TI - D - BITS

SEPTEMBER / OCTOBER 1991

OK NO MORE HORSING AROUND

The summer is over and it's time to get
back to computing again. Stop by one of our
meetings and check us out. You might
find it well worth your while.
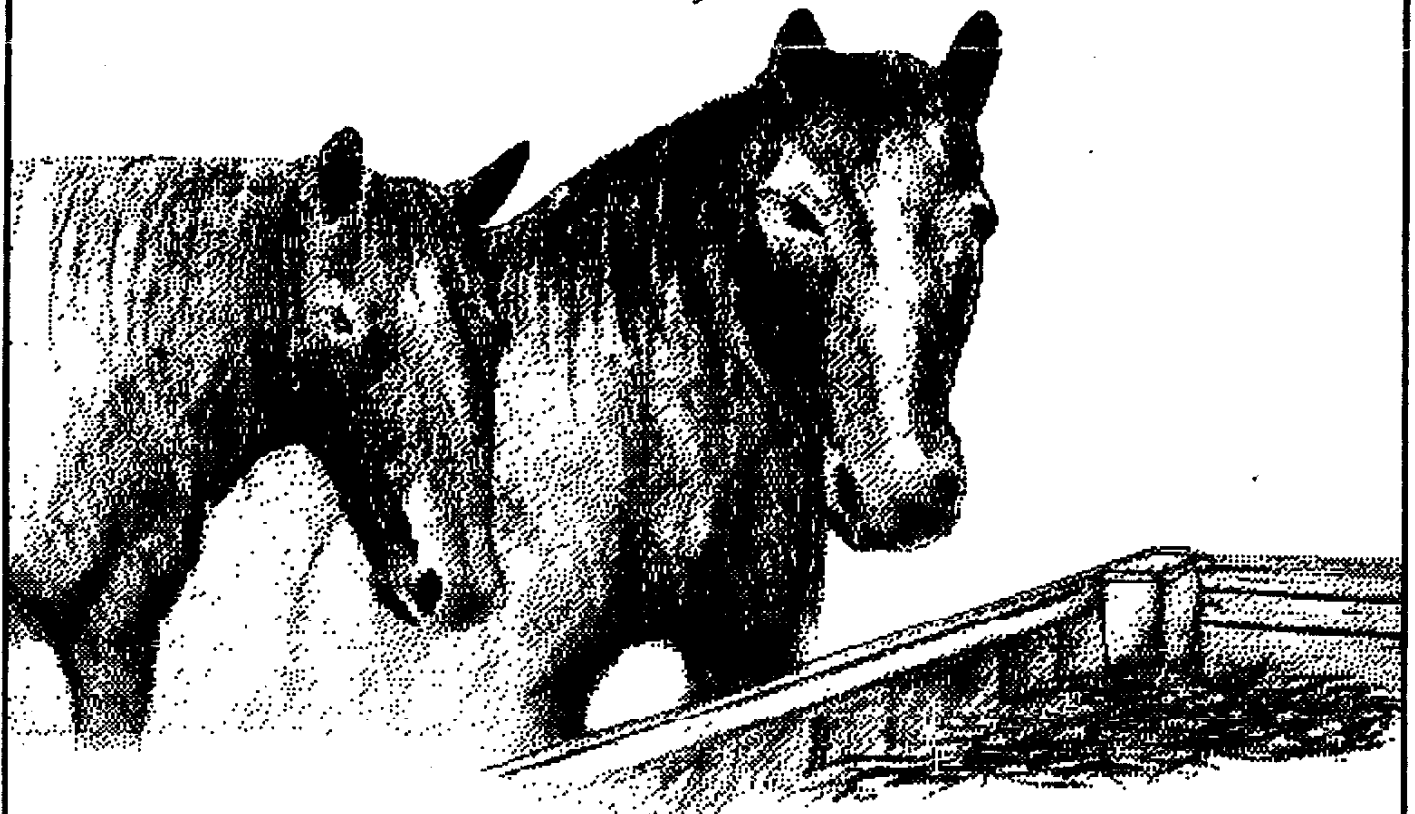
The Philadelphia Area TI-99/4A Users' Group meets twice a month. On the first Saturday of each month, at The Church of the Atonement, 6200 Green St. Germantown (Corner of Green St and Walnut Lane) at 10 A.M. And on the third Saturday of each month, we meet at Drexel University, in Matheson Hall at 34th 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

EQUIPTMENT
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, c/o 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.

TI WORLD NEWS
JULY 1991

compiled by Jim Peterson

CaDD Electronics (81 Prescott Road, Raymond NH 03077, (603)895-0119) has announced the availability of RICH GKXB by Richard Lynn Gilbertson, for $24.95 plus $2 PPM. This enhanced version of Extended Basic requires the use of a gram emulating device which supports the GK file header, such as the GRAM Kracker, Gramulator, and Geneve 9640. Users of the GRAM Karte may convert the GK file type to the GRAM Karte files with the "Universal GRAM File Converter" also available from CaDD.

Among other features, more than 40 new commands have been added to Extended Basic, as well as more than 150 combinations of commands executed with a single CALL for greater speed in execution, and even conditional combined commands in which the second part will only execute if the first part is true.

Rodger Merritt has developed a method of converting TIPS graphics into a form which can be printed from the TI-Writer or Funnelweb Editor (but not the Formatter), and a program called PIXEASE which makes it quite easy to combine these graphics with text in any combination, and even in color. He offers a flipped disk of 140 graphics for $5 (plus PPM, I presume) and has 30 such disks, available in several combinations and with a discount for the entire collection archived. The address is Comprodine, 1949 Evergreen Ave., Fullerton CA 92635.

Mike McCann's HQ_STACKS for the Geneve is such an advanced program that I have no clear idea what it is all about, but it is said to offer a graphical user interface screen, browser-stackware editing environment, MDOS command line interface utility, built-in F7 pop-up windows help system, and HQ_Stacks artwork resource program. If that sounds interesting, you can get a demo disk.

****

TI WORLD NEWS
AUGUST 1991

compiled by Jim Peterson

Mike Maksimik is now shipping his MIDI Master 99 Version 2.2. This has been publicized as having been developed using a CASIO keyboard costing $80, and is stated to be compatible with any keyboard having a MIDI interface. Unfortunately, one of the first buyers has reported great difficulty finding an inexpensive keyboard which has such an interface. It seems that they are usually only found on the expensive professional models.

I sent a letter to JP Software, addressed to 2390 El Camino Real #107, Palo Alto CA 94306. The letter was postmarked at Columbus OH on 28 September 1990. On 15 July 1991 I received it back, stamped "Addressee no longer at this address/ return to sender" and "Returned to sender/ addressee unknown". Two users have informed me that they sent an order to JP Software and their check was cashed but merchandise was never received.

In June, Shane Truffler of ESD posted a BBS message stating that the ESD Hard Drive Controller Card would support IDE drives instead of MFM, and that he had sent refund checks to those who had previously ordered in advance. He stated that ESD would ship a 40MB IDE drive with the controller card for a price, to be announced soon, in the $225-$250 range; that the IDE drives have faster acces times than all the MFM type drives; and that the controller card is directly on the hard drive, so that all that is needed is a simple interface to the TI. This is said to insure compatibility and to eliminate the problem that ESD has had with availability of WD MFM controller chips.

According to an unsigned article in the Kawartha Kronicle, Gary Bowser of O.P.A. has acknowledged that his RAMBO is not compatible with Super Synthesizer nor with the Speech Synthesizer, and the writer found that it was also incompatible with TI Extended Basic. He also found serious shortcomings in the RECALLIT +2 program - the two-column printout is

too wide for a standard printer, there is no way to input a country name and no line to print it on a label, and the "remarks" field cannot be printed in the label format.

An article in the OSHAWA TI UG newsletter reports that O.P.A.'s TI Image Maker (TIM) in its present version is incompatible with many monitors. It requires an RGB analog monitor - not RGB digital (TTL). It works well on a 15.75 kilohertz scan, but most inexpensive RGB monitors are 31.5 kilohertz. They also use separate horizontal and vertical scan lines, but TIM is set up to use only a composite synch line. The article states that a monitor with a pixel size of 0.42mm is acceptable but a smaller size is definitely better. The monitor should do a sceen size of 640 x 480 pixels, which is fairly standard on most VGA systems. Paying more for a higher size would be a waste of money, because TIM can only do 512 x 424. Among the monitors said to be compatible are the Commodore AMIGA 1080 and 1084, MAGNAVOX 8CM515, Thompson 4120, and SONY XBR series (KV1311CR) - the article does not definitely state that all of these completely meet the requirements.

The same article reports a compatibility problem between O.P.A.'s S.O.B. (Son of a Board) and the MYARC floppy disk controller.

TEXAMENTS (53 Center St., Patchogue NY 11772) is now buying, selling and trading used TI-99/4A and Geneve hardware, software, resource material and accesories. They will make a formal offer within two days if you mail them a list of items you want to sell. You can get a listing of items available for sale by sending them a self-addressed postpaid envelope, or by calling their BBS (516) 475-6463. Used hardware is sold with a 30-day warranty, and non-defective items can be returned within 15 days for a 20% restocking fee.

Chris Bobbitt has announced that Harry Brashear has joined Asgard Software in the capacity of limited partner, with responsible for almost all filling of orders. He expects that this will eliminate the long delays that customers have experienced in dealing with Asgard, and that he will now have time to resume

publication of Reflections/Asgard News and to concentrate more on new product development. Chris will cotinue to handle all customer service.

****

PROGRAMMING MUSIC THE EASY WAY

Part 3

by Jim Peterson

In Part 1 of this series, I showed you the simple routine to set up a musical scale, and showed you how easy it was to merge in various routines to create different effects in single-note music. In Part 2 I showed you how to key in single-note melodies from sheet music. Now, we will get into 3-part harmony.

But first, there are a few more things I should have told you about reading music. You will often see curved lines arching over two or more notes. If the notes are not all the same, ignore those lines - they call for phrasing which you cannot really accomplish. But, if the line curves over two or three of the same note, you will get a better effect if you add all their duration values together and program them as a single note. For instance, if your chart gives a whole note a value of 8 and a half-note a value of 4, and the music has a curved line over a whole note followed by a half-note, just program one note with a duration of 12.

You may find a heavy black bar at the beginning of a measure, with a colon to its right, and somewhere later in the music will be a heavy bar with a colon at its left. This means that the notes between those bars are to be played through twice - and naturally you will want to save time by programming them in a GOSUB as I showed you in Part 2. It can get more complicated than that, but generally you can follow the lyrics to decipher what to do.

Rather rarely, you may find three notes, usually joined together, with a 3 above them. These are called a triplet, and all three of them are to be played, with the same duration for each, in the length of time it would

normally take to play one of them. These can create a problem under any method of music programming. The best method is to divide the duration of the note by three and write individual CALL SOUNDs in your music, rather than a GOSUB to a routine, to handle those notes.

Now, let's get on to 3-part harmony. It is just the same as keying in single note music, except that you must also give frequency values to B and C - and, as before, you have to give those values only when they change.

So, load the SCALE routine from the first lesson, and key in this bit of music to experiment with. Notice that I found three repeating phrases and put them in subroutines in 500, 600 and 700 to make this shorter.

```
110 GOSUB 500 :: T=4 :: A=15
:: B=11 :: C=9 :: GOSUB 100
0 :: T=8 :: A=18 :: GOSUB 10
00 :: T=2 :: A,B,C=0 :: GOSU
B 1000 :: T=2 :: A=23 :: B=1
8 :: C=15 :: GOSUB 1000 :: G
OSUB 600
120 T=2 :: A=21 :: B=18 :: C
=15 :: GOSUB 1000 :: A=23 ::
GOSUB 1000 :: T=12 :: A=20
:: B=16 :: C=11 :: GOSUB 100
0
130 T=2 :: A,B,C=0 :: GOSUB
1000 :: GOSUB 500 :: T=4 ::
A=21 :: B=16 :: C=13 :: GOSU
B 1000 :: T=10 :: A=25 :: GO
SUB 1000
140 T=2 :: A=28 :: GOSUB 100
0 :: GOSUB 600
150 T=2 :: A=27 :: B=23 :: C
=18 :: GOSUB 1000 :: A=30 ::
GOSUB 1000 :: T=10 :: A=28
:: B=23 :: C=20 :: GOSUB 100
0
160 T=2 :: A,B,C=0 :: GOSUB
1000 :: T=3 :: A=28 :: B=23
:: C=20 :: GOSUB 1000 :: T=1
:: A=27 :: GOSUB 1000 :: GO
SUB 700
170 T=6 :: A=25 :: B=21 :: C
=9 :: GOSUB 1000 :: T=2 :: A
=23 :: B=18 :: C=15 :: GOSUB
1000
180 T=10 :: A=20 :: B=16 ::
C=11 :: GOSUB 1000 :: T=2 ::
A,B,C=0 :: GOSUB 1000
```

```
190 T=3 :: A=28 :: B=23 :: C
=20 :: GOSUB 1000 :: T=1 ::
A=27 :: GOSUB 1000 :: GOSUB
700
200 T=4 :: A=25 :: B=21 :: C
=16 :: GOSUB 1000 :: A=21 ::
B=18 :: C=15 :: GOSUB 1000
210 T=14 :: A=20 :: B=16 ::
C=11 :: GOSUB 1000 :: T=2 ::
A,B,C=0 :: GOSUB 1000 :: ST
OP
500 T=2 :: A=23 :: B=20 :: C
=16 :: GOSUB 1000 :: A=28 ::
GOSUB 1000 :: A=27 :: GOSUB
1000 :: A=28 :: GOSUB 1000
:: A=27 :: GOSUB 1000
510 A=28 :: GOSUB 1000 :: A=
23 :: B=20 :: C=16 :: GOSUB
1000 :: A=20 :: B=16 :: C=11
:: GOSUB 1000 :: A=16 :: B=
11 :: C=8 :: GOSUB 1000 :: R
ETURN
600 T=2 :: A=27 :: B=23 :: C
=18 :: GOSUB 1000 :: A=23 ::
B=18 :: C=15 :: GOSUB 1000
:: A=21 :: GOSUB 1000 :: A=2
3 :: GOSUB 1000
610 A=27 :: GOSUB 1000 :: A=
23 :: GOSUB 1000 :: RETURN
700 T=4 :: A=27 :: B=21 :: C
=16 :: GOSUB 1000 :: T=8 ::
A=25 :: GOSUB 1000 :: T=3 ::
A=27 :: B=23 :: C=18 :: GOS
UB 1000
710 T=1 :: A=21 :: GOSUB 100
0 :: T=4 :: A=25 :: B=21 ::
C=16 :: GOSUB 1000 :: T=8 ::
A=23 :: B=20 :: C=16 :: GOS
UB 1000
720 T=3 :: A=25 :: B=21 :: C
=16 :: GOSUB 1000 :: T=1 ::
A=23 :: GOSUB 1000 :: T=2 ::
A=23 :: B=18 :: C=15 :: GOS
UB 1000
730 A=21 :: GOSUB 1000 :: A=
20 :: GOSUB 1000 :: A=21 ::
GOSUB 1000 :: RETURN
```

Save that under the filename ROSES. clear the memory with NEW, and key this in -

```
1000 CALL SOUND(D*T,N(A),V1,
N(B),V2,N(C),V3):: RETURN
```

Save that by SAVE DSK1.PLAIN3,MERGE. Load ROSES again and merge it in by MERGE DSK1.PLAIN3 . Add a line - 105 D=200 and RUN it.

Sounds rather raw and harsh, doesn't it? Try changing that line 105 to -
105 D=200 :: V2=5 :: V3=8

Try it again. Sound better? The first time, all 3 voices were being played at the loudest volume. Usually computer music will sound better if the harmony notes are given a lower volume.

Experiment and find the volumes you like best. Is the music too slow for you? Just change the value of D. Is it not in your singing key? Just change the value of F in line 100, as I showed you before.

But, does the music still have too strong a beat for your taste? Clear the memory again and key this in -

```
1000 CALL SOUND(-4250,N(A+Z)
,V1,N(B+Z),V2,N(C+Z),V3):: G
OSUB 1010 :: RETURN
1010 FOR W=1 TO T*D :: NEXT
W :: RETURN
```

Save that as NEG3,MERGE because it uses negative duration for 3 voices. Then load ROSES again and merge it in. This time, try line 105 with D=50 and with V2 and V3 as you wish. Sound smoother?

In lines 110, 130, 160, 180 and 210 of ROSES, you will find A,B,C=0 . That makes all three voices silent, because in line 100 N(0) is given a frequency of 40000 which is above the range of human hearing. This is how I programmed those silent pauses, the "rests" which were written in the music.

On a piano or guitar, the strings continue to vibrate during a rest, so that the sound gradually fades out. However, the electronically generated tones of a computer stop very suddenly. That is why I often add the duration of the rest to the duration of the preceding note, and play it right on through. Some people think that doesn't sound right, so here is another solution. Clear memory again and key this in -

```
2000 FOR W=2 TO 8 STEP 8 ::
CALL SOUND(-999,N(A+Z),V1+W,
N(B+Z),V2+W,N(C+Z),V3+W):: G
OSUB 2010 :: NEXT W :: RETUR
N
2010 FOR Y=1 TO T*D/4 :: NEX
```

```
T Y :: RETURN
```

Save that as REST,MERGE. Load ROSES again, merge in SCALE and NEG3 (this will not work well with PLAIN3) and merge in REST. Now go to lines 110, 130, 160, 180 and 210, delete the A,B,C=0 :: and change the GOSUB 1000 after it to GOSUB 2000. Add line 105, run it and see if you like that better.
Anyway, keep it for now because we will use it again.

You will probably want to have the music play through more than once. Just add :: FOR J-1 TO 4 to the end of line 105 (if you want it to play 4 times) and change the end of line 210 to read NEXT J :: STOP .

I said that you could change the key of the music just by changing the value of F in line 100. There is also a way to change it while the music is playing. After the FOR J=1 TO 4 in 105 put :: Z=Z-(J=2)*3-(J=3)*1+(J=4)*4 That is somewhat complicated but it just means to play the second time three whole keys higher, the third time one key higher still (I know the *1 is unnecessary!) and drop back 4 keys for the 4th time, so you can take it from there and modify it as you wish. If you want to use that routine with silent rests, change the GOSUB after each rest to 3000 instead of 1000, and add this line -

```
3000 CALL SOUND(-4250,N(A),V
1,N(B),V2,N(C),V3):: GOSUB 1
010 :: RETURN
```

This tune happens to end in a rest, which is unusual. If you key in another tune and it seems to end too abruptly, just after that NEXT J and before the STOP, put in a long duration such as T=12 and a GOSUB 2000 to that REST routine to fade out more slowly.

Now, when you are keying in your own tunes, the notes on your sheet music will usually have two or three of those little eggs on the stem. It is best to use the upper one for A, the next one for B, and the lower one for C; the computer could care less, but you will find it easier to keep track

of what you are doing. If there are less than three, just go directly below to the bass clef and find a note there. If you still don't have enough, you can always use 0 to make that voice silent. Or, you can usually just let the previous note continue. If your sheet music has guitar chords - those little square grids with dots on them - above the staff, they will give you some help - if there is no guitar chord above the note you are working on, the chord has not changed and it is safe to use the previous harmony notes.

There are many other CALL SOUND routines you can use for different effects. This is similar to the one that Bill Knecht used for his hymns - I call it VIBRA.

```
105 D=1 :: V1=1 :: V2=5 :: V
3=11
1000 FOR J=1 TO T*D :: CALL
SOUND(-99,N(A),V1,N(B),V2,N(
C),V3):: CALL SOUND(-99,N(A)
*1.01,V1,N(B),V2,N(C),V3)::
NEXT J :: RETURN
```

This one I call WUBBA, for no good reason -

```
105 D=1 :: V1=1 :: V2=5 :: V
3=11
1000 FOR J=1 TO T*D :: CALL
SOUND(-99,N(A),V1,N(B),V2,N(
C),V3):: CALL SOUND(-99,N(A)
*1.01,V1,N(B),V3,N(C),V2)::
NEXT J :: RETURN
```

And this one I call TREM -

```
105 D=1 :: V1=1 :: V2=5 :: V
3=11
1000 FOR J=1 TO T*D :: CALL
SOUND(-999,N(A),V2,N(B),V2,N
(C)*1.01,V3):: CALL SOUND(-9
99,N(A)*1.01,V1,N(B),V2,N(C)
,V3):: NEXT J :: RETURN
```

I included line 105 in those, to merge in the duration and volumes along with the sound routine. Change the value of D to suit yourself, even in decimal increments such as D=1.5 .

It is easy to play a song repeatedly but with a different effect each time. Merge in VIBRA and change its line

number to 1010. You can do this by typing 1000 and FCTN X, Enter, FCTN 8 to bring it back, type over the line number, and Enter. Merge in WUBBA and change it to line 1020 in the same way, then TREM and change it to line 1030.
Add :: FOR R=1 TO 3 to the end of line 105. Put in a new line 1000 - 1000 ON R GOSUB 1010,1020,10 30 :: RETURN
And change the end of line 210 to NEXT R :: STOP.

Next time - more different effects, and autochording.

****

TI WRITER TIPS

Fm LA99er Newsletter

Nesting INCLUDE FILE's and saving paper with TI-Writer.

Page 109 of the TI-Writer Reference Guide says that the .IF command does not permit nesting of files - calling a second file from the first and a third from the second. However it can be done by using ALTERNATE INPUT - *n*. The "n" is a number from 1 to 99. ALTERNATE INPUT can be used with or without DEFINE PROMPT - .DP.
Take a blank disk and create the following six files with filenames FN1 to FN6. These files will demonstrate how .IF (INCLUDE FILE) can be accomplished by entering all or part of a .IF DSKn.FILENAME as prompted.

FILENAME FN1

```
0001 TEST 1
0002 .DP 2:ENTER "n" of DSK1.FN"n"
0003 .IF DSK1.FN*2*
```

FILENAME FN2

```
0001 TEST 2
0002 .DP 3:ENTER "FNn" of DSK1."FNn"
0003 .IF DSK1.*3*
```

FILENAME FN3

```
0001 TEST 3
0002 .DP 4:ENTER "DSKn.FNn"
0003 .IF *4*
```

FILENAME FN4

```
0001 TEST 4
0002 .DP 5:ENTER "n" of DSK"n".FN5
0003 .IF DSK*5*.FN5
```

FILENAME FN5

```
0001 TEST 5
0002 .DP 6:ENTER ".IF DSKn.FNn"
0003 *6*
```

FILENAME FN6

```
0001 TEST 6
0002 .DP 7:PRESS "FCTN 9" TO EXIT
0003 *7*
```

Enter the FORMATTER with disk in drive 1. Enter DSK1.FN1 in the initial FORMATTER prompts.

```
    TEST 1 printed
        at prompt enter "2"
    TEST 2 printed
        at prompt enter "FN3"
    TEST 3 printed
        at prompt enter "DSK1.FN4"
    TEST 4 printed
        at prompt enter "1"
    TEST 5 printed
        at prompt enter ".IF DSK1.FN6"
    TEST 6 printed
        prompt will be "PRESS FCTN 9
        to QUIT"
```

Pressing FCTN 9 returns to TI-Writer menu. This last sample can also be used (with or without a prompt) to save paper, by making the "n" ALTERNATE INPUT the last line of the file.

Incidently the ALTERNATE INPUT with DEFINE PROMPT can also be used with the HEADER or FOOTER commands when printing documents that are not in consecutive page number order.

****

DV80APPEND

BY Westley R. Richardson
Northcoast 99ers, Cleveland, Oh.
Taken from LA99ers Newsletter

The Extended Basic program, DV80APPEND, will append a Display Variable 80 (DV80) format file to the end of an existing DV80 file. One purpose of this utility program is to combine files which are too large for them to both be loaded into the TI-Writer Editor.

Since the TI-Writer uses the last line of the file to store the margin and tab settings, you may wish to load the appended to file using the Editor/Assembler Editor and then save the file back to disk to remove the tab settings line. The only caution using DV80APPEND is that if the first character of a line has an ASCII value greater than 127, then that line will not be included in the output file.

The program works best with the input file on one disk drive and the output file on another disk drive because the disk is accessed for each input and output line.

```
100 REM DV80APPEND
110 REM WESLEY R. RICHARDSON
, DECEMBER, 1990
120 REM NORTHCOAST 99ERS, CL
EVELAND, OH
130 PRINT "DSKX. FILE TO ADD
?"
140 INPUT " ":F1$
150 PRINT "DSKX.APPEND TO FI
LE?"
160 INPUT " ":F2$
170 OPEN #1:F1$,INPUT
180 PRINT "READING ";F1$
190 OPEN #2:F2$,APPEND
200 LINPUT #1:W$
210 IF EOF(1)THEN 260
220 IF ASC(W$)>127 THEN 260
230 N=N+1
240 PRINT #2:W$
250 GOTO 200
260 PRINT "CLOSING ";F1$
270 CLOSE #1
280 PRINT "CLOSING ";F2$
290 CLOSE #2
300 PRINT N;"LINES ADDED:
310 END
```

****

TIPS FROM THE TIGERCUB

No. 62

Tigercub Software
156 Collingwood Ave.
Columbus, OH 43213

Dec. 1990

My stock of Tigercub Software catalogs is depleted and it would not pay me to reprint it. Therefore I have released all copyrighted Tigercub programs, except the Nuts & Bolts Disks, for free distribution providing that no price or copying fee is charged. All of my Tigercub programs have been added to my TI-PD library and are cataloged, by category, in TI-PD catalog #4.

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 consists of 452 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 #4 listing all titles and authors, is available for $1 which is deductible from the first purchase.

According to Charles Good, running a program containing CALL SAY on a beige console without the speech synthesizer attached will cause a lockup.

On a black and silver console, there is no lockup but program execution can be greatly delayed. To avoid that, CALL PEEK(-28672,@) at the beginning of the program and add IF @=96 before each CALL SAY (remember that, IF causes program execution to skip to next program line if not true!), or IF @<>96 THEN to skip over the CALL SAYs.

In Tips #60 I presented a routine to find the lowest power of 7 which contains six 7s in sequence. My version took 24 minutes to find the answer on my TI-99/4A. Several users tried this on a Geneve. The NUTI News of the Nittany UG, Oct 1990 reports that on a 9640 (MDOS 0.97H) with TI XBasic loaded through GPL (speed 5) it ran in 11 min. 33.86 seconds, and with MYARC Advanced Basic V2.99A loaded through GPL it ran in 4 min. 58.62 seconds!

Now, from the TI*MES of England, here is a method using a level of math beyond my comprehension that will solve the problem on an ordinary TI in 6 minutes and 17 seconds!

```
100 ! FASTER WAY John Seager
110 CALL CLEAR :: DIM ELEM(2
6):: ELEM(0)=7 :: POWER,SS=0
 :: DISPLAY AT(1,1):"7 TO TH
E POWER OF"
120 ELM=SS :: SS,CARRY=0 ::
POWER=POWER+1
130 DIS$=STR$(ELEM(ELM)):: F
OR I=ELM-1 TO 0 STEP -1 :: D
IS$=DIS$&RPT$("0",10-LEN(STR
$(ELEM(I))))&STR$(ELEM(I))::
 NEXT I
140 DISPLAY AT(1,19):STR$(PO
WER);"=": : :DIS$
150 FOR I=6 TO LEN(DIS$)STEP
 6 :: IF SEG$(DIS$,I,1)<>"7"
THEN 190
160 FOR J=I-5 TO I :: IF SEG
$(DIS$,J,6)<>"777777" THEN 1
80 ELSE DISPLAY AT(24,1):"AN
```

```
Y KEY TO CONTINUE"
170 CALL KEY(0,K,S):: IF S=0
 THEN 170 :: DISPLAY AT(24,1
):: J=I
180 NEXT J
190 NEXT I
200 ELEM(SS)=ELEM(SS)*7+CARR
Y :: IF ELEM(SS+1)=0 AND ELE
M(SS)<1.E+10 THEN 120
210 CARRY=INT(ELEM(SS)/1.E+1
0):: ELEM(SS)=ELEM(SS)-CARRY
*1.E+10
220 SS=SS+1 :: GOTO 200
```

And if you think that is fast, the Autumn '90 edition of TI*MES contains a Minimemory program to solve the program in 2 SECONDS! And an assembly version that will search to the 10,000 power and find 52 strings of six 7's in an hour and a half!

Here's a puzzler for you. Can you figure out why that 1000-microsecond CALL SOUND is cut short?

```
100 CALL CLEAR
110 DISPLAY AT(12,1):"Filena
me? DSK" :: ACCEPT AT(12,14)
BEEP:F$
120 ON ERROR 130 :: OPEN #1:
"DSK"&F$ :: STOP
130 GOSUB 140 :: RETURN 110
140 CALL SOUND(1000,110,0,-4
,0):: DISPLAY AT(24,1):"CAN'
T OPEN FILE" :: RETURN
```

I recently programmed a diskfull of gospel songs, and in each one I used this formula to set up an array containing the frequencies for 3 octaves:
DIM N(36) :: F=110 :: FOR J
=1 TO 36 :: N(J)=INT(F*1.059
463094^(J-1)+.5):: NEXT J
At the end of each selection I put CALL INIT :: CALL LOAD(-31961,149) I don't remember where I learned that one, but it clears the screen, sets all colors and characters to default, deletes sprites, and looks

for a LOAD program on DSK1.

The LOAD program has a routine to play each song one after another, but one song crashed with a BAD VALUE error even though it had previously been OK. I found that this was the only song that actually used N(1). The value should have been 110 but it had somehow changed to 24263 which the program line multiplied by 2, therefore out of range.

I found that the routine was correctly giving N(1) a value of 110 the first time but after the CALL LOAD it always had the 24263 value. Substituting other values for 110, I found that any value was being multiplied by 220.5727273, rounded off.

Further experimentation revealed that the problem was being caused by the ^ (exponentiation sign, shift 6 on your keyboard, in case someone prints this through the Formatter!). So I wrote this little routine to experiment with:

```
100 FOR J=1 TO 10 :: PRINT
2^J  :: NEXT J :: CALL INIT
:: CALL LOAD(-31961,149)
```

I saved that as DSK1.TEST and then wrote another one
```
100 RUN "DSK1.TEST",
```
saved that as DSK1.LOAD, and then entered RUN "DSK1.TEST".

It printed out the proper values time after time, so I changed the 2^J to read 2^(J-1). The first time around, the first value was 1 as it should be - the computer will consider any number to the power of 0 to have a value of 1. But, the next time around, the first value was F0.57000101 !

That was not even a valid numerical representation, so I changed the formula to 2^(J-1)*2, expecting it to

crash. Instead, it gave me a value of 441.140002 !

Further experimentation showed that 2^(J-1)+1 gave a value shown as 1<1.570001.

Changing the +1 to +10 gave 1=0.570001 and to +100 gave 2<0.570001 !

So, poking a value of 149 into -31961 will cause any number taken to the power of zero to have a value of 220.5727273, which will be represented on screen in some apparently undocumented format - it's not even radix 100. I wonder if the fellows who built this computer could explain that!

ATTENTION all newsletter editors! If you print the above through the Formatter PLEASE transliterate the caret sign!

This one requires the TEII module and the Speech Synthsizer. Want to make the computer so mad it will fuss and fume and cuss and mutter? Run this program and answer the prompt with 1.
```
100 CALL CLEAR
110 OPEN #1:"SPEECH",OUTPUT
120 INPUT X
130 PRINT #1:"//"&STR$(X)&"
"&STR$(X*3.17)
140 PRINT #1:"THIS IS THE SE
CRET METHOD OF MAKING THE CO
MPUTER SPEAK IN A WHISPER"
150 GOTO 120
```

Want to make it whisper to you? Answer the prompt with 0 or -10.

Why did I get an INPUT ERROR when the strings in this routine got too long?

```
100 CALL CLEAR :: X=1
110 X=X*2 :: A$=RPT$("A",X):
: B$=RPT$("B",X):: C$=RPT$("
C",X):: D$=RPT$("D",X):: PRI
NT A$:B$:C$:D$
120 OPEN #1:"DSK1.TEST",VARI
ABLE 254,OUTPUT :: PRINT #1:
```

```
A$:B$:C$:D$ :: CLOSE #1
130 OPEN #1:"DSK1.TEST",INPU
T :: INPUT #1:A$,B$,C$,D$ ::
 PRINT A$:B$:C$:D$ :: CLOSE
#1 :: GOTO 110
```

Thanks to Irwin Hott for the answer to that one. I don't think it's in the books anywhere, but the TI won't input multiple records in a single INPUT if the total number of bytes is too high - less than 154 for two records to less than 144 for six records.

I still think computers should be fun, so here is a quickie for the kids, or for the kid in you -

```
100 PRINT TAB(9);"QUICK DRAW
": : : :" How good a gunslin
ger are":"you?": :" Can you
outdraw":"Deadeye Joe?": :
110 PRINT " Watch the countd
own from 1":"to 10.": :" Wai
t for the gun....": :" Then
hit any key FAST!! - ": :" -
and HOLD IT DOWN": :
120 PRINT " I got down to 20
 once - can":"you beat that?
": :" Press any key to start
"
130 CALL KEY(0,K,ST):: IF ST
=0 THEN 130
140 CALL CLEAR :: S@=300 ::
CALL CHAR(58,"009F9191919191
9F"):: CALL CHAR(42,"0000FCF
E171F0707")
150 CALL KEY(0,K,ST):: IF ST
=-1 THEN 150
160 CALL CLEAR :: FOR M=1 TO
 10 :: CALL HCHAR(12,16,M+48
):: FOR N=1 TO 100
170 NEXT N :: CALL KEY(0,F,X
):: IF F=70 THEN 330
180 NEXT M :: CALL CLEAR ::
FOR J=1 TO 500
190 NEXT J :: IF F=70 THEN 3
30
200 CALL KEY(0,K,ST):: IF ST
<>0 THEN 330
210 CALL HCHAR(12,16,42):: F
OR D=1 TO S@
220 NEXT D :: CALL KEY(0,Z,X
```

```
)::  IF X=0 THEN 240
230 GOTO 270
240 CALL CLEAR :: PRINT :: P
RINT "YOU'RE DEAD!"
250 FOR D=1 TO 200
260 NEXT D :: GOTO 160
270 PRINT "OUCH!" :: IF S@<5
1 THEN 290
280 S@=S@-50 :: GOTO 320
290 IF S@<31 THEN 310
300 S@=S@-5 :: GOTO 320
310 S@=S@-1
320 PRINT S@ :: GOTO 250
330 PRINT "YOU CHEATED!" ::
GOTO 150
```

I always wondered about those recipe programs. Does the cook lug the computer out to the kitchen to read the screen, or use a printer to make a hardcopy of a file that was keyed in from a hardcopy in the first place?

Anyway, some of those programs do convert quantities for different servings, so here is a little program to do that. It provides input and output in fractions instead of decimals, because that is the way recipes are written.

```
100 DISPLAY AT(3,6)ERASE ALL
:"RECIPE CONVERTER"
110 DISPLAY AT(6,1):"Enter f
ractional quantities separat
ed by a space from   whole q
uantities."
120 DISPLAY AT(9,1):"For ins
tance, to enter threeand one
-half, type 3 1/2"
130 DISPLAY AT(12,1):"Result
s will be rounded to  the ne
arest 8th."
140 DISPLAY AT(24,7):"press
any key" :: DISPLAY AT(24,7)
:"PRESS ANY KEY" :: CALL KEY
(0,K,S):: IF S=0 THEN 140
150 DISPLAY AT(12,1)ERASE AL
L:"TURN PRINTER ON!"
160 OPEN #1:"PIO" :: PRINT #
1:CHR$(27);"@" :: CALL CLEAR
170 DISPLAY AT(5,1):"Name of
 recipe?" :: ACCEPT AT(7,1):
M$ :: PRINT #1:M$;"";""
```

```
180 DISPLAY AT(3,1)ERASE ALL
:"Recipe is for how many
 servings?" :: ACCEPT AT(4,
11)VALIDATE(DIGIT)BEEP:R
190 DISPLAY AT(6,1):"You wan
t to cook how many    serving
s?" :: ACCEPT AT(7,11)VALIDA
TE(NUMERIC):S :: X=S/R
200 DISPLAY AT(10,1):"Name o
f ingredient? (just    enter
if finished)" :: ACCEPT AT(1
3,1)BEEP:A$ :: IF A$="" THEN
 STOP
210 DISPLAY AT(15,1):"Unit o
f measure?" :: ACCEPT AT(17,
1)BEEP:M$
220 ON ERROR 310 :: DISPLAY
AT(19,1):"Quantity in recipe
?" :: ACCEPT AT(21,1)BEEP:AX
$ :: A=VAL(AX$)
230 Q=X*A :: J=INT(Q):: P=Q-
J :: IF P=0 THEN X$=STR$(J):
: Y$="" :: GOTO 290
240 IF J=0 AND P<=.0625 THEN
 X$="" :: Y$="less than 1/16
" :: GOTO 290 ELSE IF P<=.06
25 THEN X$=STR$(J):: Y$="" :
: GOTO 290
250 IF P>.9375 THEN X$=STR$(
J+1):: Y$="" :: GOTO 290
260 DATA .8125,7/8,.6875,3/4
,.5625,5/8,.4375,1/2,.3125,3
/8,.1875,1/4,.0625,1/8
270 RESTORE 260
280 READ M,N$ :: IF P>M THEN
 Y$=N$ :: X$=STR$(J)ELSE 280
290 IF J<1 THEN X$=""
300 PRINT #1:A$&" "&X$&" "&Y
$&" "&M$ :: GOTO 200
310 P=POS(AX$," ",1):: Q=POS
(AX$,"/",1):: IF Q=0 THEN 34
0
320 ON ERROR 340 :: IF P=0 T
HEN A=0 ELSE A=VAL(SEG$(AX$,
1,P-1))
330 B=VAL(SEG$(AX$,P+1,Q-1-P
)):: C=VAL(SEG$(AX$,Q+1,255)
):: A=A+B/C :: RETURN 230
340 DISPLAY AT(24,1):"OOPS!
TRY AGAIN" :: CALL SOUND(1.1
10,0,-4,0):: RETURN 220
```

And here is an oldie - a utility to get the bugs out of your programs.

```
100 ! MOSQUITO #2 by Jim Pet
```

```
erson from a PEEK by Crag Mi
ller
110 CALL CLEAR :: CALL SPRIT
E(#1,42,2,100,100)
115 DISPLAY AT(22,1):"Don't
let the mosquito get":"out o
f the TV!":"Press any key -Q
UICK!"
120 RANDOMIZE :: CALL PEEK(-
31808,A,B):: CALL MOTION(#1,
A-128,B-128):: CALL KEY(0,K,
S):: IF S=0 THEN 120
130 CALL CLEAR :: CALL COLOR
(1,2,8):: CALL SCREEN(2):: C
ALL CHAR(32,"FF888888FF88888
8"):: GOTO 120
```

Long live the TI-99/4A!

Jim Peterson

The Tigercub