KINGS 99ers USERS GROUP
299 W. Birch Ave.
Hanford, Ca. 93230

Apr 86

Bill Mills, Pres. (209)582-1385

April Meeting:

Well at last we are back to our fourth Monday of the month
regular meeting (watch out for May, because of the holiday we'll need
to change again). Anyway, we will meet at 7:00pm on Monday. Let's all
show up and share what we know.

> KINGS 99ers USERS GROUP
> MONDAY, APRIL 28, 1986
> * * * * 7:00 * * * *
> 1255 Beulah Street, Hanford

March Minutes:

We had several visitors, the meeting was spent discussing
Jeani's and my trip to the L.A. TI Fest-West on March 1st. It was
great seeing 35 (yes, count them-35) booths selling and
demonstrating nothing but TI products. The really great thing was
meeting people like Craig Miller (of Miller's Graphics Inc.) in
person The really big announcement of the show was when Craig
introduced their newest product. They have a card for the expansion
box that will allow you to use any IBM (or IBM clone) board
(keyboards like this were selling for $55 at the computer faire).
Basically, you have all the advantages of the IBM keyboard with the
smarts of the TI, such things as dedicated curser, programable
function keys, numeric keyboard, noncramped keys, etc. Production is
expected this spring.

Information Wanted:

While at the TI Fest-West I saw a list of the "known" users
groups throughout the world. Since I have a daughter in Oklahoma, I
wrote to only listed group in Oklahoma, asking for membership info.
etc. to pass on to my daughter. Can you imagine my dissapointment
when my letter came back marked "no such number". If anyone knows of
any existing users groups in Oklahoma I'd appreciate it greatly if
you'd take the time to send me their address. Thanks, Bill.

Membership renewal:

A timely reminder-1986-87 dues are due soon. Next month's
newsleter will contain a renewal form.

Raffle Reminder:

Bring items to donate.

Gram Kracker:

If you are tired of inserting, changing cartridges, etc. I
haghly recommend you consider purchasing a Gram Kracker. For a
fantastic review see pages 4&5 of the April edition of Spirit Of 99
(Central Ohio).

Speech Tutorial:

If you have the TE II and the speech synthesizer and arn't
getting the maximum use out of them may I refer you to a very good,
short tutorial on pages 6 & 7 of the April issue of Spirit of 99.

DV/80 to Program:

Interested in running Display Varible 80 Ti Writer files as a
program? See page 50 ot the December MICROpendium.

Sh! :

Page 3 of the April OnLine (Edmunton, Canada) newsletter has a
short article on how to quiet that noisy expansion box fan.

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 d— all 'REM' and '!' comment lines from 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:

TIPS FROM THE TIGERCUB

#32

Copyright 1986

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 130 original programs in Basic and Extended Basic, available on casette or disk, only $3.00 each plus $1.50 per order for PPM. Entertainment, education, programmer's utilities. Descriptive catalog $1.00, deductable from your first order.

Tips from The Tigercub, a full disk containing the complete contents of this newsletter Nos. 1 through 14, 50 original programs and files, just $15 postpaid.

Tips from the Tigercub Vol. 2, another diskfull, complete contents of Nos. 15 through 24, over 60 files and programs, also just $15 postpaid. Or, both for $27 postpaid.

Nuts & Bolts (No. 1), a full disk of 100 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 100 utility subprograms in merge format, all new and fully compatible with the last, and with 10 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!

I've found a bug in the Tigercub Menuloader V.05 which won't let you print a disk catalog if the disk contains the maximum 127 files. This should fix it.
340 I=I+1 :: IF I>127 THEN K=X :: GOTO 430
520 DISPLAY AT(X+5,12)SIZE(12):" #?" :: ACCEPT AT(X+5,15)SIZE(3)VALIDATE(DIGIT):KD :: IF KD<1 OR KD>NN THEN 520

I think that all program listings should be printed in 28-column format, exactly as they appear on the screen - it makes it so much easier to key them in without errors. I combined parts of two of my programs to make

the following. It is written for the Gemini 10X but the lines of printer control codes are annotated to help others make adjustments.
100 DIM K$(240):: LN=100 :: DISPLAY AT(3,4)ERASE ALL:"TIGERCUB PROGLISTER": :" Will convert a program":"listing to 28-column format,"
110 DISPLAY AT(7,1):"exactly as it appears on the":"screen, and print it in 4":"columns."
120 DISPLAY AT(11,1):" Program must be RESequenced":"and LISTed to disk by":"RES (enter)":"LIST DSK1.(filename) (Enter)"
130 DISPLAY AT(18,1):"Filename? DSK" :: ACCEPT AT(18,14)BEEP:F$
140 OPEN #1:"DSK"&F$,DISPLAY ,VARIABLE 80,INPUT
150 IF EOF(1)=1 THEN 260 :: LINPUT #1:A$
160 IF LEN(A$)<80 THEN LN=LN+10 :: GOTO 210
170 LINPUT #1:B$ :: IF POS(B$,STR$(LN),1)=1 THEN FLAG=1 :: LN=LN+10 :: GOTO 210
180 A$=A$&B$ :: IF LEN(A$)<160 THEN LN=LN+10 :: GOTO 210
190 LINPUT #1:B$ :: IF POS(B$,STR$(LN),1)=1 THEN FLAG=1 :: LN=LN+10 :: GOTO 210
200 A$=A$&B$ :: LN=LN+10
210 S=1
220 L$=SEG$(A$,S,28)
230 IF L$<>"" THEN 240 :: IF FLAG=1 THEN FLAG=0 :: A$=B$ :: GOTO 160 :: ELSE GOTO 150
240 X=X+1 :: K$(X)=L$ :: S=S+28 :: IF X=240 THEN 250 :: GOTO 220
250 X=0 :: CALL PRINTER(K$()):: GOTO 220
260 CLOSE #1 :: FOR J=X+1 TO 240 :: K$(J)="" :: NEXT J :: CALL PRINTER(K$()):: PRINT #2:CHR$(12):: END
270 SUB PRINTER(B$()):: IF F=1 THEN 340 :: F=1
280 OPEN #2:"PIO.LF",VARIABLE 132 :: PRINT #2:CHR$(15);CHR$(27);"N";CHR$(6);!condensed print and perforation skip
290 PRINT #2:CHR$(27);"G";!

- double-struck printing, optional
300 PRINT #2:CHR$(27);CHR$(42);CHR$(0);!download normal characters - required if lines 310-330 are used
310 PRINT #2:CHR$(27);CHR$(42);CHR$(1);CHR$(48);CHR$(0);CHR$(64);CHR$(30);CHR$(96);CHR$(17);CHR$(72);CHR$(5);CHR$(66);CHR$(61);CHR$(0);!slash the zero - optional
320 PRINT #2:CHR$(27);CHR$(42);CHR$(1);CHR$(42);CHR$(0);CHR$(8);CHR$(34);CHR$(8);CHR$(0);CHR$(62);CHR$(0);CHR$(8);CHR$(34);CHR$(8);!broaden the asterisk - optional
330 PRINT #2:CHR$(27);CHR$(36);CHR$(1);!activate redefined characters - required if lines 310-320 are used
340 FOR C=1 TO 60 :: IF B$(C)="" THEN 360 :: PRINT #2:TAB(10);B$(C);TAB(41);B$(C+60);TAB(72);B$(C+120);TAB(103);B$(C+180);CHR$(10)
350 NEXT C
360 SUBEND

I had trouble in debugging that program because printing the control codes gave me unwanted line feeds, and using semicolons to prevent line feeds will interfere with tabs in the first line of text. An article by Art Byers in the Central Westchester UG newsletter gave me the solution - suppress all the line feeds by opening the printer with PIO.LF, and put them back in where you need them with CHR$(10)!

We haven't had a random music player in a long time. This one is called ECHO but I don't know where it came from.
100 RANDOMIZE :: DEF X=INT(RND#7):: FOR B=0 TO 6 :: A(B)=VAL(SEG$("247262294330349392440",(B+1)*3-2,3)):: NEXT B :: B,C,D=X
110 CALL SOUND(-900,A(B),0,A(C),9,A(D),19):: D=C :: C=B :: B=X :: GOTO 110

Sound effects - thanks to Greg Healy in the Edmonton User Group newsletter -
```
100 CALL INIT
110 FOR J=2000 TO 2300 STEP 10 :: CALL LOAD(-31568,J):: NEXT J
```

To go directly from XBasic to console Basic - thanks to Greg Healy in the Edmonton User Group newsletter -
```
CALL INIT :: CALL LOAD(-31962,8787)
```
Enter. Ignore the error message. Type NEW and Enter.
```
> TI BASIC READY
```

This routine will read a file of 28-character records and scroll them up the lower half of the screen without disturbing the upper half.
```
100 DISPLAY AT(12,1)ERASE ALL:"FILENAME? DSK" :: ACCEPT AT(12,14)BEEP:F$ :: CALL CLEAR
111 OPEN #1:"DSK"&F$,INPUT
112 DIM M$(400)
113 X=X+1 :: LINPUT #1:M$(X)
120 DISPLAY AT(24,1):M$(X)
125 R=24
130 FOR T=X-1 TO 1 STEP -1 :: IF R>13 THEN R=R-1 :: DISPLAY AT(R,1):M$(T)
140 NEXT T :: IF EOF(1)<>1 THEN 113 ELSE CLOSE #1
```

```
10 !ONE-LINE MORTGAGE PAYMENT CALCULATOR BY SAM MORABITO
100 CALL CLEAR :: INPUT "ENTER P,R,N WHERE P=AMOUNT, R=RATE, N=YEARS":P,R,N :: PRINT "$";INT((P*R/1200)/(1-1/(1+R/1200)^(N*12))*100+.5)/100;"PER MONTH"
```

A number always prints out with a blank space before and after it (except that a negative number is preceded by -). This is not always desirable when formatting a screen or printout. The solution is to change the number to a string by using STR$ -
```
100 CALL CLEAR
110 PRINT " MULTIPLICATION TABLES": :
120 FOR J=1 TO 9
130 FOR K=1 TO 9
140 PRINT TAB(K*3-2);STR$(J*K);
150 NEXT K
160 PRINT : :
170 NEXT J
```

Regarding the CHECKER program in Tips #31, I should have mentioned that the two programs to be compared must first be LISTed to one disk by -
```
LIST "DSK1.(filename)
```
- using a different filename for each.

We are still finding new ways to skin the kitty. In Tips #26 I listed three algorithms to alternate between the two joysticks. Rick Humburg sent me another which is the simplest and fastest of all -
```
100 Z=2
110 Z=3-Z :: CALL JOYST(Z,X,Y).......and back to 110!
```

Here are some more dark secrets Texas Instruments didn't tell us. The User's Reference Guide claims that the computer can produce frequencies up to 44733 Hz, "well above human hearing limits", but then admits "the actual frequency produced may vary from 0 to 10 percent depending on the frequency." According to Jim Hindley, the highest frequency actually produced is 37287 (which is certainly not above the hearing range of some humans, but neither is 44733!), and the maximum error rate far exceeds 10 % because any frequency you call for from 31953 to 43733 ends up as exactly 37287! Not to worry, the frequencies in the normal range of music are accurate enough and your TV speaker probably can't reproduce frequencies above 20000 anyway.
And did you know that TI really gave us only 13 values, not 30? Listen and count them -
```
100 FOR V=0 TO 29 STEP 2
110 CALL SOUND(1000,500,V)
120 CALL SOUND(1000,500,V+1)
130 FOR D=1 TO 500
140 NEXT D
150 NEXT V
```

And the duration values are just as inaccurate. Experimenting with a series of 8 CALL SOUNDs in a loop repeated 100 times, I found that execution time was 48 seconds for any duration between 1 and 49, or a negative duration; 54 seconds for any duration between 50 and 66; 67 seconds between 67 and 83; 80 seconds between 84 and 99; 94 between 100-116; 106 between 117-133....!
I guess I've been neglecting those who don't have the Extended Basic module, so -
```
100 CALL SCREEN(16)
110 CALL CLEAR
120 PRINT TAB(0);"GREENSLEEVES": : : : : : : : : : : : :"programmed by Jim Peterson"
130 DIM S(15)
140 FOR N=1 TO 12
150 READ S(N)
160 NEXT N
170 M$="4218##995ABDC324E7D8A5186699182498425A88DBC33A66A5243C7E8199428857E66DD3CA5423C187E423C8D5A818899FFC3"
180 RANDOMIZE
190 FOR R=1 TO 12
200 CALL COLOR(R+1,1,1)
210 CALL CHAR(32+R*8,CH$&CH$)
220 FOR T=R TO 25-R
230 CALL HCHAR(T,R,32+R*8,34-2*R)
240 NEXT T
250 NEXT R
260 CALL SCREEN(2)
270 FOR R=1 TO 12
280 CALL COLOR(R+1,R+2,1)
290 CH$=SEG$(M$,INT(17*RND+1)*2-1,8)
350 CALL CHAR(32+R*8,CH$&CH$)
310 NEXT R
320 DATA 247,277,294,311,330,370,392,440,494,523,554,587
330 DATA 2,5,5,4,7,5,2,8,5,3,9,5,1,10,1,2,9,3,4,8,3,2,6,3,3,3,1,1,5,3
340 DATA 2,6,1,4,7,5,3,5,2,1,4,2,2,5,2,4,6,1,2,4,4,4,1,1
350 DATA 2,5,1,4,7,5,2,8,5,3,9,5,1,10,5,2,9,5
360 DATA 4,8,3,2,6,3,3,3,3,1,5,3,2,6,3,3,7,5,1,6,2,2,5,1
370 DATA 3,4,1,1,2,2,2,4,1,4,5,1,2,1,5,6,5,1
380 DATA 2,12,9,2,12,7,2,12,3,3,12,12,1,11,9,2,9,7
390 DATA 4,8,6,2,6,3,3,3,3,1,5,5,2,6,3,4,7,5,2,5,3
400 DATA 3,5,5,1,4,4,2,5,5,4,6,1,2,4,1,6,1,1
410 DATA 6,12,9,3,9,12,1,11,8,2,9,7,4,8,6,2,6,3,3,3,3
420 DATA 1,5,3,2,6,2,3,7,5,1,6,6,2,5,5,3,4,1,1,2,2,2,4,4,6,5,1,1,1,5,7,5,1
430 FOR J=1 TO 223 STEP 3
440 READ T,A,B
450 GOSUB 530
460 FOR TT=1 TO T
470 CALL SOUND(-999,S(A),0,S(B),7)
480 NEXT TT
490 NEXT J
491 FOR V=0 TO 20
492 CALL SOUND(-999,S(A),V,S(B),V+7)
493 NEXT V
500 CALL SCREEN(INT(14*RND+2))
510 RESTORE 330
520 GOTO 270
530 CALL COLOR(A+1,INT(14*RND+2),1)
540 CALL COLOR(B+1,INT(14*RND+2),1)
550 RETURN
```

```
1 !from 9 T 9 UG newsl. Aug 85
100 PRINT """Hallo"" said TI"
110 PRINT "Press ""ENTER"" to continue"
```

If you bite the hand that feeds you, you'll go hungry tomorrow. Don't be a pirate!

MEMORY FULL TO BUSTIN'

Jim Peterson

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 130 original programs
in Basic and Extended Basic,
available on casette or
disk, only $3.00 each plus
$1.50 per order for PPM.
Entertainment, education,
programmer's utilities.
Descriptive catalog $1.00,
deductable from your first
order.
Tips from The Tigercub, a
full disk containing the
complete contents of this
newsletter Nos. 1 through
14, 50 original programs and
files, just $15 postpaid.
Tips from the Tigercub Vol.
2, another diskfull, com-
plete contents of Nos. 15
through 24, over 60 files
and programs, also just $15
postpaid. Or, both for $27
postpaid.
Nuts & Bolts (No. 1), a full
disk of 100 Extended Basic
utility subprograms in merge
format, ready to merge into
your own programs. Plus the
Tigercub Menuloader, a tuto-
rial 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 100 utility
subprograms in merge format,
all new and fully compatible
with the last, and with 10
pages of documentation and
examples. Also $19.95

postpaid, or both Nuts Bolts
disks for $37 postpaid.
Tigercub Full Disk Collec-
tions, 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!

    I found a bug in Nuts &
Bolts #2 which prevents
using HIGHCHAR after HEAVY-
CHAR. To fix it, remove the
write-protect tab,
MERGE DSK1.HEAVYCHAR
RES 21000,1
SAVE DSK1.HEAVYCHAR,MERGE
Replace write-protect tab.

    While they last, and the
supply is limited, I will
sell a single Texas Instr.
cassette interface cable for
$2.00 with any order for
cassette software.

    Did you ever wonder how a
computer sort actually
worked? This program will
let you actually see it in

action. It will also show
you the value being held in
the temporary variable T$,
and the total number of
swaps and comparisons made.
    Then you can change any of
the variables and resort.
Try AAA in the last position
or ZZZ in the first. You
will find that some of the
fastest sorts are not so
fast when a list is already
almost in sequence.

```
100 CALL CLEAR :: CALL SCREE
N(16):: FOR SET=2 TO 9 :: CA
LL COLOR(SET,5,16):: NEXT SE
T :: ON WARNING NEXT :: RAND
OMIZE
110 DISPLAY AT(21,1)ERASE AL
L:">>>TIGERCUB SORT WATCHER<
<<": :"Wait, please - genera
ting":"random array....." ::
DIM A$(101),B$(101),ST(25,2)
120 FOR J=1 TO 100 :: FOR L=
1 TO 3 :: B$(J)=B$(J)&CHR$(I
NT(26*RND+65)):: NEXT L :: X
=J :: A$(X)=B$(X):: GOSUB 32
767 :: NEXT J
130 DISPLAY AT(3,1)ERASE ALL
:"(1) BUBBLE SORT": :"(2) SH
AKER SORT": :"(3) SWAP SORT"
: :"(4) SHUTTLE SORT": :"(5)
 EASY SORT"
140 DISPLAY AT(13,1):"(6) QU
ICK SORT": :"(7) RESORT SORT
": :"(8) SHELL SORT": :"(9)
RESERVED": :"Type number of
choice"
150 ACCEPT AT(21,23)VALIDATE
(DIGIT)SIZE(2)BEEP:K :: IF K
<1 OR K>10 THEN 150
160 DISPLAY AT(24,1):"Size o
f array? (10-100)" :: ACCEPT
 AT(24,25)VALIDATE(DIGIT)SIZ
E(3):G :: IF G<1 OR G>100 TH
EN 160
170 ON K GOSUB 230,300,430,5
00,550,650,850,910,250000 ::
DISPLAY AT(22,1):W;"SWAPS":C
;"COMPARISONS" :: C,W=0
180 DISPLAY AT(24,1):"Choose
 (1)Menu or (2)Resort" :: AC
CEPT AT(24,7)VALIDATE("12")S
IZE(1):Q :: IF Q=1 THEN 130
190 DISPLAY AT(24,1):"Change
 which position? 0" :: ACCEP
T AT(24,24)VALIDATE(DIGIT)SI
ZE(-3):P :: IF P=0 THEN 210
ELSE IF P<1 OR P>G THEN 190
200 DISPLAY AT(24,1):"Change
 to?" :: ACCEPT AT(24,12)SIZ
E(3):A$(P):: X=P :: GOSUB 10
20 :: GOTO 190
210 DISPLAY AT(22,1):" ":" "
:: GOSUB 1010 :: N=G :: ON
K GOSUB 240,310,440,510,560,
660,860,920,25010 :: DISPLAY
AT(22,1):W;"SWAPS";C;"COMPA
RISONS" :: C,W=0 :: GOTO 180
220 REM *BUBBLESORT*
230 CALL CLEAR :: GOSUB 980
240 FOR J=2 TO N :: C=C+1 ::
 IF A$(J)>=A$(J-1)THEN 260
250 T$=A$(J):: GOSUB 1050 ::
 A$(J)=A$(J-1):: X=J :: GOSU
B 1020 :: A$(J-1)=T$ :: X=J-
1 :: GOSUB 1020 :: W=W+1 ::
F=1
260 NEXT J :: C=C+1 :: IF F=
0 THEN 280
270 W=W+1 :: F=0 :: W=W+1 ::
 N=N-1 :: GOTO 240
280 RETURN
290 REM *SHAKERSORT*
300 CALL CLEAR :: GOSUB 980
310 W=W+1 :: L=1 :: W=W+1 ::
 R=N
320 W=W+1 :: F=0 :: FOR J=L
TO R-1 :: C=C+1 :: IF A$(J)<
=A$(J+1)THEN 340
330 T$=A$(J):: GOSUB 1050 ::
 A$(J)=A$(J+1):: X=J :: GOSU
B 1020 :: A$(J+1)=T$ :: X=J+
1 :: GOSUB 1020 :: W=W+1 ::
F=1
340 NEXT J :: C=C+1 :: IF F=
0 THEN 410
350 W=W+1 :: R=R-1 :: C=C+1
:: IF R=L THEN 410
360 W=W+1 :: F=0 :: FOR J=R
TO L+1 STEP -1 :: C=C+1 :: I
F A$(J)>=A$(J-1)THEN 380
370 T$=A$(J):: GOSUB 1050 ::
 A$(J)=A$(J-1):: X=J :: GOSU
B 1020 :: A$(J-1)=T$ :: X=J-
1 :: GOSUB 1020 :: W=W+1 ::
F=1
380 NEXT J :: C=C+1 :: IF F=
0 THEN 410
390 W=W+1 :: L=L+1 :: C=C+1
:: IF L=R THEN 410
400 GOTO 320
410 RETURN
420 REM *SWAPSORT*
430 CALL CLEAR :: GOSUB 980
440 FOR J=1 TO N-1 :: W=W+1
:: R=J :: FOR JJ=J+1 TO N ::
 C=C+1 :: IF A$(R)<=A$(JJ)TH
EN 460
```

```
450 W=W+1 :: R=JJ
460 NEXT JJ :: C=C+1 :: IF R
   =J THEN 480
470 T$=A$(J) :: GOSUB 1050 ::
   A$(J)=A$(R) :: X=J :: GOSUB
   1020 :: A$(R)=T$ :: X=R :: G
   OSUB 1020
480 NEXT J :: RETURN
490 REM ***SHUTTLE SORT*****
500 CALL CLEAR :: GOSUB 980
510 FOR J=1 TO N-1 :: FOR JJ
   =J TO 1 STEP -1 :: C=C+1 ::
   IF A$(JJ)<=A$(JJ+1)THEN 530
   :: T$=A$(JJ) :: GOSUB 1050 ::
   A$(JJ)=A$(JJ+1) :: X=JJ :: G
   OSUB 1020
520 A$(JJ+1)=T$ :: X=JJ+1 ::
   GOSUB 1020 :: NEXT JJ
530 NEXT J :: RETURN
540 REM ****EASY SORT******
550 CALL CLEAR :: GOSUB 980
560 W=W+1 :: D=1
570 W=W+1 :: D=2*D :: C=C+1
   :: IF D<=N THEN 570
580 W=W+1 :: D=INT(D/2):: C=
   C+1 :: IF D=0 THEN 630
590 FOR J=1 TO N-D :: W=W+1
   :: Y=J
600 W=W+1 :: Z=Y+D :: C=C+1
   :: IF A$(Y)<=A$(Z)THEN 620 :
   : T$=A$(Y) :: GOSUB 1050 :: A
   $(Y)=A$(Z) :: X=Y :: GOSUB 10
   20 :: A$(Z)=T$ :: X=Z :: GOS
   UB 1020
610 W=W+1 :: Y=Y-D :: C=C+1
   :: IF Y>0 THEN 600
620 NEXT J :: GOTO 580
630 RETURN
640 REM *QUICKSORT*
650 CALL CLEAR :: GOSUB 980
660 W=W+1 :: L=1 :: W=W+1 ::
   R=N :: W=W+1 :: T=0
670 T$=A$(INT((L+R)/2)):: GO
   SUB 1050 :: W=W+1 :: J=L ::
   W=W+1 :: JJ=R
680 C=C+1 :: IF A$(J)>=T$ TH
   EN 710
690 W=W+1 :: J=J+1
700 GOTO 680
710 C=C+1 :: IF A$(JJ)<=T$ T
   HEN 730
720 W=W+1 :: JJ=JJ-1 :: GOTO
   710
730 C=C+1 :: IF A$(J)<>A$(JJ
   )THEN 760
740 C=C+1 :: IF J>=JJ THEN 7
   60
750 W=W+1 :: J=J+1 :: GOTO 7
   30
760 C=C+1 :: IF J>=JJ THEN 7
```

```
80
770 W=W+1 :: H$=A$(J):: A$(J
   )=A$(JJ):: X=J :: GOSUB 1020
   :: A$(JJ)=H$ :: X=JJ :: GOS
   UB 1020 :: GOTO 680
780 W=W+1 :: J=J+1 :: W=W+1
   :: JJ=JJ-1 :: C=C+1 :: IF J>
   =R THEN 800
790 W=W+1 :: T=T+1 :: W=W+1
   :: ST(T,0)=J :: W=W+1 :: ST(
   T,1)=R
800 W=W+1 :: R=JJ :: C=C+1 :
   : IF L<R THEN 670
810 C=C+1 :: IF T=0 THEN 830
820 W=W+1 :: L=ST(T,0):: W=W
   +1 :: R=ST(T,1):: W=W+1 :: T
   =T-1 :: GOTO 670
830 RETURN
840 REM ***RESORT SORT******
850 CALL CLEAR :: GOSUB 980
860 FOR J=2 TO N :: C=C+1 ::
   IF A$(J)>=A$(J-1)THEN 900
870 T$=A$(J):: GOSUB 1050 ::
   FOR L=J-1 TO 1 STEP -1 :: A
   $(L+1)=A$(L):: X=L+1 :: GOSU
   B 1020
880 C=C+1 :: IF A$(L-1)>=T$
   THEN 890 :: A$(L)=T$ :: X=L
   :: GOSUB 1020 :: GOTO 900
890 NEXT L
900 NEXT J :: RETURN
910 REM *SHELLSORT*
920 CALL CLEAR :: GOSUB 980
930 W=W+1 :: M=N
940 W=W+1 :: M=INT(M/3)+1
950 FOR J=1 TO N-M :: FOR JJ
   =J TO 1 STEP -M :: C=C+1 ::
   IF A$(JJ)<=A$(JJ+M)THEN 970
   :: T$=A$(JJ) :: GOSUB 1050
960 A$(JJ)=A$(JJ+M):: X=JJ :
   : GOSUB 1020 :: A$(JJ+M)=T$
   :: X=JJ+M :: GOSUB 1020 :: N
   EXT JJ
970 NEXT J :: C=C+1 :: IF M>
   1 THEN 940 :: RETURN
980 REM *RENEW ARRAY*
990 FOR J=1 TO G :: A$(J)=B$
   (J):: X=J :: M$=A$(J):: GOSU
   B 1020
1000 NEXT J :: N=G
1010 DISPLAY AT(24,1):"A to
   abort   P to pause" :: RETUR
   N
1020 RR=X
1030 IF RR>20 THEN RR=RR-20
   :: GOTO 1030
1040 CC=1-(X>20)*5-(X>40)*5-
   (X>60)*5-(X>80)*5 :: DISPLAY
   AT(RR,CC):A$(X);:: W=W+1 ::
   GOSUB 1050 :: RETURN
```

```
1050 DISPLAY AT(22,14):"T$="
   ;T$ :: W=W+1 :: GOSUB 1060 :
   : RETURN
1060 CALL KEY(3,K1,SS):: IF
   SS=0 THEN 1090
1070 IF K1=65 THEN 130
1080 CALL KEY(3,K2,SS):: IF
   SS<1 THEN 1080
1090 RETURN
```

Don't try timing these sorts, because the screen display distorts the speed. Option 9 has been left open so that you can add your own favorite sort routine, in the same format, starting in line 2500.

These routines may not be the most efficient forms, and their names may not be correct. If you know better ones, let me know!

```
100 !BASKET WEAVING by Jim P
eterson
110 CALL CLEAR :: W=11 :: T=
2 :: CH$="A5A5A5A5A5A5A5FF
00FF00FF00FF" :: CALL CHAR
(142,CH$):: CALL COLOR(14,2,
W,13,2,W):: CALL SCREEN(W)
120 CALL HCHAR(1,1,143,768):
: CALL CHAR(134,CH$):: CH=14
2
130 FOR C=1 TO 31 STEP T ::
FOR R=1 TO 23 STEP T :: CALL
HCHAR(R,C,CH):: NEXT R :: F
OR R=24 TO 2 STEP -T :: CALL
HCHAR(R,C+1,CH):: NEXT R ::
NEXT C
140 CH=ABS((CH=142)*135+(CH=
134)*143):: RANDOMIZE :: T=I
NT(3*RND+2)
150 FOR R=1 TO 23 STEP T ::
FOR C=2 TO 32 STEP T :: CALL
HCHAR(R,C,CH):: NEXT C
160 FOR C=31 TO 1 STEP -T ::
CALL HCHAR(R+1,C,CH):: NEXT
C :: NEXT R :: CH=CH-1 :: W
=INT(14*RND+3):: T=INT(3*RND
+2)
170 IF CH=134 THEN CALL COLO
R(13,2,W):: GOTO 130 ELSE CA
LL COLOR(14,2,W):: GOTO 130
```

The following routine will create a D/V80 file named GRAPHPAGE, to be loaded into TI-Writer as a 77x57 grid numbered along the left and bottom. Arrow keys can then be used to create a line graph of asterisks or whatever, annotated with text as desired.

```
100 OPEN #1:"DSK1.GRAPHPAGE"
,OUTPUT :: PRINT #1:TAB(4);R
PT$("_",75):: FOR J=1 TO 57
:: J$=STR$(J)
105 IF J<10 THEN J$=" "&J$
110 PRINT #1:J$&RPT$("_",38
)&"!" :: NEXT J
120 FOR T=1 TO 2 :: PRINT #1
:" ";:: FOR J=1 TO 77 :: J$
=STR$(J)&" " :: PRINT #1:SEG
$(J$,T,1);:: NEXT J :: PRINT
#1 :: NEXT T :: CLOSE #1
```

```
1 !TO PRINT A HANDY REFERENC
E CHART OF ASCII TO HEX CODE
- MODIFIED FROM READING-BERK
S AUG 85
90 OPEN #1:"PIO" :: PRINT #1
:CHR$(27);CHR$(77);CHR$(5)
100 FOR X=32 TO 63 :: FOR Y=
X TO X+64 STEP 32 :: CALL CH
ARPAT(Y,Y$):: PRINT #1:Y;" "
;CHR$(Y);" ";Y$;:: NEXT Y ::
PRINT #1:"" :: NEXT X
```

```
100 CALL CLEAR :: CALL MAGNI
FY(2):: RANDOMIZE :: DISPLAY
AT(3,2):"TIGERCUB SPEED TYP
ING TEST": :TAB(12);"SPEED"
:: T=10
110 DISPLAY AT(5,18):100-T :
: X=INT(26*RND+65):: CALL SP
RITE(#1,X,2,96,120):: FOR D=
1 TO T :: CALL KEY(3,K,ST)::
ON (K=X)+2 GOTO 120,130
120 T=T-1 :: GOTO 110
130 NEXT D :: T=T+1 :: GOTO
110
```

The UG newsletters are full of good editorials, reminding people that they had better pay for their freeware or there won't be anymore. I totally agree with that - but I can't help thinking that if there had been as much emphasis on paying for commercial software instead of pirating it, there would still be a lot more good programmers supporting the TI!

MEMORY FULL
Jim Peterson

The following program is the original TI MAILLIST. I have
modified it to Extended Basic. Disk drive is required. The
program will file and produce a print out for a maillist or a
print labels. If you need a print label program this will do it.
The program starts on line 490. If you do not want the header and
screen wipe you can start entry at 490. Those of you that start
at line 100 note that the header and screen wipe is from the
Tiger Tips, courtesy of Jim Peterson, published monthly in our
news letter. Also, note the program is in 28 column format, just
as you see it on the screen. Another Jim Peterson "Tiger Tip"

```
100 !***********************
110 !*                     *
120 !*    TI MAIL LIST      *
130 !*                     *
140 !*   Modified for ExB.  *
150 !*        by            *
160 !*    HARRY ALLSTON     *
170 !*                     *
180 !*     Loose files      *
190 !*                     *
200 !*   Save as:TI/MAIL/XB *
210 !*                     *
220 !*    Revision:3/3/1988 *
230 !*                     *
240 !***********************
250 !
260 CALL CLEAR :: CALL SCREE
N(6)
270 RANDOMIZE
280 DATA TI MAIL LIST,,Modif
ied for ExB,,by,Harry Allsto
n,,Loose Files,Save as:TI/MA
IL/XB,,TOUCH ANY KEY"
390 N$="1290665AC34208667E58
100995AC2A5E781428024086F023
142992100785AC3A53C24180FFF0
85AFFFEFF0099189100650010"
300 FOR P=1 TO 10
310 READ A$
320 PRINT TAB(15-LEN(A$)/2);
A$;" "
330 NEXT P
340 GOSUB 380
350 CALL KEY(0,K,ST)
360 IF ST=0 THEN 340
370 GOTO 490
380 CALL CHAR(128,SEG$(N$,IN
T(43*RND+1)*2-1,16))
390 Y=INT(14*RND+3)
400 Y=INT(14*RND+3)
```

```
410 IF Y=X THEN 490
420 CALL COLOR(19,X, )
430 CALL HCHAR(1,2,128,31)
440 CALL HCHAR(24,2,128,31)
450 CALL VCHAR(1,2,128,24)
460 CALL VCHAR(1,31,128,24)
470 RETURN
480 CALL HCHAR(1,3,128,768)
490 CALL CLEAR :: CALL SCREE
N(6)
500 FOR SET=0 TO 14
510 CALL COLOR(SET,15,6
520 NEXT SET
530 LINE$="==================
==========="
540 ON ERROR 2570
550 DIM LN$(50),NA$(50),AD$(
50),CP$(50),PC$(50)
560 DISPLAY AT(10,8)ERASE AL
L:"TI MAIL LIST" :: DISPLAY
AT(15,3):"PRESS ENTER TO BEG
IN..." :: GOSUB 2290
570 DISPLAY AT(20,1):"WHAT I
S THE NAME OF YOUR";"PRINTIN
G DEVICE? PIO" :: ACCEPT AT(
21,18)SIZE(-10)BEEP:P$
580 G$="PLEASE WAIT...

               WHILE PRINTER IS WOR
KING"
590 CALL SCREEN(6):: DISPLAY
AT(3,1)ERASE ALL:"MAIN INDE
X":"-----------"
600 DISPLAY AT(6,1):"1 View
name list";"2) Search for a
name";"3) Add names";"4) Ch
ange names"
610 DISPLAY AT(10,1):" 5) Del
ete names";"6) Alphabetize l
ist";"7) Save data file ";"8)
```

```
Load data file"
620 DISPLAY AT(14,1):"9) Pri
nt labels/list";"10) Finish
session"
630 DISPLAY AT(21,1):"SELECT
AN ITEM:" :: ACCEPT AT(21,1
6)SIZE(2)BEEP VALIDATE(DIGIT
);P :: IF P<1 OR P>10 THEN 6
30
640 CALL CLEAR
650 ON P GOSUB 670,750,910,1
050,1340,1510,1730,1860,2200
,2240
660 GOTO 590
670 T=0 :: FOR I=1 TO N :: T
=T+1
680 PRINT NA$(I),LN$(I);AD$(
I);CP$(I);PC$(I) : : : :
690 IF T<2 THEN 720
700 DISPLAY AT(21,1):"PRESS
(ENTER) TO CONTINUE"; :"PRES
S '9' TO RETURN TO MENU";
:: ACCEPT AT(23,28)BEEP:X$ ::
CALL CLEAR :: IF X$="9" OR
X$="9" THEN 740
710 T=0
720 NEXT I
730 DISPLAY AT(23,8):"END O
F FILE"; :" PRESS ENTER TO
CONTINUE" :: GOSUB 2290
740 RETURN
750 DISPLAY AT(2,1)ERASE ALL
:"LAST NAME? "; :"ENTER ALL
OR PART OF NAME:" :: ACCEPT
AT(6,1)BEEP:K$
760 FOR I=1 TO N :: IF POS(S
EG$(LN$(I),1,10),K$,1)=0 THE
N 880
770 DISPLAY AT(2,1)ERASE ALL
:"IS THE PERSON:"; :" ";NA$
(I);" ";LN$(I)
780 DISPLAY AT(9,1):"(Y-N)?
Y" :: ACCEPT AT(9,8)SIZE(-2)
BEEP VALIDATE("YNyn"):X$
790 IF SEG$(X$,1,1)="N" OR S
EG$(X$,1,1)="n" THEN 880
800 DISPLAY AT(11,1):NA$(I),
LN$(I);AD$(I);CP$(I);PC$(I)
810 DISPLAY AT(18,1):"DO YOU
WISH TO PRINT";"A LISTING O
```

```
R LABEL? (Y-N) Y" :: ACCEPT
AT(19,27)SIZE(-27)BEEP VALID
ATE("YNyn"):Z$
820 IF SEG$(Z$,1,1)="N" OR S
EG$(Z$,1,1)="n" THEN 860
830 OPEN #2:P$ :: ON ERROR 5
90
840 PRINT #2:TAB(5):NA$(I):"
";LN$(I);TAB(5);AD$(I);TAB(
5);CP$(I);" ";PC$(I); : : :
850 CLOSE #2
860 DISPLAY AT(23,1):"SEARCH
FOR MORE NAMES (Y-N)Y" :: A
CCEPT AT(23,28)SIZE(-28)BEEP
VALIDATE("YNyn"):X$
870 IF SEG$(X$,1,1)="Y" OR S
EG$(X$,1,1)="y" THEN 750 ::
GOTO 900
880 NEXT I
890 DISPLAY AT(10,1)ERASE AL
L:"THE ";K$;" YOU ARE SEARC
HING FOR";" IS NOT IN THIS
FILE." :: GOTO 860
900 RETURN
910 A=N+1 :: FOR I=A TO 50
920 DISPLAY AT(2,1)ERASE ALL
:"ENTER DATA: ";"#";I;" (M-
X:75)"
930 DISPLAY AT(4,1):"LAST NA
ME:"; : :"FIRST NAME:"; : :"
STREET ADDRESS:"; : :"CITY/S
TATE:"; : :"ZIP CODE:"
940 ACCEPT AT(5,1)BEEP:LN$(I
):: ACCEPT AT(8,1)BEEP:NA$(I
):: ACCEPT AT(11,1)BEEP:AD$(
I):: ACCEPT AT(14,1)BEEP:CP$
(I):: ACCEPT AT(16,10)BEEP:P
C$(I)
950 V=I
960 DISPLAY AT(3,1)ERASE ALL
:"ENTRY";"#";V
970 DISPLAY AT(4,1):"YOU ENT
ERED:"; :LN$(V);", ";NA$(V);
:AD$(V); :CP$(V); :PC$(V)
980 DISPLAY AT(23,1):"CHANGE
ANYTHING? (Y-N) N" :: ACCEP
T AT(23,24)SIZE(-24)BEEP VAL
IDATE("YNyn"):Y$
990 IF SEG$(Y$,1,1)="N" OR S
EG$(Y$,1,1)="n" THEN 1010
```

```
1000 C=N+1 :: CALL CLEAR :: GOSUB 1100
1010 DISPLAY AT(24,1):"ADD MORE NAMES?   (Y-N) N" :: ACCEPT AT(24,24)SIZE(-24)BEEP VALIDATE("YNyn"):X$
1020 N=N+1 :: IF X$="N" OR X$="n" THEN 1050
1030 NEXT I
1040 DISPLAY AT(22,1):"DATA FILE IS FULL....    PRESS ENTER TO CONTINUE" :: GOSUB 2290
1050 RETURN
1060 DISPLAY AT(2,1)ERASE ALL:"ENTER ALL OR PART OF NAME:" :: ACCEPT AT(4,1)BEEP:C$ :: FOR C=1 TO N+1
1070 IF POS(SEG$(LN$(C),1,10),C$,1)=0 THEN 1250
1080 DISPLAY AT(6,1):"IS THIS THE PERSON:";NA$(C):"   ";LN$(C)
1090 DISPLAY AT(9,1):"(Y-N)? Y" :: ACCEPT AT(9,8)SIZE(-8)BEEP VALIDATE("YNyn"):X$ :: IF X$="Y" OR X$="y" THEN 1100 ELSE 1250
1100 DISPLAY AT(9,1)ERASE ALL:"PRESS (#) TO CHANGE"
1110 DISPLAY AT(10,3):"1. LAST NAME:"  2. FIRST NAME:"  3. STREET ADDRESS"
1120 R=C
1130 R$="  *ENTER THE NEW DATA:"
1140 DISPLAY AT(13,3):"4. CITY/STATE:"  5. ZIP CODE:"  6. NO CHANGE"
1150 DISPLAY AT(23,1):"SELECT ONE OF THE ABOVE:" :: ACCEPT AT(23,25)BEEP:P :: CALL CLEAR
1160 IF P<1 OR P>5 THEN 1100
1170 ON P GOSUB 1280,1290,1300,1310,1320,590
1180 DISPLAY AT(15,1):"MORE CHANGES FOR:"  "  ";NA$(R):"   ";LN$(R)
1190 DISPLAY AT(20,1):"(Y-N)? N" :: ACCEPT AT(20,8)SIZE(-8)BEEP VALIDATE("YNyn"):Y$
1200 IF SEG$(Y$,1,1)="Y" OR SEG$(Y$,1,1)="y" THEN 1100
1210 DISPLAY AT(22,1):"CHANGE DATA FOR OTHER NAMES?"
1220 DISPLAY AT(24,1):"(Y-N) N" :: ACCEPT AT(24,7)SIZE(-7)BEEP VALIDATE("YNyn"):Z$ :: CALL CLEAR
1230 IF SEG$(Z$,1,1)="Y" OR SEG$(Z$,1,1)="y" THEN 1060
1240 RETURN
1250 NEXT C
1260 DISPLAY AT(10,1)ERASE ALL:"THE ";C$;"  YOU ARE SEARCHING FOR:"  IS NOT IN THIS FILE." :: GOTO 1210
1270 RETURN
1280 DISPLAY AT(2,1):"LAST NAME WAS:"  ;LN$(R):: GOSUB 2300 :: ACCEPT AT(4,1)SIZE(-25)BEEP:LN$(R):: RETURN
1290 DISPLAY AT(2,1):"FIRST NAME WAS:"; NA$(R):: GOSUB 2300 :: ACCEPT AT(4,1)SIZE(-25)BEEP:NA$(R):: RETURN
1300 DISPLAY AT(2,1):"ADDRESS WAS:"  ;AD$(R):: GOSUB 2300 :: ACCEPT AT(4,1)SIZE(-25)BEEP:AD$(R):: RETURN
1310 DISPLAY AT(2,1):"CITY/STATE WAS:"  ;CP$(R):: GOSUB 2300 :: ACCEPT AT(4,1)SIZE(-25)BEEP:CP$(R):: RETURN
1320 DISPLAY AT(2,1):"ZIP CODE WAS:"  ;PC$(R):: GOSUB 2300 :: ACCEPT AT(4,1)SIZE(-10)BEEP:PC$(R):: RETURN
1330 RETURN
1340 CALL SCREEN(9):: DISPLAY AT(2,1)ERASE ALL:"To prevent errors in the":"delete mode it is necessary":"to type the ENTIRE last name"
1350 DISPLAY AT(5,1):"exactly as it is in the data":"file.":"If you are not sure, return":"to the main menu by pressing"
1360 DISPLAY AT(9,1):"""R"",  Access the search mode.":"verify the correct name.":"return to the delete mode":"and continue."
1370 DISPLAY AT(15,1):"Press:" :: DISPLAY AT(17,4):"""R""  "to return to menu.":"  (ENTER) to continue."
1380 CALL KEY(0,K,S):: IF S=0 THEN 1380 :: IF K=82 THEN 590
1390 CALL SCREEN(6)
1400 DISPLAY AT(2,1)ERASE ALL:"LAST NAME?" :: ACCEPT AT(3,1)BEEP:X$ :: FOR I=1 TO N :: IF LN$(I)<>X$ THEN 1490
1410 DISPLAY AT(5,1):"IS THE PERSON:"  ";NA$(I):"   ";LN$(I)
1420 DISPLAY AT(10,1):"(Y-N)? Y" :: ACCEPT AT(10,8)SIZE(-8)BEEP VALIDATE("YNyn"):Y$ :: IF Y$<>"Y" OR Y$="y" THEN 1490
1430 A=I
1440 FOR D=A TO N
1450 LN$(D)=LN$(D+1):: NA$(D)=NA$(D+1)
1460 AD$(D)=AD$(D+1)
1470 CP$(D)=CP$(D+1):: PC$(D)=PC$(D+1)
1480 NEXT D :: N=N-1 :: GOTO 1500
1490 NEXT I
1500 DISPLAY AT(22,1):"MORE DELETIONS? (Y-N)? N" :: ACCEPT AT(22,23)SIZE(-24)BEEP VALIDATE("YNyn"):X$ :: IF X$="Y" OR X$="y" THEN 1400 :: RETURN
1510 DISPLAY AT(10,7):"PLEASE WAIT...":" ":"THE LIST IS BEING ARRANGED"
1520 B=1
1530 B=2*B
1540 IF B<=N THEN 1530
1550 B=INT(B/2)
1560 IF B=0 THEN 1640
1570 FOR Y=1 TO N-B
1580 X=Y
1590 I=X+B
1600 IF LN$(X)=LN$(I)THEN 1620 :: IF LN$(X)<LN$(I)THEN 1630 :: GOSUB 1670 :: Y=Y-B
1610 IF X>0 THEN 1590 :: GOTO 1620
1620 GOSUB 1650
1630 NEXT Y :: GOTO 1550
1640 RETURN
1650 IF NA$(X)<NA$(I)THEN 1660 :: GOSUB 1670
1660 RETURN
1670 N$=LN$(X):: LN$(X)=LN$(I):: LN$(I)=N$
1680 N$=NA$(X):: NA$(X)=NA$(I):: NA$(I)=N$
1690 N$=AD$(X):: AD$(X)=AD$(I):: AD$(I)=N$
1700 N$=CP$(X):: CP$(X)=CP$(I):: CP$(I)=N$
1710 N$=PC$(X):: PC$(X)=PC$(I):: PC$(I)=N$
1720 RETURN
1730 DISPLAY AT(5,1)ERASE ALL:"PRESS:"  (ENTER) TO CONTINUE.":" ":"""R"" TO RETURN TO MENU."
1740 DISPLAY AT(12,1):LINE$ :: DISPLAY AT(13,1):"**LOAD MODE**SAVE MODE**" :: DISPLAY AT(14,1):LINE$
1750 CALL KEY(0,KK,SS):: IF SS=0 THEN 1750 :: IF KK=82 THEN 590
1760 DISPLAY AT(17,1):"WHAT IS THE NAME OF YOUR ":"DATA FILE:DSK1.";&L$
1770 DISPLAY AT(22,1):"*NOTE*:THE FILE IN MEMORY IS:""";L$;"]"
1780 ACCEPT AT(19,18)SIZE(-27)BEEP:L$
1790 OPEN #1:"DSK1.";&L$,INTERNAL,OUTPUT,FIXED 192 :: ON ERROR 590
1800 PRINT #1:N
1810 FOR I=1 TO N
1820 PRINT #1:LN$(I),NA$(I),AD$(I),CP$(I),PC$(I)
1830 NEXT I
1840 CLOSE #1
```

```
1350 RETURN
1360 DISPLAY AT(5,1)ERASE ALL:"PRESS:";:" (ENTER) TO C
ONTINUE.";:" 'R' TO RETURN
 TO MENU."
1370 DISPLAY AT(12,1):LINE$
 :: DISPLAY AT(13,1):"**LOAD
 MODE***LOAD MODE**" :: DIS
PLAY AT(14,1):LINE$
1380 CALL KEY(0,KK,SS):: IF
 SS=0 THEN 1380 :: IF KK=82
THEN 590
1390 DISPLAY AT(17,1):"WHAT
IS THE NAME OF YOUR.";:"DAT
A FILE:DSK1."
1900 ACCEPT AT(19,16)SIZE(-2
7)BEEP:L$
1910 OPEN #1:"DSK1."&L$:INTE
RNAL,INPUT ,FIXED 150 :: ON
ERROR 590
1920 INPUT #1:N
1930 FOR I=1 TO N
1940 INPUT #1:LN$(I),NA$(I),
AD$(I),CP$(I),PC$(I)
1950 NEXT I
1960 CLOSE #1
1970 DISPLAY AT(9,1)ERASE AL
L:"THIS FILE HAS";N;"
ENTRIES.";:"50 ENTRIES IS M
AXIUM"
1980 DISPLAY AT(21,1):"PRESS
 ENTER TO CONTINUE" :: GOSUB
 2330
1990 RETURN
2000 DISPLAY AT(10,1)ERASE A
LL:"DO YOU WANT TO SET THE T
YPE STYLE FOR YOUR PRINTER?"
2010 DISPLAY AT(14,1):"(Y-N)
? Y" :: ACCEPT AT(14,3)SIZE(
-9)BEEP VALIDATE("YyNn"):ANS
$
2020 IF SEG$(ANS$,1,1)="Y" O
R SEG$(ANS$,1,1)="y" THEN GO
SUB 2390
2030 OPEN #1:"PIO" :: PRINT
#1:CHR$(27);CHR$(78):: CLOSE
 #1
2040 DISPLAY AT(10,1)ERASE A
LL:"PRESS A # TO:";:" 1 P
RINT MAILING LABELS";:" 2
 PRINT MAILING LIST";:" 3
 RETURN TO MENU" :: DISPLAY

AT(18,1):"SELECT ONE:"
2050 ACCEPT AT(18,12)BEEP SI
ZE(1):P
2060 IF P=3 THEN CALL CLEAR
:: GOTO 590
2070 IF P<1 OR P>3 THEN 2000
2080 DISPLAY AT(10,1)ERASE A
LL:G$ :: IF P<>1 THEN 2170
2090 FOR I=1 TO N STEP 2 ::
GOSUB 2120
2100 NEXT I
2110 RETURN
2120 OPEN #2:P$
2130 PRINT #2:TAB(5);NA$(I);
" ";LN$(I);TAB(41);NA$(I+1);
" ";LN$(I+1);TAB(5);AD$(I);T
AB(41);AD$(I+1)
2140 PRINT #2:TAB(5);CP$(I);
" ";PC$(I);TAB(41);CP$(I+1);
" ";PC$(I+1);
2150 CLOSE #2
2160 RETURN
2170 FOR I=1 TO N
2180 GOSUB 2200
2190 NEXT I :: RETURN
2200 OPEN #2:P$
2210 PRINT #2:LN$(I);" ";NA
$(I);TAB(33);AD$(I);TAB(60);
CP$(I);TAB(76);PC$(I)
2220 CLOSE #2
2230 RETURN
2240 DISPLAY AT(10,1)ERASE A
LL:"DO YOU WISH TO TERMINATE
";:"THIS SESSION (Y-N)? N"
 :: ACCEPT AT(12,21)SIZE(-2
)BEEP VALIDATE("YNyn"):X$
2250 CALL CLEAR :: IF X$<>"Y
" OR X$="y" THEN 590
2260 DISPLAY AT(10,1):"HAVE
A NICE DAY"
2270 FOR XX=1 TO 500 :: NEXT
 XX
2280 STOP
2290 CALL KEY(0,KK,SS):: IF
 SS=0 THEN 2290 :: RETURN
2300 DISPLAY AT(5,1):R$
2310 RETURN
2320 OPEN #2:P$
2330 !
2340 PRINT #2:TAB(5);NA$(I);
" ";LN$(I);TAB(5);AD$(I)
2350 PRINT #2:TAB(5);CP$(I);
" ";PC$(I);

2360 CLOSE #2
2370 RETURN
2380 CALL CLEAR :: OPEN #1:P
$ :: A$="1 ITALIC CURSIVE"
:: B$="2 STANDARD ELITE-LIST"
 :: C$="3 CORRESPONDENCE PIC
A"
2390 D$="4 CORRESPONDENCE EL
ITE-LIST" :: E$="5 PROPORTIO
NAL-LIST" :: F$="SELECT: 1
2 3 4 5" :: CALL CHAR(64,"00
3C3C7E7EFF18")
2400 ! ** DISPLAY MENU **
2410 DISPLAY AT(5,1):A$; :B$
; :C$; :D$; :E$; :TAB(5);F$
2420 DISPLAY AT(23,2):"SPACE
 to change else ENTER"
2430 C=15 :: DISPLAY AT(17,C
):"@" :: VALU=1
2440 GOTO 2530
2450 C=C+2 :: VALU=VALU+1 ::
 IF VALU>5 THEN 2430
2460 DISPLAY AT(17,1):" ";TA
B(C):"@" :: GOTO 2440
2470 ON VALU GOTO 2490,2490,
2500,2510,2520
2490 PRINT #1:CHR$(27);"B"
:: CLOSE #1 :: CALL CLEAR ::
GOTO 2040
2490 PRINT #1:CHR$(27);"E" :
: CLOSE #1 :: CALL CLEAR ::
GOTO 2040
2500 PRINT #1:CHR$(27);"H" :
: CLOSE #1 :: CALL CLEAR ::
GOTO 2040
2510 PRINT #1:CHR$(27);"Q" :
: CLOSE #1 :: CALL CLEAR ::
GOTO 2040
2520 PRINT #1:CHR$(27);"P" :
: CLOSE #1 :: CALL CLEAR ::
GOTO 2040
2530 CALL KEY(0,K,S):: IF S=
0 THEN 2530
2540 IF K=13 THEN 2470
2550 IF K=32 THEN 2450
2560 GOTO 2530
2570 CALL ERR(A,B):: DISPLAY
 AT(12,3)BEEP ERASE ALL:"ERR
OR CODE";A;"TYPE";B :: FOR X
=1 TO 700 :: NEXT X :: RETUR
N 590
```
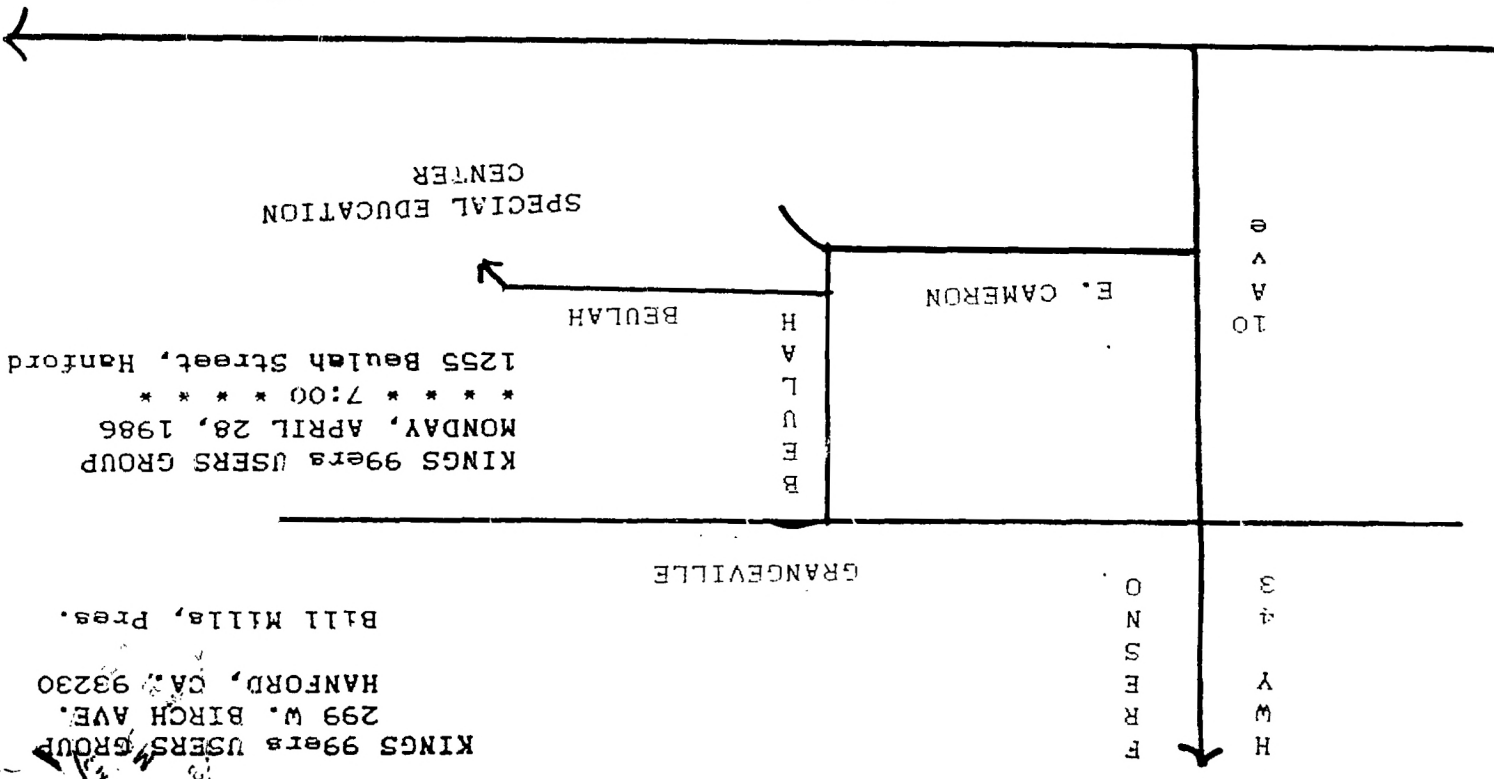
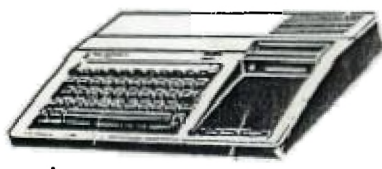HARRY ALLETON
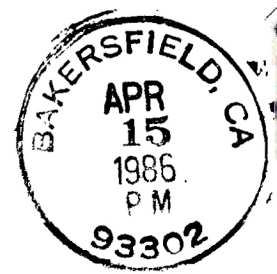
10300 KINGS RIVER ROAD #67

REEDLEY, CA 93654

209/638-7558