Vol.3 No.3
MAR. 1985

THE SPRITE

a monthly newsletter of

THE 9900 USER'S GROUP, INC.

A voluntary organization for the sharing of knowledge and resources of people having interests in, or ownership of 9900 processor based Home Computers.

THE SPRITE is published monthly by **THE 9900 USER'S GROUP, INC.** for the enjoyment and furthering the knowledge of it's members in the use of 9900 processor based Home Computers. Address all correspondence to the EDITOR, THE 9900 USER'S GROUP, INC. P.O. Box K, Moorestown, N.J. 08057.

Articles from other newsletters are welcome and will be included to broaden our readerships base of knowledge and experience level. All submissions will be noted and credit given to the author. Articles from this newsletter may be reprinted for use by other user groups so long as the user group in question is an existing viable entity for the benefit of those wishing computer literacy. User groups that exist without a membership at large will not be considered user groups per-se. All articles or letters sent to the Editor for publication are subject to the unrestricted right to edit and comment.

THE SPRITE is available by subscription for $13.00 per year for 12 issues. Send all subscription payments and/or change of address to THE 9900 USER'S GROUP, INC., Subscription Service, P.O. Box K, Moorestown, N.J. 08057.

It will be the policy of THE 9900 USER'S GROUP, INC. not to pass, copy, or sell copyrighted text, cassettes, diskettes, or any other copyrighted medium thru The Group without the permission of the copyright owner.

Membership in THE 9900 USER'S GROUP, INC. does not impart to the general member any corporate authority or status to act for the corporation. Membership in THE 9900 USER'S GROUP, INC. is for one year from the month membership is acquired. Present membership rates are: $19/yr. as of January 29, 1985. Rates are subject to change without notice.

THE SPRITE is soliciting advertisers at the following rates:

FULL PAGE	$15.00	Note: All submitted
1/2 PAGE	$8.00	advertisements must
1/4 PAGE	$5.00	be printer ready.

We must receive all submissions by the 12th of each month for the next month's printing. Prepaid Orders Only!

THE GROUP OFFICERS and STAFF:

PRESIDENT - Michael J. Baker
VICE-PRESIDENT - Larry Wittenberg
SECRETARY - (vacant)
TREASURER - (vacant)
LIBRARY COMMITTEE: Diskette- Mike Harte
Errol Lansberry
John Bagocious
Cassette- Ray Osowski
Correspondence Librarian- Doug Ferguson

THE SPRITE STAFF:

Editor-in-Chief - Michael J. Baker
Research Editor - Errol Lansberry

TI*BBS BULLETIN BOARD:

The Bulletin Board is available to all callers at no charge. Common courtesies prevail. The BBS is up most days 8AM - 11PM. The phone # for the BBS is 609-435-7301.

INTRODUCTION:

Almost on schedule! The last couple of months have been very time consuming and the newsletter timetable has suffered. We are just about on track and even have a few articles for the next newsletter lined up. The up front news is about the Millers Graphics newletter, The Smart Programmer. Due to their schedule having slipped so bad they are no longer accepting subscriptions. If you have one now, when it runs out, thats it! They plan on putting out an expanded newsletter about 3-4 times a year. The 'August' newsletter should be out soon. They expect it back from the printer this (24 Mar) week. I guess he's been pretty busy with all his other ventures.

Also in from Millers Graphics is the new Advanced Diagnostics. A VERY worthwhile utility. It's a diskette fixer (HEX or ASCII modes) plus a maintenance aid in keeping your drive speeds within tolerance as well as being able to control Head-Seek times from the keyboard! Now when you get any of the DSDD controller cards you can determine what seek times to set your drives to without having to open the card up for each speed check. There is lot's more. Come to the NEXT meeting to see it in action.

The MYARC DSDD controller card is almost out! Soon you will be able to get a DSDD controller from MYARC up in South Jersey. As far as I know though they do not sell direct to the consumer yet. You have to go mail order or find a local retailer who will handle it. Ahem!!! Hint! Hint! No word so far 'officially' on the new 99/8 clone from MYARC. What we want to do is get a signature sheet passed at the next meeting for those interested in that letter from RYTE DATA we had in the last newsletter. We know many people just don't get a chance to sit and write letters so we felt a list sent to them from us (this goes for all you other user groups too) would help also. I've heard too that in order to get the newsletter mentioned you must send them about $7.00 ???

This meeting we'll have THREE demos! One on Advanced Diagnostics, one on LOGO II by Doug Ferguson and the other by Errol Lansberry our librarian. Errol will be demonstrating some of the Public Domain and Freeware software he has acquired for the group. The disk library is taking shape. We now need to get the cassette library in gear also. We have right now 5 volunteers to maintain that library. That's good. As soon as they wish they may begin. We wanted to first however get some of the disk library organized so we would have a good base to pull from. Things are looking better. Thanks so far to those who are involved.

Next month if all goes well we may have the newest graphics program to hit the streets. It's called GRAPHX. It does all of the things many of the graphics programs out do now, plus more! An example would be in it's comparison to devices that employ mechanical means. This is not to say they are not worthwhile but that technique is excellant for REPRODUCING graphic designs whereas GRAPHX is excellant for DEVELOPING graphics! Don't miss the demo presently scheduled for APRIL.

BBS: UL/DL

The BBS is hanging in there. I am in the process of writing the assembly for UL/DL section of the BBS. You will know I'm close since the first thing that will change is you will see a noticeable speed increase. I hope. We could use a little more activity and note passing etc.

TI WRITER: The LAST record revealed!

This information comes via Millers Graphics and clears up an interesting mystery for me anyway. In using TI Writer and if you work with records a lot what you discover is an annoying collection of 'trash' in the last record of TI WRITER. It's always there!

What that stuff is is the TAB data for TI Writer! Interestingly enough I have NEVER used the TAB functions while working with TI Writer so that data never changed for me. Hence, it was merely invisible junk, until I tried using my TI Writer files in many other applications and then I would get a few characters and a whole bunch of U's.

Ok. What you need to do to avoid that stuff is to use the PRINT FILE (PF) method of saving to disk rather than SAVE FILE (SF) method. Have fun!

TI-WRITER TUTORIAL Part 1 by Michael Kelly

For me the most useful software package is TI-Writer. In the following months I hope to lead you through some of the useful procedures in TI-Writer. To use the TI-Writer software you need 32K memory expansion, one disk drive (or two), RS232 card, and a printer. We will begin using the text editor to write a letter.

FIRST DRAFT OF LETTER.

1. INSERT TI-WRITER CARTRIDGE.
2. PRESS ANY KEY.
3. SELECT TI-WRITER BY PRESSING **2**.
4. INSERT TI-WRITER DISK.
5. SELECT TEXT EDITOR BY PRESSING **1**.
6. SELECT SCREEN COLOR BY PRESSING **CTRL 3**.
7. SET TABS, INDENT AND MARGINS:

-SET LEFT MARGIN (L) AT 10 BY MOVING CURSOR OVER 10 AND PRESSING **L**.

-SET INDENT (I) AT 13 BY MOVING CURSOR OVER 13 AND PRESSING **I.**

(INDENT must be set from left of page and not the margin.)

When INDENT (I) is used ENTER and NEW PARAGRAPH automatically indent the next line.)

(TAB uses INDENT as another tab.)

-SET TAB (T) AT 40 BY MOVING CURSOR OVER 40 AND PRESSING **T.**

(A TAB set at 40 is 40 spaces from the left of the page, not 40 spaces from the margin.)

(Other tabs can be eliminated by putting the cursor over the T and pressing the **SPACE BAR.**)

- SET RIGHT MARGIN (R) AT 70 BY MOVING CURSOR OVER 70 AND PRESSING **R.**

8. EXIT TAB BY PRESSING **ENTER.** YOU ARE NOW IN EDIT MODE.

9. START TEXT ON LINE 0001.

10. TYPE LETTER EXACTLY AS SHOWN BELOW USING THE FOLLOWING OPERATIONS:

-CURSOR UP	**FCTN E** OR **CTRL E**
-CURSOR DOWN	**FCTN X** OR **CTRL X**
-CURSOR LEFT	**FCTN S** OR **CTRL S**
-CURSOR RIGHT	**FCTN D** OR **CTRL D**
-TAB	**FCTN 7** OR **CTRL I**
-NEXT WINDOW	**FCTN 5**
-ROLL UP	**FCTN 6** OR **CTRL B**
-ROLL DOWN	**FCTN 4** OR **CTRL A**
-LEFT MARGIN RELESE	**CTRL Y**
-NEW PARAGRAPH	**CTRL 4** OR **CTRL J**

(See **TI-WRITER QUICK REFERENCE CARD** for explanation of operations.)

(Since you are using word wrap (solid cursor), do not hit enter at the end of each line. you will automatically be shifted to the next line. Hit enter only at the end of a paragraph.

```
0001                                  Street
0002                                  City, State Zip
0003                                  Date
0004  Dear joe,
0005    Hoe are you? hope all is well with you.
0006
0007
0008    this past wek I won $9,000,000.06 pllllllaying
0009  bingo at church.  Ihav no idea what todo with the
0010  money.  IF you can help me spend some of the money,
0011  send give me a call (111)BIG-BUCK.
0012  be ablt
0013  THANK you fror reeeading this 7urgent call for help.
0014                                  gratefully yours
0015  Name
```

11. EXIT EDIT MODE BY PRESSING **FCTN 9.**

12. SAVE FILE. (DSK1.LETTERA) -TYPE **SF** (SAVE FILE) THEN PRESS **ENTER.**

-TYPE **DSK1.LETTERA** (FILE NAME) THEN PRESS **ENTER.**

13. EXIT EDIT MODE BY PRESSING **FCTN COMMAND/ESCAPE.**

14. PRINT FILE.

-TYPE **PF** (PRINT FILE) THEN PRESS **ENTER.**

-TYPE **RS232** (REST OF PRINTER NAME IF NECESSARY) THEN PRESS **ENTER**

15. EXIT TEXT EDITOR BY PRESSING **Q** (QUIT) THEN **ENTER** AND **E** (EXIT) THEN **ENTER.**

16. EXIT TI-WRITER BY PRESSING **FCTN QUIT.**

EDIT LETTER

1. REENTER TI-WRITER. (See page 1.)
2. SELECT TEXT EDITOR BY PRESSING **1.**
3. LOAD FILE.

-TYPE **LF** THEN PRESS **ENTER.**

-TYPE **DSK1.LETTERA** (FILE NAME) THEN PRESS **ENTER.**

4. EDIT LETTER AS SHOWN BELOW USING EXPERIMENTATION AND THE FOLLOWING COMMANDS:

-FOUR WAY CURSOR MOVEMENT.

-TYPE OVER EXISTING TEXT.

-INSERT LINE **FCTN 8** OR **CTRL O**

-DELETE CHARACTER **FCTN 1** OR **CTRL F**

-DELETE LINE **FCTN 3** OR **CTRL N**

-WORD TAB **CTRL 7** OR **CTRL W**

-NEXT PARAGRAPH **CTRL 4** OR **CTRL J**

-LAST PARAGRAPH **CTRL 6** OR **CTRL H**

-OOPS **CTRL 1** OR **CTRL Z** (Will not work after reformat.)

-HOME CURSOR **CTRL L**

-INSERT CHARACTER **FCTN 2** OR **CTRL G**

(After insertion is completed, reformat will close the text.)

-REFORMAT **CTRL 2** OR **CTRL M**

(See **QUICK REFERENCE** GUIDE for explination of operations.)

(If inserting a letter at the end of a word, skip a space then reformat.)

(Reformat will adjust all lines which follow cursor location. Therefore, the use of the carriage return character is important, If carriage return characters are not used unwanted reformatting can occur.)

```
0001                                  Street
0002                                  City, State Zip
0003                                  Date
0004
0005
0006  Dear Joe,
0007
0008    How are you? Hope all is well with you.
0009
0010    This past week I won $9,000,000.06
0011  playing bingo at church.
0012
```

```
0013     I have no idea what to do with the money.
0014   If you can help me spend some of the money,
0015   give me a call at (111)BIG-BUCK.
0016
0017     Thank you for reading this urgent call for
0018   help.
0019
0020                          Gratefully yours,
0021
0022
0023
0024
0025                          Name
```

5. EXIT EDIT MODE BY PRESSING **FCTN 9.**
6. SAVE FILE. (DSK1.LETTERA)
   -TYPE **SF** (SAVE FILE) THEN PRESS **ENTER.**
   -TYPE **DSK1.LETTERA** (FILE NAME) THEN PRESS **ENTER.**
   (Using the same name (**DSK1.LETTERA**) deletes the first draft of the letter and replaces it on the disk with the edited letter.)
7. EXIT EDIT MODE BY PRESSING **FCTN 9.**
8. PRINT FILE.
   -TYPE **PF** (PRINT FILE) THEN PRESS **ENTER.**
   -TYPE **RS232** (and rest of printer name if necessary) THEN PRESS ENTER.
   (Printed letter will not show carriage return character or line numbers.)
9. EXIT TEXT EDITOR BY PRESSING **Q** (QUIT) THEN ENTER AND E (EXIT) THEN **ENTER.**
10. EXIT TI-WRITER BY PRESSING **FCTN QUIT.**

I will end here for this month. If you have saved a copy of the letter used above keep it. Next month we will use the letter to show some different commands.

MUNCHMAN

Did you know MUNCHMAN had a hidden test mode? Try this to enter the test mode but remember you must be very quick (within 3 seconds).

SHIFT 8 SHIFT 3 SHIFT 8

1. RND (0-2) appears which means WHAT ROUND
2. SCN (0-19) appears which means WHAT SCREEN
3. MM (1-9) appears which means HOW MANY MUNCHMEN

UNHEARD SUBOCTAVES

As we all know, normally 110 HZ is as low as the sound processor SHOULD go. (Heh,heh,heh). Play with the following program if your looking for some low bass notes.

```
100 FOR S=4000 TO 110 STEP -10
110 CALL SOUND(400,24000,30,24000,30,S,30,-4,0)
120 NEXT S
130 END
```

PASCAL ??

WHAT IS PASCAL?

1. A Soutnern French city
2. Foot fungus
3. A sham machine
4. None of the above.

If you answered 4 than continue on to the next sentence else stay with Basic your better off. So let's take a look and see what the mysterious P-System really is.

The P-System was developed at the University of Calf. at San Diego and is usually called the UCSD P-SYSTEM. The P stands for pseudomachine because the machine language for the system was designed around a hypothetical processor. This hypothetical processor was given an architecture which lends itself well for execution of Pascal programs. Webster's dictionary defines pseudo as a sham, so if you picked 3 above than you were very close.

I hope this will solve some of the mystery about odd names you may come across in the computer world. If not here is a little test to practice on.

*******COMPUTER QUIZ*******

SOFTWARE
1. Angora sweater
2. Foam rubber forks

DISK DRIVE
1. Campaign to collect old records.
2. Saucer shaped parking space

FLOW CHART
1. Map for rivers in the area
2. Graph that fell into the soup.

INVENTOR OF THE COMPUTER
1. Bill Cosby
2. Mr. Chips

WRITER OF COMPUTER MANUALS
1. ET
2. Marquis De Sade

MICROCHIPS

1. Eaten with micro dip
2. What a herd of micros drop on the open prairie

FLOPPY DISK

1. Record album they can't sell
2. Painful lower back problem
3. Rubber frisbee

FORTRAN

1. Between Threetran and fivetran
2. How computers get excited prior to interface

RELIABLE COMPUTER ORGANIZATION

1. IBM
2. CIA
3. PLO

Quiz courtesy of "HOCUS" Milwalkie Area.

INFOCOM HINTS:

Zork I:

1. Kill the thief with the knife.
2. Remember to stop at the Gallery for the painting.
3. Don't tinker with the skeleton in the Mase or he might be you next time around.

Zork II:

1. Here are some of the treasures you need to collect:

A.	Moby ruby	E.	Statuette
B.	Portrait	F.	Bills
C.	Rare stamp	G.	Zorkmid
D.	Gaudy crown	H.	Gold key

Zork III:

1. Some water is dirty but clean water can be had from the Shady Brook.
2. Try to REZROV the egg in the Jewel Room.

WHY USE RELATIVE FILES

Most people are generally afraid of all files. Don't ask me why, it is just true. When they finally get around to tackling files at all, it is at cassette files that they take their first crack. One reason is that cassette files don't seem as forbidding as disk files. The other obvious reason is that until recently most people only had cassettes to work with.

Cassette files are necessarily 'sequential', and relative files can only be used with disks. As disk owners became more numerous, those who learned enough about cassette files to actually program with them, changed from cassette files to disk files by just changing 'CS1' to 'DSK1.8888' and went merrily plugging along with their same sequential file set up; only the access time was faster because disks are faster.

Those same people who were afraid of cassette files but finally took the plunge are now in many cases very reluctant to start dealing with RELATIVE files. If this applies to you, Welcome to the club. I am also numbered among you.

Now that I have mentioned why many don't use relative files; lets cover reasons that they are good for you. Since this is a introduction and not a lengthy tutorial, this will be brief.

One advantage is that relative files can be used sequentially; so it is possible to keep using files the way you are used to using them. You could sneak up on relative files as it were. The key advantage to relative files is that you can use pieces of files without having to go through the whole thing. For example: You could address the fifth record of a file without having to read in the first four just to get to it.

Imagine you had a check book program where each record was set up like this: check #, amount of check, and paid to whom. You had a very long list of checks listed numerically and you wanted to look up just check numbers 200 through 250. Using a REC number in a file INPUT statement could save you a lot of search time. It would appear like this: INPUT #1,REC number:check number. IF check number < 200 THEN add 15 to the REC number. This is not meant to be proper programming. It is just to illustrate how you could do it.

Notice in the program listed below that the REC number is both printed to the file and input from the file starting with REC 1 (K=1). If I didn't specify; the computer would have started with REC 0.

If you had long lists of checks or whatever; you can see how adding some INPUT and IF/THEN statements would let you search through a disk file without having to input the whole thing into memory. That is a big advantage of Relative files. Computer memory space is saved when only part of a long file has to be loaded into the computer.

```
100 OPEN #3:"DSK2.MESS",RELATIVE,INTERNAL,UPDATE
110 READ A(1),B(1),C$(1),A(2),B(2),C$(2),A(3),B(3),C$(3),
    A(4),B(4),C$(4)
125 FOR K=1 TO 4
130 PRINT #3,REC K:A(K),B(K),C$(K)
135 NEXT K
140 FOR K=1 TO 4
150 INPUT #3,REC(K):H(K),J(K),KK$(K)
160 NEXT K
170 CLOSE #3
180 PRINT H(1);H(2);H(3);H(4)
189 DATA 201,500.00,TOM,202,300.00,DICK,203,700.00,HARRY,
    204,100.00,FRED
190 END
```

Gary Matthews

TIPS FROM THE TIGERCUB

#18

TIGERCUB SOFTWARE
156 Collingwood Ave.,
Columbus OH 43213

My new catalog #5 is now available for $1.00, which is deductable from your first order. It contains over 130 programs in Basic and Extended Basic at only $3.00 each (plus $1.50 per order for casette, packing and postage, or $3.00 for diskette, PP&H).

The entire contents of Tips from the Tigercub Nos. 1 through 14, with more added, are now available as a full disk of 50 programs, routines and files for only $15.00 postpaid.

Nuts & Bolts is a diskfull of 100 (that's right, 100!) XBasic utility subprograms in MERGE format, ready for you to merge into your own programs. Contents include 13 type fonts, 14 text display routines, 12 sorts and shuffles, 9 data saving and reading routines, 9 wipes, 8 pauses, 6 music, 2 protection, etc., etc., all for just $19.95 postpaid!

New programs this month -

TCX-1058 SCRUM, now available in Extended Basic. I'm told that this challenging puzzle-game has been programmed for other computers under the name Merlin. I haven't seen it, but I don't think you can beat my version - it's 511 puzzles in one!

TCX-1137 SOUNDMAKER, a very versatile utility program to develop sound effects, then save them in the form of actual program lines. Requires Extended Basic; disk only.

I must first thank all those newsletter editors and other users' group officers who are trying so hard to help me keep my kitchen table enterprise alive. One users group reprinted my entire catalog in their newsletter, another is putting it on their BBS, another made me an honorary life member, many others have mentioned and recommended my software in their newsletters.

Unfortunately, all that support hasn't helped very much. From reading the editorials in many newsletters, I can easily see that most users groups consist of a few dedicated hard-working individuals and a lot of.....well, frankly, freeloaders. And freeloaders don't buy software!

To borrow a few quotable quotes from the newsletters, "too many getters and not enough givers", and "users are users!". That is why users groups are fading away, software producers are going out of business, and the TI-99/4A will die before its time.

In the last Tips, I mentioned the one remaining bug in my 28-Column Converter. I have found a fix for it. The version published in Tips#15 was a horrible example of sloppy programming, so I have rewritten it entirely -

```
100 DISPLAY AT(1,4)ERASE ALL
:"28-COLUMN CONVERTER" :: DI
SPLAY AT(3,12):"by Jim Peter
son"
110 DISPLAY AT(5,1):" To con
vert a program, saved":"with
 LIST ""DSK1.FILENAME"",":"i
nto 28-column format which":
"can be merged into the text
"
120 DISPLAY AT(9,1):"buffer
of TI-Writer."
130 DISPLAY AT(11,1):" Optio
nally with transliter-":"ate
d #, &, %, ^ and . for":"pri
nting from formatter":"mode.
"
140 DISPLAY AT(16,1):" Progr
am should be RES in":"steps
of 10 starting at 100":"befo
re LISTING to disk."
150 DISPLAY AT(20,1):" Do yo
u want to print the":"file f
rom the":" (E)ditor":" (F)o
rmatter?"
160 ACCEPT AT(24,1)VALIDATE(
"EF")BEEP:Q$
170 LN=100 :: CALL CLEAR ::
INPUT "What is the FILENAME?
      DSK1.":FN$ :: FN$="DS
K1."&FN$ :: PRINT : :
180 INPUT "what is the new F
ILENAME?  DSK1.":PN$ :: PN$
="DSK1."&PN$ :: OPEN #1:FN$,
DISPLAY ,VARIABLE 80,INPUT :
: OPEN #2:PN$,DISPLAY ,VARIA
BLE 80,OUTPUT
190 IF Q$="E" THEN 200 :: PR
INT #2:".TL 126:94;" :: PRIN
T #2:".TL 123:64;" :: PRINT
#2:".TL 125:38;" :: PRINT #2
:".TL 124:42;" :: PRINT #2:"
.TL 92:46;" :: PRINT #2:".NF
"
200 IF EOF(1)=1 THEN 300 ::
LINPUT #1:A$
210 IF LEN(A$)<80 THEN LN=LN
+10 :: GOTO 260
220 LINPUT #1:B$ :: IF POS(B
$,STR$(LN),1)=1 THEN FLAG=1
:: LN=LN+10 :: GOTO 260
230 A$=A$&B$ :: IF LEN(A$)<1
60 THEN LN=LN+10 :: GOTO 260
240 LINPUT #1:B$ :: IF POS(B
$,STR$(LN),1)=1 THEN FLAG=1
:: LN=LN+10 :: GOTO 260
250 A$=A$&B$ :: LN=LN+10
260 S=1
270 L$=SEG$(A$,S,28):: IF Q$
="E" THEN 280 :: GOSUB 320
280 IF L$<>"" THEN 290 :: IF
 FLAG=1 THEN FLAG=0 :: A$=B$
 :: GOTO 210 :: ELSE GOTO 20
0
290 PRINT #2:L$ :: S=S+28 ::
 GOTO 270
300 IF Q$="E" THEN 310 :: PR
INT #2:".FI;AD;"
310 CLOSE #1 :: CLOSE #2 ::
END
320 DATA (see instructions below)
330 RESTORE 320 :: FOR W=1 T
O 5 :: READ CH$,K$
340 X=POS(L$,CH$,1):: IF X=0
 THEN 360
350 L$=SEG$(L$,1,X-1)&K$&SEG
$(L$,X+1,LEN(L$)):: GOTO 340
360 NEXT W :: RETURN
```

The DATA elements to be typed in line 320, separated by commas, are - the "at" sign above the 2, the left brace on the front of the F key, the

ampersand above the 7, the right brace on the front of the 6, the carat sign above the 6, the tilde on the front of the N, the asterisk above the 8, the whatsit? on the front of the A, the period, and the backslash on the front of the Z. If you don't want to revert to FILL and ADJUST, delete the second statement in line 300.

Beware the A6 bug! The asterisk in the above program is transliterated because of an odd quirk of TI-Writer which causes it to change A*256 into A6! It happened to me, and I've seen it in two published programs.

If my Autoloader gives you a couple of asterisks instead of the number of sectors, it's because you have files over 99 sectors long. You can change the image in line 170 to *** if you want to.

Here is probably the last word on the challenge to write a 1-line XBasic program which would scramble the numbers 1 to 255 into a random sequence without duplication. This one runs in 17 seconds!

```
100 ! FROM TIGOFT (BELGIUM)
NEWSLETTER V.6 #4 JULY-SEPT
84 - ANONYMOUS
110 DIM R(255):: FOR I=0 TO
255 :: R(I)=I :: NEXT I :: F
OR I=0 TO 255 :: RANDOMIZE :
: CALL PEEK(-31806,J):: K=R(
J):: R(J)=R(I):: R(I)=K :: N
EXT I
120 FOR J=0 TO 255 :: PRINT
R(J);:: NEXT J
```

I believe that Craig Miller is due the credit for publishing the PEEK used in that routine. He also found a PEEK to get two random numbers, which I fooled around with until I discovered I had a mosquito trapped behind my TV screen.

```
100 ! MOSQUITO  by Jim Peter
son from a PEEK by Craig Mil
ler
110 CALL CLEAR :: CALL SPRIT
E(#1,42,2,106,100)
120 RANDOMIZE :: CALL PEEK(-
31808,A,B):: CALL MOTION(#1,
A-128,B-128):: GOTO 120
```

If you're worried about the mosquito getting out, you can put a screen on the window by adding a statement to line 110 - CALL CHAR(32,"FF888888FF888888")

Here's one for the kiddies -

```
100 REM - DANCING STICKMAN p
rogrammed by Jim Peterson
110 CALL CLEAR
120 DIM S(26),T(60),NN(60)
130 FOR CH=48 TO 80 STEP 8
140 CALL CHAR(CH,"00002B107C
1028")
150 NEXT CH
160 GOSUB 590
170 FOR SET=3 TO 7
180 CALL COLOR(SET,1,1)
190 NEXT SET
200 DATA "  H  000  P","  H
 000  P","  H   0   P","  00
0000000","  8  000  @","  8
 000  @"
210 DATA " 88  000  @@","  H
HH000PPP","  H  8 @  P","  H
  8 @  P","HHH  8 @  PPP","
   8 @","     8 @","   888
@@@"
220 PRINT "     dancing stic
kman": : : :
230 RESTORE 200
240 FOR J=1 TO 14
250 READ A$
260 PRINT TAB(6);A$
270 NEXT J
280 CALL COLOR(3,16,5)
290 CALL COLOR(4,16,7)
300 CALL COLOR(5,5,16)
310 GOTO 690
320 ON INT(3*RND+1)GOSUB 340
,400,460
330 RETURN
340 CALL COLOR(4,1,1)
350 CALL COLOR(6,16,5)
360 GOSUB 560
370 CALL COLOR(6,1,1)
380 CALL COLOR(4,16,7)
390 RETURN
400 CALL COLOR(5,1,1)
410 CALL COLOR(7,16,7)
420 GOSUB 560
430 CALL COLOR(7,1,1)
440 CALL COLOR(5,7,16)
450 RETURN
460 CALL COLOR(4,1,1)
470 CALL COLOR(5,1,1)
480 CALL COLOR(6,16,5)
490 CALL COLOR(7,16,7)
500 GOSUB 560
510 CALL COLOR(6,1,1)
520 CALL COLOR(7,1,1)
530 CALL COLOR(4,16,7)
540 CALL COLOR(5,5,16)
550 RETURN
560 FOR D=1 TO 30
570 NEXT D
580 RETURN
590 F=262
600 FOR N=1 TO 25
610 S(N)=INT(F*1.059463094^N
)
620 NEXT N
630 S(26)=40000
640 RESTORE 740
650 FOR J=1 TO 60
660 READ T(J),NN(J)
670 NEXT J
680 RETURN
690 FOR J=1 TO 60
700 CALL SOUND(T(J)*100,S(NN
(J)),0,S(NN(J))+5,5)
710 GOSUB 320
720 NEXT J
730 GOTO 690
740 DATA 4,6,4,13,4,13,4,13,
4,17,4,13,4,17,4,15,4,12,4,1
3,4,13,4,15,4,17,8,13,4,12
750 DATA 4,8,4,13,4,13,4,15,
4,17,4,18,4,17,4,15,4,13,4,8
2,4,8,4,10,4,12,8,13,4,13,4,
26
760 DATA 4,10,4,12,4,10,4,9,
4,10,4,12,8,13,4,8,4,10,4,8,
4,6,4,5,4,6,8,6
770 DATA 4,10,4,12,4,10,4,9,
4,10,4,12,4,13,4,10,4,6,4,13
,4,12,4,15,8,13,4,13,4,26
```

I used to sign off with "happy hackin'", but the Vandals and thieves have made hacking a disreputable word, so

Meeowww
The Tigercub
Jim Peterson

TIPS FROM THE TIGERCUB

#19

TIGERCUB SOFTWARE
156 Collingwood Ave.
Columbus, OH 43213

The entire contents of Tips from the Tigercub Nos. 1 through 14, with more added, are now available as a full disk of 50 programs, routines and files for just $15.00 postpaid!

Nuts & Bolts is a diskfull of 100 (that's right, 100!) XBasic utility subprograms in MERGE format, ready for you to merge into your own programs. Contents include 13 type fonts, 14 text display routines, 12 sorts and shuffles, 9 data saving and reading routines, 9 wipes, 8 pauses, 6 music, 2 protection, etc., and now also a tutorial on using subprograms, all for just $19.95 postpaid!

And I have about 140 other absolutely original programs in Basic and XBasic at only $3.00 each!(plus $1.50 per order for cassette, packing and postage, or $3.00 for diskette, PPM) Some users groups charge their members that much for public domain programs! I will send you my descriptive catalog for a dollar, which you can then deduct from your first order.

Folks, I just can't afford to keep mailing out these Tips if you don't BUY something once in awhile! I am hearing from more and more groups who want to get on my mailing list, but I am having to cut back. I am dropping those groups which don't give any indication that their members ever get to see the Tips, and I'll have to cut further. If you do send me an order, or even ask for my catalog, mention your users group so I'll know there is someone still alive out there!

If you know of any schools in your area, especially elementary schools, that have TI-99/4As in the classroom, won't you please give me their address? I'll send them a free catalog.

Danny Michael has improved his graphics screen dump to include rotate and double size! It is in assembly, very fast, and runs out of XBasic, E/A module or Mini Memory. He has also written an assembly Neatlist program which lists an XBasic program to a printer in single line statements, indented, expanded, etc., very useful for debugging, setting up pre-scan, etc.

These are freeware, pay if you want and whatever you want. Just send an initialized disk for either one, or two disks (or SSDD or flippy) for both, in a returnable mailer with ENOUGH RETURN POSTAGE, to

Danny Michael,
Rt 9 Box 460
Florence, AL 35630.

John Hamilton of the Central Iowa Users Group will send you his 22-page booklet of "99 Tips" for the TI-99/4A, for just $4.00. The address is

John Hamilton,
4228 E. Clinton, Des Moines IA 50317.

I have been experimenting with TI-Writer, and this issue of the Tips is being printed in 4 columns, right justified directly from the printer. Here's how -

Use TI-Writer, editor mode, in any line length you want. The first line should be .RM 27;FI;AD but don't use any other formatter codes. Don't indent paragraphs. Use some other character as a temporary substitute for any ^, @, & or $ in the text. Don't include any program listings, yet.

Save the file as DSK1.TEXT. Print an edit copy. Then go into formatter mode. Select DSK1.TEXT to be printed, but instead of your printer spec, type DSK1.TEXT2. Your file will now be in 28-column format and right justified, and indented.

If the text is to include any program listings, run them through my 28-Column converter (see Tips #18), using the Editor option of that program.

Go back to TI-Writer editor and load DSK1.TEXT2. Merge in the program listings. Then PF to print file, but instead of a printer spec, type C DSK1.TEXT3. When it has printed to disk, LF the DSK1.TEXT3 and you will find that all control characters are gone.

Now for a bit of editing. Delete the 3 blank lines at the beginning, and the 6 blank lines that have appeared after every 60th line. Center the title by erasing with the space bar and retyping - do NOT use FCTN 2! Also replace any temporary characters with the ^, @, & or $.

You will print 4 columns of 60 lines per page, so the total lines in your file must be a multiple of 240. Add enough blank lines to the end of the file to reach that count.

Save that file back to disk as DSK1.TEXT3. Now go into XBasic, key in this program and RUN!

```
100 OPEN #1:"DSK1.TEXT3",INP
UT :: OPEN #2:"PIO",VARIABLE
 255 :: PRINT #2:CHR$(15);CH
R$(27);CHR$(69):: DIM B$(240
)
110 FOR A=1 TO 2 :: FOR B=1
TO 240 :: LINPUT #1:B$(B)::
NEXT B
120 FOR C=1 TO 60 :: PRINT #
2:TAB(10);B$(C);TAB(41);B$(C
+60);TAB(72);B$(C+120);TAB(1
03);B$(C+180):: NEXT C :: PR
INT #2:CHR$(27);CHR$(97);CHR
$(6):: NEXT A :: CLOSE #1 ::
 CLOSE #2 :: END
```

The A loop is for a 2-page printout of 480 lines, of course.

You can modify this routine to print in 2 or 3 columns, adjust the margins, change the type font or size, rewrite for your own printer, etc. And the column width can be anything you want, just change that .RM 27 in the first line of the text (don't forget that the left margin is set at 0, not 1).

If you want a 2-column page, you can dump the file back to disk instead, and then print it out of TI-Writer editor. Use this routine, modified as you wish.

```
100 !Opens a file TEXT3 of 2
40 lines 35 char long and co
nverts it into a file which
can be printed out of TI-wri
ter Editor as 2 pages in 2 c
olumns
110 OPEN #1:"DSK1.TEXT3",INP
UT :: OPEN #2:"DSK1.TEXT4",O
UTPUT :: DIM B$(120)
120 FOR A=1 TO 2 :: FOR B=1
TO 120 :: LINPUT #1:B$(B)::
NEXT B
130 FOR C=1 TO 60 :: PRINT #
2:"    "&B$(C)&RPT$(" ",38-
```

```
LEN(B$(C)))&B$(C+60):: NEXT
C :: FOR D=1 TO 6 :: PRINT #
2:" " :: NEXT D :: NEXT A ::
 CLOSE #1 :: CLOSE #2
```

It is best to run a program to set up your printer, and leave it turned on, before printing that file out of the Editor. It is not at all easy to imbed control characters in the file, because they affect the line in all coluans and also shift the lines out of alignment.

I understand that there a couple of kids who wait every month for their dad to key them in a bit of nonsense from the Tigercub, so -

```
100 !KEYZAP - by Jim Peterso
n
110 DISPLAY AT(6,11)ERASE AL
L:"KEYZAP" :: DISPLAY AT(12,
1):"  Zap the Zprite by typ
ing the key in the correspon
dingposition on the keyboard
."
120 DISPLAY AT(24,10):"Press
 any key" :: CALL KEY(0,K,S)
:: IF S=0 THEN 120
130 RANDOMIZE
140 CALL CHAR(47,"817EA58199
A5423C")
150 CALL CLEAR :: T=0 :: CAL
L FLASH(T)
160 CALL KEY(3,K,ST):: IF ST
=0 THEN 180
170 C=C+1 :: IF C=101 THEN 1
90 ELSE CALL KEYBOARD(K,T)
180 CALL MOTION(#1,25*RND-25
*RND,25*RND-25*RND):: CALL C
OINC(#1,#2,16,A):: IF A=0 TH
EN 160 ELSE CALL FLASH(T)::
GOTO 160
190 CALL DELSPRITE(ALL):: DI
SPLAY AT(12,9):"GAME OVER" :
: DISPLAY AT(14,9):"SCORE";T
 :: DISPLAY AT(16,9):"PLAY A
GAIN?"
200 CALL KEY(3,K,S):: IF S<1
 THEN 200
210 IF K=89 THEN C=0 :: GOTO
 150 ELSE END
220 SUB KEYBOARD(K,T)
230 IF FLAG=1 THEN 250 :: FL
AG=1
240 KEY$="1234567890=QWERTYU
IOP/ASDFGHJKL;"&CHR$(13)&"ZX
CVBNM,."
250 IF (K=47)+(K=61)+(K=13)T
HEN SUBEXIT ELSE X=POS(KEY$,
CHR$(K),1):: Y=ABS(X)11)-(X)
22)-(X)33)+1 :: R=Y*6 :: C=(
(X+(Y)1)*(Y-1)*11)*3)
260 CALL SPRITE(#2,42,16,R*8
-7,C*8-7):: CALL COINC(#1,#2
,16,N):: IF N=0 THEN SUBEXIT
270 CALL FLASH(T):: SUBEND
280 SUB FLASH(T):: FOR N=1 T
O 10 :: CALL SCREEN(16):: CA
LL SCREEN(8):: NEXT N :: CAL
L SPRITE(#1,47,2,1,1):: T=T+
1 :: DISPLAY AT(1,20):T :: S
UBEND
```

And here's another -

```
100 ! QUICK & DIRTY DOODLER
      by Jim Peterson
Use joystick #1. Press fire
button to change color or
pattern, Enter to clear the
screen.
110 DATA FFFFFFFFFFFFFFFF,FF
,0101010101010101,0000000000
0000FF,8080808080808080,01020
4081020408,8040201008040201,
FF818181818181FF
120 CALL CLEAR :: FOR J=1 TO
 8 :: READ CH$(J):: NEXT J
130 FOR CH=32 TO 136 STEP 8
:: FOR CN=CH TO CH+7 :: X=X+
1 :: CALL CHAR(CN,CH$(X)):: 
NEXT CN :: X=0 :: NEXT CH ::
 CALL CHAR(32,"0")
140 CALL SCREEN(16):: FOR S=
2 TO 14 :: CALL COLOR(S,S+1,
1):: NEXT S :: R=12 :: C=16
:: CH=33
150 CALL HCHAR(R,C,CH):: CAL
L FASTJOY(C,R,Q):: IF Q=18 T
HEN CH=CH+1+(CH=143)*110
160 CALL KEY(0,K,S):: IF K=1
3 THEN CALL CLEAR :: GOTO 15
0 ELSE 150
170 SUB FASTJOY(C,R,Q):: CAL
L JOYST(1,X,Y):: CALL KEY(1,
Q,S):: X=SGN(X):: Y=-SGN(Y):
: C=C+X+(C=32)-(C=1):: R=R+Y
+(R=24)-(R=1):: SUBEND
```

And a pretty one -

```
100 CALL CLEAR :: CALL SCREE
N(2):: FOR S=2 TO 8 :: CALL
COLOR(S,15,1):: NEXT S :: DI
SPLAY AT(12,7):"KALEIDOSQUAR
ES" ! by Jim Peterson
110 FOR CH=40 TO 136 STEP 8
:: FOR L=1 TO 4 :: RANDOMIZE
 :: X$=SEG$("0018243C425A667
EB199A5BDC3DBE7FF",INT(16*RN
D+1)*2-1,2)
120 B$=B$&X$ :: C$=X$&C$ ::
NEXT L :: CALL CHAR(CH,B$&C$
):: B$,C$=NUL$ :: NEXT CH
130 FOR S=2 TO 14 :: X=INT(1
5*RND+2)
140 Y=INT(15*RND+2):: IF (Y=
X)+(Y=8)THEN 140
150 CALL COLOR(S,X,Y):: NEXT
 S
160 AR,R,AVR,VR=1 :: AC,C,AH
C,HC=4 :: TT=24 :: XX,XT=13
170 FOR L=1 TO 12 :: T=TT ::
 XT=XX :: R=AR :: VR=AVR ::
C=AC :: HC=AHC
180 FOR J=1 TO XT :: X=INT(1
3*RND+2)*8+24 :: CALL HCHAR(
R,HC,X,T):: CALL HCHAR(25-R,
HC,X,T):: CALL VCHAR(VR,C,X,
T)
190 CALL VCHAR(VR,31-C,X,T):
: T=T-2 :: HC=HC+1 :: VR=VR+
1
200 NEXT J :: AR=AR+1 :: AVR
=AVR+1 :: AC=AC+1 :: AHC=AHC
+1 :: TT=TT-2 :: XX=XX-1 ::
NEXT L
210 IF INT(2*RND)<>0 THEN 23
0
220 FOR S=INT(12*RND+2)TO 14
 :: CALL COLOR(S,1,1):: NEXT
 S
230 FOR J=1 TO INT(20*RND+1)
:: S=INT(13*RND+2):: X=INT(1
5*RND+2):: Y=INT(15*RND+2)::
 CALL COLOR(S,X,Y):: NEXT J
240 CALL SCREEN(INT(15*RND+2
)):: ON INT(5*RND+1)GOTO 130
,160,220,230,240
```

The challenge in Tips #16 was - how can you store a hundred or more values of any size, positive or negative, integer or non-integer, even in exponential notation, without dimensioning an array or opening a file, and then link to another program with a RUN statement and recover those values - not by reading them from the screen? I had just one reply! Was it too easy, too hard, or doesn't anyone care? Anyway -

```
20591 SUB CHARSAVE2(CH,N)::
N$=STR$(N):: N$=RPT$("0",16-
LEN(N$))&N$
20592 IF POS(N$,".",1)=0 THE
N 20593 :: N$=SEG$(N$,1,POS(
N$,".",1)-1)&"A"&SEG$(N$,POS
(N$,".",1)+1,LEN(N$))
20593 IF POS(N$,"+",1)=0 THE
N 20594 :: N$=SEG$(N$,1,POS(
N$,"+",1)-1)&"B"&SEG$(N$,POS
(N$,"+",1)+1,LEN(N$))
20594 IF N<0 THEN N$=SEG$(N$
,1,POS(N$,"-",1)-1)&"F"&SEG$
(N$,POS(N$,"-",1)+1,LEN(N$))
20595 CALL CHAR(CH,N$):: SUB
END
```

And to recover the values -

```
20596 SUB READCHAR(CH,N):: C
ALL CHARPAT(CH,CH$)
20597 IF POS(CH$,"A",1)=0 TH
EN 20598 :: CH$=SEG$(CH$,1,P
OS(CH$,"A",1)-1)&"."&SEG$(CH
$,POS(CH$,"A",1)+1,LEN(CH$))
20598 IF POS(CH$,"B",1)=0 TH
EN 20599 :: CH$=SEG$(CH$,1,P
OS(CH$,"B",1)-1)&"+"&SEG$(CH
$,POS(CH$,"B",1)+1,LEN(CH$))
20599 IF POS(CH$,"F",1)<>0 T
HEN CH$="-"&SEG$(CH$,POS(CH$
,"F",1)+1,LEN(CH$))
20600 N=VAL(CH$):: SUBEND
```

Here's a jewel of a routine from Danny Michael, to avoid those lockups and other foul-ups that occur when you CALL INIT after you have already CALLed INIT -

```
CALL PEEK(8198,A):: IF A<>17
0 THEN CALL INIT
```

The best way to edit a program is to type NUM and the first line number, then Enter will take you through line by line with no danger of accidentally deleting a line. The edit functions will still work, and FCTN 4 gets you out of the NUM mode.

MEMORY FULL!

Jim Peterson

RAMON's NOTES:

```
       DEF  MENU1,MENU2
VSBW   EQU  >2020                  Address obtained from E/A Manual P.416
OFFSET DATA >6000                  Extended Basic Offset.  We are assigning it a label  so  that
*                                  we can refer to it by a name rather than a number
EXIT   BSS  2                      Set  up  a variable called "EXIT".  We will use this variable
*                                  to hold the X-Basic return address.
M1     TEXT 'MENU1              '  M1 has a total of 320 bytes (10 rows of 32 columns.)
       TEXT '-------------------'
       TEXT 'PRESS:             '
       TEXT '                   '
       TEXT '   1 - TO LOAD MENU 2'
       TEXT '                   '
       TEXT '  2 - TO END PROGRAM'
       TEXT '                   '
       TEXT '-------------------'
*
M2     TEXT 'MENU2              '  M2 has a total of 320 bytes (10 rows of 32 columns.)
       TEXT '*******************'
       TEXT '                   '
       TEXT 'PRESS:             '
       TEXT '                   '
       TEXT '  1 - TO LOAD MENU 1'
       TEXT '                   '
       TEXT '  2 - TO END PROGRAM'
       TEXT '                   '
       TEXT '*******************'
*
MENU1  MOV**R11,@EXIT              Save the return address in EXIT variable
       BL   @CLS                   Do a Call Clear
       LI   R0,32                  VDP Ram Screen Address for Row 2, Col 1
       LI   R2,M1                  Load R 2 with M1 address
       LI   R3,320                 Load R 3 with the number of bytes to write.  We will use this
*                                  Register as a counter.
L1     MOVB *R2+,R1                Treat  the data in R 2 as an address, move the left byte into
*                                  R1, and increase the valve in R2 by one
*
*
       AB   @OFFSET,R1             Add XB Offset
       BLWP @VSBW                  "Gosub" VSBW
       INC  R0                     INCrease R1 by one.  Remember this register holds the VDP Ram
*                                  address of the byte to be written.
       DEC  R3                     DECrement R3 by 1
       JNE  L1                     If R3<>0 then Menu1
       MOV  @EXIT,R11              Put return address in R11
       RT                          Return to XB
**********************************
MENU2  MOV  R11,@EXIT
       BL   @CLS
       LI   R0,96                  VDP Ram address of Row 4, Col 1
       LI   R2,M2                  Load R2 with address of the second menu
       LI   R3,320                 Number  of  bytes  to  write.  We will use this register as a
*                                  counter.
L2     MOVB *R2+,R1                Treat data in R2 as address,  move  left  byte  into  R1  and
*                                  increment value in R2 by 1
```

```
       AB   @OFFSET,R1
       BLWP @VSBW
       INC  R0
       DEC  R3
       JNE  L2
       MOV  @EXIT,R11
       RT
*************************************
CLS    LI   R0,767
       LI   R1,)80
CLEAR  BLWP @VSBW
       DEC  R0
       JOC  CLEAR
       RT
       END
```

The CLS routine above is the same as a CALL CLEAR in basic. The only thing that is being done is to fill the entire screen (767 spaces) with blank characters ()20).

Here is a practical use for the 8K that is reserved in EX. Basic for assembly language programs (or subprograms). What you will have to do is place in the text statements what you want then adjust the VDP Ram Screen Address (LI R0,???) and The Number of Bytes To Write (LI R3,???). You will have to practice before you get the hang of it. When you use this subprogram it can be called with a CALL LINK("MENU1") or "MENU2". Here is the subroutine to use.

```
10  CALL CLEAR :: CALL INIT
20  CALL LOAD("DSK1.CODE")
30  CALL LINK("MENU1")
40  CALL KEY(0,K,S)
50  IF S=0 OR K()49 AND K()50 THEN 40
60  IF K=50 THEN STOP ELSE CALL LINK("MENU2")
70  CALL KEY(0,K,S)
80  IF S=0 OR K()49 AND K()50 THEN 80
90  IF K=50 THEN STOP ELSE 30
100 END
```

ED NOTE: In the above short program line 100 is not needed. There is no way for the program to get to this line. Line 90 uses an ELSE condition to force a jump to line 30 if K()50. Also, to keep from 'losing' STOPs and ENDs there should only be one place in any program for it to come to a halt. For example what could have been done in line 90 is to replace the STOP command within the IF ... THEN to the line number 100. As a program gets larger and larger a programmer might just 'forget' about a STOP or END stuck in some routine within an IF ... THEN and be very surprised somewhere down the road.

```
Add XB Offset
"Gosub" VSBW
INCrement R0
DECrease the counter
If not done (R3()0) then goto L2


This is what a CALL CLEAR looks like in machine language.
VDP Ram address
Space Character ()20) Plus Offset ()60)
"Gosub" VSBW
DECrease R0
Jump On Carry to CLEAR
Return
```

MEETING AGENDA:

7:00PM - 7:15PM Introduction, words of wisdom and whit.
7:15PM - 7:45PM LOGO II demo
7:45PM - 8:00PM Public Domain & FREEWARE demo
8:00PM - 8:15PM Advanced Diagnostics Demo
8:15PM - 9:00PM Open session, walk around, look at set up systems, buy stuff, etc.

MEETING DATES:

MONTH	GENERAL MEETING	SPLINTER MEETING
MAR	27	13
APR	---) 29 (--note change	15
MAY	29	14

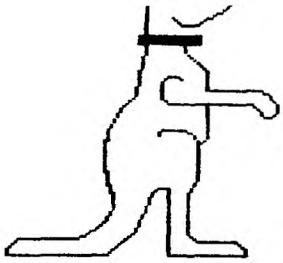
A GPL ROUTINE

BY JOHN PHILLIPS - VIDEO MAGIC

```
DEF GPL
REF VMBW
*********************************************
* The following program demonstrates        *
* the method for calling a GPL routine      *
* without using the @GPLLNK utility in      *
* the EDITOR/ASSEMBLER. The word at         *
* @XML must contain a GROM address which    *
* contains >0F, followed by >F0. The        *
* address is different for different        *
* GROM releases. In the example, the        *
* first part of the code scans through      *
* GROM memory until it finds the >0FF0.     *
* When it does, it stores that value        *
* makes the program MACHINE DEPENDENT:      *
* it will run on any version of /4As.       *
* The GPLLNK is called by a BL, not a       *
* BLWP.                                     *
*********************************************
MYWS   EQU >8300          SET MY WORKSPACE IN PAD
GRMWA  EQU >9C02          GROM WRITE ADDRESS REGISTER
GRMRD  EQU >9800          GROM READ DATA REGISTER
XML    DATA 0             1 WORD FOR XML BRANCH ADDRESS TO BE FOUND
       EVEN
*********************************************************
* START SCANNING FOR A >0FF0 IN GROM FOR XML RETURN     *
* THIS FIRST PIECE OF CODE MUST BE EXECUTED BEFORE      *
* ANY CALLS TO @GPLLNK CAN BE MADE.  YOU ONLY HAVE TO   *
* EXECUTE THIS TOP CODE, ONCE!                          *
*********************************************************
GPL    LWPI MYWS          LOAD MY WORKSPACE
       LI R3,>0300        START SCANNING ABOUT >300 GROM
*
LOOP   LIMI 0             DISABLE INTERRUPTS
       MOVB R3,@GRMWA     SET HIGH BYTE
       SWPB R3
       MOVB R3,@GRMWA     SET LO BYTE
       SWPB R3            RESTORE
*
       CLR R4             CLEAR REGISTER
       MOVB @GRMRD,R4     GET DATA FROM GROM
       CI R4,>0F00        >0F?
       JNE INCR           NO, SO TRY NEXT BYTE
       MOVB @GRMRD,R4     GOT THE >0F, SO NEED THE >F0
       CI R4,>F000        >F0?
       JNE INCR           NO, SO TRY NEXT BYTE
       JMP GOTIT          FOUND RETURN ADDRESS
*
INCR   INC R3             TRY NEXT SET OF BYTES
       JMP LOOP
*
GOTIT  MOV R3,@XML        SET XML ADDRESS
       LIMI 2             ENABLE INTERRUPTS AGAIN
```

```
*************************************
* PROGRAM TO TEST GPLLNK SUBROUTINE *
* NOW ALL YOU EXTENDED BASIC NUTS   *
* CAN HAVE A GPLLNK OF YOUR VERY    *
* OWN!                              *
*************************************
YORPGM CLR R8             ZERO OUT R8
       MOVB R8,@>837C     CLEAR STATUS BYTE
       BL @GPLLNK         CALL ROUTINE
       DATA >36           BAD  SOUND ROUTINE
*
       LI R0,>FFFF        DELAY
DLY    DEC R0
       JNE DLY
       JMP YORPGM         KEEP BUMPING!
*************************************
* THIS IS THE SUBROUTINE CODE TO    *
* REPLACE GPLLNK. REGISTERS 0 AND 1 *
* OF YOUR CALLING WORKSPACE ARE     *
* ALTERED. SO IS >8300 CPU RAM.     *
*************************************
```

```
GPLLNK LIMI 0
       MOVB @>8373,R1     FETCH GPL SUBSTACK POINTER
       SRL R1,8           MAKE IT A WORD
       AI R1,>8302        ASS PAD OFFSET
       MOV @XML,R1        PUT XML INSTRUCTIONS ADDRESS ON STACK
       SWPB R1            HIGH BYTE CONTAINS OFFSET INTO PAD
       MOVB R1,@>8373     UPDATE GPL SUBSTACK POINTER
       MOV *R11+,R0       GETCH GPL ROUTINE ADDRESS
       MOVB R0,@GRMWA     SET UP GPL PROGRAM COUNTER
       SWPB R0             SENDING HIGH BYTE FIRST
       MOVB R0,@GRMWA      FOLLOWED BY LOW BYTE
       LI R0,RTN          GET DESIRED RETURN ADDRESS
       MOV R0,@>8300      PUT RETURN ADDRESS IN XML TABLE
* NOTE: THIS ADDRESS MUST ALWAYS BE >8300!!
       LWPI >83E0         LOAD GPL WORKSPACE
       B @>006A           BRANCH TO GPL CODE
RTN    LIMI 2
       LWPI MYWS          RESTORE MY WORKSPACE
       RT                 RETURN TO MAIN PROGRAM
*
       END
```



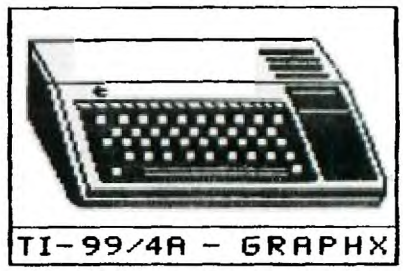
GRAPHX

(SYDNEY) : 7139506

FREE POSTER OFFER

LATE EDITION APRIL 1985

NEW GRAPHICS PROGRAM NAILABLE!



TI-99/4A - GRAPHX

EXCLUSIVE

If you have an expanded TI-99/4A congratulations, you too can become part of the ever-growing group of satisfied 'GRAPHX' owners situated all over the U.S.A. and Australia.

GRAPHX is a graphic design program written in Assembler which sets a new standard of quality and user friendliness for third party T.I. software. As an example of the power of the program everything in this newsletter, except the daisy wheel printed text (like you are now reading) was drawn with GRAPHX, this includes all headings, pictures, lines, etc.

Response from GRAPHX users has been overwhelmingly favourable, e.g.:

"The very best single program I have ever seen for the TI."
.. CO-ORDINATOR, TASMANIAN TI USERS GROUP

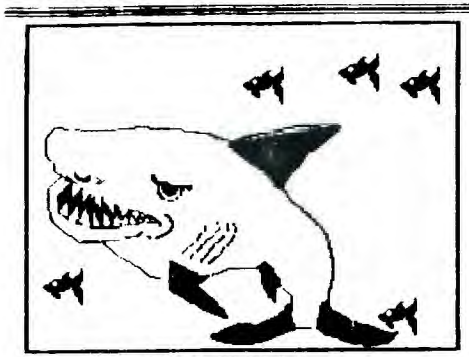
"The best program for the TI that any of us have seen".
.. MEMBER, EXECUTIVE BOARD, SOUTHERN CALIFORNIA COMPUTER GROUP

"A super program ... GRAPHX is really needed."
.. VICE PRESIDENT, CIN-DAY USERS GROUP

On the following pages we will tell you more about the powerful features of GRAPHX, how you can order your own copy and about some special offers which apply to quantity purchases.

POSTER # 1

All orders of 5 or more copies of GRAPHX will receive a FREE poster print produced using a new utility program currently under development. You may select a design from any in the newsletter which have 'Poster #' indicated, for example you may select the picture above by requesting 'Poster #1'.



POSTER PRINT # 2

G·R·A·P·H·X SETTING A STANDARD

"GRAPHX?", you might say to yourself, "strange name, what does it do?". Well, GRAPHX is a graphics processor which means that it works with images in the same way that a word processor works with words. Using GRAPHX you can create images, copy them or move them around, you may file them away then retrieve and recombine them in many different ways.

To fully describe all the features of the program is beyond the scope of a newsletter this size, it takes the 48 page Users Guide to do that, but very briefly they include free-hand drawing and erasing with full control over color and speed; a powerful zoom mode for detailed work; automatic circle creation; straight lines using advanced 'Rubber Banding' techniques and an easy to use palette allows you to select colors for any part of the screen; you can fill shapes, move parts of the screen around or copy a section many times; four different styles and size of print are available and of course you may save your picture to disk for later recall.

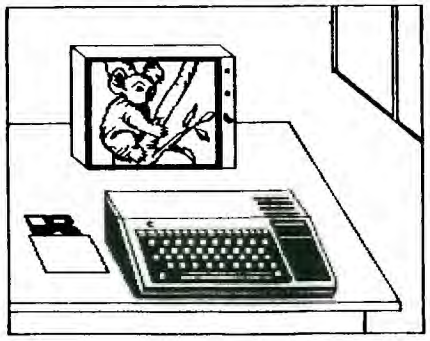
One VERY powerful feature is the CLIPBOARD'. With it you may create and save sets of objects such as special alphabets or symbols so they are available for use whenever required. The clipboard is useful for many other functions, for example it may be used as a garbage bin from which your trash may be retrieved if you wish. The clipboard will even let you move images from one screen to another or try a little computer animation! Each copy of GRAPHX comes complete with ready to use clipboards featuring a range of special alphabets and examples of animation.

All features of the program are easy to access and use, for example to create a circle you need only press a single labelled function key. You are then given a circle on the screen which you can move around, stretch, squash or change in size, using only the joystick, until you have created a circle precisely the size and shape you want, then press the joystick button to fix it in place.

Whenever you need to make a choice in GRAPHX, for example when choosing a printer style, a menu is super-imposed over part of the screen presenting you with all the available options and you select the one you want simply by using the joystick.

One of our aims when designing GRAPHX was to make the program easy to learn and use. Working on the assumption that people would rather not spend hours reading a manual, we decided to put all the necessary instructions in the program itself. When the program is running two lines of 'HELP' information are displayed prompting you for your next action. These lines operate in such away that none of the drawing screen is wasted, they are "intelligent" and automatically move away from your cursor when they sense that it is close. You can, of course, turn them off to view your picture unobstructed.

As well as being fun to use, GRAPHX can do useful work, for example it is being used to produce training video tapes for the Australian Institute of Sport, to illustrate manuals for other software packages, and if your club produces a news-letter we know you will find it invaluable for headlines, logos, etc.



POSTER PRINT # 3

GRAPHX requires the TI-99/4A computer (will not operate on the old 99/4), at least one disk drive, 32K expansion and a joystick. If you wish to print you will also require the R.232 card and an Epson MX-80 compatible printer (see next article for info on other printers). You will also need one of the following modules : Mini Memory, Editor Assembler or Extended Basic. There is a separate version for each of the modules, all have identical functions however there are minor differences in loading speeds and clipboard size, in order of preference they are:

1. Mini Memory - 6K Clipboard
   63 second load time;
2. Editor Assembler - 4K Clipboard
   65 second load time;
3. Extended Basic - 4K Clipboard
   251 seconds load time.

Built into GRAPHX are four types of screen dumps, two sizes and two densities. The Bulldog on page one is cut from a large size print and the picture above is a small size, exact original sizes are 8" x 5" and 4.0" x 2.6".

If your printer is not MX-80 compatible help will soon be at hand. Coming (mid year) is a new utility program which will greatly extend GRAPHX'S printer capabilities. It will provide all normal GRAPHX printer modes but for a much wider range of printers.

Among its functions will be "POSTER PRINTING" which will print your GRAPHX screen over six A4 sized sheets, poster printing is quite spectacular, to give an idea of the size we have reproduced (see right), a small fragment taken from a poster print of the picture at left, you will need to look at it from about 10 ft away for the best effect! The utility will also enable you to pre-print your stationery with GRAPHX designed letterhead, a feature we developed in response to requests from early users.

A handy facility will allow you to create banners, simply type a phrase and it will be automatically printed in letters 8 inches high!.

Below you will see examples of the way GRAPHX'S circle and line functions allow you to lay out the framework of a drawing very easily. You can then use the zoom mode to add the finishing touches!

T. I. LOGO

by Aleta Duey

Last month, in my first article, I gave some background information on the Logo computer language. I neglected to say that Logo was developed in the Artificial Intelligence Laboratory at Massachusetts Institute of Technology in the 1970's. Logo can be used at all age levels and at all ability levels -- from toddlers to adults -- from nursery school games to graduate school engineering, physics, and A.I. research.

From now on, whenever I speak of T. I. Logo, I shall be referring to T. I. Logo II, which greatly improved on the original T. I. Logo. This has been true of all Logo publishers who are constantly coming out with new versions with more commands and greater capability.

And now to begin! The T. I. Logo cartridge is necessary as well as extended memory (not Extended Basic), and a disk system. After the title screen appears and the user chooses T. I. Logo, you are ready to begin.

1. When the screen reads, "Welcome to T. I. Logo!", you are in NOTURTLE mode -- one of three modes. Type TELL TURTLE ... be sure to leave a space between words. This puts you into the "Turtle Mode" and places the turtle on the screen, ready to draw. The bottom six lines on the screen are reserved for your input and the computer's replies.

2. To type in a procedure, type the title preceded by the word "TO"; for example, TO TQUARE. Press ENTER and this puts you into the "Edit Mode."

3. In the "Edit Mode," press ENTER at the end of every line. Type all lines of the procedure and finish with END on a line by itself.

4. Leave the "Edit Mode" by using FUNCTION 9. You can use the arrow keys to move around while in the "Edit mode." To erase in any mode, use FUNCTION 3.

5. When back in "Turtle Mode," type the procedure title, omitting the word TO, for example, SQUARE, and your procedure will run. You can repeat this as many times as you want and even save it on a disk. To clear the screen each time, type the primitive CS.

6. Procedures need not be written immediately, and the best way to begin is just by experimenting with FORWARD, BACK, RIGHT TURN, LEFT TURN, and CLEARSCREEN. Abbreviations may be used: FD, BK, RT, LT, and CS. The first four commands (called primitives) must be followed by a space and a number, telling the turtle how many turtle steps to move or how many degrees to rotate.
   An example is: FD 20, RT 90, FD 20, RT 90, FD 20, RT 90, and FD 20, which draws a square.

7. To write a procedure so that this can be easily repeated, type TO SQUARE and press ENTER. Now you are in the "Edit Mode." Press ENTER again to start the second line. Then type the program as follows:

```
TO SQUARE
FD 20
RT 90
FD 20
RT 90
FD 90
RT 90
FD 20
END
```

Use FUNCTION 9 to return to "Turtle Mode" and type SQUARE. You will see your procedure redrawn every time you type the command SQUARE (Remember to clear the screen by typing CS). For fun, try typing HT (HIDE TURTLE) and watch how much faster the figure is drawn when the Turtle is invisible. To make it appear again, type ST (SHOW TURTLE).

"Challenges" in Logo are ideas that are suggested for the user to try. Some beginning challenges might be DRAW A TRIANGLE, DRAW A RECTANGLE or DRAW A HOUSE (not so easy!). Then try to write procedures to show them over and over. However, some people are more interested in just "playing Turtle" and, in a spontaneous way, discovering what develops. This is the fun of Logo. Remember, in Logo, A COMPUTER IS A MACHINE TO THINK WITH!

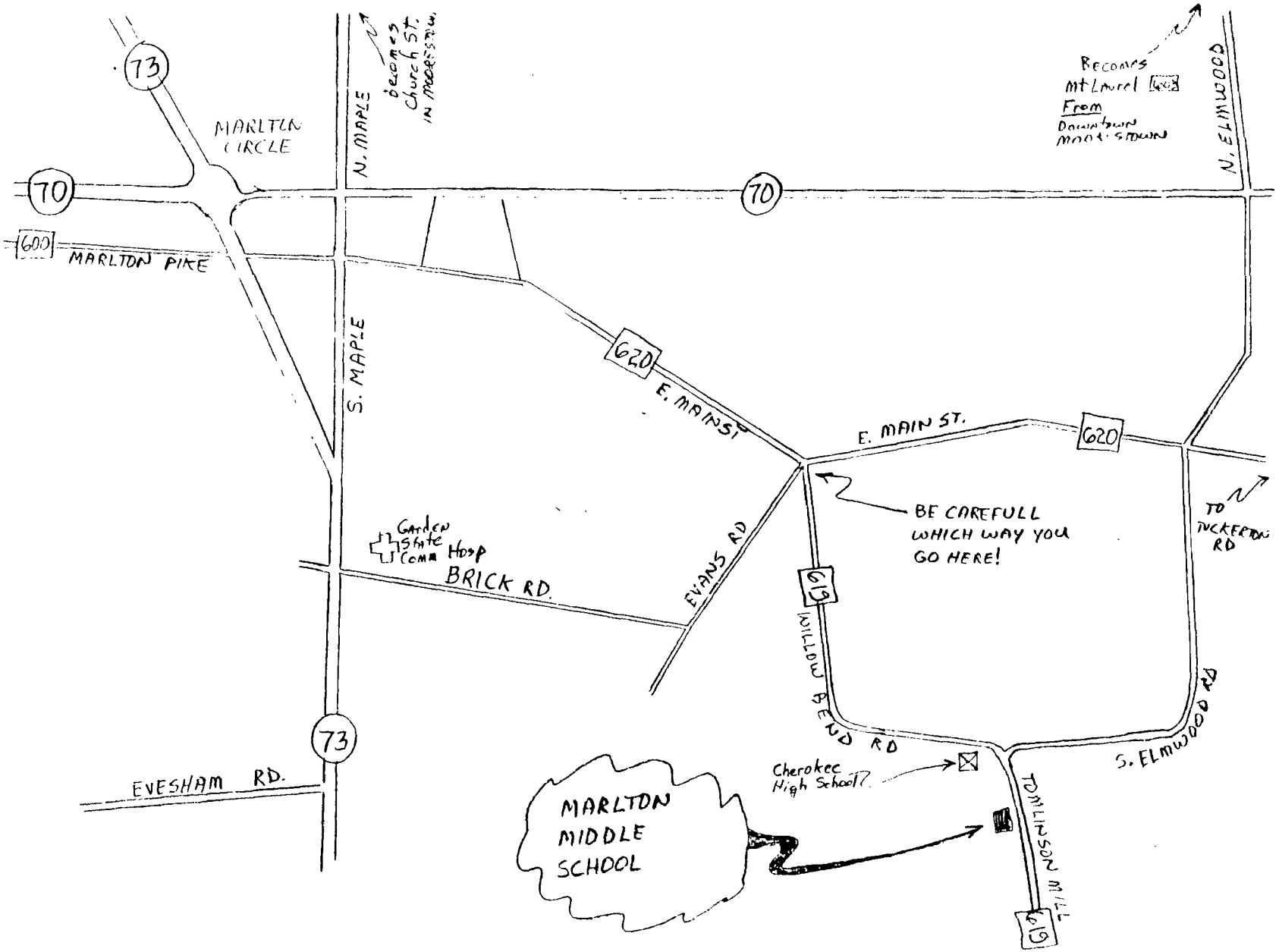
Next month, I'll tell you how to save procedures and give you an easy form of Logo using one-letter commands.

"BYE -- AND A PLEASANT DAY TO YOU!"

==========

THIS TUTORIAL VIA:

THE PHILADELPHIA AREA
TI99/4A USERS GROUP


73
MARLTON CIRCLE
70
600
MARLTON PIKE
N. MAPLE
becomes Church ST. in Moorestown
Becomes Mt Laurel From Downtown Moorestown
N. ELMWOOD
70
S. MAPLE
620
E. MAIN ST
E. MAIN ST.
620
BE CAREFULL WHICH WAY YOU GO HERE!
TO PUCKERTON RD
Garden State Comm Hosp
BRICK RD.
EVANS RD
619
WILLOW BEND RD
S. ELMWOOD RD
Cherokee High School
TOMLINSON MILL
619
73
EVESHAM RD.
MARLTON MIDDLE SCHOOL

THE 9900 USER'S GROUP, INC.
P.O. BOX K
MOORESTOWN, N.J. 08057

Edmonton 99'ers Users Society
PO Box 11983, Edmonton
Alberta, CANADA T5J-3L1

Exchng