :: M=M*F*1ØØ^(E-64)
817 PRINT #PP:M
87Ø FOR P=1 TO NN-1 :: PRINT
 #2:PG$(P);TAB(15);V(P,3);TA
B(2Ø);T$(ABS(V(P,1)));TAB(25
);V(P,2);TAB(31);CHR$(89*ABS
(V(P,I)<Ø)):: NEXT P :: CLOS
E #2
```

The change in the last line is my own, because it was pointed out to me that the catalog output to the printer did not indicate protected files.
That last line is a good example of the power of relational expressions to accomplish compact programming. The variable V(P,1) picks up its value from the variable A which is read from the disk directory in line 35Ø. This is a number from 1 to 5, indicating the type of file, and if the file is write-protected the number is negative. A true expression has a relational value of -1. If the file is protected, V(P,1)<Ø is true, and its value is -1, converted by ABS to +1 and multiplied by 89 to give ASCII 89, converted by CHR$ to "Y". If not protected, V(P,1) is a positive number, V(P,1)<Ø is false and has a relational value of Ø; 89 times Ø is still Ø, and CHR$(Ø) prints nothing.

George also mentioned in a letter that my remarks on the UPDATE mode applied only to VARIABLE files; that RESTORE without a number,to return the record pointer to the beginning of a file, works only with VARIABLE files; that RESTORE with a number works only with RELATIVE files; and that therefore the only way to RESTORE a SEQUENTIAL FIXED file is to close it and reopen it.
On trying this out, I find that you can write to a FIXED SEQUENTIAL file and still be able to read the following records - but you can't simply "read a record, change it in some way, and then write the altered record back out on the file", as the Reference Guide indicates, because you will change the record FOLLOWING the one you read!
It is possible to UPDATE a FIXED SEQUENTIAL file without reading it all into an array and writing it back out, but you must read sequentially to the record you want, close the file, reopen the file, read back to the record just before the one you want to update, then write in the updated record.

I have received several other suggestions regarding the Menu Loader, too many to describe here. You can all modify it to your own tastes and needs. Remember to turn off the pre-scan and ON ERROR while you're working on it, then add any new variable names or CALLs to the pre-scan. And remember, that last line MUST be the LAST line of the program! You can resequence it higher, and change the GOTO accordingly, but don't put anything after it!
I did change my version to slash the zero, since this will carry over into a program that is loaded. If you do this, be sure to add a CALL CHAR to the list in line 15Ø!

```
19Ø CALL CLEAR :: FOR S=1 TO
 14 :: CALL COLOR(S,7,16)::
NEXT S :: CALL COLOR(Ø,2,16)
:: CALL CHAR(48,"ØØ3A444C546
444B8")
```

When you just want to load a program, waiting for it to be read from the disk directory can be a drag. And, you may have trouble recognizing the filename. So, here is the Tigercub Quickloader which I have placed on all my Collection Disks.
First you will need Catwriter, another program that writes a program. This

one will read the disk directory, ignore everything other than programs, ask you for a complete program name for each filename, and write all that into a MERGE format program called CATMERGE.

```
100 !CATWRITER by Jim Peters
on
110 OPEN #1:"DSK1.",INPUT ,R
ELATIVE,INTERNAL :: INPUT #1
:N$,A,J,K :: OPEN #2:"DSK1.C
ATMERGE",VARIABLE 163 :: LN=
1000 :: FN=1100
120 X=X+1 :: INPUT #1:P$,A,J
,B :: IF LEN(P$)=0 THEN 160
:: IF ABS(A)=5 OR ABS(A)=4 A
ND B=254 THEN 130 ELSE X=X-1
 :: GOTO 120
130 DISPLAY AT(12,1)ERASE AL
L:P$;"    PROGRAM NAME?" ::
ACCEPT AT(14,1)SIZE(25):F$
140 PRINT #2:CHR$(INT(FN/256
))&CHR$(FN-256*INT(FN/256))&
CHR$(147)&CHR$(200)&CHR$(LEN
(F$))&F$&CHR$(0):: FN=FN+1
150 M$=M$&CHR$(200)&CHR$(LEN
(P$))&P$&CHR$(179):: IF X<11
 THEN 120
160 IF M$="" THEN 180
170 PRINT #2:CHR$(INT(LN/256
))&CHR$(LN-256*INT(LN/256))&
CHR$(147)&SEG$(M$,1,LEN(M$)-
1)&CHR$(0):: LN=LN+1 :: M$="
" :: X=0 :: IF LEN(P$)<>0 TH
EN 120
180 PRINT #2:CHR$(INT(LN/256
))&CHR$(LN-256*INT(LN/256))&
CHR$(147)&CHR$(200)&CHR$(3)&
"END"&CHR$(0)
190 PRINT #2:CHR$(255)&CHR$(
255):: CLOSE #1 :: CLOSE #2
```

Next, key in the Quickloader. Do not change the line numbers, do not RESequence, because CATMERGE will be merged into the middle of it and that last line must be the last. Then, enter MERGE DSK1.CATMERGE and then SAVE DSK1.LOAD .

```
100 CALL CLEAR :: DIM M$(48)
:: CALL CHAR(94,"3C4299A1A19
9423C"):: CALL SCREEN(2):: F
OR SET=1 TO 14 :: CALL COLOR
(SET,15,1):: NEXT SET :: DIS
PLAY AT(1,4):"TIGERCUB QUICK
LOADER"
110 X=X+1 :: READ M$(X):: IF
 M$(X)<>"END" THEN 110
115 CALL PEEK(8198,A):: IF A
<>170 THEN CALL INIT
120 R=3 :: FOR J=1 TO X-1 ::
 READ X$ :: DISPLAY AT(R,1):
STR$(J);TAB(4);X$ :: R=R+1 :
: IF R<23 THEN 150
130 DISPLAY AT(24,1):"CHOICE
? OR 0 TO CONTINUE 0" :: ACC
EPT AT(24,26)VALIDATE(DIGIT)
SIZE(-2):N
140 IF N<>0 THEN 155 :: R=3
150 NEXT J :: DISPLAY AT(24,
1):"CHOICE?" :: ACCEPT AT(24
,9)VALIDATE(DIGIT):N
160 IF SEG$(M$(N),LEN(M$(N))
,1)="*" THEN DISPLAY AT(12,1
)ERASE ALL:"Return to BASIC"
: :"Type OLD DSK1."&M$(N)::
STOP
170 CALL CHARSET :: CALL CLE
AR :: CALL SCREEN(8):: CALL
PEEK(-31952,A,B):: CALL PEEK
(A*256+B-65534,A,B):: C=A*25
6+B-65534 :: A$="DSK1."&M$(N
):: CALL LOAD(C,LEN(A$))
180 FOR J=1 TO LEN(A$):: CAL
L LOAD(C+J,ASC(SEG$(A$,J,1))
):: NEXT J :: CALL LOAD(C+J,
0):: GOTO 30000
30000 RUN "DSK1.1234567890"
```

If you don't want to give your Basic-only programs a filename ending in an asterisk, you can leave out that warning routine, or you can modify it to warn of E/A or MiniMemory programs. If Catwriter has picked up any unloadable program-format files, etc., just delete them from the DATA lines.

The first issue of the GENIAL TRAVelER has arrived, and it is SUPERB! This is a magazine-on-a-disk, a SS/SD flippy loaded with 700 sectors of some of the finest articles and programs you'll ever see! And the programs are ready to run, you don't have to key anything in. The subscription price, until the end of 1985 at least, is $30 for 6 issues, which computes out to $5 per disk - many of you are paying your own user group that much for a one-sided disk of public domain!

If the subscribers will only have the guts to refuse to let their friends copy this for free, this venture will surely survive and contribute greatly to the advancement of the TI. The address is -

GENIAL COMPUTERWARE, 835 Green Valley Drive,
Philadelphia PA 19128.

Gene Burchfield asked if I had a program to print banners vertically. I had never heard of such a thing, so I wrote one.

```
100 DISPLAY AT(12,1)ERASE AL
L:"TIGERCUB STREAMER PRINTER
" !by Jim Peterson
110 DATA 0000,0001,0010,0011
,0100,0101,0110,0111,1000,10
01,1010,1011,1100,1101,1110,
1111
120 RESTORE 110 :: DIM B$(16
):: FOR J=1 TO 16 :: READ B$
(J):: NEXT J :: P$(0)=" " ::
 P$(1)=CHR$(230)
130 INPUT "TEXT TO BE PRINTE
D? ":T$ :: PRINT :: INPUT "P
RINTER DESIGNATION? ":PD$ ::
 OPEN #1:PD$
140 PRINT :: INPUT "SIZE? (1
-10) ":Z :: IF Z<1 OR Z>10 T
HEN 140
150 FOR J=1 TO LEN(T$):: A=A
SC(SEG$(T$,J,1)):: IF A=32 T
HEN GOTO 200
160 CALL CHARPAT(A,H$):: FOR
 W=1 TO 15 STEP 2 :: K$=SEG$
(H$,W,2):: FOR L=1 TO 2 :: L
$=SEG$(K$,L,1):: B=POS("0123
456789ABCDEF",L$,1)
170 M$=B$(B):: FOR M=1 TO 4
:: N=VAL(SEG$(M$,M,1)):: N$=
N$&RPT$(P$(N),Z):: NEXT M
180 NEXT L :: FOR Q=1 TO Z/2
+.5 :: PRINT #1:TAB((81-Z*8)
/2+.5);N$ :: NEXT Q :: N$=""
 :: NEXT W :: FOR R=1 TO Z/2
+.5 :: PRINT #1:"" :: NEXT R
190 NEXT J :: STOP
200 FOR T=1 TO Z*4 :: PRINT
#1:"" :: NEXT T :: GOTO 190
210 CALL KEY(0,K,S):: IF S=0
 THEN 210 ELSE RETURN
```

If your printer doesn't have the special characters of the Gemini, substitute 88 instead of 230 in line 120, to print X's, or whatever else you want. If you do have the special characters, try some others, such as 239, for this and other graphics printing programs. This routine will print a handy reference chart of them.

```
100 IMAGE ### #    ### #    ##
# #    ### #    ### #    ### #
110 P$=RPT$(CHR$(251)&CHR$(2
53),21):: X=0
120 OPEN #1:"PIO" :: PRINT #
1:CHR$(27);"E"
130 PRINT #1:P$:"  ASCII COD
ES FOR GEMINI SPECIAL CHARAC
TERS":P$
140 FOR J=160 TO 175 :: K=J-
X
150 PRINT #1,USING 100:K,CHR
$(J),K+16,CHR$(J+16),K+32,CH
R$(J+32),K+48,CHR$(J+48),K+6
4,CHR$(J+64),K+80,CHR$(J+80)
:: NEXT J
160 IF FLAG=1 THEN STOP ELSE
 FLAG=1 :: PRINT #1:"":"":P$
:"TI-WRITER CODES FOR GEMINI
 SPECIAL CHARACTERS":P$ :: X
=128 :: GOTO 140
```

Another one that just looks pretty -

```
100 !KALEIDOSPRITES by Jim P
eterson
110 CALL CLEAR :: FOR CH=100
 TO 128 STEP 4 :: FOR L=1 TO
 4 :: RANDOMIZE :: X$=SEG$("
00182430425A667E8199A5BDC3DB
E7FF",INT(16*RND+1)*2-1,2)
120 B$=B$&X$ :: C$=X$&C$ ::
NEXT L :: CALL CHAR(CH,RPT$(
B$&C$,4)):: B$,C$="" :: NEXT
 CH :: Z=2 :: CALL SCREEN(5)
130 CALL MAGNIFY(Z):: K=1 ::
 FOR J=1 TO 7 :: S=96+4*J ::
 R=16*J :: C=100*RND+20
140 IF J>5 AND Z=4 THEN T=5
:: GOTO 160
150 T=INT(15*RND+2):: IF T=5
 THEN 150
160 CALL SPRITE(#K,S,T,R,C,#
K+1,S,T,177-R,C,#K+2,S,T,R,2
41-C,#K+3,S,T,177-R,241-C)::
 K=K+4 :: NEXT J
170 Z=INT(2*RND+1)*2 :: GOTO
 130
```

```
100 !DISK MATCHER by Jim Pet
erson
110 DISPLAY AT(8,9)ERASE ALL
:"DISK MATCHER": : : :" To c
ompare a backup disk":"with
a master and list any":"file
s found on one but not"
120 DISPLAY AT(15,1):"on the
 other.": : : :"         Press
 any key"
130 CALL KEY(0,K,S):: IF S=0
 THEN 130
140 DISPLAY AT(12,1)ERASE AL
L:"INSERT MASTER - PRESS ENT
ER" :: CALL KEY(0,K,S):: IF
S=0 THEN 140
150 OPEN #1:"DSK1.",INPUT ,R
ELATIVE,INTERNAL :: INPUT #1
:D1$,A,J,K :: DIM F1$(127)
160 X=X+1 :: INPUT #1:F1$(X)
,A,J,B :: IF LEN(F1$(X))<>0
THEN 160 ELSE CLOSE #1
170 DISPLAY AT(12,1)ERASE AL
L:"INSERT BACKUP DISK": :"PR
ESS ENTER" :: CALL KEY(0,K,S
):: IF S=0 THEN 170
180 OPEN #1:"DSK1.",INPUT ,R
ELATIVE,INTERNAL :: INPUT #1
:D2$,A,J,K :: DIM F2$(127)
190 Y=Y+1 :: INPUT #1:F2$(Y)
,A,J,B :: IF LEN(F2$(Y))<>0
THEN 190 ELSE CLOSE #1
200 DIM F(127):: FOR J=1 TO
X :: FOR L=1 TO Y :: IF F2$(
L)=F1$(J)THEN F(L)=1 :: GOTO
 220
210 NEXT L :: PRINT F1$(J);"
 NOT ON BACKUP"
220 NEXT J
230 FOR M=1 TO Y :: IF F(M)=
0 THEN PRINT F2$(M);" NOT ON
 MASTER"
240 NEXT M :: END
```

A very useful tip from Jim Swedlow, in the Orange County ROM newsletter - INPUT respects any trailing print separator on a preceding PRINT command. Try it -

```
100 PRINT TAB(20);:: INPUT B
$
```

MEMORY FULL IN LINE 480
Jim Peterson

MERGE FILE EDITOR
Makes Programming Easier

By Michael C. Amundsen
New Horizons, January 1986

TI EDITOR IS GOOD, BUT

In the time I have spent writing TI BASIC and XBASIC programs, I have come to appreciate the TI Line Editor built into the console. If all the home computers, TI's Line Editor is about the best I've worked with. Few computers offer the easy editing of a single line (typing NUM XXX or EDIT XXX and using arrow keys, etc.) or the global resequencing of program lines (great when you have to insert a line later) that the TI Line Editor has. In fact, in many machines, you need to use a word-processor to generate your original textfile for the basic programs (goodbye automatic line-numbers!).

There are some times when I could use some more flexibility than the current TI Editor offers, though. There are four editing actions that I often need, but are not allowed by the built-in console editor. They are: 1) delete a series of lines (say a whole subroutine); 2) copy a series of lines to another file for use in other programs; 3) move a series of lines to another area in the same program (for example, move all data statements to the end of the program); and 4) delete only the REM lines to save memory space once the program is completed.

To meet my needs for a more flexible editor (and my need to continue to write programs!), I wrote a program called MFE (Merge File Editor) that allows the editing actions I described above. This program works only on XBASIC's MERGE Format files and requires a disk drive, expansion Memory and, of course, the XB cartridge. Below is a run-down of the capabilities of this small, but powerful programming aid.

WHAT THE MFE CAN DO

The MFE is great for doing little "spot-editing" in your programs. It allows you to copy or delete any line or sequence of lines in your program, delete only the comment lines, and resequence any line or group of lines including moving a group of lines from one part of the program to another. All these functions can be done on any BASIC or XBASIC program as long as it has been SAVEd in XBASIC's MERGE format.

DELETE-ing Lines

If you suddenly realize that the subroutine you just wrote is a duplicate of some other lines in your program, you could use the built-in editor to erase each line, one at a time (and sit and wait around!) or you could use the MFE to do it all at once.

MFE asks you what the starting and ending lines to delete are and then creates a new program file with the offending lines removed.

COPY-ing Lines

I often discover that the subroutine I need has already been written in some other program. Instead of getting the printout and sitting at the console typing the thing in again, I just use the MFE to copy the desired lines from the original program into another file for use in my new project. This saves time, effort and reduces the chance of typing errors in transferring the routine.

Deleting REM Lines

I tend to write a lot of comments in my programs as I am designing them. It helps me remember where I am headed when I come back to the project later on. But these comments use up precious memory and need to be removed to improve the speed of the program. I use the MFE to delete all 'REM' and '!' comment lines from my completed programs.

RESEQUENCING Lines

This is by far the most handy of the MFE functions. It allows me to outline a specific set of lines (say 1050-2015) and to resequence them using any starting line number (say 3000).

This may not seem handy at first, but I have come to love this feature of MFE. Below are some examples of the use of resequencing to help improve programs:

1 - KEEPING THINGS NEAT

I like to keep things easy to ready and edit when I write a program. I try to start all major routines with similar line numbers like 1000, 2000, 3000, etc. and I try to keep all line numbers in increments of 10.

When I am de-bugging, however, things get a bit messed up, discovering the need to add an extra line can mess up the line numbers, and using the TI editor to resequence can botch up my 1000, 2000, 3000 sections too!

I can use MFE to fix this, though. I can tell MFE to resequence lines 1000-1135 in increments of 10 (or 5, 20, etc.) starting at 1000. No other lines will be affected and every jump-reference (GOTO, GOSUB, etc.) will be adjusted if needed. Handy, eh?

2 - MOVING THINGS AROUND

The MFE can also move entire sections of code from one part of the program to another. How many times have you discovered you have just written some program code underneath an XBASIC Subprogram? The program won't run because all Subprograms must be at the end of the program code! How about when you wish you had put that subroutine at the end of the file instead or the middle? Or how about wanting to put all your DATA statements in one section instead of scattered throughout your program? Do you delete the code and write it all again in the proper place? Not if you have MFE.

With MFE you can move any line of code by just changing the starting address of the resequencing. For example, say I wanted to move the DATA statements now at lines 350-460 down to the end of the file at around 1500. All I need to do is tell MFE to resequence starting at 350 and ending at 460 and start the new line numbering at 1500 in increments of 10. MFE does the rest!

MFE DISK AVAILABLE

MFE has become a standard tool in my programming arsenal, and I highly recommend it for anyone who does a lot of BASIC and XBASIC programming.

A program disk including on-line instructions is available for $5 by contacting:

Michael Amundsen c/o SubFile99
POB 533, Bowling Green, OH 43402
CIS: 71706,625 STC: TI5361

...FORTH

```
SCR #28
( DOODLE #1 -TEXT -GRAPH -GRAPH2 )
( WESLEY R RICHARDSON  MARCH 1986 )
: IT ; BASE->R DECIMAL
Ø VARIABLE XP Ø VARIABLE YP Ø VARIABLE PP
: ALP1 CLS Ø 1Ø GOTOXY ." put ALPHA LOCK " ;
: ALP2 ." , then press ENTER" KEY DROP CLS ;
: SETV 64 XP ! 158 YP ! Ø PP ! ;   HEX
: ISPR 38ØØ ' SATR ! 38ØØ SSDT 1 MAGNIFY
  FFFF FFFF FFFF FFFF 15 SPCHAR 1Ø1Ø 28C6 281Ø 1ØØØ 16 SPCHAR
  8244 28ØØ 2844 82ØØ 17 SPCHAR F88Ø 8ØEØ 8Ø8Ø F8ØØ 18 SPCHAR
  88C8 E8A8 B898 88ØØ 19 SPCHAR FØ98 8888 8898 FØØØ 1A SPCHAR
  39 97 F 17 1 SPRITE 91 AE F 15 2 SPRITE B1 AE F 18 3 SPRITE
  C1 AE F 19 4 SPRITE D1 AE F 1A 5 SPRITE ; DECIMAL
: DA 9 DO DUP I SWAP DOT LOOP DROP ;
: DB 24Ø DCOLOR ! 173 242 DA 182 137 DA 191 242 DA ;
  -->

SCR #29
( DOODLE #2 )
: DC 8 * 17 + DUP 8 - DO 191 174 DO J I DOT LOOP LOOP ;
: DD 16 Ø DO I 16 * DCOLOR ! I DC LOOP DB ;
: INIT ALP1 ." up" ALP2 GRAPHICS2 SETV ISPR DRAW DD ;
: DLY Ø DO 1 1 / DROP LOOP ;
: SA CASE 4 OF  1 ENDOF Ø OF Ø ENDOF 252 OF -1 ENDOF ENDCASE ;
: SB CASE 4 OF -1 ENDOF Ø OF Ø ENDOF 252 OF  1 ENDOF ENDCASE ;
: SC SA XP @ + 9 MAX 242 MIN XP ! ;
: SD SB YP @ + 9 MAX 19Ø MIN YP ! ;
: PDOT PP @ IF XP @ YP @ DOT ENDIF ;
: PSPR PP @ IF 22 1 SPRPAT ELSE 23 1 SPRPAT ENDIF XP @ 7 -
  YP @ 7 - 1 SPRPUT ;
: FINISH DELALL TEXT ALP1 ." down" ALP2 CR CR
  ." type DOODLE to restart " CR CR
  ." type FORGET IT to end  " CR CR QUIT ;
  -->

SCR #3Ø
( DOODLE #3 )
: CJOY 1 JOYST SWAP SC SD ;
: ART 18 = IF PP @ Ø= PP ! ENDIF PSPR PDOT ;
: CCOL DUP 9 - 8 / DUP 2 SPRCOL DUP 2 < IF 15 1 SPRCOL ELSE
  DUP 1 SPRCOL ENDIF 16 * DCOLOR ! ;
: CSCR DUP 9 - 8 / SCREEN ;
: OPT PSPR 18 = IF XP @ DUP 137 < IF YP @ 182 < IF CCOL
  ELSE CSCR ENDIF ENDIF 176 > IF FINISH ENDIF ENDIF ;
: DOOOLE INIT BEGIN 3Ø DLY CJOY YP @ 173 < IF ART ELSE
  OPT ENDIF AGAIN ;
  DOODLE     R->BASE

SCR #89
( FORTH DISK  DIRECTORY )
( SCR #ØØ-2Ø  FORTH FORTHSAVE ERRORS )
( SCR #22-23  PYTHAGORAS -TEXT -PRINT )
( SCR #24-27  FINCALC  -FLOAT -PRINT )
( SCR #28-3Ø  DOODLE  -TEXT -GRAPH2 -GRAPH )
```

PEEKS & POKES

Reprinted from BYTE-LINE, January, 1986
Decatur 99er Users' Group

Be sure to do a 'CALL INIT'. The variables P and Q are used for CALL PEEK, and the numbers are for CALL LOAD.

ADDRESS	VALUE	MEANING IN EXTENDED BASIC
CALL VERSION(X)		IF X=100 THEN NEWEST VERSION OF XB
8192	P	USE (PEEK,P) IF P IS NOT 70 OR 121, THEN DO A CALL INIT
8194		FIRST FREE ADDRESS IN LOW MEMORY
8196		LAST FREE ADDRESS IN LOW MEMORY
-28672	P	IF P=0 SPEECH NOT ATTACHED, IF P=96 OR 255, SPEECH IS ATTACHED
-31572	0 TO 255	VARY KEYBOARD RESPONSE
-31740	P, Q	TO CHANGE BEEPS, WARNINGS, ETC.
	192	NO AUTO SPRITE MOTION OR SOUND
	224	NORMAL OPERATION
	225	MAGNIFIED SPRITES
	226	DOUBLE SIZE SPRITES
	227	MAGNIFIED & DOUBLE SIZE SPRITES
	232	MULTICOLOR MODE 48 BY 64 SQUARES
-31794	P	CALL SOUND TIMER, COUNTS 255-0
-31804	X, Y	USE PEEK(2,X,Y) FOR TITLE SCREEN
	P˙	CHANGE CURSOR FLASH RATE D-255
-31806	0	NORMAL OPERATION
	16	DISABLE QUIT KEY (FCTN =)
	32	DISABLE SOUND, USE NEGATIVE FOR CONTINOUS SOUND
	48	DISABLE SOUND & QUIT KEY
	64	DISABLE AUTO SPRITE MOTION
	80	DISABLE SPRITES AND QUIT KEY
	96	DISABLE SPRITES AND SOUND
	128	DISABLE SPRITES SOUND & QUIT KEY
-31808	P, Q	DOUBLE RANDOM NUMBERS (0 TO 255) NEED 'RANDOMIZE'
-31860	4	FROM X-BASIC TO BASIC NEED 'NEW'
	8	AUTO RUN OF DSK1.LOAD
-31866	P, Q	END OF CPU PRGM ADDRESS (P*256+Q)
-31868	0	NO 'RUN' OR 'LIST' AFTER 'BREAK'
	0, 0	TURNS OFF 32K MEMORY EXPANSION
	255, 231	TURNS ON THE 32K MEMORY EXPANSION
-31873	3 TO 30	'PRINT' SCREEN COLUMN TO START AT
-31877	P	P&32= SPRITE COINCIDENCE P&64= 5 SPRITES ON A LINE
-3187B	P	HIGHEST NUMBER SPRITE IN MOTION,
	0	STOPS ALL SPRITES
-31879	P	TIMER FOR VDP INTERRUPTS EVERY 1/6D OF A SEC (0 TO 255)
-31880	P	RANDOM NUMBER (0 TD 99), NEED 'RANDOMIZE'
-31884	0 TO 5	KEYBOARD MODE LIKE 'CALL KEY(K,,)'
-31888	63, 255	DISABLE ALL DISK DRIVES, USE 'NEW' TO FREE MEMORY
	55, 215	ENABLE ALL DISK DRIVES, USE 'NEW' TO FREE MEMORY
-31931	0	UNPROTECT X-BASIC PROTECTION
	2	SET 'ON WARNING NEXT' COMMAND
	4	SET 'ON WARNING STOP' COMMAND
	14	SET 'UNTRACE' COMMAND
	15	SET 'UNTRACE' & 'NUMBER' COMMAND
	16	SET 'TRACE' COMMAND
	64	SET 'ON BREAK NEXT' COMMAND
	128	PROTECT X-BASIC PROGRAM
-31952	P	PEEK IF P=55 THEN 32K EXPANSION MEMORY IS OFF, P NOT 55 MEANS ON
-31962	32	RETURN TO THE TITLE SCREEN
	255	RESTART X-BASIC WITH 'DSK1.LOAD'
-31974	P, Q	END OF VDP STACK ADDR. (P*256+Q)
-32112	8	SEARCHES DISK FOR ?
-32114	2	RANDOM GARBAGE
	13	SCREEN GOES WILD
	119	PRODUCE LINES
-32116	2	RANDOM CHARACTERS ON SCREEN
	4	GO FROM X-BASIC TO BASIC
-32187	0	UNPROTECT X-BASIC PROGRAM
	2	SET 'ON WARNING NEXT' COMMAND
	4	SET 'ON WARNING STOP' COMMAND
	9	SET 0 LINE NUMBER
	14	SET 'UNTRACE' COMMAND
	15	SET 'UNTRACE' & 'NUMBER' COMMAND
	16	SET 'TRACE' COMMAND
	64	SET 'ON BREAK NEXT' COMMAND
	128	PROTECT X-BASIC PROGRAM
-32188	1	SET COLOR & RECEIVE SYNTAX ERROR
	127	SET COLOR & RECEIVE BREAKPOINT
-32630	128	RESET TO TITLE SCREEN
-32699	0	UNPROTECT X-BASIC PROGRAM
	2	SET 'ON WARNING NEXT' COMMAND
	4	SET 'ON WARNING STOP' COMMAND
	14	SET 'UNTRACE' COMMAND
	15	SET 'UNTRACE' & 'NUMBER' COMMAND
	16	SET 'TRACE' COMMAND
	64	SET 'ON BREAK NEXT' COMMAND
	128	PROTECT X-BASIC PROGRAM
-32700	0	CLEARS SCREEN FOR AN INSTANT
-32729	0	RUN 'DSK1.LOAD'
-32730	32	RESET TO TITLE SCREEN
-32961	51	RESET TO TITLE SCREEN
	149	SET 'ON BREAK GOTO', LOCKS SYSTEM

ADDRESS	VALUE	MEANING IN E/A OR MINI-MEM
784	P	USE 'POKEV(784,P)' P=16 TO 31 TO CHANGE CURSOR BACKGROUND
-24574	8	? 24K STORAGE WITH MINI-MEM ?
-30945	0	WHITE EDGES
-32272	0, ** ,	-30945, 0 WILL CHANGE TO TEXT MODE
-32766	0	BIT MAP MODE
-32768	0	GRAPHICS (NORMAL MODE)
-32280	D	MULTI-COLOR MODE
-32352	107	BLANKS SCREEN, ANY KEY RESTORES

ADDRESS	VALUE	MEANING IN PASCAL
14586	0, 0	ALLOWS YOU TO DO A RUN-TIME WARM START FROM PASCAL TO BASIC

MILWAUKEE AREA 99-4 USER GROUP
4122 N. GLENWAY
WAUWATOSA, WI 53222

Edmonton 99UG
box 11983
Alberta
Canada
T5J 3L1