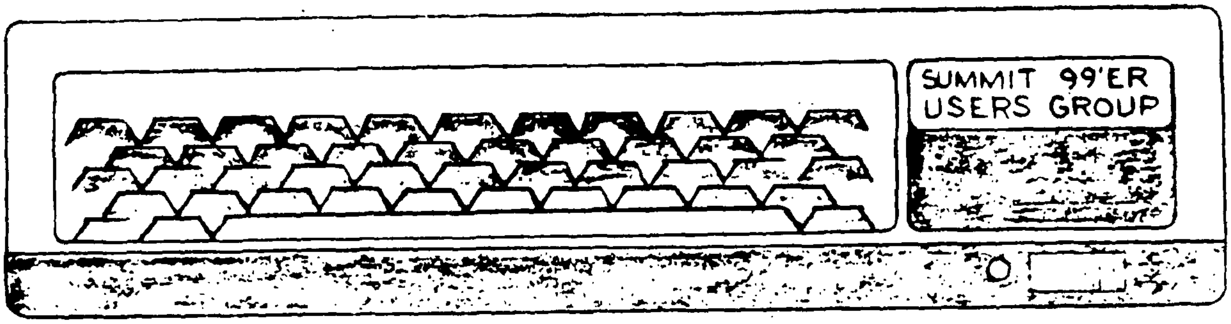

NOVEMBER 1984 Vol. 2 No. 11

The November meeting will be held on Thursday, November 15 at Cuyahoga Falls High School at the corner of Fourth and Stow Streets in Room -413 - Physic's Lab. The December meeting will be held on Thursday, December 20. Please remember to sign in.

NEW ADDRESS

For any correspondence or exchanging of newsletters please write to us at the following address:

Summit 99er Users Group
P.O. Box 3201
Cuyahoga Falls, Ohio 44223

BEGINNER'S CLASS

Rich will be covering chapter 2 in the Beginner's class. Those interested should bring their Blue book with them.

We would like to set up programs for the coming month's meetings. If you have any ideas or know of someone who would like to give a program, contact John Tuesday.

Is there any interest in setting up an Extended Basic class? If you are interested in either taking the class or teaching it, please let one of the board members know.

As you know, we exchange newsletters with different users groups across the country. Last month The Spirit of 99 ran John Tuesday's program Ohio Lottery. Congratulations, John.

PRESIDENTS CORNER

Are you tired of working in the garden,yard,or raking leaves? Are you bored with the senseless drivel on T.V.? Are you getting the most out of your computer? Would you like to meet people who would like to help you with your computing questions? Would you like to run your cassette recorder effectively,program in Extended Basic or Forth .You may even be interested in new software,suppliers. or how a Modem works, or how to work files.

We have the answer for your needs right here at our meetings on the 3rd Thursday of every month. If you noticed I said "our meetings" because you joined a users group because you were looking for somthing and we filled that need .

The officers and several of the members are here to help you and we have made a commitment to be the users group that you were looking for and more !

I would like to say how pleased I am that Ian Mariano and Dan Fedak have agreed to start teaching a beginning Forth class this month. These two fine young gentlemen are quite an asset to this group and I believe we are lucky to have them. Even if their abilities make me feel a little inadequate

The latest addition to the library is from INFORMATION ASSOCIATES. We received from them four disks "Copy-Cat" a disk copy utility,that is very quick and EASY to use (I love it). "Graphics Grabber" a very user friendly screen dump. And Two copies of "SPRITEmaker" one in 16K and one in 32K they do just what the name says it's just as easy as drawing on paper, but much much better.

As everyone knows by now I can't find enough good things to say about this fine company.They will also be helping Santa supply my daughters with Logo II. (I hope the girls can run it,I'm lost)

NORM SORKIN

This article comes to us from HUG, The Houston Users Group, 11/84.

MUSIC PROGRAM

The following is an Extended Basic Music Program of the song GREEN, GREEN by Bill Knecht. It has nice graphics and sprites.

```
100 REM GREEN/GREEN BY BILL KNECHT * TI EXTENDED BASIC
110 CALL CLEAR :: CALL SCREEN(4):: CALL CHAR(42,RPT$("F",16)):: CALL CHAR(48.RPT
$("F",16)):: CALL CHAR(90,RPT$("F",16))
120 FOR I=1 TO 7 :: READ CH,CH$ :: CALL CHAR(CH,CH$):: NEXT I
130 DATA 43,"0707070707070707",44,"0080C0E0F0F8FCFE",45,"7F3F1F0F070301"
140 DATA 49,"0707070707070707",50,"0080C0E0F0F8FCFE",51,"7F3F1F0F070301",56,"A7B
34FD90143CB7E"
150 CALL CHAR(128,"000000000000000001030707030100000003C3C2020202020A0E0E0E0C0800 000")
160 CALL COLOR(2,4,16,3,3,16,4,4,16):: FOR I=5 TO 8 :: CALL COLOR(I,16,1):: NEXT I
170 DISPLAY AT(7,1):RPT$("8",28):RPT$("Z",28):"Z***ZZ***ZZ***ZZ***ZZ***ZZZZ":"Z*
ZZZZ*Z*ZZ*ZZZZ*ZZZZ*Z*ZZZZ":"Z*+*ZZ***ZZ***ZZ***ZZ*Z*ZZZZ"
180 DISPLAY AT(12,1):"Z*Z*000*,000ZZ000ZZ000Z*000Z":"Z***0Z*-*0Z0**0Z***0Z*Z*0Z0
Z":"ZZZZ010ZZ000ZZ000ZZ000ZZ0Z0Z"
190 DISPLAY AT(15,1):"ZZZZ0Z0ZZ002ZZ0ZZZZ0ZZZZ0Z0Z":"ZZZZ000ZZ030ZZ000ZZ000ZZ0Z0
Z":RPT$("Z",28):RPT$("8",28)
200 CALL VCHAR(7,2,56,12):: CALL VCHAR(7,31,56,12):: DISPLAY AT(20,4):"PROGRAM B
Y BILL KNECHT"
210 LLA=110 :: LLB=123 :: LC=131 :: LD=147 :: LE=165 :: LF=175 :: LG=196 :: LA=2 20 :: LB=247
220 C=262 :: D=294 :: E=330 :: F=349 :: G=392 :: A=440 :: B=494 :: HC=523 :: HD=
587 :: HE=659 :: W=40000
230 CALL MAGNIFY(4):: FOR I=1 TO 7 :: CALL SPRITE(#I,128,17-I*1.2,I*6+90,RND*180
+I*4,RND*-5-1,0):: NEXT I
240 GOTO 260
250 CALL DELSPRITE :: GOSUB :: ACCEPT AT :: IF :: THEN :: ELSE :: AN$ :: STOP
```

Continued next page

MUSIC PROGRAM (Cont.)

```
260 !@P-
270 V=3 :: X=W :: Y=W :: Z=LG :: GOSUB 650 :: Z=LLA :: GOSUB 650 :: Z=117 :: GOS
UB 650 :: Z=LLB :: GOSUB 650 :: Z=LC :: GOSUB 650
280 Z=LD :: GOSUB 650 :: Z=156 :: GOSUB 650 :: Z=LE :: GOSUB 640
290 X=E :: Y=C :: GOSUB 650 :: X=LG :: Y=W :: GOSUB 650 :: X=D :: Y=LB :: Z=LG :
: GOSUB 650 :: X=E :: Y=C :: GOSUB 650 :: GOSUB 640
300 X=W :: Y=W :: Z=C :: GOSUB 650 :: Z=LG :: GOSUB 650 :: Z=LLA :: GOSUB 650 ::
 Z=117 :: GOSUB 650
310 Z=LLB :: GOSUB 650 :: Z=LC :: GOSUB 650 :: Z=LD :: GOSUB 650 :: Z=156 :: GOS
UB 650 :: Z=LE :: GOSUB 640 :: X=E :: Y=C :: GOSUB 650
320 X=LG :: Y=W :: GOSUB 650 :: X=D :: Y=LB :: Z=LG :: GOSUB 650 :: X=E :: Y=C :: GOSUB 650
330 GOSUB 640 :: X=W :: Y=W :: Z=C :: GOSUB 650 :: Z=LG :: GOSUB 650 :: Z=LLA ::
 GOSUB 650 :: Z=117 :: GOSUB 650 :: V=4 :: Z=LLB
340 GOSUB 650 :: Z=LC :: GOSUB 650 :: Z=LD :: GOSUB 650 :: Z=156 :: GOSUB 650 ::
 V=2 :: GOSUB 420 :: GOSUB 550 :: GOSUB 530 :: GOSUB 590
350 GOSUB 420 :: GOSUB 550 :: GOSUB 530 :: GOSUB 590 :: GOSUB 420 :: GOSUB 420 :
: V=4 :: GOSUB 500 :: V=6 :: GOSUB 500 :: V=8 :: X=G :: Y=W
360 GOSUB 650 :: Z=LF :: GOSUB 650 :: GOSUB 650 :: X=A :: Y=E :: Z=LC :: GOSUB 6
40 :: X=HC :: GOSUB 640 :: X=HD :: Y=F :: Z=LC :: GOSUB 650
370 V=10 :: GOSUB 640
380 X=HC :: Y=G :: Z=LC :: GOSUB 650 :: GOSUB 640 :: CALL DELSPRITE(ALL):: CALL
SOUND(-400,LC,30)
390 DISPLAY AT(24,6):"PLAY AGAIN  Y N" :: ACCEPT AT(24,22)VALIDATE("YNyn")SIZE(1):AN$
400 IF (AN$="N")+(AN$="n")THEN CALL CLEAR :: STOP :: ELSE 410
410 DISPLAY AT(24,1):"" :: GOTO 230
420 X=W :: Y=W :: Z=LE :: GOSUB 640 :: X=HE :: Y=G :: GOSUB 640 :: X=HC :: Z=LC
:: GOSUB 650 :: Z=LG :: GOSUB 650 :: X=G :: Y=W
430 GOSUB 650 :: X=A :: Y=C :: Z=LF :: GOSUB 650 :: X=HC :: GOSUB 640 :: Z=LC ::
 GOSUB 640 :: X=G :: Y=W :: Z=LF :: GOSUB 650 :: GOSUB 650
440 X=A :: Y=E :: Z=LC :: GOSUB 640
450 X=HC :: GOSUB 640 :: Y=G :: Z=LE :: GOSUB 650 :: GOSUB 650 :: Z=LG :: GOSUB
650 :: X=HD :: Y=B :: GOSUB 650 :: GOSUB 650 :: GOSUB 650
460 X=W :: Y=W :: Z=LLA :: GOSUB 650 :: Z=117 :: GOSUB 650 :: Z=LLB :: GOSUB 650
 :: Z=LC :: GOSUB 650 :: Z=LD :: GOSUB 650
470 Z=156 :: GOSUB 650 :: Z=LE :: GOSUB 640
480 X=HE :: Y=HC :: GOSUB 640 :: X=HC :: Y=466 :: Z=LC :: GOSUB 640 :: Z=LG :: G
OSUB 650 :: X=G :: Y=W :: GOSUB 650 :: X=A :: Y=C :: Z=LF
490 GOSUB 650 :: X=HC :: GOSUB 650 :: GOSUB 650 :: Y=F :: GOSUB 650 :: Z=LC :: GOSUB 650
500 X=G :: Y=W :: GOSUB 650 :: Z=LF :: GOSUB 650 :: GOSUB 650 :: X=A :: Y=E :: Z
=LC :: GOSUB 640 :: X=HC :: GOSUB 640 :: X=HD :: Y=F
510 Z=LG :: GOSUB 650 :: GOSUB 640 :: X=HC :: Y=G :: Z=LC :: GOSUB 650 :: GOSUB
640 :: X=W :: Y=W :: Z=LE :: GOSUB 650
520 Z=LG :: GOSUB 650 :: Z=LD :: GOSUB 650 :: Z=LE :: GOSUB 650 :: RETURN
530 X=D :: Y=LB :: Z=LG :: GOSUB 650 :: Y=LF :: GOSUB 640 :: X=E :: Y=C :: Z=LG
:: GOSUB 650 :: Z=LC :: GOSUB 640 :: X=W :: Y=W :: Z=LE
540 GOSUB 650 :: Z=LG :: GOSUB 650 :: Z=LD :: GOSUB 650 :: X=HC :: Z=LE :: GOSUB  650
550 X=HC :: Z=LE :: GOSUB 650 :: GOSUB 650 :: Y=G :: Z=LC :: GOSUB 640 :: Y=E ::
 GOSUB 640 :: X=B :: Y=G :: Z=LE :: GOSUB 650 :: Y=E
560 GOSUB 650 :: Y=G :: GOSUB 650 :: Y=E :: GOSUB 650 :: X=A :: Y=F :: Z=LF :: G
OSUB 650 :: Y=C :: GOSUB 650 :: Y=F :: GOSUB 650
570 X=G :: Y=C :: Z=LC :: GOSUB 650 :: GOSUB 640 :: Y=E :: GOSUB 650 :: GOSUB 65
0 :: X=C :: Y=LA :: Z=LF :: GOSUB 640
580 Y=W :: GOSUB 650 :: Y=LA :: GOSUB 650 :: RETURN
590 X=C :: Y=LA :: Z=LF :: GOSUB 650 :: X=D :: Y=LF :: Z=LG :: GOSUB 650 :: X=C
:: GOSUB 650 :: X=D :: Y=LA :: GOSUB 650 :: X=C :: Y=LE
600 Z=LC :: GOSUB 650 :: GOSUB 650 :: Y=W :: Z=LG :: GOSUB 650 :: X=W :: Z=LLA :
: GOSUB 650 :: Z=117 :: GOSUB 650
610 Z=LLB :: GOSUB 650 :: Y=C :: Z=LC :: GOSUB 650
620 Y=D :: Z=LD :: GOSUB 650 :: Y=C :: Z=156 :: GOSUB 650 :: RETURN
630 !@P+
640 FOR I=1 TO 4 :: CALL SOUND(-400,X,V,Y,V+3,Z,V+6):: NEXT I :: RETURN
650 FOR I=1 TO 2 :: CALL SOUND(-400,X,V,Y,V+3,Z,V+6):: NEXT I :: RETURN
```

This article comes to us from Spirit of 99, October 1984 issue.

HOW TO FIX DISKS

By Niraj N. Shah
Counseled by Mike Ballman

Did you ever try to catalog a disk and find out the Disk Controller thinks the disk is NOT Initialized? But you know better! What do you usually do with the blown disk? Most people Delete the file giving them the problem. Usually that does correct the problem, but it also gets rid of that file forever. The ultimate solution is to use DISK FIXER by Navarone Industries

The DISK FIXER enables one to examine and change the contents of any disk on a sector-by- sector basis. I think it is worth its forty-dollar list price. It is available from some TI retailers INFOWARE, MICROSTUPH & ZETTLERS OR directly from Navarone Industries.

Here is the process to fix up a blown disk...
First acquire a DISK FIXER from a friend or buy one, they're worth it. Get a hard-copy catalog of the blown disk, or even better, get a complete(old) catalog of what should be on the disk. If a complete catalog is not available, try to remember what should be on the disk and write those names down on paper. Once you have a catalog of the disk, you are ready to start using DISK FIXER

Insert the DISK FIXER cartridge and select option 2 from the TITle Screen. Upon doing so you should see the DISK FIXER menu. Do the following if the most recent catalog of the disk tells you there are more sectors used/free than is logically posssible: 358 for single-sided & 718 for double-sided disks
For example, IF the catalog lists 5ØØ sectors used/free on a single-sided disk THEN do the following ELSE GOTO the paragraph on "SECTOR ONE".

This part tells you how to fix up Sector Ø; which is the sector containing the information concerning the name of the disk and the number of sectors used/free on the disk. If the disk catalog tells you the used/free sector information is erroneous then Sector Ø needs to be fixed. The easiest way to this is to copy a good Sector Ø from another disk to the blown disk. Here is how to do that:

1) Insert a good disk in drive
2) Read Sector Ø of that disk:
   R Ø,1 <enter>
3) Put the blown disk in drive
4) Write good Sector Ø to disk:
   W Ø,1 <enter>

If you catalog the bad disk, you will see that the diskname and the used/free information is the same as the good disk. But do not let that alarm you. We did that to fool the Disk Controller into thinking the bad disk is at least partially restored to normalcy. Now we need to fix up the blown disk as much as we can This is done by changing Sector 1.

Here is how to fix Sector one. First, get the most complete catalog and the most recent catalog of the bad disk in front of you. Then compare the two catalogs to see which filenames are missing. Next, compile an alphabetical list of all the filenames which are and should be in the catalog.

Then you need to find the corresponding sector for each filename. This is done by using the Find String function of the DISK FIXER

1) Put the bad disk in drive
2) Find a filename by:

   F Ø,2DØ,1 <enter>
   type in the filename <enter>
3) Ignore the "ERROR N SECTOR" message
4) Write down the sector number for that filename
5) If that filename could not be found make sure you typed it ni correctly and and try again; otherwise that file does not exist on the disk.
6) Repeat the process from step two for all the filenames

CONTINUED

FIX CONTINUED

You should now have an alphabetical list consisting of two columns: filenames and sectors. With that information in hand you are ready to begin fixing up the bad disk. This is done by modifying Sector one of the blown disk. First you have to read Sector 1 from the bad disk by doing this:

1) Put the bad disk in drive
2) Read Sector 1 of disk by:
   R 1,1 <enter>

Then you want to Alter the contents of sector one. This is done by using the Alter function of the DISK FIXER. This process is best learned by observing a concrete example Thus, lets say the blown disk has 14 files (filenames) on it. Thus there should be 14 entries on sector 1; one entry for each file. The rest of the sector should be all zeroes. Lets Alter Sector 1:

1) Keep the bad disk in drive
2) Enter the Alter function:
   A Ø <enter>
3) Type in the following just as shown, including the spaces:
   1 2 3 4 5 6 7 8 9
   A B C D E
4) Do not press enter) yet!
5) If you saw a non-zero entry after the "E" entry in the first column then type in a (Ø) zero <space> and repeat until the first column shows a zero.
6) Press <enter>
7) Write the revised Sector 1 to the disk:
   W 1,1 <enter>

You have just entered a table of pointers to the files on the disk. The table points to the corresponding sector for each file name. This is the table that is updated and sorted if you add/delete files to the disk.

Leave the DISK FIXER by typing <Q> for QUIT and press (enter>. Then catalog the disk. Lets call this new catalog the "mixed" catalog. You will see the reason once the disk has been cataloged. Notice how the catalog is NOT in alphabetical order! It does however contain all the filenames that you expected to be on the disk! The next step is to alphabetize the catalog. This is done by first alphabetizing the catalog on paper and carrying along the appropriate sector number of each filename. Here is an example of a "Mixed" Catalog.

MIXED CATALOG

FILENAME	SECTOR
CAT	1
SCREEN	5
VOTE	2
FIRE	6
APPLE	E
HELLO	9
SCROLL	C
LOAD	3
TIME	8
DEMO	7
QUICK	4
JUSTIFY	D
PLOT	B
LOGO	A

SORTED CATALOG

FILENAME	SECTOR
APPLE	E
CAT	1
DEMO	7
FIRE	6
HELLO	9
JUSTIFY	D
LOAD	3
LOGO	A
PLOT	B
QUICK	4
SCREEN	5
SCROLL	C
TIME	8
VOTE	2

The above example shows how you should alphabetize the filenames and the corresponding sector numbers on paper. Once you have done this, you are ready to enter this information into Sector 1. You do not have to enter the filenames, just the sector numbers. Here is how to do that:

1) Put the blown disk in drive
2) Read Sector 1 by:
   R 1,1 <enter>
3) Enter the Alter function:
   A Ø <enter>
4) Type in the sector numbers in the ororder as shown for the above sorted example catalog. Seperate each number by a space:

FIX CONTINUED

```
   E 1 7 6 9 D 3 A B
   4 5 C 8 2
5) Then press <enter>
6) Write revised sec-
   tor to disk:
        W 1,1 <enter>
7) Put a Write-Protect
   tab on the disk!
```

You have now fixed up the disk. For verification, Quit the DISK FIXER program and catalog the disk. You should encounter no problems during the cataloging process. But you are not completely done yet! DO NOT add/delete any files or programs to the disk!
Get a fresh disk and inititalize it to the same configuration as the blown disk. Then backup the blown disk to the fresh disk. Then catalog the fresh disk and you will see that the used/free sector information is now correct. Thus, the fresh disk is now your working disk and the blown disk is now a disk for your archives.

Keep the blown disk in a safe place just in case you remember a file that was not previously recovered from the blown disk. Go through the above procedures to recover that new-but-old file.

If you have any questions on how to fix up blown disks please write to this newsletter in care of the Blown Disks department

Happy fixing!

END OF LINE

This article comes to us from the Atlanta 99/4A Computer Users Group, Sept/Oct 1984 issue.

MAGAZINES

Has another source of written material for the 99/4A dried up and blown away?

Just as we were getting all the books one could hope for, it appears that the magazines are deserting us. Compute who for the last year or so had good articles and good programs for the 99/4A has this month seemed to drop us. Only one article from Regena and no programs.

Home Computer Magazine (formerly 99'er) has just released a newsbulletin that they are no longer going to have advertising in their magazine, just articles and programs. I don't know of any publication that can do that - no magazine, no newspaper, circulation is just never enough. But, HCM says that they have studdied the idea, and they can do it. In the future a seperate flyer containing the advertising will be distributed nine times a year. Well that's better then they've done so far this year on the magazine. To date I've only counted three issues of the "new expanded" HCM, and it's already September. I'd be very wary about renewing. Sorry to lose my usual optimism about this magazine but under current conditions it's hard to be optimistic.

There are two Newsletters that are still out that I can recommend. The first is from Millers Graphics. It's called the **SMART PROGRAMMER** and if you want to program its tips and hints are absolutely essential. Cost $12.50 per year. Order from;

MILLERS GRAPHICS
1475 W. Cypress Ave.
San Dimas, CA 91773

The other is **MICROPENDIUM** it covers news, reviews, and profiles of people and issues relavent to the 99/4A. An execellent example of what a newsletter should be. Cost $12.00 per year. Order from;

MICROpendium
P.O. Box 1343
Round Rock, Tx 78680

Although neither of these two are magazines. They need our support if they are to continue to succeed and supply us with reliable information, reviews, and programing tips.

Marshall

This article comes to us from 99'ERS UGA NL #9 -7/84.

TIGERCUB TIPS #13

Copyright 1984; Tigercub Software; 156 Collingwood Ave.; Columbus, Ohio 43213. May be reprinted by non- profit users' groups, with credit to Tigercub Software.

These Tips are distributed to Users' Groups in exchange for their newsletters - and in the faint hope that someday, somewhere, someone may buy some of my original programs. I have over 130 of them, at only $3 each - some of the users'groups charge their own members almost that much for public domain programs! My catalog costs a dollar, refundable on your first order, or refundable anyway if you ask. I give one-day service by 1st Class mail, I give bonus programs for repeat orders, I give free programs on disk orders, and I'm still not getting any orders!

I'm told that someone actually found a practical use for my number-scrambling routine, so here is an expanded version. It will scramble any sequence beginning with 1 and ending with any number less than 256 or any number greater than 256 which is evenly divisible by any number less than 256 and greater than 1, within the limits of computer memory. In Extended Basic with Memory Expansion, the limit is about 10,700; if you reformat it to Basic and run it bare bones, you might get close to 13,000.

```
100 CALL CLEAR :: OPEN #1:"P
IO",OUTPUT
110 INPUT "HIGHEST NUMBER? "
:HN :: IF HN<256 THEN TN=HN
:: XX=1 :: GOTO 150
120 FOR TN=255 TO 2 STEP -1
:: IF HN/TN=INT(HN/TN)THEN
40
130 NEXT TN :: PRINT HN;"IS
NOT DIVISIBLE BY":"ANYTHING
LESS THAN 256  - ":"CANNOT U
SE" :: GOTO 110
140 XX=HN/TN
150 DIM M$(50)
```

```
160 CALL CLEAR :: FOR J=1 TO
 TN :: M$(1)=M$(1)&CHR$(J)::
 NEXT J :: FOR J=1 TO XX ::
M$(J)=M$(1):: NEXT J :: FOR
J=1 TO HN :: TT=1+INT((J-1)/
255)
170 RANDOMIZE :: X=INT(XX*RN
D+1):: IF LEN(M$(X))=0 THEN
170
180 Y=INT(LEN(M$(X))*RND+1)
190 PRINT #1:ASC(SEG$(M$(X),
Y,1))+TN*(X-1);
200 M$(X)=SEG$(M$(X),1,Y-1)&
SEG$(M$(X),Y+1,LEN(M$(X)))::
NEXT J
```

Here's a little routine you can use to jazz up your title screen or text.

```
100 CALL CLEAR
110 DATA "THIS IS A DEMONSTR
ATION","OF THE","TIGERCUB SO
FTWARE","TWO-WAY PRINT ROUTI
NE"
112 FOR T=1 TO 4
113 READ M$
120 IF LEN(M$)/2=INT(LEN(M$)
/2)THEN 135
130 M$=M$[" "
131 GOTO 140
```

```
135 M$=M$&" "
140 L=LEN(M$)
150 C=16-L/2
160 FOR J=L/2 TO 1 STEP -1
170 CALL HCHAR(10+T*2,C+J,AS
C(SEG$(M$,J,1)))
180 CALL HCHAR(10+T*2,16+L/2
-J,ASC(SEG$(M$,L-J,1)))
190 NEXT J
200 NEXT T
```

Did you ever go through your checkbook 5 times in order to add up your gas bill, then your electric bill, etc.? With this little handy- dandy, you can do it all in one pass.

```
100 CALL CLEAR
110 REM -  ADDER-UPPER by Ji
m Peterson
120 A$="ABCDEFGHIJKLMNOPQRST
UVWXYZ"
130 DIM C$(26),T(26)
140 PRINT "        ADDER-UPP
ER": : :
```

```
150 PRINT "WITH THIS PROGRAM
 YOU CAN GO THROUGH YOUR CHE
CKBOOK, OR ANYTHING ELSE, AN
D ADD UP   AMOUNTS IN SEVERA
L CATE-"
160 PRINT "GORIES ALL AT ONE
 TIME.": :
170 PRINT " FIRST, LIST THE
```

```
CATEGORIES":"YOU WANT TO ADD
 UP.":" TYPE 'END' WHEN FINI
SHED.": :
180 PRINT " NEXT, ENTER THE
CATEGORY":"CODE AND AMOUNT F
OR EACH":"BILL."
190 PRINT : :"WHEN YOU HAVE
ENTERED ALL":"THE BILLS, TYP
E =": :
200 N=N+1
210 PRINT "CATEGORY #";N
220 INPUT "             ":C$(N
)
230 IF C$(N)="END" THEN 340
240 W$=SEG$(C$(N),1,1)
250 IF POS(A$,W$,1)<>0 THEN
290
260 PRINT :"CODE LETTER ";W$
;" ALREADY USED - PICK A CO
DE LETTER."
270 INPUT W$
280 GOTO 250
290 X=POS(A$,W$,1)
300 A$=SEG$(A$,1,X-1)[SEG$(A
$,X+1,LEN(A$))
310 X$=X$[W$
320 PRINT :"CODE LETTER FOR
";C$(N);" WILL BE ";W$: :
330 GOTO 200
340 C$(N)=""
350 N=N-1
360 X$=X$&"="
370 IF FLAG=1 THEN 420
380 FLAG=1
390 PRINT : :"READY TO START
- ": : :
```

```
400 PRINT "WHEN FINISHED, TY
PE =": :
410 INPUT "DO YOU WANT TO VE
RIFY EACH  INPUT? ":V$
420 PRINT :"CODE (";X$;")"
430 INPUT Q$
440 IF Q$="=" THEN 600
450 IF POS(X$,Q$,1)<>0 THEN
510
460 PRINT "THAT IS NOT ONE O
F THE CODES
470 INPUT "IS IT A NEW CATEG
ORY?(Y/N) ":Q$
480 IF SEG$(Q$,1,1)<>"Y" THE
N 420
490 X$=SEG$(X$,1,LEN(X$)-1)
500 GOTO 200
510 Y=POS(X$,Q$,1)
520 INPUT "AMOUNT ?":A
530 ID SEG$(V$,1,1)="N" THEN
 580
540 PRINT :C$(Y);A: :
550 INPUT "CORRECT? (Y/N)":L
$
560 IF SEG$(L$,1,1)="Y" THEN
 580
570 IF SEG$(L$,1,1)="N" THEN
420 ELSE 550
580 T(Y)=T(Y)+A
590 GOTO 420
600 FOR J=1 TO N
610 PRINT :C$(J);T(J)
620 TT=TT+T(J)
630 NEXT J
640 PRINT :"GRAND TOTAL OF A
LL IS";TT
650 END
```

And, did you ever wish that you could make numbers smaller, so that you could squeeze more of them onto a chart or graph? The problem is that resolution is so poor, at least on my TV screen, but maybe you'll find a use for this.

```
100 REM - NUMBER SCRUNCHER -
 programmed by Jim Peterson
110 CALL SCREEN(5)
120 FOR S=2 TO 14
130 CALL COLOR(S,15,1)
140 NEXT S
150 CALL CLEAR
160 RANDOMIZE
170 DATA 75557,22222,25127,6
1216,55571,74616,74757,71222
,75257,75711
180 FOR J=0 TO 9
190 READ C$
200 CH$(J)="00"[C$
210 NEXT J
220 CH=91
230 INPUT "NUMBER? ":RX
240 N$=STR$(RX)
```

```
250 IF LEN(N$)/2=INT(LEN(N$)
/2) THEN 270
260 N$="0"&N$
270 FOR J=1 TO LEN(N$)STEP 2
280 P1=VAL(SEG$(N$,J,1)
290 P2=VAL(SEG$(N$,J+1,1))
300 FOR T=1 TO 7
310 Z$=Z$&SEG$(CH$(P1),T,1)&
SEG$(CH$(P2),T,1)
320 NEXT T
330 CALL CHAR(CH,Z$)
340 Z$=""
350 P$=P$&CHR$(CH)
360 CH=CH+1
370 NEXT J
380 PRINT N$;" ";P$
390 P$=""
400 N$=""
410 GOTO 230
```

Almost OUT OF MEMORY.
Happy hackin'
Jim Peterson

These last two articles come from 99'ERS UGA NL #9 - 7/84.

FORTH Arrays (concluded)
by J.W.Vincent

This, I promise, is my last article on arrays. Not that there aren't other things regarding arrays that we could explore (like automatic range checking). It's just that there are so many facets to FORTH, and I don't want to bore you. So, next month we'll deal with a practical application of disk I/O. More on that later...

This month we will take one last look at arrays. In case you hadn't noticed, TI FORTH is a little tight on dictionary space. The arrays we have previously defined can eat up that dictionary space real fast. Yet, in all but bit-map mode, large amounts of VDP RAM are unused. So, let's use it.

The screen below defines a set of array words that (application wise) operate identically to those defined last month. However, all of the variable space they allot is in VDP RAM. While they will operate slower, the dictionary space saved can be worth the comprimise in speed. As with the previous array words, they assume the first element to be row 1, column 1 (OPTION BASE 1 in BASIC) and using a zero is asking for trouble. Since the dictionary isn't automatically keeping track of memory allocation for us, I have defined a new variable called V_ADDR. Its function is to keep track of the next available VDP address for use in defining arrays. If your application is using some of the free VDP memory (like for I/O buffers) you must update V_ADDR so it doesn't allocate your memory area to arrays.

Since the definitions are shown in screen format, I have also illustrated use of the Conditional LOAD word to prevent this screen from loading if it has previously been loaded or if bit-map is loaded. Rather than discuss each word in detail, I encourage you to work through the definitions yourself, refering to our previous array words if necessary.

```
( VDP RAM arrays - JWVincent 6/27/84)
0 CLOAD VW! 0 CLOAD LINE ( if this or bit-map loaded  don't )
BASE->R    HEX     68 USER V_ADDR    1400 V_ADDR !    R->BASE
  ( character array words )
: VC_ARRAY <BUILDS DUP V_ADDR @ DUP , SWAP OVER VSBW
          ROT ROT * + 1+ V_ADDR !         ( r c VC_ARRAY name )
          DOES> @ ROT 1 - OVER VSBR * + + ;
: VC@ VSBR ;                              ( r c name VC@ ___ b )
: VC! VSBW ;                              ( b r c name VC! ___ )
  ( word array words )
: VW_ARRAY <BUILDS DUP V_ADDR @ DUP , SWAP OVER VSBW
          ROT ROT 2 * * + 1+ V_ADDR !   ( r c VW_ARRAY name )
          DOES> @ ROT 1 - OVER VSBR * + + ;
: VW@ DUP VSBR SWPB SWAP 1+ VSBR + ;     ( r c name VW@ ___ n )
: VW! OVER SWPB OVER VSBW 1+ VSBW ;      ( n r c name VW! ___ )
```

Next month I'll provide you with some screens which will both illustrate disk I/O and give you the useful capability of transfering screens to or from variable 80 format files. This should make exchanging FORTH screens much easier and facilitate transfering them via modem. Till next time ... Jim.

I GET QUESTIONS??? and hope to give answers!

by G-S Romano

This will be a occasional column of some of the more interesting questions that I receive from people all over. I may at times include a question for which I have been unable to find an answer in hopes that some kind soul out there may be able to help. In such cases, please send the answer directly to me at 116 Carl Street, San Francisco CA 94117. I will then include that reply in an upcoming column. Of course, if you have any suggestions for alternative answers to those listed here, please do send them on.

I have heard that computer systems that draw in air over the components and then exhaust it with a cooling fan subject the components to an unnecessary amount of dust. Since the 99/4A uses this method for cooling, what can I do to filter out any dirt that might enter the back of the PEB (Peripheral Expansion Box)?

Without realizing it TI has already been generous in supplying you with some washable reusable PEB filters at no extra cost. Any of the cards you bought for the Expansion system came packed in some grey sponge rubber cushioning material. The flat pieces make perfect filters. But first, go to the supermarket and buy a can of "STATIC GUARD", an antistatic spray. Spray it lightly all over the sponge sheet and let dry about 5 minutes. Then hold the sheet against the back of the PEB. With a razor blade, cut it to fit and cover the whole back section (but not the cooling fan exhaust port). Then using the razor blade, make slits in the sheet so the protruding sections of all cards can fit through them. Friction will hold the "filter" tight against the PEB so that is all there is to it! If you think of it, once every six months, take the filter off, wash it in sudsy water, dry it and respray. My "first filter" is still fine after three years.

In some forms of Basic one is allowed to use a MODULO function of numbers. This exists in TI-UCSD Pascal but not in Basic. Is there any way I can implement this in Extended Basic?

For MOD(XX) functions you can create an small algorithm with a DEF statement, but the DEF statement can handle only one modulus without some really complicated programming. This way you could easily create the function for a singular purpose in any given program. You may want to toy with the HIDDEN modulo functions that exist in Basic/Ex.Basic that TI never told us about. For example, if A=3.22532 then in a statement like CALL HCHAR(3,A,32), A will be returned as "3" since the VCHAR and HCHAR macros have built in "INT" functions for conversion of numbers before execution. It's just like saying A=INT(A) except that in these macros rounding off is also built in -- unlike the INT function. The third position (usually the character number) in these macros is a MOD 256 function. It is perfectly legitimate to use something like CALL HCHAR(3,2,18767) (That third position number cannot be greater that 32767, however). The macro just keeps doing a loop subtracting 256 from the number until it gets one that is within range. One consequence of this is that although Ex.Basic will not allow a CALL CHAR(154,"") it WILL allow a CALL HCHAR(1,1,154,32).

Another MOD 256 function is CHR$(). This can be useful in several ways for space saving in files (if the number of items is less than the 256 maximum). Since a digit takes up three bytes vs. a character's one, you can see the advantage. For example, instead of saving a record number and a string as in:

```
FOR I=1 TO 200
PRINT #1: I,A$(I)
NEXT I
```

a more efficient use of space could be utilized by:

```
FOR I=1 TO 200
PRINT #1:CHR$(I)&A$(I)
 NEXT I
```

Then when reading in the data it is simple to translate by reading in A$ and stating that your counter is ASC(A$) and A$=SEG$(A$,2,LEN(A$)-1). It is also an easy way to imbed hidden codes in records. A program line might state CHR$(18342) and be very confusing to the observer.

Yet another set of MOD functions exists in DISPLAY AT and ACCEPT AT. The first position in either of these is MOD 24 and the second is MOD 28. Stating DISPLAY AT(18645,5766) is totally acceptable because of this. So long as either value is within range (32767) no error is detected and the number is just looped until brought into the MOD limits. One space saving use of this in programming is where you want to do this:

```
DISPLAY AT(1,1):"MY FIRST GAME PROGRAM"
DISPLAY AT(24,1):"PRESS ANY KEY TO CONTINUE"
```

It is much simpler to state:

```
DISPLAY AT(24,1):"PRESS ANY KEY TO CONTINUE":"MY FIRST GAME PROGRAM"
```

Because the numbers are MOD 24 when the row display number is greater that 24 it just starts at "1" again so you achieve exactly the same thing.

Now just to show I am human at the onset, I need some help in helping someone else. We all know the lengths that TI went to to hide the architecture of the system. This is what is creating this particular dilemma. Does anyone out there know of a way to alter the shape of the cursor character (ASC-30) through Extended Basic? Poking new values into the Pattern Descriptor Table has no effect. If you know the answer please let me know.

Until then, with a scratch to the head, I await your questions.

LIST OF BOARD MEMBERS AND THEIR PHONE NUMBERS

President, Norm Sorkin	678-2360
Vice-President, Bruce Remmy	923-9315
Secretary, Open	
Treasurer, Betty Duncan	633-5217
Library, Bert Haase	753-7846
Educational Director, Rich Williams	626-2423
Editor, Kathi Anderson	923-7530
Program, John Tuesday	644-2616
Member at Large, Bruce Rodenkirch	928-7873

CHRISTMAS PARTY

We will have a Christmas party at the December 20th meeting. We will furnish cups, napkins, punch and coffee. We are asking for volunteers to bring two dozen cookies to the meeting that night. If you would like to bring some cookies you can reach Betty at 633-5217 or myself at 923-7530. We would like to see all the members come and share the holiday spirit with us.

HARDWARE & SOFTWARE

Hardware & Software is published for the computer hobbist and user. It will have advertisements, reviews, articles, news and other items of interest. This is a local publication that specializes in computer hardware and software. A free copy is sent to anyone who advertises. If you have an ad, article, cartoon, or just something to say let us know. Ad prices are 1.50 for each 25 words. This magazine is for Apple, Commordore and TI. For information about this magazine contact:

Grayg Ralphsnyder, P.O. Box 9119, Akron, OH 44305.

I would like to thank all the users groups for their articles. If you have something to contribute to the newsletter, I will include it in the newsletter. The deadline for the December issue is Saturday, December 8th. See you at the meeting.

Kathi Anderson, Editor

SUMMIT 99'er USERS GROUP
Kathi Anderson, Editor
P.O. Box 3201
Cuyahoga Falls, OH 44223

AKRON OHIO
NOV 13 '84
U.S. POSTAGE
17

USA 20c

EDMONTON 99'ERS USERS SOCIETY
P.O. BOX 11983, EDMONTON
ALBERTA CANADA.