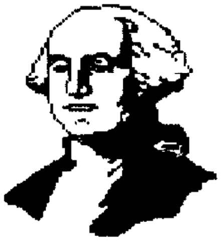
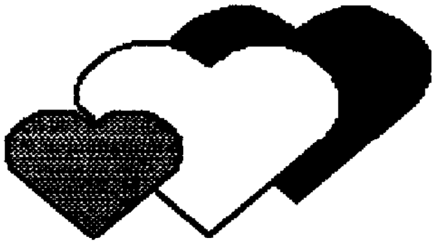
TI-D-BITS

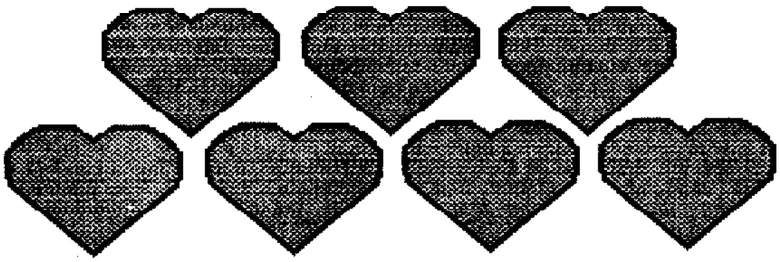
PHILADELPHIA AREA USERS GROUP NEWSLETTER
COVERING THE TI99/4A
AND MYARC 9640 COMPUTERS

FEBRUARY 1992

Volume 12 Number 2







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

VICE PRESIDENT
Allen Silversteen. 215-885-7910

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

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

Committees consists of:

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

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

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

EDUCATION
Barry Traver
Allan Silversteen

EQUIPMENT
Allen Silversteen

PROGRAM
Barry Traver

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

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

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

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

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

LANGUAGE DRILLS

by Don Shorock
Reviewed by Jim Peterson

Don Shorock has written a shell program to give multiple-choice quizzes in word recognition, using disk files of words, and another program to be used to create these word files. By using the latter program, and by redefining letters peculiar to a specific language, he has created fairware disks of language drills in many languages. Each disk contains one or more word files, and he asks that users key in their own additional files for study, and share them with him.

My TI-PD catalog offers these disks for Modern Greek, Danish, Vietnamese, Czech (two disks), German, French, Polish, Hungarian, Russian, Japanese (three disks), Spanish, Finnish, Swedish, Nor'vegian, Latin, and Hebrew. Don also offers them in Arabic, Bulgarian, Croatian, Dutch, Esperanto, Gaelic, Icelandic, Indonesian, Italian, Rumanian, Serbian, Slovakian, Swahili, Turkish, Ukrainian and Yiddish, and is willing to consider requests for other languages, although he considers some Asiatic alphabets to be beyond the capabilities of the TI computer.

Most of these languages required only the modification of a few characters - the German umlaut, Spanish tilde, Scandinavian slashed O, etc. The Russian, Greek and Hebrew disks required a complete new alphabet. Vietnamese was a real challenge because their alphabet uses Roman letters with accent marks both above and below, which will not fit within the 8x8 dot matrix of a standard character definition.

Japanese, however, was the most difficult of all. In ancient times, the Chinese developed a most impractical system of using a picture to represent each word in their language, and these pictures evolved into a stylized form which do not even give a clue as to their meaning. One must therefore memorize several thousand of these characters in order to read fluently.

Since China was the Asiatic center of learning, the Chinese ideographs were also adopted in Annam (now Vietnam), Korea and Japan. The Vietnamese and Koreans have had the good sense to abandon them in favor of their own alphabet systems, and even the Chinese Communist regime streamlined them for easier writing and learning.

The Japanese, however, cling to the traditional form, although they are even more impractical for writing Japanese. Chinese verbs are not conjugated, therefore the same pictograph can represent all tenses. Japanese verbs, however, can have many different endings - and so can the adjectives and adverbs. The Japanese therefore had to develop a phonetic alphabet in order to tack these endings onto their words - and for some reason, they invented not one phonetic alphabet but two! Japanese newspapers are now written, usually in vertical columns, in Chinese kanji characters combined with the hiragana alphabet for the conjugations, with the katakana alphabet used for foreign words which have been adopted into Japanese, and often also with Roman letters used for foreign names and foreign words. The Japanese schoolchildren must therefore memorize thousands of kanji characters plus three alphabets in order to read their schoolbooks - and they are still the best-educated people in the world!

Don has written three Japanese drill disks. The one uses the ordinary Western alphabet, with the long vowels overlined. This is the Romaji or Romanji system which is used in phrase books for tourists, etc., and is adequate for learning the spoken language. Since the Japanese have already purchased Hawaii and are negotiating to buy Ohio, it might pay anyone to try out this disk - just in case!

The second disk provides drills in recognizing the characters of both the hiragana and katakana phonetic alphabets, and a considerable number of the less complex kanji ideographs.

The third disk provides multiple choice quizzes in the meaning of words written in either the hiragana or katakana alphabet. This disk also provides the option to run a program in console Basic, using the TE II module and Speech Synthesizer, and take a spoken rather than printed quiz! The speech of the Speech Synthesizer is never very clear, of course, and it was programmed for English rather than Japanese sounds, but the result is not all that bad. In fact, since Japanese is spelled phonetically with far fewer exceptions than English, most of the words are pronounced correctly although not too intelligibly.

****

YOU DON'T HAVE TO HAVE IT ALL!

by Jim Peterson

Do the conversations at your user group meeting sound like a coffee break in Silicon Valley? Are you confused by talk of GROMs and GRAMs, puzzled by references to HFDCs, intimidated by discussions of megabytes and frightened by talk of burning EPROMs? Well, join the crowd, buddy - so am I!

There are basically three types of people interested in computers. First, there are those who use a computer to run programs, to accomplish something useful or just to have fun. I believe that those people are still in the great majority, although we don't hear much from them.

Then, there are those who get their kicks out of writing programs, of creating software for others to use. There aren't too many of those left in the TI world.

And finally, there are those who like to tinker with the computer, soup it up, plug in doohinkies and thingama- jigs, and talk in that strange language I mentioned above. I don't know how many of those folks there are, but they are certainly the most knowledgeable, active, and interested, and they tend to dominate the conversations and the printed material in the TI world nowadays.

I presume that those fellows also do actually run programs on their souped up systems. And, some of them must be skilled programmers, because many of their hybrid hardware creations would be useless without specialized software.

I'm very glad that those people are around. Once in a while they invent something that I actually find useful, and they are a lifesaver when my equip- ment breaks down.

But, don't be intimidated by all that high-tech talk, and don't think that the computer world is passing you by. There are so many things to do with a computer that no one could possibly find time to do them all. Do your own thing and don't worry about the rest.

I have operated a TI software company for seven years, and I also spend a lot of time writing programs, using the computer as a word processor, etc. I probably spend more time on my TI than 90% of the users. So, what does my equipment consist of?

I have a console with the Extended Basic module plugged in, attached to a P-box which contains a TI disk controller, two double sided drives, the 32k card, RS232 card, and a Horizon Ramdisk. Also plugged into the RS232 card is an old Gemini 10X printer and an Avatex 1200 baud modem.

I also have a Speech Synthesizer, a pair of TI joysticks, a TEII module and an Editor Assembler module, all of which I plug in occasionally when I need them; also, a cassette recorder and cable which hasn't been used in a long time.

I use Triton's Super Extended Basic module because it has some editing features which are useful when programming. It also has some limited plotting capability which I have never used - and have never heard of anyone who has. If you don't program, it would hardly pay to switch from the old TI Extended Basic. I also have the Mecha- tronics module but never got around to trying it.

I had a Gram Kracker but soon sold it and bought a Ramdisk instead. The Gram Kracker has fantastic capabilities if you have the skill and knowledge to take advantage of them.

but most users don't seem to have done much beyond personalizing the title screen.

I had a widget, and I guess it is still collecting dust around here some place. It was a nuisance, and since I use XBasic 99% of the time I didn't need it. There are now widgets or "module expanders" that allow you to access more than one module from within a program. That is, if you have the skill to write such a program. I don't know that anyone has released such programs to the public domain, and I can't think of any practical use except to access TEII speech from XBasic - but you can do that with the Text-To-Speech disk.

The ram disk is the one tool that I would not be without. In order to assemble my TI-PD catalog, I screened over 4000 programs, debugged and modified, merged in help files, conversions to XBasic and loaders, and assembled over 400 disks of programs. It took me hundreds of hours of work - without a ram disk it would have taken thousands of hours and I would not even have attempted it.

The ram disk enables me to switch from one program to another almost instantly, and with John Johnson's Boot program I can just as quickly catalog a disk or view a file. Mine has 256k of memory. I could get one with much more memory but I see no reason to do so; I have every program on it that I am apt to use even once a month, and it is only half full. That leaves plenty of room for temporary storage and downloading.

However, if you only use your computer to play games, do a little word processing and a bit of record keeping, a ram disk would be an expensive convenience rather than a necessity.

Since my ram disk is only half full, I would consider a hard drive to be about as useful as the mammalian appendages on a swine of the masculine persuasion. If I was running a BBS, sure - or if I was doing a lot of work with those memory-gobbling graphics and needed everything quickly accessible.

My old Gemini printer has been a faithful workhorse, although the hood over one sprocket wheel has lost its spring and is being held down by a loop of elastic cord. I will have to give it up soon, because the Gemini printer codes are becoming obsolete and I need to be able to write and test Epson codes. But, I hate to give up these 79-cent typewriter ribbons and start getting ripped off on $2.50 cartridges!

As for a color ribbon, the temperature will have to go way down, down under, before I pay for one of those.

Once in a while, when someone sends me a double-density diskfull of stuff, I wish I had a CorComp disk controller. Otherwise, with diskettes selling for a quarter or less, it wouldn't pay to change.

If I ever get around to subscribing to GENie or Delphi, it will pay me to get a 2400 baud modem.

I can't think of anything else I need, and I don't want what I don't need. If I really wanted to play joystick games, I would certainly get something better than the TI joystick. And if that MIDI interface cable becomes a reality, I wil be sorely tempted.

I can't see any advantage in putting the 32k under the hood, or anyplace other than where it is now. If I used speech a great deal, it would be nice to get rid of the synthesizer - but I know only one user who uses speech that much. I don't need a clock built in because I have a watch on my wrist. If I really did a lot of serious writing, an 80-column card would be wonderful. But then I would have to buy a monitor capable of displaying 80 columns. I certainly don't want to give up color, and high-resolution color monitors cost more. I would still want to use my old monitor for programming, because I like to write programs for folks who have basic equipment. I don't have room on my computer desk for two monitors, so I think I'll pass.

I'm a three-finger typist, so a RAVE keyboard wouldn't speed up my typing very much. If I really wanted an IBM keyboard and 80-column capability, I would throw in a few bucks more and get a Geneve.

So, what about the Geneve? If I had an irresistible urge to run the few great programs that have been written for it, or if I wanted to explore its great programming capabilities, I would get one. But, I like to write programs for other people to use. When so few are interested in programs that I write for a computer that sold in the millions, why would I write programs for a computer purchased by a couple of thousand people?

I am sure that many folks will disagree with what I have written. That's why I wrote it. I hope they will disagree so strongly that they will immediately boot up Funlweb and compose a blistering reply. But don't send it to me - send it to your newsletter editor. The newsletters are badly in need of more articles by more writers!

****

TERMINOLOGY AT THE HEARTBEAT

BY Chip Chapin
Taken From Boston TI User Group

During the last month or so, I have been spending my time in the nation's capital (the nation's kinder and gentler' Heartbeat) and, while some things are exactly as you would expect, some things are just that little bit different. when I ran across some computer terminology definitions in a U.S. Government publication, I just knew that they were going to clear up any old confusion that might have crep into our non-offical lingo over the years. I was just a teensy bit worried that the definitions might not relate to the TI, it being an orphan an' all, but shucks, it all fit in just as fine as frog hair. Them Gub'nent folks has really got it together. Why, I'll even be able to join in when my friends with IBM's talk about using those AUTOEXEC.BATs. Anyhow, here's some of the latest definitions making the rounds in the DC area.

ADVANCED USER: A person who has managed to remove a computer from its packing materials.

POWER USER: A Person who has mastered the brightness and contrast controls on any computetr's monitor.

AMERICAN-MADE: Assembled in America from parts made abroad.

ALPHA TEST VERSION: Too buggy to be released to the paying public.

BETA TEST VERSION: Still too buggy to be released.

RELEASE VERSION: An alternate pronounciation of "BETA TEST VERSION".

SALES MANAGER: Last weeks's new sales associate.

CONSULTANT: A former sales associate who has mastered at least one-tenth of the dBase III manual.

SYSTEMS INTEGRATOR: A former CONSULTANT who understands the term AUTOEXEC.BAT.

AUTOEXEC.BAT: A sturdy aluminum or sooden shaft used to coax AT hard disks into performing.

That's all folks. This came from MICROCOMPUTING, government computer news, who credited the WIC Connection, the newsletter of the EPA's Washington Information Center. The WIC obtained it from an EPA employee who picked up a copy in New Hampshire. And guess what - the author or authors are unknown.

****

TI-BASE - From INSCEBOT TUTORIAL 2 By Martin Smoley NorthCoast 99'ers - Sept. 1, 1988

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.

First some of Marty's shorthand from last month. The letters TIB will refer to TI-Base. MT: will signify the beginning of some text which should be considered Marty's Theory. Marty's Theory should not be taken as fact, but as my interpretation of an item. FYI: designates text that is For Your Information. FE will stand for For Example. DP will stand for Dot Prompt. <E> means press ENTER. (FEL) means Further Explanation Later, and last for now is ">", the greater than sign. I will use ">" when program segments are displayed at the left of every line. The position immediately to the right of the ">" will be column one. Take the example >12345. You should think of the number 1 as column one. The > does not exist. It is for reference only, the same as when you type in an XBasic program, at the head of each line you see > but it is not part of the program. ALSO! In Tutorial 2, I have listed some Command Files with line numbers instead of ">" in the left most column. This is to allow for explanation of specific lines only. Line numbers are not used in Command Files, but from now on you will have to use FunnelWeb or the E/A Editor to create the Command Files, and this will be easier on me. Since we're on the subject I might as well fill you in. The editor which comes with TIB is not bad. By editor I am referring to the part of TIB you use to write and save Command Files. However, in TIB version 1.02 when you enter about 33 lines you run out of memory space. If you want or need to use the TIB editor you could produce a bunch of Command Files that run each other and get the job done quite well. I prefer to have the luxury of writing larger files if needed. I also prefer the use of embedded control codes as printer commands, which at this piont are not available in the TIB Editor. There are two more reasons to contemplate an outside editor. The first is that the Command Processor that runs your Command Files truncates or chops off all lines at 40 characters. This means you can set tabs at 40 columns and after typing commands on the right half of the page you can tab over past 40 and type in comments. TIB will never see the comment so they won't interfer with the program logic or slow the speed down. Last, I print out lots of hard copies to check my work. It's hard to print files created by the TIB Editor in Int/Fix 40 Format.

Now I'd like you to make a correction in the dataoase we created for Tutorial 1. The problem is in the "XP" field of the database, "TNAMES". As it was displayed in SCREEN FOUR the XP dates were "Month-Year", (02-88), etc. This configuration does not sort to a desirable conclusion in a character field (FEL). In order to get what we want out of a "SORT ON XP" command we need the year first and the month second, ie. "Year-Month", or (88-02). Since we only have five names in TNAMES you can edit the file and change them. I have placed a printout of TNAMES at the bottom of this page for your convenience. You are, of course, going to have to learn something along the way. Let's say that you are really trying to learn TI-Base and you were working frantically on something when this newsletter arrived. Reading to this point you want to start editing immediately. In order to get going you must CLOSE your present file, point TIB at disk 3 (which is where you have the database named TNAMES), un-SORT the file, and you don't like the present screen colors. If you had the little program that's listed below, you could type DO EDTN and TIB would do the rest. So let's make one. Fire up FunnelWeb and press 1 for EDITOR. When you get into the editor press (CTRL 0) to get out of word wrap mode. You should then see a hollow cursor. At that point you can type in the Command File, EDTN. When this is done save it to disk under the name EDTN/C, and print out a hard copy which you can compare against the listing below. Remember, you don't type in the line numbers, and any line with an asterisk in the first column is a comment line.

```
0001  * Command File to EDIT TNAMES
0002  *       PROGRAM NAME = EDTN
0003  *       SAVED AS   EDTN/C
0004  *
0005  CLOSE ALL
0006  SET DATDISK=DSK3.
0007  USE TNAMES
0008  SORT OFF
0009  TOP
0010  COLOR WHITE DARK-BLUE
0011  EDIT
0012  CLOSE ALL
0013  RETURN
```

Let's attack this little CF (CF = Command File). Lines 1 through 4 can be anything you need to refresh your memory about this program. Line 5 is a good idea for every CF you own. This line has saved me many times. If there aren't any Dbs (Db = Database) open, then 5 will do nothing. Line 6 is not really needed and you can leave it out. I do change drives on occasion with this statement, but you should remember to change it back at the end of the CF with a similar line. The reason its here is to demonstrate that the CLOSE ALL should come at the very beginning of the CF before you do something like line 6 and confuse the system. Line 7 will open TNAMES on drive 3

Continued Next Page

TNAMES

REC	LN	FN	MI	SA	CT	ST	ZP	PH	XP	6P	ID
0000	Vivannovitch	Elexxie	I.	111 E. 98th. St.	Cleveland	OH	91023	541-5415	88-05	NOCO	0712881
0001	Aardvark	Grant	E.	9995 State Rt. 84	Geneva	OH	44014	1-465-9876	88-02	NOCO	0717851
0002	Whitman	Raymond (Slim)	A.	2574 East 254th.	Eastlake	OH	44094	951-2345	88-09	NOCO	0921861
0003	Jones	Quincy	W.	37285 Burgandy Laine	Mentor-on-the-Lake	OH	44060	257-1029	88-08	NOCO	0820871
0004	Smoley	Martin	A.	6149 Bryson Drive	Mentor	OH	44060	257-1661	89-02	NOCO	0713831

TI—BASE Tutorial 2 Page 2

NorthCoast 99'ers (C) Martin A. Seeley

as per line 6 or where ever DATDISK is located if line 6 is omitted. You can also use "USE DSKx.TNAMES" where x is any drive number, including a HORIZON Ramdisk No. 6, "which I use". SORT OFF will un-SORT the file and TOP will point TIB at the first record in the Db, as BOTTOM would point TIB at the last record. Line 10 is all you need to change the foreground and background colors. See page 4-2 of the manual for colors available.

AND NOW THE EDITOR!

Line 11, EDIT, will put you in edit mode using whatever Db file is open. In this case TNAMES. While you are in EDIT you can use arrow keys or enter to move around. You can then type over any item you want to change. At this time it is the XP field. "This is important!". You can also use FCTN 6 to page up, or enter, or FCTN 5 to page down to the next record. This could cause a problem as the changes you have made will not always be saved. If you make any changes you should always use FCTN 8 to register, or save, your changes and move to the next record. If you are on the last record in the file, you should still press FCTN 8. This will not end the editing session and you will remain in the last record. You can then press FCTN 6 to page up, or FCTN 9 to leave the editor. In this case you would be returned to line 12 of our CF and TNAMES would be closed. RETURN will end the program and take you back to the DP. MT: If I am editing a file without a program, I close the file with CLOSE or CLOSE ALL as soon as I have finished. This allows TIB to update all of the records. One last idea on this CF. I either SET TALK ON in the first line of this CF or more often it is already on when I run EDTN. This will allow me to read lines 1 through 7 on the screen while the Db is being un-SORTed. I can then see if this is actually the program I wanted, with the right Db, and that I have changed the location of the DATDISK. The beauty of little CF programs like this is that you can build on them and add things you realize you want as you go along. The CF will not forget any of the details from one day to the next like I do. Also, once you have the first one done you can copy it to a new name, (COPY DSK2.EDTN/C DSK2.NEWED/C). You can then use MODIFY COMMAND NEWED to edit this new CF to handle another Db, or do whatever you wish. It's much easier than typing a completely new CF from scratch.

IMPORTANT TIP!

I have discovered that a CF created with FunnelWeb in DV/80 format can be copied or edited by TIB and the DV/80 format will not be changed. Therefore, if you create the CF below with FunnelWeb and save it to the name BLNK/C on your DATDISK, you can then copy it to a new name, re-edit it, and save it with MODIFY COMMAND, and it will remain a DV/80 file.

```
>SET TALK ON
>*
>*      Command File BLNK
>*
>*        Save as BLNK/C
>*
>*  Use as a seed file for DV/80
>*
>*  Copy to a name of your choice
>*    and type over this stuff.
>*
>RETURN
```

This means you will then be able to print the CF with FunnelWeb for a hard copy. I try to make sure that all m hardcopies have the program name and pertinent comments at th top. Then if I'm writing a new CF I can look over these hardcopies and then merge chunks of previously written material with FunnelWeb LF Merge capabilities.

Let's get started on this months project. We need another Db to try some new routines. Create TNTST2 using the instructions below. Some of this is a repeat so skip over the parts you know and get right to the data entry. If this doesn't look slightly familiar you should refer back to Tutorial number one for more help.

```
>CLOSE ALL <E>
>CLEAR <E>
>CREATE TNTST2 <E>
```

When the CREATE screen comes up enter the following fields, anc when you enter the 0 (zero) in the last column of field 4 press FCYN 8 and wait for TIB to create the file for you.

arrows to move, enter to advance

FIELD	DESCRIPTOR	TYPE	WIDTH	DEC
1	TDATE	D	8	
2	NUM1	N	7	2
3	NUM2	N	7	2
4	ID	N	7	0

After pressing FCTN 8 TIB will ask if you want to enter data now. Answer yes and enter the data supplied below. Take your time, there are a lot of numbers here and you may get confused.

REC	TDATE	NUM1	NUM2	ID
0000	03/16/88	100.11	100.22	071288
0001	02/29/88	200.11	200.22	071383
0002	08/27/88	300.11	300.22	071785
0003	03/03/88	400.11	400.22	082087
0004	12/30/87	500.11	500.22	092186
0005	06/06/88	600.11	600.22	071785
0006	04/22/88	700.11	700.22	092186
0007	01/21/88	800.11	800.22	071383
0008	05/12/88	900.11	900.22	082087
0009	06/17/88	1000.11	1000.22	071383
0010	03/01/88	1100.11	1100.22	092186
0011	08/03/88	1200.11	1200.22	071383

I double spaced the data above to make it as clear as possible. If you make any mistakes, this is a good time to convert EDTN. Type COPY DSK2.EDTN/C DSK2.EDTST2/C (E). After copying it use Modify Command to change lines 1 through 4, and change line 7 to USE TNTST2. Press FCTN 8 to save and you're done.

Continued Next Page

TI—BASE Tutorial 2 Page 3

NorthCoast 99'ers (C) Martin A. Smoley

```
01   * Command File TNTST2
02   *
03     SET TALK OFF
04     SET RECNUM OFF
05     SET HEADING OFF
06     SET LINE=80
07   CLOSE ALL
08     SET DATDISK=DSK2.
09   CLEAR
10   COLOR WHITE DARK-RED
11   WRITE 2,8," TI-Base Demonstration to"
12   WRITE 4,8,"open two Databases at one"
13   WRITE 6,6,"time and find data in File #2"
14   WRITE 8,6,"which is related to an ID No."
15   WRITE 10,7,"in File #1. With some very"
16   WRITE 12,7,"simple math implementation."
17   WRITE 14,9,"************************"
18   WRITE 16,9,"   Running:  TNTST2      "
19   WRITE 18,9,"************************"
20   USE TNAMES
21     LOCAL CDATE D 8
22     WRITE 20,2," Enter the Date  MM/DD/YY"
23     WRITE 21,2," Within Quotes"
24     READ 21,18,CDATE
25     WRITE 22,5,"Current Date: ",CDATE
26    SORT ON ID
27   CLOSE
28   USE TNTST2
29    SORT ON TDATE
30   CLOSE
31   SELECT 2
32   USE TNTST2
33   TOP
34   SELECT 1
35   USE TNAMES
36   TOP
37        LOCAL BLNK C 4
38        REPLACE BLNK WITH "'ьE'ьG"
39        PRINT BLNK
40   LOCAL TESTID N 7 0
41   LOCAL TEMP C 60
42     WHILE .NOT. (EOF)
43      REPLACE TEMP WITH "'ьE'ьG" | TRIM(LN) ;
44              | "  " | TRIM(FN) | " " | MI ;
45              | "  " | ID
46           PRINT   TEMP
47             REPLACE TESTID WITH 1.ID
48          DO DSK2.NUMTST2
49         COLOR WHITE DARK-BLUE
50     WRITE 17,9,"  Running: > TNTST2 <    "
51         SELECT 1
52        MOVE
53     ENDWHILE
54   CLOSE ALL
55     SET RECNUM ON
56     SET HEADING ON
57     SET TALK ON
58   RETURN
```

The CF on this page may look complicated, but its not. We will g through it together, and I will try to explain the important parts I hope you have read Tutorial 1 so I can skim over the routine part and concentrate on the rest. Remember don't enter the line numbers. Lines 1 throug 9 are strictly housekeeping except for CLOSE ALL, and from now on I will consider i housekeeping. CLOSE ALL should be part of every MAIN CF. By MAIN I'm referring to a CF that may run other CFs, but is not itself run by a previous CF. This CF run NUMTST2 as you can see in line 48. You would not want to close all the files i NUMTST2 it would bomb the program. I intend to have my data disk in drive 2, line clears the screen and changing the screen colors has no real value. The WRITE statements from 11 through 19 are to demonstrate user prompts. The lines I hav included are not important, but it will give you some idea of ROW-COLUMN display. Line 20 is the beginning of the real stuff. USE TNAMES opens that Db which it expect to find on drive 2. After line 8, TIB will expect t find all CFs and Dbs on drive 2, and I will not wast space bringing it up again. There is a three lin cluster which is important. The lines are 20, 26, an 27. Their purpose is to open, sort and close a file. This is identical to lines 28, 29, and 30. However lines 21 through 25 are of interest. Line 2 initializes the LOCAL variable named CDATE, which is D (date type) entry with a length of 8 characters. variable is a place to store some type of information In this case it will be the Current DATE (CDATE) which you will type in when asked. Lines 22 and 23 will as you to enter the date and 24 will place the cursor or the screen one space after "Within Quotes", and wai for your input. NOTE: with Version 1.02 all Characters or Dates, which are characters, must be input enclose in quotation marks, "09/01/88". Line 25 will write th message "Current Date:" and display whatever you type in for CDATE. FYI: I have initialize CDATE close to its use for your benefit. I will continue this procedure in this program, bu from then on variables should be initialized at the beginning of a CF. This little chunk (LN 21-25) was stuck in here because TIB likes to have a Db open before you READ to a variable AND! line 25 WRITEs to screen line 22. You will notice that screen line 22 doesn't scroll lik the rest. You can put a message there and it will stay put until a CLEAR or another WRITE 22, removes it. Some of these things will be apparent when you run this program, or DO TNTST2 Line 31 leads us into a very complicated and confusing area. I will try to cover it as thoroughly as possible. I will re-analyze it many times in the future, "It's that important."

Think of a Lazy Suzan, or a rotatable table. This table has 5 areas o it with low partitions between each area. You can take from one to fiv file folders which are filled with sheets of paper and place one in eac of the five areas. You must stand in one spot, but you can SELECT one o the five areas to be positioned directly in front of you. The are SELECTed, (1-5), is the one where you can do the most work, but you can see over the partitions to do limited things with th information in the files which are not directly i front of you. If you can grasp this concept an visualize the 5 different slots, or areas, you'r going to catch on fast. Remembering, or keeping trac of what can be done in non-SELECTed slots is challange. NOW! line 31 and beyond. Line 31 SELECT slot #2, 32 opens the Db named TNTST2 in slot #2 an 33 points TIB at the first record in the file. Line 3 SELECTs slot #1, 35 opens the Db named TNAMES in slot #1 and 36 points TIB at the first record in that file. Remember that both of these Dbs wer previously SORTed to our specifications. Well, we have done it. At this point we have opene two databases at the same time. TNTST2 is open in slot #2 and TNAMES is open in slot #1, an if we don't count all the junk I put in to add flash to the program, we did it with about 1 lines of code. I told you that we'd get through this somehow. If you examine and keep trac of this stuff one peice at a time, you'll get the hang of it sooner than you think.

Continued Next Page

TI—BASE Tutorial 2 Page 4

NorthCoast 99'ers (C) Martin A. Smoley

NOTE: Although there is a CF on this page we will be discussing CF on the previous page until we reach line 48. Don't get confus. We just left line 36. Lines 37, 38 and 39 make up a small group

Their purpose is to initialize the LOCAL BLNK for 4 Characters. Fill it with the control codes that set the printer to Emphasized Doublestrike, and send the codes to the printer. Entering the control codes can only be done with TIW or FunnelWeb at this time. Th will be some useless repetition concerning control codes. I'll explain later. Line 40 and 41 initialize two more variables to be us in the WHILE loop. TESTID is to hold a Number with the length of 7 and 0 decimal places, and TEMP has been discussed previous "Let's get into the meaty part." The way I have set this loop up it will continue to do everything from line 42 through line 53 until reaches the End Of File marker, (EOF), for the Db TNAMES which we have located in slot #1. NOTE: TNAMES is in slot #1, and slot #1 our currently SELECTed slot because the last slot we worked with was slot #1, in line #34. "I'll keep at this concept as we go alon Line 42 has a simple but important job. It immediately checks to see if we have hit the EOF in whatever file is open, in the slot are facing. In this case it is slot 1 and the file is TNAMES. WHILE it does .NOT. encounter the (EOF), in that file it proceeds line 43. If it does encounter the (EOF), it goes directly to the line after the ENDWHILE, which in this case is line 54. Lines 43, and 45 are seen by TIB as one continous line because of the semicolon (;) at the end of lines 42 and 43. So in this line TIB is go to take the 2 control codes directly after WITH and hold them. It will then TRIM the trailing blanks from LN and attach it behind control codes, and then it will stick 2 spaces behind that. It will TRIM the trailing blank spaces from FN, attach it to our grow string, and then throw in another blank space. It will then tack MI on that followed by 2 more spaces, and last but not least ID. did not TRIM MI because MI doesn't have any extra blank spaces. We did not TRIM ID because it is of (N)umeric type (a number) TRIMming is only used on (C)haracter strings. Then TIB takes this whole mess we have put together and sticks it into the variable call TEMP. NOTE: If you look back at line 41 you'll see we made TEMP with 60 spaces. When you fill up a variable with all kinds of jur like we just did, you must make sure the variable is big enough to hold it all. In line 46 we PRINT all the junk we just put into TE I'd like to also mention that the junk we filled TE with was related to TNAMES, (LN,FN,MI,ID). Using th type of data gathering it is up to us to be sure TIB pointed at the right slot and that slot contains the that holds the information we want. In line 47 we p the same ID number from above into the holding area named TESTID. The phrase 1.ID is another way to te TIB that we want the ID number from slot #1. At th time the 1 is FYI only, and doesn't have any real effe on the program except to asure me that I am getting from slot #1. TIBs Author uses this form of da gathering in the CF named PROCESS, manual page 5-5.

"Well here comes line 48." When TIB hits line 48 it leaves the CF named TNTST2 wi everything exactly as it is and executes the CF named NUMTST2 on Disk 2. So now start looking at the lines in NUMTST2. Line 3 CLEARs the stuff left on the scre by that other CF. Lines 4, 5 and 6 put up a new message. TIB WAITs 2 seconc then it changes the COLORs to WHITE on DARK-GREEN. In the same instant it reWRIT a new message to line 17, followed by the TESTID. This is the Identificat Number we brought with us from that other CF. We will use it to find related da in the Db you typed all those numbers into a short time ago. TIB WAITs a coup more seconds just for kicks and we're on our way. Lines 11 through 15 initial all the variables we will need in this program. We can also use variables fr that other CF, but we cannot send these variables (lir 11-15), back there. If we needed to send something back o there, we could put it in one of the variables from that (like TEMP) just before we RETURN and then we could use th information when we RETURNed to that other CF. I did not u the names of the 2 CFs in that explanation because it w even more confusing that way. Line 16 places some bla spaces and the word TOTAL into T for later use. In line we SELECT slot #2, which is where TIB holding the Db TNTST2. TOP in line 18 is only to make me fe secure. We should already be at the TOP of t file. Lines 19 and 20 stick the control code for Italics into BLNK and PRINT it. NOTE TWO THINGS: One, we needed a blank l printed anyway, which this gives us, and Two, that BLNK came over from t other CF. "NOW!", in line 21 we have another WHILE loop. The WHILE loop t runs from line 21 through line 31 has the same definition I gave earlier, we will do different things while we are inside this loop, and it will looking for the (EOF) for TNTST2 in slot #2.

```
01   * Command file NUMTST2
02   *
03   CLEAR
04     WRITE 15,9,"************************"
05     WRITE 17,9,"    Running:  NUMTST2   "
06     WRITE 19,9,"************************"
07   WAIT 2
08   COLOR WHITE DARK-GREEN
09     WRITE 17,9,"Looking For ID No.",TESTID
10   WAIT 2
11     LOCAL TNUM1 N 10 2
12     LOCAL TNUM2 N 10 2
13     LOCAL STNUM1 N 10 2
14     LOCAL STNUM2 N 10 2
15     LOCAL T C 8
16   REPLACE T WITH "    TOTAL"
17   SELECT 2
18   TOP
19    REPLACE BLNK WITH "'ь4 "
20    PRINT BLNK
21   WHILE .NOT. (EOF)
22      IF TESTID = ID
23        PRINT TDATE,BLNK,NUM1,BLNK,NUM2 ;
24              BLNK,ID
25        REPLACE STNUM1 WITH TNUM1 + NUM1
26        REPLACE TNUM1 WITH STNUM1
27        REPLACE STNUM2 WITH TNUM2 + NUM2
28        REPLACE TNUM2 WITH STNUM2
29      ENDIF
30    MOVE
31   ENDWHILE
32    REPLACE TEMP WITH "-------------------";
33    ! "-----------------------"
34    PRINT TEMP
35   PRINT  CDATE,TNUM1,TNUM2,T
36    PRINT BLNK
37     REPLACE BLNK WITH "'ь5 "
38    PRINT BLNK
39   RETURN
```

Continued Next Page.