BITS,BYTES&PIXELS

LIMA 99/4A USERS GROUP
VOLUME 2 NO 3 MARCH 1986

COMMENTARY

Every now and again we will buy piece of software that either doesn't fulfil the need for which we buy it. Or,perhaps,is is difficult to use,the documentation is poor or all of the above.

This type of problem is what this news letter was designed to solve. Your User Group should also be a place where you meet persons who can help you and save you from making a mistake in buying software.

If you are planning to buy some special type of software ask before buying. Maybe we have some thing in the library or someone else has had experience with the particular software that you are interested in.

If you have a special piece of program material you'd like to describe send it to us and we'll publish it.

ERRATA

We will not be able to attend the next meeting as we will be out of the country for a while. However,we will be back towards the end of the month. Let us know before the 11th of April if you have any copy for the Bits,Bytes and Pixels issue of April.

TIPS

Apparantly we missed sending you your "Tips from the Tiger Cub"#30 with the January issue. We have included this issue with the corrent one,#32,and it is attached. Both are provided by Jim Peterson,the tireless TI Tiger Cub. If you need any of his material the address is on the issure.

BRAIN TEASER

Arrange the numerals 1 thru 9 so that when added they wil equal 100. Hint:some have to be double digit. Answer on page 2

LIBRARY INFORMATION

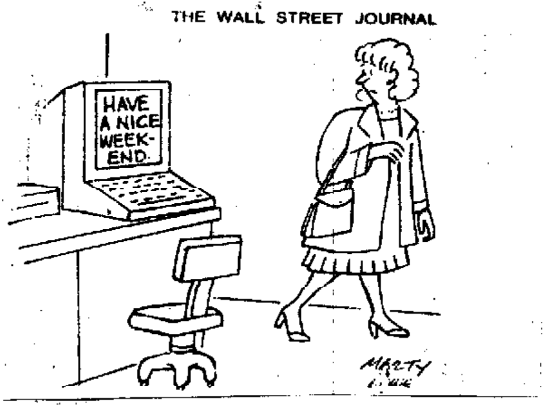
Dave Szipple is the librarian of our group. If you want a piece of software between meetings or want to return one get iH :ouch with Dave. resides at 4 Poulston Place 'phone 228 7109.

MISCELLANY

The hostess at a children's party was chatting with one of the girls,"My,your little brother certainly shy. He hasn't moved away from that corner all afternoon."

"He's not shy," answered the little girl."He never has been forced to wear a necktie before and he thinks he's tied to something."

THE WALL STREET JOURNAL



SOFTWARE REVIEW-TI 1-2-3

Marketed by Datex and sold for $39.95+S&H this seemed to be the answer to the need for a disk operated spread sheet. Indeed, we wrote in January that it seemed to have a lot of promise. That, unfortunatly, was before we tried to use it.

In the interest of fairness it should be said that this is the perception of only one person. Maybe others will see it differently.

We found that the spreadsheet was formatted similarly to Microsoft Multiplan(copyright), however, manipulation is difficult and the minimum print cycle is ten columns. Thus if you are preparing a two column five item sheet it will print your numbers and a humungous number of zeros, confusing at the least.
The word processor does not measure up to T-I Write in any of its attributes.
Finally the third part is a so called memo writer that we could not find useful for any purpose.
Editor---

ANSWER TO BRAIN TEASER

15+36+47=98+2= 100 See! it was easy
wasn't it?

BITS,BYTES & PIXELS

Published by Lima

99/4A User Group

Hal Sehnert Editor
Charles Good Tech. Editor

Material contained herein may be copied by any user group as long as credit is given to source.
Address:2225 High Ridge
Lima,Oh 45805

Published Monthly

CHARLIE SAYS:

DISPLAY vs. INTERNAL FILES

DISPLAY FILES--A couple of years ago I wrote a program to index my personal library of books and magazine articles by title, author, date, and keywords. Each library entry was assigned a maximum of 10 keywords, and I could search for all entries listed under a particular keyword as well as by title, author, and date. After alot of trial and error, and alot of referring to the USERS REFERENCE GUIDE, I found a file structure that would save my data on a disk: DISPLAY, VARIABLE 254, SEQUENTIAL, APPEND (or INPUT)

I used DISPLAY format because the USERS REFERENCE GUIDE suggested that data to be read by people (on the screen or on a printer) should use this format. The following program lines add data for a single book or magazine article onto the end of an open disk file opened for APPEND from my program:

NEXT PAGE PLEASE

```
500 PRINT #1:T$!Store title
510 PRINT #1:A$!Store author
520 PRINT #1:D$!Store date and/or
name of magazine.
530 FOR I=1 TO 10!Start routine to
store 10 keywords
540 PRINT #1:K$(N)!Store keyword to
disk
550 NEXT I!End routine to store 10
keywords.
```

These lines require 13 disk records to store the data for one library entry. You can't store multiple data items in a single record in DISPLAY files. "PRINT #1:T$,A$,D$" is not possible.

The program that I wrote using the above file structure and PRINT #1 statements could store about 500 books or magazine articles in a single massive file on the same single sided disk side as the program before filling the disk. A complete scan through this entire file searching for a particular entry took 20 MINUTES. This was a VERY LONG TIME, considering that I had 6 data disks, each of which might require a 20 minute search.

INTERNAL FILES--Recently a very able programmer, John Clulow, suggested that my search time could be speeded up if I used INTERNAL format in my file structure. To check this out I rewrote the program with the file structure INTERNAL,VARIABLE 254, SEQUENTIAL, APPEND (or INPUT).

The following program lines add data for a single book or article to the end of an open disk file:

```
500 PRINT #1:A$,D$!Store author and
date of publication
510 PRINT #1:T$!Store title. Some
of my titles are qutie long.
520 PRINT #1:K$(1),K$(2),K$(3),
K$(4),K$(5)!Store the first 5
keywords.
530 PRINT #1:K$(6),K$(7),K$(8),
K$(9),K$(10)!Store last 5 keywords.
```

NEXT COLUMN PLEASE

This time ONLY 4 RECORDS ARE NEEDED to store the data from one book or article since INTERNAL files can store more than one piece of data in each record. More importantly, SEARCH TIME HAS BEEN CUT IN HALF TO ONLY 10 MINUTES per disk side. This is due in part to the fact that the program only has to access 3 disk records for each article or book rather than 13 records. However, experimentation has shown me that a large part of the decrease in search time is due to the computer's ability to handle INTERNAL files more efficiently than DISPLAY files. Even if I still had to look through 13 disk file records for each book or article, search time would still be greatly reduced.

There is one disadvantage of INTERNAL text files, however (notice that I said TEXT files). They take up more space on the disk. That is apparently one reason why TI-WRITER uses DISPLAY format. It saves disk space in the storage of its text files.

The USERS REFERENCE GUIDE is confusing on this point. On page II-120 it says:

> "You will find that INTERNAL format is more efficient for recording data on a storage device such as a cassette tape. It requires less space."

This is sometimes true of files containing numerical data. Each number, no matter how many significant figures, requires 9 bytes of storage space. This is often less than that used to store numbers in DISPLAY files. However, string data in INTERNAL format requires an extra byte at the beginning of each string to tell the computer how long the string is. This extra byte is not used in DISPLAY files. To store exactly the same 500 books and articles using the rewritten faster searching INTERNAL file version of my program TAKES 12% MORE DISK

NEXT PAGE PLEASE

SECTORS when compared to the old DISPLAY file version of my program.

TAKE HOME LESSON-- I consider the increased disk storage space in INTERNAL format a minor problem. My trial and error with DISPLAY vs INTERNAL files suggests to me that I should probably ALWAYS USE INTERNAL FILES. The difference between a 20 minute and a 10 minute search is really significant. So why do you suppose that TI made DISPLAY the automatic default option in file processing????

SCRUNCHED DIGIT DISPLAY

Vic Shattner of our Lima User Group has developed the following series of CALL CHARs which will squeeze two numerals (0-9) in the 8x8 pixel grid normally occupied by only one numeral or letter. Althugh quite small, these mineature numerals are readable on a TV display as well as a color monitor.

With scruched digits, an extra dot is sometimes useful in keeping track of columns and the spacing between digits. A scrunched digit may be displated in the center of the 8x8 grid with or without a dot immediately above by using the following CHAR codes. All digit CHAR codes start with the same six codes 000800. Make the first 8 a 0 to turn off the dot.

Codes for centered small digits:

Digit 1-	000800180808081C
Digit 2-	0008001C0408101C
Digit 3-	0008001C041C041C
Digit 4-	00080014141C0404
Digit 5-	0008001C101C141C
Digit 6-	000800040818141C
Digit 7-	0008001C04081010
Digit 8-	0008001C141C141C
Digit 9-	0008001C141C0404
Digit 0-	0008001C1414141C

NEXT COLUMN PLEASE

For example,
CALL CHAR(62,"0000001C0408101C")
redefines character 62 (the >) as

```
  ___
 |   |
 | 2 |
 |___|
```

To put two digits in an 8x8 grid use the two of the following codes and alternate the numbers of the two codes. As above, all CALL CHARs start with the same six digits 000800. To delete the dot, change the first 8 to a 0.

Codes for digits on right and/or left sides of 8x8 pixel grid:

Digit 1-	62227
Digit 2-	71247
Digit 3-	71317
Digit 4-	55711
Digit 5-	74717
Digit 6-	12657
Digit 7-	71244
Digit 8-	75757
Digit 9-	75711
Digit 0-	75557

These are the codes for 2 and 6 alternating with each other

For example: {__________}
CALL CHAR(62,"0008007112264577)
redefines the > as

```
  ___
 | . |
 |2 6|
 |___|
```

and CALL CHAR(62,"0000007112264577)
redefines the > as

```
  ___
 |   |
 |2 6|
 |___|
```

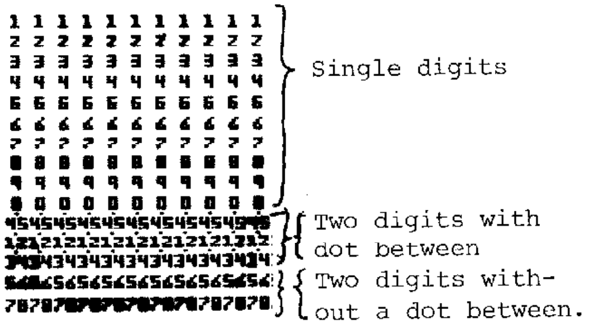
To place a single digit on the left side of the 8x8 grid, alternate the immediately above digit codes with zeros.
CALL CHAR(62,"0000000505070101")
displays:

```
  ___
 |   |
 |  4|
 |___|
```

NEXT PAGE PLEASE

```
CALL CHAR(62,"0008007040701070")
displays:   ___
           | . |
           |5  |
           |___|
```

Immediately below are some actual examples of scrunched digits so you can see what they look like. They were printed from a screen display via a screen dump program.


Single digits
Two digits with dot between
Two digits without a dot between.

VOODO CASTLE Words

The following are the first and second words recognized by Scott Adams adventure #4, Voodo Castle. As with most other Scott Adams games, the computer actually responds to only the first three letters of each word in a two word command. These letters are shown below in upper case. I am not as familiar with this adventure as I am with some of the others. Thus, there are more than the usual number of question marks indicating that I am not sure what the word is beyond the first three letters.

FIRST WORD (usually a verb)	SECOND WORD (usually a noun)
AUT?	33
BREak	34
CIRcle	35
CLEan	36
CLImb	37
CLOse	38
CRY	AMO?
CUT	ANImal
DANce	ANY
DIA?	ARMory
DIG	armor
DOO?	AROund
DRInk	BAG
DROp	BALl
DUSt	ballroom
EAT	BAR
ENTer	BOArds
EXAmine	BOOk
GET	BREw
GO	BUTton
HAMmer	CELl
HEAt	CHArm
HELp	chamber
HUG	chant
INVentory	chapel
KICk	CHEmical
LEAve	CHImney
LISten	CHUte
MIX	CLEar
MOVe	CLOver
ON	COFfin
OPEn	CRAck
PICk	CRIsto
PREss	CRYstal
PULl	DOLl
PUSh	DOOr
PUT	DOWn
QUIt	EASt
REAd	FIReplace
REMove	FLO?
RES?	FLUe
RUB	FOOt
RUN	GAMe
SAVe	GLAss
SAW	GRAves
SAY	grating
SCRape	graveyard
SHA?	HAMmer
SHOve	HEAds
SHR?	HOLe
SLIde	IDOl
SMAsh	INVentory
SMOke	KETtle
STArt	KNIfe
stay	LAB
SUMmon	LEAflet
TAKe	LEDge
THRow	LOCk
TURn	MAN
WALk	MEDium
WAVe	MOA?
YELl	MUM?

NEXT COLUMN PLEASE

CONTINUED NEXT MONTH

TIPS FROM THE TIGERCUB

#3Ø

TIGERCUB SOFTWARE
156 Collingwood Ave.
Columbus, OH 43213

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

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

I goofed again! if you tried the Quickloader in Tips #29 with a disk containing more than 2Ø programs, you may have already noticed that line 14Ø should go to 16Ø, not 155.

Here's another Tigercub Challenge - can you run this and get these results?

```
>LIST
1ØØ PRINT PI
11Ø PRINT MAX
12Ø PRINT PI
13Ø PRINT MAX
>RUN
 Ø
 Ø
 3.141592654

* SYNTAX ERROR IN 13Ø
```

Some of you sharp-eyed newsletter editors may have noticed that this text is being hyphenated to avoid some of those gaping blanks that occur when only a few long words will fit on a right-justified line. The only way that I have found to accomplish this is to set the TI-Writer right tab for the actual column width to be printed and then, whenever a word is hyphenated, backspace and replace the blanks on that line with carets, adding enough extra carets to justify the line - like this -

```
whenever^a^word^^is^^hyphen-
```

It helps to go into fixed mode with CTRL Ø when you are inserting extra carets.

When using this method, it is also necessary to set the paragraph indentation with IN Ø on the command line; if indentations are desired, they can be filled with caret signs, like this:

```
^^When using this method,
```

I am told that my old 3D Sprite Routine made it to the Golden Quickies section of CompuServe, so here is an updated version. I have found that sprites can be controlled much more easily (although not moved as rapidly) with CALL LOCATE, rather than turning them loose with CALL MOTION and then trying to catch up with them!

```
1ØØ CALL CLEAR :: CALL SCREE
N(5):: FOR SET=2 TO 8 :: CAL
L COLOR(SET,8,5):: NEXT SET
:: DISPLAY AT(3,12):"3-D SPR
ITE DEMO"
11Ø DISPLAY AT(22,1):"BY TIG
ERCUB" :: CALL CHAR(4Ø,"FF81
8181B18181FF818181818181 81FF
FFØ1Ø1Ø1Ø1Ø1Ø1FFØ1Ø1Ø1Ø1Ø1Ø1
Ø1FF")
12Ø CALL CHAR(36,RPT$("F",64
)):: CALL MAGNIFY(4):: FOR X
=2 TO 22 STEP 2 :: CALL SPRI
TE(#X,36,X/2+1-(X>7)-(X>13),
32+X*6,4Ø+X*6):: NEXT X
13Ø S=1 :: CALL SPRITE(#S,4Ø
,16,46,7):: FOR C=6 TO 42 ST
EP 2 :: CALL LOCATE(#S,46,C)
:: NEXT C :: FC=44 :: FR=46
:: Y=Ø
14Ø FOR C=FC TO FC+44 STEP 2
 :: CALL LOCATE(#S,FR,C):: N
EXT C :: FC=FC+44 :: CALL SP
RITE(#S+2,4Ø,16,FR,FC):: CAL
L DELSPRITE(#S):: TC=FC-32
15Ø FOR C=FC TO TC STEP -2 :
: CALL LOCATE(#S+2,FR,C):: N
EXT C :: TR=FR+34 :: FOR R=F
R TO TR STEP 2 :: CALL LOCAT
E(#S+2,R,TC):: NEXT R
16Ø CALL SPRITE(#S,4Ø,16,TR,
TC):: CALL DELSPRITE(#S+2)::
 FR=TR :: TR=FR-72 :: FOR R=
FR TO TR STEP -2 :: CALL LOC
ATE(#S,R,TC):: NEXT R
17Ø CALL SPRITE(#S+2,4Ø,16,T
R,TC):: CALL DELSPRITE(#S)::
 FR=TR :: TR=FR+5Ø :: FOR R=
FR TO TR STEP 2 :: CALL LOCA
TE(#S+2,R,TC):: NEXT R
18Ø Y=Y+1 :: IF Y=11 THEN CA
LL DELSPRITE(#S+2):: GOTO 13
Ø ELSE S=S+2 :: FC=TC :: FR=
TR :: GOTO 14Ø
```

Ian Swales in Belgium can write some of the most intricate routines, and pull them into the tightest knot. I had searched everywhere for a sorting routine for 2-dimensional arrays, and invented some ridiculous ones, before Ian sent me this jewel.

```
1ØØ !DEMO of two-dimensional
 sorting routine
11Ø !Set up array to be sort
ed
12Ø CALL CLEAR :: DIM A$(2Ø,
4):: RANDOMIZE :: DEF X$=CHR
$(26*RND+65)
13Ø FOR J=1 TO 2Ø :: A$(J,1)
=X$&X$&X$ :: A$(J,2)=STR$(IN
T(1ØØ*RND+1):: A$(J,3)=X$&ST
R$(INT(1Ø*RND)):: A$(J,4)=IN
T(1Ø*RND))&X$ :: NEXT J
14Ø INPUT "SORT BY?(1-4)":K
15Ø J=2Ø !2-dimensional arra
y sorting routine by Ian Swa
les
```

```
16Ø DIM Q(2Ø):: FOR X=1 TO 2Ø :: Q(X)=X :: NEXT X
17Ø M=Ø
18Ø FOR X=1 TO J-1 :: IF A$(Q(X),K)<=A$(Q(X+1),K)THEN 21Ø
19Ø M=-1
2ØØ T=Q(X):: Q(X)=Q(X+1):: Q(X+1)=T
21Ø NEXT X
22Ø IF M THEN 17Ø
23Ø FOR X=1 TO 2Ø :: FOR L=1 TO 4 :: PRINT A$(Q(X),L);" ";:: NEXT L :: PRINT :: NEXT X :: GOTO 14Ø
```

Did you ever need a routine that would accept either a string or a numeric value? Try this -

```
1ØØ N=Ø :: ON ERROR 11Ø :: ACCEPT M$ :: N=VAL(M$):: GOTO 12Ø
11Ø ON ERROR STOP :: RETURN 12Ø
12Ø ON (N=Ø)+2 GOTO 13Ø,14Ø
13Ø PRINT M$ :: GOTO 1ØØ
14Ø PRINT N :: GOTO 1ØØ
```

A useful tip from Stephen Shaw in England - if you have a long program which wil run only in Basic, and which will load from disk with CALL FILES(1) but runs out of memory when you try to run it; and if you have the MiniMemory module -
Insert MiniMemory module, select Basic, enter CALL FILES(1), Enter NEW, enter OLD DSK1.(filename). When loaded, enter SAVE EXPMEM2. When SAVEd, enter CALL LOAD(-31888,63,255), enter NEW, enter OLD EXPMEM2, and enter RUN. That is still a lot faster than loading a long program from tape!

Another reason for never using the default mode of so-called UPDATE when opening a file (without specifyying INPUT or OUTPUT) is that you will get an I/O ERROR #1 if the file is write-protected.

Has anyone found a way to go from Extended Basic to Basic without losing the program in memory, or at least fouling it up?
CALL LOAD(-32116,4) has been published in many newsletters as a way to do this, but has anyone actually made it work?

If you are printing out of TI-Writer Editor, finish your letter with CTRL U, SHIFT L, CTRL U and when it is printed the paper will automatically feed to the top of the next sheet.

To make a note to yourself while programming, just type 1! and whatever you want to make note of, then LIST "PIO":1, and then type 1 and enter to delete the line.

TI-Writer puts an extra space after every period that is followed by a space. If you don't want this extra space after abbreviations such as "Mr." or St.", use a caret sign ^ instead of a space after the period, Mr.^Jones. But TI-Writer puts only one space after ? or ! so if you want two, put a caret after the symbol !^

One of the very best tips for this month comes from Paul A. Meadows, in the September 85 newsletter of T.I.N.S. (Nova Scotia, Canada) -
How to print up to 132 characters in a line (condensed print, of course) out of TI-Writer! Just prepare your file as usual but in line ØØØ1 put formatter commands such as .LM 1Ø;RM 132; IN +5;FI;AD . The Fill and Adjust are necessary, the Indent is up to you, as are the left and right margins - but notice that right margin set way over at 132?
Now, instead of saving the file with SF, type PF and then C DSK1.(filename) to print to the disk. This not only strips out the control C characters, it also erases the TI-Writer tab line that was applied to the last line of the file.
So now, with your printer opened and initialized for condensed print, go into the TI-Writer formatter mode and print your file!

I have made the following changes to my working copy of the Tigercub Menuloader. This sets up my Gemini printer to skip over the perforations and print full page width in elite print with a wide left margin for ring-binder punching. Other printers may need changes in these codes.

```
62Ø DISPLAY AT(12,1)ERASE ALL:"PRINTER? PIO" :: ACCEPT AT(12,1Ø)SIZE(-18):P$ :: GOSUB 895 :: PP=3
84Ø DISPLAY AT(24,1):"PRINTER NAME? PIO" :: ACCEPT AT(24,15)SIZE(-14):PP$ :: GOSUB 895 :: PRINT #2:SEG$(D$,1,4)&" - Diskname= "&N$
895 OPEN #3:P$,VARIABLE 132 :: PRINT #3:CHR$(27);"B";CHR$(2);CHR$(27);"M";CHR$(1Ø);CHR$(27);"N";CHR$(6):: RETURN
```

I always keep a backup of everything, on the flipped side of another disk, and I often want to verify that the backup has everything that is on the master, and vice versa.

```
1ØØ DISPLAY AT(3,6)ERASE ALL:"TIGERCUB DOUBLECAT": :" To compare the contents of": :"a disk with a backup." !by Jim Peterson
11Ø DISPLAY AT(12,1):"INSERT MASTER DISK": :"PRESS ENTER"
12Ø CALL KEY(Ø,K,S):: IF S=Ø THEN 12Ø
13Ø DATA DF,DV,IF,IV,P
14Ø RESTORE :: FOR I=1 TO 5 :: READ T$(I):: NEXT I
15Ø DIM F$(127):: OPEN #1:"DSK1.",INPUT ,RELATIVE,INTERNAL :: INPUT #1:A$,J,J,K :: F$(Ø)=A$&" "&STR$(K)
16Ø X=X+1 :: INPUT #1:F$(X),I,J,K :: IF F$(X)="" THEN 17Ø :: F$(X)=F$(X)&" "&T$(ABS(I)):: GOTO 16Ø
17Ø X=X-1 :: CLOSE #1 :: DISPLAY AT(12,1)ERASE ALL:"REMOVE MASTER DISK": :"INSERT BACKUP DISK": :"PRESS ENTER"
18Ø CALL KEY(Ø,K,S):: IF S=Ø THEN 18Ø
19Ø OPEN #1:"DSK1.",INPUT ,RELATIVE,INTERNAL :: INPUT #1:A$,J,J,K :: DISPLAY AT(1,1)ERASE ALL:F$(Ø);:: DISPLAY AT(1,15):A$&" "&STR$(K);
2ØØ Y=Y+1 :: R=R+1 :: GOSUB 29Ø :: INPUT #1:A$,I,J,K :: IF A$="" THEN 26Ø :: K$=A$&" "&T$(ABS(I))
21Ø IF K$=F$(Y)THEN DISPLAY AT(R+1,1):F$(Y);:: DISPLAY AT(R+1,15):K$;:: GOTO 25Ø
22Ø IF K$<F$(Y)THEN DISPLAY AT(R+1,15):K$;:: Y=Y-1 :: GOTO 25Ø
23Ø DISPLAY AT(R+1,1):F$(Y);:: R=R+1 :: GOSUB 29Ø :: Y=Y+1
24Ø IF K$=F$(Y)THEN 21Ø ELSE IF K$<F$(Y)THEN 22Ø ELSE IF Y<X THEN 23Ø ELSE DISPLAY AT(R,15):K$;
25Ø GOTO 2ØØ
26Ø IF Y>X THEN 28Ø
27Ø R=R+1 :: GOSUB 29Ø :: FOR J=Y TO X :: DISPLAY AT(R,1):F$(J):: R=R+1 :: GOSUB 29Ø :: NEXT J
28Ø DISPLAY AT(24,1):"      PRESS ANY KEY" :: CALL KEY(Ø,K,S):: IF S=Ø THEN 28Ø ELSE CLOSE #1 :: END
29Ø IF R<23 THEN RETURN
3ØØ DISPLAY AT(24,1):"PRESS ANY KEY" :: DISPLAY AT(24,1):" " :: CALL KEY(Ø,K,S):: IF S=Ø THEN 3ØØ
31Ø CALL CLEAR :: R=1 :: RETURN
```

And that is just about

MEMORY FULL!

Jim Peterson

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 13Ø original programs in Basic and Extended Basic, available on casette or disk, only $3.ØØ each plus $1.5Ø per order for PPM. Entertainment, education, programmer's utilities.
Descriptive catalog $1.ØØ, deductable from your first order.
Tips from The Tigercub, a full disk containing the complete contents of this newsletter Nos. 1 through 14, 5Ø original programs and files, just $15 postpaid.
Tips from the Tigercub Vol. 2, another diskfull, complete contents of Nos. 15 through 24, over 6Ø files and programs, also just $15 postpaid. Or, both for $27 postpaid.
Nuts & Bolts (No. 1), a full disk of 1ØØ Extended Basic utility subprograms in merge format, ready to merge into your own programs. Plus the Tigercub Menuloader, a tutorial on using subprograms, and 5 pages of documentation with an example of the use of each subprogram. All for just $19.95 postpaid.
Nuts & Bolts No. 2, another full disk of 1Ø8 utility subprograms in merge format, all new and fully compatible with the last, and with 1Ø pages of documentation and examples. Also $19.95 postpaid, or both Nuts Bolts disks for $37 postpaid.
Tigercub Full Disk Collections, just $12 postpaid! Each of these contains either 5 or 6 of my regular $3 catalog programs, and the remaining disk space has been filled with some of the best public domain programs of the same category. I am NOT selling public domain programs - my own programs on these disks are greatly discounted from their usual price, and the public domain is a FREE bonus!

TIGERCUB'S BEST
PROGRAMMING TUTOR
PROGRAMMER'S UTILITIES
BRAIN GAMES
BRAIN TEASERS
BRAIN BUSTERS!
MANEUVERING GAMES
ACTION GAMES
REFLEX AND CONCENTRATION
TWO-PLAYER GAMES
KID'S GAMES
MORE GAMES
WORD GAMES
ELEMENTARY MATH
MIDDLE/HIGH SCHOOL MATH
VOCABULARY AND READING
MUSICAL EDUCATION
KALEIDOSCOPES AND DISPLAYS

For descriptions of these send a dollar for my catalog!

I've found a bug in the Tigercub Menuloader V.Ø5 which won't let you print a disk catalog if the disk contains the maximum 127 files. This should fix it.

```
34Ø I=I+1 :: IF I>127 THEN K
=X :: GOTO 43Ø
52Ø DISPLAY AT(X+5,12)SIZE(1
2):" Ø?" :: ACCEPT AT(X+5,15
)SIZE(3)VALIDATE(DIGIT):KD :
: IF KD<1 OR KD>NN THEN 52Ø
```

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 1ØX but the lines of printer control codes are annotated to help others make adjustments.

```
1ØØ DIM K$(24Ø):: LN=1ØØ ::
DISPLAY AT(3,4)ERASE ALL:"TI
GERCUB PROGLISTER": :" Will
convert a program":"listing
to 28-column format,"
11Ø DISPLAY AT(7,1):"exactly
 as it appears on the":"scre
en, and print it in 4":"colu
mns."
12Ø DISPLAY AT(11,1):" Progr
am must be RESequenced":"and
 LISTed to disk by":"RES (en
ter)":"LIST DSK1.(filename)
(Enter)"
13Ø DISPLAY AT(18,1):"Filena
me? DSK" :: ACCEPT AT(18,14)
BEEP:F$
14Ø OPEN #1:"DSK"&F$,DISPLAY
 ,VARIABLE 8Ø,INPUT
15Ø IF EOF(1)=1 THEN 26Ø ::
LINPUT #1:A$
16Ø IF LEN(A$)<8Ø THEN LN=LN
+1Ø :: GOTO 21Ø
17Ø LINPUT #1:B$ :: IF POS(B
$,STR$(LN),1)=1 THEN FLAG=1
:: LN=LN+1Ø :: GOTO 21Ø
18Ø A$=A$&B$ :: IF LEN(A$)<1
6Ø THEN LN=LN+1Ø :: GOTO 21Ø
19Ø LINPUT #1:B$ :: IF POS(B
$,STR$(LN),1)=1 THEN FLAG=1
:: LN=LN+1Ø :: GOTO 21Ø
2ØØ A$=A$&B$ :: LN=LN+1Ø
21Ø S=1
22Ø L$=SEG$(A$,S,28)
23Ø IF L$<>"" THEN 24Ø :: IF
 FLAG=1 THEN FLAG=Ø :: A$=B$
 :: GOTO 16Ø :: ELSE GOTO 15
Ø
24Ø X=X+1 :: K$(X)=L$ :: S=S
+28 :: IF X=24Ø THEN 25Ø ::
GOTO 22Ø
25Ø X=Ø :: CALL PRINTER(K$()
):: GOTO 22Ø
26Ø CLOSE #1 :: FOR J=X+1 TO
 24Ø :: K$(J)="" :: NEXT J :
: CALL PRINTER(K$()):: PRINT
 #2:CHR$(12):: END
27Ø SUB PRINTER(B$()):: IF F
=1 THEN 34Ø :: F=1
28Ø OPEN #2:"PIO.LF",VARIABL
E 132 :: PRINT #2:CHR$(15);C
HR$(27);"N";CHR$(6);!condens
ed print and perforation ski
p
29Ø PRINT #2:CHR$(27);"G";!
- double-struck printing, op
tional
3ØØ PRINT #2:CHR$(27);CHR$(4
2);CHR$(Ø);!download normal
characters - required if lin
es 31Ø-33Ø are used
31Ø PRINT #2:CHR$(27);CHR$(4
2);CHR$(1);CHR$(48);CHR$(Ø);
CHR$(64);CHR$(3Ø);CHR$(96);C
HR$(17);CHR$(72);CHR$(5);CHR
$(66);CHR$(61);CHR$(Ø);!slas
h the zero - optional
32Ø PRINT #2:CHR$(27);CHR$(4
2);CHR$(1);CHR$(42);CHR$(Ø);
CHR$(8);CHR$(34);CHR$(8);CHR
$(Ø);CHR$(62);CHR$(Ø);CHR$(8
);CHR$(34);CHR$(8);!broaden
the asterisk - optional
33Ø PRINT #2:CHR$(27);CHR$(3
6);CHR$(1);!activate redefin
ed characters - required if
lines 31Ø-32Ø are used
34Ø FOR C=1 TO 6Ø :: IF B$(C
)="" THEN 36Ø :: PRINT #2:TA
B(1Ø);B$(C);TAB(41);B$(C+6Ø)
;TAB(72);B$(C+12Ø);TAB(1Ø3);
B$(C+18Ø);CHR$(1Ø)
35Ø NEXT C
36Ø 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$(1Ø)!

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.

```
1ØØ RANDOMIZE :: DEF X=INT(R
ND*7):: FOR B=Ø TO 6 :: A(B)
=VAL(SEG$("247262294330349392
44Ø",(B+1)*3-2,3)):: NEXT B
 :: B,C,D=X
11Ø CALL SOUND(-9ØØ,A(B),Ø,A
(C),9,A(D),19):: D=C :: C=B
:: B=X :: GOTO 11Ø
```

Sound effects - thanks to Greg Healy in the Edmonton User Group newsletter -

```
1ØØ CALL INIT
11Ø FOR J=2ØØØ TO 23ØØ STEP
1Ø :: 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(-3196
2,8787)
```

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.

```
1ØØ DISPLAY AT(12,1)ERASE AL
L:"FILENAME? DSK" :: ACCEPT
AT(12,14)BEEP:F$ :: CALL CLE
AR
111 OPEN #1:"DSK"&F$,INPUT
112 DIM M$(48Ø)
113 X=X+1 :: LINPUT #1:M$(X)
12Ø DISPLAY AT(24,1):M$(X)
125 R=24
13Ø FOR T=X-1 TO 1 STEP -1 :
: IF R>13 THEN R=R-1 :: DISP
LAY AT(R,1):M$(T)
14Ø NEXT T :: IF EOF(1)<>1 T
HEN 113 ELSE CLOSE #1
```

```
1Ø !ONE-LINE MORTGAGE PAYMEN
T CALCULATOR BY SAM MORABITO
1ØØ CALL CLEAR :: INPUT "ENT
ER P,R,N WHERE P=AMOUNT, R=R
ATE, N=YEARS":P,R,N :: PRINT
 "$";INT((P*R/12ØØ)/(1-1/(1+
R/12ØØ)^(N*12))*1ØØ+.5)/1ØØ;
"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$ -

```
1ØØ CALL CLEAR
11Ø PRINT "  MULTIPLICATION
TABLES": :
12Ø FOR J=1 TO 9
13Ø FOR K=1 TO 9
14Ø PRINT TAB(K*3-2);STR$(J*
K);
15Ø NEXT K
16Ø PRINT : :
17Ø 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 -

```
1ØØ Z=2
11Ø Z=3-Z :: CALL JOYST(Z,X,
Y).......and back to 11Ø!
```

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 Ø to 1Ø 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 1Ø % 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 2ØØØØ anyway.

And did you know that TI really gave us only 15 volumes, not 3Ø? Listen and count them -

```
1ØØ FOR V=Ø TO 29 STEP 2
11Ø CALL SOUND(1ØØØ,5ØØ,V)
12Ø  CALL SOUND(1ØØØ,5ØØ,V+1
1)
13Ø FOR D=1 TO 5ØØ
14Ø NEXT D
15Ø NEXT V
```

And the duration values are just as inaccurate. Experimenting with a series of 8 CALL SOUNDs in a loop repeated 1ØØ times, I found that execution time was 4Ø seconds for any duration between 1 and 49, or a negative duration; 54 seconds for any duration between 5Ø and 66; 67 seconds between 67 and 83; 8Ø seconds between 84 and 99; 94 between 1ØØ-116; 1Ø6 between 117-133....!

I guess I've been neglecting those who don't have the Extended Basic module, so -

```
1ØØ CALL SCREEN(16)
11Ø CALL CLEAR
12Ø PRINT TAB(8);"GREENSLEEV
ES": : : : : : : : : : : : : :
 :"programmed by Jim Peterso
n"
13Ø DIM S(15)
14Ø FOR N=1 TO 12
15Ø READ S(N)
16Ø NEXT N
17Ø M$="4218ØØ995ABDC324E7DB
A51Ø6499182ØØ425AØØDBC35A66
A5243C7EB19942ØØA57E66BD3CA5
423C187E423CBD5A81ØØ99FFC3"
18Ø RANDOMIZE
19Ø FOR R=1 TO 12
2ØØ CALL COLOR(R+1,1,1)
21Ø CALL CHAR(32+R*8,CH$&CH$
)
22Ø FOR T=R TO 25-R
23Ø CALL HCHAR(T,R,32+R*8,34
-2*R)
24Ø NEXT T
25Ø NEXT R
26Ø CALL SCREEN(2)
27Ø FOR R=1 TO 12
28Ø CALL COLOR(R+1,R+2,1)
29Ø CH$=SEG$(M$,INT(47*RND+1
)*2-1,8)
3ØØ CALL CHAR(32+R*8,CH$&CH$
)
31Ø NEXT R
32Ø DATA 247,277,294,311,33Ø
,37Ø,392,44Ø,494,523,554,587
33Ø DATA 2,5,5,4,7,5,2,8,5,3
,9,5,1,1Ø,1,2,9,3,4,8,3,2,6,
3,3,3,1,1,5,3
34Ø 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
35Ø DATA 2,5,1,4,7,5,2,8,5,3
,9,5,1,1Ø,5,2,9,5
36Ø 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
37Ø DATA 3,4,1,1,2,2,2,4,1,4
,5,1,2,1,5,6,5,1
38Ø DATA 2,12,9,2,12,7,2,12,
3,3,12,12,1,11,9,2,9,7
39Ø DATA 4,8,6,2,6,3,3,3,3,1
,5,5,2,6,3,4,7,5,2,5,3
4ØØ DATA 3,5,5,1,4,4,2,5,5,4
,6,1,2,4,1,6,1,1
41Ø DATA 6,12,9,3,9,12,1,11,
8,2,9,7,4,8,6,2,6,3,3,3,3
42Ø 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
43Ø FOR J=1 TO 223 STEP 3
44Ø READ T,A,B
45Ø GOSUB 53Ø
46Ø FOR TT=1 TO T
47Ø CALL SOUND(-999,S(A),Ø,S
(B),7)
48Ø NEXT TT
49Ø NEXT J
491 FOR V=Ø TO 2Ø
492 CALL SOUND(-999,S(A),V,S
(B),V+7)
493 NEXT V
5ØØ CALL SCREEN(INT(14*RND+2
))
51Ø RESTORE 33Ø
52Ø GOTO 27Ø
53Ø CALL COLOR(A+1,INT(14*RN
D+2),1)
54Ø CALL COLOR(B+1,INT(14*RN
D+2),1)
55Ø RETURN
```

```
1 !from 9 T 9 UG newsl. Aug
85
1ØØ PRINT """Hello"" said TI
"
11Ø PRINT "Press ""ENTER"" t
o 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