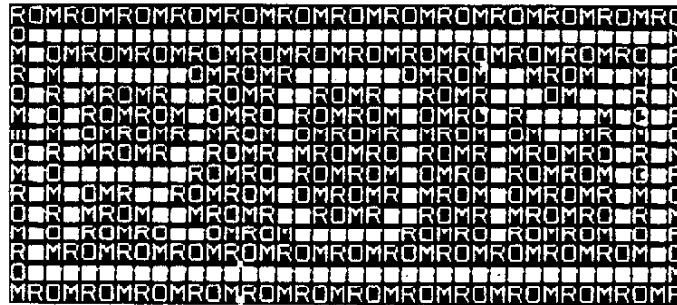


THE ROM NEWSLETTER  
 USERS GROUP OF ORANGE COUNTY  
 17301 SANTA ISABEL STREET  
 FOUNTAIN VALLEY, CA 92708

3/89 WELCOME  
 DALLAS TI COMPUTER GROUP (DTIHCG)  
 PO Box 29863  
 DALLAS,  
 TX 75229



APRIL 1989

SERVING THE TI 99/4A HOME COMPUTER COMMUNITY

## WE MEET AT MERCURY

TIME AND PLACE OF MEETING  
 The FIRST Thursday of each month at

MERCURY SAVINGS and LOAN  
 7:30 PM

West of Beach at 7813 Edinger Ave., Huntington Beach, Cal.  
 Use the WEST entrance. Park on the west side of the building. All are welcome.

### U.S.O.C. OFFICERS

PRESIDENT.....JIM SWEDLOW.....897-9209  
 VICE-PRES.....JIM JOLLY.....531-7782  
 SECRETARY.....RICHARD ATKINS.....637-7410  
 TREASURER.....JERRY RASH.....631-0579  
 PAST-PRES.....ROBERT HARPER.....744-2517

### COMMITTEE CHAIRMEN

JIM MORRIS.....MEMBERSHIP.....546-8354  
 SILES BAZERMAN...NEWSLETTER.....897-2868  
 BEN HATHEWAY....BULLETIN BOARD...751-4332

### SOFTWARE LIBRARY

KNUTE ERSLAND.....842-0859

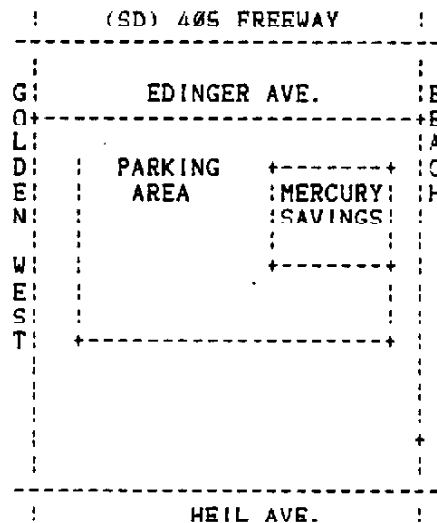
### NEWSLETTER LIBRARY

EARL RAGUSE....(Phone for time)...847-5875

### PUBLICATION NOTICE

The ROM is published by the Home Computer Users Group of Orange County, a non-profit organization. Permission is granted to similar groups to reproduce articles published in the ROM provided author and source credits are included. Opinions expressed in this publication are those of the author and do not necessarily represent those of the ROM, it's Editor, or the HCUGOC, it's officers or members.

We solicit letters and articles of interest to the TI-99/4A user community. Material accepted may be edited for fit and format. No payment is offered nor intended (other than your byline).



### NEWSLETTER CONTRIBUTORS

NEWT ARMSTRONG.....EDITOR EMERITUS  
 EARL RAGUSE.....PRODUCTION  
 EARL RAGUSE.....TI FORTH  
 ADRIAN ROBINSON.....ASSEMBLY  
 ROBERT SHAFFER.....CIRCULATION  
 JIM SWEDLOW.....AT LARGE  
 BILL NELSON.....GRAPHICS

### TI CLUB ACTIVITIES

CLUB	ACTION	DATE	INFO
BUG	GENERAL MEETING	MAY 01	532-1554
UGOG	GENERAL MEETING	MAY 04	897-9209
UGOC	LIBRARY, FTNVLY	MAY 08	842-0859
UGOC	ASSEMBLY SIG	MAY 11	537-1839
ET99	GENERAL MEETING	MAY 13	837-8757
UGOC	LIBRARY, FTNVLY	MAY 15	842-0859
UGOC	BOARD MEETING	MAY 18	897-9209
UGOC	ASSEMBLY SIG	MAY 25	537-1839
UGOC	NSLETTER LIBRARY	CALL	847-5875

## FROM THE PRESIDENT

By Jim Swedlow

### APRIL MEETING RECAPPED

Hope you were able to attend the April meeting. The hardware demonstrations were fascinating. Seeing the TI work with two hard drives was really impressive.

Special thanks to Frank Aylstock for bringing in his hard drives and controller and sharing them with us.

I would also like to thank Bob August for his demonstration of the SuperSpace cartridge. A very useful extension of EA.

### MAY MEETING WILL BE INTERESTING

While we are talking about meetings, May looks interesting. Of course we will go through our annual elections but beyond that we have:

- BOOT will be demonstrated by Ben Hatheway.
- GEE, a graphics language, will be put through its paces by Jerry Rash.
- GOLF will be played on our screen by Newt Armstrong.

Come and enjoy.

### LIBRARY UPDATES

Only two updates thus far for the May Library Update:

- MAGIC FILE MANIPULATOR version 2.1. Ben Hatheway's extraordinary program for moving files between two computers.
- MULTIPLAN FONTS lets you send printer controls from your Multiplan spread sheets.

### GENIAL TRAVELER

We will have Volume 2 Number 3 of GT available at the May meeting (I know we were supposed to have it in April but things ran over).

'Nuff Said.

=====

## RIBBON RE-INKING

We will have equipment available at the next meeting for reinking your printer ribbons. There will be a nominal charge for this service, so bring your old ribbons in and take advantage of this opportunity.

## UGOC HALL of FAME

This month we have THREE inductees to our Hall of Fame. All three are TI'ers who have made significant contributions to the TI Community. It is with great pleasure that we recognize their efforts:

Tony and Will McGovern: These two residents of Australia are best known for Funnelweb. This program, which has been proclaimed by some as "the most significant program ever written for the TI", takes TI writer to its maximum abilities and then adds features never consolidated into one program before Funnelweb. It is, simply put, an entire operating environment that will support just about any disk based TI application.

They have tweaked TI Writer and added new control keys, a better character font on screen, faster word wrap and much, much more.

The McGoverns have also contributed to the TI knowledge base. Their newsletter articles, which are often copies, cover many areas of interest for Extended BASIC and Assembly programmers.

Barry Boone: Barry has taken Barry Traver's Archiver from a basic program to one that is truly "elegant". The operation of Version 3.02 is simple but comprehensive. Once you understand the function of an archiver program, the learning curve is almost flat.

The availability of an Archiver is a key element in the electronic network that helps support the TI. Barry has also written many other fine programs.

Jim Peterson: This gentleman has given of himself to those of us who use the TI extensively and unselfishly. His "Tips from the Tigercub" have appeared in almost every TI news letter. You may not know this, but Jim distributes his material free of charge. He regularly sends care packages out to TI clubs at his own expense.

His "Nuts and Bolts" disks are full of programs, large and small. If you can think of an Basic or Extended Basic program, Jim has probably written it. Every one of his offerings is first rate.

His latest endeavor is distribution of TI programs to TI owners. He is charging only \$1.50 a disk (which barely covers expenses).

## AND NO FORTH #01

By Earl Raguse

Please note the title change, I was afraid the Editor would type cast me if I kept on writing about Forth, and wouldn't let me write about anything else, so I have temporarily switched to XBASIC, just to show that I could do it. Actually I was encouraged by Newt's new series of Basically Basic articles. I figured that after reading such instructive programs as he writes, you would probably get smart enough to read some of mine.

FIL0 was the first real program I ever wrote. I called it Lister, because I wrote it to keep track of lists of crossword puzzle words which I had jotted down while learning to do crossword puzzles. Yes, you have to learn how to work crossword puzzles too. Unless you are a lot smarter than I. I learned by looking at the answers the next day, then looking them up in the dictionary and writing them down, especially the the words I didn't know, and particularly the strange ones which always seemed to reappear, like AMIE, NEF and CLEW. At first this was only a one page list. but one day I discovered it had grown to SEVERAL pages, and it was not alphabetical!

A fine job for a computer I thought, and thus LISTER was created. I really got the idea after attending a meeting at the Santa Ana library, and watching Knute Erslund's fascinating demonstration on how speed up handling of address files on tape. LISTER worked crudely (and slowly) at first, but better and better as I learned. Then Knute sold me a Pbox and disc drive! Yipee!

Ultimately I realized that I had written a program that could handle many types of files, so I changed it to FILER. When I decided to update it for the ROM, I changed it to FIL0, its shorter, even if the program is long. I'm going to list it in installments, and explain as we go, not like Newt is doing, his stuff is planned, mine is not. FIL0 is not a sophisticated Data Base program, but it does do a lot of things. It doesn't use every XB word available, like error trapping etc., but if you succeed in getting completely through it you will learn a lot about XBASIC, and how to manipulate files, lists etc.

There are about a dozen stand-alone programs incorporated in FIL0, just waiting to be salvaged to do something in another program that you will probably write, once you learn XBASIC. There are Menus and full prompting throughout, Data Matrices, File Editing,

with insertion, deletion, changing, and appending, Disk Storage and Retrieval, File Sorting on any field, File Displaying, including forward and backward scrolling, and random alphabetical display of file records, File Printing with two selectable formats, File Searching for any word or part thereof, and Cataloging of your data disk if you forget the file name. Sorry, I didn't include file marking like MENU, that wasn't even thought of when I wrote the program. Maybe I'll make the an addendum.

I strongly recommend that you type in this program as we go and try to understand what each line is there for. You will, however, be able to get this program and a lot of the files (if you are a crossword) from the club library, if you just can't seem to get around to typing it in. However, there are possibly better programs, like PRBASE or TIBASE, if all you want is a data base program to use.

The main advantage of FIL0, is that its written is XBASIC, so if you can understand it you can change it to suit, particularly the printed output format. Also if you don't want to understand it, its quite simple to use and you really don't have to learn how to use it, just follow the prompts and read the MENUS. So much for the commercial, lets see what its all about. The first installment takes us through Menu, Creating a file and displaying it, and starts on sorting. Sorry, I didn't plan this, so there is no way to save and retrieve files. That should have come before displaying partial files and sorting, but thats the way I wrote it. So you will have to wait until next month to get the file save and recall routines. You can however check out the menus and file creation operations.

The Menu section was originally written with PRINT commands, the result was a lot of scrolling, and loss of speed. I did not intend to update that, but when I showed it to Newt Armstrong, to make sure I wasn't stepping on the toes one of his future articles, he subtly said, "You know there are ways of stopping all that scrolling." I took the hint. He also made as his first selection: 9. SEARCH FILE. He has instincts like that. He probably knew darn well that I hadn't protected against people trying to process nonexistent files. I fixed that too, you now get kindergarden level instructions if you try save, display, search or print a file you haven't yet created or loaded. Newt also gave me a lot of info on writing and reading files. We will talk about that later, along with other stuff I learned from writing FIL0.  
C U next time.

```

100 !SAVE DSK1.FILO
110 !TI XBASIC
120 CALL CLEAR
130 DISPLAY AT(10,5):"*****
*****"
140 DISPLAY AT(12,12):"FILO"
150 DISPLAY AT(14,8):"By Ear
! Raguse"
160 DISPLAY AT(15,6):"Early
85 Rev 4/89"
170 DISPLAY AT(17,2):"HARDLY
ANY RIGHT RESERVED"
180 DISPLAY AT(19,5):"*****
*****"
190 DISPLAY AT(22,8):"press
any key"
200 CALL KEY(3,K,S):: IF S=#
THEN 200
210 F#="filename"
220 HEAD#="printer heading h
ere"
230 FLAG=#
240 OPTION BASE 1
250 DIM A$(200,5)
260 CALL CLEAR
270 DISPLAY AT(9,2):" SELECT
FROM MENU"
280 DISPLAY AT(11,4):"1. CRE
ATE FILE"
290 DISPLAY AT(12,4):"2. EDI
T FILE"
300 DISPLAY AT(13,4):"3. DIS
PLAY FILE"
310 DISPLAY AT(14,4):"4. DIS
PLAY PART OF FILE"
320 DISPLAY AT(15,4):"5. SOR
T ALPHABETICALLY"
330 DISPLAY AT(16,4):"6. STO
RE FILE TO DISK"
340 DISPLAY AT(17,4):"7. REA
D FILE FROM DISK"
350 DISPLAY AT(18,4):"8. PRI
NT FILE"
360 DISPLAY AT(19,4):"9. SEA
RCH FILE"
370 DISPLAY AT(20,4):"0. CAT
ALOG DATA DISK"
380 DISPLAY AT(22,3):"INPUT
YOUR CHOICE"
390 IF FLAG=# THEN 410
400 DISPLAY AT(24,2)BEEP:"RE
MEMBER TO SAVE CHANGES"
410 CALL KEY(0,K,S)
420 IF K<48 OR K>57 THEN 410
430 CHOICE=K-48 :: CALL CLEA
R
440 IF CHOICE=# THEN CHOICE=
10 ELSE IF CHOICE>10 THEN 41
0
450 ON CHOICE GOTO 470,2220,
770,1050,1390,1610,1920,3060
,3640,3980
460 !
470 ! **CREATE A FILE**
480 !
490 CALL CLEAR
500 DISPLAY AT(10,1):"FILE N
AY CONTAIN ABOUT 200"
510 DISPLAY AT(12,1):"RECORD
S UP TO 5 LINES EACH"
520 DISPLAY AT(14,1)SIZE(25)

```

```

:"EACH UP TO 26 CHARACTERS "
530 DISPLAY AT(16,1)SIZE(21)
:" LINES PER RECORD? 5 "
540 ACCEPT AT(16,20)BEEP SIZ
E(-2):NL
550 DISPLAY AT(16,1)SIZE(30)
:"THERE ARE";NL;"LINES PER R
ECORD"
560 DISPLAY AT(20,1):" WHAT
IS THE PRINTER FILE HEADI
NG? EDIT THEN ENTER"
570 DISPLAY AT(24,1)BEEP SIZ
E(-64):HEAD#
580 ACCEPT AT(24,1)BEEP SIZE
(-64):HEAD#
590 CALL CLEAR :: DISPLAY AT
(1,10):"ENTER RECORDS-AFTER
LAST ONE!INPUT XX"
600 DISPLAY AT(14,0):" 1234
5678901234567890123456"
610 FOR I=N+1 TO 200
620 DISPLAY AT(12,1):"INPUT
RECORD > ";I
630 FOR L=1 TO NL
640 DISPLAY AT(15+L,0):L;" ";
650 ACCEPT AT(15+L,3)SIZE(-2
0)BEEP:A$(I,L)
660 IF A$(I,L)="XX" THEN 740
670 NEXT L
680 FOR LC=1 TO NL
690 DISPLAY AT(15+LC,0)SIZE(
30):
700 NEXT LC
710 NEXT I
720 CALL CLEAR
730 GOTO 750
740 A$(I,L)=" " :: N=N-1
750 PRINT "NO. OF RECORDS =
";N :: FLAG=1 :: GOTO 200 :H
ENU
760 !
770 ! **DISPLAY A FILE**
780 !
790 CALL CLEAR
800 IF FLAG=# THEN 420
810 PRINT HEAD# :
820 PRINT "HOLD SPACE BAR TO
SHOW FILE RELEASE TO
STOP"
830 PRINT "PRESS (R) TO REVE
RSE"
840 PRINT "PRESS (M) TO RETU
RN TO MENU": :
850 J=1 :: DIR=1
860 CALL KEY(3,K,S)
870 IF S=# THEN 860
880 IF K=32 THEN 920
890 IF K=82 THEN 910
900 IF K=77 THEN 260 ELSE 86
0
910 DIR=DIR*-1 :: J=J+2*DIR
920 IF (J>N OR J<1)THEN 1010
930 PRINT TAB(6);"***";J;"***"
940 FOR L=1 TO NL
950 IF NL>1 THEN PRINT L;
960 PRINT TAB(4);A$(J,L)
970 NEXT L
980 PRINT
990 J=J+DIR
1000 GOTO 860
1010 PRINT "WHAT NOW? (M) GO
TO MENU, OR (A) SEE AGAIN,

```

```

ENTER CHOICE > "
1020 CALL KEY(3,K,S)
1030 IF K=65 THEN 790 ELSE"1
F K=77 THEN 260 ELSE 1020
1040 !
1050 ! **DISPLAY PART FILE**
1060 !
1070 IF FLAG=# THEN 4210
1080 PRINT "LINE THE FILE IS
SORTED ON?"
1090 INPUT SL
1100 INPUT "ENTER START AND
STOP LETTERS X#,Y# > ":X#,Y#
1110 NUMX=ASC(X#)
1120 NUMY=ASC(Y#)
1130 CALL CLEAR
1140 J=1 :: DIR=1 :: Q=#
1150 IF J>N OR J<1 THEN 1350
1160 S#="SEG*(A$(J,SL),1,1)
1170 NUMS=ASC(S#)
1180 IF NUMS<NUMX OR NUMS>NU
MY THEN J=J+DIR :: GOTO 1150
1190 IF Q># THEN 1230 :: Q=1
1200 PRINT "HOLD SPACE BAR T
O SHOW FILE RELEASE TO STOP"
1210 PRINT "PRESS (R) TO REV
ERSE"
1220 PRINT "PRESS (M) TO RET
URN TO MENU": :
1230 CALL KEY(3,K,S):: IF S=
# THEN 1230
1240 IF K=32 THEN 1280
1250 IF K=82 THEN 1270
1260 IF K=77 THEN 260 ELSE 1
230
1270 DIR=DIR*-1 :: J=J+2*DIR
1280 PRINT TAB(6);"***";J;"*
*"
1290 FOR L=1 TO NL
1300 IF NL>1 THEN PRINT L;
1310 PRINT TAB(4);A$(J,L)
1320 NEXT L
1330 J=J+DIR
1340 GOTO 1150
1350 PRINT : ;"WHAT NOW? (M
) GOTO MENU, OR (A) AGAIN, E
NTER CHOICE > "
1360 CALL KEY(3,K,S)
1370 IF K=65 THEN 1100 ELSE
IF K=77 THEN 260 ELSE 1360
1380 !
1390 ! **SORT FILE**
1400 !
1410 IF FLAG=# THEN 4210
1420 INPUT "WHICH LINE TO SO
RT ON? > ":L
1430 CALL CLEAR :: DISPLAY A
T(12,4):"SORTING- BACK IN A
WHILE"
1440 FOR P=1 TO N
1450 X=N-P
1460 DISPLAY AT(20,2):X
1470 FLAG=#
1480 FOR I=1 TO X
1490 IF A$(I,L)<=A$(I+1,L)TH
EN 1560
1500 FLAG=1
1510 FOR K=1 TO NL
1520 V#="A$(I,K)

```

To Be Continued Next Month

# ASSEMBLY LANGUAGE

by Adrian Robinson

Many of us have at least a few CSGD/GR graphics files lying around. These are excellent printer graphics in a standard format of 5 by 5 characters. This month we have a little assembly program that will allow XBasic programmers to easily include these graphics in XBasic program screens. Simply assemble the program, load it with your XBasic program, and CALL LINK to it. It will load much more quickly if saved in XBasic format with a template program such as Dale Loftis' SAVER3 or Alsave, Systex, FWSAVE or one of the others.

A CSGD/GR file consists of a single record with three numeric fields and one string field of 200 bytes. We use only the string field which is comprised of 8 printer code bytes for each of 25 characters(5x5). The calling program must first construct an array (list) of 25 ASCII codes to be redefined as graphics characters. Then it reads the CSGD file and passes the string field and ASCII array in the CALL LINK statement:

```
CALL LINK("CSGD",GR$,ASC())
```

The Assembly program then converts the print codes to screen pattern definition codes and writes them to the pattern definition table to redefine the ASCII codes specified in the array. Then the XBasic program can display the graphic anywhere on the screen. A sample XBasic program is included.

So much for the utilitarian part. Now, let's talk about the interesting part. In the Sept. 88 ROM, I had a screen dump program. If you were to look closely at that program you might find a routine virtually identical to the CONVRT subroutine of the current program. As we all know, the screen image of a character is formed of an 8 by 8 array of pixels. The 8 pixel ROWS are defined by the bits of the 8 pattern table bytes. When we print the character in printer dot graphics mode, the 8 pixel COLUMNS are defined by the bits of the 8 printer codes. In essence, then, when we want to convert from screen image to printer codes, we form the Transpose of the screen image matrix (i.e. exchange rows and columns). But, if the resulting printer codes are passed back through the same conversion, the original screen pattern bytes will be returned. All of this is simply to say that the same routine can be used to convert codes in either direction.

```
*****
* CSGD/GR FILE CONVERSION *
* PRINTER CODES TO SCREEN *
* PATTERN DEFINITION CODE *
* Adrian Robinson *
* April 6, 1989 *
* EXTENDED BASIC *
* CALL LINK("CSGD",GR$,ASC()) *
* GR$ = CSGD/GR PRINT STRING *
* ASC() = LIST OF ASCII CODES *
*****
```

```

*
DEF CSGD
NUMREF EQU >200C
STRREF EQU >2014
XMLLNK EQU >2018
VMBW EQU >2024
FAC EQU >834A
MYWS BSS 32
PBUF BSS 8
ASC BSS 25
      BYTE 200
GRS BSS 200
      EVEN
*
CSGD LWPI MYWS
      CLR R0          Not an Array
      LI R1,1        First Parameter
      LI R2,GRS-1
      BLWP @STRREF   Get GR$ to GRS
*
      LI R2,ASC
      INC R1          Second Parameter
C1   INC R0          Array Element #
      BLWP @NUMREF   Get Array Element
      BLWP @XMLLNK   Convert Floating
      DATA >12B8   Point to Integer
      MOVB @FAC+1,*R2+ Store in ASC
      CI R0,25
      JLT C1
*
      LI R9,ASC      ASC Pointer
      CLR R8          GRS Index
C2   BL @CONVRT      Do Subroutine
      CLR R0
      MOVB *R9+,R0    ASCII Code
      SRL R0,5        Compute
      AI R0,>0300     PDT Address
      LI R1,PBUF      Pattern Buffer
      LI R2,8         Eight Bytes
      BLWP @VMBW      Write to PDT
      AI R8,8
      CI R9,ASC+25   End of List?
      JLT C2        No
*
      LWPI >83E0     GPLWS
      CLR @>837C     STATUS
      R @>70         Return to XBasic
*
CONVRT LI R5,PBUF    Subroutine
      LI R4,>8080     Converts Eight
      LI R3,>8000     Printer Codes
      LI R6,8        (One Character)
C3   CLR R2          to Eight Screen
      MOV R8,R1      Pattern Descriptor
      LI R7,8        Bytes
      CLR R0

```

(continued on Page 7)

## In My Humble Opinion

by Bill Nelson

I've never been much in math but I do know that if you start with 8 the next number will be 9. For this month GR09. NOW WATCH ME STEAL.

Boston Computer Society  
TI-99/4A User Group  
One Center Plaza  
Boston, MA 02108

Public domain software disk

RLE Pictures

This disk contains a collection of graphic pictures stored in TI-Artist format. They may be loaded and printed with the MAXRLE program by Travis Wattford, available from the BCS library. They may also be loaded, edited, and printed using TI-Artist, a commercial program. These pictures were provided by Barry Traver via a special interest group on Compuserve.

The above information comes from the -README file on the disk which leaves little else to say other than my humble opinion on the rest of the files.

1. ABC News is a poor picture of a news anchor with a smaller worse picture of President Reagan to his rear.

2. Bambi is a cute picture with Thumper, Flower, and Bambi posed in a poor background (to dark). The most redeeming thing about this picture is that you can edit out the back ground with TI Artist and have some excellent drawings of the three characters.

3. Coke: Looks like a half can (six oz. size) of the classic flavor coke can. Looks good to me.

4. Corvette, with a little work you could have a very good drawing of a Corvette for your instance collection.

5. Drop: Have you ever seen the milk commercial where their pouring a glass and as the last drop hits there is a ripple across the top of the glass and that drop seems to hang there above the surface just for a moment. Now you have the picture. Very good, in fact I had to get up and get a glass as I contemplated this picture.

6. Ferrari: The standard side shot of a vehical. The picture has more detail than the Corvette and all that was said about it applies here also.

7. Gator: This looks more like a dragon (puff like) to me with a tee-shirt on sitting in front of a typewriter. I

thought of me. Tis an excellent drawing I could see in use in headings or whatever for the humorist effect.

8. Led Zepplin, this picture is poor at best. While the lettering is unique, I'm very unimpressed with the drawing.

9. Middle Earth: This is an outstanding drawing. Many times I use pictures to dress up my letters and this will be in the next letter going out. It is an landscape with trees, set in rolling hills and rocks in the foreground. My words do not do this picture justice.

10. Mac??? Why? Don't we do our pounding on TIs? It's out of there. Besides, I don't like it.

11. Pagoda: This drawing makes me take back my words on ranking pictures. Even though I can't find a use for this picture, it's excellent. A white on black drawing of a pagoda set in trees and two white birds in the foreground.

12. Pirate: Another Excellent drawing. A buccaneer complete with musket and parrot holding the disk(?) that he's pirated.

13. Rocky Horror: Yuck! A very poor reproduction of the a poster for the movie.

14. The Wall: Are you into what looks like a brick wall with graffiti, (Pink Floyd the Wall) on it? I'm not. This one gets a poor score.

Over all I feel this this disk of info is well worth the \$2.00 we pay the library to copy it for us.

If you like this column let me know and I'll keep it up, I mean it's not everyday I get away with saying things like I do in my humble opinion.

=====

## MEMBERSHIP CORNER

BY Jim Morris

Membership is currently at seventy one. Eleven members are on ninety day hold and twelve members are currently due.

A large number of updated Library lists were passed out at the last meeting. There are a number of excellent programs. available from our library so please patronize it if at all possible. Bulletin Board users also have a number of excellent programs available for download.

Don't forget we do need new members and you can really do a friend a favor if you ask him to join.

## BASICALLY BASIC

by N. Armstrong

Now that we have the data stored in a separate file, we will have to write algorithms to delete and add names and numbers. This won't be elegant, more like brute force.

The hard part is maintaining the integrity of the array number sequence. For instance; if we have 25 names on the list, we nominally will have elements 1 through 25 filled. If we delete the name in the 15th element, do we then renumber elements 16 through 25, or do we put up with a blank in the middle of our array? Let's renumber them.

Statements with the asterisks are new or changed. In 190 we have Add and Delete options. Statement 215 uses results of a keyboard poll to direct program flow. For the ADD option (400-450), new names/numbers are appended at the end of the file. For the DELETE option (500-540), the name/number most recently displayed is deleted and subsequent elements are renumbered. And when the STOP option is taken (330-370), updated data is stored in the file, FONE.

```
100 REM SIMPLE PHONE LIST
    MOD BY N.ARMSTRONG
110 DIM NA$(50)
120 OPEN #1:"DSK1.FONE"
130 N=N+1
140 INPUT #1:NA$(N)
150 IF EOF(1)THEN 170
160 GOTO 130
170 CLOSE #1
180 CALL CLEAR
*190 PRINT "PRESS SPACEBAR TO
    SHOW NAMES": " " OR A
    TO ADD D
    TO DELETE": :
200 CALL KEY(3,K,K)
*210 ON 1-(K=32)-2*(K=85)-3*(
    K=68)GOTO 200,220,400,500
220 X=X+1
230 P=POS(NA$(X),"#",1)
240 N#=SEG$(NA$(X),1,P-1)
250 A#=SEG$(NA$(X),P+1,LEN(
    NA$(X))-P)
260 PRINT N#,A# :
270 IF X<N THEN 200
280 X=0
290 PRINT "END OF LIST PRE
    SS <R>EPEAT
    OR <S>TOP"
300 CALL KEY(3,K,K)
310 IF K=82 THEN 180
320 IF K<>83 THEN 300
*330 OPEN #1:"DSK3.FONE1"
*340 FOR I=1 TO N
*350 PRINT #1:NA$(I)
*360 NEXT I
*365 CLOSE #1
```

```
*370 STOP
*400 CALL CLEAR
*410 INPUT "NAME,NUMBER XXX X
    XX XXXX " :N#,A#
*420 NA$(N+1)=NA$(N)
*430 NA$(N)=N#&"#"&A#
*440 N=N+1
*450 GOTO 180
*500 FOR I=X TO N
*510 NA$(I)=NA$(I+1)
*520 NEXT I
*530 N=N-1
*535 X=X-1
*540 GOTO 180
```

=====

continued from Page 5

```
C4 MOVB @GRS(1),R0 (See Text)
    COC R3,R0
    JNE $+4
    SOCB R4,R2
    SRC R4,1
    INC R1
    DEC R7
    JNE C4
    MOVB R2,*R5+
    SRL R3,1
    DEC R6
    JNE C3
    RT
*
    END
```

```
10 ! CSGD GRAPHICS
15 ! SCREEN DISPLAY
20 ! Adrian Robinson
25 ! April 12, 1989
30 !
35 DIM A(25):: FOR I=1 TO 25 ::
    A(I)=96+I :: NEXT I :: CALL CLEAR
    :: FLG=0
40 CALL SCREEN(5):: FOR I=0 TO 12
    :: CALL COLOR(1,16,1):: NEXT I
45 A$(1)="abcde" :: A$(2)="fghij"
    :: A$(3)="klmno" :: A$(4)="pqrst"
    :: A$(5)="uv wxy"
50 CALL INIT :: CALL
    LOAD("DSK1.CSGD"):: CALL CLEAR
55 DISPLAY AT(10,1)SIZE(17):"FILE
    NAME: DSK1."
60 ACCEPT AT(10,16)SIZE(-12):X$ ::
    F$="DSK"&X$&"/GR"
65 OPEN #1:F$,INPUT
    ,INTERNAL,VARIABLE 254 :: INPUT
    #1:X,X,X,GR$ :: CLOSE #1
70 CALL LINK("CSGD",GR$,A()):: IF
    FLG=1 THEN 80
75 FOR I=1 TO 5 :: DISPLAY
    AT(14+I,11):A$(I):: NEXT I :: FLG=1
80 CALL KEY(3,K,S):: IF S=0 THEN 80
    ELSE IF K<>13 THEN 55
```

## TI BITS \* Number 27

By Jim Swedlow

### MORE DEBUGGING

I mentioned the story about the origin of "debugging" to a fellow in Canada. His response was a long and sad tale:

"Yes, I did know about 'debugging'. Fifteen years ago my company decided to do a lot of the alarms to our Central Station on a computer. This unit was six feet high. I knew NOTHING about computers except that, at that time, the office environment had to be kept within close bounds. I was chosen. They said, 'You are the most careful supervisor in Ontario'.

"I asked for air conditioning and an electronic air cleaner. 'Too expensive', they said. After again reading from the instructions given me, I showed them the pertinent info. 'H'mm', they said. Said I, 'This unit has seven filters that are to be changed each month'. Said they, 'Too expensive, just clean the old ones'. "Said I, 'We must make grandfather disks in case we have problems'. 'Oh, we don't think so - that is just their way of making the whole project more expensive'.

Lastly, I told them that there was a week of training to be completed in Trenton and at least two of our people should attend. Said they, 'You are too valuable to lose for a whole week -- just pick out two of your people that you can afford to lose for that period'.

"I did and told them that the system would never work at all if this was how they were going to operate and that I disavowed any problem that reared it's head. \$950,000 later they threw their hands in the air and told the computer people that their stupid computer was no good.

"Yes Jim, I did find bugs in the unit as well as bugs in my bosses at Head Office."

### GRAPH

The other night my 13 year old just had to have graph paper for her home work. Never mind that she knew that she would need some paper for some time; she had forgotten to tell us and the need was urgent.

I remembered that I had an XB program that printed graph paper. To my surprise, I found it fairly quickly. I ran it and it worked - almost. It printed graph paper but the boxes were wider than they were long. It sufficed for her assignment but I had to fix the program.

Here it is:

```
100 ! GRAPH
110 E$=CHR$(27)
120 A$=RPT$(CHR$(128),228)
130 B$=RPT$(CHR$(255)&SEG$(A$,1,6),8)
140 B$=RPT$(B$&CHR$(255),4)
150 A$=E$&"K"&CHR$(228)&CHR$(0)&A$
160 B$=E$&"K"&CHR$(228)&CHR$(0)&B$
170 OPEN #1:"PIO.CR"
180 FOR I=1 TO 11
190 PRINT #1:E$;"@";E$;"3";CHR$(16)
200 FOR J=1 TO 8
210 PRINT #1:B$;B$;CHR$(10)
220 NEXT J
230 PRINT #1:A$;A$;E$;"3";CHR$(2)
240 NEXT I
250 PRINT #1:RPT$(CHR$(13)&CHR$(10),9)
260 PRINT #1:E$;"@"
270 CLOSE #1
```

This program will work with MOST Gemini and Epson compatible printers. There are two printer commands that can cause you problems.

The first one is <E\$;"@"> (remember that E\$ is defined in line 110 as CHR\$(27) or Escape) which appears in lines 190 and 260. This is a reset command that tells your printer to restore its default settings. Earlier Epson compatibles (including the TI Impact Printer) do not recognize this command. If you get garbage with some "@" characters, yours doesn't either. These embedded reset commands cause your printer to completely lose any idea of where the top of form is so you will have to manually reset it.

The printer command that caused my problem is <E\$;"3";CHR\$(16)> in line 190. For the Gemini 10X and most Epson MX compatible printers, this sets line height to 16/144 (or .1111) inch. For the Star NX10, NX1000, and most Epson FX compatibles, it sets the line height to 16/216 (or .0741) inch. Hence the squat squares. I changed the line to read <E\$;"3";CHR\$(24)>. This set line height at 24/216 (or .1111) inch and everything worked correctly.

The difference for the <E\$;"3";CHR\$(2)> in line 230 is so small (.0046 inch) that it makes no difference. If you really wanted to, you could change it to <E\$;"3";CHR\$(3)>.

The problem with CHR\$(27) "3" CHR\$(n) raises its ugly head in a number of programs. Most folks wrote for the Gemini 10X/Epson MX family. This change, which came with the NX/FX lines of printers, has not received wide attention.

If your printout lines are too close together, look for this code and increase "n" by a factor of 1.5. You can, that is, if it is an XB program.

Enjoy.