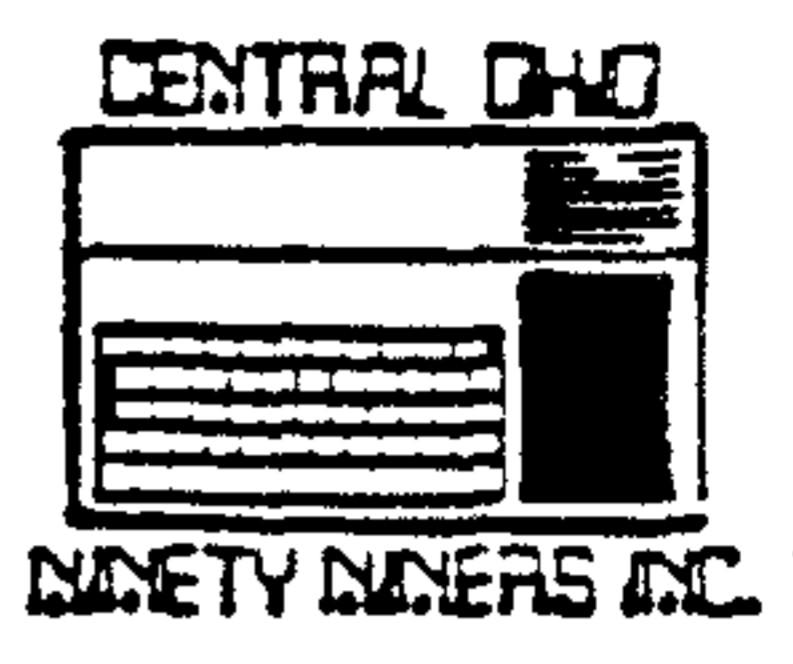


5/1/93

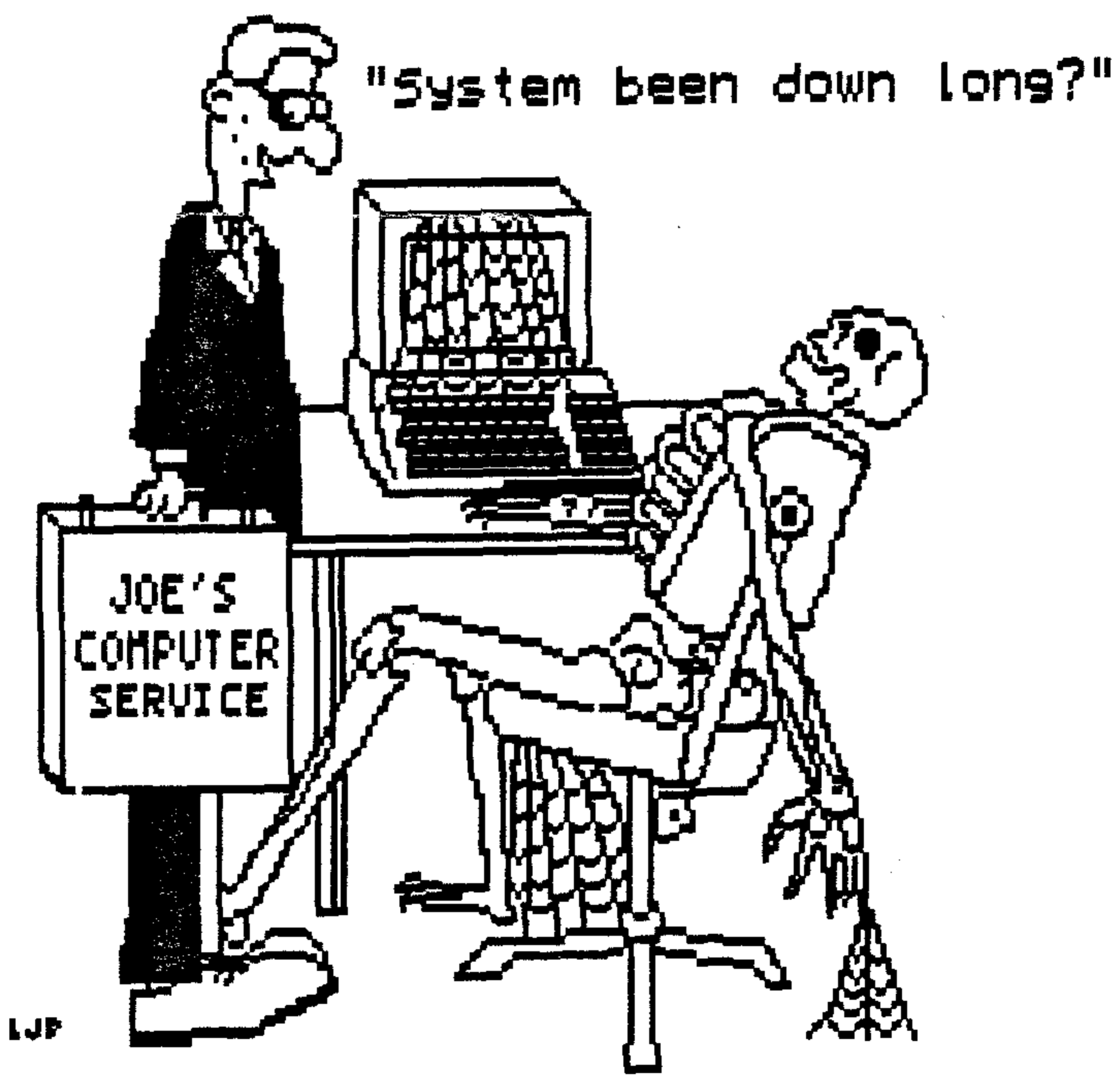
Texas Instrument 99/4A and Myarc 9640 Computers

Spirit of 99



THE OFFICIAL NEWSLETTER OF THE CENTRAL OHIO NINETY-NINERS INC.

PUBLISHED IN COLUMBUS OHIO



COPYRIGHT © 1985
Central Ohio Ninety-
Miners Incorporated
(C.O.N.N.I.), Colum-
bus Ohio 43212, USA.
All rights reserved.
Spirit of 99 is pub-
lished for Central
Ohio Ninety Miners
Inc. by C.O.N.N.I.
members and is the
official newsletter
of C.O.N.N.I. User
Group.

Editorial, address
is:
181 HEISCHMAN AVE
NORTHINGTON, OH 43085
Subscription rate
(USA) \$20.00 /1 year
\$25.00 /1 year out-
side continental U.S.
Third class postage
paid at Columbus, OH
CHANGE OF ADDRESS:

Send both OLD and
NEW address to add-
ress listed above.
We assume no respon-
sibility for manusc-
ripts, programs (tape
or disk) not accomp-
anied by return pos-
tage. Letters to the
Editor are property
of Spirit of 99. If
published, we reserve
the right to edit at
our discretion.

OPINIONS EXPRESSED
HEREIN ARE THE AUTH-
ORS AND ARE BASED ON
VALID DOCUMENTABLE
RESEARCH, THEY DO NOT
NECESSARILY REFLECT
THE OPINIONS OF THE
PUBLISHER.

We will not know-
ingly publish copy-
right material with-

out the permission
of the author and
credit due.

All programs pub-
lished herein are of
public domain unless
otherwise noted.

Other non-profit
user groups may use
material from this
newsletter only if
source and credit is
given.

Central Ohio Nine-
ty Miners Inc. is a
non-profit organiza-
tion comprised of ME
MBERS who own or use
the TI99/4A computer
and it's related pro-
ducts and have paid
a yearly membership
fee of \$20 and whose
main objective is
the exchange of Edu-

cational and Scient-
ific information for
the purpose of comp-
uter literacy.

C.O.N.N.I. meetings
are held the 3rd sat-
-urday of each month
at C h e m i c a l
Abstract, 2540
Oientangy River Road
Columbus, OH. Meet-
ing time is 8:30 AM
til 2:30PM, Meetings
are open to the pub-
lic. Membership dues
(\$20.00) are payable
yearly to C.O.N.N.I.
and cover the immedi-
ate family of the
member. Please send
check to our member-
ship registerar and
join C.O.N.N.I.

Please address it to:
Everett Wade

179 Erie Road
Columbus, OH 43214

INDEX

	PAGE
ABOUT THE DOM.....	04
ANOTHER 96 CHARACTERS...	04
CHECKSUM.....	12
CONNI MINUTES.....	02
LIMA CONFERENCE.....	05
MANAGING YOUR MONEY-3...	05
MORE ON LINE NOISE.....	10
MORE TIPS and Trix.....	06
PROGRAM OF THE MONTH....	09
TI-101.....	14
TIPS FROM THE TIGERCUB..	07
TI-SORT.....	03
USING "PRINT USING".....	06

C . O . N . N . I . MINUTES

MINUTES OF C.O.N.N.I. MEETING Saturday 17 April 1993

Prior to the meeting, we watched Brad Snyder's music programs and a French billiards game.

Former member Curt Borders sent a boxfull of TI hardware, software and publications to be given away to members.

President John Parkens opened the meeting and announced the recent decease of long-time member John Nihof.

Everett Wade gave the treasurer's report. Transfer of duties to the new treasurer has been postponed until the bank can find the signature card for the account.

Chuck Grimes reported the contents of the Disk of the Month.

A motion was passed to buy a Rave keyboard from Everett Wade, to be installed in the group's portable computer which is being assembled.

Innumerable other subjects were discussed, most of them relating to computers.

After the meeting, Irwin Hott demonstrated his Franklin Dictionary, an electronic device for the blind which gives a spoken definition for any of 80,000 words, with many cross-references and many other features. He also demonstrated the Type and Speak, a word processor for the blind which speaks the name of each key as it is typed, and also has many additional features.

Respectfully submitted
Co-Secretary

Jim Peterson

President John Parkins opened the meeting at approximately 8:20. A very loosely-organized meeting that did not always stick to the subject of TI computers but that was nonetheless a lot of fun.

General discussion included pipe organs and theatre organs, selling unwanted systems and other topics. During the Question and Answer session a user reported problems that could stem from either a bad drive or RS232.

We wondered what the featured speakers would present at the Lima fair, and what exhibitors would be present, but so far the club has received no information.

Jim Peterson raised a question concerning the status of the Compiled Basic out of Chicago that we heard about some time back. No one had any answers to that one, either. A discussion followed regarding the various merits of Super Extended Basic and Megatronics Extended Basic. Jim P. lamented that Chris Bobbitt's reported release of the new (Rich) Extended Basic still uses a 28-col. screen instead of forty columns.

John Parkins demonstrated his Selectronic Word Speller, followed by a demo of the Panasonic KX-P2123 24-pin color printer, used with a 4A. Some have said such an arrangement wouldn't work, but Harley Ryan, the owner, showed that it does. We were unable to determine whether the machine was using a 9-pin emulation or true 24-pin operation, but the quality of the printed results suggested the latter.

The meeting concluded at approximately 10:30 p.m.

Respectfully submitted,

Dick Beery
Co-Secretary

C.O.N.N.I. MEETING Wednesday, April 28, 1993.

TI-SORT reviewed by Jim Peterson

This program has been around since 1989, but hasn't received much attention. I've had it for a couple of years, but only recently had a need to use it. I was MUCH impressed - but also much disappointed by one serious flaw.

Sorting is a job for a computer, but it is not an easy job for a computer. Let me explain.

You can quite easily and quickly sort a dozen playing cards into sequence in your hand, by pulling them out and shoving them back in where they belong. But suppose those cards were spread out face down on the table. You are allowed to hold one card in your hand. You can peek at the others, find the lowest one, swap it for the one in your hand, pick up the first card in the row and replace it with the lowest card, and then start peeking for the next one. That is roughly how the computer performs a sort. It takes a lot of moves. For twice as many cards, it doesn't take twice as many, but several times as many. I once wrote a program called SORTWATCHER (it is in Tips From The Tigercub #33, with a correction in #34) which lets you watch different kinds of sorts actually taking place on the screen.

Now, the TI-99/4A is handicapped as a sorting machine, not only by its slowness but by the fact that in XBasic it can only store strings in that puny 16k of memory in the console - and not even all of that. That would limit it to about 150 80-character records.

J. Peter Hoddie wrote an assembly program called Sort Experiment which very greatly speeds up the operation and allows use of the 24k of expansion memory, or about 300 80-character records. His program has one very dangerous bug - when it has read in as many records as memory can hold, it starts sorting - without warning you that it does not have the complete file!

But suppose you need to sort a file that is larger than 24k? Dennis Faherty came up with the ingenious solution. His TI-SORT uses a single FIXED file as a batch file. Records are read into memory, sorted, and then written to the scratch file in consecutive blocks, keeping track of where each block

begins. Then the first record is pulled from each block, sorted, the lowest record written to the sorted file and another record pulled from the block it was taken from, etc.

The size of the file you can sort is limited only by the capacity of the disk that holds the scratch file. Since it is a FIXED file, it usually becomes considerably larger than the file being sorted - a 40-character record in a D/V80 file takes only 40 bytes, but it takes 80 bytes in a D/F80 file. A DS/DD disk could handle 4300 records, a ram-disk even more, and a hard drive would be limited only by the 32767 record limit of the program itself.

The program is very well written. The main menu screen offers you a choice of sorting a TI-BASE file, a delimited file, a fixed record file, or a file created by Basic. The program was actually written because of the poor sorting capability of TI-BASE, and will sort any size record that TI-BASE will support, according to the documentation. Other types of files are limited to 255 bytes, but can be fixed or variable, display or internal.

A delimited file is one which has fields of variable length, separated by a character such as an asterisk. A fixed field file is one in tabular format. I'm not sure what a "file created by Basic" is, because almost any kind of file can be created by Basic, but they seem to be handled in the same way as a fixed field sort.

The main menu also offers the option of sorting in descending order (from the highest down), or cataloging the disk (thanks for that!), or quitting.

If you elected a delimited sort, you are asked for the delimiting character. Then you are required to set up a structure file, unless you already have one, even if you just want to do a simple sort on the first character. If you asked for a TI-BASE sort, the program expects to find a structure file created by TI-BASE.

You are taken step by step through the creation of the structure file, for as many fields as you want, up to 17. For a fixed field file you must know the length of each field, and you are also asked for this information for other types of files. You are also asked

whether the field is character or numeric and, if numeric, how many decimal places you want to sort to.

The documentation tells you, although the screen prompts do not, that you exit this part with FCTN 8. You have the option of saving the structure file for future use.

Then you select the field or fields that you want to sort on, in the sequence that you want - the program can perform nested sorts up to 8 levels.

Then the sorting begins. You are shown on screen just what is going on, and it is fast! Reading from disk is very fast, sorting is extremely fast, writing to the scratch file a bit slower, the final merge and writing somewhat slower yet but, just for instance, 500 80-character records can be sorted in 90 seconds on a Horizon RAM disk, in 8 minutes and 11 seconds on a SS/SD drive.

But now for the bad news. I was unable to perform a numeric sort on any field of my fixed field files. I sent them to Dennis Faherty and he soon identified the problem, which no one had told him about in the years this program has been on the market. It was written for TI-BASE, which right-justifies its numerical fields. The fixed field files created by tabbing in Funnelweb, or by most any other means, are left-justified, and those in a delimited file are not justified at all!

The documentation also states that TI-SORT will not correctly sort display data which have strings with imbedded blanks. This could be a serious deficiency, but I have sorted such files with no problems.

I do not have TI-BASE, so did not test it with TI-BASE files. With delimited files, it seems to expect fields of fixed lengths, which defeats the whole purpose of a delimited file.

But, if you need to sort ASCII fields in large files, this program will really do the job. The documentation is thorough and well written. You can get it from Inscobot Inc., P.O. Box 291610, Pt. Orange FL 32029, for \$14.95 plus \$2.50 S&H.

END

ABOUT THE D.O.M.

Remember: "^" following a filename means you will have to use Archiver 3.03 or 3.03g to unpack the file to another disk. The same is true for most files with the designation "I/F 128". Files marked with "D/V 80" can be read/printed with any of the wordprocessors: TI-Wirter, Funnelweb, etc.

We received a request from Ray Francies, the programmer for JUMBLE, which appeared in an earlier D.O.M., that we print the following patch that will prevent the same numbers from coming up each time the game is played:

70 insert RANDOMIZE

579 delete RANDOMIZE statement there.

7610 change the 400 to 100 (the line to return to).

That's it! Apply the patch and enjoy the game even more than previously. Thanks for the correction, Ray, and also thanks for your many contributions to our D.O.M. and to the T.I. world in general.

Side A this month (April 1993) has one file for Geneve only. Its name is FADE^ and it allows you to fade in/out screens.

Also on A is MICROARC^, for either 4A or Geneve. Programs typed in from Micropendium. These included: Assembly19, Crazyeigt, Disklabel, Screenduap, Time It, Search, Sidebar 19 and Text.

Also: MIDI/SNF^, for which you can use either 4A or Geneve and for which you will also need the program Midimaster99 and some sort of Midi device. Included are several huan tunes, the popular Cielito Lindo programmed by Jim Peterson, and both Bumbleboogie and Dizzyfingers. The file is worth getting if only for the latter, which has been broken into two files that must be played separately and that are truly wonderful.

Next: DRB^, a 1-2 player game by Jon Dyer and Joe Delekto. You need joysticks to use it. Load using an EA-5 loader. See the docs in the archived file. Players shoot at each other.

Last: READ--THIS. A complete listing of all files on this issue with comments and hints.

Side B:

DRAWMASTR^, for both 4A and Geneve. Read the docs first. Load it from an EA-5 loader, using the filename "MURRY". It's a drawing program.

FED^, for the 9640 only. A sector editor from John Johnson. Edits both hard and floppy drives.

JOBIM^, Brazilian music from programmer Harold Timmons. I like it so much that I used it for a demo at a recent C.D.N.M.I. meeting. Hope you enjoy it too. Either 4A or Geneve. Loads from Extended Basic. The music was written by

the renowned composer kAntonio Carlos Jobim. Included are: Batidinha, Dialog, Lamento, Trise, Mojave and Wave. At least two of those have been widely known and loved, according to HT.

MOUSEV30^, New mouse driver. 9640 only.

VED^, Video editor from John Johnson. Doc file. 9640 only.

XFIND^, Geneve only. Not sure what it does, and can't find out for myself, as I don't have a 9640.

Now, you say! Not for me--it's almost all for the 9640 only. NOT TRUE! To quote from the READ--THIS: "you will see that the file space used for 9640 only files is small compared to the TI files, and when you unpack the Micropendium files, the Jobim music file, the Drawing Master program and the Midi Snf music files, you will see that most of what the disk has is for both machines."

Try to make it to the Lima fair on May 15. Several new things will reportedly be available. Contact Dave Szippel at (513)498-9713 evenings for more information.

Note: the terms "9640" and "Geneve" are used interchangeably.

END

ANOTHER 96 CHARACTERS?
Charlotte U6, Jim Folz
Delaware Valley U6
Great Lakes 1/93

Many Gemini printer users don't realize that they can access another 96 characters in their printers. The 99/4A uses seven data bits to send characters to be printed. The additional printer characters in the Gemini printer (and some others) become active when the eighth bit is turned on. Gemini engineers have provided an escape sequence that turns the eighth bit ON and OFF.

In TIM FORMATTER, .TL60:27,62 will make the eighth bit go ON when a < is found in the text. If you also put in a .TL62:27,61 then > will turn the eighth

bit OFF. Check your printer manual to make sure your codes are not different.

Many users will find it easier to use the CTRL-U sequence to access these codes, when working from the EDITOR, so as not to have to use the FORMATTER. In that case, the sequence would be: CTRL-U, FCTN-R, CTRL-U, >, to turn the bit ON and CTRL-U, FCTN-R, CTRL-U, =, to turn it OFF

The eighth bit simply adds 128 to the ASCII value of the character you type. Consult your printer manual for the characters you want to print, subtract 128, and use the ASCII character of the resulting number.

For a complete printout of the resul-

ting number, enter and RUN the following Basic Program.

```
100 REM PRINT CHARS 128+ !PR
INTS 2 COL.
110 OPEN #1:"PID"
120 PRINT #1:CHR$(15)!COND
130 FOR X=32 TO 80 !2-COL
140 PRINT #1:" ";X;" ";X+128
;" ";CHR$(X+128);" ";X+4
B;" ";X+48+128;" ";CHR$(X+48
+128)
150 NEXT X
160 CLOSE #1::END
```

END

Author's Note: This series of articles dealing with personal finance based solely upon my own opinions and my own experiences. I am not a trained financial advisor nor am I an investment counsellor. You are cautioned to always seek the advice of a finance professional before making any decision or taking any action that would involve what to you is a significant amount of money.

CREATING BUDGET:

Creating a budget is not as difficult as it might seem, it does take some commitment on your part. The quickest approach you can use is to go back through your checkbook registers and/or creditcard statements for at least 2-3 years to determine what you spend money on. If you don't use a checkbook or creditcards at all, then you have a lot of guess-work ahead of you. If you use them quite a bit, you will still end up with less than 100% accuracy in what you come up with because there is still bound to be a lot of unaccounted for cash that was sent along the way. But that's OK for now. The idea is to determine "what" you spend your money on and to then get some idea of "how much" you spend in each area. Once you have accomplished that, you can average the total by the number of months it took to accumulate the amount that you have totaled up for each type of expense. That average is the monthly amount you will use as a starting point for budget creation. Once you arrive at the amounts for each category you can then adjust them as the year progresses and you get a more accurate picture of your spending.

The software that you use to manage a budget is up to you. There are several programs that are or were available, among them;

-Bill Hara's Fast-Tran program.

-Irv Crowley's SYSTEM III program.

-John Taylor's Checkbook and budget Management program.

-Bill Gaskill's CHECK+ program.

TI's Household Budget Management module

-tie Babcock's Home Accounting and Control System

There were several other budget management programs reviewed in the April 1984 issue of Micropendium that might still be around, but they are not listed in any of the major mail-order catalogs. So you will just have to check them out. There are probably others available that I've not heard of also. Be careful though, you want a full-featured budget management application, not just a checkbook manager. There are at least a dozen checkbook programs around that I know of, but they are not what you are looking for to manage a budget with.

Once your initial budget is in place your software should be able to support changes to it along with adjustments in it. This

means that the program must be able to keep track of the sums of money that you take in and the sums that you expend, in any form that the transactions occur, whether they be cash, checking or creditcard. Additionally, the program should be tolerant enough to allow for adjustments in budgetted amounts throughout the year. You will find out "real quick" that some of your projected amounts for the year are not accurate. SO an adjustment is necessary. Budgets should be reviewed at least quarterly, but not more often than monthly, to ensure that you are still on course.

Under ideal circumstances, meaning if everything you projected happens as planned, you will have taken in exactly 100% of income you anticipated and spent exactly 100% of the money you projected you would spend, by December 31st. Similarly, at the end of the first quarter, March 31st, 25% of your income should be realized and 25% of all your annual expenses should be incurred. The same holds true for each quarter. Measuring your progress towards an objective against the time in which you have to attain the objective is the benchmark that you can use to determine whether or not you are on course.

For example, assume that you have projected a \$2880 contribution to your retirement fund for the year. This means that you should have \$720 of it saved by March 31st. If at the end of a quarter you determine that you don't have the projected amount available, that is a red-flag that tells you its time to find out why. Assuming that you originally anticipated that the retirement fund contributions would come out in 12 equal installments, and by mid-year say, you discover that you have not achieved 50% of the objective, you have only 6 months left to get the job done. So its time to make some adjustments if the objective is to be attained. You make the adjustments by;

1. First identify why an insufficient amount of money exists in the retirement fund account. This is usually the result of unanticipated obligations in another area of the budget, less income than projected, a bookkeeping error or failure to stay within the amount allocated for a particular expense category.

2. When the offending area is identified you may be able to make an entry adjustment if it is the result of a bookkeeping error, you may be able to re-allocate funds or you may just have to live with the situation and down-size your objective to an amount more in line with your financial capabilities.

Imagine that you are in a hot air balloon and that you have an altimeter that shows your altitude in relation to a horizontal line that represents the ground below. When you first start your flight and get up to cruising altitude its like the start of a new budget year. You have charted your flight path so that maintaining a certain altitude for a specific length of time, traveling in a pre-selected direction, will get you to your destination. In other words, you have identified a goal to head towards and you have charted the objectives that will tell you how well you are doing in realizing that goal. During the flight you check your altitude, direction and speed to make sure that you are still on course. The same thing occurs with your budget. Periodically you will check things to make sure that the budget is still on the course that you set at the beginning of the year.

NEXT PAGE

If for instance your altitude has dropped you fire up the burners to raise the balloon. If your budget had dropped in a particular area you adjust it so that it is brought back to the level needed to reach your planned destination. That is how budgets are managed, by setting a destination (a goal), charting a course to get there (setting objectives) and then navigating your craft (managing your budget) effectively enough to land where you

originally planned to (reaching your goal).

END PART 3

USING "PRINT USING" WITH YOUR PRINTER
by Richard Roberts
reprint from PUG Peripheral

One of the more obscure statements available with TI Extended Basic is one called PRINT USING. Even more obscure is the fact that this statement can be used to format variables and constants that will be dumped to your printer. The Extended Basic manual, on page 150, shows several examples of how PRINT USING can be used to format data for screen display, but nary a word of how to do the same with open files. It can be done, and is much more powerful than you may realize.

Any discussion of PRINT USING will require an understanding of the IMAGE statement, so if you are not familiar with it, you better brush up on it first. The PRINT USING statement uses IMAGE in one of two ways, either with a string expression, or a line number reference. I prefer the latter, as it allows for more flexibility, but since these different methods are explained in the manual, I will limit this to a few simple examples that are not shown in the manual.

```
100 TCOST=19.55
110 IMAGE ##.##
120 OPEN #1:"PIQ"
```

```
130 PRINT #1,USING 110:TCOST
```

Running this sample program will effectively show how the PRINT USING statement will work with an open file.

Of course, there are many other variations of IMAGE that can be used, so experiment with them and watch how it performs when line 130 dumps it to the printer. Shown below are a few more examples for use with an open file.

```
110 IMAGE "##.## ##.##"
130 PRINT #1,USING 110:COST1,COST2
```

This IMAGE statement will allow you to print two (or more) variables at a pre-determined spot on the same line. The length of the string expression in the IMAGE statement can be long as you wish, up to the limit of an Extended Basic line.

```
110 IMAGE "#####.##"
130 PRINT #1,USING 110:"TOTAL COST",TCOST
```

This version shows how you can format the printed line for string data as well as numerical data. A string variable could be used in place of the string constant, as below.

```
105 A$="TOTAL COST"
110 IMAGE "#####.##"
130 PRINT #1,USING 110:A$,TCOST
```

It is also possible to place the IMAGE statement inside the PRINT USING statement, as shown below. First, delete line 110.

```
130 PRINT #1,USING "##.##":TCOST
or
130 PRINT #1,USING "#####.##"
##.##":A$.TCOST
```

A few other points to remember include the fact that IMAGE and PRINT USING can be used to round off calculated variables. A single string expression such as "#####.##" will round off a decimal align numbers as small as 0.01 up to 999999.99, and print the number at any designated location. This function could save many hours of algorithm development for accomplishing the same thing. So in the long run, the PRINT USING statement is one that any programmer should be very familiar with, and use as much as possible.

END

More Tips and Tricks
by Andy Frue
Lima US

Have you ever been editing a line and accidentally erased it by pressing FCTN 3? The old advice was to type FCTN P (*) and then press enter. You get an error then get your line back. There is an easier way to do this. Besides using any of the keys that when used alone on a

line will generate an error, you can simply press FCTN 4 then retype the line number you accidentally erased and press the up or down arrow key.

And it's a similar thing with OLD CS1. You do NOT have to press Shift E.

Just press E. I think the reason people were worried about losing a program is that the old 99/4 usually would lock up the system when an error in cassette

loading occurred. This bug was fixed in the 4A. In fact, when loading, you see a "menu" of R)ead, C)heck, or E)xit. These apply whenever you have the option of pressing ENTER. Try it. Type OLD CS1 and instead of pressing ENTER when it asks you to, press C. Then go through the usual procedure. Instead of "READING" you see "CHECKING."

END

No. 71

Tigercub Software
156 Callingwood Ave.
Columbus, OH 43213

My three Nuts & Bolts disks, each containing 100 or more subprograms, have been reduced to \$5.00 each. I am out of printed documentation so it will be supplied on disk.

My TI-PD library now has over 600 disks of fairware (by author's permission only) and public domain, all arranged by category and as full as possible, provided with loaders by full program name rather than filename. Basic programs converted to XBasic, etc. The price is just \$1.50 per disk(!), post paid if at least eight are ordered. TI-PD catalog #6 is available for \$1 which is deductible from the first order.

Newsletters recently have been reprinting a random music player from several years ago. I thought I would try to make it a bit more musical -

```
100 REM ECHO2
110 RANDOMIZE
120 DATA 165,196,247,262,330,392,523
130 DATA 131,175,220,262,349,440,523
140 DATA 147,175,196,247,294,392,494
150 FOR J=0 TO 6 :: READ C(J) :: NEXT J
160 FOR J=0 TO 6 :: READ F(J) :: NEXT J
170 FOR J=0 TO 6 :: READ G(J) :: NEXT J
180 X,Y,Z=INT(RND*7):: GOTO 200
190 Z=Y :: Y=X :: X=INT(RND*7)
200 T=T+1+(T=18)*18 :: IF T > 12 THEN GOSUB 240 ELSE IF T > 6 THEN GOSUB 230 ELSE GOSUB
```

```
220
210 CALL KEY(O,E,M):: IF N=0 THEN 190 ELSE STOP
220 CALL SOUND(-300,C(X),O,C(Y),9,C(Z),19):: RETURN
230 CALL SOUND(-300,F(X),O,F(Y),9,F(Z),19):: RETURN
240 CALL SOUND(-300,G(X),O,G(Y),9,G(Z),19):: RETURN
```

In a previous Tip, I had a short routine to compute First Class postage. In the Bluegrass 99ers newsletter, Mark Schafer published an even shorter version -

```
100 INPUT "OUNCES? ":A :: PR
INT .06-.23*INT(-A):: GOTO 100
```

Mark gives some other useful algorithms. INT(X) will round DOWN to the nearest integer, but remember that a negative number is also rounded downward, therefore INT(-4.1)=5.

INT(X+.5) rounds a number to the NEAREST integer; a number ending on .5 is rounded up.

-INT(-X) rounds UPWARD to the next integer, so that -INT(-4.1)=5

-INT(.5-X) rounds to the NEAREST integer but a number ending in .5 is rounded down instead of up.

INT(X*10+.5)/10 rounds to the nearest tenth.

INT(X*10)/10 truncates to the nearest tenth.

Thanks, Mark

I still like to program "brain games", so here is another one. I developed it from a much simpler version in an ad for membership in the Mensa Society of really brainy people, so I call it the Super Mensa Puzzle. It's not so hard at Level 1 or 2.

```
100 CALL CLEAR :: FOR S=0 TO 12 :: CALL COLOR(8,2,16):: NEXT S :: CALL BSCREEN(5)
110 DISPLAY AT(3,3)ERASE ALL
:"THE SUPER MENSA PUZZLE": ""
:" X restart ? help @ quit": ""
P + M - T *
```

```
D /"
120 lby Jim Peterson
130 DISPLAY AT(9,1):" Put a P(lus) or +, or "-"(minus) or r -, or I(imes) or "*" (ivide by) or / in"
140 DISPLAY AT(12,1):"each blank to reach the:"total on the sprite."
150 DISPLAY AT(15,1):" Use X to start over, ? to:"give up and see answer, @ to:"quit."
160 DISPLAY AT(19,1):"Difficulty level (1 - 5)?" :: CALL CALLKEY(19,27,"12345",L):: L=VAL(L):: DISPLAY AT(9,1):RPT$( " ",254)
170 DISPLAY AT(19,1):""
180 FOR J=0 TO 4+L :: Z#=Z#&STR$(J)&" " :: NEXT J :: Z#=SE6$(Z#,1,LEN(Z#)-1)&"=" :: ZC=(4+L)*2+3
190 RANDOMIZE :: GOSUB 320
200 DISPLAY AT(12,1):Z#&STR$(T):: "":: "":: C=2 :: N=0 :: N2=1 :: T=0 :: V#="P-M*T/DX?1@" :: CALL DELSPRITE(ALL)
210 CALL MAGNIFY(2):: S#="STR$(T):: R=63 :: CC=95 :: FOR J=1 TO LEN(S#):: CALL SPRITE(#J,ABC(SE6$(S#,J,1)),5,R,C):: CC=CC+20 :: NEXT J
220 CALL CALLKEY(12,C,V#,X#)
230 ON POS(V#,X#,1)GOSUB 280,280,290,290,300,300,310,310,200,370,370,390
240 IF C<ZC-2 THEN 220 ELSE IF T=T# THEN 260
250 FOR J=1 TO 20 :: DISPLAY AT(15,6):"WRONG!" :: DISPLAY AT(15,6):"wrong!" :: NEXT J :: DISPLAY AT(15,6):"" :: GOTO 200
260 IF X#="1" OR X#="?" THEN 270 ELSE DISPLAY AT(15,6):"RIGHT!" :: CALL SOUND(200,196,5):: CALL SOUND(500,523,2) :: GOSUB 320 :: GOTO 200
270 DISPLAY AT(15,5):"PRESS ANY KEY" :: DISPLAY AT(15,5):"press any key" :: CALL KEY(O,K,S):: IF S=0 THEN 270 ELSE GOSUB 320 :: GOTO 200
280 DISPLAY AT(12,C):"+": "" :: T=T+N2 :: GOSUB 380 :: RETURN
290 DISPLAY AT(12,C):"-": "" :: T=T-N2 :: GOSUB 380 :: RETURN
```

```
300 DISPLAY AT(12,C):"*": "" :: T=T*N2 :: GOSUB 380 :: RETURN
310 DISPLAY AT(12,C):"/": "" :: T=T/N2 :: GOSUB 380 :: RETURN
320 T=0 :: FOR J=1 TO 4+L
330 X=INT(RND*4+1):: IF X=1 THEN T=T+J :: P$(J)="+" :: GOTO 350 ELSE IF X=2 THEN T=T-J :: P$(J)="-" :: GOTO 350 ELSE IF X=3 THEN T=T*J :: P$(J)="*" :: GOTO 350
340 IF T/J<>INT(T/J)THEN 330 ELSE T=T/J :: P$(J)="/"
350 NEXT J
360 IF T>9999 OR T<-999 THEN 320 ELSE T=T :: RETURN
370 DC=2 :: FOR J=1 TO 4+L :: DISPLAY AT(12,DC):P$(J):: DC=DC+2 :: ON POS(V#,P$(J),1)GOSUB 280,280,290,290,300,300,310,310 :: NEXT J
380 T#="STR$(T):: T#=SE6$(T#,1,13-L):: DISPLAY AT(12,ZC):T# :: C=C+2 :: N=N+1 :: N2=N2+1 :: RETURN
390 CALL CLEAR :: STOP
400 SUB CALLKEY(R,C,V#,K#)
410 CALL HCHAR(R,C+2,30):: FOR T=1 TO 3 :: CALL KEY(3,K,S):: IF S<>0 THEN 440
420 NEXT T :: CALL HCHAR(R,C+2,20):: FOR T=1 TO 3 :: CALL KEY(3,K,S):: IF S<>0 THEN 440
430 NEXT T :: GOTO 410
440 IF POS(V#,CHR$(K),1)=0 THEN 410 ELSE K#=CHR$(K)
450 SUBEND
```

Programmers might be interested in the CALLKEY subprogram in lines 400-450 which allows a one-character ACCEPT without pressing Enter, with blinking cursor and with validation. Since the - and * and + symbols require the Shift key, I provided alternative P, M, T and D key input - in either upper or lower case, since key mode 3 accepts either as upper case.

The Lima newsletter has a tip by Andy Frueh on converting a QV40 file to QV80 by using a sector editor to find find byte >11 of the

NEXT PAGE

header sector and change it from >28 to >50. If you dislike sector editing and hex numbers as much as I do, you might prefer this method. It comes in handy to convert program listings, listed as DV28 files in 28-column size with Super Extended Basic, into DV80 files so that I can get them into Funnelweb to incorporate them into my text, such as this -

```
100 DISPLAY AT(12,1)ERASE ALL:
L:"Input file? DSK" :: ACCEPT AT(12,16):IN$ :: DISPLAY AT(14,1):"Record length?" :: ACCEPT AT(14,16):IL
110 DISPLAY AT(16,1):"Output file? DSK" :: ACCEPT AT(16,17):OUT$ :: DISPLAY AT(18,1):"Record length?" :: ACCEPT AT(18,16):OL
120 OPEN #1:"DSK"&IN$,VARIABLE IL :: OPEN #2:"DSK"&OUT$,VARIABLE OL
130 LINPUT #1:M$ :: PRINT #2:M$ :: IF EOF(1)<>1 THEN 130 ELSE CLOSE #1 :: CLOSE #2
```

After I published an article about writing a program to add blanks to records in a file, Stephen Shaw wrote to me from England and Chas. Stringer wrote from Decatur to point out that I could have done the same thing with Tl-Writer or Funnelweb Replace String. I had never realized that you could specify columns with the RS command.

For instance, to get a left margin of two spaces, use CTRL O to get the open cursor, put the cursor at the beginning of the first line, use FCTN 8 to put a blank first line above it, then FCTN 9 and RS and enter O O // / and then A for all lines.

Stephen Shaw published in MICROpendium a routine to add fractions, and challenged anyone to improve it to reduce the result. Dean

Mah sent in a subprogram to do the reduction. Bruce Harrison sent me this greatly speeded-up version.

```
1 | ADD FRACTIONS WITH REDUCTION
2 | MODIFIED FOR REDUCTION AND SPEED ENHANCEMENT BY BRUCE HARRISON
3 | ORIGINAL PROGRAM BY STEPHEN SHAW
4 | VERSION ADFR6 OF 1 APRIL 93
90 ON WARNING NEXT
100 DISPLAY AT(10,5)ERASE ALL:
L:"--- + --- = ---"
110 ACCEPT AT(9,5)SIZE(3)VALIDATE(DIGIT):A
120 ACCEPT AT(11,5)SIZE(3)VALIDATE(DIGIT):B :: IF B=0 THEN 120
130 ACCEPT AT(9,11)SIZE(3)VALIDATE(DIGIT):C
140 ACCEPT AT(11,11)SIZE(3)VALIDATE(DIGIT):D :: IF D=0 THEN 140
150 FOR L=MAX(B,D)TO B*D STEP MAX(B,D)
160 IF INT(L/B)<L/B THEN 180
170 IF INT(L/D)=L/D THEN 190
180 NEXT L
190 N=L/B*A+L/D*C
200 FOR Y=2 TO MIN(N,L)
210 IF N/Y>INT(N/Y)THEN 230
220 IF L/Y=INT(L/Y)THEN N=N/Y :: L=L/Y :: GOTO 200
230 NEXT Y
240 DISPLAY AT(9,16):USING "####":N
250 DISPLAY AT(11,16):USING "####":L
260 DISPLAY AT(14,3):"PRESS ENTER FOR ANOTHER"
270 DISPLAY AT(1,1):"NORMAL RESULT=";A/B+C/D
280 CALL KEY(O,K,B):: IF B<>1 THEN 280 ELSE IF K=13 THEN 100
```

This one was published long ago in Tips #31 but someone asked for an encoding and program. This one will make codes that should be quite difficult to crack. Both parties must have the program with the same random code created by CODEPRINT merged in, but you can make

any number of merge files and merge them in according to the day of the week, etc.

First we need one of those those programs that write a program -

```
100 ICODEPRINT by Jim Peterson - creates a random code in a MERGE format program CODESTRING to be MERGEed into CODEMAKER
110 FOR J=1 TO 254 :: M$=M$&CHR$(J):: NEXT J
120 FOR J=1 TO 254 :: RANDOMIZE :: X=INT(RND*LEN(M$)+1): C$=C$&SE6$(M$,X,1):: M$=SE6$(M$,1,X-1)&SE6$(M$,X+1,LEN(M$)):: NEXT J
130 OPEN #1:"DSK1.CODESTRING",VARIABLE 163,OUTPUT :: PRINT #1:CHR$(0)&CHR$(1)&"C$"&CHR$(190)&CHR$(199)&CHR$(127)&SE6$(C$,1,127)&CHR$(0)
140 PRINT #1:CHR$(0)&CHR$(2)&"C2$"&CHR$(190)&CHR$(199)&CHR$(127)&SE6$(C$,128,127)&CHR$(0)
150 PRINT #1:CHR$(0)&CHR$(3)&"C$"&CHR$(190)&"C$"&CHR$(184)&"C2$"&CHR$(0):: PRINT #1:CHR$(255)&CHR$(255):: CLOSE #1 :: END
```

And now the coder/decoder - 100 ITIGERCUB CODEMAKER written by Jim Peterson

110 The MERGE format program CODESTRING created by the program CODEPRINT must be MERGEed into lines 1-3 of this program

```
120 DIM A$(254):: DISPLAY AT(3,6)ERASE ALL:"TIGERCUB CODEMAKER" :: DISPLAY AT(12,1):"Do you want to": "(1)Encode": "(2)Decode"
130 CALL KEY(O,K,B):: IF K=49 THEN 140 ELSE IF K=50 THEN 290 ELSE 130
140 OPEN #1:"DSK1.CODE",VARIABLE 254,OUTPUT
150 DISPLAY AT(5,6)ERASE ALL:"Type message in segments of": "not more than 254 characters": "and Enter. When done, type"
160 DISPLAY AT(9,1):"END and Enter. Type slowly": "to avoid skipped characters.": "Backspace with FCTN 8 to": "corr
```

```
ect.": "Press any key"
170 CALL KEY(O,K,B):: IF B=0 THEN 170
180 CALL CLEAR :: CALL LONGACCEPT(O,M$):: IF M$="END" THEN 280
190 DISPLAY AT(20,1):"WAIT, PLEASE - ENCODING"
200 FOR J=1 TO LEN(M$)
210 A$(ASC(SE6$(C$,J,1)))=SE6$(M$,J,1)
220 NEXT J
230 FOR J=1 TO 254 :: RANDOMIZE
240 IF A$(J)="" THEN A$(J)=CHR$(INT(26*RND+65))
250 CODE$=CODE$&A$(J)
260 NEXT J :: PRINT CODE$
270 PRINT #1:CODE$ :: CODE$="" :: FOR J=1 TO 254 :: A$(J)="" :: NEXT J :: GOTO 180
280 CLOSE #1 :: END
290 OPEN #1:"DSK1.CODE",VARIABLE 254,INPUT :: CALL CLEAR :: DISPLAY AT(12,10):"DECODING"
300 LINPUT #1:CODE$ :: FOR J=1 TO 254 :: M$=M$&SE6$(CODE$,ASC(SE6$(C$,J,1)),1):: NEXT J :: PRINT M$:: M$=""
310 IF EOF(1)<>1 THEN 300 :: CLOSE #1 :: END
320 SUB LONGACCEPT(L,M$):: X=0 :: IF L<>0 THEN R=L ELSE R=R+1
330 M$="" :: C=3 :: CH=140 :: CALL CHAR(140,RPT$(O",14)&"FF")
340 CALL HCHAR(R,C,CH):: CH=CH+5+(CH=160)*25 :: CALL KEY(O,K,B):: IF B<1 THEN 340
350 IF K<>8 THEN 370 :: X=X-1 :: C=C-1 :: IF C=2 THEN C=30 :: R=R-1
360 M$=SE6$(M$,1,LEN(M$)-1): GOTO 340
370 IF K=13 THEN 410
380 X=X+1 :: M$=M$&CHR$(K):: CALL HCHAR(R,C,K):: IF X=254 THEN 410
390 C=C+1 :: IF C=31 THEN C=3 :: R=R+1 :: IF R=25 THEN CALL CLEAR :: R=1
400 GOTO 340
410 R=0 :: SUBEND
```

Memory full,
Jim Peterson

PROGRAM OF THE MONTH
 by Bob August
 reprinted from
 BUG NEWS

The program this month has two versions. Both versions will read a disk in drive number one and create a catalog of the disk in drive number one on a data disk in drive number two. Version number one will catalog up to 10 disks at one time while version number two will only catalog one disk. Both versions create a DV/BO file which can be printed out or edited with TI-Writer. We left a margin of 6 on the left side so your printout could be hole punched and inserted into a ring binder.

When using the program DO NOT EXIT THE PROGRAM WITH FCTN-4. Use the "S" option to exit the program. Just leave the last disk in drive number one and press enter for another disk and then press "S". If you use FCTN-4 or FCTN- to exit the program, the data file in drive two will be left open and you will lose all of your data.

If you want to catalog more than 10 disk at one time, change line 210 of version two from for DISK=1 to 10 to any number you wish.

Hope you enjoy.

```

100 ! DISK FILE CATALOG V1 !
208
110 ! IN TI EXTENDED BASIC !
234
120 ! BY R.W. AUGUST !092
130 DIM FT$(5)!179
140 FT$(1)="DIS/FIX" :: FT$(
2)="DIS/VAR" :: FT$(3)="INT/
FIX" :: FT$(4)="INT/VAR" ::
FT$(5)="PROGRAM" !145
150 DISPLAY AT(2,3)ERASE ALL
:"<< DISK FILE CATALOG>>" !1
76
160 DISPLAY AT(8,1):"PLACE D
ISK TO BE CATALOGED": "IN D
RIVE NUMBER ONE." !144
170 DISPLAY AT(12,1):"PLACE
DISK TO SAVE DATA ON": "IN
DRIVE NUMBER TWO.": "PRESS
ENTER WHEN READY." !087
180 CALL KEY(0,K,S):: IF K<>
13 THEN 180 :: CALL HCHAR(8,

```

```

1,32,288)!085
190 ON ERROR 370 :: OPEN #2:
"DSK2.FILECAT",OUTPUT :: PRI
NT #2:TAB(6);"FILE NAME SIZ
E TYPE DISK NAME FILE CO
MMENTS" !239
200 PRINT #2:TAB(6);"-----
-----"
* !074
210 FOR DISK=1 TO 10 :: ON E
RROR 370 :: OPEN #1:"DSK1.",
INPUT ,RELATIVE,INTERNAL ::
INPUT #1:X$,M,X,Y :: U=(X-Y)
+2 !069
220 DISPLAY AT(6,1):"Disknam
e is ";X$;TAB(24);"@";STR$(D
ISK): "Available =";Y;TAB(1
8);"Use =";U !072
230 DISPLAY AT(11,1):"Catalo
g this disk Yes/No Y": "or En
ter S to stop." !154
240 ACCEPT AT(11,26)VALIDATE
("NnSsYy")SIZE(-1):YN$ :: IF
YN$="N" OR YN$="n" THEN DIS
K=DISK-1 :: GOTO 320 !226
250 IF YN$="S" OR YN$="s" TH
EN CLOSE #1 :: GOTO 360 !039
260 FOR F=1 TO 127 :: INPUT
#1:A$,A,J,K :: IF LEN(A$)=0
THEN 320 ELSE DISPLAY AT(11,
1):"FILE NAME SIZE TYPE":"-
-----" !066
270 DISPLAY AT(13,1):A$;TAB(
12);J;TAB(17);FT$(ABS(A)): :
:"Enter Comment Yes/No Y" :
: ACCEPT AT(16,22)VALIDATE("
NnYy")SIZE(-1)BEEP:YN$ !097
280 DISPLAY AT(16,1):" " :: I
F YN$="N" OR YN$="n" THEN FI
LECOM$=" " :: GOTO 300 !238
290 DISPLAY AT(16,1):"Enter
File Comment" :: ACCEPT AT(1
8,1):FILECOM$ !055
300 PRINT #2:TAB(6);A$;TAB(1
7);: PRINT #2,USING "###":
J;: PRINT #2:TAB(22);FT$(AB
S(A));TAB(30);X$;TAB(41);FIL
ECOM$ !228
310 DISPLAY AT(18,1):" " ::
NEXT F !218
320 CLOSE #1 :: DISPLAY AT(6
,1):"":"PLACE THE NEXT DISK
TO BE":"CATALOGED IN DRIV
E ONE.":"LEAVE THE DATA D
ISK IN DRIVE" !187
330 DISPLAY AT(12,1):"":"NUM
BER TWO.":"PRESS ENTER WH
EN READY": " " !137
340 CALL KEY(0,K,S):: IF K<>

```

```

13 THEN 340 :: CALL HCHAR(7,
1,32,320)!232
350 NEXT DISK !193
360 CLOSE #2 :: STOP !178
370 ON ERROR 380 :: DISPLAY
AT(12,1)ERASE ALL BEEP:"ERRO
R DETECTED ON DISK" :: CLOSE
#1 !073
380 ON ERROR 390 :: CLOSE #2
!170
390 FOR D=1 TO 1000 :: NEXT
D :: CALL CLEAR :: END !130
-----
100 ! DISK FILE CATALOG V2 !
209
110 ! IN TI EXTENDED BASIC !
234
120 ! BY R.W. AUGUST !092
130 DIM FT$(5)!179
140 FT$(1)="DIS/FIX" :: FT$(
2)="DIS/VAR" :: FT$(3)="INT/
FIX" :: FT$(4)="INT/VAR" ::
FT$(5)="PROGRAM" !145
150 DISPLAY AT(2,3)ERASE ALL
:"<< DISK FILE CATALOG>>" !1
76
160 DISPLAY AT(8,1):"PLACE D
ISK TO BE CATALOGED": "IN D
RIVE NUMBER ONE." !144
170 DISPLAY AT(12,1):"PLACE
DISK TO SAVE DATA ON": "IN
DRIVE NUMBER TWO.": "PRESS
ENTER WHEN READY." !087
180 CALL KEY(0,K,S):: IF K<>
13 THEN 180 :: CALL HCHAR(8,
1,32,288)!085
190 ON ERROR 350 :: OPEN #1:
"DSK1.",INPUT ,RELATIVE,INTE
RNAL :: INPUT #1:X$,M,X,Y ::
U=(X-Y)+2 !100
200 DISPLAY AT(6,1):"Disknam
e is ";X$: "Available =";Y;
TAB(18);"Used =";U !154
210 DISPLAY AT(15,1):"Enter
Disk Comment Yes/No Y" :: AC
CEPT AT(15,27)VALIDATE("NnYy
")SIZE(-1)BEEP:YN$ !164
220 IF YN$="N" OR YN$="n" TH
EN DISKCOM$=" " :: GOTO 240
!225
230 ACCEPT AT(17,1):DISKCOM$
:: DISPLAY AT(15,1):"":" "
!206
240 ON ERROR 350 :: OPEN #2:
"DSK2.DISKCAT",OUTPUT :: PRI
NT #2:TAB(6);X$;TAB(17);"FRE
E":STR$(Y);TAB(30);"USED":
STR$(U);TAB(41);DISKCOM$ !14
3
250 PRINT #2:TAB(6);"-----

```

MORE ON LINE NOISE
by Robert K. Ricketts
reprinted from the HUG

(Editors note: This article by R.K. Ricketts appeared in the March issue of the HUGers newsletter and I thought it would be of interest to the rest of you since we published an article in the April issue of May issue of the SPIRIT of '99. This is a different approach to the problem.)

Many people have left messages on my bulletin board asking me why there are so many /garbage/ characters on their screens and why file transfers are riddled with errors. These garbage characters are really line noise and can be introduced in many different places. One of the more common and familiar introduction points of line noise is in the telephone company's system and even here there are several ways noise is introduced. A signal is routed through multiple stations before it eventually makes it to the other end and some of these stations aren't exactly new. Older areas may have older, less sophisticated equipment that is more apt to be affected by ambient noise. This is one reason some people continue to have noise problems even after hanging up and calling back multiple times. Also, a given physical connection at one of these junctions may not be up to snuff. If your particular bout of line noise is solved by hanging up and calling back, then it's probably that you were previously connected through an intermittent or /dirty/ connection. Some of these trunk lines (large, multi-caller /pipes/) may pass through an area that has a lot of ambient RFI (Radio Frequency Interference) present, although this is not usually the case.

Another common noise introduction point is in your home. Most residential homes have televisions, radios, microwave ovens, VCR's, touch lamps, and if you are reading this, a micro-computer. All these devices radiate radio waves that can (and often do) get into the phone lines and cause noise. Electric motors and mechanical dialer controls can introduce noise into the electrical wiring in your house and cause problems. If your line noise problem does not go away after repeated hanging up and call-

ing back, then you may be suffering from one of those household problems. If you are suffering from this problem, you can take steps to eliminate it. First of all, turn off EVERYTHING except the fridge (If it is the fridge, then you're SOL. Can't live life with your ice box unplugged) and see if the noise persists. If it goes away, then start turning things back on, checking the computer each time until you see the noise start up again. It may be that a single device is not bugging you but several devices plotting together to annoy you. This elimination tournament may take awhile.

Another area to check is the wiring at the computer. Use noise suppressors on your power connections to both the computer and the modem. Use shielded RS-232 cable to connect your modem to the computer. Ribbon cables (especially long runs of it) are great antennas and will cause problems. Re-route the RS-232 cable so it does not run next to the computer power supply or any other transformer. Make sure the cable does not run near the monitor. If you are particularly adventuresome, you can line the interior of the monitor with foil and ground it with a ribbon grounding strap. BE VERY CAREFUL if you attempt this. Monitors generate THOUSANDS of volts of electricity and can knock you clear into next week. You'd best NOT attempt this unless you are experienced in electronics. If you live near a freeway or highway, then interference from CB radio can present a problem. Many interstate truckers have 100+ watts of power (illegally) on their CB rigs and frequently have a sloppy amplifier that can emit spurious radiation all over the radio spectrum.

Modems operating at 2400 BPS are much more intolerant of line noise than are modems operating at 1200 BPS. Conversely, modems capable of 2400 BPS operate better at 1200 BPS only modems.

If you are being hopelessly attacked by noise at 2400 BPS, try calling back at 1200 BPS. It's very possible that the noise will be greatly reduced or disappear altogether. I know, you didn't buy a 2400 BPS modem just to retard it to 1200 BPS. The brand of modem plays a part in the immunity to line noise. Some modems can digest more noise (lower

signal-to-noise ratio) than others. Most 2400 modems operating at 1200 have approximately -8 to -10 db error threshold while the same modem has -16 to -28 db threshold operating at 2400. For this reason, line quality is much more critical at 2400 operation.

If you get transmission errors while downloading or uploading a file, don't fret it. The Xmodem (or whatever protocol) incorporates an error checking/correction mechanism that automatically detects and corrects any error in the first place means that he caught it and corrected it. The only errors you have to worry about are the ones that Xmodem does NOT report. Any reported error has already been corrected. Xmodem, especially the CRC favored one, is a very reliable file transfer protocol. Even if you got 100 errors during transmission, chances are still pretty slim that the file got corrupted. Occasionally, a file will be corrupted after transfer, but many times this may be due to a bad ARC-ing of the file or perhaps a disk error that may have occurred sometime during the files' past.

MODEM NOISE KILLER

Here is a kit that is easy to build and may solve your noise problems, however conditions may be different due to modem type, amount of line noise, old or new switching equipment (Bell's equipment), and so on.

Here is what you will need from Radio Shack:

1 #279-374 Modular line cord if you don't already have one. (You won't need one if your phone has a modular plug in its base). \$4.95

1 #279-420 Modular surface mount jack (4 or 6 conductor) \$4.95

1 #271-1720 Potentiometer. This a 5K audio taper variable resistor. \$1.09

1 272-1055 Capacitor. any non-polarized 1.0 to 1.5 uf cap. should do. Paper, Mylar, or metal film caps should be used, although #272-996 may work well. (#272-996) \$.79.

1 100 ohm resistor - quarter or

NEXT PAGE

half watt. \$.19.

1 #279-357 Y-type or duplex modular connector. Don't buy this until you've read the section on connecting the Noise Killer below. (A,B,C) \$4.95

SUBSTITUTIONS: If you have these components laying around doing nothing

2MFD's of capacitance (two 1MFD's in parallel). Two R.S. p/n 272-1055 work fine. Remember that about 90 Volts will appear across red & green at ring, so the caps should be rated at 100VDC+.

Use resistors with a final series resistance value (100 ohm + pot) of 2.75K. One could probably use 2MFD and a fixed 2.7K resistor and do the job 90% of the time. The adjustment of the pot is not very critical. Changes of +/-1K made very little difference in performance of the circuit. add up to 100 ohms.

First off, open the modular block. You normally just pry them open with a screwdriver. Inside you'll find up to 6 wires. Very carefully cut out all but the green and red wires. The ones you will be removing should be black, yellow, white and blue. These wires won't be needed and may be in the way. So cut them as close to where they enter the plug as possible. The other end of

these wires have a spade lug connector that is screwed into the plastic. Unscrew and remove that end of the wires as well. Now, you should have two wires left. Green and red. Solder one end of the capacitor to the green wire. Solder the other end of the capacitor to the center lug of the potentiometer. Solder one end of the resistor to the red wire. You may want to shorten the leads of the resistor first. Solder the other end of the resistor to either one of the remaining lugs of the pot, doesn't matter which. Now to wrap it up, make a hole in the lid of the mod block to stick the shaft of the pot through. Don't make this hole dead center as the other parts may not fit into the body of the mod block.

See how things will fit in order to find where the hole will go.

Well, now that you've got it built you'll need to test it. First twist the shaft on the pot until it stops. You won't know which way to turn it until later. It doesn't matter which way now. You also need to determine where to plug the Noise Killer into the telephone line. It can be done by one of several ways:

A. If your modem has two modular plugs in back, connect the Noise Killer into one of them using a line cord.

B. If your phone is modular, you can unplug the cord from the back of it after you're on-line and plug the cord into the Noise Killer.

C. You may have to buy a Y-type modular adapter. Plug the adapter into a wall outlet, plug the modem into one side. Call a BBS that has known noise problems. After you've connected and garbage begins to appear, plug the Noise Killer into the adapter. If you have turned the shaft on the pot the wrong way you'll find out now. You may get a lot of garbage or even disconnected. If this happens, turn the shaft the other way until it stops and try again. If you don't notice much difference when you plug the Noise Killer in, that may be a good sign. Type in a few commands and look for garbage characters on the screen. If there still is, turn the shaft slowly until most of it is gone. If nothing seems to happen at all, turn the shaft slowly from one side to the other. You should get plenty of garbage or disconnected at some point. If you don't, reread this message to make sure you've connected it right.

END

PROGRAM

```

-----
-----
!225
260 PRINT #2;TAB(6);"FILE NAME SIZE TYPE DISK NAME FILE COMMENTS" !075
270 PRINT #2;TAB(6);"-----"
-----
-" !120
1 280 FOR F=1 TO 127 :: INPUT #1:A$,A,J,K :: IF LEN(A$)=0 THEN 340 ELSE DISPLAY AT(11,1):"FILE NAME SIZE TYPE":"-----" !112
290 DISPLAY AT(13,1):A$;TAB(

```

```

12);J;TAB(17);FT$(ABS(A)): :
:"Enter Comment Yes/No N" :
: ACCEPT AT(16,22)VALIDATE("NnYy")SIZE(-1)BEEP:YN$ !086
300 DISPLAY AT(16,1):" " :: IF YN$="N" OR YN$="n" THEN FILECOM$=" " :: GOTO 320 !002
310 DISPLAY AT(16,1):"Enter File Comment" :: ACCEPT AT(18,1);FILECOM$ !055
320 PRINT #2;TAB(6);A$;TAB(17);: PRINT #2,USING "###": J;: PRINT #2;TAB(22);FT$(ABS(A));TAB(30);X$;TAB(41);FILECOM$ !22B
330 DISPLAY AT(18,1):" " ::

```

```

NEXT F !218
340 CLOSE #1 :: CLOSE #2 ::
STOP !203
350 ON ERROR 360 :: DISPLAY AT(12,1)ERASE ALL BEEP:"ERROR DETECTED ON DISK" :: CLOSE #1 !053
360 ON ERROR 370 :: CLOSE #2 !150
370 FOR D=1 TO 1000 :: NEXT D :: CALL CLEAR :: END
3

```

END

CHECKSUM
by Tom Freeman
reprinted from
VAST News 12/92

(This article and the accompanying program first appeared in the LA Topics, newsletter of the Los Angeles 99/4A User Group.) The program in this article was modified by C.O.N.I. member, Harley Ryan.

Have you ever typed in a TI-99/4A version of a BASIC program from a magazine and noticed that the versions of the program for other computers have little numbers at the end of the lines that you don't have? They were for error checking on your typing, to ensure no mistakes.

Have you ever laboriously typed in a long program and run it, only to find that it crashes? Or doesn't work as it is supposed to, all because of a simple typing error that you can't find?

So, why doesn't TI have one? Now you do!

This may be the most useful program that I have published for general use, because almost everyone does BASIC at one time or another. It involves only a few extra steps for the programmer, and one for the user who is typing the published program in.

It is really a simple method, and depends on the manner in which the TI stores BASIC programs. However, it requires a memory expansion and disk drive and works only in Extended BASIC (although BASIC programs can be entered in XBASIC, SAVED and then RUN in BASIC).

You may remember the format in which "MERGE" type programs are stored on disk. The MERGE format is actually a duplicate of the way in which the actual program is stored in memory, or on disk. The difference is it is a display type file, with each record starting with two bytes for the line number, and then the actual program line. In memory, however, the program lines are stored contiguously, and in seemingly random order. (Actually, the order depends on the order in which they were entered).

A separate line number table is stored

below the program area and keeps track of the line numbers and the pointers to where each line begins. Each line consists of one byte "tokens" for all reserved words with all strings, including the names of subprograms such as LOAD, SCREEN, etc, being spelled directly.

When you enter any line in Extended BASIC (either a command or program line with the line number coming first), it is first moved to the so-called "edit buffer" at address >8C0 in VDP. The BASIC bias is preserved. The purpose of this is that if you press FCTN 8 (REDO) then the whole line or lines can be retrieved. Next everything is "crunched" by replacing each reserved word with its token, subtracting the BASIC bias from strings, computing their length, etc., and placing the results in the "crunch buffer" at >820 in VDP. Once it is there, it can be transferred to the appropriate place in memory expansion. This is the area that is used when my program computes the "checksum" by merely adding the value of each byte! The number is never allowed to go over hex >FF - the high byte is ignored (thus in decimal, no number over 255). The assumption is that it is extremely unlikely, probability approaching zero, that a small number of mistakes will result in a number that differs by exactly 256 or a multiple thereof. The one exception is that if you transpose two characters, there is nothing I can do about that.

When the program is completely debugged and you are satisfied it is workable, then the following steps are required to insert the checksum numbers at the end of each line.

1. SAVE your program in MERGE format.
EX: SAVE "DSK1.A" <enter>
(or any filename).
2. RUN the CHECKSUM program listed in this article. (don't include the checksum characters).
3. INPUT-> DSK1.A (file name must be same as in step 1 <enter>).
OUTPUT-> DSK1.B <enter>.
4. At the prompt SAVE TO PRINT (Y/N):
"Y" will send results to printer.
"N" will display results on screen (the results will look similar to this).

LINE#-CKSM#
100 - !128

(this will repeat until all lines are checked).

When finished, the program will still be in MERGED format. Now load the merged file back into the computers memory by typing NEW <enter> then type MERGE DSK1.B <enter> (filename must be the same as OUTPUT-> above). Now for fun go ahead and LIST the program and look for the checksum characters that appear at the end of each line. You can re-save this with any filename you choose, and your file will be in PROGRAM format. I would like to add that this program does not have a "load" feature assigned. If you save this program as DSKn.LOAD it will auto load in EXTENDED BASIC.

Notice the exclamation and three numbers at the end of each line? The program was RUN on itself!

CHECKSUM

```
1 @=0 :: [=1 :: ]=2 :: _=3 :  
: \=4 :: GOTO 100 :: AS,B$,E  
$,IN$,N$,DN$,P$,R$ :: @,C,D,  
I,K,L,N,O,P,S,T,X,Y,Z,[,\,],  
_ :: CALL CLEAR :: CALL COLD  
R :: CALL KEY :: CALL SCREEN  
!092  
2 !@P-  
100 CALL CLEAR :: CALL SCREE  
N(S):: FOR C=@ TO 14 :: CALL  
COLOR(C,16,1):: NEXT C !128  
110 DISPLAY AT(1,6)ERASE ALL  
:"CREATE CHECKSUMS": : " IS F  
ILE IN MERGE FORMAT?" :: ON  
WARNING NEXT :: I,O,P=@ !203  
120 DISPLAY AT(8,1):"INPUT->  
";"DSK " :: ACCEPT AT(8,12)  
BEEP SIZE(12):IN$ :: IF IN$=  
"" THEN 120 !172  
130 ON ERROR 320 :: OPEN #[:  
"DSK"&IN$,INPUT ,DISPLAY ,VA  
RIABLE 163 :: I=[ !032  
140 DISPLAY AT(10,1):"OUTPUT  
> ";"DSK" :: ACCEPT AT(10,12)  
)BEEP SIZE(12):DN$ :: IF DN$  
="" THEN 140 !051  
150 ON ERROR 320 :: OPEN #[:  
"DSK"&DN$,OUTPUT,DISPLAY ,VA  
RIABLE 163 :: O=[ !147  
160 DISPLAY AT(12,1):"SAVE T  
O PRINT (Y/N)" :: ACCEPT AT(  
12,21)BEEP VALIDATE("YN")SIZ
```

NEXT PAGE

```

E(1):P% :: IF P%="Y" THEN GO
SUB 250 !055
170 DISPLAY AT(16,6):"LINE#
- CKSM#" !116
180 LINPUT #1:A% :: IF LEN(A
%)=] THEN CLOSE #[ :: PRINT
#]:CHR$(255)&CHR$(255):: CLO
SE #] :: GOTO 290 !053
190 I=ASC(A%)+256+ASC(SEG$(A
%,],(1)):: B%=SEG$(A%,_,163):
; L=LEN(B%): IF L>157 THEN
230 !074
200 N=0 :: FOR X=[ TO L :: Y
=ASC(SEG$(B%,X,(1)):: N=N+Y ;
; NEXT X :: N=N AND 255 :: N
0=STR$(N):: N%=RPT$("0",_-LE
N(N%))&N% !196
210 DISPLAY AT(18,7):Z;TAB(1
4);N% :: PRINT #]:SEG$(A%,[,
L+[]&CHR$(131)&N%&CHR$(0)::
IF P=[ THEN 220 ELSE 180 !23
2
220 PRINT #_:TAB(T);Z;" - "
;N% :: GOTO 180 !047
230 DISPLAY AT(21,13):"WARNI
NG!": " LINE";Z;"IS TOO LONG!
": "PRESS ANY KEY TO CONTINUE
" !192
240 CALL KEY(0,K,S):: IF S<[
THEN 240 ELSE IF D=[ THEN P
RINT #]:A% :: GOTO 180 !133
250 DISPLAY AT(18,6):"PRINTE
R MUST BE ON!" :: OPEN #_: "P
IO",OUTPUT :: DISPLAY AT(20,
[]): "SET TAB(58max)" :: P=[
!105
260 ACCEPT AT(20,17)BEEP VAL

```

```

IDATE(DIGIT)SIZE( ):T :: E%=
CHR$(27):: PRINT #_:E%;"I";C
HR$(T):: PRINT #_:E%;"A";CH
R$(9)!000
270 PRINT #_:CHR$(15):: PRIN
T #_:TAB(T);"LINE# - CKSM#"
!157
280 DISPLAY AT(18,[]):"" :: D
ISPLAY AT(20,[]):"" :: RETURN
!253
290 IF P=[ THEN CLOSE #_ !15
3
300 DISPLAY AT(22,[]):"ANOTHE
R PROGRAM (Y/N)" :: ACCEPT A
T(22,22)BEEP VALIDATE("YN")S
IZE(1):R% :: IF R%="Y" THEN
110 ELSE 310 !088
310 CALL CLEAR :: END !222
320 DISPLAY AT(10,1)ERASE AL
L:"AN ERROR HAS OCCURED": "
CHECK DRIVE NO & FILENAME":
:"PRESS ANY KEY TO CONTINUE"
!212
330 CALL KEY(0,K,S):: IF S<[
THEN 330 :: IF I=[ THEN CLO
SE #[ :: IF D=[ THEN CLOSE #
] !097
339 !@P+ !062
340 GOTO 110 !189

```

Here is what happens: each record of the MERGE file is read in, the first two bytes ignored (we don't need the line number) and the rest are added up.

Next the identical record is printed to the output file with the addition of the

"token" for "!" (REMark) and the three characters of the checksum. This will work even if the program line already contains a REMark. The user must not type these characters since, they are not computed into checksum. At the end (it may take a while for a long program, but only needs to be run once), the programmer types NEW and MERGEs in the output file, then SAVES it in the normal mode, or LISTs it to a printer, or whatever.

Now with all that under your belt, I want to introduce you to another program called CHECK. Check is a program I use when I type in programs from MICROpendium or other sources. As I type in each line and press ENTER, the sum is printed under the line number in inverse video. If it is the same number as the one on the print copy, it is OK. If not, retype it and make any needed corrections. This program does not add the CHECKSUM number to the program.

Note from the editor:

The CHECK program is written in ASSEMBLY and I am not able to include it in this article, but you can find it on our May issue of the C.O.N.N.I. DOM under the file name "CHECK^"

END

MAY 15 LIMA MUG CONFERENCE UPDATE
from the April newsletter of the
Lima 99/4A USERS GROUP

As of March 15 the following have scheduled videotape seminars and/or display tables. We expect many additions to this list as Conference time approaches.

SEMINARS:

Tim Bodenmiller "The current status of game programming for the TI"
Mike Wright "The PC-99 emulator"
S & T software title unknown
Mike Maksimik "MIDI update"
Don Malden "New hardware for the Geneve"
Jack Sughrue "The Teaching TI: Our Computer as an Educational Tool"

Bud Mills "The SCSI card and other hardware from BMS"
Barry Traver title unknown

TABLES:

PC emulator
S&T software
Crystal Software
Cecure Electronics
D. Wright Stuff
L.L Conner Enterprises
Competition Computer
MUNCH (self help video tape on console maintenance)
Bud Mills Services
Harrison Software (debuting "The Utilities ACCEPT AT")
Recharged Computers

***** TI-101 *****

by Jack Sughrue
originally appeared
BITS, BITES & PIXELS
Lima, Ohio

TI-101, a series of nine longish articles on the educational use of the 99/4a for learners of all ages, is written by Jack Sughrue, author of the NEW-AGE 99 series.

The first of this series is a historical background to the stuff leading up to the ways computers and the TI, in particular, could be used successfully and ingeniously in the classroom or at home.

The other articles begin by exploring holism up to the ways computers can be used in the class. Then I show the variety of ways the TI can be an important part of any child's education (any adults, too) by giving many specific applications through modules, disks, books, newsletters, and tapes and even combinations of all (and not discounting direct applications in the form of programming).

Basically, the series is to make the TI new to new users: our children, grandchildren, nieces & nephews, friends, whoever. But mostly to LEARNERS of all ages (many of whom are new or have forgotten the glories of the "Educational TI" or might not have been aware of all that was and IS available for the learner on our wonderful computer.) Our "Educational President" failed the nation's children badly in that category, but our "Educational Computer" - the TI-99/4A - far exceeded (and still does) any competition. That's what these articles are about: a remarkable, viable, educational tool.

The format of the articles I simply couldn't resist as a teacher. They all take place in a college classroom. The course is "TI-101" and the professor imparts information in preparation for some test in the future. He also assigns lots of homework and notetaking along the way. Each "class" builds upon the previous one, and the "lectures" are titled and numbered: HOLISM is #2, DOCENT #3, ROOTS #4, MCGUFFEY'S #5, etc. (through all eight).

Jack Sughrue

***** TI-101 *****

OUR 4/A UNIVERSITY

by Jack Sughrue
Box 459
E. Douglas MA 01516

#1 Corpus
Historical Perspective

First, Class, if you'd look at the screen. This overhead shows the brain. Yes, Ms. Bronte, the human brain.

This is the corpus calosum, that wonderful band of billions

of nerve fibers connecting the hemispheres of the brain. Forty years ago that band was surgically severed to contain grand mal seizures in epileptic patients.

That was the beginning of a profound revolution in education that is quietly (though, at times, quite noisily) continuing through today.

Through the massive research done since that fateful slice, we, as a society, have learned more in the past quarter century about how people learn than we knew about the subject in all the tens of centuries humans have considered the process.

This educational revolution was not without its prophets. John Dewey was one. Today there are many great teachers out there operating under the umbrellas of "Process Learning," "Open Methodologies," "Whole-brain Teaching," "Open Classrooms," "Science/Logic Approach," and piles of other names, including "Whole Language." The last is probably having the most profound influence on the real education in the English-speaking World as any philosophical approach since Horace Mann "Mandated" public education in America so long ago. (So long ago that we take free, public education for all as a given, as an inalienable right.)

But there is a problem, Class.

(Isn't there always?)

When the Germans first devised an efficient way of organizing a mass education in the 19th Century, they decided to make a step-by-step system of completing a given body of work at a given chronological year of a child's life. Thus, 6-year-olds go through a first grade (and an artificially-created, adult-generated curriculum). After completing this predetermined set of tasks, the child turns seven and, if lucky, moves into the second grade where another set of artificial goals awaits HIM (no girls, of course).

Aren't you glad America has no sexist or racist bias these days?

The fact that 7-year-olds are not developmentally on the exact step at any time (any more than all the 47-year-olds are) made no difference to the people operating this 19th Century system. In order to protect the system, an achievement hierarchy was developed, which has come down to us, unfortunately, even to today in too many schools. It is a system that never worked because it created an invisible - though profound - class system. The system created a society of elitists, of average Dicks and Janes, of losers. The basal reader system (unfortunately still in place in most American schools) requires that the classroom be divided into three groups: the good readers, the average readers, the poor readers (sometimes called Bluebirds, Robins, and Snowy Egrets or Red-crested Flaminglers or whatever). But you know and I know that those groups, begun in kindergarten and carried all through elementary school, created what are perceived as the smart snobs, the struggling middle class, and the dumb (and bad) kids. By the time official tracking takes place in junior high (middle school) the system is firmly in place. You'll never guess which group has the greatest number of dropouts or which group has the greatest number of kids who go on to advanced degrees (followed by the best jobs). These determinations for the most part are made in the primary grades in elementary school.

The same 19th-Century system also created a hierarchy of adults. Prior to the institutionalization of education the teacher was the most important adult in the learning process. After the system overtook the world, administrators became the

NEXT PAGE

most important part of the system. This is usually followed by the operational staff. (Go into ANY school and see if that institution operates around the things that secretaries and custodians require before all else or whether the teachers get top priority. Surprise!)

Anyway, Class, in this topsy-turvy setup, highly-paid administrators make the decisions. These decisions (from administrators operating in an entirely separate building from a school, believe it or not) are then handed down to other administrators who have offices and secretaries. The decisions are then handed down to administrators who are in schools (principals, which means, by the way "first or highest in rank and importance"). In secondary schools these decisions are usually then handed down to department heads. Then - possibly - the teachers are told. These are the same teachers who administrators love to hold "accountable," even though they have been excluded from the decision making. Doesn't this "accountability without authority" have a bit of the ring of "taxation without representation" about it?

Generally speaking, administrators - who have the most opportunity and time to learn about all the masses of research on how children learn - know the least. They are divorced from the youngsters and from the realities of day-to-day education. They don't realize, for example, that the clientele has changed. That the students today are not made the same way, intellectually and emotionally and socially, that youngsters 25 years ago were. That the horrors of nuclear war, AIDS, street violence, fanatic consumerism, drugs, and so on were not part of our growing up, of our everyday consciousness and reality. That when I was growing up the attention span of youngsters in ELEMENTARY SCHOOL was estimated to be a little over an hour; that seven years ago for students in K-12 it was 22 minutes; that last year for that same group it was 10.8 minutes!

And education is a big - a humongous! - business. Publishers determine the curriculum in America and sell their goods to administrators who foist these materials upon the trained classroom professionals. This is a multi-billion dollar business and one that stomps out any attempt at teacher input for better ways of doing things in the classroom. Such changes may cause these influential profiteers to lose money; influential bureaucrats to lose power.

Millions of Americans sense (even if they don't have statistics at hand) that something is drastically wrong with schools that still use 19th-Century methods and materials to teach 21st-Century life skills and that still put profits and political power (inside and outside the schools) ahead of the education of our children. These parents and other friends of public education are afraid for America, for the Earth. For all our children.

Some parents (former Bluebirds) have the lucky financial fortune to put their children into expensive private schools. Others have sought to find some solace and protection from the

outside world by placing their youngsters in religious schools where they hope their own values will be inculcated. Others, who have the trained academic and intellectual background (like Barry Traver) teach their children at home. The vast majority of us parents are, however, just working class stiffs who want and expect public education to do its job by our kids.

But, wait a minute, my young scholars!

Aren't we the same society that put a man on the Moon just because Jack Kennedy set us that national goal? Didn't we (not England, not Chile, not Russia, not China, not Iraq) send those Voyager spacecraft out into the wilderness of our Solar System? Aren't we the country with the most Nobel winners?

But those achievements all stemmed from a society that prized education. Weren't these and most of the other masterful achievements of our nation developed during a high level of caring for our youngsters (our future), and of developing a liberal climate of risk-taking and experimentation?

What has happened since Nixon's Presidency to change all this? In spite of the lip service given to education by our recent Presidents, the State of the Union, educationally, has regressed catastrophically following the Kennedy/Johnson Era. And, because federal and state programs to assist and enhance the education of our nation's greatest resource - it's children - has virtually dried up and property taxes are the primary source of funding education, teacher bashing has become a national pastime. Blaming the teachers (the lower paid members of the staff who are not allowed to make important educational decisions nor even to give input in most cases) is like blaming the production line worker for the stupid concepts American car manufacturers have been promulgating. As a matter of fact, it is an interesting solution on the part of these rich conservatives to save American business (and, thus, America) by laying off the workers, as if they in some way were to blame for the decision-makers' gross and blatant stupidities.

That, of course, is another story, Class.

There is a revolution happening in American education, and it will prove to be the saving of our nation. This revolution has many names and takes many forms, but it has a commonality: holism. It's an idea whose time is long overdue, and your TI has its place in this scheme of things. We'll begin to look at those next time in TI-101.

Meanwhile, Class, for your homework I'd like you to type in any program from any source on your TI. No, it doesn't have to be an educational program, but it must be a minimum of 20 lines and work when you bring it to class next time.

Ciao!

END

Time is money, spend it wisely. "

**MEETING DATES
FOR
1993**

C.O.N.N.I. BOARD MEMBERS

3RD SATURDAY

--15 MAY 1993--(NO MEETING
19 JUN 1993 LIMA FAIR)
17 JUL 1993
21 AUG 1993
18 SEP 1993
16 OCT 1993
20 NOV 1993
18 DEC 1993

Pres. - John Parkins	614/891-4965
Treas - Bill Sheppard	614/881-5742
Sec/Sat - Jim Peterson	614/235-3545
Sec/Wed - Dick Beery	614/459-3597
Membership - Everett Wade	614/262-6346
Librarian - Ken Marshall	614/876-1670
Disk - Dick Beery	614/459-3597
Cassette - Everett Wade	614/262-6346
Cartridge - Ken Marshall	614/876-1670
NL Exchange - Jean Hall	614/885-4223
TIABS BBS	014/851-0708
Vice Pres. - Chuck Grimes	614/268-8821
Spirit of 99 BBS	614/263-3412
Irwin Hott	614/263-5319
Dick Beery	614/459-3597
Co-Editors/Spirit of 99 Newsletter:	
Jean Hall	614/885-4223
Bob DeVilbiss	614/891-0566

4TH WEDNESDAY

26 MAY 1993
23 JUN 1993
28 JUL 1993
18 AUG 1993

C.O.N.N.I.
181 HEISCHMAN AVE
WORTHINGTON, OH 43085

