



HUG

HOUSTON USERS' GROUP

JANUARY

1986

PROP. of HUG
SET "A"
c/o R. Lumpkin
Houston Texas
713-469-5089

HUG TIBBS - (713) 475-8909

24-hour BULLETIN BOARD

MEETING SCHEDULE

FIRST SUNDAY OF EVERY MONTH

(2nd Sunday if 1st Sunday

is on a holiday weekend)

AT THE NEXT MEETING

SUNDAY, JANUARY 5, 1986 2:00 P.M.

St. John's School - 2401 Claremont

Ron Albright of San Antonio will be our guest speaker. Ron has served as the TI SIG manager on Comuserve and is widely recognized for his knowledge and interest in the Texas Instruments 99/4A home computer.

IN THIS ISSUE

PRESIDENT'S CORNER LIBRARY ADDENDUM

SUPER SWIFT STRING SEARCH PROGRAM FOR MINI/MEM

MULTI-COLUMN PRINTING, VARIABLE LENGTH LISTING PROGRAMS

1986 HUG OFFICERS

President --- MARK CRUMP unlisted

VP/Membership DON LEWIS 353-5295

VP/Program -- DAVID SHOLMIRE 482-0186

VP/S.I.G. --- ROGERS MILLS .. 930-0810

Exec. Asst. - TOM JAY 850-0222

Secretary - JIM HUMPHREY . 465-6525

Treasurer - JERRY ILLING . 664-7059

Librarian - LARRY PIPKIN . 499-9991

TIBBS/SysOp BILL KNECHT .. 473-5713

Editor --- PHIL PAXTON .. 973-2362

THIS NEWSLETTER IS PUBLISHED MONTHLY BY THE HOUSTON USERS' GROUP. ANY OPINIONS OR ENDORSEMENTS ARE THOSE OF THE AUTHOR, AND MAY NOT NECESSARILY REFLECT THE OFFICIAL OPINION OF 'HUG'. PERMISSION TO REPRINT GRANTED TO OTHER USER GROUPS. SUBSCRIPTION IS FREE WITH MEMBERSHIP.

President's Corner

I'd like to start off the new year by asking you to take a fresh look at our association and your role in it. We begin our new year with a strong organization. Our association has several assets for you. Our best asset is our library. During 1985, under the care of Larry Fipkin, the library grew in quality and quantity. We have excellent titles to fulfill many needs for business, programming, and entertainment. Look over the catalog, and ask other members what their favorite programs in the library are. Order copies and take them home. You'll discover our library is the best deal around.

The monthly HUG meetings will be in a new format. We will have a one hour program followed by an hour to visit with other members. This will give you an opportunity to browse through the merchandise available, order programs from the library, or talk to someone about your personal use of the 99/4A.

Users groups around the country have done much to keep the TI home computer alive. HUG has served well in bringing people together with a common interest in the TI home computer. With this in mind I'd like to ask you to contribute to HUG. Consider writing an article for our monthly newsletter. Everyone has had unique experiences working with this computer and its software. If you've solved a particular problem, write down what you did and send it to the newsletter editor.

Look into working on a program for HUG. Few people have the talent to write entire programs, but everyone can make contributions toward a good program. Most of the best software were written not by individuals but by teams. A good piece of software needs more than programmers, it needs writers, artists, professionals, musicians, and it needs you. With you on a team we could get some of the best software for any home computer.

Use the bulletin board. The bulletin board--operated by Bill Knecht--provides for ongoing dialogue among TI users. Put your questions and comments about our computer on the board. Use it to become more involved with HUG.

Go to SIG meetings. Special Interest Groups can meet special training needs. Remember that joining a SIG is also a commitment to others in the SIG to participate.

Finally, become involved in HUG by making friends with other members. Good friendships can evolve when a common interest is in place.

A SUPER SWIFT STRING SEARCH FOR MINI MEMORY

by Gary R. Curry

When I first decided to try assembly language programming in the Mini Memory, I had several programs which used string arrays to hold data and used what seemed to me "slow as Christmas" BASIC routines to find a match for a given string in the array(s). I resolved to create an assembly routine that I could LINK from BASIC to do the job faster. The SEARCH program listed below is the fruit of my labors.

Using the example program in the MMM manual, I learned how to pass parameters from BASIC to the assembly program and back again. I decided that the minimum parameters that would have to be passed were: 1) the element number of the string array at which the SEARCH is to begin; 2) the element number at which the SEARCH is to end; 3) the string segment to SEARCH for; and 4) the name of the string array in which to SEARCH. You will notice that parameter number THREE says "string segment" rather than just string. I wanted this program to be able to find a string with the least necessary input, that is, the minimum number of characters needed to identify the particular string wanted. For example, to find my last name, "Curry", I could input only "Cu" if there were no other strings beginning with "Cu" in the array. But if there were other names like "Cutter" or "Cuervo" in the array, then the input of "Cur" would be required to pick out "Curry" from all the strings in the array. SEARCH returns a match if the array element matches to the length of the given string segment.

After repeated attempts (and failures), I finally figured out that the DATA given in the MMM manual for the ROM routine which converts an integer into floating point number format is incorrect for the Mini Memory. The DATA given is for the Editor/Assembler, NOT for the MMM. The correct DATA I found in Appendix B (page 72) which shows the memory map for the ROM in the MMM. The correct DATA to follow the BLWP@@>601C (XMLLNK) is >7200 for convert integer to floating point.

To use SEARCH with your TI BASIC program with the MMM, use the following format:

```
CALL LINK("SEARCH",START,STOP,STRING$,ARRAY$())
```

START is a number @Variable previously set equal to the element number at which the SEARCH is to begin. STOP is the number of the element at which the SEARCH is to end (probably the last element). STRING\$ is the string variable or actual string segment for which to SEARCH, and ARRAY\$() is the name given the string array in a DIMension statement in your program in which the SEARCH for STRING\$ is to be done.

Return to the BASIC program gives a change in the value of START. If there is no match for STRING\$ in the ARRAY\$(), then START will be returned as STOP+1; that is, the value of the number variable START will be set equal to the number given above as STOP plus one (START=STOP+1). If there is a match in the array for STRING\$, then START will be returned as the number of the matching element in the ARRAY\$().

Here is an example of how this might be used in a BASIC program:

```

////////////////////////////////////
140 OPTION BASE 1
150 DIM A$(200),C$(200),N$(200)
//////////////////////////////////// Loading names, addresses, and cities into arrays.
////////////////////////////////////
400 REM FIND A NAME
410 INPUT "LAST NAME? ":LN$
420 LET S=1
430 CALL LINK("SEARCH",S,200,LN$,N$())
440 IF S=201 THEN 500
450 PRINT N$(S):A$(S):C$(S)::

```

```

////////////////////////////////////
500 PRINT LN$:" NOT IN DATA.":

```

If you want the SEARCH to continue in the ARRAY\$() to find any other matches, add these lines;

```

460 S=S+1
470 GOTO 430

```

SEARCH may be entered into your Mini Memory Module by using the Line-by-Line Assembler (supplied with the MMM), or by entering and running the TI BASIC listing below. But in either case, you can SAVE the SEARCH routine from the MMM by using the EasyBug SAVE option. Be sure to SAVE from >7000 to >7FFF to get all the necessary information included.

ADDR Code LBL Assembly Mnemonics Remarks

7D00	SB	BSS	64	Set up string segment buffer
7D40	EB	BSS	64	Set up string array element buffer
7D80	0420	GN	BLWP @>6044	Begin routine for getting numeric parameter
7D82	6044			>6044 = NUMREF
7D84	0420		BLWP @>601C	Link to ROM routine
7D86	601C			>601C = XMLLNK
7D88	1200	DATA	>1200	Floating point to integer identifier
7D8A	045B	B	*R11	Return to main program
7D8C	C28B	SE	MOV R11,R10	Save return addr in R10; begin SEARCH
7D8E	04C0		CLR R0	Clear element pointer
7D90	0201		LI R1,1	Set parameter pointer
7D92	0001			
7D94	06A0	BL	@GN	Use subroutine at GN to get number in FAC
7D96	7D80			
7D98	C120	MOV	@834A,R4	Put START number in Register 4
7D9A	834A			
7D9C	0581	INC	R1	Advance parameter pointer
7D9E	06A0	BL	@GN	Get numeric parameter in FAC
7DA0	7D80			
7DA2	C160	MOV	@834A,R5	Put STOP number in Register 5
7DA4	834A			
7DA6	0585	INC	R5	ADD 1 to STOP number
7DA8	0581	INC	R1	Advance parameter pointer
7DAA	0202	LI	R2,SB	Load addr of string buffer in Register 2

7DAC	7D00				
7DAE	0203	LI	R3,>3E00	Load max string length in MSB of R3	
7DB0	3E00				
7DB2	D483	MOVB	R3,*R2	Move MSB of R3 to address in R2	
7DB4	0420	BLWP	@>604C	Get string segment in string buffer	
7DB6	604C			>604C = STRREF	
7DB8	04C6	CLR	R6	Clear string length counter	
7DBA	D192	MOVB	*R2,R6	Get string length byte in MSB of R6	
7DBC	06C6	SWPB	R6	Swap bytes in R6 for length countdown	
7DBE	0581	INC	R1	Advance parameter pointer	
7DC0	C004	MOV	R4,R0	Load element pointer with START	
	number				
7DC2	0202	NX	LI	R2,EB	Load addr of element buffer in R2
7DC4	7D40				
7DC6	D483	MOVB	R3,*R2	Set max string length for buffer	
7DC8	0420	BLWP	@>604C	Get array element in element buffer	
7DCA	604C			>604C = STRREF	
7DCC	0207	LI	R7,1	Set up R7 as index register	
7DCE	0001				
7DD0	99E7	CM	CB	@SB(7),@EB(7)	Compare bytes in string buffer and
	element				buffer as indexed by Register 7
7DD2	7D00				
7DD4	7D40				
7DD6	160E	JNE	ON		Jump if not equal to symbol ON
7DD8	0587	INC	R7		Advance index to next byte
7DDA	8187	C	R7,R6		Check for end of string segment
7DDC	12F9	JLE	CM		If not, compare next byte at CM
7DDE	C800	RE	MOV	R0,@>834A	Put current element # in FAC
7DE0	834A				
7DE2	0420	BLWP	@>601C		Link to ROM routine
7DE4	601C				>601C = XMLLNK
7DE6	7200	DATA	>7200		MMM ref for integer to floating point
	conv.				
7DEB	04C0	CLR	R0		Clear element pointer
7DEA	0201	LI	R1,1		Set parameter pointer for returning
7DEC	0001				element number to BASIC program
7DEE	0420	BLWP	@>6040		Assign value in FAC to parameter 1
	for				BASIC
7DF0	6040				>6040 = NUMASG
7DF2	045A	B	*R10		Return to BASIC program
7DF4	0580	ON	INC	R0	Advance element pointer
7DF6	8140	C	R0,R5		End of SEARCH? Check for STOP element
7DF8	11E4	JLT	NX		If not, get next element (at NX)
7DFA	10F1	JMP	RE		If end, then goto RE

The completes the program listing for use with the Line-by-Line Assembler, but the program name must be added to the REF/DEF table by doing the following:

	AORG	>7FE8
7FE8	TEXT	'SEARCH'
7FEA		4152
7FEC		4348
7FEE	DATA	SE
7FF0	AORG	>701C
701C	DATA	>7DFC
701E		7FE8
	END	

This will put the name of the SEARCH program in place of the LINES program from the Line-by-Line Assembler, and change the First Free Address pointer at >701c to read correctly the address following your SEARCH program.

If using the Line-by-Line Assembler is not your cup of tea, then the following TI BASIC program if loaded and run will put the SEARCH program into your Mini Memory Module, provided the MMM is inserted in the Command Module port on your computer at the time. This program uses the sub program LOAD to "poke" the machine language values into the proper memory addresses. Since the real work of programming in Assembly has already been done and the machine language values are easily available, all that is necessary to convert the SEARCH program into a set of numbers is to convert the hexadecimal numbers into decimal numbers. Then, the numbers are the DATA for the program below to "poke" with CALL LOAD into the memory directly. Remember, you MUST have your Mini Memory Module plugged in for this program to run. CALL LOAD is not available if the MMM is not plugged in, and LIKEWISE there will be no memory to "poke" into at the addresses being used even if you could. Once the program has run, then SEARCH will be in the RAM of the MMM and will remain there for you to use until you do something to erase it. Turning off your console will no lose what is in the MMM RAM, it is battery backed-up. Removing the MMM (with the console off) will not lose it either, but for your protection, I suggest that a tape of the contents of the MMM be made after you have made sure that all is well with SEARCH in the MMM. Use the SAVE option in the Easy Bug. If you have a disk drive, it will be more convenient to simply save the BASIC program and run it whenever you want to re-load SEARCH in your MMM.

```
100 REM *****
110 REM * SEARCH ROUTINE *
120 REM * by Gary Curry *
130 REM *****
140 REM Version 1.2MMM
150 CALL CLEAR
160 CALL INIT
170 PRINT TAB(5);"LOADING THE PROGRAM":TAB(12);"SEARCH":TAB(5);"INTO THE
MINI MEMORY MODULE":
180 LET A=32128
190 READ N
200 IF N>255 THEN 250
210 CALL LOAD(A,N)
220 A=A+1
230 GOTO 190
240 REM UPDATE REF/DEF TABLE
250 CALL LOAD(32760,83,69,65,82,67,72,125,140)
260 CALL LOAD(28700,125,252,127,248)
270 DATA 4,32,96,68,4,32,96,28,18,0,4,91,194,139,4,192,
2,1,0,1,6,160,125,128,193,32,131,74
280 DATA 5,129,6,160,125,128,193,96,131,74,5,133,5,129,2,2,125,0,
2,3,62,0,212,131,4,32,96,76
290 DATA 4,198,209,146,6,198,5,129,192,4,2,2,125,64,212,131,4,32,
96,76,2,7,0,1,153,231,125,0,125,64
300 DATA 22,14,5,135,129,135,18,249,200,0,131,74,4,32,96,28,114,0,4,
192,2,1,0,1,4,32,96,64,4,90,5,128
310 DATA 129,64,17,228,16,241,300
```

```
320 CALL CLEAR
330 END
```

HAPPY COMPUTING! Gary R. Curry

100 REM VARYLIST - Geo. F. Steffan, LA 99ers Computer Group, Oct. 1985

110 REM THIS PROGRAM WILL CONVERT ANY PROGRAM LISTED TO DISK INTO A LISTING OF ANY WIDTH YOU DESIRE

120 REM IT MAY BE A 28 COLUMN LISTING SIMULATING A SCREEN LIST

130 REM IF LISTED TO DISK AND OUTPUT WIDTH IS 80 OR LESS, OUTPUT MAY BE EDITED WITH TI-WRITER

140 REM IF A NUMBERED LINE IS EXACTLY 80, 160, OR 240 BYTES WHEN LISTED, THIS PROGRAM WILL COMBINE IT WITH THE FOLLOWING LINE

150 DATA 3,DSK,WDS,RD

160 CALL CLEAR :: PRINT TAB(11);"VARYLIST"

170 PRINT :: LINPUT "NAME OF INPUT PROGRAM LIST? ":IP\$

180 PRINT :: LINPUT "NAME OF OUTPUT FILE? ":OF\$:: IF OF\$=IP\$ THEN PRINT "INPUT AND OUTPUT NAMES MUST BE DIFFERENT!" :: GOTO 17

0

190 PRINT :: INPUT "WIDTH OF OUTPUT FILE? ":OW :: ODM=OW :: IF OW>79 THEN 220

200 READ N :: FOR I=1 TO N :: READ DN\$:: IF SEG\$(OF\$,1,LEN(DN\$))=DN\$ THEN ODM,I=80

210 NEXT I

220 OPEN #1:IP\$,DISPLAY ,VARIABLE 80,INPUT :: OPEN #2:OF\$,DISPLAY ,VARIABLE ODM,OUTPUT

230 FOR I=1 TO 9999 :: L1\$=""

240 IF EOF(1)THEN I=I+10000 :: GOTO 250 ELSE LINPUT #1:L2\$:: IF LEN(L2\$)=0 THEN GOTO 240 ELSE L1\$=L1\$&L2\$:: IF LEN(L2\$)=80 THEN

N GOTO 240

250 FOR O=1 TO LEN(L1\$)STEP OW :: PRINT #2:SEG\$(L1\$,O,OW):: J=J+1 :: NEXT O :: NEXT I

260 CLOSE #1 :: CLOSE #2 :: PRINT :I-10000;"NUMBERED LINES":J;"OUTPUT LINES" ::

END

100 REM MULTIPRINT - Geo. F. Steffan, LA 99ers Computer Group, Oct 1985

110 REM TI EXTENDED BASIC AND MEMORY EXPANSION

120 REM WILL PRINT MULTIPLE COLUMNS OF ANY TEXT FILE

130 DIM L\$(300):: CALL CLEAR :: PRINT TAB(10);"MULTIPRINT"

140 PRINT :: LINPUT "NAME OF INPUT FILE? ":IF\$:

INPUT "LENGTH OF INPUT LINES? ":LL

150 PRINT :: LINPUT "NAME OF

PRINTER? ":P\$:: INPUT "PRINTER LINE LENGTH? ":PL

160 PRINT "COLUMN SEPARATIONS WILL BE CALCULATED." ::

INPUT "LEFT AND RIGHT MARGIN SIZE? ":M :: INPUT "NUMBER OF COLUMNS? ":C

170 IF (2*(M+C-1)+C*LL)>PL THEN PRINT "WILL NOT FIT" :: GOTO 160

180 OPEN #1:IF\$,INPUT ,DISPLAY ,VARIABLE :: FOR I=1 TO 300 :: IF EOF(1)THEN 210

190 LINPUT #1:L\$(I):: IF ASC

(L\$(I))>127 THEN L\$(I)="" :: GOTO 210 ! DISREGARD TAB SETTINGS

200 NEXT I

210 CLOSE #1 :: S=INT((PL-(C*LL+2*M))/(C-1))+LL :: M=M+1

:: OPEN #2:P\$,DISPLAY ,VARIABLE PL+1,OUTPUT

220 N=INT((I/1)/C):: FOR I=1 TO N :: FOR J=0 TO C-1 :: PRINT #2:TAB(J*S+M);L\$(I+J*N)

::: IF I=1 AND LEN(L\$(1))>LL THEN J=C

230 NEXT J :: NEXT I :: CLOSE #2 :: END

MULTI-COLUMN PRINTING and VARIABLE LENGTH LISTING by George F. Steffan

Reprinted from The Data Bus-Vol3, No.10 Delaware Valley Users Group: Nov 1985

For the past several months I have used two programs to list programs in our newsletter 28 characters wide as they appear on the screen and three columns wide so they do not waste space. I received a request for the method and, at the same time, I saw a program to list programs on a wide printer. So I adapted my program to be more versatile instead of single purpose.

VARYLIST will take a program listing and convert it to whatever line length you desire. There is one bug: if the listed line is an exact multiple of 80 characters in length, the next line will be appended to it. I can think of no simple solution to this and it is such an infrequent occurrence, so it remains in the program. This program works on a program LISTed to disk. If your desired length is 80 or less, the disk file will be opened as VARIABLE 80 so that it may be edited with TI Writer. If you wish to list to a wide printer, the file will be opened with the correct length.

MULTIPRINT will take a text file and output it to the printer in multiple columns so that it may be read in normal newspaper fashion, one column after another. You determine the number of columns, but you must inform the program of the length of the input text and the length of the output device.

This program has no provisions to enable the output text to be edited. Editing must be done before using it

Before using MULTIPRINT you should prepare your text file. You should first use VARYLIST or the Formatter of TI Writer to create a text file of the desired width. Then examine the file and delete any unneeded blank lines. Make sure that the number of lines is an exact multiple of the number of columns you will be using. Insert blank lines to reach this number. You may put these blank lines any place in the text, but they should be placed so as to form pleasing column breaks. If you have used the Text Formatter to print the file, you should use the Replace String command to change all Line Feeds (Control U, Shift J, Control U), Carriage Returns (Control U, Shift M, Control U) and New Page (Control U, Shift L, Control U) to spaces. Because the text is reformatted after these changes, be sure you are not in Word Wrap Mode when you do this. If you make the first line of your text longer than the line length you plan to tell the printer, it will print across the page as on this article. In this case, you must be sure that the first two lines of succeeding columns are blank. Then save the text file or print it to the disk and run MULTIPRINT. The program is designed to accept 300 lines of text, enough for five columns of 60 lines each. If the number is increased too much, the computer will run out of memory.

The programs are listed herewith, each giving an example of itself.

HUG LIBRARY CATALOG ADDENDUM

December 1985

0176 BREAKTHRUE/A**

Fascinating version of "Breakout". Uses joysticks or keyboard. Very challenging game. Written by Clint Pulley of Ontario, Canada. 20 sectors

0177 KITEXB**

Great kite flying game written by Kris & Andrew Kirase of Vienna, Virginia. Fly your kite in all kinds of weather and dodge various obstacles. 69 sectors

0178 OILXB**

Another great game from Kris & Andrew Kirase. Lets you drill your own oil wells. You may even get rich. 28 sectors

0179 SPEED-SKIXB**

Guide your skier on a downhill race avoiding all of the obstacles in your path. A great program by Skip Frusciante. 16 sectors

1072 GRAPH PAPERXB, Printer Rad.**

Print out full sheets of graph paper 8 X 11 columns of 8 grids each. Handy aid when programming graphics. For Epson/Gemini compatible printers only. 6 sectors.

4129 BURGLAR ALARMXB**

Nice program written by R.A. Lumsden. You can wire your own burglar alarm system for home or office using contact switches from a local electronics store. Wiring instructions are included. 34 sectors

4130 XB/40XB**E/A**MM**

Program includes demo to show you various types of scrolling and true 40 column screen. Can be used with XB, Editor Assembler, or Mini Memory cartridges. Comes with complete instructions on how to implement these utilities into your own programs. 230 sectors

4131 CAT_LOADXB**

Nice single drive, 2 pass, disk copier. This program is being distributed by Disabled American Veterans of Cincinnati Ohio. 10 sectors

4132 CATALOGING LIBRARYE/A required**

This program will catalog up to 900 files on 123 disks. Written by Martin Kroll Jr. of Pittsburgh, Pennsylvania and distributed as "Freeware". Written in assembly language it will sort and catalog disks very rapidly. 130 sectors

4133 MASS/XFERXB**

This excellent program, written by Joe Nuvolini allows you to transfer up to 50 files consecutively using an XMODEM type transfer. It also has such features as auto dial, auto answer, and choice of 300/1200 baud rate. 50 sectors

4134 READERXB**

Another fine program from Bill Knecht which allows you to read 40 column DV/80 files and output them to your screen or your printer. 28 sectors

4135 RX80 INITIALIZATIONXB**

This program by John Ford allows you to input several settings for your Epson RX80 or Gemini printer and save them to disk which can be recalled when you need to use them. 39 sectors

4136 DEFINITION TABLEXB**E/A**MM**

This program written by John Clulow will display an object code (DF/80) and give you the program name as well as the rest of the program. Works with XB, Editor Assembler or Mini Memory cartridges. 15 sectors

5226 SYN-SIZERXB**

Excellent program from Scott Lennon. This program allows you to play music on a very realistic sounding music synthesizer. 14 sectors

PHILIP FOXON
EDITOR-HOUSTON USERS GROUP
9122 HAMMERLY
HOUSTON.TX 77080