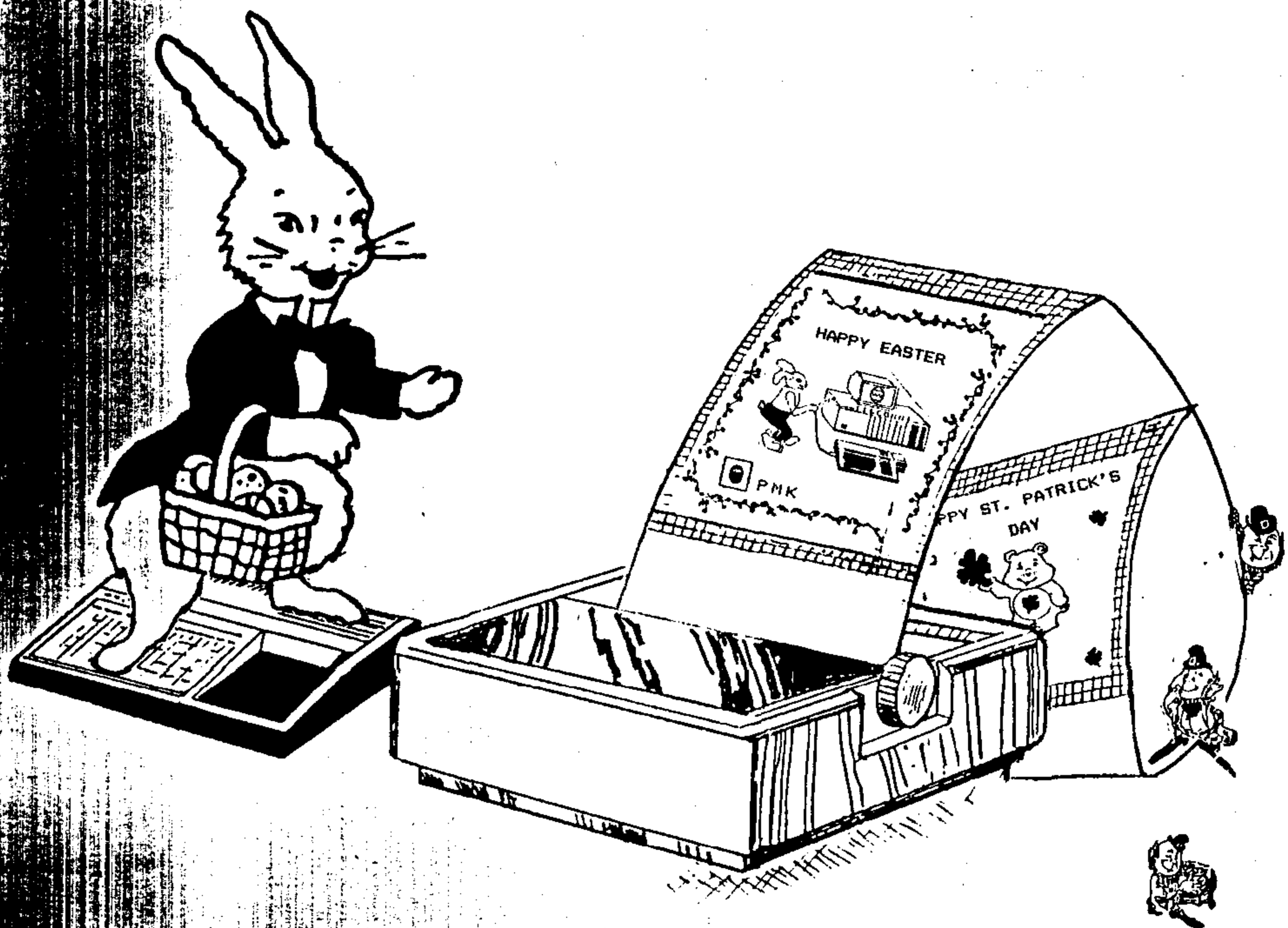
Spirit of 99

CENTRAL OHIO NINETY NINERS INC.

THE OFFICIAL NEWSLETTER OF THE CENTRAL OHIO NINETY-NINERS INC.

PUBLISHED MONTHLY IN COLUMBUS OHIO

MARCH



$1.50 VOL.7 NO.3 MAR 1989

Spirit of 99

THE OFFICIAL NEWSLETTER OF CENTRAL OHIO NINETY-NINERS

COPYRIGHT C 1985 Central Ohio Ninety-Niners Incorporated (C.O.N.N.I.). Columbus Ohio 43212, USA. All rights reserved. Spirit of 99 is published monthly for Central Ohio Ninety-Niners Inc. by C.O.N.N.I. members and is the official news letter of C.O.N.N.I. User Group.

Editorial, advertising and subscription address is:
181 HEISCHMAN AVE
WORTHINGTON,OH 43085

Subscription rate (USA) $20.00 /1 year (12 issues). Foreign subscription rates available upon request. Third class postage paid at Columbus, Ohio.

CHANGE OF ADDRESS: Send both OLD and NEW address to: Subscription address above. WE assume no responsibility for manscripts, programs (tape or disk)not ac-companied by return postage. Letters to the Editor become property of Spirit of 99. If published, we reserve the right to edit at our discretion.

OPINIONS EXPRESSED HEREIN ARE THE AUTHORS AND ARE BASED ON VALID DOCUMENTABLE RESEARCH.THEY DO NOT NECESSARILY REFLECT THE OPINIONS OF THE PUBLISHER.

We will not knowingly publish copyright material without the permission of the author and credit due.

All programs published herein are of public domain unless otherwise noted.

Other non-profit user groups may use material from this newsletter only if source and credit is given.

Central Ohio Ninety Niners Inc. is a non profit organization comprised of MEMBERS who own or use the TI99/4A computer and it's related pro-ducts and have paid a yearly membership fee of $28.00 and whose main objective is the exchange of Educational and Scientific information for the purpose of computer literacy.

C.O.N.N.I. meetings are held the 2nd Sat-urday of each month at the Martin Janis Senior Center - East Eleventh Ave. at the Ohio State fairgrounds.Meeting time is at 9 am. Meetings are open to the public. Membership dues ($28.00) are payable yearly to C.O.N.N.I. and cover the immediate family of the member. (An application has been placed in this newsletter for your convenience) Please address it to:

EVERETT WADE
179 ERIE ROAD
COLUMBUS, OH 43214

ADVERTISEMENT:

We do accept commercial advertisement at The following rates:

Business Card(2x3.5):	$5.00/issue
1/4 Page:	$25.00
1/2 Page:	$45.00
Full Page:	$75.00

Write this newsletter for other size arrang-ements.

All ads should be submitted (camera ready) to advertising address above,payment enclosed. Members ads are published at no cost. (Limit of 25 words and must not be commercial please.)

EDITOR.....JEAN HALL
ASSIST.....CAROLE PARKINS

OFFICERS

PRESIDENT.....DICK BEERY
VICE PRES.....JIM SEITZ
SECRETARY.....CHARLES OSMENT
TREASURER.....MIKE CHANEY
LIBRARIAN.....CHUCK GRIMES

ANNOUNCEMENTS

Dues are usually paid at or before the March meeting, and are $28 per year for full membership, library and voting privileges, plus the newsletter. You may also pay your dues in two installments if desired: $14 in March and $14 in September. If only the newsletter is desired, then payment is $20 per year. Those who join during other months of the year pay a lesser, pro-rated amount:

Mar---28.00	Apr---25.75	May---23.50	Jun---21.00	Jul---18.75
Aug---16.50	Sep---14.00	Oct---11.25	Nov----9.50	Dec----7.00
Jan----4.75	Feb----2.50			

Fill out an application blank (one on the back of this newsletter), make a check out to C.O.N.N.I. and give it to Everett Wade, the membership registrar, at one of the meetings or mail to him at the following address:

Everett Wade 179 Erie Rd Columbus, OH 43214

MEETING AGENDA ----- SATURDAY 11 MAR 1989

9 AM LIBRARIES OPEN
BULLETINS AVAILABLE
REGISTRATION - MEMBERSHIP
MICROpendium magazines for sale

9:25 AM QUESTION AND ANSWER SESSION

9:50 AM BUSINESS MEETING

10:20 AM DEMO-HOUSEHOLD BUDGET MANAGEMENT by JIM SEITZ

DEMO-TAX DEDUCTION FILER by JIM SEITZ

12:00 PM WE MUST BE OUT OF THE BUILDING BY NOON!!!!!

DON'T FORGET CONNI DUES

+WELCOME TO NEW MEMBERS+
+ AND NEWSLETTER +
+ SUBSCRIBERS +

DAVID L. TRUESDALE II

+ COFFEE ANYONE? +
+ SATURDAY MORNINGS +

Call Jim Seitz (875-5532) to be a host or hostess. SIGN UP IF YOU WANT ANY COFFEE!!

MAR - CHARLES OSMENT
APR -

+ WEDNESDAY EVENING +
+ MEETING - MAR 22 +

7:30 PM AT MCcDONALD'S CORNER OF CLEVELAND AVE AND MAIN IN WESTERVILLE

HOPE TO SEE YOU THERE!!

FROM THE PRESIDENT'S DESK

by DICK BEERY

With the coming of March, we turn the calendar to a new fiscal year for C.O.N.N.I. Because of the steady influx of new materials for the 99/4A and the Geneve, we feel we have many good things to look forward to in the coming weeks and months. Funnelweb(c) is now available in version 4.13; Telco(c) keeps appearing in new versions, as do support files for it. The Miami Users' Group just released a new Boot Menu program, 1989 version, as well as a new HRD Menu program for HORIZON ramdisks, version 7.36. Many programs are being expanded or modified to allow their use with the hard-disk controller now available from Myarc, Inc.

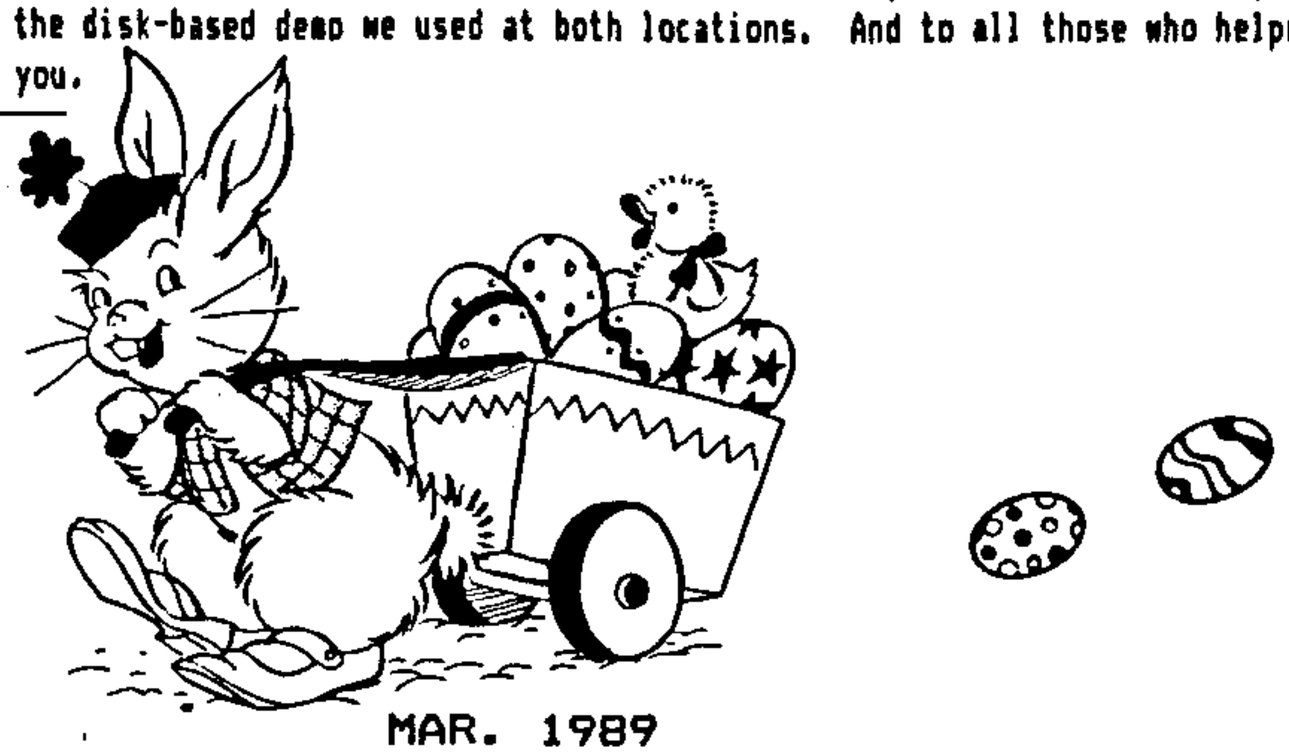
The genealogy program from Australia is still on hold for us. Jim Wright? has not yet responded to our suggestions, the major one of which was to provide a four-generation chart printout in a form recognizable to, and usable by, American genealogical researchers. We hope this change will soon be forthcoming, as well as others suggested by us and by other users around the world.

Ti-Base(c), now in version 2.2, is a truly remarkable database, from all I have been able to discover. At the evening meeting of C.O.N.N.I. on January 25th, Jim Klein ably demonstrated its capabilities. I myself have tried using it (a borrowed copy), and it seems to be very user-friendly. According to the manual, the database limitations are "255 characters per field; 17 fields per record; and 16129 records per database." One can have up to five databases open at one time and freely manipulate and interchange data among them. And there is a lot more! I like it so much that I am sending for a copy of my own this week. Other club members who are interested in doing the same should let Jim Klein, our contact person, know, so that he can make the club discount available. When I see programs such as this one and the soon-to-be-delivered Press(c), with reportedly vast file capability limited only by the amount of storage available, I feel that all of my own personal needs can probably be met through using the T.I. and its ever-expanding capabilities. And I feel sure that many could agree. We will continue to report and to demonstrate new software as it appears, so as to help you, the members, stay up-to-date and totally aware of all the excitement that the TI can and does offer.

We have positive feelings about our recent demonstrations, at Northland Mall on Saturday, January 28, and at the Ohio Expositions Center on Sunday, February 19. While we did not attract new members in quite the dramatic fashion of the results reported by the West Penn group, we were well-received, talked to many individuals, and had a show of interest on the part of between four and six people. Whether they will join the group remains to be seen. In any case, many of us feel that the exposure, the letting the public know that we are still alive and prospering, is a worthwhile goal that may, in the long run, lead to an increase in membership. Thanks to Gary Cox of the Mid-South TI99/4A
users group in Germantown, TN, for sending us information about the February 19th show. We found the management friendly and cooperative, and the show space to be roomy and well-planned. The sponsor is Shows Unlimited, located in Annapolis, MD. If other users' groups that receive our newsletter would like more information write to either Jean Hall or me at our official address on the back cover.

What excitement have you generated with your TI lately? Tried using any new programs? Found a new way to use your wordprocessor? Tried writing some program within your range of capabilities? Gone back and rediscovered some of the wonderful "oldies" of several years ago? Try your hand at writing an article for the newsletter--a review of some program you liked--or didn't, or maybe some special use you have for the computer that others might like to hear about. Etc.

In conclusion, thanks to all those who gave up precious weekend time to man the tables at both demonstrations. We appreciate it! And special thanks to Bill Wood for organizing the Northland one, and to Ken Marshall,Jr. for all his hours of work on both the banners and the disk-based demo we used at both locations. And to all those who helped in so many other ways. Good to know we can count on you.



SPEECH PART 2
BY IRWIN HOTT

Even though this is part 2, I did not write part 1. It was in the February Spirit of '99 Newsletter taken from the Hunter Valley User Group. After looking at it, I realized that something was missing. I have included a modified version of that missing program here. I also made a minor change in the original program as listed in the newsletter. It was supposed to run in Mini-Memory or E/A Basic as well as XB. Line 490 had a double colon in it which would have worked only in XB so:

```
490 CALL KEY(0,K,S)
500 IF S<1 THEN 490 ELSE IF
K=89 THEN 100 ELSE END
```

The purpose of that program is to take the 4-digit number for a CALLSAY word as found in Editor/Assembler manual (page 422) and change it into a series of numbers that will be recognized by Basic. The hex numbers for "HELLO" are 351A. If you input them into the "SPEECH" program as found in the February Disk-of-the-Month and the February Spirit of '99 Newsletter you will get this:

```
74  65  69  67  64  80
```

The program converts to decimal, adds 64, reverses the order of the numbers and adds 64, and 80 at the end.

I got that far with no problem. Now how the h— do I get speech?

I remembered a demonstration program that ran in Basic and used speech, music and graphics all at once. I found it on DOM-7/86#A as E/A-Basic. The important lines as far as we are concerned are:

```
190 @=-27648
600 CALL LOAD(@,70,"",@,65,"
",@,72,"",@,70,"",@,64,"",@,
80)
```

Why the, "" I do not know. They are not necessary.

```
600 CALL Load(@,70,@,65,@,72
,@,70,@,64,@,80)
```

will work as well.

I tried that phrase, in the normal manner, CALL SAY ("THAT IS INCORRECT") and discovered an interesting thing. It did not work. After a considerable amount of digging I found that the Speech Editor manual indicated that phrases such as that had to be written as CALL SAY("#THAT IS INCORRECT#"). This is not mentioned in the Extended Basic or Speech Synthesizer manuals as far as I can determine.

Well now what is all of this good for? The best reason to use CALL LOAD method instead of the CALL SAY routine is that the speech starts faster and program execution is not halted during the CALL LOAD as it is during the CALL SAY.

Here is the program that was not shown in the February newsletter. It is SPEECH1 on the February Disk-of-the-Month. If you want to run it in Mini-Memory or E/A Basic delete line 100.

```
100 GOTO 110 :: CALL CLEAR :
: CALL INIT :: CALL LOAD ::
I,S :: !@P-
110 CALL CLEAR
120 CALL INIT
130 S=-27648
140 PRINT "      READY TO ST
ART"
150 CALL LOAD(S,67,S,75,S,70
,S,69,S,64,S,80)
160 GOSUB 610
170 CALL LOAD(S,74,S,65,S,69
,S,67,S,64,S,80)
180 GOSUB 610
190 CALL LOAD(S,68,S,71,S,73
,S,70,S,64,S,80)
200 CALL LOAD(S,64,S,76,S,71
,S,68,S,64,S,80)
210 CALL LOAD(S,66,S,67,S,74
,S,67,S,64,S,80)
220 CALL LOAD(S,70,S,73,S,70
,S,70,S,64,S,80)
230 GOSUB 610
240 CALL LOAD(S,67,S,73,S,71
,S,67,S,64,S,80)
250 CALL LOAD(S,74,S,68,S,68
,S,67,S,64,S,80)
260 CALL LOAD(S,78,S,67,S,76
,S,66,S,64,S,80)
270 CALL LOAD(S,71,S,67,S,70
,S,65,S,64,S,80)
280 CALL LOAD(S,74,S,72,S,75
,S,67,S,64,S,80)
290 CALL LOAD(S,69,S,64,S,68
,S,68,S,64,S,80)
300 GOSUB 610
310 CALL LOAD(S,67,S,73,S,71
,S,67,S,64,S,80)
320 CALL LOAD(S,73,S,77,S,76
,S,65,S,64,S,80)
330 CALL LOAD(S,67,S,64,S,68
,S,71,S,64,S,80)
340 CALL LOAD(S,64,S,65,S,77
,S,65,S,64,S,80)
350 CALL LOAD(S,76,S,77,S,76
,S,68,S,64,S,80)
360 CALL LOAD(S,77,S,66,S,68
,S,66,S,64,S,80)
370 GOSUB 610
380 CALL LOAD(S,78,S,75,S,65
,S,71,S,64,S,80)
390 CALL LOAD(S,73,S,77,S,76
,S,65,S,64,S,80)
400 CALL LOAD(S,69,S,78,S,79
,S,68,S,64,S,80)
410 CALL LOAD(S,72,S,66,S,78
,S,66,S,64,S,80)
420 CALL LOAD(S,75,S,70,S,74
,S,71,S,64,S,80)
430 CALL LOAD(S,77,S,71,S,67
,S,68,S,64,S,80)
440 CALL LOAD(S,65,S,68,S,76
,S,68,S,64,S,80)
450 CALL LOAD(S,68,S,71,S,73
,S,70,S,64,S,80)
460 CALL LOAD(S,73,S,78,S,75
,S,67,S,64,S,80)
470 CALL LOAD(S,76,S,77,S,76
,S,68,S,64,S,80)
480 CALL LOAD(S,77,S,78,S,74
,S,67,S,64,S,80)
490 GOSUB 610
500 CALL LOAD(S,73,S,78,S,71
,S,71,S,64,S,80)
510 GOSUB 610
520 CALL LOAD(S,67,S,73,S,71
,S,67,S,64,S,80)
530 CALL LOAD(S,71,S,70,S,66
,S,68,S,64,S,80)
540 CALL LOAD(S,68,S,78,S,70
,S,65,S,64,S,80)
550 CALL LOAD(S,76,S,67,S,66
,S,66,S,64,S,80)
560 CALL LOAD(S,79,S,78,S,72
,S,66,S,64,S,80)
570 GOSUB 610
580 CALL LOAD(S,72,S,68,S,65
,S,67,S,64,S,80)
590 PRINT "              GOOD B
YE"
600 STOP
610 FOR I=1 TO 400
620 NEXT I
630 RETURN
```

It is really pretty straight forward. You will hear words and phrases through your speech synthesizer. The GOSUB 610 is a delay loop. Try substituting :

```
610
620
```

and see what happens.

Thanks to Curt Borders for typing in the original program for me. I added line 100, deleted the "", from each CALL LOAD since don't seem to be essential and made the delay loop into a subroutine

I just tried an experiment. I ran the preceeding program as is and it took 29.2 seconds in Extended Basic. I then replaced each CALL LOAD with the appropriate word or phrase. Line 150 became:

```
150 CALL SAY("#READY TO STAR
T#")
```

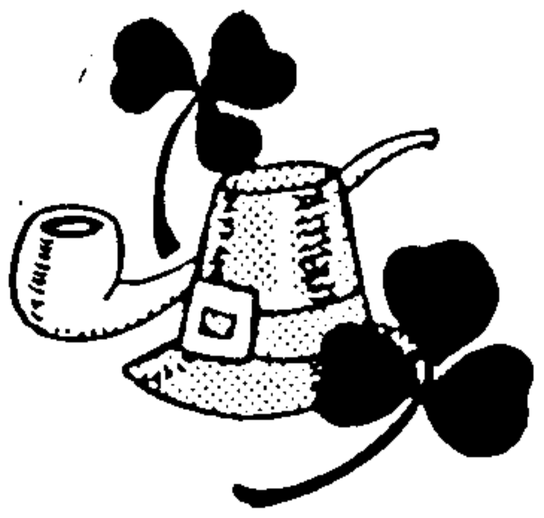
etc. That took 40.9 seconds to run. That may seem to be an inefficient way to use CALL SAY but the following modification to the program took 40.2 seconds to run:

```
100 GOTO 110 :: CALL SAY ::
CALL CLEAR :: CALL INIT :: I
 :: !@P-
110 CALL CLEAR
120 CALL INIT
140 PRINT "      READY TO ST
ART"
150 CALL SAY("#READY TO STAR
T#")
160 GOSUB 610
170 CALL SAY("HELLO")
180 GOSUB 610
190 CALL SAY("THE+NAME+IS+#T
EXAS INSTRUMENTS#")
230 GOSUB 610
240 CALL SAY("I+HAVE+FORTY+E
IGHT+K+MEMORY")
300 GOSUB 610
310 CALL SAY("I+CAN+USE+CASS
ETTE+OR+DISKETTE")
370 GOSUB 610
380 CALL SAY("YOU+CAN+PLAY+G
AMES+WITH+ME+ON+THE+KEYBOARD
+OR+JOYSTICK")
490 GOSUB 610
500 CALL SAY("#WHAT WAS THAT
#")
510 GOSUB 610
520 CALL SAY("I+MADE+A+DATA+
ERROR")
570 GOSUB 610
580 CALL SAY("GOODBYE")
590 PRINT "              GOOD B
YE"
600 STOP
610 FOR I=1 TO 400
620 NEXT I
630 RETURN
```

The + sign is used as word separator that denotes 0 seconds pause between words.

Have fun playing with speech.

!UUUUUUUUUUUUUU



TIPS FROM THE TIGERCUB

♫

#55

Tigercub Software
156 Collingwood Ave.
Columbus OH 43213

I am still offering over 120 original and unique entertainment, educational and utility programs at just $1.00 each, or on collection disks at $5.00 per disk.

The contents of the first 52 issues of this newsletter are available as ready-to-run programs on 5 *Tips Disks at $10 each.*

And my three Nuts & Bolts Disk, $15 each, each contain over 100 subprograms for you to merge into your own programs to do all kinds of wonderful things.

My catalog is available for $1, deductable from your first order (specify TIGERCUB catalog).

****

TI-PD LIBRARY

I have selected public domain programs, by category, to fill over 200 disks, as full as possible if I had enough programs of the category, with all the Basic-only programs converted to XBasic, with an E/A loader provided for assembly programs if possible, instructions added and any obvious bugs corrected, *and with an autoloader* by full program name on each disk. These are available as a copying service for just $1.50 postpaid in U.S. and Canada. No fairware will be offered without the author's permission. Send SASE for list or $1, refundable for 9-page catalog listing all titles and authors. Be sure to specify *TI-PD* catalog.

****

The Tigercub has dipped a cautious paw into the cold dark mysterious waters of asembly, while still keeping a firm grip on trusty old Extended Basic. The result is an XBasic program that writes an assembly program!

The following subprogram, when merged into any program which has reidentified characters, and called after the characters have been reidentified, will write a source code which can be assembled into object code, loaded from XBasic and linked to instantly access the character set.

The source code is based on 2FONTS/S by Barry Traver, who gives credit to Mac McCormick, David Migicovsky and Karl Schuneman.

```
19000 SUB CHARSUB(HX$())
19001 DISPLAY AT(12,1)ERASE
ALL:"Source code filename?":
"DSK" :: ACCEPT AT(13,4)SIZE
(12)BEEP:F$ :: OPEN #1:"DSK"
&F$,OUTPUT
19002 DISPLAY AT(15,1):"LINK
ABLE program name?" :: ACCEP
T AT(16,1)SIZE(6):P$
19003 DISPLAY AT(18,1):"Rede
fine characters from  ASCI
I  to ASCII"
19004 ACCEPT AT(19,7)VALIDAT
E(DIGIT)SIZE(3):F
19005 ACCEPT AT(19,21)VALIDA
TE(DIGIT)SIZE(3):T
19006 PRINT #1:TAB(8);"DEF";
TAB(13);P$ :: PRINT #1:"VMBW
   EQU  >2024" :: PRINT #1:"
STATUS EQU  >837C"
19007 NB=(T-F+1)*8 :: CALL D
EC_HEX(NB,H$):: A=768+F*8 ::
 CALL DEC_HEX(A,A$)
19008 FOR CH=F TO T :: IF CH
<144 THEN CALL CHARPAT(CH,CH
$)ELSE CH$=HX$(CH)
19009 IF FLAG=0 THEN PRINT #
1:"FONT";:: FLAG=1
19010 FOR J=1 TO 13 STEP 4 :
: M$=M$&")"&SEG$(CH$,J,4)&",
" :: NEXT J :: M$=SEG$(M$,1,
23)&"  @"&CHR$(CH)
19011 PRINT #1:TAB(8);"DATA
"&M$ :: M$="" :: NEXT CH
19012 PRINT #1:P$;TAB(8);"LI
   R1,FONT" :: PRINT #1:TAB(
8);"LI   R0,>"&A$ :: PRINT #
1:TAB(8);"LI   R2,>"&H$
19013 PRINT #1:TAB(8);"BLWP
@VMBW":TAB(8);"CLR  @STATUS"
:TAB(8);"RT":TAB(8);"END" ::
 CLOSE #1
19014 SUBEND
19015 SUB DEC_HEX(D,H$)
19016 X$="0123456789ABCDEF"
:: A=D+65536*(D>32767)
19017 H$=SEG$(X$,(INT(A/4096
)AND 15)+1,1)&SEG$(X$,(INT(A
/256)AND 15)+1,1)&SEG$(X$,(I
NT(A/16)AND 15)+1,1)&SEG$(X$
,(A AND 15)+1,1):: SUBEND
```

Now to try it out. You probably know that CALL CHARSET will restore reidentified characters below ASCII 96 to normal form, but not those above, so *let's write a routine to* restore those. Clear the memory with NEW, merge in the above, which you should have SAVED with -

SAVE DSK1.CHARSUB,MERGE

by MERGE DSK1.CHARSUB. Add a line -

100 CALL CHARSUB(HX$()) and RUN. Answer the filename prompt with DSK1.OLDLOW/S, the next prompt with OLDLOW and select ASCII 97 to 127.

When done, insert the Editor/Assembler module and its disk Part A. Select Assembler, Y to load assembler, give the source code DSK1.OLDLOW/S, object code DSK1.OLDLOW/O, just press Enter at next prompt, and R for options. You should get 0000 ERRORS.

Now key in this routine to test your program.

```
100 CALL INIT :: CALL LOAD("
DSK1.OLDLOW/O"):: FOR CH=33
TO 126 :: CALL CHAR(CH,"FF81
81818181FF"):: PRINT CHR$(
CH);:: NEXT CH
101 CALL KEY(0,K,S):: IF S=0
 THEN 101 ELSE CALL CHARSET
102 CALL KEY(0,K,S):: IF S=0
 THEN 102 ELSE CALL LINK("OL
DLOW")
110 GOTO 110
```

Press any key to restore the upper case characters by CALL CHARSET, any key again *to use the CALL LINK.*

You are now ready to use the routine to copy all kinds of character sets from the programs in your library. You don't have any such programs? Not to worry. You don't have to reidentify characters one by one with one of those graphics editor programs. You can just manipulate the existing hex codes of the normal characters. I have created nearly 50 different character sets by that method!

The space occupied by a character on the screen is *really an 8x8 square of 64* tiny dots. Various dots are turned on (colored) and off (transparent) to create a pattern - just the opposite of light bulbs on a score*board.*

And those on-and-off dots are really the binary numbers which the computer uses. But fortunately the computer lets us use hexadecimal numbers rather than binary. The following will print out a reference chart of decimal to binary to hexadecimal. You can easily convert it to dump to a printer.

```
10 DISPLAY AT(6,1)ERASE ALL:
"DEC BIN HEX"
100 FOR J=0 TO 15 :: CALL DE
C_BIN(J,B$):: CALL DEC_HEX(J
,H$):: DISPLAY AT(J+8,1):J;T
AB(5);B$;TAB(10);SEG$(H$,4,1
):: NEXT J
21020 SUB DEC_BIN(D@,B$):: D
=D@ :: IF D=0 THEN B$="0000"
 :: SUBEXIT
21021 IF D=1 THEN 21022 :: X
=D/2 :: B@$=STR$(ABS(X<>INT(
X)))&B@$ :: D=INT(X):: IF D>
1 THEN 21021
21022 B@$="1"&B@$ :: B$=RPT$
```

```
("0",4-LEN(B@$))&B@$ :: B@$=
"" :: SUBEND
21039 SUB DEC_HEX(D,H$)
21040 X$="0123456789ABCDEF"
:: A=D+65536*(D<0)
21041 H$=SEG$(X$,(INT(A/4096
)AND 15)+1,1)&SEG$(X$,(INT(A
/256)AND 15)+1,1)&SEG$(X$,(I
NT(A/16)AND 15)+1,1)&SEG$(X$
,(A AND 15)+1,1):: SUBEND
```

And this routine will show you how each letter is formed, by binary 0's (off) and 1's (on), for each key you press. I put it in merge format so you can MERGE it into any program and CALL it to examine the characters.

```
17000 SUB CHARVIEW
17001 !programmed by Jim Pet
erson Feb 1989
17002 DISPLAY AT(1,1)ERASE A
LL:"CHARACTERS IN BINARY & H
EX";;"Press any key to see
the    binary representation
 of thescreen character and
its hexcode."
17003 DISPLAY AT(8,1):"Press
 Enter to see the char-acter
."
17004 CALL KEY(0,K,S):: IF K
=13 THEN 17005 ELSE IF S=0 O
R K<32 OR K>143 THEN 17004 E
LSE 17007
17005 CALL CHAR(48,"FF"&RPT$
("81",6)&RPT$("FF",9))
17006 CALL KEY(0,K,S):: IF S
<1 THEN 17006 ELSE CALL CHAR
(48,"003844444444443800103010
10101038"):: GOTO 17004
17007 CALL CHARPAT(K,CH$)
17008 R=12 :: FOR J=1 TO 15
STEP 2
17009 H$=SEG$(CH$,J,1):: CAL
L HEX_BIN(H$,B$)
17010 DISPLAY AT(R,8):B$
17011 H$=SEG$(CH$,J+1,1):: C
ALL HEX_BIN(H$,B$)
17012 DISPLAY AT(R,12):B$ ::
 DISPLAY AT(R,18):SEG$(CH$,J
,2):: R=R+1 :: NEXT J :: DIS
PLAY AT(22,6):CH$ :: GOTO 17
004
17013 SUBEND
17014 SUB HEX_BIN(H$,B$):: H
X$="0123456789ABCDEF" :: BN$
="0000X0001X0010X0011X0100X0
101X0110X0111X1000X1001X1010
X1011X1100X1101X1110X1111"
17015 FOR J=LEN(H$)TO 1 STEP
 -1 :: X$=SEG$(H$,J,1)
17016 X=POS(HX$,X$,1)-1 :: T
$=SEG$(BN$,X$5+1,4)&T$ :: NE
XT J :: B$=T$ :: T$="" :: SU
BEND
```

And to reidentify a character, you just change the numbers and letters in the 16-digit hex code which represents the binary pattern. By writing little routines to switch those digits around, all kinds of things can be done.

For instance, the normal characters always have the top row of dots turned off, to provide spacing between lines of text on the screen. If you want taller characters you will have to double-space the lines, but you can create them by making the numerals and upper case characters consist of the 2nd-7th rows, the 7th row again, and the 8th row - it just happens to work out.

```
18000 SUB HIGHCHAR :: FOR CH
=48 TO 90 :: CALL CHARPAT(CH
,CH$):: CALL CHAR(CH,SEG$(CH
$,3,10)&RPT$(SEG$(CH$,13,2),
2)&SEG$(CH$,15,2)):: NEXT CH
 :: SUBEND
```

I made that a subprogram so you can MERGE it in and use it to modify other character sets.

If we take the hex code apart, 2 digits at a time, and reassemble it backward,

```
100 CALL CLEAR :: FOR CH=33
TO 90 :: CALL CHARPAT(CH,CH$
):: FOR J=1 TO 15 STEP 2 ::
CH2$=SEG$(CH$,J,2)&CH2$ :: N
EXT J :: CALL CHAR(CH,CH2$):
: CH2$="" :: NEXT CH
110 DISPLAY AT(12,1):"?NWOD
EDISPU":"VT EHT DENRUT OHW !
YEH" :: GOTO 110
```

That one was in my first Tips newsletter, years ago, but it is much more effective at assembly speed.

This one shades characters on their left edge by turning on the pixel to the left of the leftmost "on" pixel, if any. Also try it in combination with HIGHCHAR.

```
18001 SUB NEWCHAR3 :: FOR CH
=48 TO 122 :: CALL CHARPAT(C
H,CH$):: FOR J=1 TO 15 STEP
2
18002 CH2$=CH2$&SEG$("0367CD
EF",POS("01234567",SEG$(CH$,
J,1),1),1)&SEG$(CH$,J+1,1)::
 NEXT J :: CALL CHAR(CH,CH2$
):: CH2$="" :: NEXT CH :: SU
BEND
```

This one uses HIGHCHAR to heighten the character and then blanks out three rows. Try following it with NEWCHAR3.

```
18030 SUB NEWCHAR10 :: A$="0
0" :: FOR CH=48 TO 90 :: CAL
L CHARPAT(CH,CH$):: CH$=SEG$
(CH$,3,10)&RPT$(SEG$(CH$,13,
2),2)&SEG$(CH$,15,2)
18031 CH$=SEG$(CH$,1,4)&A$&S
EG$(CH$,7,2)&A$&SEG$(CH$,11,
2)&A$&SEG$(CH$,15,2):: CALL
CHAR(CH,CH$):: NEXT CH :: SU
BEND
```

The next one, which works only on ASCII 97-122, makes tall characters ridiculously elongated above.

```
18050 SUB NEWCHAR20 :: FOR C
H=97 TO 122 :: CALL CHARPAT(
CH,CH$):: CALL CHAR(CH,SEG$(
CH$,7,2)&RPT$(SEG$(CH$,9,2),
4)&SEG$(CH$,11,6)):: NEXT CH
 :: SUBEND
```

This one has the characters raised by one line, widened one column at left and two columns at right to make a full 8x8 character which must be double-spaced horizontally and vertically.

```
SS090 SUB NEWCHAR27 :: FOR C
H=48 TO 122 :: CALL CHARPAT(
CH,CH$):: CH$=SEG$(CH$,3,10)
&RPT$(SEG$(CH$,13,2),2)&SEG$
(CH$,15,2):: FOR J=1 TO 15 S
TEP 2
18091 CH2$=CH2$&SEG$("014589
CD",POS("01234567",SEG$(CH$,
J,1),1),1)&SEG$("0129",POS("
048C",SEG$(CH$,J+1,1),1),1)
18092 NEXT J :: CALL CHAR(CH
,CH2$):: CH2$="" :: NEXT CH
:: SUBEND
```

Those who have my Nuts & Bolts disks will see how valuable this assembly can be to make instantly available the routines for double height and double width characters, etc., etc. And if you have Todd Kaplan's amazing ALSAVE routine from the Genial Traveler Vol. 1 No. 3, you can imbed them in your XBasic program for fast loading.

And you can merge CHARSUB into any character editor or sprite defining program and, with a bit of modification, use it to convert your creations into fast-loading assembly.

These assembly loads are compatible with my BXB, so you can also load character sets into sets 15 and 16, ASCII 144-159. However, the CHARPAT statement cannot access ASCII above 143, so in this case you must dimension an array in the program you are copying from, as DIM HX$(159), and place the hex codes in the array using the ASCII as the subscript number, such as

```
CALL CHAR(CH+64,CH$) ::
HX$(CH+64)=CH$,
```

so that they will be passed to the subprogram. And don't CALL INIT after you have called BXB!

So, now you try creating your own screen fonts!

Memory full,

Jim Peterson

MINUTES

Saturday, February 11,1989

Following open library time and an opportunity to read the announcement page, there was the usual question-and-answer period, and then the business meeting was opened by president Dick Beery. One new member, David Truesdale, and two visitors, Henrietta Gaskins and David Adlerstein ,were introduced and welcomed. John Cummings read his final report as treasurer; that and the minutes were approved as read. Items being offered for sale were announced. Posters for display at many different locations were handed out and members were encouraged to help with the placing of these. Irwin Hott and Jean Hall of the nominating committee read the list of candidates. Since there was only one candidate for each office, and no further nominations from the floor, a voice vote was taken, and the slate was unanimously accepted. Much discussion followed regarding our planned participation at the Ohio Exposition Center on Feb. 19. Thanks were tendered Mr. & Mrs. Jon Veit for the fine refreshments. The business meeting was adjourned, and was followed by a program of demonstrations of cassette programs by Jim Seitz; the newly-revised Panorama drawing program by Karl Ronstedt, as well as his House Number learning game that utilizes graphics created by Panorama. The building was cleared by 12:10.

Respectfully submitted,
Dick Beery,
substituting for Jere Singleton, secretary.

DUES ARE DUE IN MARCH.

$28.00	FULL MEMBERSHIP
$20.00	NEWSLETTER FOR ONE YEAR

the ink well
quality printing

SUDHANSHU (PAT) PATEL
Owner

CHERRI PARK SQUARE

15 Cherri Park Square
Westerville, Ohio 43081
(614) 899-1403

TI-Base - [illegible] Inc.
P. O. Box [illegible]
Pt. Orange, FL 32029

IMPORTANT TIPS

NorthCoast 99'ers - Sept. 16, 1988
Late information By Martin A. Smoley

In Tutorial 2, I said that you can use FunnelWeb to enter Command Files and place comments in columns 41 through 80 which would not affect TIB. I must temporarily retract that statement. I am finding that for some reason characters on that side of the page cause TIB to issue the error message "no data base in use". This is not a constant problem, but seems to be affected by certain commands on the left side of the page. So, for now it is best to not type any characters from column 40 to 80 in Command Files. Also, concerning error messages, the message "no data base in use", seems to pop up for a multitude of errors. If you get this message and you don't think that's your problem, check for missing quotation marks, or for improper local variable statements. One reason I use FunnelWeb for CFs is so I can print out hard copies. I have a terrible time finding little mistakes like missing quotation marks on the screen. Here's something you need to remember. If you leave TALK ON you will see all the CF lines scroll up the screen. Keep an eye on the line numbers at the far-left. If you see an asterisk at the beginning of that number, it means that line was not executed. Remembering this could help you find the point where an error originated. Here's an unrelated tip. It seems that 2 is the smallest numeric variable you can create. An example would be "LOCAL I N 2 0". TIB apparently wants space available for a sign, such as "-1", or "-9". I'll admit that I don't know why many of these programming problems occur. I've only been working with TI-Base for two and a half months and I'm doing a lot of (learn as you go) programming. Sorting apparently needs some discussion. I have placed automatic sorts in many of the CFs. This is no problem when you have 5 names in the file, but when you have 100 names it takes a lot of time. With TIB a sort is saved. If you sort a file by zipcode the next time you use that file it will still be sorted by zipcode. This means you only have to resort when you add more names or edit the zipcode field. I'd like to grab this chance to ask for your help. HELP!!! I need questions from TI-Base users. If you have a problem with TIB, or something weird is going on with the program, write down some notes and send them to me, Martin A. Smoley, 6149 Bryson Drive, Mentor, Ohio, 44060. I cannot answer the letters individually, but this information will be used in future tutorials. I realize that the tutorials are wordy and complicated at this point, but when I cover almost everything in the manual the tutorials will switch to more programming and less retoric. You should also read the TIB manual and the TIB supplied tutor a couple of times. You should also create small command files with no specific purpose except to learn how something works. You need to use TIB to learn it. FYI: Here is something to think about. TI-Base is in many ways identical to one of those big database programs for the big machines, but scaled down to fit our small machine. But! you can buy a complete TI99/4A system, with disk drives, and TI-Base for less money than it would cost for the database software alone for an IBM compatable machine.

TI-BASE - From INSCEBOT TUTORIAL 2 By Martin Smoley

NorthCoast 99'ers - Sept. 1, 1988

Copyright 1988 By Martin A. Smoley

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

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 IBasic 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 database we created for Tutorial 1. The problem is in the "ZP" field of the database, "TNAMES". As it was displayed in SCREEN FOUR the ZP 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 ZP" 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	ZP	GP	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	M.	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

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. NT: 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 [illegible] FunnelWeb for a hard copy. I [illegible] sure that all my hardcopies have the program name and pertinent comments at the 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, and 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	0712881
0001	02/29/88	200.11	200.22	0713831
0002	08/27/88	300.11	300.22	0717851
0003	03/03/88	400.11	400.22	0820871
0004	12/30/87	500.11	500.22	0921861
0005	06/06/88	600.11	600.22	0717851
0006	04/22/88	700.11	700.22	0921861
0007	01/21/88	800.11	800.22	0713831
0008	05/12/88	900.11	900.22	0820871
0009	06/17/88	1000.11	1000.22	0713831
0010	03/01/88	1100.11	1100.22	0921861
0011	08/03/88	1200.11	1200.22	0713831

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.

```
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 "'␣E'␣G"
39        PRINT BLNK
40   LOCAL TESTID N 7 0
41   LOCAL TEMP C 60
42     WHILE .NOT. (EOF)
43      REPLACE TEMP WITH "'␣E'␣6" ! 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 go through it together, and I will try to explain the important parts. I hope you have read Tutorial I so I can skim over the routine parts and concentrate on the rest. Remember don't enter the line numbers. Lines 1 through 9 are strictly housekeeping except for CLOSE ALL, and from now on I will consider it 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 runs NUMTST2 as you can see in line 48. You would not want to close all the files in NUMTST2 it would bomb the program. I intend to have my data disk in drive 2, line 9 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 have 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 expects to find on drive 2. After line 8, TIB will expect to find all CFs and Dbs on drive 2, and I will not waste space bringing it up again. There is a three line cluster which is important. The lines are 20, 26, and 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 21 initializes the LOCAL variable named CDATE, which is a D (date type) entry with a length of 8 characters. A 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 ask you to enter the date and 24 will place the cursor on the screen one space after "Within Quotes", and wait for your input. NOTE: with Version 1.02 all Characters, or Dates, which are characters, must be input enclosed in quotation marks, "09/01/88". Line 25 will write the message "Current Date:" and display whatever you type in for CDATE. FYI: I have initialized CDATE close to its use for your benefit. I will continue this procedure in this program, but from then on variables should be initialized at the beginning of a CF. This little chunk (LNs 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 like the rest. You can put a message there and it will stay put until a CLEAR or another WRITE 22,x 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 on it with low partitions between each area. You can take from one to five file folders which are filled with sheets of paper and place one in each of the five areas. You must stand in one spot, but you can SELECT one of the five areas to be positioned directly in front of you. The area 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 the information in the files which are not directly in front of you. If you can grasp this concept and visualize the 5 different slots, or areas, you're going to catch on fast. Remembering, or keeping track of what can be done in non-SELECTed slots is a challange. NOW! line 31 and beyond. Line 31 SELECTs slot 02, 32 opens the Db named TNTST2 in slot 02 and 33 points TIB at the first record in the file. Line 34 SELECTs slot 01, 35 opens the Db named TNAMES in slot 01 and 36 points TIB at the first record in that file. Remember that both of these Dbs were previously SORTed to our specifications. Well, we have done it. At this point we have opened two databases at the same time. TNTST2 is open in slot 02 and TNAMES is open in slot 01, and if we don't count all the junk I put in to add flash to the program, we did it with about 12 lines of code. I told you that we'd get through this somehow. If you examine and keep track of this stuff one peice at a time, you'll get the hang of it sooner than you think.

Continued Next Page.

NOTE: Although it looks like a [illegible] CF on the previous page [illegible] 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 and Doublestrike, and send the codes to the printer. Entering the control codes can only be done with TIM or FunnelWeb at this time. There will be some useless repetition concerning control codes. I'll explain later. Line 40 and 41 initialize two more variables to be used in the WHILE loop. TESTID is to hold a Number with the length of 7 and 0 decimal places, and TEMP has been discussed previously. "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 it 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 is 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 along." 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 we 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 to 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, 44 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 going 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 the control codes, and then it will stick 2 spaces behind that. It will TRIM the trailing blank spaces from FN, attach it to our growing 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. We 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) and TRIMming is only used on (C)haracter strings. Then TIB takes this whole mess we have put together and sticks it into the variable we 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 junk, 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 TEMP. I'd like to also mention that the junk we filled TEMP with was related to TNAMES, (LN,FN,MI,ID). Using this type of data gathering it is up to us to be sure TIB is pointed at the right slot and that slot contains the Db that holds the information we want. In line 47 we put the same ID number from above into the holding area we named TESTID. The phrase 1.ID is another way to tell TIB that we want the ID number from slot #1. At this time the 1 is FYI only, and doesn't have any real effect on the program except to assure me that I am getting ID from slot #1. TIBs Author uses this form of data gathering in the CF named PROCESS, manual page 5-5.

```
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 "%4 "
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 "%5 "
38   PRINT BLNK
39  RETURN
```

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

Continued Next Page.

I can't believe I'm on page 5. Well, since I'm using up so much Newsletter space, here is a promo for my sponsor.

Join The NorthCoast 99'ers UG

NorthCoast has 3500 plus programs in it's library and produces this great little Newsletter. You can take full advantage of the club's services by mail, and you will be certain of receiving my wonderful tutorials in the future. The membership cost for someone living in the continental United States is only $15.00. You can send your membership fee to me, Martin A. Smoley, 6149 Bryson Drive, Mentor, Ohio, 44060. Make all checks payable to NorthCoast 99'ers User Group, DO NOT send cash, and I'll expedite your membership personally.

"OK, NUMTST2, line 22." When we get to this point TIB is looking at the first record in NUMTST2, which we have SELECTed in slot #2. Therefore, in line 22, IF the value in TESTID matches or is equal to the value in ID, then TIB will execute all the lines between the IF (line 22) and the ENDIF (line 29). Remember, TESTID holds the ID number which matches the LN, FN and MI we just printed in from the Db TNAMES. ID holds the ID number from the current record of the DB TNTST2. I will not follow the program accurately because TIB will not find a match to make the IF true until the sixth record of this Db. So lets say it finds a match which makes line 22 true. Line 23 prints the information held in TDATE, NUM1, NUM2, and ID under the persons name from TNAMES. Lines 25 and 26 make up an accumulator that keeps a running total of the NUM1 part of any matching records. Similarly lines 27 and 28 keep a running total of the numbers in NUM2 if the ID match is true. Coming from line 28 to line 29, TIB ignores 29 and goes directly to line 30. This line tells TIB to MOVE its pointer to the next record in the file. So we are now looking at the next record in the Db TNTST2. The ENDWHILE in line 31 is not ignored by TIB, and TIB is sent back to line 21 to test the new ID we now have against TESTID which remains the same. This loop goes around and around. Each time it does, it moves to the next record and then checks for (EOF). If its not the End Of File, and it has data to work on it immediately tests to see IF the ID numbers match, etc. When it runs out of data or hits the (EOF), line 21 sends TIB directly to line 32, the first statement after the ENDWHILE. TIB then puts the dashed line into TEMP and prints it. TIB then prints the current date (CDATE), which you entered at the beginning of that other CF, the totals in TNUM1 and TNUM2, and the word TOTAL. In lines 37 and 38 TIB turns Italics off, at the printer. We then RETURN to that other CF named TNTST2. In doing so we throw away all the LOCALs we initialized in this CF. When we land back in the CF named TNTST2 we land on line 49, which changes the screen colors. Line 50 WRITEs this CFs name to the screen over screen line 17, which was left there by that other CF. Line 51 SELECTs slot #1, so we are once again working with TNAMES. Line 51 MOVEs TIBs pointer to the next record, for a new name, and line 53 sends us back to line 42 to start the whole process over again. These two loops will ratchet through the names in TNAMES one at a time, and for each name in TNAMES, will completely search TNTST2 for any information that is related to that name by comparing ID numbers in TNAMES to ID numbers in TNTST2. It will continue to search until it runs out of names, or records, in TNAMES. At that time 42 will send TIB to line 54. ALL Dbs will be CLOSEd, things that were turned off will be turned back on and the whole thing is finished. In line 58 you are RETURNed to the Dot Prompt. That just about raps this tutorial up except for a few things I said I'd get back to. I threw around a lot of control codes in this set of CFs. If your using FunnelWeb to produce your CFs, you can carry these ideas back to the LABEL program we did last month. Fire up FunnelWeb and retype the CF called LBLS1/C, but this time name it LBLS2/C. There are only about 32 lines and most of them are very short. Leave out the present line that reads LOCAL BLNK C 1. Next, add lines 37, 38 and 39 from TNTST2. Insert them between the line that says TOP and WHILE .NOT.(EOF). This will cause your printer to print in Emphasized and Doublestrike Mode. If you don't like that, try what I did in line 43. You can concatenate (;) control codes on the front and rear of a character string. There are lots of ways to do it. Before my mind goes completely I'm giving up. I copied the printout from this months stuff below. I'd also like to add that this set of CFs make a nice club demo.

```
Vivannovitch Elexxie I.  0712881
03/16/88      100.11      100.22      0712881
-------------------------------------------------
09/11/88      100.11      100.22      TOTAL

Smoley Martin A.  0713831
01/21/88      800.11      800.22      0713831
02/29/88      200.11      200.22      0713831
06/17/88     1000.11     1000.22      0713831
08/03/88     1200.11     1200.22      0713831
-------------------------------------------------
09/11/88     3200.44     3200.88      TOTAL

Aardvark Grant E.  0717851
06/06/88      600.11      600.22      0717851
08/27/88      300.11      300.22      0717851
-------------------------------------------------
09/11/88      900.22      900.44      TOTAL

Jones Quincy W.  0820871
03/03/88      400.11      400.22      0820871
05/12/88      900.11      900.22      0820871
-------------------------------------------------
09/11/88     1300.22     1300.44      TOTAL

Whitman Raymond (Slim) A.  0921861
12/30/87      500.11      500.22      0921861
03/01/88     1100.11     1100.22      0921861
04/22/88      700.11      700.22      0921861
-------------------------------------------------
09/11/88     2300.33     2300.66      TOTAL
```

Continued Next Month.

GETTING THE MOST FROM YOUR CASSETTE SYSTEM

BY MICKEY SCHMITT

NUMBER 15

UNDERSTANDING - CREATING - AND USING - CASSETTE FILES

PART IV

THIS MONTH I AM CONTINUING WITH THE TOPIC OF UNDERSTANDING - CREATING - AND USING - CASSETTE FILES. MORE SPECIFICALLY, I WILL BE CONTINUING WITH THE TOPIC OF... "HOW TO "OPEN" UP A CASSETTE FILE"... WHICH I FIRST BEGAN DISCUSSING TWO MONTHS AGO... IN PART II OF THIS PARTICULAR SERIES.

THE "OPEN MODE" ENTRY INSTRUCTS THE COMPUTER TO PROCESS THE CASSETTE FILE EITHER IN THE "INPUT MODE" WHERE FILES MAY BE READ ONLY... OR THE "OUTPUT MODE" WHERE FILES MAY BE WRITTEN TO ONLY. KEEP IN MIND THAT THE "NEW" "OUTPUT" FILE WHICH IS BEING CREATED WILL HAVE ALL THE CHARACTERISTICS GIVEN IT BY THE "OPEN" STATEMENT SPECIFICATIONS... INCLUDING THE STANDARD DEFAULT CHARACTERISTICS, AS WELL. NOTE... THAT IF YOU ARE USING TWO CASSETTE RECORDERS... ONLY CSI CAN BE SPECIFIED FOR AN "INPUT" FILE AND... BOTH CSI AND CSII CAN BE SPECIFIED FOR AN "OUTPUT" FILE.

AS A WORD OF WARNING: THE "OPEN MODE" SPECIFICATION IS REQUIRED. THERE IS NO DEFAULT SPECIFICATION FOR THIS ENTRY.

THE "FILE ORGANIZATION" ENTRY MUST BE "SEQUENTIAL" FOR A CASSETTE RECORDER. RECORDS ON A "SEQUENTIAL" FILE ARE READ OR WRITTEN... ONE AFTER THE OTHER... IN SEQUENCE... FROM BEGINNING TO END. YOU MAY HOWEVER, "OPTIONALLY" SPECIFY THE INITIAL NUMBER OF RECORDS ON A FILE BY FOLLOWING THE WORD "SEQUENTIAL" WITH A NUMERIC EXPRESSION.

IF YOU OMIT THE "FILE ORGANIZATION" SPECIFICATION... NO PROBLEM... THE T.I. COMPUTER WILL AUTOMATICALLY ASSUME THE DEFAULT SPECIFICATION... AS "SEQUENTIAL" ORGANIZATION.

IF ALL OF THIS SOUNDS WAY TOO CONFUSING FOR YOU... FEAR NOT... I FELT THE SAME WAY MYSELF! WITH THAT PARTICULAR THOUGHT IN MIND... I HAVE DECIDED TO CREATE A "REFERENCE CHART"... IN ORDER TO GET A BETTER UNDERSTANDING OF ALL THE "NEW MATERIAL" THAT I HAVE EXAMINED SO FAR. (PLEASE KEEP IN MIND... THAT THIS PARTICULAR "REFERENCE CHART" IS A CONTINUATION OF THE "REFERENCE CHART" WHICH FIRST APPEARED LAST MONTH... IN PART III OF THIS PARTICULAR SERIES.

TO "OPEN" UP A CASSETTE FILE... FOLLOW THESE STEPS...

10	11	12	13	14
NEED COMMA	OPEN MODE ========== INPUT....READ OUTPUT..WRITE = YOUR CHOICE =	NEED COMMA	FILE..... ...ORGANIZATION ========== SEQUENTIAL SEQUENTIAL	NEED COMMA

NEXT MONTH I WILL CONTINUE WITH THE TOPIC OF UNDERSTANDING - CREATING - AND USING - CASSETTE FILES. MORE SPECIFICALLY, I WILL BE CONCLUDING THE TOPIC OF "HOW TO "OPEN" UP A CASSETTE FILE"... INCLUDING... THE CONCLUSION OF THE ABOVE CHART.

TAX DEDUCTION FILER
By Jim Seitz

(Ed. note: This is a reprint that was published in the Spirit of 99-Jan 1987. Jim will be giving a demo of this program during the March meeting)

One of the most basic reasons to have a computer is to keep yourself organized. Are you organized at tax time each year? Wouldn't be handy to have a file system to keep track of your Form 1040 Schedule A deductions throughtout the year? Maybe Tax Deduction Filer is what you need! This program appeared in Vol 4 Num 4(Sept.84) of Home Computer Magazine as a type-it-in yourself program.

The program, written in XBasic, is easy to use and completely menu driven. The first menu asks you which part of the program you wish to access:

1) ADD DATA
2) CHANGE DATA
3) DISPLAY DATA
4) TOTALS
5) PRINT REPORT
6) LOAD DATA FILE
7) SAVE DATA FILE
8) EXIT PROGRAM

There are 17 deduction categories generally corresponding to the Form 1040 Schedule A format. To activate a category just Press its File Number, you will then be asked to give a description of the tiem. The description cannot exceed 27 characters. I use a short description and check number in case I need to find the actual check later. You are then asked to enter the amount. Before Saving the data I have just entered I usually Display the data or Print the data using one of the 3 printing options. Choosing the TOTALS printout gives you the Total deduction for each category. The ALL RECORDS IN A CATEGORY printout allows you to print out each item entered in the deduction category of your choice. ALL CATEGORIES prints out a list of all the items in the order in which they were entered into the program

I use this program every couple of months and I find it to be not only a help in doing my current taxes but also as a legible explanation of the deductions for use in the future.

$$

TEXAS TAXES
Copyright 1986 Steven Karasek

(Ed. Note: Thanks to New Horizon NL Feb 1989)

These programs will help to save you time and mistakes when preparing your income tax return. They include Forms 1040 and 1040A, Schedules A, B, C, D, E, F, and SE and Forms 2106 and 2441. Each form works like a spreadsheet, so when a data item is entered, all lines which depend on that data are immediately updated. Also, relevant data is copied from one form to another. You can save your data on disk and recall it later to make corrections. If you have a printer, you can print out the completed forms (except for Forms 1040 and 1040A) and mail them directly, so you don't have to copy the information onto the government forms. A working copy of any of the forms can be printed at any time.

There are three programs in the package: 1040, CONSISTENT, and PRINT. They are described individually in later sections of the manual

Not all of the information on the forms is requested by the programs (names, dates, descriptions, etc.), so be sure to fill these in on the government forms. If you use the PRINT program to print out the forms, the information you must fill in is generally marked >__________for names and descriptions, __/__/__ for dates, etc. In the following descriptions of the programs, there are very few references to line numbers on the tax forms, since these numbers are bound to change in future years.

Most of the data that you enter will be either a yes-or-no answer or a numeric value. The yes-or-no questions will look for an answer starting with capital Y for yes or capital N for no, so the ALPHA LOCK button on your keyboard should be depressed. Unless otherwise noted in the manual, if you just press ENTER, it will be interpreted as a no response.

A menu will be presented when you boot the disk. Press 1, C, or P for the appropriate program. Leave the disk in the drive, since the tax tables and other data are on the disk. There is enough room on the disk to store your data files. The other files on the disk are protected so that you won't accidently overwrite them.

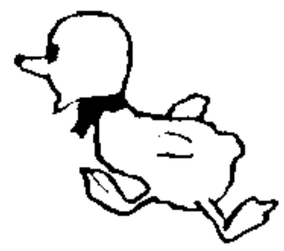
There is room left over on the disk for your data file(s). Since the programs read other data from the disk as well as your data, it is best to have everything on one disk. Don't put tape over the write-protect notch! It is a good idea to make a backup copy of the disk and keep it in a safe place. If you do accidently destroy part of the disk and need a new copy, send $4 to me at the address listed below, and I'll send you a new disk.

To obtain an updated version of the program each year, send a check or money order for $10 plus $2 (to cover the cost of postage and a new disk) to:

Steven Karasek
855 Diversey Dr.
St Louis, MO 63126
(314)961-2052

The new version should be ready about the middle of January each year.

IMPACT/99



BY JACK SUGHRUE
Box 459
East Douglas, MA 01516

IMPACT/99 BLUE RIBBON 1989 WINNER

If this annual award could be given to the same company two years in a row, Asgard Software (with its incredibly varied and impressive catalog) would certainly be very much in contention again. So I'm glad I didn't have to make that decision this year.

Instead, it was a clear choice: MYARC is the winner of the 1989 IMPACT/99 BLUE RIBBON AWARD.

MYARC is one of the few companies still making anything for TI owners on a steady basis. But it isn't just ANYTHING that they are making; they have given us the most powerful hardware and software that exists for us. They haven't just provided enhancements; they have given us a future.

MYARC (the vision, the dream, of former TI employee Lou Phillips) has been around a long time. Since 1982, actually, when Lou developed Winchester hard-disk capabilities which sold better in other countries than here (as we were mostly all fledglings at the time). Later he produced a not-very-successful competitor to the TI PE Box (still flooding the interested market at the time). So he moved into the card development. And there MYARC (which is a mutilated acronymic form of "Microcomputer Architects") began to blossom.

From a personal viewpoint (as this column has always been - for better or worse), MYARC and I have had a perfect relationship. I own lots of their products, and I have never had to speak to or write to anyone about them. They have been easy to use and have never broken down. And they have made my computing life much richer.

A few years ago my TI Disk Controller Card was behaving erratically. Lots of my friends recommended the MYARC card.

Got it.

Loved it from the minute I pulled out my old card and plugged in the new. It immediately made my original Shugart SSSD into a DSSD drive, so I doubled my potential on every disk and no longer had to "flippy" anything.

Not only did the MYARC Controller work smoothly, but it was faster than my old controller, and it had inside a built-in disk catalogUer which could be accessed from anywhere by CALL DIR(n). I foget how wonderful this is until I get to someone else's non-MYARC TI.

And it had MYARC's legendary Disk Management System. Still my first choice among a pile of excellent systems and one that remains constantly configured in FUNNELWEB on my RAM. (But I'm getting a bit ahead of myself.) Lots of programmers learned a lot of techniques from this DM, but for users like myself it opened up a large world (particularly within its futuristic Utility Menu).

Now my drive was old, so I thought I'd get a new DSDD one and add a power supply for my old one. I did. Again, the controller took everything in stride. Switched from one kind of drive to another with no heavy breathing.

As my computer madness grew I knew I'd never be happy without a RAMdisk or some extended memory. MYARC had just come out with their 512 card to go along with their 256 and 128 cards.

As I had such great fortune with MYARC, I bought their 512. Took out my 32K card, plugged in the new. Just like the controller, it worked perfectly from that moment.

I had a LARGE RAMdisk that I could partition as a buffer for my printer and have lots of options available. But did I really need all that space? I didn't think so at the time. I wondered why I hadn't purchased the smaller cards with my hard-earned pennies.

However, within a couple weeks, I had all the FUNNELWEB and PLUS! files I use regularly (and some other very specific utilities and games) all on a RAM load with an automatic 80K set aside for buffing (which turned out to be one of the greatest enhancements I ever added to my TI).

The RAM portion is wonderful to operate. Everything I need is THERE at the moment I want it. All the word processing tools. All the assembly tools. All the utilities, in short, that I always used to load one-by-each as needed. In those days the thing NOT in memory was the thing I needed most at any given time.

And my controller? Well, I just designated my 512 card as Drive 3, and it went about its business as if I was hardly given it an adult task. Its "ho-hum" manner showed me that the design of the thing was ingenious. No fuss. No muss. No bother. I like things that way.

Now, here I was with a MYARC-stuffed full-blown system when my extra SSSD original drive (in the power-supply box) died after much faithful service. Six years is a long time, I've been told. Particularly for the kind of use I give the drives. So I bought a couple DSDD half-heights on sale, put them in the P-Box, put the DSDD from the box into the

ADDED POWER SUPPLY, AND RAN MY SOFTWARE. BUT ALL MY SOFTWARE HAD BEEN GEARED TO MAKING DRIVE 3 AS MY RAMDISK. MY CONTROLLER WINKED AT ME. "CALL THE EXTRA DRIVE DRIVE 4," IT SAID, "AND KEEP THE RAM AT 3." I TOOK ITS ADVICE. NOW I HAVE ALL FOUR DRIVES (WITH 512 AT 3) OPERATING QUICKLY AND FLAWLESSLY AND WONDERED HOW I EVER DID WITH THREE DRIVES OR TWO. CAN'T EVEN IMAGINE HOW I SURVIVED WITH ONE.

(THERE'S SOMETHING VERY OBSESSIVE ABOUT THIS KIND OF BEHAVIOR.)

ALTHOUGH I AM THE ULTIMATE NON-TECHIE, EVEN I CAN PLUG IN CARDS AND (AS A LAST RESORT) READ MANUALS. MYARC MAKES IT SO EASY, YOU DON'T HAVE TO READ THE MANUALS IN MOST CASES, THOUGH THEY WARN THE USER NEVER TO DO ANYTHING WITHOUT FIRST READING THE MANUAL COMPLETELY.

AFTER A FEW YEARS OF BLISS WITH MYARC, I WAS PLEASED AS PUNCH TO LEARN THAT THEY WERE DEVELOPING A NEW COMPUTER THAT WOULD BE COMPATIBLE WITH THE TI. NOT JUST AN UPGRADE. BUT A NEW COMPUTER.

WELL, LIKE ALL (WITHOUT EXCEPTION) NEW PRODUCTS IN THE COMPUTER INDUSTRY WORLD WIDE, THE ANNOUNCEMENTS OF ITS COMING DRAGGED ON AND ON. BUT EACH STAGE WAS PUBLICIZED TO THE POINT OF ANNOYANCE. PROBABLY WHAT WAS MOST ANNOYING WERE THE DOOMSAYERS. THEY DUMPED ALL OVER MYARC FOR THE DELAYS. IT'S TOO BAD, REALLY. THE KINDS OF STUFF COMING OUT FOR STILL-MANUFACTURED COMPUTERS DOES NOT RAISE THE IRE WITH THE ENDLESS DELAYS BECAUSE THERE IS SO MUCH ELSE BEING MANUFACTURED AND RELEASED. WITH MYARC, IT WAS THE ONLY SHOW IN TOWN. SO IT GOT SPOTLIGHTED. AND, IN SOME PEOPLE'S MINDS, GOT A BAD REP. NOT DESERVED. NOT DESERVED AT ALL.

IF YOU'RE THE ONLY COMPANY MAKING A COMPATIBLE UPGRADE FOR AN ORPHANED COMPUTER, YOU ARE TAKING A GREAT RISK TO BEGIN WITH. YOU GET NO SUPPORT TO CONTINUE WITH. AND YOU GET TO LIVE WITH WHAT YOU HAVE CREATED TO END WITH.

WHAT MYARC ENDED WITH IS A MINOR MIRACLE. THE GENEVE (9640) COSTS ABOUT TWICE WHAT THE KEYBOARDS SOLD SEPARATELY COSTS. LESS THAN TWICE WHAT THE DIFFERENT RAMDISKS COSTS. FOR UNDER $500 99ERS CAN NOW BUY A COMPUTER THAT WAS ALMOST 100% COMPATIBLE WITH EVERY PIECE OF SOFTWARE THEY OWN. IT HAS 640K BUILT IN. IT HAS A FULL-SIZE ENHANCED KEYBOARD. CAN PARTITION A HUGE BUFFER FOR THOSE NOVELS OF YOURS. IT HAS THE BEST GRAPHIC RESOLUTION IN THE BUSINESS. IT COMES WITH SOME PRETTY IMPRESSIVE SOFTWARE AND PORTS FOR MOUSE, PRINTER, MODEM, ETC.

THE GENEVE IS THE ONLY ANSWER FOR TI UPGRADING. THANK GOODNESS IT'S A GREAT ANSWER. IN ADDITION TO THE POWERFUL DOS, THE SOFTWARE INCLUDES MYWORD (AN EXCELLENT 80-COLUMN PROCESSOR), ADVANCED BASIC (THAT GOES FAR BEYOND EXTENDED BASIC), PASCAL, GPL, AND A CARTRIDGE DOWNLOADER.

EARLY OWNERS (LIKE MYSELF) HAVE BEEN RECEIVING UPDATES OF ALL THE SOFTWARE FREE. SO OUR MACHINE KEEPS GETTING BETTER AND BETTER. AS A MATTER OF FACT, THERE IS ANOTHER WHOLE PACKAGE BEING SENT OUT BY MYARC THIS MONTH. I CAN'T WAIT. WHAT A SERVICE THIS IS!

THIS COMPUTER HAS SO MUCH SPEED THAT YOU HAVE TO SET MOST SOFTWARE ON SLOWER MODES IN ORDER TO HANDLE THE DIFFERENCE.

AND, LIKE ALL THE OTHER STUFF FROM MYARC, THIS COMPUTER IS ON A CARD THAT JUST PLUGS RIGHT INTO YOUR P-BOX. (THE MANUAL IS HUGE AND INCLUDES QUITE A SECTION ON THE SUPERB ADVANCED BASIC.) IT WILL TAKE QUITE A BIT OF TIME AND EFFORT ON THE USER'S PART TO USE THE GENEVE TO ITS FULL POTENTIAL (IF ONE CAN EVER REACH THE FULL POTENTIAL OF ANY COMPUTER). THERE ARE ALSO MANY OPTIONS (SUCH AS A 512 CARD) THAT CAN BE ADDED TO THE GENEVE. THERE IS ALSO A GROWING SOFTWARE SUPPORT. MYART IS A MOUSE-SERVED, HIGH-RESOLUTION PACKAGE. MOST TI SOFTWARE MAKERS ARE CREATING GENEVE COMPATIBILITY RIGHT AT THE START.

AND, NOW!!! BEFORE I EVEN GET A CHANCE TO START TO MASTER THE GENEVE, MYARC HAS DONE IT AGAIN!

THEY HAVE JUST RELEASED THE FIRST HARD AND FLOPPY DISK CONTROLLER WITH STREAMER TAPE BACKUP SUPPORT WITH MYARC DM-V, THE MOST INTUITIVE DM ON THE MARKET.

THE CONTROLLER INCLUDES A REAL BUILT-IN TIME CLOCK FOR FILE STAMPING; INTERFACES WITH STANDARD FLOPPY, HARD, AND STREAMER DRIVES; SUPPORT OF UP TO FOUR 5 1/4 AND/OR 3 1/2 DRIVES IN ANY CONFIGURATION; PROVIDES RAMDISK SPEED OF A HARD-DRIVE TRANSFER RATE OF 5MBIT PER SECOND. AND SO ON.

I HAVE NO PLANS IN THE IMMEDIATE FUTURE FOR HARD-DRIVING, BUT IT SURE IS NICE TO KNOW THAT MYARC IS PROVIDING THE OPTIONS IF I DO. IT IS ALSO NICE TO KNOW THAT SOME OF THE BEST MINDS IN THE TI WORLD COMMUNITY HAVE PARTICIPATED IN THE CREATION OF THESE GREAT MYARC ADVANCES.

IT IS A REAL PLEASURE TO PRESENT THIS ANNUAL AWARD TO A COMPANY THAT HAS THE TI OWNERS IN MIND AND WHO HAS BROUGHT US INTO THE HI-TECH AGE ENJOYED BY SO MANY OTHER COMPUTERS. THEIR CONTINUED SUPPORT IN THE FACE OF A LOT OF ADVERSITY IS NOT JUST COMMENDABLE BUT ASTOUNDING. MYARC DOESN'T DESERVE THE BUM REP GIVEN TO IT BY THE LOUD (BUT FORTUNATELY SMALL IN NUMBER) COMPLAINERS WHO SEEM TO NEED A SCAPEGOAT FOR THEIR OWN SELF ESTEEM.

CONGRATULATIONS, MYARC! YOU'RE DOING A GREAT JOB, LOU! KEEP IT UP.

HOUSEHOLD BUDGET MANAGEMENT

by Jim Seitz

(Ed Note: This is a reprint. This article appeared in the April 1985 issue of Spirit of 99 followed by SUCCESS, another article in March 1986 issue of Spirit of 99. Both relate to the HOUSEHOLD BUDGET MANAGEMENT program. Jim's address listed in the first article has been changed to his present address.)

I bought my computer during the Great Computer Sale of November 1983. One of the first programs I bought was "Household Budget Management". We put this program on line January 1984 and we are using it this year also. The program consists of 99 preselected categories of which 34 can be active at any one time; thus you can customize your budget to your needs. The categories are classed as either income or expenses. After choosing your categories, you can assign a budgeted amount to the category or enter a full amount. One of my compaints about the program is you cannot rename any of the categories; for instance we wanted to keep track of pet care expenses but there is not a pet care category. We had to use the "Dry Cleaning" category and remember it was really "Pet Care".

As with any budget the biggest problem is keeping track of your records so they can be entered. We had to develop several ideas to improve our record keeping which I would like to share with you. We keep a small box on our desk to put receipts in after we go shopping. It helps to label the items purchased on the receipt to insure they are entered into the proper category. We put paystubs near the box to be entered into the computer also. I do as much entering from the individual receipts as I can; you cannot keep an accurate budget if you only use the monthly statements or your charge cards. After entering the receipts and paystubs I look at the checkbook for other expenses that might have been mixed.

When I finish with the checkbook I draw a line under the last item entered into the computer so I know where I left off. I do the same for the notebooks we keep in the cars to record the car expenses. The last place I look for expenses is on the calendar we keep in the kitchen. We record baby sitting and other cash expenses that generally do not issue a receipt there. It also helps to keep a menu of which category you enter hard-to-define items. For example: are computer expenses "Household" or "Education"? I only enter our budget about 2 times a month and this whole process takes about 30 minutes.

After entering your data you can analyze your budget using several different options. I use monthly and the year-to-date options the most. There are also options to change your budgeted amount or to correct mistakes. You can add or substract categories as you choose, but remember to go back and update your entries! The monthly and year-to-date options also include graphs and projections that can be helpful also.

I have only two chief complaints against this program. The first is-it treats Savings as an expense, I wish the program operated under three main classes: Income, Expenses and Savings. My second complaint is the program does not include the option of a printout. If anyone knows how to get a printout please contact me: Jim Seitz at 2167 Keller Pl W., Grove City, OH 43123 or call (614)875-5532.

I think this is a good program and is worth the investment for the person who does not have access to the more expensive spreadsheet programs.

****

SUCCESS!

by Jim Seitz

In April 85 issue of "The Spirit of 99" I wrote an article entitled HOUSEHOLD BUDGET MANAGEMENT reviewing the module of the same name. In the article I mentioned one large drawback to the program was the lack of a printout. Last October I received a letter from Mr. Bob Lawson of Houston, TX, stating he had written a program, available as "freeware", to print out the HBM files. Would I be interested?....YOU BET! In late November I received my copy of HBMPRINT and used it right away.

You will need the following to use the program: 99/4A console, 32K memory, disk drive(s), RS232 and printer, Editor/Assembler, and your data disk. The program is easy to run; just turn on the hardware, insert the E/A module, put the program disk in the disk drive, select "Load and Run" from the E/A menu and load the program. After the program loads you are walked through a hardware checklist to identify the hardware being used. After identifying the name of your data file disk, place it in the disk drive; press any key; and the file will be read. After answering a few questions and setting up the printer the printout menu will appear. You can choose from the following printouts: 1) All Categories for One Month, 2) All Categories Year to Date, 3) All Categories Total Year, 4) One Category by Month, 5) All Categories by Month, 6) All Income by Month, 7) All Expense by Month. I use the first printout monthly and the rest of the printouts as needed. You can also customize your printout by using this program in conjunction with TI-Writer. I consider the program to be the missing link needed to complete my monthly budgeting. This program is available through the library, and let's support the author of this great program.

These are examples of the category type printout (4,5,6,7)

Categories by Month for: SEITZ'S on: 12/31/85

Category: 46 Telephone				Category: 47 Telephone			
Month	Budget$	Actual$	$Difference	Month	Budget$	Actual$	$Difference
JAN	27	25.82	- 1	JAN	32	77.17	45
FEB	27	24.07	- 3	FEB	32	0.00	-32
MAR	27	17.90	- 9	MAR	32	38.04	6
APR	27	28.60	2	APR	32	31.10	- 1
MAY	27	54.83	28	MAY	32	33.97	2
JUN	27	33.82	7	JUN	32	30.91	- 1
JUL	27	17.21	-10	JUL	32	33.17	1
AUG	27	34.33	7	AUG	32	37.67	6
SEP	27	22.71	- 4	SEP	32	26.14	- 6
OCT	27	17.48	-10	OCT	32	31.42	- 1
NOV	27	32.29	5	NOV	32	32.72	1
DEC	27	24.55	- 2	DEC	32	35.91	4
TOTALS	$324	$333.31	$ 9	TOTALS	$384	$408.22	$24

ANSWERS TO COMPUTER-TERMS#2 by CHUCK GRIMES

COMPUTER_TERMS#2

```
**********************************************************
*   S                   K   T E R M I N A L              *  WORKSPACE       PROM
*   Y             R       E   E R A W T F O S            *  ZERO-BIT        PROGRAMMER
*   N             E         E                    T       *  WRITE           PROCESSOR
*   T         T   M E           P E T I R W   A          *  VIEWDATA        PRINTER
*   H         U   M L   E   P U N E P     P     N        *  VIDEO           POWER
*   E         R   A B   N                   E     A        *  VARIABLE        POKE
*   S         T N R B   I   P A S C A L         M        *  UPC             PIXEL
*   I         L E G I   T   R O S S E C O R P   -        *  TURTLE          PILOT
*   Z         E H O N   U     E   N E W         C        *  TREE            PENUP
*   E       R   T R     O     D                 A        *  TRANSISTOR      PEEK
* W R       E     P     R     I       R         P        *  THEN            PCB
* O         G           B     V     O     K A N          *  TERMINAL        PASCAL
* R P       I       P   U         T L E                  *  TAPE            PAC-MAN
* K R       S     Z   R S         S E G V A R I A B L E  *  SYNTHESIZER     OUTPUT
* S I       T       E   O R   I X R N             M      *  SYNTAX          NIBBLE
* P N M     E   X     R A M S I E 2   R         I P      *  SUBROUTINE      NEW
* A T O     R   A   T M O N P M   3     U     A P C      *  STRING          NAK
* C E D         T     O A -       2       T T S B        *  SOFTWARE        MIPS
* E R E     U   N   S R L   B     - R O M A E     R      *  SAVE            MODEM
*     M K P     Y   T     I   I   S     D     R     E    *  RS-232          MERGE
*       C O     S   R   T   P   T R   W           V W    *  ROM
*           P       I     R         E           A   O    *  RAM
*                   N       E     I           S     P    *  RETURN
*                   G         E V       T U P T U O      *  REGISTER
**********************************************************
```

MEETING DATES FOR 1989

2ND SATURDAY
11 MAR 1989
08 APR 1989
13 MAY 1989
10 JUN 1989
08 JUL 1989
12 AUG 1989
09 SEP 1989
14 OCT 1989
11 NOV 1989
09 DEC 1989
13 JAN 1990
10 FEB 1990

4TH WEDNESDAY
22 MAR 1989
26 APR 1989
24 MAY 1989
28 JUN 1989
26 JUL 1989
23 AUG 1989

C.O.N.N.I. BOARD MEMBERS

Pres. - Dick Beery (614) 459-3597
Vice Pres. - Jim Seitz (614) 875-5532
Treas. - Mike Chaney (614) 272-2281
Sec. - Charles Osment (614) 833-1698
Membership - Everett Wade (614) 262-6346
Librarian - Chuck Grimes (614) 268-8821
Cassette - Sonny Grubb
Cartridges - John Rupert
Newsletter Exchange - Curt Borders
TIBBS - Irwin Hott (614) 263-5319
Dick Beery (614) 459-3597
Editor- Jean Hall (614) 885-4223
Assist- Carol Parkins (614) 891-4965

Spirit of 99

Bulk Rate
U.S. Postage
PAID
COLUMBUS 43221
Permit No. 1945

NEW ADDRESS
C.O.N.N.I.
181 HEISCHMAN AVE
WORTHINGTON, OH 43085

TIME SENSITIVE MATERIAL
POSTMASTER - PLEASE DELIVER PROMPTLY

SMAUG USER'S GROUP/99
RT 4, BOX 23
BREWTON, AL
36426

*** MEMBERSHIP APPLICATION ***

NAME_____AGE_____
ADDRESS_____
CITY_____STATE_____ZIP_____
AREA CODE_____HOME PHONE_____BUSINESS PHONE_____EXT#____
WHAT IS YOUR PROFESSION/VOCATION_____
HOW LONG HAVE YOU OWNED YOUR COMPUTER?_____
DATE OF APPLICATION_____ACCEPTED BY_____