

# TI - D - BITS

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

## NOVEMBER 1992

Volume 12 Number 9

Happy Thanksgiving



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 Greene St. Germantown (Corner of Greene 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 5 AND FINAL

by Jim Peterson

In previous installments I have shown you how to program music by an easy method which requires you to specify a duration or a frequency only when it changes from one note to the next. Now, here is an even easier method - auto chording.

With this method, you do not have to key in the accompaniment - you just specify the chord and GOSUB to the proper line to play the type of chord.

Almost all sheet music has guitar chords printed above the upper staff - those little 6x4 grids with black dots on them. And those guitar chords are always labeled with the name of the chord they represent.

The most common chord is a major chord, represented by a letter - A, C or whatever, or a letter followed by a flat or sharp sign. For those, use GOSUB 1000. The second most common chord is the 7th chord, which has the letter followed by a 7, such as C7. For those, GOSUB 1100.

You might come across a minor chord, denoted by a small m after the letter, such as Cm. In that case, GOSUB 1200. And for a minor 7th, such as Cm7, GOSUB 1300.

There are many more complex chords, but I have not tried to allow for them all in this easy method. If you come to one of them, just try playing on through with the previous chord - it will usually sound alright.

To program music in this way, use the scale that I showed you in Part 1, but you will probably have to set the starting frequency considerably higher than 110. Merge in one or the other of the following routines, then program the music just as I showed you before, but only A and B. Give A the number for the melody and B the number for the chord, then GOSUB to the proper line number for that type of chord. If the next note does not have a guitar chord above it, it is the same chord so you do not have to give B a value again, just GOSUB to the same line number.

Now, here is the first routine, to

play simple harmony. Let me give you a tip to save you some time. When you are keying in a series of program lines which are all nearly the same, key in the first one, Enter it, then use FCTN 8 to bring it back to the screen. Use the editing keys to change the line number and make other necessary changes. Enter it, use FCTN 8 to bring it back, etc.

```

110 D=3 :: V1=1 :: V2=9 :: V
3=9
1000 X=X+1+(X=4)*4 :: ON X G
OSUB 1010,1020,1030,1040 ::
RETURN
1010 FOR J=1 TO T*D :: CALL
SOUND(-999,N(A),V1,N(B),V2,N
(B)/1.585,V3):: NEXT J :: RE
TURN
1020 FOR J=1 TO T*D :: CALL
SOUND(-999,N(A),V1,N(B),V2,N
(B)/1.334,V3):: NEXT J :: RE
TURN
1030 FOR J=1 TO T*D :: CALL
SOUND(-999,N(A),V1,N(B),V2,N
(B)/2,V3):: NEXT J :: RETURN
1040 FOR J=1 TO T*D :: CALL
SOUND(-999,N(A),V1,N(B)/1.58
5,V2,N(B)/1.334,V3):: NEXT J
:: RETURN
1100 X=X+1+(X=9)*4 :: ON X G
OSUB 1110,1120,1130,1140 ::
RETURN
1110 FOR J=1 TO T*D :: CALL
SOUND(-999,N(A),V1,N(B)/1.49
7,V2,N(B)/1.585,V3):: NEXT J
:: RETURN
1120 FOR J=1 TO T*D :: CALL
SOUND(-999,N(A),V1,N(B)/1.49
7,V2,N(B)/1.334,V3):: NEXT J
:: RETURN
1130 FOR J=1 TO T*D :: CALL
SOUND(-999,N(A),V1,N(B)/1.49
7,V2,N(B)/2,V3):: NEXT J ::
RETURN
1140 FOR J=1 TO T*D :: CALL
SOUND(-999,N(A),V1,N(B)/1.58
5,V2,N(B)/1.334,V3):: NEXT J
:: RETURN
1200 X=X+1+(X=4)*4 :: ON X G
OSUB 1210,1220,1230,1240 ::
RETURN
1210 FOR J=1 TO T*D :: CALL
SOUND(-999,N(A),V1,N(B),V2,N
(B)/1.679,V3):: NEXT J :: RE
TURN
1220 FOR J=1 TO T*D :: CALL
SOUND(-999,N(A),V1,N(B),V2,N
(B)/1.334,V3):: NEXT J :: RE

```

```

TURN
1230 FOR J=1 TO T*D :: CALL
SOUND(-999,N(A),V1,N(B),V2,N
(B)/2,V3):: NEXT J :: RETURN
1240 FOR J=1 TO T*D :: CALL
SOUND(-999,N(A),V1,N(B)/1.67
9,V2,N(B)/1.334,V3):: NEXT J
:: RETURN
1300 X=X+1+(X=4)*4 :: ON X G
OSUB 1310,1320,1330,1340 ::
RETURN
1310 FOR J=1 TO T*D :: CALL
SOUND(-999,N(A),V1,N(B)/1.49
7,V2,N(B)/1.679,V3):: NEXT J
:: RETURN
1320 FOR J=1 TO T*D :: CALL
SOUND(-999,N(A),V1,N(B)/1.49
7,V2,N(B)/1.334,V3):: NEXT J
:: RETURN
1330 FOR J=1 TO T*D :: CALL
SOUND(-999,N(A),V1,N(B)/1.49
7,V2,N(B)/2,V3):: NEXT J ::
RETURN
1340 FOR J=1 TO T*D :: CALL
SOUND(-999,N(A),V1,N(B)/1.67
9,V2,N(B)/1.334,V3):: NEXT J
:: RETURN
    
```

That routine will play straight 3-part harmony, but I like this one better, although it does not work well with some pieces.

```

110 D=30 :: S=1 :: V1=1 :: V
2=5 :: V3=7
1000 FOR J=1 TO T :: X=X+1+(
X=4)*4 :: ON X GOSUB 1010,10
20,1030,1040 :: GOSUB 2000 :
: NEXT J :: RETURN
1010 CALL SOUND(-999,N(A),V1
,N(A)*1.01,V1,N(B),V3):: RET
URN
1020 CALL SOUND(-999,N(A),V1
,N(A)*1.01,V1,N(B)/1.585,V3)
:: RETURN
1030 CALL SOUND(-999,N(A),V1
,N(A)*1.01,V1,N(B)/1.334,V3)
:: RETURN
1040 CALL SOUND(-999,N(A),V1
,N(A)*1.01,V1,N(B)/2,V3):: R
ETURN
1100 FOR J=1 TO T :: X=X+1+(
X=4)*4 :: ON X GOSUB 1110,11
20,1130,1140 :: GOSUB 2000 :
: NEXT J :: RETURN
1110 CALL SOUND(-999,N(A),V1
,N(A)*1.01,V1,N(B),V3):: RET
URN
1120 CALL SOUND(-999,N(A),V1
,N(A)*1.01,V1,N(B)/1.585,V3)
    
```

```

:: RETURN
1130 CALL SOUND(-999,N(A),V1
,N(A)*1.01,V1,N(B)/1.334,V3)
:: RETURN
1140 CALL SOUND(-999,N(A),V1
,N(A)*1.01,V1,N(B)/1.497,V3)
:: RETURN
1200 FOR J=1 TO T :: X=X+1+(
X=4)*4 :: ON X GOSUB 1110,11
20,1130,1140 :: GOSUB 2000 :
: NEXT J :: RETURN
1210 CALL SOUND(-999,N(A),V1
,N(A)*1.01,V1,N(B),V3):: RET
URN
1220 CALL SOUND(-999,N(A),V1
,N(A)*1.01,V1,N(B)/1.679,V3)
:: RETURN
1230 CALL SOUND(-999,N(A),V1
,N(A)*1.01,V1,N(B)/1.334,V3)
:: RETURN
1240 CALL SOUND(-999,N(A),V1
,N(A)*1.01,V1,N(B)/2,V) :: R
ETURN
1300 FOR J=1 TO T :: X=X+1+(
X=4)*4 :: ON X GOSUB 1110,11
20,1130,1140 :: GOSUB 2000 :
: NEXT J :: RETURN
1310 CALL SOUND(-999,N(A),V1
,N(A)*1.01,V1,N(B),V3):: RET
URN
1320 CALL SOUND(-999,N(A),V1
,N(A)*1.01,V1,N(B)/1.679,V3)
:: RETURN
1330 CALL SOUND(-999,N(A),V1
,N(A)*1.01,V1,N(B)/1.334,V3)
:: RETURN
1340 CALL SOUND(-999,N(A),V1
,N(A)*1.01,V1,N(B)/1.497,V3)
:: RETURN
2000 FOR Y=1 TO D :: NEXT Y
:: RETURN
    
```

Both of those routines cycle through four inversions of the chord, to avoid a monotonous drone.

There are many ways to vary those routines. Just for instance, right after each N(B) put \*2 to raise the harmony above the melody. Also try \*4. Or alternate \*2 and \*4. Experiment! Have fun!

# USING THE PACS BBS

by Whit Bauer

Philadelphia Area TI User Group

The Philadelphia Area Computer Society (PACS) has supported a Bulletin Board System (BBS) for many years. It is a large, fast system with a very large hard drive capacity, and is available 24 hours a day, 7 days a week. It supports as many as six online users simultaneously, using six separate phone lines and, of course, six modems. The modems can support data transfer speeds from 300 bits per second (bps, also called "baud") to 9600 bps, though not every modem can support every speed. (More on this in a moment.) The BBS seems to have something for everyone, with some 40 conferences (topics of interest) covering everything from Artificial Intelligence to Word Processing. And, yes, it includes a conference which embraces the TI and the Geneve!

## How to access the PACS BBS.

215-842-9600, 9601, 9602, 9603, 9604, 9605

Even though there are six telephone lines to use, in general you need only remember 215-842-9600. Better yet, load it into the AutoDialer of your Communications program, such as Telco or MassTransfer, etc. There are two exceptions:

1. If you have a 300 bps modem, use 842-9601.
2. If you have a 9600 bps modem, use 842-9603.

Here's why. There are currently four popular modem speeds available: 300, 1200, 2400, and 9600 bps. Some 9600 bps modems have compression and other features which increase their effective speed to 19,200 bps. (Do not become confused by the coincidence that the phone number of the BBS, 842-9600, is similiar to a 9600 bps transmission rate.) At this writing in the summer of '92 (no, not a movie title), few people who own a modem rated at 300 bps want to access a bulletin board frequently. It's just too slow for the average user. Similarly, few people own 9600 bps modems, as they are fairly expensive. Only those who use their modems frequently and extensively on long-distance phone lines or other per-minute services, or who use them for business, can justify the expense. So, the average user has a 1200 or 2400 baud modem. For these reasons, here's the way the PACS board is set up: First, in the language of the telephone company, "trunk hunting" is installed on the first four phone lines; that is, when you use 842-9600, that line connects if it is not busy with another call. If it is busy, your call "hunts" for the next phone line, 9601, without your even knowing it. If 9601 is also busy, your call is again "bumped" to 9602, again without your knowledge. And so on, through 9604. The last line, 9605, is equipped with a 9600 bps modem capable of handling special PEP tones necessary for a particular use. The modems on all the lines are compatable with 300, 1200, and 2400 bps, except the one on the first line, 842-9600, which cannot support 300 bps. The modems on 842-9603 and higher also support 9600 bps. Because of this "hunting" feature, if your modem is capable of 1200 or 2400 bps, you should use 842-9600 to access the modems on any of the first 5 lines, all of which handle 1200 and 2400 bps. If you own a 9600 bps modem, you should use 842-9603, since the modems on the 9603 and 9604 lines can go that high. If you can achieve only 300 bps, then use 842-9601 to access any except the first modem.

### How To Login to the PACS BBS.

The BBS is physically located at LaSalle University in Philadelphia, so message-unit or long-distance telephone line charges may apply, depending on where you are calling from. Set your communications program to transmit at 8N1 and full duplex. Then use the ATDT commands to your modem to dial 215-842-9600 or the proper number as discussed above. Your modem will report "CONNECT" or whatever your modem does. Next, the BBS will request you to login, or you may get nothing, or even garbage on the screen. DO NOT DISPAIR if you get garbage. The first remedy is to hit ENTER (or RETURN) once. If you still get garbage, you simply have to send a BREAK signal to the BBS's modem to properly connect with your modem's speed. This is called "handshaking" by the two modems. You send the BREAK signal from your TI using MassTransfer by holding down the CTRL key and simultaneously depressing the "slash" key, the one to the right of the "P" key. (From a Geneve using Telco, use a CTRL 5. I don't know whether the difference is caused by the TI/Geneve difference or by the MassTransfer/Telco difference. MassTransfer I,m told uses FCTN 4. In short, different configurations use different key combinations to send a BREAK -- you may have to experiment! (Oh, no! Experiment!)) You may have to send the BREAK signal several times -- just wait a couple of seconds after each BREAK to see if the request to login is received on your screen. (You may not always have this "break" problem. 2400 and 9600 bps users probably never will. 300 and 1200 bps users may or may not, depending upon which modem you get.) When the request to login is received, unlock the alpha-lock key and type in lower case the letters, "bbs" (without the quotes), then press Enter. You will receive a WELCOME message, then be invited to complete your login as a NEWUSER, or, if you have logged in before, you should log in with your last name, then your password. Remember to always use lower-case in all your replies. You will now be invited to JOIN a conference. For a list of all conferences available, respond "help" (no quotes). For the TI conference, respond "ti". Remember, lower-case, without quotes. If you are new to the board and unfamiliar with its operation, I believe you should type "menu" and select the executive menu. I believe it will help you feel your way around the BBS.

### What is Caucus? What is Magpie?

At the present time, the PACS BBS is simultaneously running two separate bulletin board systems on one computer. Caucus has been available for many years, while Magpie is a recent addition. Magpie is being tested and reviewed, with the possibility of replacing Caucus in the future. For now, at least, either system is available to you at login time. When the login request is received on your screen, respond with "bbs" for Caucus or "magpie" for that system. This discussion reflects Caucus operation.

### How to Use Caucus on the PACS BBS.

As I said early in this article, there are about 40 conferences (topics of interest) on Caucus. Here is the selection list. Surely, one of these would be of interest to you:

ai	amiga	apple	atari
atari_st	bbs	bible	caucus
classifieds	coco	commodore	comped
consult	cpm	databus	dhase
dbms	for_sale	general	graphics
ibm	jobs	mac	midi
m sword	naic	naic2	pacboard
pacsgeneral	pascal	prodigy	rbase
robot	tandy	tandyfiles	telecom
ti	unix	weather	wp

Each conference consists of "Items", or specific subject within that conference. Each item has "responses" as various people add their thoughts or comments to the item. The items can be listed, and when you see an item that seems to be of interest, you can show the item with all the responses. They can be lengthy, so be aware that you can pause to view your screen with CTRL S, then resume scrolling with another CTRL S. To cancel the rest of the responses, CTRL C will abort. If you are on a pay-per-minute phone line, just capture the info on a disk file, then view or print it offline. If the BBS pauses every so often to ask whether you want to continue, at any prompt you can "set screensize 0" to eliminate the pauses. While this is far from a complete text on the use of Caucus, it should be enough to get you interested in BBSing!

#### How to Access the Files Section.

It's simple, but it's totally undocumented in the menus: At any prompt, type "files" (yes, lower case, without the quotes) and Enter. To move around in the files section, type SH to show a list of groups of files. The BBS responds with a list of file groups. Now, respond with the name of the group of files you wish to see the detailed list of files. For example, to look at the list of files for the Geneve, respond "9640" and the list of files for the Geneve will be shown. To look elsewhere, type "sh" again, then select "utilities" to see that list of files. To return to the main menu, enter "q" (for quit).

#### Other Bulletin Boards.

There are many Bulletin Boards around the country, some of which are partially or entirely dedicated to the TI/Geneve. In the Philadelphia area, the TI-D-BITS board has been available in the recent past, and has been very good with messages, especially. It has not been available for some months, and I have no idea whether its return is imminent or even likely. Phone is 672-4051. Another board, which even runs on a TI, is the Bullwinkle Board. It, too, is an interesting board. Phone is 729-0401.

Every BBS is different in the way it operates, even when running the same BBS software, because every SYSOP has his/her style of implementation. So, expect a learning curve on every board you try. They all have something that you'll like or learn. But do start with the local boards where the phone line charges are low or free, depending on your location and phone services used. Remember weekend and late night rates are lowest. When you have a few hours under your belt, then spend a few dollars on some out of town BBSing. Best of luck!

+++

by Jim Peterson

The hard part of learning to program is not in learning what the various commands do - it is learning how to put them together to do what you want them to do! Key in this little routine, run it to see what it does, then read the explanation of how it does it.

```

100 DISPLAY AT(12,1):"Input
filename?":"DSK" :: ACCEPT A
T(13,4):IF$
110 DISPLAY AT(15,1):"Output
filename?":"DSK" :: ACCEPT
AT(16,4):OF$
120 DISPLAY AT(18,1):"Put bl
ank lines between paragr
aphs? Y/N" :: ACCEPT AT(19,1
7)SIZE(1)VALIDATE("YN"):Q$
130 OPEN #1:"DSK"&IF$.INPUT
:: OPEN #2:"DSK"&OF$.OUTPUT
:: C$=CHR$(13)
140 IF EOF(1)THEN 170 :: LIN
PUT #1:M$ :: IF Q$="Y" THEN
160
150 IF M$="" THEN PRINT #2:C
$:M$:: GOTO 140 ELSE IF ASC
(M$)<33 THEN PRINT #2:C$:M$;
:: GOTO 140 ELSE PRINT #2:""
:M$:: GOTO 140
160 IF M$="" OR M$=" " THEN
PRINT #2:C$ :: GOTO 140 ELSE
IF ASC(M$)<33 THEN PRINT #2
:C$:C$:M$:: GOTO 140 ELSE P
RINT #2:"":M$:: GOTO 140
170 PRINT #2:C$ :: CLOSE #1
:: CLOSE #2
    
```

This program will add carriage returns to a file, such as those which are nowadays being ported over from IBM. However, the file must have indented header lines and indented paragraphs, to give a clue as to where the CRs should be. You are also given the option of putting blank lines between paragraphs.

The first two lines get the name of the file to be worked on, and the filename to be used for the revised file. Note that the ACCEPT AT cursor is placed right after DSK, to make it plain that the input should be a drive number, period and filename.

In line 130 the files are opened and ASCII 13, the carriage return, is defined as C\$ so it can be more conveniently referred to hereafter.

In line 140, the EOF end-of-file check is placed before the input, because execution keeps returning here until all the file is read.

Records are read in by LINPUT rather than INPUT because if the record contains a string INPUT will stop reading it at that point. If the option to put blank lines between paragraphs was selected, execution jumps to line 160. All the work is done in 150 or 160.

A blank line may be a null string, containing nothing at all, or it may contain a single ASCII 32, the space character, depending on how it was

created. In line 150, if the input record is a null string, a CR is printed to place a CR at the end of the previous record, which has always been left open. The colon starts a new record and the null string is printed to it, followed by the semi-colon to hold the record open. If the first character of the record is less than 33 (i.e., the space character 32), it is either a blank line or the first line of an indented paragraph, and the same action is taken. It is a peculiarity of XBasic that this cannot be written as IF M\$="" OR ASC(M\$)<33 - in spite of the OR, the program will attempt to find the ASCII of a null string and will crash.

If the line is not a null string and does not begin with a blank, it is the second or subsequent line of a paragraph. A null string is printed to close the previous open record, then the record is printed and held open in case it turns out to be the last line of a paragraph and needs to have a CR added next time.

Line 160 is similar. If the record is a null string or a single blank, a CR is printed to close the previous record. If the first character is a blank, the CR is followed by another CR, to place a blank line between paragraphs.

In all cases, execution goes back to line 140 for another input but first checks to see if the end of the file has been reached. In that case it jumps to 170 where a CR is printed to close the final pending record before the files are closed.



**TI-BASE Tutorial 4 Page 2**  
**NorthCoast 99'ers (C) Martin A. Sooley**

Here's one quick reminder. "I keep several copies of my databases on different disks. If one of these new CFs I'm testing wipes out the database, I want to have a backup." "So!" with all this in mind, I decided to CREATE a new database, leave it empty and transfer what I wanted from the old DB (T NAMES), into the new one. I typed CREATE NEWNAMES. When TIB gave me the structure entry screen, I typed in everything you see below.

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

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

[ NEWNAMES STRUCTURE ]

Compare NEWNAMES structure to T NAMES structure. You will see that there are now 12 fields instead of 11. Notice that the first field is now NM, a N umeric field, the size is 4 and 0 decimal places. The rest of the fields match T NAMES except ID which I have changed to a C)haracter field with a length of 7. When I entered all the information above and I was on the last 7 in ID, I pressed FCTN 8 to save and end the creation screen. When TIB asked if I wanted to enter data now, I answered N)o. At that point I had created NEWNAMES and it was completely empty. NEWNAMES is the DB I really need, so all I have to do is move all the data from T NAMES over to it and I'll be happy.

I whipped up MOVED which I later turned into MOVED1 and MOVED2 (FEL). I still prefer FunnelWeb to produce my CFs, but the TIB way is to type MODIFY COMMAND MOVED1 (E). TIB will then start the procedure of producing the CF named MOVED1 and place you in the EDITOR screen. Type in all the lines you see down to and including the first RETURN. At that point press FCTN 8 to SAVE/END and TIB will finish making MOVED1 for you. You may have to press FCTN 9 to break out of the editor at this point. When you get back to the DP start over, and with MODIFY COMMAND MOVED2 (E), type the second CF as you see it. I'll start with MOVED1 and go through it. Remember that # in the first column means comment line and TIB will not execute that line when it reads the #. So you are probably looking at # SET TALK OFF and scratching your head. Normally I set talk off to keep the screen clear. In this case it made me nervous because I now that TIB was doing some real thrashing on two databases and it concerned me. Seeing all the lines go by on the screen doesn't give you much sore control, but at least you don't feel

so left out, and you can see certain problems if you watch which lines execute. You can remove the # and turn off the screen junk any time you wish. SELECT 2 is where the good stuff starts. I'm going through this again. I hope I don't bore the people who already understand the aspect of SELECT. TIB has 5 areas. A different DB can be opened in each area. These DBs will remain open and you can work on all of them, but not all at once. Lets say that each area is a cardboard box. You have 5 boxes. You must do any major work on only one box at a time, however, you can do minor work on the other 4 by reaching over into those boxes and picking items out. If you use the command SELECT, you can change the box that you wish to do major work on. If you are unsure of yourself, you should not only SELECT the important work area, you should also tell TIB exactly where things can be found by using the (x.) directive, where x is one of the area numbers (1-5) (FEL). Therefore, SELECT 2 means take box number 2 in your hands. USE NEWNAMES means place all the stuff called NEWNAMES into whatever box you are holding. In this case it is box 2. TOP means make sure that when we look into this box later the first thing we see is the first record in NEWNAMES. With NEWNAMES this is not important because it is empty, but with T NAMES it could be important. NOTE: TIB does not associate the name NEWNAMES with box 2. You must remember what you have placed in which box. SELECT 1 means put box 2 down and pick up box number 1. USE T NAMES and TOP is the same as above but using T NAMES this time. LOCAL NUMT 4 0, is the variable I will use to feed numbers into the field named NM. REPLACE NUMT WITH 1, puts the number 1 into NUMT. I previously said I would like to start numbering with 1. The next line is DB DSK2.MOVED2. This line runs the CF named MOVED2 which is similar to a sub-program and is located on drive 2. I mention drive 2 because you could change the 2 to any drive you wish. If you have this CF on drive 3, make it DSK3. etc. When this line is executed TIB runs MOVED2 and begins to do all the real work. When TIB hits the statement WHILE .NOT. (EOF) we are working with box number 1. That is because it was selected last and therefore is still the current selection. It is also the box where we are keeping T NAMES. If we selected 2, where NEWNAMES is, the file is empty, so we would get an EOF signal and the WHILE would not execute. In our case it does execute, so we go through all the statements between WHILE and ENDDO. I immediately SELECT 2. This is necessary because I want to APPEND BLANK. To TIB this means, append one complete record (which in this case is all 12 fields) onto the end of the DB which is in box 2 (which in this case is NEWNAMES). Since there is now some real space in NEWNAMES that we can fill with data, we will do so. REPLACE 2.NM WITH NUMT, takes the 1 which we previously placed in the variable NUMT and copies it into the field named NM in box 2. The 2. is to tell TIB box 2. It is not really necessary because we are in box 2 from the previous SELECTION. However, it helps me understand what is going on when I read over old CFs a few weeks after I'm done with them and I can't remember what they were for in the first place. REPLACE 2.LN WITH 1.LN tells TIB to copy the last name from box 1 into the last name field in box 2. TIB works only with the box numbers, but for our understanding, we are saying take the last name we have in T NAMES and copy it into the blank last name field we have created in the new DB NEWNAMES. Notice I use the term COPY and not MOVE.

Continued Next MONTH

# GENEVE 9640



RS232 card

MYARC HFDC CARD

MYARC FLOPPY CARD

2 - 30 MEG HARD DRIVES

3 - 1/2 Ht. 720K Floppies

1 - 1/2 Ht. 1.44 Meg Floppy

RGB COLOR MONITOR

Hundreds of disks of programs and  
graphics

ALL DOCUMENTATION

MANY OTHER EXTRAS

CALL (215)368-0446

\$450.00 OBO for all