TI - D - BITS

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

DECEMBER 1992

Volume 12 Number 10

Happy Holidays

The Philadelphia Area TI-99/4A Users' Group meets twice a month. On the second and forth 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.

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 thingamajigs, 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 equipment 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 Mechatronics 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!

****

WHY SHOULD YOU LEARN TO PROGRAM?

by Jim Peterson

Why should you learn to program? To make money? No way! If you could write a program to guarantee world peace, eliminate hunger and cure AIDS, you couldn't make money selling it to the TI world!

Why should you learn to program? To contribute something to the TI world? OK, but don't expect any thanks! Contributing a program to the TI public domain is like dropping a pebble into a bottomless dry well - you will never hear a splash, not even a thud.

Why should you learn to program? Because no one has written the program you need? Well, now you have a good reason! Since there is neither money nor recognition in programming, the programmers tend to write what they feel like writing, not what you want them to write.

Why should you learn to program? For one reason, because I know that you would like to make some changes in the programs that you use frequently. I know that, because the only feedback I ever get is from people who wish that I would change this or that! You really wouldn't have to learn very much to change colors, add or silence a beep or a burp, output to disk instead of printer, etc., etc.

Beyond that, unraveling someone else's code can be tricky and frustrating (and I pity anyone who tries to unravel my code!) Often I find it easier to just rewrite the basic idea in my own way.

If you do modify someone's program, please put a note on the title screen, or at least in a REM, that you did so - and unless you are very sure that you have not introduced a bug, don't distribute your version! Programmers do not like to be blamed for other people's mistakes, and the sales of good programs have been ruined by the bad reputation resulting from pirated, modified and bugged copies.

But, the real reason for learning to program - it's fun, it's challenging, it's creative! There is something very satisfying about getting an idea to make the computer do something it has never done (as far as you know!) and then succeeding in making it do what you want. There is a thrill in pushing the limits of that obsolete tiny TI pea brain just a little bit farther.

There are those who prefer to exercise their creativity with the soldering iron, those who can plug in chips and soup up a Model T computer to run like a Ferrari. I regard them with awe and wonderment, and I'm glad they are around. Without them, I wouldn't have my RamDisk, and my equipment wouldn't get repaired.

Personally, I am the ultimate klutz. If I approach my car with a screwdriver, all four tires go flat. My one feeble attempt to repair my P-Box resulted in failure, expense and embarrassment. But, without having more than a faint idea what goes on beneath that keyboard, I have learned to punch the keys (two right fingers and a left thumb) and create hundreds of programs and routines which have given me a great deal of satisfaction.

It's been fun! You should try it sometime.

****

TIPS FROM THE TIGERCUB

#44

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 deductable 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, KALEIDOSCOPES 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, tips 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.

Thanks to Steve Chapman and Bill Wallbank of Stone & Webster Engineering Corp. TIUG for this one. If V=21 you are in Extended Basic, otherwise you are in Basic. I am not sure it will work with all consoles and modules. -

```
100 RANDOMIZE (0)
110 V=INT(RND*100)
```

How can you input a blank (CHR$ 32) with ACCEPT AT? As far as I know, you can't. With LINPUT, just hit the space bar, and with INPUT, type " ". But with ACCEPT AT the space bar gives a null string and " " gives " " ! However, you can code around it -

```
X$=CHR$(34)&CHR$(32)&CHR$(32
):: ACCEPT AT(1,1):T$ :: IF
T$=X$ THEN T$=CHR$(32)
```

And, to clear up the puzzling behavior of the "quote marks" -

```
100 CALL CHARPAT(34,CH$):: C
ALL CHAR(35,CH$)!written by
Jim Peterson
110 DISPLAY AT(1,7)ERASE ALL
:"THE # PUZZLE":" You can't
enter PRINT # or PRINT ### -
 the computer   demands an
even number of #."
120 DISPLAY AT(5,1):"1 PRINT
 ## !prints a null string (n
othing)":"2 PRINT #*# !print
s *"
130 DISPLAY AT(8,1):"3 PRINT
 #### !prints #":"4 PRINT ##
*## !crashes as  STRING-NUM
BER MISMATCH"
140 DISPLAY AT(11,1):"5 PRIN
T ##**## !crashes as  SYNTAX
 ERROR"
150 DISPLAY AT(13,1):"6 PRIN
T ###### !prints ##":"7 PRIN
T ###*### !prints #*#":"8 PR
INT ###**### !print #**#"
160 DISPLAY AT(16,1):"9 PRIN
T ######## !prints ###":"10
PRINT ####*#### !crashes as
STRING-NUMBER MISMATCH"
170 DISPLAY AT(19,1):"11 PRI
NT ####**#### !crashes as SY
NTAX ERROR":"12 PRINT ######
#### !####"
```

```
180 DISPLAY AT(22,1):"13 PRI
NT #####*##### !##*##":"14 P
RINT #####**#####!##**##"
190 DISPLAY AT(24,1):"TRY IT
! LINE NO.(1-14)?" :: ACCEPT
 AT(24,25)VALIDATE(DIGIT)SIZ
E(2)BEEP:LN :: IF LN<1 OR LN
>14 THEN 190
200 CALL CLEAR :: ON LN GOSU
B 230,240,250,260,280,290,30
0,310,320,330,340,350,360,37
0
210 PRINT :;:;:"Press any ke
y"
220 CALL KEY(0,K,S):: IF S=0
 THEN 220 ELSE 110
230 PRINT "" :: RETURN
240 PRINT "*" :: RETURN
250 PRINT """" :: RETURN
260 PRINT ""*"" !crashes as
STRING-NUMBER MISMATCH - the
 * is misinterpreted as a mu
ltiplier!Same with +,-,/
270 !with anything else, inc
luding numerals, crashes as
SYNTAX ERROR - but inserts a
 space before the character!
280 PRINT ""**"" :: !crashes
290 PRINT """""" :: RETURN
300 PRINT """*""" :: RETURN
310 PRINT """**""" :: RETURN
320 PRINT """""""" :: RETURN
330 PRINT """"*"""" !crash
340 PRINT """"**"""" !crash
350 PRINT """""""""" :: RETU
RN
360 PRINT """""*""""" :: RET
URN
370 PRINT """""**""""" :: RE
TURN
```

The method of closing an "ajar" file, described in Tips #28, doesn't always work, but this one seems to be reliable -

```
100 ON ERROR 500 :: OPEN #1:
"DSK1.TEST" :: INPUT #1:A$ :
: PRINT A$ :: STOP
500 ON ERROR 510 :: CLOSE #1
510 INPUT "CHECK DISK AND DR
IVE, PRESS ANY KEY":DUMMY$ :
: RETURN 100
```

This one is just for the fun of it - it uses the contents of computer memory to create designs -

```
100 DISPLAY AT(3,10)ERASE AL
L:"COLORPEEK": :TAB(7);"by J
im Peterson": : :" Watch the
 computer's memory": :"displ
ayed in color."
110 DISPLAY AT(12,1):"Choose
": :"(1) plain colors": :"(2
) bars & checks": :"(3) patt
erns" :: ACCEPT AT(12,8)VALI
DATE("123")SIZE(1):Q :: CALL
 CLEAR :: IF Q=1 THEN 170
120 DISPLAY AT(12,5):"wait,
please" :: IF Q=3 THEN 140
130 FOR CH=32 TO 143 :: CALL
 CHAR(CH,RPT$("F0",8)):: NEX
T CH :: GOTO 160
140 RANDOMIZE :: FOR CH=32 T
O 88 :: FOR J=1 TO 4 :: X$=S
EG$("0018243C425A667E8199A5B
DC3DBE7FF",INT(16*RND+1)*2-1
,2):: B$=B$&X$ :: C$=X$&C$ :
: NEXT J :: CALL CHAR(CH,B$&
C$)
150 CALL CHAR(CH+55,B$&C$)::
 B$,C$="" :: NEXT CH
160 FOR SET=0 TO 14 :: CALL
COLOR(SET,SET+1,16-SET):: NE
XT SET :: CALL SCREEN(2):: G
OTO 180
170 FOR SET=0 TO 14 :: CALL
COLOR(SET,SET+2,SET+2):: NEX
T SET :: CALL SCREEN(16)
180 FOR J=-1 TO -2000 STEP -
1 :: CALL PEEK(J,A):: A=A-(A
<33)*(A+32):: A=A+(A>143)*(A
/2):: R=R+1+(R=24)*24 :: CAL
L HCHAR(R,1,A,32)
190 C=C+1+(C=32)*32 :: CALL
VCHAR(1,C,A,24):: NEXT J ::
GOTO 100
```

Unlike most of the number games played against the computer, you can win this one -

```
100 CALL CLEAR :: CALL SCREE
N(16):: DISPLAY AT(3,8):"THE
 `37' GAME" !by Jim Peterson
110 DISPLAY AT(5,1):" We wil
l take turns picking":"a num
ber from 1 to 5, but":"not t
he number that was just":"pi
cked."
120 DISPLAY AT(10,1):" The n
umbers we pick will be":"add
ed to the total count."
130 DISPLAY AT(13,1):" Whoev
er reaches 37 is the":"winne
r, but if you go over":"37 y
ou lose."
140 CALL SHOW(20,1,"Press an
y key to start")
150 CALL KEY(0,K,S):: IF S=0
 THEN 150
160 DATA 4,11,17,24,30,37
170 DATA 262,330,392,523,523
180 DATA 1047,784,659,523,52
3
190 C,P=0 :: CALL CLEAR :: C
ALL MAGNIFY(2):: R=10 :: FOR
 J=1 TO 5 :: CALL SPRITE(#J,
48+J,5,R,10):: R=R+30 :: NEX
T J
200 CALL SHOW(24,1,"(Y)ou or
 (C)omputer first?"):: ACCEP
T AT(24,28)VALIDATE("YC")SIZ
E(1):Q$ :: DISPLAY AT(24,1):
""
210 IF Q$="C" THEN CALL SHOW
(22,8,"I pick 4"):: CALL COL
OR(#4,1):: P=4 :: C=4 :: CAL
L SHOW(3,10,"COUNT=4")
220 CALL SHOW(20,8,"Pick you
r number"):: ACCEPT AT(20,26
)VALIDATE("12345"):N :: IF N
=P THEN 220
230 IF P>0 THEN CALL COLOR(#
P,5)
240 CALL COLOR(#N,1):: P=N :
: C=C+N :: CALL SHOW(3,10,"C
OUNT= "&STR$(C)):: IF C=37 T
HEN 320 ELSE IF C>37 THEN 34
0
250 RESTORE 160
260 READ X :: IF C<X THEN B=
X-C ELSE IF X<37 THEN 260
270 CALL SHOW(22,8,"I'm thin
king..."):: FOR Y=1 TO 700 :
: NEXT Y
280 IF B>5 AND B/2=INT(B/2)T
HEN B=B/2
290 IF B>5 OR B=P THEN B=1-(
P=1)
300 CALL SHOW(22,8,"I pick "
&STR$(B)):: CALL COLOR(#P,5)
:: CALL COLOR(#B,1):: P=B ::
 C=C+B :: CALL SHOW(3,10,"CO
UNT= "&STR$(C))
310 IF C=37 THEN 340 ELSE IF
 C>37 THEN 320 ELSE 220
320 RESTORE 170 :: FOR J=1 T
```

```
0 5 :: READ F :: CALL SOUND(
100,F,5,F*1.03,5):: NEXT J :
: CALL SHOW(12,8,"YOU WIN!")
330 CALL SHOW(15,8,"Play aga
in? (Y/N)"):: ACCEPT AT(15,2
6)VALIDATE("YN"):Q$ :: IF Q$
="N" THEN STOP ELSE 190
340 RESTORE 180 :: FOR J=1 T
0 5 :: READ F :: CALL SOUND(
300,30000,30,30000,30,F,30,-
4,5):: NEXT J :: CALL SHOW(1
2,8,"YOU LOSE!"):: GOTO 330
350 SUB SHOW(R,C,T$):: FOR J
=1 TO 10 :: DISPLAY AT(R,C):
" " :: DISPLAY AT(R,C):T$ ::
 NEXT J :: SUBEND
```

A couple more peculiarities of the computer -

```
100 DISPLAY AT(3,8)ERASE ALL
:"POS PUZZLE #1": :"        f
rom Tigercub"
110 DISPLAY AT(9,1):"Why doe
s the computer say":"that X=
1 if you answer the":"prompt
 with the Enter key":"(null-
string) ?"
120 DISPLAY AT(14,1):"110 IN
PUT M$"
130 DISPLAY AT(15,1):"120 X=
POS(""TESTING"",M$,1)::":"PR
INT X :: GOTO 100"
140 !POS PUZZLE #1 - why doe
s the computer say that X=1
if you answer the prompt wit
h Enter (null-string) ?
 - Jim Peterson
150 INPUT M$
160 X=POS("TESTING",M$,1)::
PRINT X :: GOTO 140
```

And -

```
100 DISPLAY AT(3,8)ERASE ALL
:"POS PUZZLE #2": :"        f
rom Tigercub"
110 DISPLAY AT(7,1):"Why doe
s the computer say":"that th
e first position of":"null-s
tring is at whatever":"posit
ion it is told to start":"se
arch at?"
120 DISPLAY AT(13,1):"100 M$
="""""""""
130 DISPLAY AT(14,1):"110 DI
SPLAY AT(20,1):""POS?"" :: A
CCEPT AT(20,6):P"
140 DISPLAY AT(16,1):"120 X=
POS(""TESTING"",M$,P):: DISP
LAY AT(22,1):""X="";X :: GOT
0 110"
150 M$=""
160 DISPLAY AT(21,1):"POS?"
:: ACCEPT AT(21,6):P
170 X=POS("TESTING",M$,P)::
DISPLAY AT(23,1):"X=";X :: G
OTO 160
```

Here is an improvement to the PRINTSPEAKER in Tips #40 - in lines 130 and 160, change the CHR$(1)&"1" to CHR$(3)&"255" . This will avoid problems if the program being converted opens FILE #1.

Irwin Hott informs me that assembly routines which have been imbedded into XBasic programs, using ALSAVE or SYSTEX, can be saved to cassette and reloaded. This could be very useful for those who have a stand-alone or "matchbox" 32k.

And, a mini-game for you to have fun with or improve on -

```
1 !      2-LINE GAME
      by Jim Peterson
 - use S&D keys to paint the
white line on the highway
2 !if it is too easy, change
the 6 in A$=RPT$(CHR$(143),6
) to 5 and the 5 in C>T+5 to
 4
100 CALL CLEAR :: A$=RPT$(CH
R$(143),6):: CALL COLOR(14,2
,2,2,16,16):: CALL SCREEN(4)
:: T=11 :: C=14 :: CALL HCHA
R(22,C+2,42):: RANDOMIZE
110 T=T+INT(3*RND-1)+(T=21)-
(T=1):: PRINT TAB(T);A$ :: C
ALL KEY(3,K,S):: C=C+(K=83)-
(K=68):: CALL HCHAR(22,C+2,4
2):: IF C<T OR C>T+5 THEN ST
OP ELSE 110
```

And finally, one of the best examples of compact programming I have ever seen -

```
1 !JOHN WITTE'S 3-LINE VERSI
ON OF JOHN WILLFORTH'S WAVE
POWER - PUBLISHED IN GREATER
 OMAHA UG NEWSLETTER
100 CALL CLEAR :: A$(1)="ABC
DEFGFEDCBA" :: FOR I=1 TO 7
:: CALL CHAR(72-I,RPT$("0",2
*I-2)&"FFFF",47,"30303EFF7F3
E1E04"):: A$(I+1)=SEG$(A$(I)
,2,12)&SEG$(A$(I),2,1):: NEX
T I
110 CALL SPRITE(#5,47,2,180,
180,-23,0,#6,47,2,80,100,-23
,0):: CALL MAGNIFY(2)
120 FOR I=1 TO 12 :: PRINT A
$(I+(I>7)*2*(I-7))&A$(1+I+(I
>6)*2*(I-6)):: NEXT I :: GOT
0 120
```

Memory full
Jim Peterson

TI—BASE Tutorial 4 Page 3

NorthCoast 99'ers (C) Martin A. Smoley

The next 10 REPLACE commands are the same as the one I just described. There is however, a difference in what is happening to the field in REPLACE 2.ID WITH 1.ID. Take a look at ID on the structure screens on the previous pages. You should notice that ID in TNAMES is a N)umeric field and it is a C)haracter field in NEWNAMES. This is one way we can change a field type. This change is not important at this time, but I received a question on the matter so I thought others might be having this problem. You can also convert C)haracter fields to N)umeric fields in the same manner. If that is the case, you must remove all characters from every field first. In other words the field may contain numbers only, at the time of conversion. You can also leave a particular field blank and use EDIT to type in data at a later time. "OK, the last REPLACE." REPLACE NUMT WITH NUMT + 1 is an accumulator (remember from last month?). We are telling TIB to take whatever number is in the variable NUMT, add 1 to it, and place the new total back into NUMT. So the next time we REPLACE 2.NM WITH NUMT the result will be 2, 3, 4 and so on. The next line (SELECT 1) is important. We must reSELECT number 1 (TNAMES) before the MOVE directive which is in the following line. In NEWNAMES we are at the EOF and have no place to move to. Also, when we hit the ENDWHILE and loop back to the WHILE .NOT. (EOF) statement we must already be in the full database for the same reason. We will continue to jump back and forth, add new space to NEWNAMES and move data from TNAMES to NEWNAMES until we hit the End Of File (EOF) in box number 1 (which is TNAMES). "Sounds easy right? Well it is for the CF." At the EOF TIB jumps out of the loop CLOSEs ALL the open DBs and RETURNs to the CF named MOVED1. In MOVED1 it simply turns ON all the stuff we turned OFF previously and RETURNs you to the DP. FYI: First, I am covering the business of SELECTing a slot over again because if you wish to really use TI-BASE you must fully understand the basics. If TIB can handle 17 fields in a database and it can open 5 databases at the same time (slots 1-5), then you have the potential of actively working with 85 fields at the same time. We have just worked with 23 fields at the same time. Think of how complicated a situation it could be with 85. This is why you must take the time to fully understand the basics of this language. Next, I still prefer to use FunnelWeb to write and edit my CFs. I realized that this will not be convenient for you under certain conditions, for example, if you have TIB running and you would like to change the name of the DB you will use in slot number 1. If the CF is to big to load with MODIFY COMMAND, you must leave TIB, load FunnelWeb, make the correction and then reload TIB. So, from now on I will try to keep the CF segments small enough to be modified without leaving TIB. I still recommend that you use some means to produce D/V 80 type CFs. Last for now, I have changed my mind about creating large menu-type systems at this time. You may recall my mentioning this idea last month. I will stay with smaller and I hope more easily understood utility type CFs for a while. Several people have told me that they are already lost. So I'll slow down a little and try to accommodate everyone, if possible.

"OK, back to work." While working on this tutorial and writing MOVED1 and MOVED2 I definitely did not get things right the first time. This created a related problem that covers some new items in that old manual. When I ran MOVED1 it fille NEWNAMES with junk, I literally mean junk. Since APPEND means stick more stuff on the end, the second time there was twice as much junk in NEWNAMES. What I needed was another CF that woul clean out NEWNAMES. So I whipped up the CF named CLEARD.

```
*                     10/28/88
* Command File CLEARD
* Save as CLEARD/C
*   Clear Data from NEWNAMES
CLEAR
CLOSE ALL
* SET TALK OFF
SET RECNUM OFF
SET HEADING OFF
USE NEWNAMES
SORT OFF
TOP
WHILE .NOT. (EOF)
 DELETE RECORD
 MOVE
ENDWHILE
 PACK
* Second time through.
TOP
WHILE .NOT. (EOF)
 DELETE RECORD
 MOVE
ENDWHILE
 PACK
* That should do it.
CLOSE ALL
SET RECNUM ON
SET HEADING ON
SET TALK ON
RETURN
```

A SPECIAL NOTE FROM MARTY

I am presently testing TI-Base Version 2.0.

The improvements look very very impressive.

This CF opens or USEs NEWNAMES, unSORTs the DB and starts a the TOP or first record. The WHILE loop loops until it hit the EOF, and while it is looping, TIB is DELETEing records. I this application DELETE RECORD actually means mark the recor TIB is presently looking at for later removal. This is for record only. Therefore, we MOVE to each record, one at a time and mark them all. When we hit the EOF, the WHILE kicks ou and we execute PACK. PACK is a program segment of TIB and i resides on the TIB program disk. It permanently removes th records which have been marked for deletion. For some reason unknown to me, I kept winding up with a record still left i NEWNAMES. Not always the same record but a record. Th easiest way to handle this was to rerun the loop. This bring NEWNAMES up empty every time. You can check this by typing US NEWNAMES at the DP and then DISPLAY ALL. You will get database empty message. Well, I'm going to break off here. still have enough material for hundreds of tutorials unfortunately it's all in my head. Remember that I have tutorial disk available and you can join the NorthCoast User Group, see tutolial 3 for details. And send those questions t me, I need to know what you need to know. Good luck. Marty.

TI-BASE - From INSCEBOT TUTORIAL 5.1 By Martin Smoley

NorthCoast 99'ers - Dec. 17, 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.

Well here it is December already. Gee! time flies when you're having fun. This month I'm going to change my mind again. I said I didn't like System type setups, so this month I'm doing a system for you. This is my Version 1.02 and my 1988 finale. This tutorial will contain practically all programming, with only a couple comments from me. The whole thing works, so if this is what you wanted, your time is here. This type of program runs too slow for me, but once it's finished you can run the whole thing with a few number entries. TIBSYS, which is listed below, runs all the other CFs (more or less). So, to get the system going you would type DO DSK2.TIBSYS <E>. You can find your way through the system by the order of the DO DSKn.XXX commands. You'll notice that all the remark statements are at the end of the CFs. This is because the processor doesn't read anything after RETURN. Putting your remarks after that point will speed things up. I have also kept the size of the CFs down so you can edit any of them using MODIFY COMMAND. I try to use the same fieldnames (NM, FN, LN, MI) for all my databases. This allows me to use this type of programming on several DBs by merely changing USE NEWNAMES to USE (WHATEVER). Except for possible minor field length problems this System CF should be usable for many things.

```
 LOCAL ? N 2 0
 LOCAL SEL N 2 0
REPLACE ? WITH 0
     DO DSK2.PREP1
     DO DSK2.SYSSCR
USE NEWNAMES
TOP
  WHILE .NOT. (?)
     DO DSK2.INFSCR1
     DO DSK2.SLCASE
  ENDWHILE
     DO DSK2.FIN1
RETURN
*
* TIBSYS          Save as TIBSYS/C
* ******     TI-Base System 12/1/88
*
 *********************************
```

NOTE: Don't type in the last two lines of each CF. I'm refering to the *, and the ***********s. I put those in to keep things separated.

```
CLEAR
*   Pre-Program Preparation
*
* PREP1  Save as  PREP1/C
* *****
*
CLOSE ALL
SET HEADING OFF
SET RECNUM OFF
COLOR WHITE DARK-BLUE
SET TALK OFF
WAIT 5
RETURN
*
 *************************
```

```
CLEAR
 COLOR BLACK GRAY
 WRITE 3,9,"This is a TI-Base System."
 WRITE 5,9,"It is a club type system"
 WRITE 7,9,"to produce a club Roster,"
 WRITE 9,9,"a complete set of labels,"
 WRITE 11,9,"or search for individual"
 WRITE 13,9,"names and print more thar
 WRITE 15,9,"one label for a specific"
 WRITE 17,9,"name on the list."
 WRITE 19,12,"** USEs NEWNAMES **"
  WAIT 4
  COLOR WHITE DARK-BLUE
RETURN
*
* SYSSCR          Save as SYSSCR/C
* ******     System Screen 12/1/88
*
 ********************************
```

```
CLEAR
   REPLACE SEL WITH -1
 WRITE 2,8,"** Make A Selection **"
 WRITE 4,10,"> 0 < To Quit CF"
 WRITE 6,10,"> 1 < Print Roster"
 WRITE 8,10,"> 2 < Print All Labels"
 WRITE 10,10,"> 3 < Print Spec. Labels
 WRITE 12,10,"> 4 < Edit NEWNAMES"
 WRITE 14,10,"> 5 < Append To NEWNAMES
WHILE (SEL<0) .OR. (SEL>5)
   WRITE 22,4,"Enter 0-5"
   READ 22,15,SEL
   WRITE 22,3,"                       "
ENDWHILE
CLEAR
RETURN
*
* INFSCR1       Save as INFSCR1/C
* *******    Info Screen 1  12/1/88
*
 *********************************
```