006 8504

VOLUME NO. 2 ISSUE 20 SEPTEMBER 1985

Bayou 99 Users Group, P.O. Box 921, Lake Charles, La. 70602

BAYOU BYTE

BAYOU 99 USERS GROUP

Price One Dollar Distribution: 150 copies

MEETING NOTICE

The September meeting of the Bayou 99 Users' Group will be at 7:00 P.M. on September 12th at the Nelson Elementary School. Anyone interested in learning to use the capabilities of the 99/4A is invited.

****

CONTENTS

****

EDITORIAL

Freeware

The July issue of Micropendium relates objections voiced to the concept of Freeware. The logic seems to be that Freeware is being developed which will duplicate the output of programs being sold through commercial software companies. The same timeworn argument heard so often in condemning piracy are being used to condem Freeware. The argument is that if the original programmer is not allowed to realize substantial sales to pay for the time and effort to produce programs, he will cease to produce programs. Freeware, it is said, deprives the original author of income he would otherwise receive.

Those forwarding this argument would be in favor of high tariffs, cartels, monopolies, and protectionist legislation to grow fat. While piracy is theft and should be condemned, Freeware is competition and is to be praised. Neither our patent or copyright laws offer protection for ideas; only the product produced from the idea. The free enterprise system is dependent on competition and price is sensitive to supply and demand.

The originator of a program is protected against unauthorized use of his program, but has no more rights than any other business to be protected against competition. Competition brings the normally inflated price of software down to where the users of home computers will buy. Too many try to collect too much in too short a time to have any chance of reaching markets which include the average user. Enough rhetoric arguing pros and cons, the user will choose, and part with his hard earned dollars wherever he can find the program with the utility he requires at the price he can afford. Quality will always have the edge over inferior products and if the value is there, its price will be paid.

Wilkerson

FREEWARE

From

TOPICS - LA 99ers

1. DM1000 Bruce Caron ... POB 460 Route 9, Florence, AL 35620. A marvelous disk-based Disk Manager which rivals CorComp's manager.
2. MASSCOPY Steve Lawless ... 2514 Maple Avenue, Wilmington, Delaware 19808. EXCELLENT disk cloner; features ability to copy to 2 drives at once and uses the Foundation 128K card to copy a disk in ONE PASS!
3. X_DISASM Fred Hawkins ... 1020 North 6th Street, Allentown, PA. 18102. An XB disassembler with many unique features and terrific documentation for those that PAY!
4. SUPER DISK DUPLICATOR Tom Knight ... 7266 Bunion Drive, Jacksonville, FL. 32222. Allows inputting start and stop sector number for copying disks.
5. TK WRITER Tom Knight (See Above) Loads TI WRITER from XB or E/A. No cartridge needed!
6. NEATLIST Danny Michaels ... Route 9, Box 460, Florence, AL. 35630. XB utility to list multi-statement lines to printer or disk for easy reading and references program variables to line number used.
7. SCREENDUMP Danny Michaels (See Above). Screen dump to EPSON compatible printer with double or single size and verticle or horizontal page printout.
8. The DIRECTOR Ron Rutledge ... 1020 3rd Street, Waukee, IA 50363. XB program database that allows cataloging disk-based programs.
9. FAST TERM Paul Charlton ... 1110 Pinehurst Court, Charlottesville, VA 22901. Simply, THE BEST TERMINAL EMULATOR IN THE WORLD!
10. SPRITE BUILDER John Taylor ... 2170 Estaline Drive, Florence, AL. 35630. XB graphics generating program with assembly language routines for speed at crucial places. Includes a full disk of preformed graphics.
11. PILOT 99 Thomas Weithofer ... 1000 Harbury Drive, Cincinnati, OH 45220. An ENTIRE lanugage for the TI that is the simplest programming language known to us (or anyone else!).
12. MASTER CATALOG Mack McCormick ... 215 A Yorktown, Ft. Lee, Virginia 23801. A 100% asembly language disk catalog program that is super fast; handles up to 2000 different disk files.
13. EASYSPRITE Tom Freeman ... 515 Alma Real Dr., Pacific Palisades, CA 90272. An extremely fast XB program with assembly routines to create graphics sprites with easy cursor control saving for program insertion.
14. DISASSEMBLER Marty Kroll ... 218 Kaplan Avenue, Pittsburg, PA 15227. Superfast disassembler, 100% assembly and full featured.
15. TECHIE BBS Monty Schmidt ... 121 N. Blair, Madison, WI. 53703. Freeware BBS system for the 99/4A.
16. COMPACTOR Monty Schmidt (See Above). Assembly language program that takes an uncompressed D/F80 AL program and will compress to about 2/3 the disk space and yield faster load times.
17. UNCOMPACTOR Monty Schmidt (See Above). Opposite of above.
18. PRO 99er BBS Mark Hoogendoorne ... 21 Long Street, Burlington, MA 01803. TI BBS system with TRUE TE2 transfer capabilities.
19. DISK MANAGER Todd Kaplan ... 5802 N. Western Apt. 3S, Chicago, IL. 60659 INCREDIBLE Disk Manager on disk; forget TI's DM2.

20. TOMB OF DEATH John Behnke ... 5755 W. Grace, Chicago, IL. 60634. Details not available.
21. ASSAULT THE CITY John Behnke (See Above).
22. FAST FORTH Tim Curran ... 4153 Four Pole Road, Huntington, WV. 25701. XBasic Loader, fast editor, fast editor locator, 40 column auto-repeat.

MYARC 32/128K MEMORY EXPANSION

STAFF

The first of two of the Myarc 32/128K Memory Expansion Cards in use by B99UG members were Chuck Robertson and Roger Hickerson. Chuck and Roger received their cards at about the same time and Chuck put his in his P-Box with his Myarc Disk Controller while Roger installed his with his CorComp Controller. This review would have appeared in last month's issue if Chuck and Roger both had had the same experiences in using the card. The problems which appeared were first thought to be compatibility problems with operation of the Memory Expansion and the CorComp Disk Controller. The problem turned out to be the card which was exchanged with absolutely no delay or other problems.

The 128K card is 32K of RAM used by TI Extended BASIC and other command modules. The additional 96K can be used as a RAM disk or print spooler. The card can also be expanded to a total of 512 Kbytes although the necessary chips are rather expensive. The TI TMS4464 is not on the market as yet, but the Toshiba TMS4464 chip is available. Replacing the original 16 chips arranged in 4 banks of 4 (16 x 4's) with 16, 64 x 4's, you will obtain 512K for about $10 per chip. The price will come down...sometime.

The user may partition the added 96K into RAM disk and spooler. We ordinarly use 6K for the print spooler and use the remaining 90 like a SSSD disk and copy the disk we want to use to the RAM disk. (This is easily done with the Disk Manager II if you aren't using the Myarc Disk Controller; however, with new Myarc Level III Disk Manager and Disk Controller, direct transfers are f-a-s-t.) When you have the program in the RAM disk, any OLD or SAVE commands are practically instantaneous. For example, TI-Writer is loaded from the RAM disk in 1 second. The disk can be renamed for use with your programs that search for a disk name or made to emulate any of your disk drives for programs written to use a specific drive. For example, you may choose to use the RAM disk as an extra disk drive with the corresponding next disk drive number, then load TI-Writer to it and with a CALL EMDK(1), make it emulate Drive No. 1 as though your TI-Writer disk was in Drive 1. In this case, the real Drive No. 1 is not functional.

Chuck and Roger both recommend Myarc's 128K card very highly, and report no problems using it with any of their programs. We did read an article by Gary Mathews, President of the Atlanta 99/4A Computer Users Group, in their newsletter "Call Newsletter" that some programs such as Masscopy and Quick Copier 2 had memory address conflicts. According to Gary's article, Larry Hughes has changed Quick Copier 2 so that programs would work with the Myarc Memory Card.

An added feature is the connector which allows an external power source to maintain memory in the RAM Disk as long as it is kept energized. If you hook up an outside power source, it should have 500 milliamps current capacity at 7.5 to 9 volts DC.

TRIPLE TECH

The latest from CorComp is a real-time clock for the TI-99/4A. The clock is available as a stand-alone unit which connects to the I/O port on the console in a daisy chain with other stand-alones such as the speech synthesizer and the CorComp 9900 Micro Expansion System or as a "card" for P-Box installation.

The card containing the clock for the P-Box also has a 60K buffer for use with parallel printer and an edge card for mounting the circuit board from your Speech Synthesizer. This feature packed card is being marketed by CorComp under the name "Triple Tech." Unlike the other cards for the P-Box, the Triple Tech is not enclosed in a metal can, but is a bare printed circuit board like the expansion boards in the big machines.

Accessing the clock is like accessing a peripheral; that's the way the TI operating sytem works. Open a file for the clock and INPUT from file three string variables. The variables may then be printed on the screen to display the day of the week (1 through 7), the date as MM/DD/YY and the time in hrs:min:sec. format using a 24 hr. clock.

A simple program opening a file; i.e., OPEN #1: "CLOCK" and obtaining values for three string variables with INPUT #1: A$, B$, C$ can be written which will assign the name of the day of the week for the numbers read from the file. DISPLAY AT may be used to display the time followed by a GOTO back to the INPUT # statement. The clock will then tick off the seconds until you "break" the program.

CorComp's 60K print buffer is for printers connected to the parallel port. A jumper is provided to link the TRIPLE TECH to the RS232's parallel port connection and your printer plugs into the TRIPLE TECH card. Two little such button switches have been provided; the top button will reprint the contents of the buffer and the lower button will clear the buffer, erasing anything that had been sorted.

LAGNIAPPE

- * The 99'ers Association is presently formulating plans to publish a cagalog of manufacturers, producers, distributors, retailers, and service organizations supporting the 99/4A. Look for this to be a "where to find it 'Bible' for TI users."

- * We read that Craig Miller's expected new product will hit the market before the first of December. We heard it called a ROM GROM SIMULATOR or RAM GRAM SIMULATOR (LA 99'ers, Terrie's Corner), and GRAM CRACKER (Super 99 Monthly). The appearance will be that of a long module which is plugged into the cartridge port. Your module will plug into Craig's "Gidget" where it can be "dumped."

- * Another indispensible item for everyone's library is "99 Tips for the 99/4A." Copies may be obtained at $4.50 (First Class mail) from:

Mr. John Hamilton, President
Central Iowa 99/4A UG
Box 3043
Des Moines, IA 50316

BASIC PROGRAMMING

Part 4

R. N. Hickerson

This will be the fourth in a series of articles on BASIC Programming. We have started a BASIC FILES program which can be utilized for a number of different data files by merely changing a DATA statement. For purposes of illustration, an address file was selected and the DATA statement was made up with the prompts for the input of data for an address file. Changing the DATA statement to provide the input prompts for other files and the title is all that is required. So, if you want a Home Inventory File, for example, changing Lines 220 and 230 and "saving" the change to another cassette, would provide you a program for two separate files.

In Part 3 we had written the program from the beginning, Line 100, to the start of the "DATA ENTRY" option. We had OPENed a new file by CS1 by jumping to a routine starting on Line 1080, and then returned to the main program at Line 280 using GOTO statements. If we had entered "N" to answer no to the question, "Will this be the start of a new file?", program execution would have resumed with the following line, Line 270, which contains a GOSUB statement. The GOSUB 1410 sends execution to Line 1410 where it continues until a RETURN statement is encountered.

Line 1410 opens the file. Line 1430 gets the filename, the cassette name and the tape counter values entered when the tape file was originally opened. I will be the first record and the value of I is, therefore, set equal to one. A FOR...NEXT loop starting at F equal to one reads the file and stores the data in the array FILE$. The records are read sequentially until a record is found that has "END" as the first entry in the record or until F is greater than 100. When either END or F equals to 101 occurs, the program jumps to Line 1500 and the file is closed. Line 1480 will also cause the file to be closed if an asterisk is included in the third item in the record. After the file has been closed, the screen is cleared and the RETURN starts program execution again at the statement following the GOSUB. At this time, the value of F is the number of the next record to be added for an existing file and "1" for a new file.

Line 580 prints our warning message from Line 240 and Lines 590, 600, and 610 print the message telling the user how to quit whenever there is no more data to be entered. In our program, entering END at the prompt for "LAST NAME" will termiante the data entry routine. Line 630 next prints the title you have assigned to the file.

Line 650 converts the numerical value of R to a string variable. This step is included since a string variable array is being used for entries from the keyboard. FILE$ being a string variable name means only string variables can be entered. Entering a number will result in a "STRING-NUMBER MISMATCH" error message on your screen. Since the STR$ statement converts numerical values to string values, the program continues with R$, which is the record number, in FILE$(R,1).

In the next few lines, the prompt is printed on the screen and the input from the keyboard is assigned to successive columns in the record row. First

A$ (LAST NAME) followed by B$, C$, D$, E$, and finally F$(BUSINESS NAME) is assigned to FILE$(R,7).

When all the entries for the record have been completed, Line 810 increases the record number, R, by one and execution returns to Line 590 for entry of the next record.

When "END" is entered in Line 680, the program jumps to Line 830 where the remainder of the file is filled with asterisks and a new menu is scrolled onto the screen. The menu in Line 860 allows several choices. Pressing A and <ENTER> records the data, a B starts the EDIT routine, a C returns you to the menu (this step was valuable in checking the program execution), and D allows you to look over the entire file before it is recorded on tape. Lines 890 and 900 allow the choice to be made and Line 910 utilizes the ASC statement which will return the ASC11 value for the first character in the string. In our program, if an "A" was entered, CHOICE would be assigned the value of 65, the number used to denote an A in the ASC11 Tables. Sixty-four is subtracted from the ASC11 value so that CHOICE will be a 1, 2, 3, or 4 depending on what letter A through D was entered. Line 920 assures the entry was an A, B, C, or D and Line 930 causes the program to jump to a different part of the program depending on which key stroke was entered. If an "A" was pressed, CHOICE will equal 1 and the jump will be to Line 960 where we have a GOSUB to Line 940.

The subroutine starting at Line 1160 records the data in the array on the cassette tape. First the tile is OPENed; the title, tape name or number and the tape counter reading is recorded and a second GOTO sends program execution to Line 1160 where the data in the array FILE$ is recorded on tape. At Line 1230 we have included a statement "PRESS ANY KEY TO CONTINUE." This is followed by a GOSUB 1010 where we included a CALL KEY statement. *

At this point in our program, the data entered from the keyboard has been recorded on the tape and should we want to add more records, everything is stored back in your computer. The additions can be added and the new file recorded on tape.

* The expression in paranthesis are called parameters. The S has an initial value of zero and remains equal to zero until a key is pressed. When a key is pressed, the value of S changes so the S is usually called the status parameter. In Line 1020 we keep jumping back to 1010 as long as S equals zero. When S is no longer zero,which occurs when a key is pressed, the program continues to the RETURN from where it begins with the first line after the GOSUB. In our program that will be Line 1050 which directs executions back to 290 and the menu is displayed again on your screen.

With all the data recorded on tape, our record is complete and can be stashed away until needed. When necessary to review the file or get some information out of the file, it is necessary for us to have a method for reading the file. When the program menu comes up on the screen, we must enter "2" to access the Print File option. This selection assigns the value "2" to CHOICE and the ON CHOICE statement (Line 480) sends program execution to Line 1280.

The screen is cleared and a message telling us that we can have our data displayed on the screen or printed by our printer is displayed. The file is opened and the data from the tape is read into the FILE$ array by the subroutine starting at Line 1410. After the data is in memory, the program asks for either a "1" or a "2" to be entered. The program then branches to either Line 1530 or Line 1610.

IF a "1" was entered, a FOR...NEXT loop is used to display each record on the screen. The GOSUB 1010 takes program execution back to the CALL KEY statement which halts the program until a key is pressed. The record will stay on the screen until a key is pressed to permit the record to be read easily.

After a key is pressed, the program resumes operation at Line 1580. The K parameter in the CALL KEY statement takes the ASC11 value of the key that was pressed. If a "Q" was pressed, K will be equal to 81 and the program will return to the menu. When K has any other value, the next record is displayed. The loop will continue until all the records have been displayed or until the "Q" is pressed. When either of these alternatives occurs, the program returns to Line 290 and the menu.

Entering a "2" in Line 1320 would have caused the program to branch to Line 1610 where the screen will display a request for the user to type the name of the print device. A parallel connected printer would be "PIO" if the printer was accessed through the serial port "RS232" followed by any other parameters required, such as .BA=1200. Once the printer has been properly entered, the printer buffer is opened and the file is printed out. When the file has been printed, the program again returns to the menu.

After the menu selection routine is completed and the menu is redisplayed, the program can be terminated by entering a "7" which causes a branch to Line 1350. A simple "GOODNIGHT" is displayed on the screen while a "do-nothing" loop is executed and then the program ends.

```
100 OPTION BASE 1
110 CALL CLEAR
120 REM ********************
130 REM
140 REM         BASIC
150 REM         FILES
160 REM
170 REM ********************
180 REM BY R.N. HICKERSON
190 REM FOR BAYOU 99 USERS G
ROUP
200 REM OCTOBER 13,1983
210 DIM FILE$(70,10),AR$(70)
,LR(70),RR(70),FAR$(70,10)
220 DATA Last Name,First Nam
e&Initial,Street,City/State/
ZIP Code,Home Phone,Business
 Phone
230 TITLE$="ADDRESS BOOK"
240 ST$="DO NOT TYPE ANY COM
MAS WHEN RUNNING THIS PROGRA
M OR YOU WILL CAUSE AN ERROR
."
250 INPUT "WILL THIS BE THE
START OF A NEW FILE,Y/N? ":S
$
260 IF SEG$(S$,1,1)="Y" THEN
 1080
270 GOSUB 1410
280 CALL CLEAR
290 PRINT TAB(14);"MENU"
300 PRINT TAB(8);"1. ADD TO
FILE"
310 PRINT TAB(8);"2. PRINT F
ILE"
320 PRINT TAB(8);"3. DELETE
RECORD"
330 PRINT TAB(8);"4. SEARCH
FILES"
340 PRINT TAB(8);"5. EDIT FI
LES"
350 PRINT TAB(8);"6. SORT FI
LES"
360 PRINT TAB(8);"7. SIGN OF
F"
370 PRINT : : : :
380 PRINT
390 PRINT TAB(4);"ENTER NO.
OF YOUR CHOICE"
```

```
400 INPUT CHOICE
410 IF (CHOICE<1)+(CHOICE>7)
THEN 420 ELSE 480
420 PRINT "YOU MUST SELECT Y
OUR CHOICE"
430 PRINT "BY ENTERING A NUM
BER FROM  "
440 PRINT " 1 TO 7"
450 FOR R=1 TO 800
460 NEXT R
470 GOTO 290
480 ON CHOICE GOTO 510,1280,
2320,1710,1710,2590,1340
490 GOTO 290
500 REM ***** ADD TO FILES *
*****
510 CALL CLEAR
520 R=F
530 GOSUB 550
540 GOTO 570
550 READ A$,B$,C$,D$,E$,F$
560 RETURN
570 PRINT TAB(10);"DATA ENTR
Y "
580 PRINT ST$
590 PRINT TAB(6);"TO EXIT DA
TA ENTRY"
600 PRINT TAB(7);"ENTER 'END
' FOR"
610 PRINT TAB(10);A$
620 PRINT : : : :
630 PRINT "TITLE: ";TITLE$
640 PRINT
650 R$=STR$(R)
660 FILE$(R,1)=R$
670 PRINT A$
680 INPUT FILE$(R,2)
690 IF FILE$(R,2)="END" THEN
 830
700 PRINT B$
710 INPUT FILE$(R,3)
720 PRINT C$
730 INPUT FILE$(R,4)
740 PRINT D$
750 INPUT FILE$(R,5)
760 PRINT E$
770 INPUT FILE$(R,6)
780 PRINT F$
790 INPUT FILE$(R,7)
800 PRINT : : : :
810 R=R+1
820 GOTO 590
830 FOR C=2 TO 7
840 FILE$(R,C)="*"
850 NEXT C
860 PRINT "A. DATA CORRECT-R
ECORD       B. ERROR IN DATA-
EDIT        C. JUST TESTING-G
O TO MENU   D. RECHECK DATA"
870 PRINT
```

```
880 PRINT
890 PRINT "ENTER CHOICE BY T
HE LETTER."
900 INPUT CHOICE$
910 CHOICE=ASC(CHOICE$)-64
920 IF (CHOICE<1)+(CHOICE>4)
THEN 860
930 ON CHOICE GOTO 970,940,2
90,32767
940 OPEN #2:"CS1",INTERNAL,O
UTPUT,FIXED 128
950 RETURN
960 OPEN #2:"CS1",INTERNAL,I
NPUT ,FIXED 128
970 RETURN
980 GOSUB 940
990 PRINT #2:TITLE$,NM$,NM
1000 GOTO 1160
1010 CALL KEY(0,K,S)
1020 IF S=0 THEN 1010
1030 RETURN
1040 IF K=78 THEN 1150
1050 IF K=89 THEN 1070
1060 IF K<>89 THEN 1150
1070 REM ******* NEW FILE
1080 INPUT "ENTER YOUR FILET
APE NAME OR NUMBER ":NM$
1090 INPUT "AT WHAT NUMBER D
O YOU START YOUR TAPE FILE?
 ":NM
1100 PRINT "REWIND TAPE TO C
OUNTER NO. ";NM
1110 OPEN #2:"CS1",SEQUENTIA
L,INTERNAL,OUTPUT,FIXED 128
1120 PRINT #2:TITLE$,NM$,NM
1130 F=1
1140 GOTO 280
1150 REM ***** OLD FILE ****
*
1160 FOR I=1 TO R
1170 PRINT #2:FILE$(I,1),FIL
E$(I,2),FILE$(I,3),FILE$(I,4
),FILE$(I,5),FILE$(I,6),FILE
$(I,7)
1180 IF FILE$(I,3)="*" THEN
1200
1190 NEXT I
1200 CLOSE #2
1210 PRINT : : : :
1220 PRINT "YOUR DATA HAS BE
EN RECORDED ON TAPE ";NM$
1230 PRINT "PRESS ANY KEY TO
 CONTINUE"
1240 GOSUB 1010
1250 GOTO 290
1260 REM
1270 REM***PRINT FILES****
1280 CALL CLEAR
1290 PRINT "ARE FILES TO BE
PRINTED ON:":" "
```

```
1300 PRINT TAB(3);"1. SCREEN
 OR 2. PRINTER? "
1310 GOSUB 1410
1320 INPUT "CHOICE ":CH
1330 ON CH GOTO 1530,1610
1340 CALL CLEAR
1350 PRINT TAB(8);"SAY GOODN
IGHT!"
1360 CALL SCREEN(12)
1370 PRINT : : : : : : : :
1380 FOR DELAY=1 TO 350
1390 NEXT DELAY
1400 STOP
1410 OPEN #2:"CS1",INTERNAL,
INPUT ,FIXED 128
1420 I=0
1430 INPUT #2:TITLE$,NM$,NM
1440 I=1
1450 FOR F=I TO 100
1460 IF FILE$(F,2)="END" THE
N 1500
1470 INPUT #2:FILE$(F,1),FIL
E$(F,2),FILE$(F,3),FILE$(F,4
),FILE$(F,5),FILE$(F,6),FILE
$(F,7)
1480 IF FILE$(F,4)="*" THEN
1500
1490 NEXT F
1500 CLOSE #2
1510 CALL CLEAR
1520 RETURN
1530 PRINT
1540 FOR R=1 TO F
1550 PRINT FILE$(R,1):FILE$(
R,2):FILE$(R,3):FILE$(R,4):F
ILE$(R,5):FILE$(R,6):FILE$(R
,7)
1560 PRINT "PRESS Q TO QUIT,
ANY OTHER   KEY TO CONTINUE.
"
1570 GOSUB 1010
1580 IF K=81 THEN 290
1590 NEXT R
1600 GOTO 290
1610 PRINT "ENTER DEVICE NAM
E (RS232 OR PIO)."
1620 INPUT DEV$
1630 OPEN #1:DEV$
1640 FOR R=1 TO F
1650 PRINT #1:FILE$(R,1):FIL
E$(R,2):FILE$(R,3):FILE$(R,4
):FILE$(R,5):FILE$(R,6):FILE
$(R,7)
1660 PRINT #1:
1670 NEXT R
1680 CLOSE #1
1690 GOTO 290
1700 REM *** EDIT & SEARCH *
**
1710 CALL CLEAR
```

TEXAS INSTRUMENTS POSTAL SYSTEM

CLUB TIPS

MONTREAL

MAXIMEM

LE "MAXI-MEM" (48K ram)
Ce module concu pour le TI-99-4a permet d'augmenter la capacite memoire a 96k Ram, tout en y ajoutant plusieurs fonctions decuplant ses possibilitees.

L'EXECUTION DE TOUS LES MODULES.
(prealablement sauver sur disque)
Il se connecte a l'endroit reserve aux modules.
Il ne necessite aucune alimentation exterieure.
Sa capacite memoire de 48k Ram + 8k Rom controleur, s'ajoute au 16kVDP et 32kRam, portant son total a 96k.

FONTIONNEMENT DU MODULE "MAXI-MEM")
Le Rom de 8k permet d'afficher apres la mire de couleur, les choix suivants:

-1 : TI-BASIC
-2 : MAXIMEM
(permet de lancer l'execution d'un programme binaire qui peut etre en n'importe quel module de votre choix.
-3 : EDITEUR ASSEMBLEUR
(version amelioree de l'editor/assembler.

Pour lancer l'initialisation de MAXIMEM, un Rom de 8k, commutable manuellement a l'aide d'un interrupteur 2 positions place sur le module est utilise.

USAGE PERSONNEL

Aux fins d'usage personnel: l'utilisateur pourra enregistrer tous ses modules sur disquette.
Il pourra les modifies selon ses criteres et crees d'autres versions, de tous ses modules.
Un avantage primordial: Le module MAXIMEM peut rester en place indefiniment, ce qui evite l'usure du port d'entree module de la console.

EXEMPLES
MAXIMEM permet de charger et utiliser plusieurs modules, en meme temps.

-1 Editeur assembleur,basic etendu,Zerozap
-2 TI-writer,Princess&Frog,E/A,connect 4,..
-3 Editeur ass,disk manager,etc....

100% POUR 100% COMPATIBLE.

Plus de 85 modules provenant de divers constructeurs ont ete essayes.Tous fonctionnent a 100%.(LOGO II,EXTENDED BASIC,MULTIPLAN .)

COMMENT VOUS LE PROCURER

Club T.I.P.S Montreal

$199.99 US

Re:"MAXIMEM"
11680 De St-Real
Montreal, Qc H3M 2Y4
Canada

THE "MAXI-MEM" (48K Ram
This module was conceived for the TI-99-4a, and allows to expand the memory of your system to 96K. Also adding many new functions that will improve your system.

EXECUTION OF THE MODULES.
(previously saved on disk)
You connect the MAXIMEM at the same place as the module.(cartridge)
There is no need of external power source.
It increase to 96k the capacity of the console, controler and 32k.

HOW DOES IT WORK?
After the color bars you will have three choices:

1-TI-BASic
2-MAXIMEM
(allows you to run a binary format program previously save on disk. ex: any module

3-EDITEUR/ASSEMBLEUR
(improve version of the editor/assembler.

To start the MAXIMEM you have a 2 positionswitch, on the module.

PERSONNAL USE ONLY

The user can save all his module to disk an recall them when needed.
The user will also be able to modify the contents of any module to his own convenience.
Also the MAXIMEM can stay in place permanently and thus save tear and wear on your console's cartridge connectors.

EXAMPLES

MAXIMEM allows you to load into it a few program at the same time.
1-E/A,EXTENDED BASIC,ZERO-ZAP
2-TI-writer,Frogger,E-A,connect 4
3-E/A,disk manager etc....

100% COMPATABLE

More than 85 modules(cartridges) has been tried.From different software firm and there is no exception.They turned out to be all compatable.Even the big modules works, ex: EXTENDED BASIC LOGO-II, MULTIPLAN and many more.

HOW TO GET IT?
Club T.I.P.S Montreal

$ 199.99

Re:"MAXIMEM"
11680 De St-Real
Montreal, Qc H3M 2Y4
Canada

#23

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 casette, packing and postage, or $3.00 for diskette, PPM) I will send you my descriptive catalog for a dollar, which you can then deduct from your first order.

Several different routines have been published which will extract and save a specified series of lines out of a program, but this one by George Steffen of the L.A. 99ers is certainly the

```
1 !SUBROUTINE EXTRACTOR by G
eorge F. Steffen. SAVE in ME
RGE format. MERGE into any p
rogram (with line # starting
 above 8). RUN to extract
2 !selected lines. Deletes i
tself. Then BE SURE to SAVE
the selected lines in MERGE
format because the remaining
 lines are still in memory!
3 CALL CLEAR :: CALL INIT ::
 INPUT "Line numbers of rout
ine to be saved: First,Last?
      ":L,M :: G=256 :: CAL
L PEEK(-31952,H,I,J,K)
4 C=INT(M/G):: D=M-C*G :: F=
(J-G)*G+K :: FOR E=(H-G)*G+I
 TO F STEP 4 :: CALL PEEK(E,
A,B):: IF A=C AND B=D THEN 6
5 NEXT E :: PRINT :"LINE";M;
"NOT FOUND!" :: STOP !@P-
6 H=INT(E/G):: I=E-(G*H):: H
=H+G :: C=INT(L/G):: D=L-C*G
 :: FOR E=E+4 TO F STEP 4 ::
 CALL PEEK(E,A,B):: IF A=C A
ND B=D THEN 8 !@P-
7 NEXT E :: PRINT :"LINE";L;
"not found!" :: STOP !@P-
8 E=E+3 :: J=INT(E/G):: K=E-
(G*J):: J=J+G :: CALL LOAD(-
31952,H,I,J,K):: STOP !@P-
```

The enhancements to my Menu Loader, published in Tips #22, contained an error. Please change line 413 to read -

```
413 LINPUT #2:M$ :: PRINT M$
 :: IF EOF(2)THEN 416
```

Some folks were interested in the idea of a program that writes a program, so let's write a program that will write a program to list the token codes that you need to use to write a program that will write a program -

```
100 OPEN #1:"DSK1.TOKENLIST"
,OUTPUT,DISPLAY ,VARIABLE 16
3 :: FOR N=129 TO 254 :: L1=
INT(N/256):: L2=N-256*L1
110 PRINT #1:CHR$(L1)&CHR$(L
2)&CHR$(131)&CHR$(N)&CHR$(0)
:: NEXT N
120 PRINT #1:CHR$(255)&CHR$(
```

Key that in and SAVE it just in case, then RUN it. When READY, type NEW, then MERGE DSK1.TOKENLIST. Now LIST it and you will see a list of ASCII codes 129 through 254 and their token meanings. Delete lines 171 through 175, 185, 190, 226 through 231, and 242. Change the definition of 199 to QUOTED STRING, of 200 to UNQUOTED STRING, and 201 to LINE NUMBER, and add line 255 END OF FILE.

You don't need all those exclamation points, so change the program to a DIS/VAR 80 file by LIST "DSK1.TOKENLIST". Then key in this little routine.

```
100 OPEN #1:"DSK1.TOKENLIST"
 :: OPEN #2:"PIO"
110 LINPUT #1:A$ :: PRINT #2
:SEG$(A$,1,4)&SEG$(A$,6,LEN(
A$)):: IF EOF(1)<>1 THEN 110
120 CLOSE #1 :: CLOSE #2 ::
END
```

RUN it, and print out a list of all the token codes. More on this next month - if someone buys a few programs so that I can afford another month.

Now that we've done about all that we can with the Menu Loader, here is another version to use on your finalized library disks of programs. It lacks the features that you will no longer need, but will list your programs by their full names, up to 24 characters long.

```
100 !NAMELOADER by A. Kludge
/M. Gordon/T. Boisseau/J. Pe
terson/etc.
110 CALL CLEAR :: CALL SCREE
N(5):: FOR S=1 TO 14 :: CALL
 COLOR(S,7,16):: NEXT S :: C
ALL VCHAR(1,31,1,96):: CALL
COLOR(0,2,16)
120 OPTION BASE 1 :: DIM PG$
(99),M$(99)
 the programs on the disk in
 the DATA statements, in the
 sequence in which they are
 listed by an ordinary disk
 cataloger program
140 !Then SAVE this program
under the filename LOAD
150 DATA
160 DATA
170 DATA
180 DATA
190 DATA END
200 FOR J=1 TO 99 :: READ M$
(J):: M$(J)=SEG$(M$(J),1,24)
210 IF M$(J)="END" THEN M$(J
)=" " :: GOTO 230
220 NEXT J
230 IMAGE ##
240 DISPLAY AT(1,4):"TIGERCU
B NAMELOADER"
250 D$="DSK1." :: OPEN #1:D$
,INPUT ,RELATIVE,INTERNAL ::
 INPUT #1:P$
260 FOR X=1 TO 99 :: IF X/20
<>INT(X/20)THEN 290
270 DISPLAY AT(24,1):"Type #
 of choice or Enter 0" :: AC
CEPT AT(24,27)VALIDATE(DIGIT
)SIZE(-3):K :: IF K=0 THEN 2
80 :: IF K>0 AND K<NN+1 THEN
 390 ELSE 270
280 X=1
290 I=I+1 :: IF I>127 THEN K
=X :: GOTO 370
300 INPUT #1:P$ :: NN=NN+1
310 IF LEN(P$)=0 THEN 350
320 DISPLAY AT(X+3,2):USING
230:NN :: DISPLAY AT(X+3,5):
M$(NN):: PG$(NN)=P$
330 CALL KEY(0,KK,ST):: IF S
T=0 THEN 340 :: FLAG=1 :: GO
TO 350
340 NEXT X
350 DISPLAY AT(X+4,1):" " ::
 DISPLAY AT(X+5,2):USING 230
:NN+1 :: DISPLAY AT(X+5,6):"
Terminate"
360 DISPLAY AT(X+6,1):"    C
hoice?" :: ACCEPT AT(X+6,16)
SIZE(2)VALIDATE(DIGIT):K ::
IF K<>NN AND K<>NN+1 THEN 38
0
370 IF K=NN+1 THEN CALL CLEA
R :: CLOSE #1 :: END
380 !IF K<1 OR K>99 OR LEN(P
G$(K))=0 THEN 350
390 CLOSE #1
400 CALL INIT :: CALL PEEK(-
31952,A,B):: CALL PEEK(A*256
+B-65534,A,B):: C=A*256+B-65
534 :: A$=B$&PG$(K):: CALL L
OAD(C,LEN(A$))
410 FOR I=1 TO LEN(A$):: CAL
L LOAD(C+I,ASC(SEG$(A$,I,1))
):: NEXT I :: CALL LOAD(C+I,
0)
420 CALL VCHAR(1,3,32,672)::
 CALL SCREEN(8):: FOR S=0 TO
 14 :: CALL COLOR(S,2,1):: N
EXT S :: DISPLAY AT(12,2):"L
OADING ";M$(K)
430 RUN "DSK1.1234567890"
```

Last month I forgot to have anything for the kids, or anything in Basic, so -

```
100 CALL CLEAR
110 REM by Jim Peterson of
Tigercub Software
120 PRINT TAB(1);"****AUTOMA
TIC MOUSE MAZE****": : : :"
  Choose your mouse and":"wa
tch it try to find its way"
130 PRINT "through the maze.
": :"   When one of the mice
 has":"taken 50 extra steps,
 the":"cat gets it!"
140 PRINT : :"Touch any key"
150 CALL KEY(0,K,ST)
160 IF ST<1 THEN 150
170 CALL CLEAR
180 CALL CHAR(120,"0078FEFFF
E78")
190 CALL CHAR(121,"1038387C7
C7C7C38")
200 CALL CHAR(122,"387C7C7C7
C383810")
210 CALL CHAR(123,"001E7FFF7
F1E")
220 CALL CHAR(128,"001E61816
11E")
230 CALL CHAR(129,"384444444
4242410")
240 CALL CHAR(130,"102828444
4444438")
250 CALL CHAR(131,"007886818
678")
260 CALL SCREEN(5)
270 T1=610
280 T2=610
290 CALL CHAR(136,"FFFFFFFFF
FFFFFFF")
300 CALL COLOR(14,16,16)
310 CALL COLOR(13,2,16)
320 CALL COLOR(12,2,16)
330 R=10
340 GOSUB 1460
350 R1=10
360 C=2
370 C1=2
380 CALL HCHAR(R,C,136,2)
390 C=C+1
400 M=120
410 M2=128
420 RANDOMIZE
430 A=(INT(2*RND)+1)*2
440 B=INT(10*RND)+1
450 ON B GOSUB 470,470,470,4
70,510,510,550,550,590,590
460 GOTO 420
470 IF C+A>30 THEN 630
480 CALL HCHAR(R,C,136,A)
490 C=C+A
500 RETURN
510 IF R+A>20 THEN 540
520 CALL VCHAR(R,C,136,A)
530 R=R+A
540 RETURN
550 IF R-A<2 THEN 580
560 CALL VCHAR(R-A+1,C,136,A
)
570 R=R-A
580 RETURN
590 IF C-A<3 THEN 620
600 CALL HCHAR(R,C-A+1,136,A
)
610 C=C-A
620 RETURN
630 CALL HCHAR(R,C,136)
640 C=C+1
650 IF C<31 THEN 630
660 R2=R
670 C2=C
680 CALL HCHAR(R1,C1,M)
690 CALL HCHAR(R2,C2,M2)
700 Y=Y+1+(Y=2)*2
710 IF Y=2 THEN 1020
720 CALL HCHAR(R1,C1,136)
730 ON M-119 GOTO 800,900,74
0,850
740 IF C1=31 THEN 950
750 CALL GCHAR(R1,C1+1,G)
760 IF G=32 THEN 850
770 C1=C1+1
780 M=120
790 GOTO 950
800 CALL GCHAR(R1-1,C1,G)
810 IF G=32 THEN 740
820 R1=R1-1
830 M=121
840 GOTO 950
850 CALL GCHAR(R1+1,C1,G)
860 IF G=32 THEN 900
870 R1=R1+1
880 M=122
890 GOTO 950
900 CALL GCHAR(R1,C1-1,G)
910 IF G=32 THEN 800
920 C1=C1-1
930 M=123
940 GOTO 950
950 CALL HCHAR(R1,C1,M)
960 IF (C1=31)*(C2=2)THEN 13
20
970 IF C1<31 THEN 700
980 T2=T2-10
990 CALL SOUND(50,T2,5)
1000 IF T2=110 THEN 1340
1010 GOTO 700
1020 CALL HCHAR(R2,C2,136)
1030 ON M2-127 GOTO 1040,120
0,1090,1150
1040 CALL GCHAR(R2+1,C2,G)
1050 IF G=32 THEN 1090
1060 R2=R2+1
1070 M2=129
1080 GOTO 1250
1090 IF C2=2 THEN 1250
1100 CALL GCHAR(R2,C2-1,G)
1110 IF G=32 THEN 1150
1120 C2=C2-1
1130 M2=128
1140 GOTO 1250
1150 CALL GCHAR(R2-1,C2,G)
1160 IF G=32 THEN 1200
1170 R2=R2-1
1180 M2=130
1190 GOTO 1250
1200 CALL GCHAR(R2,C2+1,G)
1210 IF G=32 THEN 1040
1220 C2=C2+1
1230 M2=131
1240 GOTO 1250
1250 CALL HCHAR(R2,C2,M2)
1260 IF (C2=2)*(C1=31)THEN 1
320
1270 IF C2>2 THEN 700
1280 T1=T1-10
1290 CALL SOUND(50,T1,5)
1300 IF T1=110 THEN 1370
1310 GOTO 700
1320 CALL HCHAR(1,1,32,768)
1330 GOTO 330
1340 GOSUB 1460
1350 PRINT "THE CAT GOT THE
WHITE MOUSE": :
1360 GOTO 1390
1370 GOSUB 1460
1380 PRINT "THE CAT GOT THE
BLACK MOUSE": :
1390 PRINT "TO PLAY AGAIN, T
OUCH ANY KEY"
1400 CALL KEY(0,K,ST)
1410 IF ST<1 THEN 1400
1420 T1=610
1430 T2=610
1440 CALL HCHAR(1,1,32,768)
1450 GOTO 330
1460 CALL HCHAR(23,1,32,32)
1470 PRINT CHR$(120);(610-T1
)/10;TAB(20);CHR$(128);(610-
T2)/10
1480 RETURN
```

Did you know that ACCEPT AT(1,0) will accept a full line of 28 characters? Did you know that ACCEPT AT (R,0)SIZE(-28) and Enter will accept everything on row R? And did you know that ACCEPT N$ will accept a string of 255 characters?

Need a filler, so -

```
100 !MUSICAL BARGRAPH by Jim
 Peterson
110 CALL CLEAR :: CALL SCREE
N(5):: FOR J=2 TO 14 :: X=J-
(J>4):: CALL COLOR(J,X,X)::
NEXT J
120 DIM N$(13),N(13):: M$="(
08@HPX'hpx"&CHR$(128)&CHR$(1
36):: FOR J=1 TO 13 :: N$(J)
=SEG$(M$,J,1):: DISPLAY AT(J
+6,1)SIZE(1):N$(J):: NEXT J
130 X=110 :: FOR J=1 TO 13 :
: N(J)=X*1.059463094^(J-1)::
 NEXT J
140 A=INT(13*RND+1):: B=INT(
25*RND+1):: DISPLAY AT(A+6,2
)SIZE(28):RPT$(N$(A),B):: CA
LL SOUND(B*40,N(A),0,N(A)*2+
4,0,N(A)*4+6,0)
150 DISPLAY AT(A+6,2):"" ::
GOTO 140
```

MEMORY FULL

Jim Peterson

We wish to give credit to the PUGET SOUND 99'ERS who printed this All Purpose Handy Dandy Reference sheet in their APRIL 85 Newsletter.

COLOR CODES

COLOR	VALUE
TRANSPARENT	1
BLACK	2
MED. GREEN	3
LT. GREEN	4
DARK BLUE	5
LT. BLUE	6
DK. RED	7
CYAN	8
MED. RED	9
LT. RED	1Ø
DK. YELLOW	11
LT. YELLOW	12
DK. GREEN	13
MAGENTA	14
GRAY	15
WHITE	16

PATTERN IDENTIFIER CONVERSION TABLE

Ø	Ø	Ø	Ø	Ø
Ø	Ø	Ø	1	1
Ø	Ø	1	Ø	2
Ø	Ø	1	1	3
Ø	1	Ø	Ø	4
Ø	1	Ø	1	5
Ø	1	1	Ø	6
Ø	1	1	1	7
1	Ø	Ø	Ø	8
1	Ø	Ø	1	9
1	Ø	1	Ø	A
1	Ø	1	1	B
1	1	Ø	Ø	C
1	1	Ø	1	D
1	1	1	Ø	E
1	1	1	1	F

ERROR CODES

FIRST	COMMAND OR STATEMENT
Ø	OPEN
1	CLOSE
2	INPUT
3	PRINT
4	RESTORE
5	OLD
6	SAVE
7	DELETE
9	EOF
SECOND	**TYPE OF ERROR**
Ø	DRIVE NOT FOUND
1	DEVICE or FILE WRITE PROTECTED
2	BAD OPEN ATTRIBUTE
3	ILLEGAL OPERATION
4	OUT OF SPACE
5	ATTEMPT TO READ PAST END OF FILE
6	DEVICE ERROR or HARDWARE ERROR
7	FILE ERROR - File or disk does not exist

ASCII CODES

CODE		CODE		CODE		CODE		CODE		CODE		CODE		CODE	
3Ø		48	Ø	65	A	82	R	99	c	116	t	133	GO	15Ø	NEXT
31		49	1	66	B	83	S	1ØØ	d	117	u	134	GOTO	151	READ
32		5Ø	2	67	C	84	T	1Ø1	e	118	v	135	GOSUB	152	STOP
33	!	51	3	68	D	85	U	1Ø2	f	119	w	134	RETURN	151	DELETE
34	"	52	4	69	E	86	V	1Ø3	g	12Ø	x	137	DEF	154	REM
35	#	53	5	7Ø	F	87	W	1Ø4	h	121	y	138	DIM	155	ON
36	$	54	6	71	G	88	X	1Ø5	i	122	z	139	END	156	PRINT
37	%	55	7	72	H	89	Y	1Ø6	j	123	{	14Ø	FOR	157	CALL
38	&	56	8	73	I	9Ø	Z	1Ø7	k	124	|	141	LET	158	OPTION
39	'	57	9	74	J	91	[	1Ø8	l	125	}	142	BREAK	159	OPEN
4Ø	(	58	:	75	K	92	\	1Ø9	m	126	~	143	UNBREAK	16Ø	CLOSE
41	)	59	;	76	L	93	]	11Ø	n	127	DEL	144	TRACE	161	SUB
42	*	6Ø	<	77	M	94	^	111	o	176		145	UNTRACE	162	DISPLA
43	+	61	=	78	N	95		112	p	129	ELSE	146	INPUT	163	IMAGE
44	,	62	>	79	O	96	`	113	q	13Ø	::	147	DATA	164	ACCEPT
45	-	63	?	8Ø	P	97	a	114	r	131	!	148	RESTORE	165	ERROR
46	.	64	@	81	Q	98	b	115	s	132	IF	149	RANDOMIZE		
47	/														

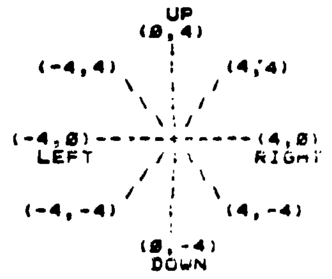
CHARACTER SETS

SET	ASCII CODES
Ø	3Ø-31
1	32-39
2	4Ø-47
3	48-55
4	56-63
5	64-71
6	72-79
7	8Ø-87
8	88-95
9	96-1Ø3
1Ø	1Ø4-111
11	112-119
12	12Ø-127
13	128-131
14	136-143
15	144-151\BASIC
16	152-159/ONLY

CALL KEY VALUE OF KEYSTROKES

CODE	KEYSTROKE
1	FCTN 7
2	FCTN 4
3	FCTN 1
4	FCTN 2
5	FCTN =
6	FCTN 8
7	FCTN 3
8	FCTN S
9	FCTN D
1Ø	FCTN X
11	FCTN E
12	FCTN 6
13	ENTER
14	FCTN 5
15	FCTN 9

JOYSTICK RETURN VALUES



CONTROL CODES

ASC	PRESS	COMMENTS
1	CTRL A	START OF HEADING
2	CTRL B	START OF TEXT
3	CTRL C	END OF TEXT
4	CTRL D	END OF TRANS.
5	CTRL E	INQUIRY
6	CTRL F	ACKNOWLEDGE
7	CTRL G	BELL
8	CTRL H	BACKSPACE
9	CTRL I	HOFIZ. TAB
1Ø	SHFT/ENTR	LINE FEED
11	CTRL K	VEPTICAL TAB
12	CTRL L	FORM FEED
13	ENTER	CARRIAGE RETURN
14	CTRL N	SHIFT OUT
15	CTRL O	SHIFT IN
16	CTRL P	DATA LINK ESCAPE
17	CTRL Q	DEVICE CONTROL 1
18	CTRL R	DEVICE CONTROL 2
19	CTRL S	DEVICE CONTROL 3
2Ø	CTRL T	DEVICE CONTROL 4
21	CTRL U	NEG. ACKNOWLEDGE
22	CTRL V	SYNCHRONOUS IDLE
23	CTRL W	END OF TRANSMIS.
24	CTRL X	CANCEL
25	CTRL Y	END OF MEDIUM
26	CTRL Z	SUBSTITUTE
27	CTRL .	ESCAPE
28	CTRL M	FILE SEPARATOR
29	CTRL J	GROUP SEPARATOR
3Ø	CTRL 8	RECORD SEPARATOR
31	CTRL 9	UNIT SEPARATOF
127	SHIFT V	DELETE CHARACTER

EXTENDED BASIC STATEMENTS

ACCEPT	DISTANCE	KEY	ON WARNING	SCREEN
CHAR	END	LET	OPEN	SOUND
CHARPAT	ERR	LINK	OPTION BASE	SPSET
CHARSET	FOR	LINPUT	PATTERN	SPRITE
CLEAR	GCHAR	LOAD	PEEK	STOP
CLOSE	GOSUB	LOCATE	POSITION	SUB
COINC	GOTO	MAGNIFY	PRINT	
COLOR	HCHAR	MOTION	RANDOMIZE	
DATA	IF THEN	NEXT	READ	
DEF	IMAGE	ON BREAK	REM	
DELSPRITE	INIT	ON ERROR	RESTORE	
DIM	INPUT	ON GOSUB	RETURN	
DISPLAY	JOYST	ON GOTO	SAY	

CALL LOADS

ADDRESS	PARAMETERS	DESCRIPTION
-3274Ø	POKE Ø-255	RATE of FLASH of CURSOR
-31786	POKE 192	DISABLE SPRITE action
-318Ø4	PEEK X,Y	Returns to Title Screen
	POKE X,Y	
-318Ø6	POKE 16	DISABLE QUIT KEY(Fctn =)
	POKE 32	DISABLES SOUND
	POKE 64	DISABLES AUTOSPRITE MOTION
-3186Ø	POKE 4	Goes to Console BASIC after 'NEW' is typed
-3186Ø	POKE Ø	Runs DSK1.LOAD
-31878	POKE Ø-26	Highest # SPRITE in motion
-31888	POKE 63,255	Disables Disks,NEW fre.mem
-31931	POKE Ø/128	Unprotects/Protects progrm
-31962	POKE ˉ2	Returns to TITLE SCREEN
-31962	POKE 55	Runs DSK1.LOAD
-31952	PEEK A,B,C,D	Recovers program with LOAD
-28672	PEEK A	96=Speech Syn. Ø=No Speec

BAYOU 99 USERS GROUP

OFFICERS

PRESIDENT	MARK WILSON
VICE PRESIDENT	ROGER HICKERSON
SECRETARY	BRUCE WYMAN
TREASURER	BOB NORDAN, JR.

COMMITTEE CHAIRMEN

NEWSLETTER EDITOR	DOUG HARGETT
PROGRAM	RICHARD MITCHELL
LIBRARY	SONNY HOFFPAUIR
RECRUITING	PETE STILL
EQUIPMENT	TIM HILL
EDUCATION	JOHN SINGLETON
TIBBS - SYSOP	ROGER HICKERSON

****

ARTICLES AND OTHER MATERIAL APPEARING IN THE BAYOU BYTE MAY BE COPIED BY OTHER USER GROUPS OFFERING RECIPROCAL PRIVILEGES IF BOTH BAYOU BYTE AND BYLINE CREDIT IS GIVEN.

****

ADVERTISEMENT RATE SCHEDULE

FULL PAGE	$14.00
HALF PAGE	8.00
QUARTER PAGE	6.00
EIGHTH PAGE	4.00
MINIMUM 4 LINES	3.00

Members are entitled to three lines free each month for notices or ads for personally owned hardware or software.

"NOTICE"

BAYOU 99 USERS GROUP
P.O. BOX 921
LAKE CHARLES. LA. 70602

MEETING 2nd. THURSDAY EACH MONTH AT 7:00 P.M. A. A. NELSON ELEMENTARY, 1001 COUNTRY CLUB ROAD, WEST OF LAKE STREET ABOUT 2 BLOCKS ON SOUTH SIDE OF COUNTRY CLUB ROAD.

Learn More About Your Texas Instrument Computer

Join A Users Group Now

1985 MEETING DATES

JAN	FEB	MAR	APRIL
10	14	14	11
MAY	**JUNE**	**JULY**	**AUG**
9	13	11	8
SEPT	**OCT**	**NOV**	**DEC**
12	10	14	12

The opinions and views expressed in articles published are those of the authors and do not necessarily reflect the views of the officers or members of the Bayou 99 Users Group or the Editor of the BAYOU BYTE.

Mention of a company or product in the BAYOU BYTE does not constitute an endorsement by either the BAYOU BYTE or the Bayou 99 Users Group.

****

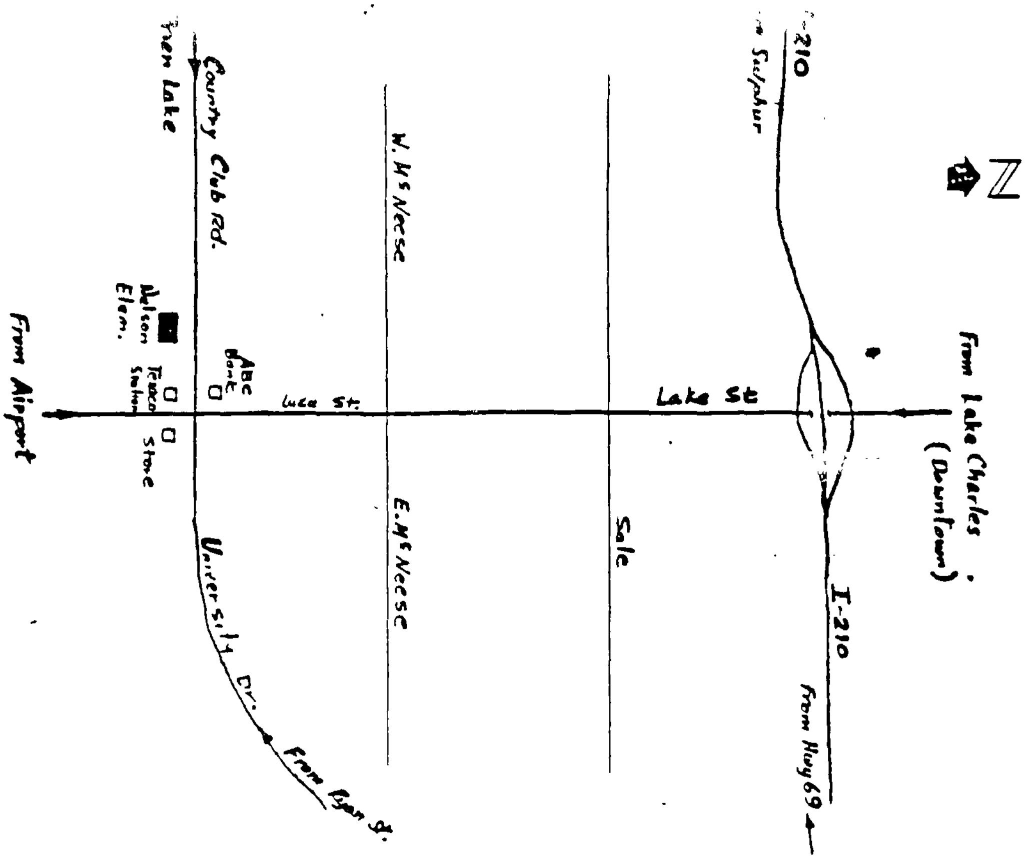
MINUTES

The members of the Bayou 99 Users Group met August 8th at the Nelson School with 13 members and 2 guests present. Roger Hickerson brought everyone up-to-date on the application to the I.R.S. for a letter of recognition of our Group as a non-profit organization. The letter is required for reducing the postage cost for mailing of the Newsletter.

Our President, Mark Wilson, then gave a preview of the Navarone Paint and Print Program which had just been received by the Group. This program will be demonstrated by Mark at our next meeting which will allow the time needed to show off the extensive capabilities of this program sent to us by Navarone. The Paint and Print preview was followed by Richard Mitchell's presentation of Sprite Builder. Sprite Builder is available as "Freeware" on disk or cassette by sending $5 to the author - John Taylor.

Following the presentations, the SIG groups represented convened into separate discussion groups.

BAYOU 99 USERS GROUP
MEETING LOCATION
A. A. NELSON ELEMENTARY
1001 COUNTRY CLUB ROAD



BAYOU 99 USERS' GROUP
POST OFFICE BOX 921
LAKE CHARLES, LA 70602

EDMONTON USERS GROUP
PO BOX 11983
EDMONTON-ALBERTA
CANADA T5J-3L1,