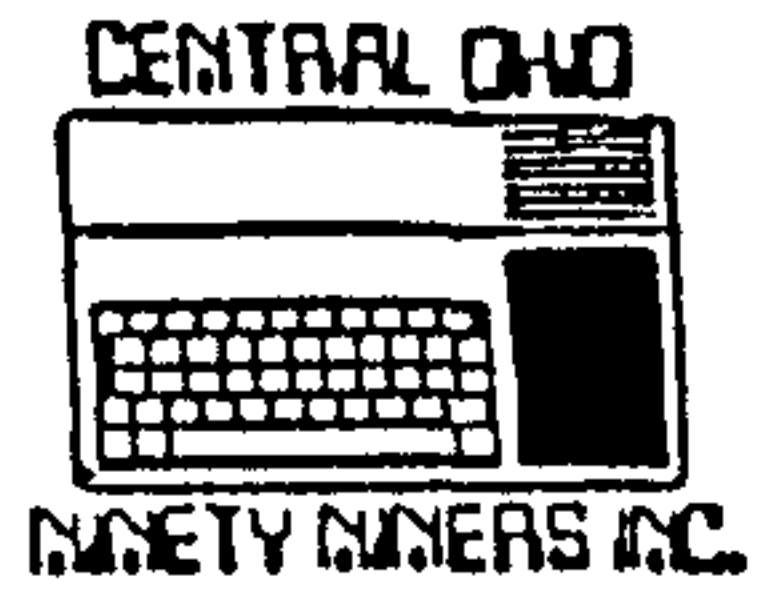


Spirit of 99



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

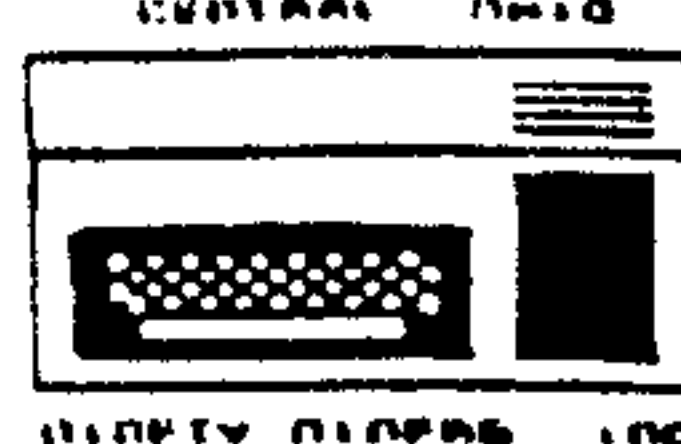
PUBLISHED MONTHLY IN COLUMBUS OHIO

HALLOWEEN



Spirit of 99

THE OFFICIAL NEWSLETTER OF CENTRAL OHIO NINETY-NINERS



COPYRIGHT © 1985
Central Ohio Ninety-Niners Incorporated (C.O.N.N.I.), Columbus Ohio 43212, USA. All rights reserved. Spirit of 99 is published monthly for Central Ohio Ninety-Niners Inc. by C.O.N.N.I. members and is the official newsletter of C.O.N.N.I. User Group.

Editorial, advertising and subscription address is:

181 HEISCHMAN AVE
WORTHINGTON, OH 43085

Subscription rate (USA) \$30.00 /1 year for newsletter and Disk of the Month: \$15.00/1 year, newsletter only. Third class postage paid at Columbus, Ohio.

CHANGE OF ADDRESS: Send both OLD and NEW address to:

Subscription address above. WE assume no responsibility for manuscripts, programs (tape or disk) not accompanied by return postage. Letters to the Editor become property of Spirit of 99. If published, we reserve the right to edit at our discretion.

OPINIONS EXPRESSED HEREIN ARE THE AUTHORS AND ARE BASED ON VALID DOCUMENTABLE RESEARCH. THEY DO NOT

NECESSARILY REFLECT THE OPINIONS OF THE PUBLISHER.

We will not knowingly publish copyright material without the permission of the author and credit due.

All programs published 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 Ninety-Niners Inc. is a non profit organization comprised of MEMBERS who own or use the TI99/4A computer and it's related products and have paid a yearly membership fee of \$28.00 and whose main objective is the exchange of Educational and Scientific information for the purpose of computer literacy.

C.O.N.N.I. meetings are held the 3rd Saturday of each month at Chemical Abstracts, 2540 Olentangy River Road Columbus, OH. Meeting time is 8:30 AM til 2:30PM. Meetings are open to the public. Membership dues (\$28.00) are payable yearly to C.O.N.N.I. and cover the immed-

iate family of the member. (An application has been placed

in this newsletter for your convenience) Please address it to:

Harley Ryan Jr
4178 Chandler Drive
Whitehall, OH 43213

ADVERTISEMENT:

We do accept commercial advertisement at The following rates:
Business Card(2x3.5):

\$5.00/issue

1/4 Page: \$25.00

1/2 Page: \$45.00

Full Page: \$75.00

Write this newsletter for other size arrangements.

All ads should be submitted (camera ready) to advertising address above, payment enclosed. Members ads are published at no cost. (Limit of 25 words and must not be commercial please.)

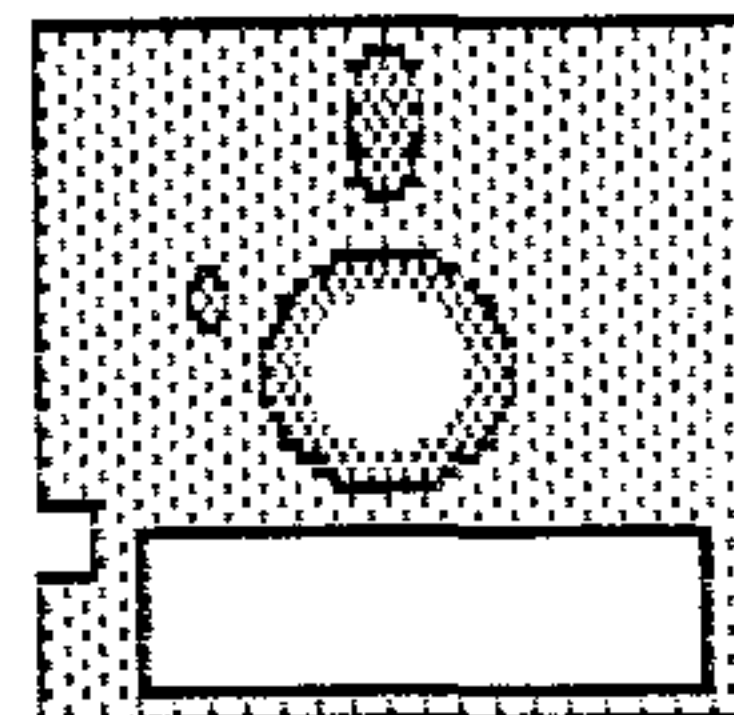
** INDEX **

	PAGE
ANNOUNCEMENTS.....	03
ARCHIVING-A HEADACHE....	12
C.O.N.N.I. MINUTES.....	04
FROM THE TEACHER'S DESK.	14
FROM THE EDITOR.....	19
MAKING FLASH CARDS.....	11
MEETING SCHEDULE.....	05
NEED GRAPHIC PAPER?.....	15
PUTTING IT ALL TOGETHER.	08
RANDOM ACCESS.....	
TO MY MEMORY.....	07
TI BITS.....	16
TIPS FROM TIGERCUB.....	09
TI WORLD NEWS.....	06

DUES ANNOUNCEMENT

Dues are usually paid at or before the March meeting, and are \$28 per year for full membership, library and voting privileges, plus the newsletter. You may also pay your dues in two installments if desired: \$14 in March and \$14 in September. If only the newsletter is desired, then payment is \$15 per year. Those who join during other months of the year pay a lesser, pro-rated amount:

MAR---28.00	APR---25.75	MAY---23.50	JUN---21.00	JUL---18.75
AUG---16.50	SEP---14.00	OCT---11.25	NOV---9.50	DEC---7.00
JAN---4.75	FEB---2.50			



**A DEAL
YOU CAN'T
PASS UP!**

SUBSCRIBE TO BOTH!

*Now you can have the best of both worlds--
Keep up to date on the latest news from
the TI-99/4A world with a subscription to
the Spirit of 99 Newsletter AND get an
up-to-date collection of new public domain
and shareware programs with the Disk of
the Month--both brought to you by the
Central Ohio Ninety-Niners, Inc.*

SUBSCRIPTION RATES

Newsletter only----\$15/yr. (Continental U.S.)
 \$25/yr. (Outside Continental U.S.)
Newsletter PLUS----\$30/yr. (Continental U.S. EXCEPT
DISK of the MONTH Delaware, Fairfiled, Franklin,
 Licking, Madison, Pickaway
 and Union Counties, Ohio)
 \$40/yr. (Outside Continental U.S.)
CONNI Club \$28/yr (see above information)
membership

CONTACT

HARLEY RYAN, Membership
Central Ohio Ninety-Niners, Inc
4178 Chandler Dr, Whitehall, OH 43213
(614) 231-1497

C.O.N.N.I. MINUTES

Wednesday, August 28, 1991.

President Grimes opened the meeting at about 8:20p.m. A small group was present. Everett Wade presented the treasurer's report, which was accepted. Chuck Grimes described the files on the August D.O.M. He reviewed the items that had been covered in the August Saturday meeting. An article appearing in the Snuglet newsletter for July, 1991 was discussed. The thrust of the article was that implementation of our Clearing House BBS has taken too long. It was suggested that SNUG will use the services of Genie, CompuServe or Delphi to satisfy the same need, and that other user groups should do the same.

Chuck and Irwin Hott gave a progress report on the Clearing House. It was asserted that it should be up and running in a week and a half. Also discussed were an article on Reweaving by Wayne Garrison in the St. Louis Computer Bridge for Aug 91 and an article by Jan Knapp in the same issue on "Women and Computing".

Dick Beery raised a question about a printer problem where the head groans and does not move. Suggested solution: change the ribbon. (Note: it worked!). The whereabouts of videocassettes ordered at the Lima Fair was discussed. They have not yet reached us. Chuck will call Charles Good regarding status.

It was announced that Irwin Hott will take over the disk library. He will need to number all the disks and will need help. No volunteers were forthcoming. Harley Ryan offered to catalog the cassette library for Everett Wade. Several members commented on the beauty of some of the older programs.

The business meeting was adjourned and was followed by these demonstrations: Music Maker by RAY Francies, with Strangers in the Night and Mona Lisa as illustrations; a programmable calculator on the 4A by Jim Peterson; and Bob DeVilbiss' on cataloging disks and labeling files via the Formatter, using Ti-Writer or Funnelweb. General discussion followed. The evening ended at about 10:45 p.m.

Respectfully submitted,

Dick Beery

Wednesday secretary.

Saturday, September 21, 1991.

Attendance was poor but we welcomed a visitor from far away, Ben Takach from the TISHUG user group in Sydney, Australia.

President Chuck Grimes conducted the business meeting, and treasurer Everett Wade gave the reasurer's report. SYSOP Irwin Hott reported on the status of the Clearinghouse BBS project. It is hoped to have the hardware tested and running within a week, but the software is not yet completed.

Chuck described the contents of the Disk of the Month. We currently have 95 members, of whom 53 are out-of-town subscribers to the newsletter and disk of the month.

There was no planned program but we discussed many subjects including the MIDI interface, the Chicago Faire, TIM, the disappearance of JP Software and unavailability of PC-Transfer, Al Beard's projected math processor card, Barry Boone's sound digitizer, etc.

Respectfully submitted,

Jim Peterson

Saturday secretary.

Wednesday, September 25th

The meeting started a little after 8:30 p.m. The only officers present, the Wednesday and Saturday secretaries, conducted the meeting with a small group. There was no equipment and there were no demonstrations, but group participation in discussion was lively and helpful.

In the absence of a planned agenda, the group arranged for several future demos, made suggestions for club activities during the Fall months, and suggested restoration of the meeting agenda to the newsletter.

Classes are to be restored to the Fall schedule, and the October meeting will be set up much like some earlier meetings.

Minutes for the Aug. 28th meeting were read and approved. Through oversight, no treasurer's report was given, although the treasurer was present.

An Assembly-language S.I.G. will be started at the Saturday meetings on a trial two-month basis, to be continued if successful.


Demonstrations to be held in the next two months include: Neatlist, Tips Manipulator, XBDetective, and configuring Funnelweb. To save Irwin Hott more time for restructuring and managing the soon-to-be-implemented Clearing House, Dick Beery agreed to take over the disk library. He will be supported by Harley Ryan and Bob DeVilbiss. It was likewise suggested that Harley Ryan spend the time he had planned to use in cataloging the cassette library by helping with the disk library catalog. The meeting adjourned shortly after 9:30p.m.

Respectfully submitted,

Dick Beery

Wednesday secretary

October 1991

SUN	MON	TUE	WED	THU	FRI	SAT
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17		19
20	21	22	23	24	25	26
27	28	29	30	31		

SATURDAY MEETING 19 OCT 1991
Chemical Abstracts Building -- Columbus

8:30AM Setup, coffee and doughnuts

9:00AM Disk of Month,
Micropenduim,
Beginners help
Libraries open

11:15AM Demos:
Dick Beery
TIPS Manipulator

9:45AM TI-Artist class

12:15PM Lunch
Question and
Answer Period

10:30AM Business
Meeting

1:15PM Assembly class

2:30PM Departure

WEDNESDAY MEETING -- 23 OCT 1991
McDONALD'S -- Cleveland and Kain -- Westerville

Demos TI-Writer configuration
by Harold Timmons and Mike Ballman

TI WORLD NEWS, OCTOBER 1991

Compiled by Jim Peterson

Mark Schafer of the Bluegrass 99ers has written a Defragmenter program to defragmentize a disk on the disk itself, rather than copying files from one disk to another. It is available from him for \$7, or \$6 and a disk, at 539 Whitaker Street, Morehead KY 40351.

Bill Gaskill's newsletter for TI-Base users has ceased publication with Volume 2 No. 4. Bill says that refunds have been sent to all subscribers and the newsletter is no longer available; neither are any other products offered by PRK DataBasics and Junction Softworks.

Tony McGovern has released Version 4.4 of Funnelweb.

Texaments has released Fonts, Frames Fun, a 3-disk package of 11 fonts, 39 frame borders and 22 instances for use with TI-Artist and TI-Artist Plus, for \$12.95; and The Missing Link Font Pak, with 29 screen fonts for use with The Missing Link, priced at \$7.95. The price of the TI-Artist Companion 2-disk sets numbered 2 through 13 has been reduced from \$9.95 to \$7.95. The address is 53 Center Street, Patchogue NY 11772, phone (516) 475-3480 or BBS (516) 475-6463.

According to an article in the West Penn newsletter, Tony Lewis and Al Beard have been collaborating for over a year on a math coprocessor card for the P-box, which is now functional although not 100% complete. It uses the Motorola 68881 math coprocessor chip and is expected to be accessible from the TI-99/4A, the Geneve, or the 99105 accelerator card.

The Hunter Valley 99ers User Group of New South Wales, Australia has disbanded. The Nittany Users of Texas Instruments, in Pennsylvania, has discontinued regular meetings but will continue to get together occasionally.

Bruce Harrison of Harrison Software has finished an assembly game called Scudbusters, based on the shooting down of Iraqi Scuds by Patriot missiles. It will be available at the Chicago Faire. He has also written, and released to the public domain, a diskfull of random number generating routines in assembly.

Tigercub Software is now offering a free bonus disk with each ten TI-PD disks purchased.

Barry Boone has written a program to convert digitized sound from PC format to TI/Geneve format, and another program to play the sound on the TI/Geneve. The sounds are reported to be incredibly realistic, but they also apparently use up an incredible amount of memory. The programs will become available through Texaments.

RANDOM ACCESS TO MY MEMORY BANKS

by Jim Peterson

The 99105 Accelerator, to be priced at \$250, will speed up the TI by a factor of 5; in combination with the PEB RAM interface card, to be priced at \$90, the increase will be 10-fold. These are remarkable devices, and I am sure the hardware hackers will love them - but do the computer USERS really need them?

Personally, every program I'm apt to use is on my Ramdisk, for almost instant access. Prescanning has eliminated those initialization delays. The programs generally run as fast as I want them to.

I know that Stephen Shaw over in England is writing graphics plotting routines that take hours and hours to run, but that is the kind of thing you boot up and start just before you go to bed; nothing is going to speed it up so much that you would want to sit and wait on it.

Yes, sorting routines are a bit slow, even in assembly, but the TI just doesn't have the memory to sort any sizeable amount of data anyway. TI-SORT gets around that limitation, to some extent, by writing and reading a scratch file on disk - but disk access is necessarily slow. Other chores you might like to speed up, such as archiving, are slow because of disk access. I know, I know - the hard drive overcomes that limitation. Sorry, my time is valuable but not THAT valuable!

And maybe you would like to speed up Multiplan, or TI-Base. Here again, the TI is handicapped by lack of memory. It was designed as a HOME computer, not a business computer. I have only two projects I would like to put in a database, and both are too large to be practical on a TI. If I ever get around to working on them, I'll put my money into a second-hand PC (pardon the word) rather than add more hardware to my little home computer.

But if you absolutely have to speed things up, maybe there are cheaper ways. For years, there have been references to changing a crystal, costing about \$4.50, in the console. Finally, in the Hoosier newsletter of August 1991 I ran across an article downloaded from DELPHI, in which Jesse Slicer explains exactly how to do it. He says you will gain an increase of 19.3% in microprocessor speed, but you will experience problems with terminal programs and graphics-intensive programs. So, he tells how to install the new crystal in addition to the old one, with a switch to alternate between them.

In the same newsletter are two other DELPHI downloads. One is a step-by-step description of Mike Ballman's method of adding 32k of RAM memory on the 16 bit bus, for a speed increase of about 50%. It requires two chips costing about \$13 and a couple of other doohinkies, price not mentioned.

The other file describes a modification to this method, by the late John Guion. Now, all of this is complete Greek to a technoklutz like me, but if some hardware hacker was to start buying up \$25 consoles, making those modifications and reselling them at a reasonable price, I might be interested.

Perhaps a better way to speed things up is through the software rather than the hardware. A few years ago, RYTE Data was marketing a so-called Basic compiler. It wasn't really that, but it did speed up XBasic programs quite drastically. Unfortunately it had so many limitations and created so many bugs that it was worthless. Last year, someone on the East Coast announced that they were writing a true Basic compiler, but I have heard no more about it. I believe that J. Peter Hoddie and Paul Charlton examined that idea previously, and decided that it was impossible within the memory limitations of the TI.

Ideally, programs should be written in assembly for maximum speed and efficiency - as Mike Ballman would be quick to point out. Trouble is, there are very few programmers with his skill, and very few programs being written in assembly.

But actually, XBasic is a perfectly satisfactory programming language, easy to write and easy to modify, and can be fast enough for any purpose if it is supplemented by CALL LINKs to assembly to do whatever XBasic can't do fast enough or can't do at all. Some of the best programmers have gone that route, but most of us don't know how to write those assembly routines to link to - we need someone to do it for us. Some extremely useful routines have been written for us by such skilled programmers as Karl Romstedt, Barry Traver, Bruce Harrison, Bud Wright, Ross Mudie and others. We need more of them.

Just think. If, back in 1984 when I put out my first 100 Nuts Bolts XBasic subprograms, some assembly genius had started putting out disks full of assembly subprograms. What a powerful programming language we would have by now, and what wonderful programs might have been written!

PUTTING IT ALL TOGETHER No. 10

by Jim Peterson

The hardest part of learning to program is not in learning what the various commands do - it is in learning how to put them all together to do what you want them to do!

Key in this simple routine, and run it, to see what it does. Then read the explanation of each line and see how they do what they do.

```
100 CALL CLEAR
110 INPUT "KNOWN VALUE? ":C
120 X=1
130 GOSUB 180
140 IF A<C THEN Y=X :: X=X*2
   :: GOSUB 180 :: GOTO 140 EL
SE 160
150 IF A>C THEN Y=X :: X=X/2
   :: GOSUB 180 :: GOTO 150
160 Z=(ABS(X-Y))/2 :: Y=X ::
  IF A<C THEN X=X+Z ELSE X=X-
Z
170 GOSUB 180 :: GOTO 160
180 A=X X/X
190 PRINT X;:: IF A=C OR A=B
  THEN STOP ELSE B=A :: RETUR
N
```

Equations such as the one in line 180 are very difficult to solve mathematically, but the computer can solve them quickly by systematic trial and error. You can substitute any other equation of one unknown value, using A for the known value and X for the unknown.

Line 110 gets the known value in C. Line 120 gives X a starting value of 1. Actually we could start with any value except 0, but we must give X some value or the computer will take it to be zero.

Execution jumps to line 180 to try this value in the equation. Line 190 prints the value of X, just so we can see what is going on, and then checks to see if the value obtained for A is the value we specified for C; in this case the problem is solved and execution stops. We will explain the B later. If not solved, execu-

tion returns to 130 and then to 140.

If $A < C$, meaning A is less than C, the value we received is too small, so X must be too small. In this case we save the value of X in Y, then double the value of X, GOSUB to the equation and check in 190 whether we have solved it. If not, we return to 140, jump back to the beginning of 140, see if A is still smaller than C and, if so, repeat the process again. Eventually the doubling of X will cause A to be more than C, if we do not hit it exactly in 190, and the ELSE 160 jumps us to that line.

However, if the first value we obtained for A was more than C, execution would drop right through line 140 to line 150. Here we would do exactly the same thing except that we would halve the value of X each time until A became less than C.

In either case, we end up in line 160 for the final step. Now here is why we were saving the previous value of X in Y each time. We subtract Y, the previous value, from X, the current value. Either one may be larger than the other, so we may get a negative value. ABS changes the negative to a positive. We divide the result by 2, and give that value to Z. If A is less than C we add Z to X, otherwise we subtract it. Then we GOSUB to 180 repeatedly until 190 finds that we have found the answer. Each time around the difference between Y and X will be half of what it was the previous time, and the result of the equation will alternate between $A > C$ and $A < C$ until it finally centers on the exact value.

Now, the reason for the B in line 190. The true value of X will usually run out to more decimal places than even our 16-bit TI-99/4A can handle and, since the exact full value cannot be reached, the program would go on forever. So, we save the value of A in B each time, and check each time to see if the next value received for A is that same value. If so, we have reached the limit of computer accuracy, so we stop. The same number will probably appear on screen several times at the end, because the screen shows only 10 digits but the computer continues calculating to 13 or 14 digits.

No. 66

Tigercub Software
156 Collingwood 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 well over 500 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 #5 and the latest supplement is available for \$1 which is deductible from the first order.

In Tips #65 I said that the TI could calculate to 14-digit accuracy, rather than the 8-digit accuracy of a PC. Actually the number in memory is calculated to 13- or 14-digit accuracy, depending on the number, but is rounded to 10 digits on the screen display, or shown in exponential notation if the number is extremely large or extremely small. If you want to see the complete number, this routine will show the normal screen display and the full number in memory. To see the complete range of numbers our little TI can handle, try inputting -9.999999999999999E127 and -1.000000000000000E-128 and 1.000000000000000E-128 and 9.999999999999999E127.

```
100 OPEN #1:"DSK1.INT2",INTERNAL,RELATIVE,UPDATE
110 INPUT N
120 PRINT #1,REC 1:N
130 INPUT #1,REC 1:N$
140 X=ASC(SEG$(N$,1,1)):: Y=ASC(SEG$(N$,2,1)):: IF Y>99 THEN Y=256-Y :: N$=SEG$(N$,1,1)&CHR$(Y)&SEG$(N$,3,255)
150 FOR J=2 TO LEN(N$):: X$=STR$(ASC(SEG$(N$,J,1)))
160 IF ASC(SEG$(N$,J,1))<10 THEN X$="0"&X$
170 P$=P$&X$ :: NEXT J
180 IF X<63 THEN Y$=".&RPT$( "00",63-X)&P$ :: GOTO 230
190 IF X>191 THEN Y$=".&RPT$( "00",X-192)&P$ :: GOTO 230
200 IF X>185 THEN Y$=SEG$(P$,1,14-(X-185)*2)& "."&SEG$(P$,14-(X-185)*2+1,255):: GOTO 230
210 Y$=SEG$(P$,1,(X-63)*2)& "."&SEG$(P$, (X-63)*2+1,255)
220 IF ASC(Y$)=48 THEN Y$=SEG$(Y$,2,255)
230 IF N<0 THEN Y$="-"&Y$
240 PRINT TAB(2);N :: PRINT TAB(3);Y$ :: P$="" :: GOTO 10
```

But even the smart little TI has its limits. Try this- $X=2/3-1/3-1/3$:: PRINT X. See the TI User's Reference Guide page III-13 for the explanation of all this.

Solving an equation such as $X^X/X-X$ would be very difficult to solve by mathematical means, but our computer can find the answer quickly by systematic trial and error, to the 14-point limit of its accuracy.

```
100 DISPLAY AT(3,1)ERASE ALL
:"This program will solve even the most difficult equations with one variable."
110 DISPLAY AT(6,1):"Put your own equation in line 210, using A for the known value and X for the unknown value."
120 DISPLAY AT(24,6):"PRESS ANY KEY" :: DISPLAY AT(24,6):"press any key" :: CALL KEY(0,K,S):: IF S=0 THEN 120
```

```
130 CALL CLEAR
140 DISPLAY AT(8,1):"KNOWN VALUE? " :: ACCEPT AT(8,14):C
150 X=1 :: DISPLAY AT(12,1):""
160 GOSUB 210
170 IF A<C THEN Y=X :: X=X#2 :: GOSUB 210 :: GOTO 170 ELSE 190
180 IF A>C THEN Y=X :: X=X/2 :: GOSUB 210 :: GOTO 180
190 Z=(ABS(X-Y))/2 :: Y=X :: IF A<C THEN X=X+Z ELSE X=X-Z
200 GOSUB 210 :: GOTO 190
210 A=X^X/X-X/2
220 IF A<C THEN DISPLAY AT(11,5):X ELSE IF A>C THEN DISPLAY AT(13,5):X
230 IF A=C OR A=B THEN DISPLAY AT(12,5)ERASE ALL:X :: GOTO 140 ELSE B=A :: RETURN
```

In Recreational and Educational Computing, published 8 times a year at 909 Violet Terrace, Clarks Summit PA 18411, \$27 per year, I found a neat routine to find the greatest common divisor and least common multiple of any two numbers - so I converted it to TI Basic and modified to do the same with multiple numbers.

```
100 CALL CLEAR :: PRINT "PROGRAM TO FIND THE GREATEST COMMON DIVISOR AND LEAST COMMON MULTIPLE OF ANY NUMBER OF NUMBERS.":
110 DIM N(100)
120 PRINT "INPUT ZERO WHEN FINISHED":
130 T=T+1 :: INPUT "NUMBER "&STR$(T)&"? ":N(T):: IF N(T)=0 THEN 140 ELSE IF N(T)<>INT(N(T))OR N(T)<1 THEN T=T-1 :: GOTO 130 ELSE 130
140 AA=N(1):: GCD=N(2)
150 GOSUB 170 :: FOR J=3 TO T-1 :: AA=N(J):: GCD=ABS(GCD):GOSUB 170 :: NEXT J
160 GOTO 210
170 R=AA-INT(AA/(GCD+ABS(GCD=0)))&GCD
180 IF R<2 THEN GCD=R+GCD*(1-R):: GCD=GCD&ABS(GCD>0)+ABS(GCD=0):: GOTO 200
190 AA=GCD :: GCD=R :: GOTO
```

```
170
200 RETURN
210 PRINT "THE GREATEST COMMON DIVISOR OF YOUR";T-1;"NUMBERS IS";GCD
220 L=N(1)*N(2)/GCD :: FOR J=3 TO T-1
230 IF L/N(J)<>INT(L/N(J))THEN L=L*N(J)
240 NEXT J
250 LL=L/2 :: FOR J=1 TO T-1 :: IF LL/N(J)<>INT(LL/N(J)) THEN J=T-1 :: GOTO 270
260 NEXT J :: L=LL :: GOTO 250
270 PRINT "AND THE LOWEST COMMON MULTIPLE IS";L
```

Joy Warner called from the L.A. group, and mentioned that it would be nice to have a program to print out a page of math problems, and a page of answers. So here is one that will randomly create any number of either addition or subtraction problems, within any specified range of numbers, and output the desired number of copies to a printer in two columns of expanded print, numbered in sequence, plus a numbered answer sheet to make it easy for the teacher.

```
100 DISPLAY AT(1,4)ERASE ALL:"MATH PROBLEM PRINTER" !by Jim Peterson
110 DIM A(200),H(200),L(200)
:: OPEN #1:"PIO" :: PRINT #1:CHR$(27)&"@"&CHR$(27)&"W"&CHR$(1);
120 M$(1)="ADDITION" :: M$(2)="SUBTRACTION" :: D$(1)="+" :: D$(2)="-" :: ON$=CHR$(27)&"-"&CHR$(1):: OFF$=CHR$(27)&"-"&CHR$(0)
130 DISPLAY AT(3,1):"Do you want?":":":1. "&M$(1):"2. "&M$(2):: ACCEPT AT(3,14)VALIDATE("12")SIZE(1)BEEP:C
140 DISPLAY AT(8,1):"Range of numbers?":"From":"To" :: ACCEPT AT(9,6)VALIDATE(DIGIT)BEEP:LN :: ACCEPT AT(10,4)VALIDATE(DIGIT)BEEP:HN :: IF LN>=HN THEN 140 ELSE HN=HN-LN
150 DISPLAY AT(13,1):"How many problems?" :: ACCEPT AT(1
```

```

3,20)VALIDATE(DIGIT)BEEP:P
160 DISPLAY AT(15,1):"How ma
ny copies?" :: ACCEPT AT(15,
18)VALIDATE(DIGIT)BEEP:CC
170 FOR J=1 TO P :: GOSUB 29
0 :: H(J)=N1 :: L(J)=N2
180 IF C=1 THEN A(J)=H(J)+L(
J)ELSE A(J)=H(J)-L(J)
190 NEXT J
200 FOR J=1 TO CC :: GOSUB 3
10 :: FOR K=1 TO P STEP 2
210 T1$=STR$(K)&". "&STR$(
H(K)):: T2$=STR$(K+1)&".
"&STR$(H(K+1))
220 PRINT #1:TAB(15-LEN(T1$
));T1$;TAB(35-LEN(T2$));T2$
230 T1$=D$(C)&STR$(L(K)):: T
2$=D$(C)&STR$(L(K+1))
240 PRINT #1:TAB(15-LEN(T1$
));ON$&T1$&OFF$&RPT$(" ",20-L
EN(T2$))&ON$&T2$&OFF$
250 PRINT #1:"":"":" " :: IF
K/19=INT(K/19)THEN PRINT #1:
CHR$(12);
260 NEXT K :: PRINT #1:CHR$(
12);:: NEXT J
270 PRINT #1:TAB(16);"ANSWER
S":"":" "
280 FOR J=1 TO P STEP 2 :: P
RINT #1:TAB(6);STR$(J)&". "
;A(J);TAB(26);STR$(J+1)&".
";A(J+1):: NEXT J :: STOP
290 RANDOMIZE :: N1=INT(RND*
HN+LN):: N2=INT(RND*HN+LN)::
IF N1=N2 THEN 290
300 IF C=2 AND N2>N1 THEN T=
N2 :: N2=N1 :: N1=T :: RETUR
N ELSE RETURN
310 PRINT #1:" "&M$(C)
&" PROBLEM PRINTER":"":" "
:"" :: RETURN

```

And this one will do the same with multiplication problems.

```

100 DISPLAY AT(1,4)ERASE ALL
:"MULTIPLICATION PROBLEMS":"
PRINTER" !by Jim P
eterson
110 DIM A(200),H(200),L(200)
:: OPEN #1:"PIO" :: PRINT #1
:CHR$(27)&"@&CHR$(27)&"W"&C
HR$(1);
120 ON$=CHR$(27)&"-"&CHR$(1)
:: OFF$=CHR$(27)&"-"&CHR$(0)
130 DISPLAY AT(8,1):"Range o
f multiplicand?":"From":"To"
:: ACCEPT AT(9,6)VALIDATE(D
IGIT)BEEP:L1 :: ACCEPT AT(10

```

```

,4)VALIDATE(DIGIT)BEEP:H1 ::
IF L1>=H1 THEN 130 ELSE H1=
H1-L1
140 DISPLAY AT(12,1):"Range
of multiplier?":"From":"To"
:: ACCEPT AT(13,6)VALIDATE(D
IGIT)BEEP:L2 :: ACCEPT AT(14
,4)VALIDATE(DIGIT)BEEP:H2
150 IF L2>=H2 THEN 140 ELSE
R=LEN(STR$(H2)):: H2=H2-L2
160 DISPLAY AT(16,1):"How ma
ny problems?" :: ACCEPT AT(1
6,20)VALIDATE(DIGIT)BEEP:P
170 DISPLAY AT(18,1):"How ma
ny copies?" :: ACCEPT AT(18,
18)VALIDATE(DIGIT)BEEP:CC
180 FOR J=1 TO P :: GOSUB 31
0 :: H(J)=N1 :: L(J)=N2
190 A(J)=H(J)*L(J)
200 NEXT J
210 FOR J=1 TO CC :: GOSUB 3
20 :: FOR K=1 TO P STEP 2
220 T1$=STR$(K)&". "&STR$(
H(K)):: T2$=STR$(K+1)&".
"&STR$(H(K+1))
230 PRINT #1:TAB(15-LEN(T1$
));T1$;TAB(35-LEN(T2$));T2$
240 T1$="X "&STR$(L(K)):: T2
$="X "&STR$(L(K+1))
250 PRINT #1:TAB(15-LEN(T1$
));ON$&T1$&OFF$&RPT$(" ",20-L
EN(T2$))&ON$&T2$&OFF$
260 FOR S=1 TO R+3 :: PRINT
#1:"" :: NEXT S
270 LC=LC+5+R :: RC=LC+5+R :
: IF RC>=60 AND K<P THEN PRI
NT #1:CHR$(12):: LC=5
280 NEXT K :: PRINT #1:CHR$(
12);:: NEXT J
290 PRINT #1:TAB(16);"ANSWER
S":"":" "
300 FOR J=1 TO P STEP 2 :: P
RINT #1:TAB(3);STR$(J)&". "
;A(J);TAB(23);STR$(J+1)&".
";A(J+1):: NEXT J :: PRINT #
1:CHR$(12):: STOP
310 RANDOMIZE :: N1=INT(RND*
H1+L1):: N2=INT(RND*H2+L2)::
RETURN
320 PRINT #1:" " MULTIP
LICATION PROBLEMS":"":"":" "
:"" :: LC=5 :: RETURN

And division -
100 DISPLAY AT(1,6)ERASE ALL
:"DIVISION PROBLEMS":"
PRINTER" !by Jim Peterso
n
110 DIM A(200,2),H(200),L(20

```

```

0):: OPEN #1:"PIO" :: PRINT
#1:CHR$(27)&"@&CHR$(27)&"W"
&CHR$(1);
120 DISPLAY AT(8,1):"Range o
f dividend?":"From":"To" ::
ACCEPT AT(9,6)VALIDATE(DIGIT
)BEEP:L1 :: ACCEPT AT(10,4)V
ALIDATE(DIGIT)BEEP:H1 :: IF
L1>=H1 THEN 120
130 DISPLAY AT(12,1):"Range
of divisor?":"From":"To" ::
ACCEPT AT(13,6)VALIDATE(DIGI
T)BEEP:L2 :: ACCEPT AT(14,4)
VALIDATE(DIGIT)BEEP:H2
140 IF L2>=H2 THEN 130 ELSE
R=LEN(STR$(INT(H1/H2)))$2 ::
H2=H2-L2 :: H1=H1-L1
150 DISPLAY AT(16,1):"How ma
ny problems?" :: ACCEPT AT(1
6,20)VALIDATE(DIGIT)BEEP:P
160 DISPLAY AT(18,1):"How ma
ny copies?" :: ACCEPT AT(18,
18)VALIDATE(DIGIT)BEEP:CC
170 FOR J=1 TO P :: GOSUB 31
0 :: H(J)=N1 :: L(J)=N2
180 A(J,1)=INT(H(J)/L(J))::
A(J,2)=H(J)-A(J,1)*L(J)
190 NEXT J
200 FOR J=1 TO CC :: GOSUB 3
20 :: FOR K=1 TO P STEP 2
210 LC=LC+1 :: T1$=STR$(K)&".
"&RPT$(" ",LEN(STR$(L(K
))))&RPT$("_",LEN(STR$(H(K)
)))
220 T2$=STR$(K+1)&". "&RP
T$(" ",LEN(STR$(L(K+1))))&RP
T$("_",LEN(STR$(H(K+1))))
230 PRINT #1:TAB(15-LEN(T1$
));T1$;TAB(35-LEN(T2$));T2$
240 T1$=STR$(L(K))&"!"&STR$(
H(K)):: T2$=STR$(L(K+1))&"!"
&STR$(H(K+1))
250 LC=LC+1 :: PRINT #1:TAB(
15-LEN(T1$));T1$;TAB(35-LEN(
T2$));T2$
260 FOR S=1 TO R+3 :: LC=LC+
1 :: PRINT #1:"" :: NEXT S
270 IF 66-LC<5+R AND K<P THE
N PRINT #1:CHR$(12):: LC=5
280 NEXT K :: PRINT #1:CHR$(
12);:: NEXT J
290 PRINT #1:TAB(16);"ANSWER
S":"":" "
300 FOR J=1 TO P :: PRINT #1
:TAB(3);STR$(J)&". ";A(J,1)
;"REMAINDER ";A(J,2):: NEXT
J :: PRINT #1:CHR$(12):: STO
P
310 RANDOMIZE :: N1=INT(RND*
H1+L1):: N2=INT(RND*H2+L2)::

```

```

RETURN
320 PRINT #1:" " DIV
ISION PROBLEMS":"":"":" "
:: LC=5 :: RETURN

Bud Wright wrote this one
for Irwin Hott, so he could
listen to lower case text
with the Speech Synthesizer.
Imbed it with ALSAVE, access
it with CALL LINK("CAPS",A$)
and it will instantly con-
vert any lower case letters
to upper case. I found it
invaluable in writing key-
word search programs.

* CAPS/S BY BUD WRIGHT
* VERSION 1.1 10/17/86
STRREF EQU >2014
STRAS6 EQU >2010
MREG BSS 32
STRBUF BYTE 255
BSS 255
DEF CAPS
CAPS LWPI MREG
CLR R0
LI R1,1
SETO @STRBUF
LI R2,STRBUF
BLWP @STRREF
MOVB @STRBUF,R2
SRL R2,8
JEQ CAPOUT
LI R1,STRBUF+1
CAPS2 MOVB #R1,R3
SRL R3,8
CI R3,96
JGT CAPS1
CAPS3 SWPB R3
MOVB R3,#R1+
DEC R2
JNE CAPS2
CLR R0
LI R1,1
LI R2,STRBUF
BLWP @STRAS6
CAPOUT LWPI >83E0
B @>006A
CAPS1 CI R3,122
JGT CAPS3
AI R3,-32
JMP CAPS3
END

```

Memory full,

Jim Peterson

MAKE YOUR OWN FLASH CARDS

by Tony Falco

Last summer a friend came to me with a programming problem. He wanted a program to display arithmetic flash cards, with any numbers, and problems in text book format. It was not as easy a task as I thought it might be, but the string commands in BASIC's DISPLAY AT and ACCEPT AT commands did the job.

The user picks one of three options. Then he picks his own numbers. Entering <Q> for the first number will end the program. The program works best if the child parent work at the computer together.

In a future article, I will show how to adapt the program so the computer generates the problem.

```

10 CALL CHAR(104,"FF80808080
BO80FFFF0101010101FF")
20 DISPLAY AT(12,6)ERASE ALL
:"PICK ONE-->+-x  +"
30 ACCEPT AT(12,22)SIZE(-1)V
ALIDATE("+-X")BEEP:OP$
40 CALL CLEAR :: CALL FLASH
:: CALL CHAR(95,"0000FFFF")
50 DISPLAY AT(3,4)SIZE(-6)BE
EP:"RIGHT:" :: DISPLAY AT(3,
15)SIZE(-6):"WRONG:"
60 DISPLAY AT(5,10)SIZE(-6):
"SCORE:"
70 FOR J=12 TO 15 :: DISPLAY
AT(J,1)SIZE(-25):" " :: NEX
T J
80 DISPLAY AT(13,7)SIZE(-1)B
EEP:OP$
90 ACCEPT AT(12,9)VALIDATE(D
IGIT,"Q")SIZE(-4):A$
100 IF A$="Q" THEN 280 ELSE
ACCEPT AT(13,9)VALIDATE(DIGI
T)SIZE(-10):B
110 A=VAL(A$)
120 C=-(A+B)*(OP$="+")-(A*B)
*(OP$="X")-(A-B)*(OP$="-")
130 B$=STR$(B):: C$=STR$(C):
:M=MAX(LEN(A$),LEN(B$)):: N
=MAX(M,LEN(C$))
140 DISPLAY AT(13,1)SIZE(-5)
:" " :: DISPLAY AT(13,7-M+LE
N(A$))SIZE(-10):OP$
150 DISPLAY AT(13,9+LEN(A$)-
LEN(B$))SIZE(-10):B$
160 DISPLAY AT(14,8+LEN(A$)-
N)SIZE(-10):RPT$(CHR$(95):N+
2)
170 ACCEPT AT(15,9+LEN(A$)-L
EN(C$))SIZE(-LEN(C$))VALIDAT
E(DIGIT,"-")BEEP:D
180 IF D=C THEN R=R+1 :: CAL
L SAY("#THAT IS RIGHT):: CA
LL DELAY(200):: GOTO 250
190 DISPLAY AT(12,19)SIZE(-L
EN(A$)):A$
200 DISPLAY AT(13,17-M+LEN(A
$))SIZE(-1):OP$
210 DISPLAY AT(13,19+LEN(A$)
-LEN(B$))SIZE(-LEN(B$)):B$
220 DISPLAY AT(14,18+LEN(A$)
-N)SIZE(-9):RPT$(CHR$(95):N+
2)
230 DISPLAY AT(15,19+LEN(A$)
-LEN(C$))SIZE(-LEN(C$)):C$ :
: W=W+1
240 CALL SAY("#THAT IS INCOR
RECT):: CALL DELAY(700)
250 S=INT(100*R/(W+R)+.5)
260 DISPLAY AT(3,10)SIZE(3):
R :: DISPLAY AT(3,21)SIZE(-3
):W
270 DISPLAY AT(5,17)SIZE(-4)
:STR$(B)&"%" :: GOTO 70
280 CALL SAY(STR$(R)):: CALL
SAY("CORRECT AND):: CALL S
AY(STR$(W))
290 CALL SAY("NOT CORRECT"):
: CALL CLEAR :: END
300 SUB DELAY(X):: FOR D=1 T
O X :: NEXT D :: SUBEND
310 SUB FLASH :: CALL SCREEN
(12):: FOR Z=1 TO 8 :: CALL
COLOR(Z,2,15):: NEXT Z
320 CALL COLOR(9,2,2,10,12,1
2):: CALL HCHAR(1,1,104,768)
330 FOR Z=2 TO 6 :: CALL HCH
AR(Z,5,32,22):: NEXT Z
340 FOR Z=10 TO 18 :: CALL H
CHAR(Z,3,32,27):: NEXT Z
350 CALL HCHAR(7,6,96,22)::
CALL VCHAR(3,27,96,4)
360 CALL HCHAR(19,4,96,27)::
CALL VCHAR(11,30,96,8):: SU
BEND

```


ARCHIVING—A HEADACHE?

By: Andy Frueh, Lima UG

A lot of people are puzzled by archiving and how to use Barry Boone's Archiver. What follows is both a reference guide and explanation of Archiver III. It is not meant to totally replace the documentation for this program. Actually, I haven't seen a distribution copy that comes with a set of instructions. There may be hidden features of ArcIII that aren't obvious to me (for example, Disk Utilities by John Birdwell has a feature to figure decimal-to-hex conversions).

What exactly is archiving? Putting it simply, when you archive you take a file or a set of files, and group them as one file then compress them so they take up less disk space. Some software comes archived. These ALMOST always include the archiving program. Examples are Jack Sughrue's PLUS! and the Complete Adventure disk set.

What is the purpose of archiving? Well it started out as a money saver for modem users. It is faster, and thus cheaper, to send 90 archived sectors as 1 file, than 120 sectors for 3 programs. Now it is also a means of backing up disks. You can save each of your disks as a one file, squashed archive. You can specify whether you want compressed files or not. The reason you have a choice is that some unusual files actually take up more space when they are compressed. Another useful application of archiving is when you have programs you want to keep, but don't need ready to use. You can keep archives of all these files instead of taking up disk space.

OK, now that you have the "what", here's the "how". As far as I know, the only archiver is Barry Boone's program. Its operation is completely different from Archiver II. Rather than add new features to past versions, Archiver was completely re-written. It usually contains an XB LOAD program, but may be loaded from E/A. The program's filename is usually ARC1. It can be found on almost all of the bulletin boards, as a commercial version with Geneve utilities, in user group libraries, with other Fairware programs or from the author. Chances are, you can definitely get a copy.

First things first, so get the program loaded. After that, you should see a Fairware notice. Press any key to pass this. You then see a menu. Each menu option is described in detail below.

1) Archive Files - These options are largely self-explanatory. As you may have guessed, this option archives files. Pressing one will deliver a set of prompts. These are "Source Drive (1-Z)". Yes, you can have drive numbered from 1-9 and A-Z. Then comes, "Output Drive (1-Z)". You may use one drive. Archiver will prompt you to change disks when needed. It is highly recommended that you use a blank output disk, since archives may fill or almost fill a disk. Next comes "Output Filename". This is usually the name of the disk you are archiving, or some related heading. For example, a set of D/V 80 articles may be named "ARTICLES". The following prompt is "Pack all Files? (Y/N)". If you answer "Y" then all the files on the source disk are archived. If you answer "N", then when Archiver is working, you are asked "Include filename? (Y/N)". If you answer "Y" then that file is archived, otherwise it is ignored. This is a handy feature if you have programs and files for example, and need them seperated. This process repeats for each of the files on the source disk. The final prompt is "Compress? (Y/N)". Saying "Y" and Archiver attempts to squash each file so it takes up less space. Remember that some unusual file types will actually get LARGER if compression is attempted. When all the prompts are answered, press REDO to correct an error in your answers, BACK to return to the menu, or any other key to continue. When Archiver is done performing any operation, pressing a key goes back to the main menu.

2) Extract Files - This is the opposite of archiving. It will let you pull (extract) files from an ARC file. You are first asked for the source drive. Next you input the source filename. After that, you are asked for the output drive. It must be stressed that the output drive for ALL operations of Archiver should be different than the input drive. You may run out of space or overwrite a file accidentally. Output disks should be blank.

The next prompt asks, "Extract all files?" If you answer "Y" then every file stored in the ARC file will be taken out. If you answer "N" then when extracting starts, the program asks, "Include filename?" for every separate file in the archive. Again, press REDO (to restart this option), BACK (returns to main menu), or any other key to continue.

3) Catalog Disk - This is fairly self explanatory. Simply input the source drive name. The program will ask if you want a printout. If you answer yes, then you are asked for the printer name. If there are more files than can be displayed, then [more] is printed on the screen and pressing a key advances the screen.

4) Catalog ARC File - If you aren't sure what files are contained in an archive file, than this option tells you. You are asked for the source drive, source filename, and whether or not you want a printout of the list of files.

5) File Copy - This option will copy a file (obviously). Simply supply the source drive and filename, and the output drive and filename.

6) File Rename - Again, this option should explain itself. Give the source drive and filename, then the output filename.

7) File Delete - Supply the source drive and filename.

8) File Un/Protect - You first supply the source drive and filename. You are then asked "Protect?" If you answer "Y" the file is protected. Otherwise, file protection is lifted.

9) List Text File - This will display or print a D/V 80 file. Give the source drive and filename. You are then asked if you want the file printed or not.

10) Load FW - This returns to Funnelweb. Simply give the drive number on which the UTIL1 file is located.

NOTE: When an I/O error occurs, pressing a key returns to the main menu. If you have a Geneve, this is for you. Using a sector editor, find the string 04E08C00 and replace it with D8018C00.

I think that this should get people on the road to understanding archiver. Remember that it is fairware, so if you find it very useful, send the author (Barry Boone) a donation.

[This article/item comes from the January 1991 issue of BITS, BYTES PIXELS (Charles Good, editor), the newsletter of the Lima OH 99/4A User Group, P.O. Box 647, Venedocia, OH 45894.]

FROM THE TEACHER'S DESK-

Reprinted from
ERIE 99-ER U.S.
April 1991

Over the years, you've read much about the virtues of computers in schools - much of it in this Newsletter. Therefore, it is only fair to view the other side of the coin. To do this, I've selected comments found in a recent issue of SCHOOL TECH NEWS entitled:

"WHAT THE CRITICS ARE SAYING ABOUT COMPUTERS IN EDUCATION"

Are we spending too much money on computers in school? Sometimes we are, says Michael Wessells, Professor of Psychology at Randolph-Macon College in Ashland, Va.

Wessells thinks computers are used too often for "electronic page-turning" - presenting lengthy texts that could be handled just as well in book form - and for displaying "electronic flashcards" in learning languages. He thinks the money spent on such seemingly misguided use of computers in the classroom might be used more wisely by allocating costly computer time for "richer, more satisfying, more interactive activities that use the power of the computer" to better advantage. Wessells says he is neither skeptical nor "starry-eyed" where computers are concerned. He sees his job as "encouraging people to ask hard questions" about computers, especially computers in school.

Computers have a significant place in education and a great contribution to make there, Wessells says. But as with other technologies, such, such as educational television, "naive idealism" tends to make some people "see computers as the salvation of education," he points out. "I don't think they are."

Setting priorities for computer education is important in this age of fast-evolving technologies, Wessells says. What goal should schools aim for to make every student capable of using a state-of-the-art computer? Programming is less important in the curriculum now, but schools still struggle with the issue of what to teach about computers, he adds.

Another aspect of computer education that concerns Wessells is, "the social context of computers in education," as he puts it. Consider gender inequities, for example. At early ages (up to third grade or so), boys and girls use computers about equally. Then a divergence appears: boys start using computers more, while girls use computers less. This divergence reflects the emphasis and expectations of a culture that treats computers as as a masculine domain, Wessells says.

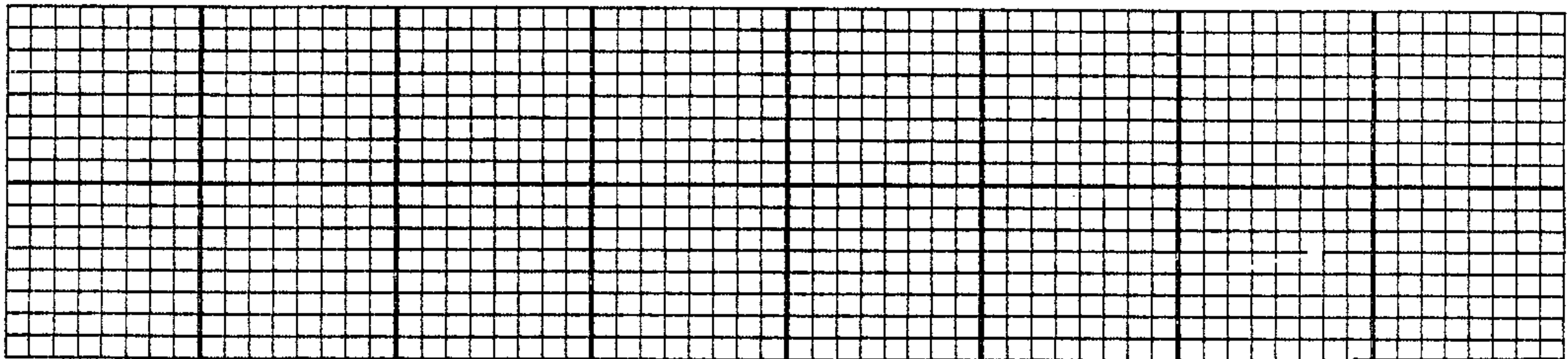
Class inequities are a problem too, Wessells goes on. Using sixth-graders as a case in point, Wessells says that computer use at school is proportional to computer use at home - so that children from middle-class and upper-class households enter the computing environment at school with an advantage over lower-class children who have less opportunity to use computers at home.

Schools in areas where lower-class children predominate, find that students use computers less frequently and in less productive ways than in schools serving more affluent communities. This is one

way computers "can be used to reinforce the dominant social order," Wessells says.

This last observation of Wessells is precisely why I loan computers to my computer students whose families cannot afford their own. I credit the ERIE 99'ers User Group for making this possible.

Need Graph Paper In A Hurry? Here's A Program To Do It



```

100 ! GRAPH
110 E$=CHR$(27)
120 A$=RPT$(CHR$(128),228)
130 B$=RPT$(CHR$(255)&SEG$(A$,1,6),8)
140 B$=RPT$(B$&CHR$(255),4)
150 A$=E$&"K"&CHR$(228)&CHR$(0)&A$
160 B$=E$&"K"&CHR$(228)&CHR$(0)&B$
170 OPEN #1:"PIO.CR"
180 FOR I=1 TO 11
190 PRINT #1:E$;"@";E$;"3";CHR$(24)
200 FOR J=1 TO 8
210 PRINT #1:B$;B$;CHR$(10)
220 NEXT J
230 PRINT #1:A$;A$;E$;"3";CHR$(2)
240 NEXT I
250 PRINT #1:RPT$(CHR$(13)&CHR$(10),9)
260 PRINT #1:E$;"@"
270 CLOSE #1

```

TI BITS * Number 29
by Jim Swedlow

This article originally appeared in the User Group of Orange County, California ROM.

I had a problem recently making TI Writer print out some revisions to our club bylaws and I thought you might be interested. Some background is needed first.

TALKING TO YOUR COMPUTER

Some time ago I mentioned that there are almost no standards for software codes to control printer functions (font, appearance, line spacing, etc). There are, however, a few things that are uniform. This month's effort centers on two of them: carriage return and line feed.

For simplicity, I will refer to them as <CR> and <LF>. <CR> is ASCII CODE 13 and <LF> is 10 (or, in hex, 0D and 0A, respectively).

What do they do? Just what they say. A <CR> causes your printer head to return to the beginning of the current line. A <LF> causes the print head to advance one line, in the same print position. Sending both <CR><LF> caused the print head to move to the beginning of the next line.

This is almost universal. Many printers, however, have the ability to add a missing <CR> or <LF>. When in this mode, receipt of either command causes the print head to move to the beginning of the next line. Receiving both causes a double space. Usually this is controlled with DIP switches and is discussed in the printer's manual.

When you open a printer as "PIO" (I won't discuss RS232 names here because they act just the same), your RS232 card sends your printer a <CR> and <LF> after every print line. You cannot control this as it is automatic.

You have some software control, however, by adding ".LF" or ".CR" to the printer name.

Here is a simple program that will show you what these extensions do:

```
100 DATA "PIO", "PIO.LF",  
"PIO.CR"  
110 FOR I=1 TO 3  
120 READ A$ :: OPEN #1:A$  
130 PRINT #1:A$:A$  
140 CLOSE #1  
150 OPEN #1:"PIO"  
160 PRINT #1: : :  
170 CLOSE #1 :: NEXT I
```

When the printer name is PIO, your RS232 card will add a <CR> and <LF> to each line and your printer will print

```
PIO  
PIO
```

When the printer name is PIO.LF, your RS232 card will only add a <CR> so you will get:

```
PIO.LF
```

but it will be printed twice (bold). With PIO.CR, neither <CR> nor <LF> is sent so you should have:

PIO.CRPIO.CR

PRINTERS AND TI WRITER

Wait, you say, in TI Writer (of FUNNELWEB, ect), I can add <CR> to lines. Does this change things? The answer is yes and no.

The primary function of the <CR> is to tell TI WRITER that you have reached the end of a paragraph. It uses it in the Editor when you cause a reformat and the Formatter when you FILL or FILL and ADjust.

What does it do to your printer? Well, lets say you have this text in your Editor:

UGOC<CR>

and then print it with PrintFile and PIO as your printer name. Your TI Writer would send this to your printer:

UGOC<CR><CR><LF>

In other words, it adds a <CR><LF> to whatever is on the line. The double <CR> is the same as a single <CR> as the print head can only move to the beginning of the print line once.

WHY DOES THE FORMATTER USE A ".LF"?

In short, the answer is to enable you to use bold (@@) and underscore (&&). Remember, when your printer name is PIO.LF, your RS232 card only adds a <CR> to each print line. Here is how it works.

If your entry line was:

@@UGOC

The formatter would send the following to the printer:

UGOC<CR>
UGOC<CR>
UGOC<CR><LF>

This would cause triple printing of UGOC or bold. Underline is similar. If your entire line was:

&&UGOC

The formatter would send the following to your printer:

UGOC<CR>
____<CR><LF>

On the page, UGOC would be underlined.

STRIKE OUT AN ITALICS

I am almost done with the background. As I said at the beginning, I was working on an update to our bylaws. The normal way to show changes is to strike out old text and put in new text in italics. For italics, I used Transliterate to change the open bracket "{" to start and the close bracket "}" to stop italics:

```
.TL 123:27,52
.TL 125,27,53
```

Whenever the formatter runs into an open bracket "{" or ASC 123, it will send the ASC codes 27 and 52 to the printer, which switches to italics. Similarly, "}" or 125 causes the formatter to send 27 and 53 which returns my printer to its normal font. It made editing easy as anything I wanted in italics I simply enclosed in brackets like {print this in italics}.

For strike out, I used transliterate to change the underscore to a dash:

```
.TL 95:45
```

Then when I used the underscore command, the Formatter printed strike out. So ~~&&old &&text~~ would print as:

```
&old &text
```

NOW FOR THE PROBLEM (FINALLY)

When I mixed strike out and italics on the same line and the italics were first, I got garbage. Why? Because TI Writer and my printer count characters differently.

Suppose I had this line:

```
{new text} for &&old &&text
```

When it went through the formatter, the printout looked like this:

```
new text for &old &text
```

Why? Well, you see, TI Writer counted the open and close brackets as characters while the printer, recognizing them as control codes, didn't. So TI Writer thought it was sending this:

```
..new text.. for old text<CR>
--- ----<CR><LF>
```

The double dots ".." stand for the control codes that the brackets were transliterated into. TI Writer, as you can see, counts the number of characters sent to the printer, not the number in the Editor print line.

The problem was that the printer counts the print characters this way:

```
new text for old text
--- ----
```

The result was that the strike out was not correctly aligned.

The solution was to make sure that, in the same line, italics did not precede strike out. It could follow or be on a different line, but not precede.

So I wrote this:

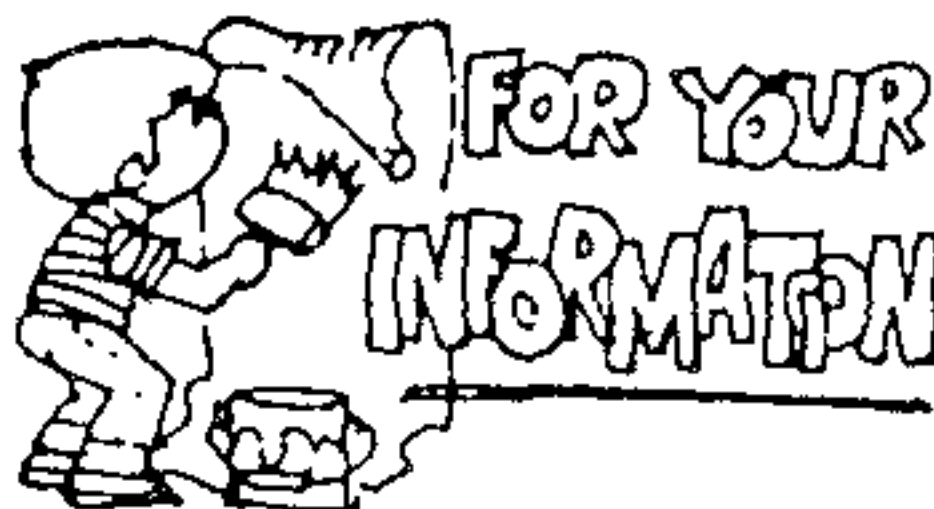
```
&&OLD &&TEXT becomes {new text}
```

and it printed like this:

&Old &Text becomes new text

I left ADjust off (using only FILL) because the difference in character counts defeats the way the Formatter right justifies text.

Enjoy.



From The Editor

Last month I was in too much of a hurry to get our September newsletter to the printer. As a result, I did not give recognition to articles reprinted from other sources, and above all, I did not include the entire article "GREENHORNS" by Sister Pat. For these blunders, I wish to apologize. It is our intention to give due credit to all authors and publication sources.

Below you will find the article and where they were obtained, if known.

Page 11 PROGRAMING TIPS - Sorry, source unknown.

Page 12 GEMINI 10-X - Source and author not available.

Page 15 KEYBOARD READER - Pomona Valley 99ers 7/91.

Here is the last part of Sister Pat's article GREENHORNS that started on page 13 in the September issue of the newsletter.

Take heart, if I can, you can! One last experience. I knew the formatter would consider a period a dot command, and did not know how to write about code using the dot in front of the code as a sample. I knew I needed a space marker and thought Function W was it as it looked like it. It is Shift 6. It only took three retypings to straighten that out!

Do I find this discouraging? Absolutely not, when you learn this way, you tend to remember it well. The end product is eventually you do know what you are doing in one more tiny area of computer experience!

Sr. Pat Taylor, BVM

**MEETING DATES
FOR
1991-1992**

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

3RD SATURDAY
19 OCT 1991
16 NOV 1991
21 DEC 1991
18 JAN 1992
15 FEB 1992
21 MAR 1992
18 APR 1992
16 MAY 1992
20 JUN 1992
18 JUL 1992
22 AUG 1992

4TH WEDNESDAY
23 OCT 1991
27 NOV 1991

Pres. - Chuck Grimes 614/268-8821
Treas - Everett Wade 614/262-6346
Sec/Sat - Jim Peterson 614/235-3545
Sec/Wed - Dick Beery 614/459-3597
Membership - Harley Ryan 614/231-1497
Librarian - Chuck Grimes 614/268-8821
Disk - David Truesdale 614/238-0719
Cassette - Everett Wade 614/262-6346
Cartridge - Jim Seitz 614/875-5519
NL Exchange - Jean Hall 614/885-4223
Chuck's BBS 614/268-1994
TIABS BBS 614/852-4579
Vice Pres. - Bill Sheppard 614/881-5742
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



Spirit of 99

**Bulk Rate
U.S. Postage
PAID
COLUMBUS 43221
Permit No. 1945**

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

**TIME SENSITIVE MATERIAL
POSTMASTER - PLEASE DELIVER PROMPTLY**

FX 127911 D
