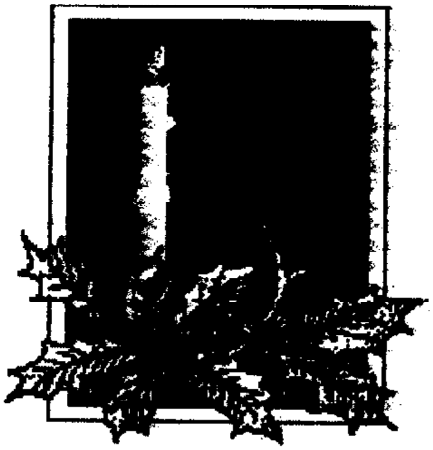
TI-D-BITS

PHILADELPHIA AREA USERS GROUP NEWSLETTER
COVERING THE TI99/4A
AND MYARC 9640 COMPUTERS

DECEMBER 1991

Volume 11 Number 10

Merry Christmas



The Philadelphia Area TI-99/4A Users' Group meets twice a month. On the first Saturday of each month, at The Church of the Atonement, 6200 Green St. Germantown (Corner of Green St and Walnut Lane) at 10 A.M. And on the third Saturday of each month, we meet at Drexel University, in Matheson Hall at 34th and Marker St. Phila. Pa Check the room chart posted at Matheson Hall for the current Room No. Membership to The Philadelphia Area TI-99/4A Users' Group is available to all. We invite anyone that is interested in the TI-99/4A to visit us. Stop in and see what is available to you for your TI and how membership can benefit you!

Current executive board consists of:

PRESIDENT
Barry Traver...... 215-483-1379

VICE PRESIDENT
Allen Silversteen. 215-885-7910

SECRETARY
Tim Coyne......... 215-947-5881

TREASURE
Don Arsenault..... 215-368-0446

Committees consists of:

TI-D-BITS
Ralph Field....... 215-362-2534
Tim Coyne......... 215-947-5881

LIBRARY
Rich Mascara...... 215-441-4060

MEMBERSHIP
Don Arsenault..... 215-368-0446

EDUCATION
Barry Traver
Allan Silversteen

EQUIPMENT
Allen Silversteen

PROGRAM
Dr. Eric Bray

REMEMBER to be considerate when calling any of the above people. Limit your calls to the early evening hours. (6pm to 9pm)

The opinions expressed herein are those of the individual authors are not necessarily those of the Philadelphia Area TI-99/4A Users' Group or its officers. Nor is the Philadelphia Area TI-99/4A Users' Group or any of its officers responsible for any damage, inconvenience, or loss which may result as a consequence of the use of any written material herein.

TI-d-Bits is published monthly by the Philadelphia Area TI-99/4A Users' Group, c/o Ralph E. Field, 603 N. Broad St., Lansdale, PA 19446. All material herein may be reprinted freely by other non-profit User Groups, (unless otherwise stated), as long as proper credit is given to both source and author. Contributions are encouraged, but no payment is made. Editorial, advertising, and classified, copy MUST be in by the LAST day of the previous month. You can either mail your copy to: TI-d-Bits, The Philadelphia Area TI-99/4A Users' Group, c/o Ralph E. Field, 603 N. Broad St., Lansdale, Pa. 19446 or send it via modem by contacting Ralph E. Field at (215)-362-2534. If your piece contains any, diagrams, charts, or code, send a paper copy AT FINAL PUBLICATION SIZE.

The editor of TI-d-Bits or the executive board of The Philadelphia area TI-99/4a Users' Group reserve the right to reject any material submitted for publication for any reasons.

The Philadelphia Area TI-99/4A Users' Group's program library is available to all active members at NO CHARGE for copying to your disk. A charge of $2.00 per disk is made for club supplied disks for members. Non members may obtain copies of the library for a fee of $5.00 per disk. A catalog of the library's contents is given to all new members upon request and updates will appear in this publication from time to time. To obtain material from the library, contact the librarian for the best procedure to obtain your requests.

PUTTING IT ALL TOGETHER 8

by Jim Peterson

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

Key in this little program and run it to see what it does, then read the explanation of how it does it.

In the early days, when computers had tiny memories, much emphasis was placed on efficient programming - the pioneer David Ahl called it "elegant" programming. The old 99'er magazine published some one-liners. My Tips From The Tigercub contained some one-line programs, even some no-line programs that could be keyed in and run in the immediate mode. In order to cram over a hundred subprograms onto a disk, I made great use of compact programming techniques on my Nuts Bolts disks. Later, Mike Stanfill originated the name "tinygram" for a program that would fit on one screen, and wrote some great ones. Ed Machonis wrote a diskfull of tiny printer utilities.

Richard Mitchell in his Super 99 Monthly once called me the "king of the one liners", but this title rightly belongs to John Martin. The following one-line disk cataloger is an example.

```
1 IF F THEN INPUT #1:A$,A,J,
K :: IF J THEN PRINT A$;TAB(
12);J;TAB(18);SEG$(B$,ABS(A*
2)+1,2);K;TAB(27);A<0 :: GOT
O 1 ELSE RUN ELSE B$="AVDFDV
IFIVPG" :: INPUT "DSK":F ::
OPEN #1:"DSK"&STR$(F)&".",IN
TERNAL,RELATIVE,INPUT :: GOT
O 1 !BY JOHN M
```

An undefined numeric variable has a value of 0, which is the value of F when the program is first run. IF F THEN is interpreted as "if F is other than 0" so program execution jumps to the first unpaired else. IF J is paired with ELSE RUN so execution jumps to ELSE B$; a string is assigned to B$, the INPUT asks for a disk number, and file #1 is opened, without a filename, as an internal relative file, for input. When it is opened, the first sector of the disk can be read; it contains information regarding the disk and its contents. GOTO 1 goes back to start over. The variable F now as a value other than 0 (from the INPUT disk number) so the values for A$, A, J and K are read from the disk. On the first pass, these are the disk name, a 0, the number of sectors initialized, the number of sectors available, and a 0. IF J THEN is interpreted as "if J is other than 0" and it is because it contains the number of sectors, so the disk name is printed, followed by the number of initialized sectors at tab 12. Since a 0 was read into A, the ABS(A*2)+1 is 0 times 2 plus 1, which is 1, so the segment of "AVDFDVIFIVPG" starting with the first character and consisting of two characters (AV) is printed (meaning "available"), followed by the number of available sectors read into K (preceded by a space because it is numeric). Since a 0 was read into A, the statement A<0 (A is less than 0) is false and has a truth value of 0, so a 0 is printed at tab 27. Execution returns to the beginning, and values are read into the variables again. Now, A$ will be a filename. A will be a number from 1 to 5, indicating the type of file - 1 for display fixed, 2 for display variable, 3 for internal fixed, 4 for internal variable, 5 for a program. If the file is protected, the number will be negative. J will be the number of sectors occupied by the file, and K will be the record length of the file (0 in the case of a program). The filename is printed, and its sector length at tab 12. ABS converts the A from negative to positive, if necessary, and the formula selects the letters DF, DV, IF, IV or PG to print, followed by the record length from K. If the file is protected, A has a negative value and A<0 therefore has a truth value of -1, otherwise a 0, printed at tab 27. Execution goes back to the beginning and this continues until blank records are read. J will then have a value of 0 so execution jumps to ELSE RUN, which re-runs the program, thereby zeroing out the value of F.

****

WHATEVER HAPPENED TO THE FUN OF IT?

by Jim Peterson

Yes, whatever happened? Were you with us back in the early days, way back back in 1983? Do you remember the days before the BBS's and user group libraries could supply you with programs by the hundreds, when every program you could acquire was a prized possession to be run and used and marveled at, to be shared with your friends, (even though it might have a copyright notice on it!), when people actually keyed in programs from listings and brought them to meetings to get help in debugging them?

Do you remember when almost everyone was trying to learn to program, and helping each other? The late Earl Dodd was writing music programs, in his unique barbershop quartet style, and bringing them to me to add graphics. Paul Powers, once our user group president, liked to program advanced math theorems and to reprogram more efficiently the programs written by others - it was he who suggested to me the use of mergeable subprograms, which led to the Nuts Bolts series, the only profitable thing I have ever done. It was teenage Brian Beery who pointed out a ridiculous error that was driving me bonkers - Brian might have become another J. Peter Hoddie if he hadn't taken up the guitar.

Do you remember when the IUG was the only source of public domain programs? I was fascinated by the program descriptions in the IUG catalog. I wanted to see every program, to see what some other programmer had been able to do. I swapped the IUG for every program I could, and bought as many more as I could afford. Often I was disappointed, but I never got over that curiosity. I still have it - I read the descriptions of new library acquisitions in user group newsletters, and I get the itch to see the program. Often I write and ask for it.

But, am I the only one left who has that curiosity? I spent hundreds of hours gleaning out the best from my library of several thousand PD programs, arranging them by category, improving them, filled nearly 300 disks (now 400), published a 13-page catalog listing them all; offered them for a copying fee less than most user groups charge their own members, cheaper than downloading them from GENIE - and in 1989, only 175 people in all the TI world were curious enough to send me an order!

Of course, many users have large libraries of programs that they never get around to even looking at. And, the potential uses of the computer have become so varied that many users have specialized in one field and have little interest in anything else. Some are mainly interested in increasing the speed and memory capacity of their machine, and have little time to make use of that speed and memory by actually running programs. Many others nowadays are hooked on graphics. To each his own. Personally, if I want to decorate my walls with pictures of nudes, I will buy a Playboy magazine and rip out pictures far better than any monitor screen will ever show or any dot matrix printer will ever produce!

Of course, even in the early days all was not sweetness and light. It seemed that everyone was out to make a buck, and those who made the buck were mostly those with questionable business ethics. When I first made contact with the rest of the TI world, I had already written about 90 programs, and I soon met people who wanted to form a business partnership with their one or two programs and my ninety. It was mainly to get them off my back that I decided to go into business for myself - a decision that I have regretted a thousand times.

Charlie LaFara started the International User Group as a nonprofit exchange of public domain software, and converted it into a business for his own profit. An entrepreneur in California acquired his programs and copied his catalog, leading to a lawsuit. Later on, a TI business in Florida called itself a "group" and sold my public domain programs, which led me to announce that "Tigercub Software is a one-man user group pretending to be a business, not a business pretending to be a user group!"

Of course, not everyone was a crook - most simply started out with unrealistic expectations, got in over their heads and faded away, leaving their creditors holding the bag. Emerald Publishing Co. extended credit to too many software advertisers who never paid up; then they got ridiculous and tried to publish a magazine with no advertising! Finally, they ripped off all their remaining subscribers, as did more than one other TI publication which never delivered or refunded subscriptions. There were exceptions, honest companies such as Random Access and another one-man operation in Texas whose name I wish I could remember.

I learned very early not to extend credit to anyone who decided to start up a software business. And I learned not to send an order to anyone for anything until I knew that someone else had actually received their order. There are not many TI old-timers who have not been ripped off at least once!

But, whatever happened to the fun of it? I remember demonstrating my programs at a local school, at a library, at computer fairs - the kids were fascinated! If only the Apple peddlers had not succeeded in brainwashing the educational system! Are Jack Sughrue and Eunice Spooner the only educators still using the TI in the classroom? In the days when I was exchanging my Tips From The Tigercub newsletter with nearly 200 user groups, I twice asked them to let me know of any schools in their area where the TI computer was being used - only two ever responded! If we had harnessed all that youthful energy and enthusiasm, our user groups might now have replacements for all those who are abandoning us for Big Blue.

Whatever happened to the fun of it? Whatever happened to the HOME computer? (remember, that was what the TI-99/4A was called!). They tell me that the COCO is the only home computer left, because there is no way to make it anything more than that. I may just take a look at it - maybe that's where I'll find the fun that I'm missing!

****

MUSIC PRO VERSION 1.4

Reviewed by Jim Peterson

This program, written by David Caron, won first prize in a software contest held by the Ottawa TI-99/4A User Group in 1988. It is now being sold by Asgard Software. I am sure that Asgard is selling it by arrangement with Bruce and the Ottawa group - in fact, Asgard supplies it with a companion disk containing a utility program for use with Music Pro, also written by David Caron, and the documentation file for this utility refers users to Asgard to obtain Music Pro. However, the status of Music Pro itself is confusing because the program still bears a fairware notice requesting a donation to the Ottawa UG!

The program is written in Extended Basic with many links to assembly, which take up a great deal of memory, and VDP memory is completely remapped. The documentation contains warnings about doing things that could corrupt this memory, and the documentation for the utility program contains many more warnings. They seem to be needed - when I tried keying in a short piece of music and tried to save it, the program crashed, and I found that my filename had been replaced by a long string of garbage!

I must first say that this is a truly remarkable piece of programming. Lucie Dorais has contributed some help files which make it even better. The documentation is also very well written.

The Music Editor displays three blank staff lines on the screen, treble and bass. Notes are entered on these staffs by pressing the appropriate key on the computer keyboard, from Q for low A to the period for high F - alternate rows on the keyboard provide sharps and flats. The disk contains a file which will print out a chart, but a person working with this program would probably soon memorize the keys. Provision is made for shifting to a higher octave but the bass octaves, which are accessible in Extended Basic programming through the -4 noise, are not available.

The length of the note is controlled by holding down the key while its pattern on the staff cycles from a 32nd note to a whole note - I found myself wishing that it would cycle back around when I held it too long. Many editing keys are available, as well as full control of volume. Notes can be "tied together" for a smoother sound; otherwise, the music has a rather "toot-toot" sound.

Only one voice can be programmed at a time. This is saved as a file, and a second and third voice can then be programmed and saved. The Compiler will read all three files and play them together in three-part harmony. While programming the music, the current voice can be played through the Editor, but the effect of combined voices can only be checked by saving the file and going to the Compiler.

This program was obviously written to enable a person with no knowledge of music to simply copy sheet music onto the screen. It does that extremely well. The person would still need some knowledge of key signatures, accidentals, and the notations used to indicate repetitions. I also think they would find it difficult to key in the three separate files without error, and to track down the errors.

However, this is certainly not the most efficient way of programming music, nor the most creative way, and it does not take advantage of musical effects, such as rapid attack and decay, which could only be achieved in assembly. To do so would have probably taken too much memory, and would have made the program too complex for its intended purpose.

I perhaps did not give the program a fair test but I did key in a melody in two voices. When I played them together, I found that I had made mistake somewhere, so that one voice was a quarter-note ahead of the other. The result was still musical, but quite unusual!

I decided that in the length of time I had spent on this, I could have programmed a couple of songs in all three voices in very elementary Basic, with the option of going to various subroutines for a wide variety of musical effects, and with many other features available.

****

SOME UTILITY PROGRAMS

Fm BYTE-LINE Newsletter of the DECATOR 99er User Group

```
90 ! METRONOME PROGRAM
100 D=100 :: PRINT "METRONOM
E PROGRAM":"PRESS SPACE BAR
TO SPEEDUP, AND ANY OTHER KE
Y TO          SLOWDOWN"!GARY B
ISHOP CV99UG 1991
110 FOR I=1 TO D :: NEXT I :
: CALL SOUND(5,220,1,300,1):
: CALL KEY(0,K,S): D=D ((K=
32)+(K=32)+1)(1+(S=0)):: GOT
TO 110
```

```
100 ! GRAPH PAPER PRINTER
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";C
HR$(24)
200 FOR J=1 TO 8
210 PRINT #1:B$;B$;CHR$(10)
220 NEXTJ
230 PRINT #1:A$;A$;E$;"3";C
HR$(2)
240 NEXT I
250 PRINT #1:RPT$(CHR$(13)&C
HR$(10),9)
260 PRINT #1:E$;"@"
270 CLOSE #1
```

TIPS FROM THE TIGERCUB

No. 64

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. I am out of printed documentation so it will be supplied on on disk.

My TI-PD library now has 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 will probably be printed by the time these Tips appear, and is available for $1 which is deductible from the first order.

Back in the days of David Ahl's Creative Computing magazine, when computers were to expensive for hardware hacking and had memory too small to run much of a program, the emphasis was on "recreational computing", and the British TI'ers carry on that tradition. A recent issue of their excellent TI*MES newsletter had this challenge - write a program to set up a circle of any chosen number of objects; starting at one, count them off by 10's, removing every 10th object. What are the numbers of the last two left?

This is my solution. It is not the best one, but it does show how strings can be used to perform math.

```
100 INPUT "NUMBER?":N
110 FOR J=1 TO N :: N$=N$&CH
R$(J):: NEXT J :: IF N<10 TH
EN 140
120 N$=SEG$(N$,11,255)&SEG$(
N$,1,9):: IF LEN(N$)>9 THEN
120
140 FOR J=1 TO 10 :: N$=SEG$
(N$,2,255)&SEG$(N$,1,1):: NE
XT J :: N$=SEG$(N$,1,LEN(N$)
-1):: IF LEN(N$)>2 THEN 140
150 FOR J=1 TO 2 :: PRINT AS
C(SEG$(N$,J,1)):: NEXT J
```

Which reminds me that I forgot to give you the answer to that short CALL SOUND puzzler in Tips #62. A CALL SOUND, even with a positive duration, will be interrupted by a BEEP.

Here's a bit of nonsense I worked up from an idea by Tim Brooks. Save this by SAVE DSK1.BUGS,MERGE . Then when you get a chance, load one of your friend's favorite programs, add this to it by MERGE DSK1.BUGS, and in the middle of the program somewhere put a line with CALL BUGS .

```
32000 !@P+
32001 SUB BUGS
32002 CALL CLEAR :: CALL CHA
RSET :: CALL DELSPRITE(ALL):
: CALL SOUND(225,220,0):: PR
INT "*ERROR 4 IN LINE 150" :
: PRINT "*BUGS IN PROGRAM"
32003 CALL SCREEN(8):: FOR A
=1 TO 500 :: NEXT A :: A$(1)
="997E3CFF3C7EBD99" :: A$(2)
="DB3CBD7E3CFFBD99" :: X=1 :
: CALL CHAR(96,A$(X))
32004 RANDOMIZE :: CALL MAGN
IFY(2):: FOR T=1 TO 2 :: FOR
 A=1 TO 20 :: X=X+1+(X=2)*2
:: CALL CHAR(96,A$(X)):: FOR
 D=1 TO 20 :: NEXT D
32005 CALL SPRITE(#A,96,2,19
5,RND*240,-5,0):: NEXT A ::
NEXT T :: CALL CLEAR :: CALL
 DELSPRITE(ALL):: SUBEND
```

Here is a puzzle game for you brainy types. I worked it up from a game by Jack Sughrue -

```
100 ! PSYCHO by Jim Peterson
110 CALL CLEAR :: RANDOMIZE
:: CALL SCREEN(2):: FOR S=1
TO 12 :: CALL COLOR(S,2,16):
: NEXT S :: CALL VCHAR(1,31,
31,96):: CALL KEY(3,K,S)
120 RANDOMIZE :: Y$(1)="+" :
: Y$(2)="-" :: Y$(3)="x" ::
Y$(4)="/"
130 CALL VCHAR(1,3,32,672)::
 D$="" :: Y(0),X=INT(10*RND+
5)
140 DISPLAY AT(2,11):"PSYCHO
":" Enter P(lus), (M)inus,
   (T)imes or (D)ivided by"
150 FOR J=1 TO 4 :: Y(J)=INT
(10*RND+5):: Z(J-1)=INT(4*RN
D+1)
160 IF Z(J-1)=1 THEN X=X+Y(J
):: GOTO 180 ELSE IF Z(J-1)=
2 THEN X=X-Y(J):: GOTO 180 E
LSE IF Z(J-1)=3 THEN X=X*Y(J
):: GOTO 180
170 IF X/Y(J)=INT(X/Y(J))THE
N X=X/Y(J)ELSE Z(J-1)=INT(3*
RND+1):: GOTO 160
180 NEXT J :: R=6 :: FOR J=0
 TO 3 :: DISPLAY AT(R,12):Y(
J):: R=R+2 :: NEXT J :: DISP
LAY AT(R,12):Y(4)
190 DISPLAY AT(R+1,12):"____
" :: DISPLAY AT(R+3,12):X
200 FOR J=0 TO 3 :: D$=D$&ST
R$(Y(J))&Y$(Z(J)):: NEXT J :
: D$=D$&STR$(Y(4))&"="&STR$(
X):: FOR J=1 TO 4
210 ACCEPT AT(J*2+5,12)SIZE(
1)VALIDATE("PMTD"):A$ :: IF
A$="" THEN 210
220 ON POS("PMTD",A$,1)GOSUB
 270,280,290,300
230 DISPLAY AT(J*2+4,12):""
:: DISPLAY AT(J*2+6,12):Y(J)
240 NEXT J
250 IF Y(4)=X THEN DISPLAY A
T(19,9):"RIGHT!" :: GOTO 260
 ELSE DISPLAY AT(19,9):"  WR
ONG! OFF BY";ABS(X-Y(4)):: D
ISPLAY AT(21,3):D$
```

```
260 DISPLAY AT(23,2):"PLAY A
GAIN? Y/N" :: ACCEPT AT(23,1
8)SIZE(1)VALIDATE("YN"):Q$ :
: IF Q$="N" THEN CALL CLEAR
:: STOP ELSE 130
270 Y(J)=Y(J-1)+Y(J):: RETUR
N
280 Y(J)=Y(J-1)-Y(J):: RETUR
N
290 Y(J)=Y(J-1)*Y(J):: RETUR
N
300 Y(J)=Y(J-1)/Y(J):: RETUR
N
```

Someone uploaded the New Testament books of the Bible to DElphi, probably ported over from IBM files. They included a program to break them into individual verses and another to display them on screen. Neither program worked properly, so I wrote this one to do it right.

```
100 CALL CLEAR :: CALL SCREE
N(16):: FOR J=1 TO 12 :: CAL
L COLOR(J,2,1):: NEXT J :: D
ISPLAY AT(2,8):"BIBLE READER
" !by Jim Peterson
110 DIM I$(127),L$(24)
120 DISPLAY AT(24,1):"DRIVE
#?" :: ACCEPT AT(24,10)VALID
ATE(DIGIT)SIZE(1)BEEP:DN ::
CALL CLEAR :: ON WARNING NEX
T
130 X=0 :: OPEN #1:"DSK"&STR
$(DN)&".",INPUT ,RELATIVE,IN
TERNAL :: INPUT #1:N$,A,A,A
140 INPUT #1:F$,A,B,C :: IF
LEN(F$)=0 THEN 160
150 IF C=80 AND ABS(A)=2 THE
N X=X+1 :: I$(X)=F$ :: DISPL
AY AT(X+(X>23)*23,1-(X>23)*1
3):STR$(X);" ";I$(X):: GOTO
140 ELSE 140
160 DISPLAY AT(23,1):"Read f
ile #" :: ACCEPT AT(23,12)VA
LIDATE(DIGIT):FL :: IF FL<1
OR FL>X THEN 160
170 CLOSE #1 :: OPEN #1:"DSK
"&STR$(DN)&"."&I$(FL),INPUT
:: CALL CLEAR :: DISPLAY AT(
3,1):"Press any key at the b
eep" :: X=0
180 LINPUT #1:M$
190 IF POS(SEG$(M$,1,5),":",
1)=0 THEN 220
200 IF FLAG=0 THEN FLAG=1 ::
 GOTO 220
210 X$=M$ :: GOTO 250
220 IF T$<>"" THEN M$=T$&" "
&M$ :: T$="" :: GOSUB 320 EL
SE GOSUB 320
230 IF LEN(T$)>27 THEN M$=T$
 :: T$="" :: GOSUB 320 :: GO
TO 230
240 IF EOF(1)<>1 THEN 180
250 IF T$<>"" THEN X=X+1 ::
L$(X)=T$ :: T$=""
260 CALL SOUND(1,500,8)
270 CALL KEY(0,K,S):: IF S=0
 THEN 270
280 FOR J=1 TO X :: DISPLAY
AT(9+J,1):L$(J):: NEXT J ::
FOR J=10+X TO 24 :: DISPLAY
AT(J,1):"" :: NEXT J :: X=0
290 IF X$<>"" THEN M$=X$ ::
X$="" :: GOSUB 320 :: GOTO 2
30
300 IF EOF(1)<>1 THEN 180 EL
SE IF X>0 THEN 250 ELSE CLOS
E #1 :: CALL SOUND(1,500,5)
310 CALL KEY(0,K,S):: IF S=0
 THEN 310 ELSE 100
320 IF LEN(M$)<29 THEN X=X+1
 :: L$(X)=M$ :: RETURN
330 IF SEG$(M$,28,1)=" " THE
N X=X+1 :: L$(X)=SEG$(M$,1,2
8):: T$=SEG$(M$,29,255):: RE
TURN
340 IF SEG$(M$,29,1)=" " THE
N X=X+1 :: L$(X)=SEG$(M$,1,2
8):: T$=SEG$(M$,30,255):: RE
TURN
350 P=27
360 IF SEG$(M$,P,1)=" " THEN
 X=X+1 :: L$(X)=SEG$(M$,1,P-
1):: T$=SEG$(M$,P+1,255):: R
ETURN
370 P=P-1 :: IF P>1 THEN 360
 ELSE X=X+1 :: L$(X)=SEG$(M$
.1.28):: T$=SEG$(M$,29,255):
: RETURN
```

Files ported over from IBM lack carriage returns, which can be a problem if you want to do any editing. I think this tinygram will do a good job of adding CRs to any text file which has centered headers and indented paragraphs.

```
100 DISPLAY AT(3,4)ERASE ALL
:"CARRIAGE RETURN ADDER":"":
"  This tinygram program wil
ladd carriage returns to any
 text file which has centere
d"
110 DISPLAY AT(8,1):"headers
 and indented para-  graphs.
"
120 DISPLAY AT(12,1):"Input
filename?":"DSK" :: ACCEPT A
T(13,4):IF$
130 DISPLAY AT(15,1):"Output
 filename?":"DSK" :: ACCEPT
AT(16,4):OF$
140 OPEN #1:"DSK"&IF$,INPUT
:: OPEN #2:"DSK"&OF$,OUTPUT
150 LINPUT #1:M$
160 IF M$="" THEN PRINT #2:C
HR$(13):M$;ELSE IF ASC(M$)<3
3 THEN PRINT #2:CHR$(13):M$;
ELSE PRINT #2:"":M$;
170 IF EOF(1)<>1 THEN 150 EL
SE CLOSE #1 :: CLOSE #2
```

Note that the program does all its work in line 160!

When text files are reformatted to a shorter line length, using the Funlweb Formatter, there are often long gaps at the ends of the lines, or between words if Fill and Adjust is used, due to long words which would have been hyphenated if the text had been originally typed in the shorter length. This little program will reformat text (containing carriage returns) to any shorter length and allow you to optionally hyphenate words which do not fit at the end of a line.

```
100 CALL CLEAR :: CALL SCREE
N(5):: FOR SET=0 TO 12 :: CA
LL COLOR(SET,2,16):: NEXT SE
T
110 CALL CLEAR
120 DISPLAY AT(12,1):"Input
filename?":"DSK" :: ACCEPT A
T(13,4)BEEP:IF$ :: OPEN #1:"
DSK"&IF$,INPUT
```

```
130 DISPLAY AT(15,1):"Output
 filename?":"DSK" :: ACCEPT
AT(16,4)BEEP:OF$ :: OPEN #2:
"DSK"&OF$,OUTPUT
140 DISPLAY AT(18,1):"Reform
at to what length?" :: ACCEP
T AT(18,26)SIZE(2)VALIDATE(D
IGIT):R
150 IF EOF(1)THEN 270 :: CAL
L CLEAR :: LINPUT #1:M$ :: M
$=P$&M$ :: P$=""
160 L=LEN(M$)+(POS(M$,CHR$(1
3),1)<>0):: IF L<=R AND POS(
M$,CHR$(13),1)<>0 THEN PRINT
 #2:M$ :: GOTO 150 ELSE IF L
<R THEN P$=M$&" " :: GOTO 15
0
170 C$=SEG$(M$,1,R):: CALL L
ASTPOS(C$," ",Y)
180 IF Y<>0 THEN 190 ELSE PR
INT #2:C$ :: M$=SEG$(M$,R+1,
255):: GOTO 160
190 IF R-Y<3 THEN C$=SEG$(M$
,1,Y):: M$=SEG$(M$,Y+1,255):
: PRINT #2:C$ :: GOTO 160
200 X=POS(M$," ",Y+1):: IF X
=0 THEN X=LEN(M$)ELSE IF X=R
+1 THEN PRINT #2:C$ :: M$=SE
G$(M$,Y+2,255):: GOTO 160
210 DISPLAY AT(2,1):M$ :: DI
SPLAY AT(8,1):SEG$(M$,1,R)
220 DISPLAY AT(12,1):SEG$(M$
,Y+1,R-Y-1)&"-"&SEG$(M$,R,X-
R+1):: Z=R-Y
230 DISPLAY AT(15,1):"Hyphen
ate?" :: ACCEPT AT(15,12)SIZ
E(1)VALIDATE("YNyn"):Q$ :: I
F Q$="N" OR Q$="n" THEN 260
240 ACCEPT AT(18,1)SIZE(Z):H
$ :: IF POS(H$,"-",1)=0 THEN
 240
250 C$=SEG$(C$,1,Y)&H$ :: M$
=SEG$(M$,Y+1+LEN(H$)-1,255):
: PRINT #2:C$ :: GOTO 160
260 PRINT #2:SEG$(C$,1,Y)::
M$=SEG$(M$,Y+1,255):: GOTO 1
60
270 CLOSE #1 :: CLOSE #2 ::
STOP
280 SUB LASTPOS(A$,B$,Y):: X
,Y=0
290 X=POS(A$,B$,X+1):: IF X>
0 THEN Y=X :: GOTO 290
300 SUBEND
```

I really think that all program listings should be published in 28-column format, as my Tips from the Tigercub have always been published, because that is how they appear on screen, making it much easier to key them in accurately. However, if you absolutely MUST reformat them, I think that this program will accurately reformat to/from any length up to 79 PROVIDING that you first put a carriage return at the end of every program line.

```
100 DISPLAY AT(3,6)ERASE ALL
:"PROGRAM RELISTER":"":"  Wi
ll reformat a LISTed    XBas
ic program from any lineleng
th to any other length."
110 DISPLAY AT(8,1):" Each
program line (not    file li
ne) must end in a    carriag
e return."
120 DISPLAY AT(12,1):"Input
filename?":"DSK" :: ACCEPT A
T(13,4):IF$ :: DISPLAY AT(15
,1):"Output filename?":"DSK"
 :: ACCEPT AT(16,4):OF$
130 DISPLAY AT(18,1):"Presen
t line length?" :: ACCEPT AT
(18,22)SIZE(2)VALIDATE(DIGIT
):A
140 DISPLAY AT(20,1):"Reform
at to what length?" :: ACCEP
T AT(20,26)SIZE(2)VALIDATE(D
IGIT):X :: IF X=A THEN 130
150 OPEN #1:"DSK"&IF$,INPUT
:: OPEN #2:"DSK"&OF$,OUTPUT
:: IF X<A THEN 230
160 IF EOF(1)THEN 270 :: LIN
PUT #1:M$ :: L=LEN(M$):: IF
POS(M$,CHR$(13),1)=0 THEN 18
0
170 IF P+L<X+1 THEN PRINT #2
:M$ :: P=0 :: GOTO 160 ELSE
PRINT #2:SEG$(M$,1,X-P):SEG$
(M$,X-P+1,255):: P=0 :: GOTO
 160
180 IF L<A THEN M$=M$&RPT$("
 ",A-L):: L=A
190 IF P=0 THEN PRINT #2:M$;
:: P=L :: GOTO 160
200 IF P+L<X THEN PRINT #2:M
$;:: P=P+L :: GOTO 160
210 IF P+L=X THEN PRINT #2:M
$ :: P=0 :: GOTO 160
220 PRINT #2:SEG$(M$,1,X-P):
SEG$(M$,X-P+1,255);:: P=LEN(
SEG$(M$,X-P+1,255)):: GOTO 1
60
230 IF EOF(1)THEN 270 :: LIN
PUT #1:M$
240 L=LEN(M$):: IF L+P>X THE
N PRINT #2:SEG$(M$,1,X-P)::
M$=SEG$(M$,X-P+1,255):: P=0
:: GOTO 240
250 IF M$=CHR$(13)THEN 230
260 IF POS(M$,CHR$(13),1)<>0
 THEN PRINT #2:M$ :: P=0 ::
GOTO 230 ELSE PRINT #2:M$;::
 P=LEN(M$):: GOTO 230
270 CLOSE #1 :: CLOSE #2
```

MEMORY FULL

Jim Peterson

APPEARING IN THE NEXT ISSUE:

>>>Beginning a series of TI-Base Tutorials by Martin Smoley of NorthCoast 99'ers (Ohio)

>>>A Tutorial by Tim Coyne on getting started in using TELCO, Charles Earl's great telecommunications software.

See ya then.

MINUTES FOR OCTOBER 5, 1991 MEETING

Church of the Atonement, Walnut & Green Lanes, Germantown, Philadelphia

Present: B. Traver, A. Silversteen, R. Fields, W. Bubeck, M. Kline, A. Knous, H. Beilstein, R. Muscaro, N. Sellers, C. Starnes, T. Coyne

A TI-Artist video demo tape was viewed briefly. Barry offered to make copies for interested members.

There was discussion about topics for future meetings:

Mentioned was the possibility of forming a TI-Base group since 4 or 5 people use this program.

Ralph is accomplished at using graphics and art programs, so a Picasso demonstration was suggested.

Organize a group to learn how to use PLUS!

A workshop on getting full use of printers, especially how to use transliteration in TI-Writer to control printer operation. Tim offered to present a demo on transliteration at the next Church meeting.

A workshop on how to configure FunnelWeb 4.40.

Many members concurred that it was important that the Club's software library once again be made available for copying. We need to get the original library masters from Ted Chemey.

Barry and Caesar announced that the Church is considering setting up an Early Childhood Learning Center. It would feature access to computers to enhance the development of basic basic educational skills. We discussed the role of the Club in using our equipment to help supply the facility. This would give us a chance to have all our equipment set up on a permanent basis instead of hauling it around for each meeting. Barry has a lot of learning software the Club could buy and donate or make available to the Center. We need to get club systems set up for hands-on use at meetings. At the conclusion of the meeting, Caesar showed several members the room on the second floor which might be used. Most agreed the facility has much potential.

We're still having problems getting repaired equipment back from Lou Phillips. We will have DSDD capacity at meetings when Ralph's H/FDC is repaired so he can return the club's.

A motion was passed authorizing Allen to spend $200 to buy a complete TI system. The club will sell the items that we don't need to keep for our use.

Respectfully submitted, Timothy C. Coyne, PATIUG Secretary

MINUTES FOR OCTOBER 19, 1991 MEETING

Drexel University, Matheson Hall, 34th & Market, Philadelphia

Present: Attendance not taken.

Tim offered the Club use of a free newspaper ad to which he is entitled in Montgomery (County) Newspapers. It could be used to solicit equipment donations for the Church's proposed Early Childhood Learning Center or to help advertise the Club and its activities.

Attendees confirmed that at the next Church meeting we would try to have equipment ready for work on TI-Writer transliteration codes and TI-Base. Members were encouraged to bring printer manuals to the next meeting so that transliteration files could be personalized for a member's use.

Ralph said he would try to be ready for a Picasso demonstration.

Barry announced that he would probably not be at the next meeting if he was able to work out a trip to the big TI Faire in Chicago.

Tim Coyne is interested in a Pinball simulation more sophisticated than what can be found in TI's Video Games 1 module.

A man named Steinbrook who was formerly a TI user stopped by the meeting to see if his old equipment might still have any relevance and utility. The membership enthusiastically educated him and invited him to the next meeting to see a demonstration of how much we could do now years later with a new generation of software.

Tim Coyne requested that he be given credit as the author of the article on how to make a modem cable which appeared in the November issue of the Newsletter.

Respectfully submitted, Timothy C. Coyne, PATIUG Secretary

MINI-NOTES ON THE NOVEMBER 2, 1991 MEETING:

Tim Coyne presented a workshop on creating a mnemonic transliteration file for controlling your printer. If you would like a copy of his notes, send a long self-addressed envelope with $.29 postage affixed to 535 Anne Street, Huntingdon Valley, PA 19006

Lot No.	Min.Bid	Description...		Due Date	Donor................
100	75.00	PEB with 32K Mem/ Interface/TI Disk controller Card/1 FH SSSD Drive		11.30.91	PATIUG
101	100.00	Myarc 512K Mem card with 128K OS Eprom/ Aux. Power Supply		11.30.91	PATIUG
102	50.00	TI RS232 card will set for RS232/3 & RS232/4 at n/c		11.30.91	PATIUG
103	10.00	Witco Track Ball w/Power Supply for joystick port			"
104	30.00	TI 99/4A Console Beige w/P.S. and RF Modulator			"
105	20.00	TI 99/4A Console Black and Silver			"
106	20.00	TI 99/4A Console Black and Silver			"
138	25.00	TI Disk Controller Card for PEB			"
139	25.00	Doryt..Paraprint Sidecar for console to use a parallel port printer only			BAT
107	10.00	Case for 18 TI Cartridges			PATIUG
108	3.00	Beginning Grammar TI-Command Module PHM	3003		"
109	2.00	Indoor Soccer	3024		"
110	5.00	Star Trek Sega Ent.	3225		"
111	1.00	Household Budget TI-C/M	3007		"
112	3.00	Disk Manager	3019		"
113	5.00	Tunnels of Doom	3042		"
114	3.00	Reading Fun Scott Foresman	3043		"
115	2.00	Early Learning Fun TI C/M	3002		"
116	2.00	Minus Mission	3118		"
117	2.00	Multiplication 1 SF	3029		"
118	2.00	Hunt the Wumpus	3023		"
119	2.00	Division 1 SF	3049		"
120	2.00	Video Chess	3008		"
121	2.00	Football	3009		"
122	3.00	Parsec	3112		"
123	3.00	Pac Man Atari for TI	----		"
124	4.00	Numeration I Scott Foresman	30213		"
125	4.00	Early Reading 1A Scott Foresman	30112		"
126	2.00	A-Maze-ing	3030		"
127	2.00	TI Invaders	3053		"
128	2.00	TI Writer	3111		"
129	2.00	Chisholm Trail	3110		"
130	2.00	Video Graphs	3005		"
131	2.00	Munchman	3057		"
132	4.00	Terminal Emulator II	3035		"
133	2.00	Facemaker Spinnaker	3177		"
134	2.00	Adventure	3041		"
135	2.00	Car Wars	3054		"
136	3.00	Shamus Atari for TI	----		"
137	3.00	Centipede Atari for TI	----		"

If you are interested in purchasing any of these items please call me. Allan Silversteen at 215 885 7910
I will submit your bid and on 11/30/91 notify you if you are successful and make the appropriate arrangements.