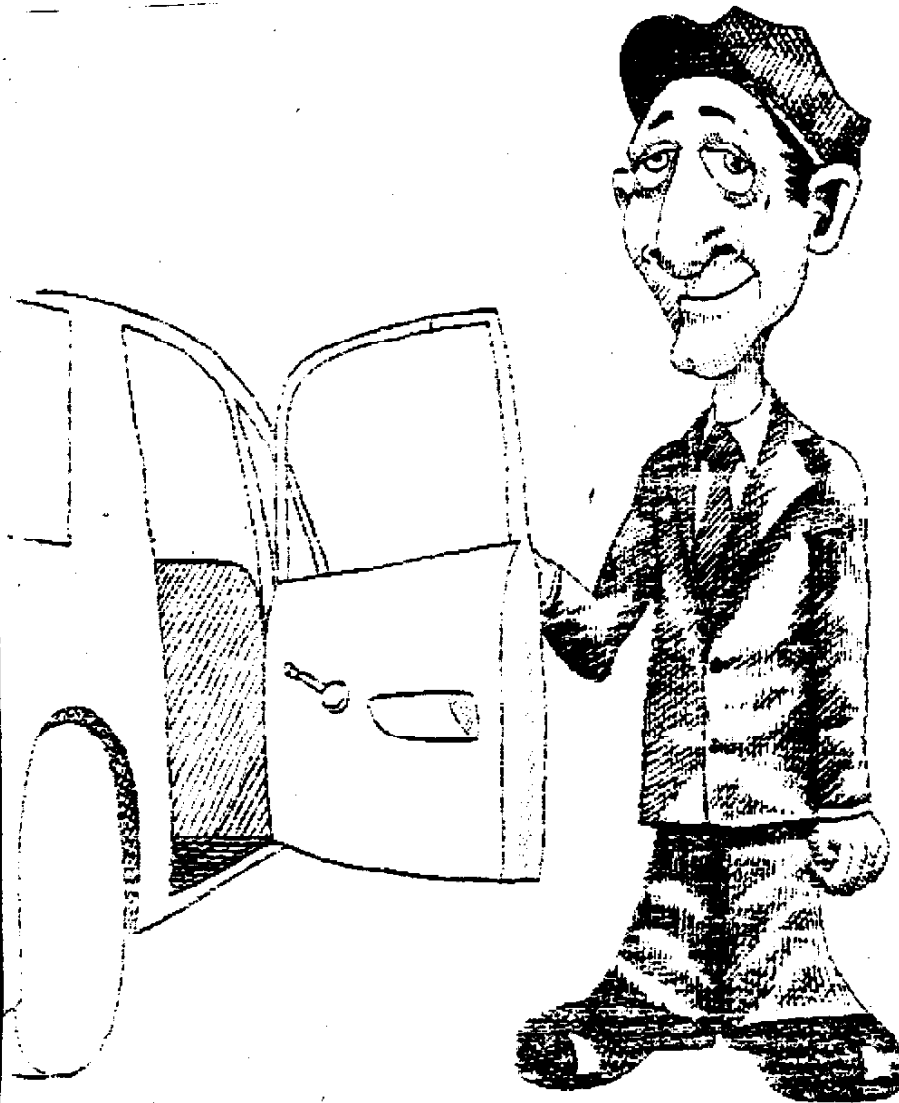


# TI - D - BITS

PHILADELPHIA AREA USERS GROUP NEWSLETTER  
COVERING THE T199/4A  
AND NYARC 9648 COMPUTERS

OCTOBER 1992

Volume 12 Number 8



TO ALL YOU  
MEMBERS THAT  
WE HAVEN'T  
SEEN FOR  
A WHILE  
"THE DOOR IS  
WIDE OPEN  
TO YOU"  
COME ON IN !

The Philadelphia Area TI-99/4A Users' Group meets twice a month. On the second and fourth saturday of the month at the Church of the ATONEMENT, 6200 Green St. Germantown (Corner of Green St and Walnut Lane) at 10 A.M. 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**

Norm Sellers..... 215-353-0475

**VICE PRESIDENT**

Allan 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 Mascaro..... 215-441-4060

**MEMBERSHIP**

Ralph Field..... 215-362-2534

**EDUCATION**

Barry Traver  
Allan Silversteen

**EQUIPMENT**

Allan Silversteen

**PROGRAM**

(OPEN)

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 and 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.

# PROGRAMMING MUSIC THE EAST WAY

## PART 4

by Jim Peterson

The first three parts of this series were written and published some time ago, so I had better review.

In Part 1, I showed you this one-line routine to set up a musical scale.

```
100 DIM N(36):: F=110 :: FOR
  J=1 TO 36 :: N(J)=INT(F*1.0
59463094 (J-1)+.5):: NEXT J
:: N(0)=40000 ::GOTO 110
101 D,T,A,B,C,V1,V2,V3,J,X,V
102 CALL SOUND
103 !P-
```

That sets up a scale of three octaves beginning with A. If you decide to change the music to a higher key, just change the 110 to 117, 123, 131, 139, 147, 156, 165, 175, 185, 196, 208 or 220. In fact, for some music you will have to change it, if the program crashes with a BAD VALUE error message.

If you have programmed the music with high notes, you can lower the key by changing 110 to 104, 98, 92, 87, 82, 78, 73, 69 or 65. Again, if you try to go too low you will get that BAD VALUE mes- sage.

I have given N(0) a value of 40000, which creates a tone too high to be heard. This can be used to silence a note, but it can also cause a crash when used with some of the following rou- tines. If you are programming three voices and want to play a single note, the easiest way is to give all three notes the same number, such as A,B,C=10. If you need a silent rest, play all the notes at an inaudible volume by V1,V2,V3=30 and then, after the GOSUB, restore their original volume by V1= (whatever is in line 110) and the same for V2 and V3.

Lines 101-103 are a pre-scan routine to start the music playing sooner. There will still be a few seconds delay while that array is set up in

line 100. You can perhaps shorten that delay slightly by changing the 36 to the highest note number you have used in programming your piece.

However, Bruce Harrison wrote for me an assembly link which eliminates any delay; this also makes it possible to change key while the music is playing. I won't list the source code here, because everyone is afraid to key in source code anyway, but it is available on my TI-PD disk #1143 and will also be on a tutorial disk on this type of music programming.

Part 2 of this series contained a listing of a program to easily give you the numbers you would need in order to key in a particular piece of sheet music. If you don't have that, you can just take a piece a paper and list the scale A Bf B C# D Ef etc., on through as many as you will need, and then number them consecutively. For the length of the notes, give the shortest note a value of 1 unless it also appears as a dotted note, in which case it must be 2, and then number the others according to their relative length - for example, 2 for a quarter note, 3 for a dotted quarter, 4 for a half note, 8 for a whole note.

Part 2 showed you how to key in single-note music, and Part 3 showed how to do 3-part harmony. To recap briefly -

First, save yourself a lot of work by identifying any groups of notes in the sheet music that are repeated two or more times. Mark them off wherever they appear. Key them in first, starting with line number 500; at the end, put RETURN. If you find another such series, label it 600 and do the same; you may find several such series. Just stay below line number 1000, which is reserved for mergeable routines. Then, while you are programming the music and come to such a series of notes, just put in GOSUB 500 or whatever.

Start keying in your music in line 120; line 110 is reserved for a line to be merged in. To key in the music, just give T the number for the length of the first note, and give A, B and C

the numbers for the melody and first and second part harmony. Then GOSUB 1000. For instance

```
T=1 :: A=23 :: B=18 :: C=12 :: GOSUB 1000
```

And for each succeeding note, give a new value to whatever changes; if T is still 1 and B and C are still the same, all you need is, for instance,

```
A=19 :: GOSUB 1000.
```

Merge in one of the following routines, put in a line 999 STOP, and after every several notes enter RUN and listen to what you have done so far, to catch any errors while it is still easier to find them.

You can merge in any of the following routines to create many different musical effects. The D in line 110 controls the tempo of the music; change it as you wish. V1, V2 and V3 are the volume (loudness) of the three voices; adjust them as you like.

Key this in and save it by SAVE DSK1.

```
110 D=500 :: V1=1 :: V2=5 :: V3=7  
CALL SOUND(D*T,N(A),V.N(B),V.N(C),V):: RETURN
```

That plays simple 3-part music, all at the same volume, which may sound rather harsh to your ears. Try changing the second V to V2 and the 3rd one to V3. Save that as PLAY2.

For a bass accompaniment in the 3rd voice, change that to

```
CALL SOUND(D*T,N(A),V1,N(B),V2,N(C)*3.75,40,-4,V3)
```

For a bass melody with accompaniment, change the A to C, V1 to V3, C to A and V3 to V1.

For the melody in two voices two octaves apart, change the C back to A and the V3 back to V1. Are you beginning to see how many different effects can be created by making changes in just this one line? Save any ones you like in merge format with

a different name for each.

Perhaps those bass notes sound too deep. Try changing the 3.75 in any of those routines to 7.5. Better yet, change it to X and add :: X=3.75 to line 110. Then you can switch back and forth in your music by simply X=7.5 or X=3.75. Getting interesting, no?

Music played in that way has a strong throbbing beat, so try this method -

```
110 D=4 :: V1=1 :: V2=5 :: V3=7  
1000 FOR J=1 TO T*D :: CALL SOUND(-4250,N(A),V1,N(B),V2,N(C),V3):: NEXT J :: RETURN
```

I'll be referring back to this one as the negative duration method. Again, you can change the tempo by changing the value of D, but sometimes not as exactly as I would like. With this method, you will find that a series of the same note runs together into a single long note. To avoid this, use different harmony notes each time, or different volumes for V2 and V3.

There's no law that says the harmony has to be lower than the melody, so try changing N(B) to N(B)\*2 or even N(B)\*4 or do the same with N(C), or both. Or, use \*X, add X=1 to line 110, and then in the middle of your music program you can switch by X=2 or X=4 (don't try 3!)

For a vibrato effect, we alternate a note with the same note multiplied by 1.01 -

```
1000 FOR J=1 TO T*D :: CALL SOUND(-4250,N(A),V1,N(B),V2,N(C),V3):: CALL SOUND(-4250,N(A)*1.01,V1,N(B),V2,N(C),V3):: NEXT J :: RETURN
```

For vibrato in the harmony rather than the melody, multiply N(C) or N(B), or both, by 1.01 instead - or multiply all three.

For a stronger vibrato, change the 1.01 to 1.02 or even 1.03. Of course, you can also multiply the harmony notes in both CALL SOUNDS by 2 or 4.

as above. Or for a "chop" effect, multiply them in one CALL SOUND but not the other. The possibilities are almost endless!

For a tremolo, we alternate the volume rather than the frequency. Add X=3 to line 110 and use this routine -

```
1000 FOR J=1 TO T*D :: CALL
SOUND(-4250,N(A),V1,N(B),V2,
N(C),V3):: CALL SOUND(-4250,
N(A),V1+X,N(B),V2,N(C),V3)::
NEXT J :: RETURN
```

You can vary the value of X as much as you want (V3+X can't total more than 30) for any amount of tremolo from a flutter to a wobble or a stutter, and you can put the +X after V1 or V2 or all three. You can even change it in the middle of your music, by X= whatever you want.

And you can multiply any or all by 1.01 for different combinations of vibrato and tremolo.

To enhance a note, play it twice in the CALL SOUND but multiply one of its voices by 1.01 -

```
110 D=4 :: V1=1 :: V2=5 :: V
3=7
1000 FOR J=1 TO T*D :: CALL
SOUND(-4250,N(A),V1,N(A)*1.0
1,V1,N(B),V2):: NEXT J :: RE
TURN
```

Of course, with this trick you can only have 2-part harmony, but you can choose to enhance the harmony rather than the melody.

Now, try combining the enhanced note with the vibrato and/or tremolo, for many more effects. For enriched vibrato, use N(A),V1,N(A)\*1.01,V1 in the first CALL SOUND and N(A)\*1.01,V1,N(A)\*1.02, V1 in the second.

The bass notes do not go well with this method because interrupting them through a loop introduces a rattle, but the baritone works well and gives a unique reedy sound. To do this, place the note you want in the 3rd position, multiply it by 7.5, give it a volume of 30, and add the -4 noise at whatever volume you want. You can also combine this with other

effects, for instance with vibrato

```
1000 FOR J=1 TO T*D :: CALL
SOUND(-4250,N(A),V1,N(B),V2,
N(C)*7.5,30,-4,V3)
1010 CALL SOUND(-4250,N(A)*1
4,V3):: NEXT J :: RETURN
```

Now for the real fun - the "piano" effects that we get by decreasing the volume gradually. This is the basic routine -

```
1000 FOR J=1 TO T*D :: CALL
SOUND(-4250,N(A),J+V1,N(B),J
+V2,N(C),J+V3):: NEXT J :: R
ETURN
```

Of course, with all of these you must also have that line 110 to define the duration and volume.

If you want a little more percussion in your piano, try this -

```
1000 FOR J=1 TO T*D :: CALL
SOUND(-4250,N(A),J*1.5,N(B),
J*1.5,N(C),J*1.5):: NEXT J :
: CALL SOUND(-4250,N(A),15,N
(B),15,N(C),15):: RETURN
```

And, of course, all those tricks we learned above - vibrato, tremolo, baritone, enhanced, high harmony, chop - can also be used with piano. This will give you the vibrato -

```
1000 FOR J=1 TO T*D :: CALL
SOUND(-4250,N(A),J+V1,N(B),J
+V2,N(C),J+V3):: CALL SOUND(
-4250,N(A)*1.01,J+V1,N(B),J+
V2,N(C),J+V3):: NEXT J :: RE
TURN
```

And an increasing tremolo -

```
1000 FOR J=1 TO T*D :: V=J*2
:: CALL SOUND(-4250,N(A),J,
N(B),J,N(C),J):: CALL SOUND(
-4250,N(A),V,N(B),V,N(C),V):
: NEXT J :: RETURN
```

And just one more, the "reverse piano" with an increasing volume

```
1000 FOR J=T*D TO 1 STEP -1
:: CALL SOUND(-4250,N(A),J+V
1,N(B),J+V2,N(C),J+V3):: CAL
L SOUND(-4250,N(A),J+V1,N(B)
,J+V2,N(C),J+V3):: NEXT J ::
```

RETURN

By the time you get through exploring all the possible combinations of those, you should have a hundred ways of making music. Save each one you like, complete with line 110, in merge format, so you can try them all with each piece of music you create.

I had intended this to be the last part of this series, but I still haven't told you about autochording, so there will have to be one more.

\*\*\*\*\*

**SCSI**

**A HARD and FLOPPY DISK CONTROLLER for the TI 99/4A And MYARC GENEVE By Western Horizon Technologies**

This Artical taken from the: LONG ISLAND USER GROUP NEWSLETTER

This advanced new peripheral for your TI 99/4A or MYARC GENEVE will expand your storage capacity to hundereds of megabytes. This high performance disk drive interface allows you to connect up to any combination of 7 SCSI Hard and Floppy drives with any capacity up to the SCSI limitations. You can connect floppy drives, both 3.5 and 5.25 inch, with current capacities up to .4 megabytes (unformatted). Or connect a Winchester drive with an astonishing 1.6 Gigabytes of hard disk storage. You can even connect a CD ROM player for access to hundreds of pictures and sounds.

This new peripheral will read and write TI floppies in all current formats as well as PC compatible floppies! That's right, you get PC TRANSFER (c) capabilities BUILT IN! You can now exchange data DIRECTLY with an IBM PC or compatible without having to convert!

Expand your disk capacity with this new SCSI Controller designed by WHT available SOON from Bud Mills Services. PRELIMINARY PRICE -- \$170 IN US.

\*\*\*\*\*

**\*\*\* FOR SALE \*\*\***

**EQUIPMENT**

- TI Expansion Box
- 32K Memory Card
- Disk Controller Card
- Internal 5 1/4 Floppy Drive
- Dust Cover
- RF Unit
- Power Supply
- 2 TI Cassette Recorders

**MODULES**

- TI XBASIC
- Personal Record Keeping and Manual
- Personal Report Generator and Manual
- Statistics and Manual
- MULTIPLAN Module and Manual
- Plus 30 additional Modules (Mostly Games)
- A 16 K Ram Expansion Module
- 2 Containers of 5 1/4 Floppy disks (A variety of home brews and commercial software)
- About 50 Cassetts (Mostly Games)
- A variety of Intro. Books and Manuals to the TI System, Power supply, TV hook up cables.
- Asking \$300.00 for WHOLE LOT or BEST OFFER!! Will sell individual pieces.
- Contact: BILL BACON at 215-441-5608 Evenings or 215-951-5655 Business Hrs.

## TI-BASE Tutorial 3 Page 4

NorthCoast 99'ers (C) Martin A. Saoley

INITPR is merely another demonstration of what you can do with DOCASE and WHILE statements. Many of the ideas I have presented in my tutorials can be done in other ways. Some of the others ways may turn out to be more efficient or more convenient to use. I still consider myself to be a beginner at TI-Base so I anticipate changing my ideas on how to optimize program power and minimize program run time. As I stated in the last tutorial, I would appreciate letters or notes from TI-Base users with comments, tips or questions on this subject. I do not have the time to write back to you. In many cases I find myself rushing to the last minute before the newsletter deadline to finish the months tutorial.

```

100 ! ##### TIB->DV/80 #####
110 ! (C) 1988 Martin A. Saoley
120 !
130 ! Extended Basic program to read TIB I/F40 files
140 ! and write D/V80 files for TIM or FunnelWeb.
150 !
160 ! You must add one blank space to the beginning
170 ! of every line in the TIB I/F40 file.
180 ! After transfer, check all lines for any
190 ! missing characters, especially the end.
200 !
210 CALL CLEAR :: CALL SCREEN(6)
220 PRINT " Enter INPUT File ALL CAPS"
230 PRINT " Example: DSK1.OPERATOR/C"
240 INPUT " " :IN$
250 IF LEN(IN$)>12 THEN OUT$=SEGS(IN$,1,12)&"$DV"
260 IF LEN(IN$)<13 THEN OUT$=IN$&"$DV"
270 PRINT "OUT File= ";OUT$ :: PRINT
280 INPUT " Is that OK Y/N ":ANS$
290 IF ANS$="N" OR ANS$="n" THEN 210
300 OPEN #1:IN$,INTERNAL,FIXED 40,INPUT :: LN=40
309 ! OPEN #1:IN$,DISPLAY ,VARIABLE 80,INPUT :: LN=80
310 OPEN #2:OUT$,DISPLAY ,VARIABLE 80,OUTPUT
320 IF EOF(1)THEN CLOSE #1 :: CLOSE #2 :: GOTO 480
330 !
340 INPUT #1:A$
350 PRINT A$
360 FOR I=1 TO LN
370 T$=SEGS(A$,I,1)
380 ON ERROR 440
390 IF ASC(T$)>126 THEN T$=" "
400 IF ASC(T$)<32 THEN T$=" "
410 B$=P$
420 IF I<1 THEN P$=T$ ELSE P$=B$&T$
430 NEXT I
440 PRINT #2:P$
450 P$=""
460 !
470 GOTO 320
480 CALL CLEAR :: PRINT " ### FINISHED ###" : : :
490 INPUT " Quit Program Y/N ":BT$
500 IF BT$="Y" OR BT$="y" THEN STOP ELSE GOTO 210
510 ! ##### TIB->DV/80 #####
520 END

```

I have also been recommending the use of FunnelWeb in the non-word wrap mode. I have had some problems with this procedure. I figured if I was having a problem, someone else must be having the same problem. The problem is hidden characters in the CF. In most cases I am in a hurry to produce code (write programs or CFs). Many times I jump into the wrong editor mode and start typing. In many instances this will not be a problem. In FunnelWeb pressing CTRL O will throw you into non-wordwrap mode, which is the same as the E/A Editor. However, if you hit the CTRL key and some other key at the same time while you are still in wordwrap mode, you can insert characters which are invisible on the screen but do crazy things when the CF is run. At one point I wasted more precious time than I could afford trying to find one of these invisible little land mines. I remembered a little Extended Basic program I had written for another task several weeks earlier. At that time I wanted to convert several Command Files (CFs) to DV/80 files so I could print them out and study them more carefully. The program I wrote was TIB->DV/80 which is listed on this page. I think some of you may get some use out of it. As is is now, it will read an I/F 40 file (like a CF), and write it to a D/V80 file for FunnelWeb. There is one thing you must do first. A control code in CFs causes the loss of the first character in every line. You can overcome this by loading the CF into TIB using MODIFY COMMAND (filename). Pressing FCTN 2 for insert mode, which stays on until you press FCTN 2 again. Then add one blank space to the beginning of every line. When you run my program the blank space will be lost instead of something you need. If you want to run the program on a DV/80 file, remove the exclamation point from line 309 and place one in front of line 300. Adding the space is only necessary with D/F40 files, not D/V80. The program will check every character in the file and will kick out all characters below 32 or above 126. That includes those invisible land mines in your CF. Unfortunately you will have to replace any printer controls.

Well, I'm running out of space and my mind is shot, so I'd like to say a couple more things and this one is finished. First I'd like to thank the people of the NorthCoast 99'ers for allowing me the space in their newsletter to write this tutorial, and a lot of miscellaneous articles in the past. The NorthCoast members are a great group of people. I'd like to throw in the fact that any TI 99/4A owner in the continental US can join the NorthCoast 99'ers for only \$15.00 per year. As you can imagine \$15.00 is little more than the cost of printing and mailing this great newsletter. If you would like to send your checks to me (Payable to the NorthCoast 99'ers UG), I will expedite your membership. Also any comments on the TIB column can be sent to Martin A. Saoley, 6149 Brycon Drive, Mentor, Ohio, 44060.

I am going to announce at this time that I will produce a TIB help disk. The disk will be a floppy and contain all of the tutorials and Command Files to date plus anything else I think may be helpful. I already have 390 sectors of tutorials. For this I would like a donation of \$3.00 to cover the Disk, Mailer, Postage, Handling and wear on my disk drives. Please make these checks payable to Martin A. Saoley at the previous address, and make two checks if you want the help disk and a membership.

**TI-BASE - From INSCEBOT  
TUTORIAL 4 By Martin Smoley  
NorthCoast 99'ers - Nov. 8, 1988  
Copyright 1988 By Martin A. Smoley**

I am reserving the copyright on this material, but I will allow the copying of this material by anyone under the following conditions. (1) It must be copied in its entirety with no changes. (2) If it is retyped, credit must be given to myself and the NorthCoast 99ers, as above. (3) The last major condition is that there may not be any profit directly involved in the copying or transfer of this material. In other words, Clubs can use it in their newsletters and you can give a copy to your friend as long as its free.

```
* Command File MOVED1      10/27/88
* Save as MOVED1/C
* Move Data from TNames to NEWNames
*
```

```
CLEAR
CLOSE ALL
* SET TALK OFF
SET RECNUM OFF
SET HEADING OFF
SELECT 2
USE NEWNames
TOP
SELECT 1
USE TNames
TOP
LOCAL NUMT N 4 0
REPLACE NUMT WITH 1
DO DSK2.MOVED2
SET RECNUM ON
SET HEADING ON
SET TALK ON
RETURN
```

```
* Command File MOVED2      10/27/88
* Save as MOVED2/C
*
```

```
WHILE .NOT. (EOF)
SELECT 2
APPEND BLANK
REPLACE 2.NM WITH NUMT
REPLACE 2.LN WITH 1.LN
REPLACE 2.FN WITH 1.FN
REPLACE 2.MI WITH 1.MI
REPLACE 2.SA WITH 1.SA
REPLACE 2.CT WITH 1.CT
REPLACE 2.ST WITH 1.ST
REPLACE 2.ZP WITH 1.ZP
REPLACE 2.PH WITH 1.PH
REPLACE 2.XP WITH 1.XP
REPLACE 2.GP WITH 1.GP
REPLACE 2.ID WITH 1.ID
REPLACE NUMT WITH NUMT + 1
SELECT 1
MOVE
ENIWHILE
CLOSE ALL
RETURN
```

Here we are again TI-Base users. In this months tutorial I intend to back track, back paddle, and change my mind on a lot of things. Last month I said I almost had the manual wrapped up and I could stop writing large tutorials. I will attempt to cut the tutorial size, but I am discovering things that people either don't understand or don't even realize that TIB can do. I will attempt to remember that every item we cover is probably brand new to you, the reader. I started to fall into that trap where the writer (me), thinks that what he is saying should be completely clear to everyone. So, I'll slow down and try to explain things more clearly.

The Command Files (CFs) on this page answers a question I received and cover some new points in the manual. The question was, "I completely set up a data base and typed in 100 names and addresses. I then realized I needed one more field and one field that was C)haracter should have been N)umeric." I will attempt to cover this problem and a multitude of other things at the same time, because they all work together. So please try to bear with me for a while. Let's work with TNames because we already have it typed in. Next, I'm going to switch the wrong field problem to, "a N)umeric field that should have been C)haracter" (FEL). So, let's say you just typed in TNames and you have entered 100 names. We only have 5 names in TNames but the CFs (MOVED1 and MOVED2) won't care if there are 5, 100 or 999 names in the database. "Now!", when you CREATED TNames you typed in the information listed below. As you can see the first 10 items are C)haracter type fields and the last item is a N)umeric type field. "Look it over."

arrows to move, enter to advance  
FIELD DESCRIPTOR TYPE WIDTH DEC

FIELD	DESCRIPTOR	TYPE	WIDTH	DEC
1	LN	C	15	
2	FN	C	15	
3	MI	C	2	
4	SA	C	25	
5	CT	C	20	
6	ST	C	2	
7	ZP	C	5	
8	PH	C	12	
9	XP	C	5	
10	GP	C	5	
11	ID	N	7	0

[ TNames STRUCTURE ]

Now, after entering all those names I decided that I should have put in another field. The field I want should be before the first field in this DB (before LN). I want it to be a number or N)umeric field, and hold a number up to 999 with no decimal places. Since this is a number for each name and address record, I'd like the first record to start with 1 instead of TIBs setup which is zero (0), and I'd like the computer to put the numbers in for me. Next I decided I wanted the last field to be a C)haracter field instead of a N)umeric field, as it is now. If you use MODIFY STRUCTURE as described in the manual, adding a field will destroy the data and the names will be lost. So we'll do it another way. Remember, this is a tutorial and this demonstrates programming techniques. The idea of adding a field or changing another doesn't have to be logical.



TIPS FROM THE TIGERCUB

#43

Copyright 1987

TIGERCUB SOFTWARE  
156 Collingwood Ave.  
Columbus, OH 43213

Distributed by Tigercub Software to TI-99/4A Users Groups for promotional purposes and in exchange for their newsletters. May be reprinted by non-profit users groups, with credit to Tigercub Software.

Over 130 original programs in Basic and Extended Basic, available on cassette or disk, now reduced to just \$2.00 each, plus \$1.50 per order for cassette or disk and PP&M. Cassette programs will not be available after my present stock of blanks is exhausted. Descriptive catalogs, while they last, \$1.00 which is deductible from your first order.

Tigercub Full Disk Collections, reduced to \$10 postpaid. Each of these contains either 5 or 6 of my regular \$2 catalog programs, and the remaining disk space has been filled with some of the best public domain programs of the same category. I am NOT selling public domain programs - they are a free bonus!

TIGERCUB'S BEST, PROGRAMMING TUTOR, PROGRAMMER'S UTILITIES, BRAIN GAMES, BRAIN TEASERS, BRAIN BUSTERS!, MANEUVERING GAMES, ACTION REFLEX AND CONCENTRATION, TWO-PLAYER GAMES, KID'S GAMES, MORE GAMES, WORD GAMES, ELEMENTARY MATH, MIDDLE/HIGH SCHOOL MATH, VOCABULARY AND READING, MUSICAL

EDUCATION, KALEIDOSCOPIES AND DISPLAYS

NUTS & BOLTS (No. 1), a full disk of 100 Extended Basic utility subprograms in merge format, ready to merge into your own programs. Plus the Tigercub Menuloader, a tutorial on using subprograms, and 5 pages of documentation with an example of the use of each subprogram. Reduced to \$15.00 postpaid. NUTS & BOLTS NO. 2, another full disk of 108 utility subprograms in merge format, all new and fully compatible with the last, and with 10 pages of documentation and examples. Also \$15 postpaid.

\*\*\*\*\*  
\* NUTS & BOLTS #3 is now \*  
\* ready, another full disk \*  
\* of 140 new merge-format \*  
\* utility subprograms, all \*  
\* compatible with the pre- \*  
\* vious. With 11 pages of \*  
\* documentation, \$15 ppd. \*  
\*\*\*\*\*

TIPS FROM THE TIGERCUB, a full disk containing the complete contents of this newsletter Nos. 1 through 14, 50 original programs and files, reduced to \$10 ppd. TIPS FROM THE TIGERCUB VOL. 2, another diskfull, complete contents of Nos. 15 through 24, over 60 files and programs, also just \$10 TIPS FROM THE TIGERCUB VOL. 3, another 62 programs, tipe and routines from Nos. 25 through 32, \$10 postpaid. TIPS FROM THE TIGERCUB VOL. 4, another 48 programs and files from issues 33 through 41, also \$10 postpaid.

If you have as much trouble as I do, trying to get the strip labels lined up in the printer, you'll

like this one -

```
100 DISPLAY AT(4,7)ERASE ALL
:"TIGERCUB LABELER": : : : "
This label maker will allow"
:"you to specify different":
"printer codes for each line
"
110 DISPLAY AT(11,1):"of a 5
-line label.": : : " You may
stop the program": "while lab
els are printing": "by pressi
ng any key, turn"
120 DISPLAY AT(17,1):"off th
e printer to adjust": "the la
bels, turn it back on,": "and
press any key to con-": "tin
ue printing."
130 DISPLAY AT(23,1): "Printe
r designation?": "PIO" :: ACC
EPT AT(24,1)SIZE(-28)BEEP:PR
$ :: OPEN #1:PR$ :: P$,E$,DS
$,CEN$="Y" :: DW$,L$,SS$,U$=
"N" :: P=1
140 CALL CHAR(95,"FF")
150 FOR J=1 TO 5 :: CALL KEY
(3,K,S)
160 DISPLAY AT(2,1)ERASE ALL
:"Line #";J;" - PRINT? "&P$
:: CALL QUERY(2,20,P$):: IF
P$="N" THEN L$(J)=" " :: GOTO
360
170 IF J>1 THEN DISPLAY AT(4
,1): "Change codes? N" :: CAL
L QUERY(4,15,Q$):: IF Q$="N"
THEN 300
180 DISPLAY AT(4,1): "Print p
itch? ";P:" (1)pica": " (2)el
ite": " (3)condensed" :: ACCE
PT AT(4,15)SIZE(-1)VALIDATE(
"123"):P
190 CI=(P=1)*-10+(P=2)*-12+(
P=3)*-17 :: L$(J)=CHR$(27)&
B"&CHR$(P):: DISPLAY AT(5,1)
: "" : "" : ""
200 DISPLAY AT(6,1): "Double
width? "&DW$ :: CALL QUERY(6
,15,DW$):: IF DW$="Y" THEN C
I=CI/2 :: L$(J)=L$(J)&CHR$(1
4)ELSE L$(J)=L$(J)&CHR$(20)
210 DISPLAY AT(8,1): "Italics
? "&I$ :: CALL QUERY(8,10,I$
):: IF I$="Y" THEN L$(J)=L$(
J)&CHR$(27)&"4" ELSE L$(J)=L
$(J)&CHR$(27)&"5"
220 DISPLAY AT(10,1): "Supers
cript? "&SS$ :: CALL QUERY(1
```

```

0,14,SS$):: IF SS$="Y" THEN
L$(J)=L$(J)&CHR$(27)&CHR$(83
)&CHR$(0)ELSE L$(J)=L$(J)&CH
R$(27)&CHR$(84)
230 IF SS$="Y" THEN 250
240 DISPLAY AT(12,1):"Double
-strike? "&DS$ :: CALL QUERY
(12,16,DS$):: IF DS$="Y" THE
N L$(J)=L$(J)&CHR$(27)&"G" E
LSE L$(J)=L$(J)&CHR$(27)&"H"
250 IF PK>1 OR SS$="Y" THEN
270 :: DISPLAY AT(14,1):"Emp
hasized? "&ES$ :: CALL QUERY(
14,13,ES$)
260 IF ES$="Y" THEN L$(J)=L$(
J)&CHR$(27)&"E" ELSE L$(J)=L
$(J)&CHR$(27)&"F"
270 DISPLAY AT(16,1):"Underl
ine? "&US$ :: CALL QUERY(16,1
2,US$)
280 IF US$="N" THEN L$(J)=L$(
J)&CHR$(27)&CHR$(45)&CHR$(0)
290 DISPLAY AT(18,1):"Center
text? Y" :: CALL QUERY(18,1
4,CEN$)
300 DISPLAY AT(18,1):"Type 1
ine";J;". Enter each":"scree
n line, enter again":"when d
one." :: DISPLAY AT(22,1):RPT
T$("_",INT(CI*3.5)):: R=21 :
: CALL KEY(5,K,S)
310 ACCEPT AT(R,1):M$ :: IF
M$="" THEN 320 :: A$=A$&M$ :
: R=R+1 :: GOTO 310
320 IF LEN(A$)>INT(CI*3.5)TH
EN DISPLAY AT(16,1):"LINE TO
O LONG!" :: CALL SOUND(300,1
10,0,-4,0):: A$="" :: R=21 :
: GOTO 310
330 L=LEN(A$):: IF US$="Y" TH
EN A$=CHR$(27)&CHR$(45)&CHR$(
1)&A$&CHR$(27)&CHR$(45)&CHR
$(0)
340 IF CEN$="Y" THEN A$=RPT$(
" ",(INT(CI*3.5)-L)/2)&A$
350 L$(J)=L$(J)&A$ :: A$=""
360 NEXT J
370 DISPLAY AT(12,1)ERASE AL
L:"Print how many?" :: ACCEP
T AT(12,17):N
380 FOR J=1 TO N :: FOR K=1
TO 6 :: PRINT #1:L$(K):: NEX
T K
390 CALL KEY(0,K,S):: IF S=0
THEN 410 ELSE CLOSE #1
400 CALL KEY(0,K1,S1):: IF S
1<1 THEN 400 ELSE OPEN #1:PR
    
```

```

$
410 NEXT J
420 DISPLAY AT(12,8)ERASE AL
L:"Another?" :: CALL QUERY(1
2,17,Q$):: IF Q$="N" THEN ST
OP ELSE 150
430 SUB QUERY(R,C,Q$):: ACCE
PT AT(R,C)SIZE(-1)VALIDATE("
YN")BEEP:Q$ :: SUBEND
    
```

More peculiarities of the TI computer -

```

90 CALL CLEAR :: PRINT TAB(7
);"SPRITE PUZZLE #1":
from Tigercub"
100 PRINT "A non-existent sp
rite can be":"created by CAL
L MOTION.": "It apparently
starts in"
110 PRINT "dot-row 1, dot-co
lumn 1, and":"has color 1, b
ut its pattern":"is not that
of any ASCII!"
120 !by Jim Peterson
130 FOR CH=0 TO 255 :: PRINT
CHR$(CH):: NEXT CH
135 PRINT "CALL MOTION(#1,5,
5):: CALL COLOR(#1,16):: CAL
L MAGNIFY(4)"
140 CALL MOTION(#1,5,5):: CA
LL COLOR(#1,16):: CALL MAGNI
FY(4)
150 GOTO 150
    
```

And another -

```

100 DISPLAY AT(3,5)ERASE ALL
:"SPRITE PUZZLE #2":
from Tigercub"
110 DISPLAY AT(7,1):"Non-exi
stent sprites can be":"creat
ed by CALL COLOR.": "Their
existence can be con-"
120 DISPLAY AT(11,1):"firmed
by CALL COINC, but":"CALL P
OSITION reports that":"they
have no position!"
130 CALL COLOR(#1,16):: CALL
COLOR(#2,16)
140 CALL COINC(#1,#2,1,X)::
DISPLAY AT(15,1):"COINC #1.#
2=";X :: CALL POSITION(#1,X,
Y)
150 CALL POSITION(#1,X,Y)::
DISPLAY AT(17,1):"POSITION #
1=";X;Y
    
```

```

160 CALL POSITION(#2,X,Y)::
DISPLAY AT(19,1):"POSITION #
2=";X;Y
170 IF FLAG=1 THEN 140 :: FL
AG=1
180 DISPLAY AT(21,1):"PRESS
ANY KEY"
190 CALL KEY(0,K,S):: IF S=0
THEN DISPLAY AT(21,1):"pres
s any key" :: GOTO 180
200 DISPLAY AT(21,1):"Until
they're set in motion!"
210 CALL MOTION(#1,5,5):: CA
LL MOTION(#2,-5,-5):: GOTO 1
50
    
```

If you have the Terminal Emulator II, Speech Synthesizer, and a pre-schooler in the house, this will help him to grasp the idea of spelling as well as letter recognition and keyboard familiarization-

```

100 REM PRE-SPELLER BY JIM
PETERSON
110 REM TI BASIC WITH TERMI
NAL EMULATOR II AND SPEECH S
YNTHESIZER
120 CALL CLEAR
130 DIM M$(100),S$(100)
140 OPEN #1:"SPEECH".OUTPUT
150 PRINT " PRE-SPELL
ER":::::
160 PRINT "TYPE WORDS TO PRA
CTICE":"TYPE 'END' WHEN FIN
ISHED"
170 X=X+1
180 INPUT M$(X)
190 IF M$(X)="END" THEN 380
200 PRINT #1:M$(X)
210 PRINT "PRONUNCIATION OK?
(Y/N)"
220 CALL KEY(3,K,S)
230 IF S<1 THEN 220
240 IF K=78 THEN 280
250 IF K<>89 THEN 220
260 S$(X)=M$(X)
270 GOTO 170
280 PRINT "TRY SPELLING PHON
ETICALLY"
290 INPUT S$(X)
300 PRINT #1:S$(X)
310 PRINT "PRONUNCIATION OK?
(Y/N)"
320 CALL KEY(3,K,S)
    
```

```

330 IF S<1 THEN 320
340 IF K=89 THEN 170
350 IF K>78 THEN 320
360 PRINT "TRY AGAIN"
370 GOTO 290
380 CALL CLEAR
390 FOR J=1 TO X-1
400 PRINT #1:"CAN YOU SPELL THIS?"
410 FOR A=1 TO LEN(M$(J))
420 CALL HCHAR(12,8+A,ASC(SEG$(M$(J),A,1)))
430 NEXT A
440 FOR B=1 TO LEN(M$(J))
450 CALL KEY(3,K,S)
460 IF (S<1)+(K=32)THEN 450
470 IF K=ASC(SEG$(M$(J),B,1))THEN 500
480 GOSUB 640
490 GOTO 450
500 C$=C$&&CHR$(K)
510 CALL HCHAR(14,8+B,K)
520 NEXT B
530 IF C$>M$(J)THEN 640
540 PRINT #1:S$(J)
550 FOR D=1 TO 500
560 NEXT D
570 PRINT #1:"VEREE GOOD"
580 FOR D=1 TO 500
590 NEXT D
600 C$=""
610 CALL HCHAR(12,1,32,100)
620 NEXT J
630 GOTO 390
640 PRINT #1:"NO THAT IS NOT RIGHT"
650 PRINT #1:"TRY AGAIN"
660 RETURN

```

And, a simple little game that is a bit different than any I've seen -

100 !FORMATION by Jim Peters on - use the S and D keys  
110 CALL CLEAR :: CALL CHAR(100,"381010FEFE383810103838F

```

EFE10103838"):: CALL SCREEN(5):: CALL MAGNIFY(2):: RANDO MIZE
120 V.W.P=0 :: FOR J=1 TO 7 :: CALL SPRITE(#J,100,7,1,250*RND+1,10,4):: FOR D=1 TO 100 :: NEXT D :: NEXT J :: CALL SPRITE(#11,101,16,160,128)
130 CALL KEY(3,K,S):: W=W+1 :: IF W=150 THEN 170 ELSE IF W=300 THEN 180 ELSE IF K=68 THEN V=V+2+(V>125)*2 ELSE IF K=83 THEN V=V-2-(V<-125)*2
140 IF P=0 THEN CALL MOTION(#11,0,V)ELSE IF P=1 THEN CALL MOTION(#11,0,V,#12,0,V)ELSE CALL MOTION(#11,0,V,#12,0,V,#13,0,V)
150 CALL COINC(ALL,A):: IF A=0 THEN 130
160 CALL SOUND(1000,-4,0):: H=MAX(H,W):: DISPLAY AT(23,1):"SCORE";W:"HIGH SCORE";H :: CALL DELSPRITE(ALL):: GOTO 120
170 P=1 :: CALL POSITION(#11,R,C):: CALL SPRITE(#12,101,16,160,C-40-(C<40)*256):: GO TO 140
180 P=2 :: CALL POSITION(#11,R,C):: CALL SPRITE(#13,101,16,160,C+40+(C>216)*256):: GOTO 140

```

If you can't figure out where all the money goes, this may be an eye-opener -

100 DISPLAY ERASE ALL AT(3,5):"THE COST OF CREDIT" ! by Jim Peterson  
110 S,T,X=0 :: DISPLAY AT(8,1):"AMOUNT OF PURCHASE?" :: ACCEPT AT(8,21):A :: B,T=A :: DISPLAY AT(10,1):"CREDIT CARD INTEREST RATE?" :: ACCEP

```

T AT(11,1):R
120 DISPLAY AT(13,1):"SAVING S ACCOUNT INT. RATE?" :: ACCEPT AT(14,1):SR
130 X=X+1 :: I=B*R/100/12 :: B=B+I :: T=T+I :: P=B/10 :: B=B-P :: S=S+P+S*SR/100/12 :: IF S<A THEN 130
140 D$="$"&STR$(INT((T-A+S-A+.5)*100)/100)
150 DISPLAY AT(17,1):"If you had saved the amount":"of your minimum 10% of the":"balance credit card payment":"each month for";X;"months,"
160 DISPLAY AT(21,1):"and used it to pay cash, you":"would have saved ";D$ :: GOTO 110

```

And this is one of the handiest routines I've seen in a long time -

10 !TURNS ALL NUMERALS AND PUNCTUATION WHITE! BY HARRY WILHELM IN TWIN TIERS UG NEWS LETTER  
20 !TURN IT OFF BY CALL LOAD(-31804,0)::TURN IT ON BY CALL LOAD(-31804,63)  
100 CALL INIT  
110 CALL LOAD(16128,2,224,38,0,2,0,8,17,2,1,63,36,2,2,0,3,4,32,32,36,2,224,131,192,3,128)  
120 CALL LOAD(16164,240,240,240)  
130 CALL LOAD(-31804,63)

Memory full

Jim Peterson