

CALL NEWSLETTER

VOLUME III NUMBER 1 January 1985

Atlanta, Georgia

Н

 \mathbf{R}

I

ELLEN WOODALL IS DYING...AND SHE NEEDS YOUR HELP

Marshall Gordon, who has been the president of the Atlanta 99/4A Computer Users Croup for the past two years, makes the U R G E N T Appeal to help save life of his only sister.

Ellen has primary biliary cirrhosis of the She is not now, nor has she ever been an alcoholic, but the cirrhosis is just as deadly as that associated with Doctors cannot explain her illness . It is a rare disease. No one has discovered what causes it.

Her Ellen lost her insurance coverage. illness keeps her from working. She s divorced and supporting herself and two young sons with savings.

Ellen's condition is deteriorating much more rapidly then predicted, and she now must have a liver transplant within three to six months. That's not much time to raise \$200.000.00. At the present time with the help of her family and friends, she has raised \$20,000.00. That will allow her to at least be evaluated and be placed a waiting list for the transplant. We must act quickly to try and save her.

P | F A S F. send contributions to:

Ellen's Liver Transplant Texas Commerce Chemical Bank Attn: Becky Lay P.O. Box 6887 Houston, Texas 77265

********** **ELECTION****ELECTION****ELECTION**

I know that this is the age of apathy but we need of three people to head up seem to be having a special spell of it here positions. And we need people to help within the club. We have had very few mail these committies. Come on beoble. within the club. We have had very few Mail in ballots returned to us. So unless all of you are planning on coming to the meeting on the 20th of January, and I do wish it was true, how about sending in your ballots. We have had a hard time finding enough people to give up some of their time to help us all, so there are no contested positions on the ballot. But, I think that the people who are running would feel better if they had your support if only on the election form.

Send in your ballot today, or come to the meeting and have your voice heard.

following is a list of those nominated, and those who have volunteered to serve as chairpersons of the groups committees;

President Vice President Sec/Treasurer Newsletter Chmn Librarian/BBS Sysop Bob Willis Publicity Chmn Program Co-Chmn

Gary Matthews Jim Hubbard Billy Glass Marshall Gordon Bill Kleinsorge Pat Cameron Jim Hubbard

However the jobs are not covered completely we still need a Recruitment Chairperson, and an Education Chairperson, and possibly a Youth Group Chairperson. In addition we have had, once again, to combine the position of Secretary and Treasurer. So we are still in

positions. And we need people to help out on these committies. Come on people, lend a hand -- the club needs your help.

TI MOVES TO A NEW LOCATION

The club received a letter from II, the text of which follows:

Dear Depot Customer:

On December 3, Texas move to a new location. Texas Instruments will The address is:

Texas Instruments Inc. 5515 Spalding Drive Norcross, Georgia 30092

The physical location is the corner of Spalding Drive and Ga.141 (north of Technology Park).

We are looking forward to seeing you in our new facility.

Regards.

John W. Dendy Atlanta Branch Manager

The new phone numbers are: 662-7900 Customer Service 662-7907 Exchange Center

PRESIDENTS CORNER

This is the last Presidents Corner I get to write. So as a last official? act let me wish all of you a Happy, Healthy New Year.

I was thinking about how to sum up computers over the last two years, but the funny thing is though that when I think back, the interesting and important points are not computers or hardware, or software, its the people - all the people. In the club, out of the club, over the phone, at the meetings, in the letters I get. They've been what has made this time really enjoyable, interesting and worth the investment of time and energy.

So for the last column I'd like to talk about people instead of computers, hope you don't mind. First I'd like to thank Gary Mathews for his help and assistance over the past two years. Especially over the last six months when he's been carrying a good many parts of this club, when I or anyone else didn't have the time he always did. I'd like to thank my wife Elise who sat out front during the meetings to make sure the business part of the meeting was taken care of. For her support and efforts on behalf of this club a large round of applause.

To Bob Willis our Librarian and Sysop of the club BBS, Bob I'll never know where you got the time and energy that you've put into the club but my hearty thanks goes to you. To all of you who have contributed to the newsletter, to stand up and speak at the meetings, give demonstrations etc, I also owe you all a large resounding round of applause. To everyone who has been at the meetings and took part in the gab session goes my thanks also, I've learned a great deal from all of you. I hope all of you got as much as I did. To Ralph Danson, and Billy Glass who started our East and South Side Groups, and to Ralph, and Pat Hester and all the others who created and ran one of the best Computer Fairs I've been to another round of applause is deserved

If I really do this properly I should name just about everybody in the club, you have all helped. Large helping or small helping, it has helped me and the club, so to one and all THANKS, THANKS, THANKS.

MARSHALL

CALL NEWSLETTER

CALL NEWSLETTER is the voice of the Atlanta 99/4A Computer Users Group. P.O.Box 19841, Atlanta, GA 30325. It is published at least 10 times a year. The A9CUG is incorporated as non-profit and is not affiliated with any commercial company or organization. CALL NEWSLETTER is published by and for the Members of the A9CUG to enhance their knowledge of home computers. CALL NEWSLETTER is composed of articles written and/or donated by members of our group and from articles appearing in other Home Computer Users Groups around the world. Opinions expressed by the authors do not necessarily represent those of the Officers or members of the A9CUG. Permission is hereby granted to any users group receiving our newsletter to

Permission is hereby granted to any users group receiving our newsletter to reproduce any article in this newsletter, unless the article is otherwise noted, provided credit is given to the author and to CALL NEWSLETTER. The APCUG freely exchanges Newsletters with other Users Groups around the country. If another club would like to receive our Newsletter but does not have one of their own to exchange; we will gladly send it to them. We do ask that they send \$5 a year or 50 cents a Newsletter to help cover costs. Membership is open to families and individuals who own or are interested in using and programming home computers. Membership includes copies of this newsletter as they are published, access to meetings of the main group and sub-groups, subject to the rules of the sub-group, and the groups Fublic Bomain Library. Annual dues are \$15.00.

Marshall Gordon
Gary Matthews
Elise Gordon
In transition
We need one
Bob Willis
HELP HELP
HELP HELP

President
Vice President
Secretary/Treasurer
Newsletter Chairman
Program Chairman
Library Chairman/BBS Sysop
Education Chairman
Recruitment Chairman

EAST SIDE CHAPTER (ESCUG)

Ralph Danson Bill Dickinson Pat Hester Robert Murphy

President Vice President Secretary/Treasurer Librarian

....

SOUTH SIDE CHAPTER

Billy Glass Bennis Hawkins We need one President Vice President Secretary/Treasurer

REVIEWREVIEW***

QS-WRITER

I just purchased a most impressive program. Let me say that this is not the first program I've purchased from **Quality 99 Software** and I've been very pleased with all the programs I've gotten from them. The nicest part about these programs is that I don't have to do a lot of work to get them to do what they are supposed to do. I can just load them and follow the screen directions.

The most interesting thing about this program is that it really doesn't do anything. It's a loader program, it only loads one program and the program doesn't even come with it. So why do I like it, because it will load and run TI-WRITER with the Extended Basic Module or with the Editor/Assembler Module or with the Mini Mem Module. This way I can run TI-WRITER without changing modules, and is that ever a benifit. If you are thinking about purchasing this program the minimum equipment required is one of the above mentioned Modules, a disk system, and 32k RAM. Anyone who wishes to be able to use TI-WRITER can purchase QS-WRITER, obtain the Public Bomain version of II-WRITER from the Library and away you go.

The program has a few problems as I look at it, number one is that it can only be used on the disk supplied by Quality ?? Software, which is single sided, single density. Therefore if you have a double sided system, as I do, you are wasting half a disk. However even after you load the TI-WRITER programs onto the disk you still have over 200 sector of useable space. I estimate that a two page letter will use 10 sector, so you would be able to write 20 letters before you fill the disk. Then you just transfer the files to another disk delete the old files and start again.

A second problem is that the program goes back to the color bars every time you want to switch from Editing to Printing to Utility, so you have to reload everything every time. This takes several extra steps that are not necessary when using the TI-WRITER Module. The third problem and one that I think is serious is that you do not have the 'SHOW DIRECTORY' function of TI-WRITER when you enter SD and hit 'EMTER' you go to the color bar screen and you have to reload the program, everything in memory is gone. I can never remember the name of the letter I wrote last week and I miss that function.

The last problem I came across is when using DRAGONSLAYER'S AUTO SPELL-CHECK, for some reason the use of small letters is lost, if you make a spelling error and use SPELL-CHECKers correction function everything you type is in capitals and you have to go back into the Editor mode of TI-WRITER to change spelling corrections back to chall letters.

Even with the problems that I have described QS-WRITER is well worth the money, and does a fine job. If you, like me, are tired of swapping modules or you have not been able to purchase a copy of TI-WRITER. This program is probably the answer you have been looking for.

This program can be purchased from: QUALITY 99 SOFTWARE 1884 Columbia Rd. #500 Washington, BC 20009 (202) 667-3574

Or from your local or mailorder dealer.

MARSHALL

Here's an offer for all of you who have just rushed out to purchase QS-WRITER after reading my review. I'd like to run a class on TI-WRITER for all those who are interested. Even if you're using the TI-WRITER Module. It will be a self help type class, that is I will present the material and go over it, but you will have to help with demonstrations of the use of what we go over. Also I will ask volunteers to research special topics and show the results of their research to the class.

If anyone is interested let me know at the next meeting. We will have to get together to determine a time and a place for the class. The first suggestion is to meet once a month just before the regular meeting from 1:30 to 2:30. The second is meeting once a week, rotating the location at the homes of the members attending the class. I'm open to any other suggestions.

MARSHALL

XXXXSIGXXXSIGXXXX

Anyone interested in starting a SIG or Special Interest Group? We could use one on Multiplan, Basic, Extended Basic, Assembly Language, Forth, Modem use, Ftc.

Anymone interested in starting a self-help group on one of the above or any topic that you're interested in?

I know, I know. It takes time, everything that you want to do takes time. But let me tell you it takes a lot less time when there's a group of you as opposed to when your doing it alone.

If you're interested get in touch with one of the officers. New or Old at the next meeting.

CURSOR CURSOR WHAT'S HAPPENED TO MY CURSOR

The next page and part of the following page contain articles on changing the look of the cursor that blinks on your screen. Please note that these programs come from many people from all over the country. They were given to their local Users Group printed in their Newsletter and exchanged with other groups around the country. Spreading the knowledge that is building about the 99/4A.

The first program actually 'POKEs'values into the low memory expansion the Assembly language area of the RAM expansion. This is the first that I've seen of programs that allow you to do what every has been crying about since they got their 99/4A. Poke values directly into RAM als the Apple.

Line 4 pokes into memory the ASCII values of C (67),U (85),R (82),S (83),O (79),R (82). Line 5 contains the decimal equivalent of the hex code for the CHAR subprogram (see pages II-76 to 78 in the Memor's Reference Guide). Decimal 48 equals 30 in Hex, 63 equals 3F, 255 = FF, 254 = FE, 124 = 7C, 24 = 18, 8 =0C. The method of getting and using the pattern codes is explained in the CHAR subprogram noted above. You can use any Hex to Decimal program to convert the values and substitute.

Here are two other set you can try, the first, if you want to look like a business cursor, is 0, 0, 0, 0, 0, 0, 126. The second is 60, 66, 165, 129, 165, 153, 66, 60.

Happy Hackin

MARSHALL

LEHIGH 99'ER COMPUTER GROUP

new cursor routines

Enough itemization. The simplest way to demonstrate the neccessary elements to get an AL program to go is to list a couple. This program was passed to us by bless-his-big-heart Jim Peterson, who hopefully is managing to get a couple bucks from us tightfisted hardhearted users. (Better to get blood from a stone, Jim.) It's a neat gimmick which turns your cursor into whatever shape you like. The Columbus, GA group in the REM statement is Columbus 99/4A UG, c/o Dr. R.G. Albright, 3434 Flint Drive, Columbus, GA 31907 -- Rom can write' and puts together a top notch newsletter.)

1 'TEXAS CURSOR from GOTO Newsletter of Columbus, GA
Users Group; unattributed
2 CALL CLEAR :: CALL INIT
3 CALL LOAD(8196,63,248)
4 CALL LOAD(16376,67,85,82,83,79,82,48,8)
5 CALL LOAD(12288,48,48,63,255,254,124,24,12)
6 CALL LOAD(12296,2,0,3,240,2,1,48,0,2,2,0,8,4,32,32,36,4,91)
7 CALL LINK("CURSOR")
8 INPUT X8 :: GOTO 8

Line 3 sets up the REF table pointer in low memory expansion at >2004 with >3F, >FB. Then at >3FF8 {16376} it sets up a program named CURSOR that starts at >3008. Next in lines 5 and 6, beginning at >3000 a DATA entry of 4 words is set up (change the next 8 numbers to change the cursor). Finally, the AL is LOADed into memory and LINKed. The ultimately curious might check to see if, indeed, it's the same program as the next one.

This version comes from HOCUS (put out by the Milwaukee Area 99/4 UG, 2007 M 71st St., Maumatosa, MI 83213), who, in turn, credit Tony Johnson of the Houston Users Group. The DATA changes the cursor to an underscore character.

DEF CURSOR

VMBW EQU >2024 IBASIC equate (IB can't REF)
NEWDEF DATA 0,0,0,>7E00 BATA forces whole words
CURSOR LI R0,>3F0 \$ 1008 in decimal

The screen image (one byte per char. on screen)
goes from VDP location >0 to 2FF (0 - 767, dec)
The Pattern table base is also at location 0, but
since each pattern takes 8 bytes and there's 256,
only >300/8=>60 or 96 patterns are lost. Anyway,
BASIC adds >60 to the CHR\$(num) to find the pattern
and since the cursor is >1E (30, dec) everything
works out thus: in MEX 1E 40 + 9 # . 3FO (FORTHed)
in decimal (30 + 96)## = 1008

LI RI, MEMDEF \

LI R2,8 > THESE THREE SET UP FOR BLUP SYMBOL / YMBOLUTILITY

RT

END

f This version is for BASICs; ED/ASH won't work.

1 For ED/ASM environment

DEF CURSOR

REF VMBN

MEMDEF DATA 0,0,0,>7600 DATA forces whole words CURSOR LI R0,>8F0 & 2288 in decimal

The Pattern table base is at >800.
works out thus: in HEX IE # \$ 800 + . 8F0
in decimal 30#8+2048 = 2288

LI R1, MEMDEF LI R2, 8 BLWP 3VMBW RT END

CURSOR CURSOR -- CONT

AORG >7000

```
# For MINIMEM's LINE-BY-LINE assembler
# EB/ASM's version will LOAD & RUN ok, but
# you can't write AL that way with L-by-L.
```

MN EQU >6028
ND DATA 0,0,0,>7E00 DATA forces whole words
CR LI R0,>8F0 % 2288 in decimal
LI R1,ND
LI R2,8
BLWP ЭНW
FF B #R11 RT may not be implemented
END

For MINIMEM you have to hand-LOAD the DEF table. # There's one pointer to its start, as well as one I for the next available byte after your program. # Together, these two are used to figure how memory # is left over: LFAM - FFAM (Last Free Add. in New. minus First Free Address in Memory) I LFAM is at >701C; it shows the start of DEF table (a list of LOADed programs) # FFAM 15 at >701E I hand-LOAD (using AORS, TEXT and DATA directives # like this: AORG >70IE we'll do FFAM lst. DATA FF+2 LABEL at B #R11 AGR6 >701C show current LFAM subtract 8 bytes from what's displayed and then DATA (new HEX value) AORG (new HEX value) TEXT 'CURSOR' puts the pgm name in DATA CR puts the pgm start in \$ (all this assumes you didn't exit the Line-by-! Line assembler until you're done -- and I'm not Certain if you can recover the values of FF and CR # if you skip this part-- it's probably in the VDP & 1 disappears when you quit. # All versions use the GPLWS at >83EO. I've had # problems with that when I. a syntax error is created \$ & 2. the routine was executing off the USER INTERRUPT at >83C4 # Recommendation; create your own workspace.

"SPEAK-N-PROOF": THE TALKING PROOFREADER Roy T. Tamashiro, Ed.D.

Talking about joint effort how is this the article below is from the Cin-Day Users Group, the program TRANSLATOR programs came from the Central Iowa 99/4A Users Group (original program), the modifications came through the Leghigh 99'er Computer Group, through Cin-Day and to us, the 99/4A will keep going thanks to the efforts of people and groups like these.

FREE PROGRAMS: by Ed York

The programs listed on the next page comes to us from two separate sources. The programs are entitled "SFEAK-H-FROOF PART ONE". "SPEAK-N-FROOF PART TWO" and "TRANSLATOR". Since notody, and I do mean notody, types perfectly, a "talking "TRANSLATUR . Since nowley, and talking nebody, types perfectly, a "talking proofresder" is the next best thing when you cannot find a willing friend to read back to you what you've typed. Whether you are writing a document on a word processor such as II-WEITER, or typing in a BASIC or EXTENDED BASIC program, and takes are unrually anevitable. If is a Bosic or EXTENDED BASIC brobram, Mistakes are virtually inevitable. If is so leasy to omit a key word or transpose a number. It can be quite difficult to detect typing errors in a BASTR on EXTENDED BASIC program especially when there are a long e a long series of numbers or It is quite nice to have the symbols. It is quite nice to have the Speech Synthesizer read back to you what you have typed, while you visually check this against the original document or check this against the original document be this against the original document of program listing. Since your eyes will be focusing on one listing rather than two, you are less likely to lose your place as you check your typing. The purpose of SPEAK-N-PROOF PART ONE is to reformat your testing and that it text file or program listing so that it can be used by SPEAK-N-FROOF PART TWO. will be asked if your text file or YOU program listing contains both upper and lower case letters. A text file would most likely contain both upper and lower case letters, whereas a program listing would most likely contain all upper case letters. The purpose of SPEAK-N-PROOF The purpose of SPEAK-M-PROOF to read back to you. via inthesizer, the text file or PART TWO is Synthesizer, the text file or listing that has been processed by -PROOF PART ONE. You will also be Speach program listi SPEAK-N-PROOF asked if your text file or program listing contains both upper and lower case letters. You should logically answer this we: did dumetion question the same speak-N-PROOF PART ONE. 35 you. You will also to banze Aon Misp asked 15 for the Speech after Synthesizer to pause after each line. Since each line of either the text file or the program listing is displayed on the screen, you may elect to have the speech synthesizer pause while you inspect each line. The purpose of TRANSLATOR is to translate a BASIC or EXTENDED BASIC EXTENDED net program listing, that has not processed with SPEAK-N-FROOF PART ONE. into a merge format which can then be called by EXTENDED BASTO only! Any files saved with either the TI-WPITER or EDITOR/ASSEMBLER Command Modules are saved in a DISPLAY/VARIABLE 80 format. However, in order to save a grogram listing in either BASIC or EXTENDED BASIC, you simply

you program into memory and type in following: LIST "DSK1.FILENAME". the following: LI Note: DO NOT TYPE IN THE ENDING PERIOD! This will save your program listing in a DISPLAY/VARIABLE 80 format which can be SPEAK-N-PROOF PART ONE AND both Now you can write EXTENDED BASIC proora TRANSLATOR. either BASIC programs using the ŢI-ŴŔĨŤĔŔ QΥ EDITOR/ASSEMBLER Command Modules and have TRANSLATOR program translate COUR into a merge format. broðram Note: After a program has been processed by TRANSLATOR, you will find that TRANSLATOR places an EXTENDED BASIC REM ("!") in places an 'EXTENDED front of each line. The advantages gained

100 REM SPEAK-N-PROOF 110 REM 120 REM PART ONE 130 REM 140 REM WRITTEN BY: 150 REM 160 REM ROY T. TAMASHIRO 170 REN 180 DISPLAY AT(4,6) ERASE ALL :"# SPEAK-N-PROOF #" :: DISP LAY AT(6,10): "PART ONE" :: DISPLAY AT (12,1): "ENTER THE FILE NAME BELOW: ": : "DSK1." 190 ACCEPT AT(14,6)51ZE(10): A\$:: B\$="DSK1."&A\$:: DISPL AY AT (16,1): DOES THE TEXT F ILE CONTAIN": : "BOTH UPPER A ND LOWER CASE" 200 DISPLAY AT(20,1):"LETTER S? Y" :: ACCEPT AT(20,10)SIZ E(-1)VALIDATE("YN"):C\$:: DI SPLAY AT(10,4) ERASE ALL: "I A M REFORMATTING THE" 210 DISPLAY AT(12,4): "FILE, SO PLEASE WAIT. .: Dr. N #1: B\$, INPUT , VARIABLE 80 :: OPE N #2:"DSK1.PROOF-FILE", OUTPU T.INTERNAL, VARIABLE 90 220 IF C\$="Y" THEN DPEN #3:" DSK1.UPPER", DUTPUT, INTERNAL, VARIABLE 90 230 LINPUT #1:D\$:: IF C\$="N " THEN 260 240 E\$="" :: FOR A=1 TO LEN(D\$):: F\$=SEG\$(D\$,A,1):: IF (ASC(F\$) >96) AND (ASC(F\$) <123) T HEN FS=CHR\$(ASC(F\$)-32) 250 E\$=E\$&F\$:: NEXT A :: PR INT \$3:E\$ 260 PRINT #2:D\$:: IF EDF(1) =0 THEN 230 270 CLOSE #1 :: CLOSE #2 :: IF C+="Y" THEN CLOSE #3 280 DISPLAY AT(10.6) ERASE AL L: "THE FILE HAS NOW": :"

BEEN REFORMATTED.

100 REM SPEAK-N-PROOF 110 REM 120 REM PART TWO 130 REM 140 REM WRITTEN BY: 160 REM RDY T. TAMASHIRO 170 REM 180 CALL CLEAR 190 PRINT TAB(6): ** SPEAK-N-PROOF *": : 200 PRINT TAB(10); "PART TWD 210 PRINT "DOES THE TEXT FIL E CONTAIN*: : 220 PRINT "BOTH UPPER AND LO MER CASE": : 230 PRINT "LETTERS?": : : : 240 CALL KEY(0, A, B) 250 IF (A()89)+(A()78)=-2 TH EN 240 260 PRINT "SHALL I PAUSE A M OMENT AFTER": : "EACH LINE? Y 270 CALL KEY(3,C,B) 280 IF (C(>89)+(C(>78)=-2 TH EN 270 290 CALL CLEAR 300 IF A=78 THEN 320 310 OPEN #3: "DSK1.UPPER", INP UT , INTERNAL, VARIABLE 90 320 OPEN #1: DSK1.PRODF-FILE ",INPUT ,INTERNAL,VARIABLE 9 330 OPEN #2: "SPEECH", DUTPUT 340 INPUT #1:A\$ 350 PRINT A\$ 360 IF A=78 THEN 380 370 INPUT #3:A\$ 380 PRINT #2:A\$ 390 IF C=78 THEN 440

by either reading or writing your programs with the TI-WRITER or the EBITOR/ASSERBLER Command modules are many. You have the full use of the editing capabilities that are available with either TI-WRITER or EDITOR/ASSEMBLER. SPEAK-N-PROOF PART ONE and TWO were written by Roy T. Tahashiro and was published in the St. TRANSLATOR was program idea by John Hamilton (of the Central Iowa 97/4A Users Group), modified by Frederick Hawkins (of the Lehigh 59 er Computer Group) and a second modification was made by Ed York (of the CIM-DAY Users Group).

400 PRINT :" PRESS ANY KEY T O CONTINUE." 410 CALL KEY(O, D, E) 420 IF E=0 THEN 410 430 CALL HCHAR(23.1,32,32) 440 IF EDF(1)=0 THEN 340 450 CLOSE #1 460 CLOSE #2 470 IF A=78 THEN 490 480 CLOSE #3 490 FND WE w18Hyou

NEW

YEAR

100 ! TRANSLATOR 110 ! 120 ! PROGRAM IDEA BY: 140 ! JOHN HAMILTON OF THE 150 ! 160 ! CENTRAL IONA 99/4A 170 ! 180 ! USERS GROUP 190 ! 200 ! PROGRAM MODIFIED BY: 210 ! 220 ! FREDERICK HAWKINS 230 ! 240 ! OF THE LEHIGH 99'ER 250 ! 260 ! COMPUTER GROUP 270 280 PROGRAM MODIFIED BY: 290 300 ! ED YORK OF THE 310 ! 320 ! CIN-DAY USER GROUP 340 DISPLAY AT(6,10) ERASE AL L: "TRANSLATOR" :: DISPLAY AT (10.1): "ENTER FILE TO BE CON VERTED: ": : DSK1." :: ACCEPT AT (12,6)SIZE (10):A\$ 350 DISPLAY AT(14.1): "ENTER FILE TO BE MERGED: :: : "DSK1. " :: ACCEPT AT(16,6)SIZE(10) :B\$:: OPEN #1:"DSK1."&A\$:: DPEN #2: "DSK1. "&B\$, VARIABLE 360 LIMPUT #1:C\$:: A=PDS(C\$ " ",1):: ON ERROR 400 :: B= VAL (SEG\$ (C\$,1,A)) 370 ON ERROR 390 :: C=INT(B/ 256):: A\$=CHR\$(B-C\$256) 380 PRINT #2:CHR#(C);A#;CHR# (131);SE6\$(C&,A+1,B0);CHR\$(0):: 60TO 360 390 PRINT #2:CHR\$(255);CHR\$(255):; CLOSE #2 :: DISPLAY A T(22,6): "PROCESS COMPLETED!" :: END 400 ON ERROR 390 11 RETURN 3

**** ASSEMBLY LOADER ****

This is an assembly loader routine that automatically loads assembly programs with no program names.(Like Atarisoft games or some TI games) It will automatically load ANY assembly program including those with program names. After loading those with program names, it returns you to the Editor/Assembler. Just select option 4 (RUN) and type the program name. It will then run.

Another way to do it is just make an extended basic CALL LOAD(FILENAME), CALL LINK(PROGRAM NAME). Then the program will run automatically from extended basic. I tried to use an external REF to get my program to run the one you select with a program name, but both programs have to be already loaded for it to work. (Self-defeating isn't it?).

There are a few things you will have to change to adapt the routine to run the specific programs you want to use with this routine. After the listing, I'll explain what you'll have to change.

```
Here is the listing:
REF DSRLNK, OMBW, VSBW, LOADER
REF KSCAN
DEF BEGIN
PABBUF EQU >1000
PAB EQU >F80
STATUS EQU >837C
                   EQU >8356
PNTR
SAVRTN BATA 0
TDATA DATA >0005, PABBUF, >5000, >0000
BATA >000B
TEXT 'DSK1.TENNIS'
                  EVEN
                   DATA >0005, PABBUF, >5000. >0000
CTIATA
                   DATA >000E
TEXT /DSK1.CENTIPEDE/
                   EVEN
CLOSE MYREG BSS >20
TEN TEXT '1. TENNIS'
CEN TEXT '2. CENTIPEDE'
BEGIN LI RO, 34
LI R1, TFN
LI R2,9
BLWP @UMBW
LI R0,98
LI R1,CEN
LI R2,12
KPREP CLR RO
MOVB RO,@>8374
LI R4,>3100
LI R2,>3200
LI R3,>2000
KCHECK CLR RI
BLWP @KSCAN
MOVB @STATUS.R5
COC R3,R5
JNE KCHECK
MOVB @S8375,R1
                   BYTE >01
CLOSE
                   MOVE @>8375,R1
                   CB R1.R4
JEQ TLOAD
                   CB R1.R2
JEQ CLOAD
JMP KPREP
                   MOU RITTOSAURTA
 TIDAD
                   LI RO, FAB
LI R1, TDATA
LI R2, 200
BLWF CYMBW
```

LI R6, FAB+9

```
MOU R 4.0PNTR
BLWP @LOADER
JMP CLOSEF
CLOAD MOV R11.0SAVRTN
LWFI MYREG
LI R1,CDATA
LI R2,\20
BLWP @VMBW
LI R4.0PNTR
BLWP @LOADER
CLOSEF MOV R4.0PNTR
MOV R 6.0PNTR
MOV R 6.0PNTR
MOV R 6.0PNTR
BLWP @CLOSE,R1
LI R0.0SEW
MOV R 6.0PNTR
BLWP @CBRLNK
BATA 8
CLR R0
MOVB R0.0STATUS
MOV @SAVRTN,R11
RT
```

Well, that's the listing. Now I'll try to explain what you'll have to change to adapt the routine to your use. You'll want to change the program names from CENTIPEDE TENNIS to whatever you want. You don't have to just have 2 programs either, that was just for simplicity.

Back up in the beginning of the listing there are 2 symbols TDATA CDATA. To put your files in place of the 2 I used, you'll want to change the command TEXT 'DSK1.TENNIS' to TEXT 'DSK1.pfile name'. In the DATA statement directly above that, there is the statement DATA >000B. >000B is the length in characters of the file name DSK1.TENNIS in hexadecimal. Change that to however long your file name is including "DSK1.". For example, "DSK1.LOADER" would be 11 characters long, so the data statement would read DATA >000B and the text below it would read TEXT 'DSK1.LOADER'. To add more programs to the routine, copy the first line of DATA from either TDATA or CDATA(it's the same), put TEXT 'DSK1. file name' below it, put another DATA statement below it with the length of the filename below it, and put an EVEN statement below it(like in the program.)

There are a few other things you have to do to add more programs to the routine. Two lines after the KPREP symbol, you'll see a command LI R4,>3100 That's the ASCII code in hex for "1". You'll have to load registers 7-15 with the hex ASCII codes for 3-9. Next you'll have to put a CB R7,R1; R8,R1 ... with a JEQ following each one telling it to jump to a symbol you create. The symbol should have the commands just like US...

SYMBOLS TLOAD CLOAD do from MOV R11, GSAVRTN TO JMP CLOSEF. All of that should be under the symbol you create. That should be all you need

Good Luck and I hope you enjoy it!.
Jim Rice

YOU

for

disk and n. If you

PLEASE READ

This is by T.I. a

before its use.

Paul Hickey

```
FORTH PROGRAMMING... by Mike Dodd
The following TI-FORTH program is a disk utility that SCR #11
                                                                0 ( DISK UTILITIES, PAGE 3)
enables you to:
                                                                 1 ADDR 4 /MOD BLOCK SWAP 100 $ + C + DUP ADDRESS !;

    Catalog the disk in BSK1.

                                                                 2 :RT ADDR CO TPE C! :
                                                                                                : PRT? T 8 AND ;
    2. Unprotect every file in DSK1.
                                                                3 : BT SD 2 2- 2 -2 SD +! :
    3. Protect every file in DSK1.
                                                                4 : PRT PRT? O= IF UPDATE T 8 + ADD C! ENDIF ;
    4. Go through the disk catalog in DSK1, (un)protecting
                                                                 5 : UMPRT PRT? IF UPDATE T 8 - ADD C! ENDIF;
        whichever files you wish.
                                                                 6: PROTECT-ALL SET-UP CNT @ 1 DO RT PRT I 5 MOD 0= IF FLUSH
The protection clause is the clause you can give the file in
                                                                    ENDIF LOOP FLUSH :
the Disk Manager by using the "MODIFY FILE PROTECTION"
                                                                 8 : UNPROTECT-ALL SET-UP CNT 2 1 DO RT UNPRT I 5 MOD 0= IF FLUSH
                                                                     ENDIF LOOP FLUSH:
The 4th item operates much the same way as the Disk
                                                                10 : YNM ." Yes or No? " ;
Manager's "DELETE FILE" routine, except all it does is
                                                                11 : YN YNH KEY 59 = ;
delete the file protection.
                                                                12 : ASK BT RT YN IF PRT ELSE UNPRT ENDIF ;
To type in the program, Press 9 EDIT and type in the listing
                                                                13 : DOFILES 128 1 DO RD O 4 GOTOXY DROP BUFR CO IF .FILE
under "SCR 49". Next, Press FCTN 4 and type in the listing
                                                                     .* PROTECT" ASK ELSE LEAVE ENDIF I 5 MOD 0= IF FLUSH ENDIF
 under "SCR #10". Repeat for screens #11 and #12.
                                                                     LOOP FLUSH :
Before loading the Disk Utility, place the TI-FORTH master
 disk in drive 1 and type _FLOAT __FILE and press ENTFR.
                                                               SCR #12
                                                                 o ( DISK UTILITIES, PAGE 4)
 To load the utility type 9 LOAD
                                                                 1 : PROTECTION SET-UP PAGE UNCAT . HEADER DUFILES CLSE ;
 To catalog the disk in DSK1, type CATALOG
                                                                 2
 To unprotect every file in DSK1, type UNPROTECT-ALL
                                                                 3
 To protect every file in DSK1, type PROTECT-ALL
 To go through the catalog in DSK1, (un)protecting whichever
                                                                 5
 files you wish, type PROTECTION.
 SCR #9
   O ( DISK UTILITIES, PAGE 1) BASE->R HEX
                                                                      Thanks to Mike Dodd and the K-TOWN 99'er the Newsletter of the K-TOWN 99/4A Users
   1 0 VARIABLE ADDRESS O VARIABLE TPE O VARIABLE ENT
                                                                 10
   2 0 VARIABLE TPEP
                                                                      Group.
                                                                 11
   3 O VARIABLE MBUFR 8 ALLOT
                                                                 12
   4 O VARIABLE BUFR AF ALLOT
                                                                 13
   5 : T TPE CO : : PAGE O O GOTOXY CLS ;
                                                                 14
   6: TP TPEP CO:
                                                                 15 R->BASE
   7 : . NAME NBUFR A 20 FILL BUFR 1+ NBUFR BUFR C2 CHOVE
   8 NBUFR A TYPE;
   9 PABS & A + BUFR 1700 FILE CATFILE
                                                                                      TERMINAL EMULATOR 3
  10 : OPNCAT CATFILE SET-PAB RLTV INPT INTRNL F-D" DSK1." OPN ;
                                                                        The Texas Instruments TERMINAL EMULATOR
  11 : NEUNV BUFR DUP Ca + 2+ SWAP 9 & + FAC 8 CHOVE FAC->3;
                                                                                                           TE3 almost has a
                                                                        III is a great program. TE3 almost has a mind of it's own. The program in short is incredible, Lets say the computer is
  12 : .SIZE 1 NCONV 4 .R;
  13 : .FLE TP 1 = IF . DS/FX " ELSE TP 2 = IF ." DS/VR " ELSE TP 3
                                                                        running behind, the program will speed up until it has all the data and them will
  14 = IF ." IN/FX " ELSE ." IN/VR " ENDIF ENDIF ENDIF 2 NCONV 3
                                                                                   to the normal speed. If you to sign off and the host computer
  15 .R;
                                                                        return to
                                                                        has started a transmission the
 SCR #10
                                                                        automatically stop the termination mode and put the data on a disk. The program
   o (DISK UTILITIES, PAGE 2)
                                                                                has many functions that are a great
   1 : .TYPE O NCONV DUP TPE C! ABS DUP TPEP C! 5 = IF ." PROGRAM "
                                                                                                              It can do some
                                                                        help to the program user.
                                                                        pretty incredible functions one of them is pulse dial (you must have a modem made for this feature). That is a great help
   2 ELSE .FLE ENDIF;
   3 : .PRO T 5 > IF . " Y" ELSE . " " ENDIF ;
                                                                               this feature). That
   4 : .FILE .NAME .SIZE 2 SPACES .TYPE SPACE .PRO CR ;
                                                                         if you have
                                                                                                             set up
   5 : .FILES 127 0 DO RD DROP BUFR CO IF .FILE ELSE LEAVE ENDIF
                                                                                           Íhe
                                                                                            The program as terminal use for
                                                                         business.
                                                                                                          U.S.#
   6 LOOP;
                                                                                                       makes
                                                                                                                  the computer
                                                                                       the other
   7 : .HEADER RD DROP ." DSK1-DISKNAME=" .NAME CR ." AVAILABLE="
                                                                         terminal
                                                                         forget it is a computer and makes it act
       2 NCONV DUP 3 .R . " USED=" 1 NCONV SWAP - 3 .R CR
                                                                                 as a terminal (This option is great
                                                                         for a RBS system). It is on a di
comes with its own documentation.
       . FILENAME SIZE TYPE P" CR
   10
                                                                         forget the functions the function
   11 : CATALOG OPNCAT CR . HEADER .FILES CR CLSE ;
                                                                                              them for
                                                                                will list
                                                                         THE DOCUMENTATION CAREFULLY!
   12 : ADD ADDRESS 0 ;
                                                                        half way completed program by T.I. and
that fact should be taken into account
  13 : SET-UP O CNT ! EMPTY-BUFFERS SP! O BLOCK 100 + ADDRESS !
```

BEGIN ADD & DUP 2 ADDRESS +! 1 CNT +! 0= UNTIL

DROP ;

15

KEY COMMANDS CHANGES PRINT ON GEMINI 10X ON GEWINI

The program below shows you how to print from the EDIT mode of TI-WRITER. By pressing 'CTRL' and 'U' your cursor changes into an underline character, denoting that you are writing Control Characters to the document that the 99/4A will obey. At the end of the article the author states that these controls also work in BASIC and EXTENDED BASIC. Try them, you'll like them.

SINGLE KEYS FUNCTIONS PREFORMED

CTRL N CTRL T	DOUBLE-WIDTH MODE
CTRL Q	PRINTER ON-LINE
CTRL S CTRL D	PRINTER OFF-LINE COMPRESSED (PICA) MODE
CTRL R	COMPRESSED (PICA) MODE CANCEL COMPRESSED MODE
ČTRL G	BELL BELL
CTRL H	BACKSPACE
CTRL I	HORIZONTAL TAB
CTRL J	LINE FEED
CTRL K	VERTICAL TAB
CTRL L CTRL M	FORM FEED
CTRL 8	CARRIAGE RETURN ENDS MACRO INSTRUCTION
CTRL .	ESCAPE (ESC)
FCTN V	DELETE

COMMON MULTI, FUNCTION

ESC 5 ESC 8 CANCEL ITALIC CANCELS PAPEROUT ESC 9 ESC 9 ESC 1 ESC E EMPHASIZED MODE CANCELS EMPHAS. ESC G DOUBLE-STRIKE ESC H CANCELS DBLSTR. ESC S O ESETS PRINTER ESC S O SUPSCRIPT ESC O LINE FEED TO 7/72	
ESC 1 LINE FEED TO 7/72 ESC 2 LINE FEED TO 1/16	

THESE KEYS CAN BE USED TO CONTROL THE ESCAPE FROM WIZARD'S KEEP PRINTER OUT OF A BASIC OR X-BASIC PROGRAM, OR DIRECT FROM THE COMMAND MODE OF THE COMPUTER. !!!! WARNING !!!! I DO NOT RECOMMEND USING THESE IN MULTIPLAN OR ANY OTHER PROGRAM WHICH IS USING CTRL OR FCTN KEYS TO CONTROL EDITING. HAVE FUN WITH THESE AND EXPERIMENT. NEW KEYS WILL BE ADDED AS THEY ARE DISCOURTED. THEY ARE DISCOVERED. (NOTE-YOU CAN USE THEY ARE DISCOVERED. (NOTE-YOU CAN USE THESE IN THE CTRL U MODE OF TI-WRITER) BY DAME K. HEATHERINGTON FROM THE SUNCOAST BEEPER NEWSLETTER OF THE SUNCOAST 99'ER USERS GROUP.

\$ DE-BUG DE-BUG DE-BUG \$\$\$\$\$\$\$\$\$\$\$

From around the country as other groups have found them.

RAY AND DAVE KAZMER but I^\prime m afraid t I^\prime ve lost their groups name. -(I) am that I sorry)

*****TACO MAN CHANGES****

SCORE=0 :: SCREEN=1 :: MEN=2 :: LAY AT(1.1):"TACO MAN'S RECORD: DISPLAY "2STR\$(HS)
530 CALL SCREEN(2):: D=247
570 MEN=MEN2 :: SCORE=SCORE*2 :: DISPLAY
AT(23,1)SIZE(10):RPT\$(CHR\$(100),MEN)::
DISPLAY AT(24,1):"SCORE:"
580 DISPLAY AT(24,7)SIZE(12):SCORE ::
DISPLAY AT(23,20):"SCREEN:" :: DISPLAY
AT(24,22):SCREEN ::SP=1 ::PP=0
885 HS=MAX(HS,SCORE)
890 DISPLAY AT(24,7)SIZE(12):SCORE ::
CALL SOUND(-1,-6,0)
1000 MEN=MEN-1 :: IF MEN=-1 THEN 1110
ELSE DISPLAY AT(23,1)SIZE(10):RPT\$(CHR\$(100),MEN) &STR#(HS) 100), MEN) 1130 DISPLAY AT(12,11)SIZE(9)BEEF:EPT*(
EHR*(100),4);" ";EPT*(CHR*(108),4)
1140 DISPLAY AT(12,11)SIZE(9):"GAME OVER"
1160 DISPLAY AT(13,1):"PRESS 'X' OR FIRE
FOR REPLAY" :: DISPLAY AT(14,1):"PRESS
ANY OTHER KEY TO EMD" :: DISPLAY AT(1,19):HS 1170 CALL KEY(1,K,S):: IF S=0 THEN 1170 1180 _IF K=18 OR K=0 THEN CALL CLEAR :: GOTO 290 1190 **** DELETE THIS LINE ENTIRELY **** 1200 CALL CLEAR :: END

2670 PRINT " PLAY AGAIN? (Y/N)" 2675 PRINT :" (FIRE BUTTON = YES)" 2713 CALL KEY(1,K,S) 2716 IF K=18 THEN 700

スススススススススススススススススス From MUNCH of Mass. we are given the following.

Good news for all who have been looking to ESCAPE FROM WIZARD'S KEEP Here is a sure-fire solution: S-W-S-W-N-N-E-S-S-E-E-N-E-E-S- SAY ROCK. Also if you get into the room with no Also if you get into the room with no exit, you really get trapped. Instead of a chance to play again, you get *SYNTAX ERROR IN 970. Line \$70 should read 970 carr council at 1.1. CALL COLOR(13,1,1)

\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$

GRISLY ADVENTURE

There's a BUG in the program. 2850 should read as follows: 2850 IF (I()G) + (P()F) THEN 2880 As written it is impossible to kill the bear by shooting either right or left. The only way to kill the bear is to shoot it from above or below.

DE-BUG CONT.

You probably already know this, but you don't have to type in the blank spaces before the := in multiple statements in Extended Basic. Just run everything together 100 CALL CLEAR::RANDOMIZE::FOR B=I TO 100::NEXT D, and the computer will separate it for you, shoving statements into additional lines if necessary.

OUT OF MEMORY

Happy hackin'

Jim Peterson

Suggested Modification to FLAGS by Dave Becker (Users Group Unknown)

The FLAGS program on the Club Library cassette has some very good educational cassette has some very good educational graphics (very compactly coded too). Perhaps because my kids are young, however, I wished the "T" feature (see a flag and guess which country) had some positive feedback for beginners. The following code identifies the "mystery country" after two wrong guesses: 391 IF TRY=1 THEN 424 (WAS TO 399) 424 IF K(11 THEN 427 425 D\$=CHR\$(K+54) 426 GOTO 428 427 D\$=CHR\$(K+47) 428 PRINT "ANSWER WAS ",D\$ 429 GOSUB 418 430 GOTO 399 (a delay)

\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$

From the AIRPORT AREA COMPUTER CLUB. Coraopolis, Pa. comes the following:

As a followup, John Dombrowski offers us this technique for switching from JOYST to KEY:

Did you ever want to change a game from joystick to keyboard or vice-versa? Here is an example of a change done in the game 'Aardvark' found in June 83's 99'en Home Computer Magazine. Use this section of 'Aardvark' as an example to change other programs:

Aardvark is like this...
630 CALL JOYST(1,X,Y)
640 IF X=-4 THEN FV=FV-1 :: GOTO 780
650 IF X=4 THEN FV=FV+1 :: GOTO 840
660 IF Y=4 THEN FH=FH-1 :: GOTO 900

670 IF Y=-4 THEN FH=FH+1 :: GOTO 970

Aardvark can be changed to this...

430 CALL KEY(O,K,S) 640 IF K=83 THEN FV=FV-1 :: GOTO 780 650 IF K=68 THEN FV=FV+1 :: GOTO 840 660 IF K=69 THEN FH=FH-1 :: GOTO 900

670 IF K=88 THEN FH=FH+1 :: GOTO 970

** NOTES **

A. This is in Extended Basic.

B. Ir CALL JOYST, the x and y are the variable of the position of the joystic... 4 or -4 is always returned. (See CALL JOYST in the User's Reference

Manual) C. In CALL KEY, we are checking for the K or Key that is pressed. The numbers 83, 68, 69, and 88 are the ASCII codes for the S, D, E, and X (arrow keys).

D. Always use the same logic and variables found in the program. It makes it easier. For example, the GOTO's are the same; FV and FH logic are the same;

and even the line numbers can be the Each Identical line number is doing the exact same logic. For example, in line 640 the X=-4 is left on the joystick, while K=83 is S (left) on Reyboard.

\$

From Doug Cerman in the Ce 99/4A UG Newsletter "4A FORUM": Central Iowa

Have you ever had difficulty finding which joystick is to be used in a program? If the program was written for joystick #1 then joystick #2 won't work and visa versa.

Add the following lines at the beginning

of the program and let the computer do the work.

100 PRINT "PRESS FIRE BUTTON TO CONTINUE

100 FRINT FRESS FIRE BUT 110 CALL KEY(1,K1,S) 120 CALL KEY(2,K2,S) 130 IF K1+K2()17 THEN 110 140 JS=INT(K1/18+K2/9+1)

replace the CALL JOYST statement in

the program with: XXX CALL JOYST(JS,X,Y) And the program will respond to whichever of the two joysticks you were holding when you pressed the fire button.

FOX CITIES From the USERS Newsletter comes a passalong from the Source.

OVERHEARD ON THE SOURCE

One of the features (of the Source) is called FOST. This is a bulletin board where individuals can write comments or questions. The TI-99/4 section is one of this writers favorite sections. The following are some excerpts of messages recently posted: recently posted:

I have been having trouble with TI's PROCRAMMING AIDS III. When using the Editor program to resequence blocks of code, I find it fails to properly resequence line references. For example: 100 ON X GOTO 200,300,400 would resequence to: 100 ON X GOTO 100,100,100. I would appreciate any help that one could lend me in debugging this utility.

Posted the next Posted the next day: I think the following line will correct the bug in the Editor program: 6215 L1=ASC(SEG*(L*,P+1,1):: L2=ASC(SEG*(L*,P+2,1)). In addition add the following to the Crefprint program: 851 RESTORE 920 875 IF P=1 THEN 885 882 GOTO 870. _day = than.

885 PRINT #2:C*