
UPTIME

MUSIC CITY 99ers USERS GROUP QUARTERLY NEWSLETTER

THIRD QUARTER 1987

TI-SDOM FROM THE PRESIDENT:

Hello again! Happy News! We had a GREAT BIG SUMMER and everyone seems to have been really busy. Our UpTime editor, Tommy Cushing has been out traveling on the road and the vast majority of fun with our "adopted" Computer has been had at the Saturday Software Swaps and Tuesday night meetings. There are now two Myarc GENEVE 9640 computer owners in the club. We have a lot of bargains in store for sale and trade, too! I am working on a shell for a new database program-more on that later.

```
M  M  EEÉÉ  EEEE  TTTT  III  N  N  GGG  DDD  AA  TTTT  EEEE  !
MM MM  E    E    T    I  NN  N  G  G  D  D  A  A  T  E  !
M M M  EEE  EEE  T    I  N  N  N  G  D  D  AAAA  T  EEE  !
M  M  E    E    T    I  N  NN  G  GG  D  D  A  A  T  E  !
M  M  EEEE  EEEE  T    III  N  N  GGG  DDD  A  A  T  EEEE  *
```

IS AGAIN SCHEDULED FOR TUESDAY OCT. 20th 7-9 p.m. in ROOM #312 at the WEST END METHODIST CHURCH. DON'T FORGET IT! WE WILL MISS YOU, TOO!

For further information, write our Post Office box or call me at home, (615)889-5852 or at work (615)242-9634. I'll be listening for you! Make it a point to get involved in this great club experience! JOIN, LEAD, ENJOY! If you would like to contribute some Time or some articles for UpTime, contact me or any of our officers and know that your input will be well received. Thanks to all our new 1987 members!! DON'T MISS YOUR CHANCE TO MEET WITH YOUR GROUP WHO UNDERSTANDS YOUR TI NEEDS AND RESPONDS TO THEM WITH ANSWERS, TUTORIALS, & TECHNICAL HELP. Our club program library is really growing and to date, we have well over 1000 great programs. These are all available to you for copying at either of our regular meetings. We manage to get together a full DISK-OF-THE-MONTH for each meeting & we have a good cassette library. Our 300/1200 BAUD TIBBS is not running and is in need of a SysOp. We will make arrangements with anyone who will dedicate some of their Time for this service. It's fun and educational too! Don't forget your \$15.00 annual membership. Your dues include membership, library, and this newsletter. The Music City 99ers Users Group offers complete support for the TI-99/4A HOME COMPUTER. Tell us how we can serve you better.

A T T E N T I O N ! ! A T T E N T I O N ! ! A T T E N T I O N ! !
We have recently gotten some new members and we welcome each and every one to our club. Please let us know your needs and we will surely help all that we can. This group is worth every thing that you put in to it and more. Sometimes if you feel that you are not getting enough out of this group or any group, try donating your Time and talents to making it better. You as a group have purchased a sometimes idling dual-disk system with double-sided/double density drives. You should determine how to make it useful to you, to benefit you, to help you, and also to nourish your desires to be productive with your own TI-99/4A Computer.
MEETING OCT. 20th, 7 p.m. BE THERE !

"LADLE RAT ROTTEN HUT"

by: H. L. Chase

Wants pawn term, dare worsted ladle gull hoe lift wetter murder inner ladle cordage, honor itch offer lodge, dock, florist. Disk ladle gull orphan worry putty ladle rat cluck wetter ladle rat hut, an fur disk raisin pimple colder Ladle Rat Rotten Hut.

Wan moaning, Ladle Rat Rotten Hut's murder colder inset, "Ladle Rat Rotten Hut, heresy ladle basking winsome burden barter an shirker cockles. Tick disk ladle basking tutor cordage offer groin-murder hoe lifts honor udder site offer florist. Shaker lake! Dun stopper laundry wrote! Dun stopper peck floors! Dun daily-dolly inner florist, an yonder nor sorghum-stenches, dun stopper torque wet strainers!"

"Hoe-cake, murder," resplendent Ladle Rat Rotten Hut, an tickle ladle basking an stuttered oft.

Honor wrote tutor cordage offer groin-murder, Ladle Rat Rotten Hut mitten anomalous woof.

"Wail, wail, wail!" set disk wicket woof, "Evanescent Ladle Rat Rotten Hut! Wares are putty ladle gull goring wizard ladle basking?"

"Armor goring tumor groin-murder's," reprisal ladle gull. "Grammer's seeking bet. Armor ticking arson burden barter an shirker cockles."

"O hoe! Heifer gnats woke," setter wicket woof, butter taught tomb shelf, "Oil tickle shirt court tutor cordage offer groin-murder. Oil ketchup wetter letter, an den -- O bore!"

Soda wicket woof tucker shirt court, an whinny retched a cordage offer groin-murder, picked inner windrow, an sore debtor pore oil worming worse lion inner bet. Inner flesh, disk abdominal woof lipped honor bet, paunched honor pore oil worming, an garbled erupt. Den disk ratchet ammonol pot honor groin-murder's nut cup an gnat-gun, any curdled ope inner bet.

Inner ladle wile, Ladle Rat Rotten Hut a raft attar cordage, an ranker dough ball. "Comb ink, sweat hard," setter wicket woof, disgracing is verse.

Ladle Rat Rotten Hut entity bet rum, an stud buyer groin-murder's bet. "O Grammar!" crater ladle gull historically, "Water bag icer gut! A nervous sausage bag ice!"

"Battered lucky chew whiff, sweat hard," setter bloat-Thursday woof, wetter wicket small honors phase.

"O Grammar, water bag noise! A nervous sore suture anomalous prognosis!"

"Battered small your whiff, doling," whiskered dole woof, ants mouse worse waddling.

"O Grammar, water bag mouser gut! A nervous sore suture bag mouse!"

Daze worry on-forgery-nut ladle gull's lest worts. Oil offer sodden, caking offer carvers an sprinkling otter bet, disk hoard-hoarded woof lipped own pore Ladle Rat Rotten Hut an garbled erupt.

MURAL: Yonder nor sorghum stenches shut ladle gulls stopper torque wet strainers.

Submitted by Tommy Cushing

October 4, 1987

Dateline-Nashville. It must have been good because it lasted for almost seven hours. Yesterday, the first Saturday in October, Roy Moore, John Treadway, Bob Sylvester (from KY), Rhea Bullington, Jr., and Richard Washington met at the residence of Dick Lasher, co-hosted by Bob Teague. Refreshments were offered but not heavily indulged because as one said it, the refreshments could not be served on a silicon chip. Some swapping of equipment was accomplished, some needed equipment modification and help was given by some more knowledgeable members and some extensive programming tutorial was given to those requesting assistance in their personal programming. Some new programs were also swapped.

One project was to modify a program which accepted data and displayed it on the screen and then optionally printed a hard copy on paper. The request was to modify the program to save the data to a disk for future printing and/or reference. We accomplished this by converting a program line from 200 PRINT #1:"PIO" To 200 PRINT #1:"DSK1.DATA", APPEND. Then we loaded whatever portion was desired into TI-WRITER and printed it. This answer was offered as a quick solution to a more complex problem which would require revisions in the program using DISPLAY and ACCEPT to input the program operators desires to name a data file, name a disk drive and whether an immediate printout was desired. Anyway the solution solved his question regarding his program and a new insight to TI-WRITER.

Another project was to modify an external disk drive to operate as disk drive two. Another member offered some dip switches that he had to accomplish this task.

Another project was to modify a Centronics male plug to use as a PIO cable. We disassembled my plug to follow the pin configuration. Mission accomplished.

I had a problem with a menu driven program where I use TIXB, MYARC XBII, MULTIPLAN, and FUNNELWEB for E/A, DM1000 and TI-WRITER. Most of my user written programs are written for MYARC but I could not load Funnelweb with anything but TI-XB. Bob Teague showed me a program to optionally load it with the MYARC XBII also. Rhea also provided me with a freeware program to print MULTIPLAN spreadsheets sideways for printouts needing wider than 8 1/2" paper plus another that he wrote to setup my Panasonic printer for various modes prior to printing. Terrific program Rhea.

Bob Teague also discovered a new option to the MYARC disk manager program that I feel is neat. At the main menu screen option, select the "setup" and change the Output Device: to read "DSK1.DATA" or some other disk device & filename, then "Exit" to the main screen again. Now catalogue a disk and use FCTN P to print the catalogue to disk. If you continue with more disk catalogues, each print appends the file. You can now load that D/V 80 file into TI-WRITER to add comments, move lines, etc before printing a catalogue listing. Fantastic Bob! (continued)

Some of us, also discussed what the differences were between a TTL-RGB, Analog-RGB and Digital-RGB and the pixel configuration needed for minimum high resolution to use 80 column display with the GENEVE 9640. I have been using an amber monochrome composite mode monitor and complaining about squinting, knowing that it can be much better. The character-pitch is also of some consideration. We have obtained information that Magnavox makes an RGB Monitor 80 #8CM515 with a 14" color screen that has 640 dots horizontal and 240 lines vertical resolution. It also has a "green" optional display switch in front with audio input and video inputs for RGB analog as well as composite video (composite is used by the TI-99/4A). There is also a cable ready to connect the GENEVE for RGB control with audio and the cost is around \$330. My experience so far has been to buy a Magnavox 14" with HI-RES and cable to fit the TI-99/4A. To my dismay, I found that the TI cable was only for composite and the RGB was only for TTL input used by IBM and others. Luckily, I was dealing with a "no questions asked" return policy store which allowed me to return it when I found that it would do no more than a regular TV or other monitor using TI or the GENEVE. I am currently "gun shy" but still interested.

The preceding article was intended to make a point. When I came into this club several years ago, I was like someone else who described themselves as "I understand maybe 10% of what I read about computers as well as the TI system". I have come a long way since I too only understood a fraction but now understand maybe 75% or more and am continuing to grow. I want to do my share to return to the club membership what it has given to me in gaining an understanding of computers. This is only accomplished by participation with others of similar interests and encouraging others to get more deeply involved in programming, equipment upgrading opportunities at a low cost by buying used equipment and/or swapping and learning the secrets of what is inside their machines and how to fix-it-yourself. Our "meet" on Saturday demonstrated a productive session or it would not have lasted as long as it did. Some apologized to my wife, but she (bless her) understood as she could see that each of us were as dedicated as I to this new learning experience in our lives. Some have said that they too get up at 3 & 4 in the morning thinking on a program that they are doing and spending several hours before work to accomplish what woke them up. These "work sessions" can help us learners because they are more of a one-on-one experience of helping one another.

Thank you to everyone who participated and Bob Teague in particular for being such a willing and helpful President. Words cannot say enough for the leadership and guidance that he has provided now for several years. Signing off: Dick Lasher

PROGRAMMING TIPS FROM LASHER

ROUNDING OFF NUMERIC INPUT TO THE NEAREST INTEGER OR DECIMAL

The TI-99/4A keyboard will handle 14 digits with integrity. For instance, if you let $A=12345678901234$ then A will return the same digits without rounding off. The A numeric value, however will be expressed in "scientific notation" as $1.23457E13$ which merely means that the decimal must be moved 13 places to the right. Although the sixth digit in the scientific notation expression has been rounded off to a "7", it still remains a "6" in the machine memory.

In layman's language, another example is in order to explain scientific notation. This example would be to let $B=123456789$ which could be expressed in scientific notation as $1.2E8$ or $1.23E8$ or, if the fourth digit is used, then the "4" should be rounded to "5" and shown as $1.235E8$. The "8" following the "E" indicates the position of the decimal moved to the right. If the "8" had been "-8", then it would mean that the decimal should be moved to the left and the B numeric value was equal to 0.000000123456789 instead. The TI-99/4A, however expresses all digits up to 10 digits in numeric format and only goes to scientific notation with 11 digits or more.

Now, to go back to our original value of $A=12345678901234$ which will print to the screen as $1.23457E13$. To retain our fundamental approach to this explanation, we will describe the "INT" (INTEger) function and the "^" (powers of) function and then show how they are used to redisplay all of the 14 digits of "A" numeric in string format and that the 14 digits have not been rounded or changed. An integer is any whole number and does not have any fractional values. The expression 12.34 contains the fractional value ".34" and the integer of that expression is "12". So $INT(12.34)$ will be returned as "12". Next, the power function can be expressed as 10^3 which means the same as $10 \times 10 \times 10$ and equals 1,000. Another would be 2^4 which is the same expression as $2 \times 2 \times 2 \times 2$ and equals 16. Still another could be 1^3 which is $1 \times 1 \times 1$ and equals 1. For my program use explanation, we will use 10^n (where "n" represents the number of times a digit is multiplied by itself) to explore how to move the decimal point left and right in the "A" numeric expression above. We saw how 10^3 means $10 \times 10 \times 10$ or 1 with 3 zeros after it. Therefore, 10^8 means 1 with 8 zeros after it. Now 10^1 means simply 10 or 10^0 simply means 1 with no zeros after it. Exploring a little further, 10^{-1} means the decimal is moved one place to the left of the 1 and equals "0.1". Now, if I multiply 1234 by 10^{-1} this would equal " 1234×0.1 " or " 123.4 ".

If we multiply the numeric value of "A" (which is 12345678901234) by 10^{-8} , we will produce the value 123456.78901234 and if we further invoke the INT function, we will now end up with the value of "123456". If we further multiply this by 10^8 , we will now have "12345600000000". Next if we subtract "12345600000000" from "12345678901234" we will now have the value of 78901234 as a remainder. The following mathematical expressions will produce the same results as follows:(See Next Page)

```

Let A=12345678901234
    (if we print A, it will be displayed as 1.23457E13)
Let X=INT(A*10^-8)
    (this sets X=123456)
Let Y=A-X*10^8
    (this multiplies X to equal 12345600000000 and subtracts it
    from A=12345678901234 and leaves the remainder of
    78901234)

```

In EXTENDED BASIC and in command mode, this can be written as:

```
A=12345678901234 :: X=INT(A*10^-8) :: Y=A-X*10^8 :: PRINT X,Y
```

The print display on the screen will be as follows:

```
"123456"      "78901234"
```

Next try this in command mode:

```
A=123456789012345 :: X=INT(A*10^-9) :: Y=A-X*10^9 :: PRINT X,Y
```

The print display on the screen will be as follows:

```
"123456"      "789012300"
```

As you can now see, using 15 digits will change the value of A by rounding off.

In the first example, where we are using only 14 digits, if we further combine the STRING equivalents of X and Y into one display, we would then always display on the screen using PRINT STR\$(X)&STR\$(Y) as follows:

```
"12345678901234"
```

The next step now is to illustrate how to round off a numeric input value to the nearest integer. If we add 10^{13} (or 10,000,000,000,000) to a numeric input value of 12.34, the machine will round this off to 10,000,000,000,012 and then by subtracting 10^{13} , we will be left with 12 as our answer or 12.34 rounded off to 12. The value 12.56 would be returned as 13.

Last of all 1E13 is the same as 10^{13} and 1E-13 is the same as 10^{-13} . Therefore the following program can be written as follows:

```

100 DEF RD=A+1E13-1E13
110 INPUT A
120 PRINT RD
130 GOTO 110

```

Another technique would be as follows:

```

100 INPUT A
110 PRINT A+10^13-10^13
120 GOTO 100

```

To carry this thought one step further, we can further round off other numeric inputs to the nearest decimal point using 1 or 2 or more decimal places. First multiply your numeric input by 10^n (where "n" represents the number of decimal places desired) before adding 1E13 and subtracting 1E-13. After this adding and subtracting, then multiply "A" times 10^{-n} . If two decimal places are desired and 12.34456 were input, the answer would be 12.34 and an input of 12.34567 would result in 12.35 as your answer.

EDITOR'S NOTE: YOUR COMPUTER IS NOT JUST A TOY!

PROGRAMMING TIPS FROM LASHER

BYTE CRUNCHING FOR DATA FILES

Speed and size are two of my most important considerations. This article deals with size and is based upon my needs which may be similar to others. I am a salesman with 65 accounts, each with 1 to 350 records per account. I have a program that keeps track of orders as they are written and sent to the factory. When they are acknowledged, I update that record with the acknowledgement number, date of acknowledgement and anticipated shipping date. Similarly, when it is invoiced, I again update with the invoice number and date shipped. My object was to keep a data disk of all these records on one DSDD disk, so space was a consideration in my file construction. Each dealer has a separate file indentified by the account number as assigned by the factory. This takes 65 sectors before the first record is written to any dealers file. I decided upon INTERNAL, FIXED 64 format for fast relative, random access to records of my choice. I also can reduce all of the data in a record to numeric values including ALPHA expressions such as REPL (replacement), SOLD, WOG (consolidate this order to ship with other goods), PO# (purchase order), etc. These ALPHA expressions (15 of them) are common to each and every dealer's records so I represent them with a string of 5 digits and retain in memory the 15 ALPHA expressions in a single element array with 15 elements (DIM A\$(14)). Zero to 14 equals 15 elements. The first digit of the 5 digit string will represent the first three elements of the ALPHA array as follows: If the first digit is 1, 3 or 5, I convert it to a numeric and subtract 1 and then divide by 2. This produces a 0, 1 or 2 and tells my program which element of the ALPHA array is to be printed (A\$(0) or A\$(1) or A\$(2)). The second segment of the 5 digit string would represent the next set of the ALPHA array (A\$(3) or A\$(4) or A\$(5)). I do this with a double loop to select the segment of the 5 digit string and which set of three elements of the array are to be printed. The reason that I use 1,3 and 5 is that if I need to use more than one Alpha expression in a set of the array, I can express each digit of the string as 4, 6, 8 or 9 also. A 4 is the combination of 1 and 3 and now will print elements A\$(0) plus A\$(1), etc. A 6 would be a combination of 1 and 5, an 8 would be a combination of 3 and 5 and a 9 would be a combination of all three elements of the array (A\$(0) plus A\$(1) plus A\$(2)). This way 5 digits represent 15 ALPHA expressions.

The rest of the numeric values in a dealers record consist of invoice number and date, acknowledgement number and date, date ordered, dollar amount of order, date of expected shipping, product number and quantities. In total, I use 98 digits for each dealer record. I separate these 98 digits into 7 different variables of 14 digits each. In another article, I have demonstrated how the TI-99/4A retains 14 digits in memory without changing them by rounding off and how to convert them from a scientific notation to a string of 14 digits. Each numeric variable takes up 9 BYTES when printed to a data file on disk. It takes 9 BYTES whether the numeric variable equals 0 or 1 or 123456789 or 12345678901234. Therefore 7 numeric variables take up 63 BYTES of space per record but I am actually storing 98 BYTES of data. Since I use a FIXED 64 file, I get 4 records per sector on the disk.

There are any number of programs that this theory may be applied to use such as I am also working on an accounting program using 2 numeric variables and 2 string variables. I store the dollar amount, the date of the entry and the account number assignment plus I still have room for additional digits, into the 2 numeric variables. This allows 28 digits in 18 BYTES leaving room for 46 BYTES to be used in my string variables when I specify a file format of FIXED 64 and this allows 4 journal entries per sector of the disk.

CONTINUE TO CRUNCH THOSE BYTES. Signing off: Dick Lasher/366-7355

=====
If you want to use your word processor to help you EDIT or break down program segments, this little program will really work for you.

```
100 DISPLAY AT(1,3)ERASE ALL:"TEXT TO PROGRAM CONVERTER"
110 DISPLAY AT(3,1):"Create or Edit your EX-BASICprogram with
TI-Writer(PF todisk)or use the E/A(SF) and CONVERT to a RUNable
program"
120 DISPLAY AT(8,1):"First LIST program to a diskwithin following
guidelines-": : " (1)Each program line has to be limited to 80
characters"
130 DISPLAY AT(14,1):" (2)The first character/s in each line must
begin with a line number."
140 DISPLAY AT(18,1):"TEXT FILE - DSK" :: ACCEPT AT(18,16)SIZE(12)
BEEP:A$ :: IF A$="" THEN 360
150 DISPLAY AT(20,1):"PRGM NAME - DSK" :: ACCEPT AT(20,16)SIZE(12)
BEEP:B$ :: IF B$="" THEN 360
160 OPEN #1:"DSK"&A$, INPUT
170 OPEN #2:"DSK"&B$,OUTPUT ,VARIABLE 163
180 ON ERROR 340
190 LINPUT #1:L$
200 S=POS(L$," ",1)
210 N=VAL(SEG$(L$,1,S))
220 A=INT(N/256)
230 B=N-A*256
240 DISPLAY AT(24,5):"OPERATION IN PROGRESS"
250 PRINT #2:CHR$(A)&CHR$(B)&SEG$(L$,S,80)&CHR$(0)
260 IF EOF(1)=0 THEN 190
270 PRINT #2:CHR$(255)&CHR$(255)
280 CLOSE #2
290 CLOSE #1
300 IF I THEN 360 :: DISPLAY AT(3,1):"Now type NEW and press ENTER":
:"Then t ype MERGE DSK";B$
310 DISPLAY AT(6,1):"Type the first line number &press FCTN X & FCTN
1 until you have gone through all ofthe line numbers, and
reSAVEyour program before RUNning."
320 DISPLAY AT(11,1): : : : : : : : : :
330 GOTO 360
340 DISPLAY AT(24,1)ERASE ALL:"TEXT FILE BAD - TAKE A LOOK" :: CALL
SOUND(150,110,0) :: FOR I=1 TO 500 :: NEXT I :: DISPLAY AT(24,1):""
350 RETURN 270
360 DISPLAY AT(24,5):" Want to EXIT (N)" :: ACCEPT AT(24,20) SIZE(-1)
VALIDATE("YnN")DEEP : X$ :: DISPLAY AT(24,1):"" :: IF X$="N"
OR X$="n" THEN 110
370 CALL CLEAR :: END ! from Bob Teague/889-5852
```