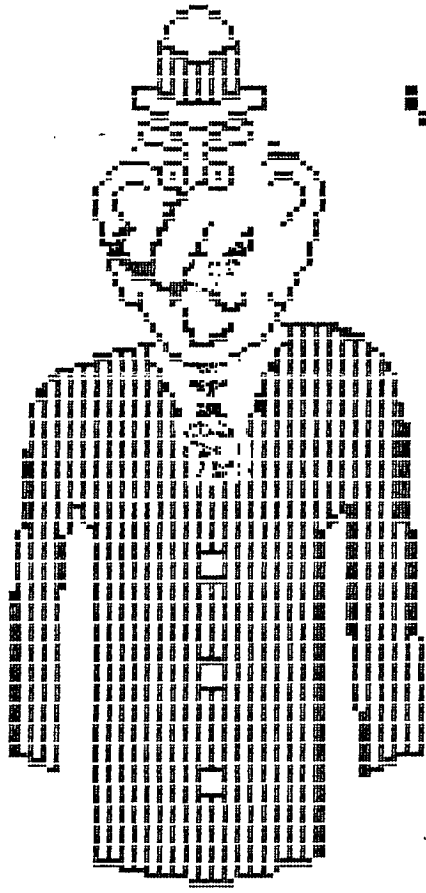


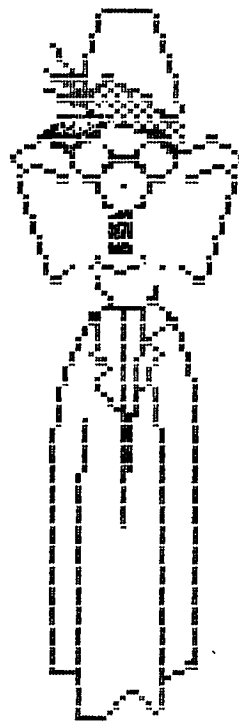
WRITERS FOR:

VICTORIA

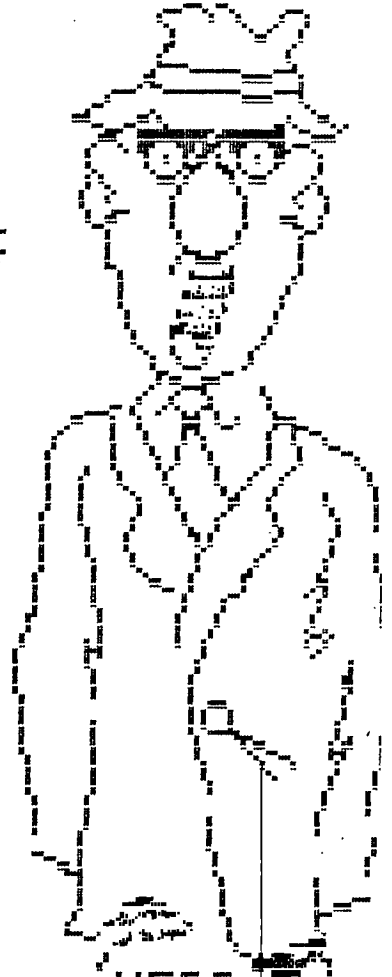
HERB



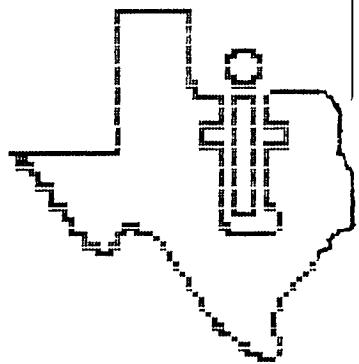
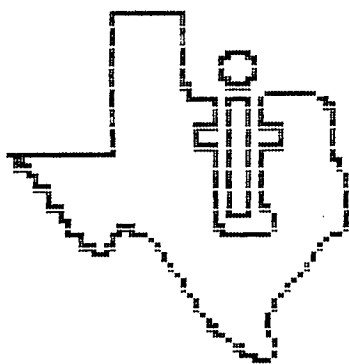
BERT



ERNIE



HERB



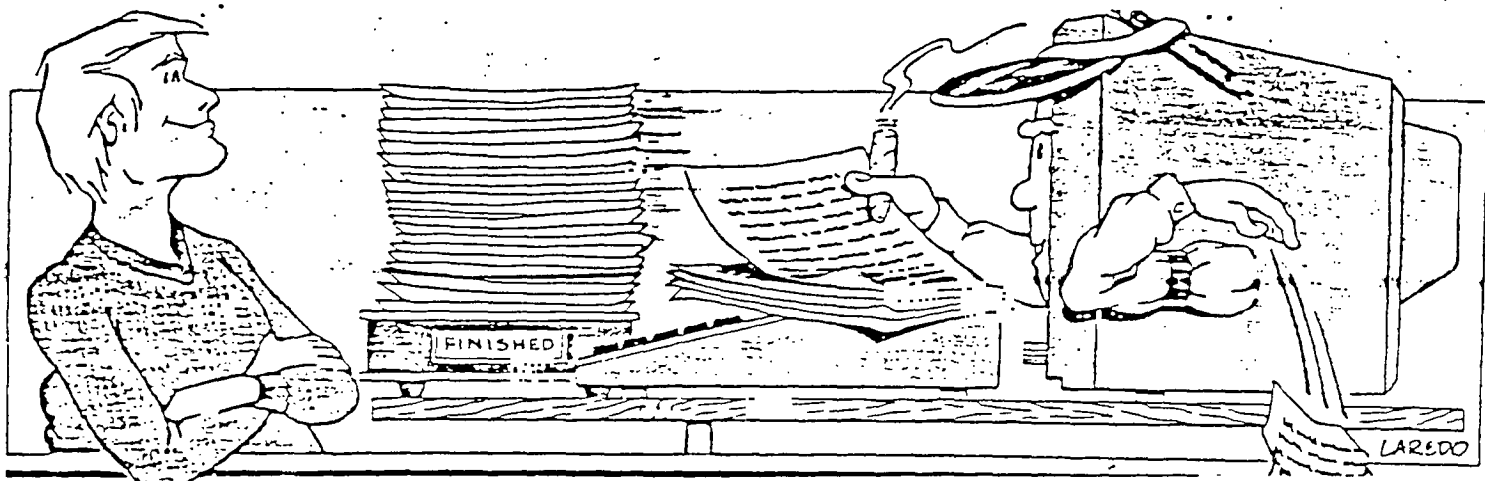
For all you lotto fans out there. Who are still waiting to win that big haul. So that you can afford to run an IBM computer instead of our little TI, for that all the difference between having money and not having as much as you want. I have pieced together a lotto 649 insta-pic that does not cost you one dollar a guess! This program let's you make up to 6 insta-pics of 6 number, and then on your request, will print these tickets to your printer. The program was put together using a screen dump, for a gemini printer by Dane heatherington, and a smashed lotto program of a artist from another users group. I had fun putting this program together especially since I own that old dreaded okidata printer which required a different screen dump. Any way have fun with the program, and pic a winner.

```

100 DIM A(49):: RANDOMIZE
110 CALL CLEAR :: DISPLAY AT(3,10):"LOTTO 649" :: DISPLAY
AT(23,1):"HOW MANY TICKETS?(6-MAX)" :: ACCEPT AT
(23,26)VALIDATE(DIGIT)BEEP:B :: FOR C=1 TO B*4 STEP 4 :: GOSUB 190 ::
GOSUB 240 :: FOR D=1 TO 12 STEP 2
120 DISPLAY AT(6+D,B+C-3):A(D):: NEXT D :: NEXT C :: DISPLAY
AT(24,1):"PRINT THIS CARD? (PRESS Y/N)"
130 CALL KEY(E,F,G):: IF G=0 THE 130
140 IF F=89 OR F=121 THEN 260 ELSE 150
150 DISPLAY AT(24,1):"ANOTHER CARD? (PRESS Y/N)"
160 CALL KEY(Q,F,G):: IF G=0 THEN 160
170 IF F=89 OR F=121 THEN 110 ELSE CALL CLEAR
180 STOP
190 RESTORE 200 :: FOR D=1 TO 49 :: READ A(D):: NEXT D
200 DATA 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19
210 DATA 20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35
220 DATA 36,37,38,39,40,41,42,43,44,45,46,47,48,49
230 RETURN
240 FOR H=1 TO 49 :: I=INT(RND*49)+1 :: J=A(H):: A(H)=A(I):: A(I)=J ::
NEXT H :: RETURN
250 END
260 OPEN #1:"PIO"
270 FOR ROW=1 TO 17
280 FOR COLUMN=1 TO 32
290 CAL GCHAR(ROW,COLUMN,X)
300 PRINT #1:CHR$(X);
310 NEXT COLUMN
320 PRINT #1:" place 32 spaces between quote "
330 NEXT ROW
340 CLOSE #1
350 GOTO 150

```

# EDITOR BIT



Congratulations to the LA 99ers users group for what appeared to be a successful FEST WEST 87. LA TOPICS, BREA USERS, SAN BERNARDINO, and TI CLUB OF OXNARD were represented by individual stalls with information, displays and some hardware for sale.

LA had a large demo table with 3 systems on the go. There library of disks was for sale at a nominal value of \$2 a disk. BREA had an item that caught my eye and probably the eyes of most of the hacker's that attended. The piggy backing of 2 widgets into the single grom port, that allowed for 6 module to be plug in at one time. Unfortunately there was no widgets for sale.

Myarc was again present with it's Geneve 9640. The rep said that software was coming, and I did see some impressive graphic using the Geneve and the old TI monitor. At the completion of this article look for some specs on the Geneve, it sounds promising. TRITON company had the Turbo XT on display with a flight simulator showing excellent graphic on the old TI monitor. But not ever playing with a flight simulator I could not make any kind of comparison.

There was other companies and club represented at the fest, such as: THE RAVE 99 KEYBOARD (gives the TI a full IBM keyboard with seperate special functions keys and numeric keypad), FAIRWARE, and BYTE MASTER.

The fest did not only cater to the needs of TI but had the whole bottom floor of the hall dedicated to Apple, Atari, Amiga, Sperry and a whole host of IBM compatibles. They had circuit boards, power supplies and modem for sale at real money saving prices. A good example was a 150 watt power supply for \$40.

I spent the better part of the afternoon in the fest, bought a couple of programs, got a couple of hints on running my oki-data printer, and was impressed to see that there is still are a lot of people who are running our little orphan. Again well done LA.

## SPECS Between MYARC 9640, ATARI ST, and AMIGA 1000 Computers

	MYARC 9640	ATARI ST	AMIGA 1000
RAM (standard)	640k	512k	256k
EXPANDABLE TO:	2 MB	4 MB	8 MB
MICROPROCESSOR:	TMS9995 12MHZ	68000 8MHZ	68000 7.1MHZ
	8/16 BIT	16/32 BIT	16/32 BIT
INTERFACES:	RS232, PARALLEL	RS232, PARALLEL	RD232, PARALLEL
KEYBOARD:	95 KEYS IBM TYPE	85 KEYS	94 KEYS

SOUND:	3 VOICE, 1 NOISE	3 VOICE	4 VOICE STEREO
REAL TIME CLOCK:	YES	NO	NO
JOYSTICK PORT:	YES	YES	YES
MOUSE:	YES (optional)	YES	YES
VIDEO PORTS:	RGB, COMPOSITE	RGB	RGB, COMPOSITE
DISK DRIVES:	5.25 DS/DD 360K	3.5 SS/DD 360K	3.5 DS/DD 880K
# OF DRIVES (max)	4	2	3
EXPANSION BUS:	YES (PE-BOX)	NO	YES
GRAPHICS: low-res	256X212	320X200	320X200
hi-res	512X424	640X400	640X400
80 COLUMNS:	YES	YES	YES
COLOR PALETTE:	512	512	4096
ON SCREEN (max)	256 COLORS	16 COLORS	32 COLORS
HARDWARE SPRITES:	32, 16 COLORS	NONE	8, 4 COLORS
SPEECH:	YES	NO	YES
HARD DRIVE PORT:	YES (optional)	YES	YES (optional)
SOFTWARE INCLUDED:	MYARC M-DOS 3.0 ADVANCED BASIC TI-WRITER MULTIPLAN CART TO DISK PROG PECAN USCD PASCAL	GEM DESKTOP ATARI BASIC ATARI WORD ATARI LOGO	AMIGADOS 1.1 AMIGA BASIC GRAPHICS DEMO AMIGA TUTORIAL
RETAIL PRICE:	\$499.00 (us)	\$995.00 (us)	\$1495.00 (us)

For the first time since I have been editor, I will submit a second editorial for your reading enjoyment. Recently our club has been talking about the disk magazine Genial Traveler by Barry Travers. I have purchased a subscription for last years editions (1 to 5), for \$30, and this years subscription for the meager price of \$30 US. Last years subscriptions gave 5 news letters, plus 2 bonus disks (PR BASE and FUNNELWEB). That bring the total cost of the magazine down to about \$4 an issue. A very realistic price for a disk, with excellent information and programs that are ready to run. No typing or debugging required.

With this last thought in mind I would like to quote, with permission given by Mr. Travers, just a couple of files off the 5th disk. These 2 files contain the docs for the XXB program that is loaded from disk.

By reading these files, I am sure there are some of us that are interested in using these new commands. This program, and many others can be acquired simply by subscribing to GENIAL TRAVELER address:  
835 GREEN VALLEY DRIVE  
PHILADELPHIA, PA. 19128

The US Mail and Canada Post seem to reasonably fast and there is no custom duty to be paid. With this info in mind, I would like to say let's support this magazine (that is entirely written for TI) so the people will continue to write software for our little orphan annie.

Views expressed are that of the editor. If you have a comment on one of your special disk, send it to me and we will share it with all our readers.

TEXT TO FOLLOW:

XXB/DOCS1

What is XXB? Well, just as XB is Extended BASIC, so XXB is Extended Extended BASIC. And just as XB gives you 40 commands not found in TI BASIC, so XXB (Version 1.0) provides you with 25 commands not normally available in Extended BASIC.

Those commands, in alphabetical order, are as follows: ACCEPT, BYE, CHRSET, CLRTXT, COLOR, DSPLY, INIT, KEYR, MODE28, MODE40, NEW, NEWCLR, NEWSET, NOQUIT, OKQUIT, PEEKC, PEEKV, POKEC, POKEV, READ, RELOAD, RETURN, RUN, TXTCOL, and WRITE.

Some of these new commands are very powerful -- e.g., ACCEPT, DSPLY, and INIT, generously contributed by J. Peter Hoddie -- while others are of a more limited utility, but I think you will agree that the total offers up some tremendous facilities, especially for the creative Extended BASIC programmer.

Since XXB is disk-based, the good news is that XXB itself can be Extended. (The LOAD program on this disk is just one example of such a possible extension, since it includes Peter's useful ARRAY program along with the other 25 commands.) In fact, you can make your own XXB, using the approach set forth in my article on "Extended BASIC Assembly Language 1-2-3" in the January 1986 issue of Super 99 Monthly.

We will attempt to make our later versions of XXB, however, compatible with earlier ones, so that you won't have problems running the programs in GENIAL TRAVELER. Thus XXB (Version 1.0) includes a form of "RAW" (READ and WRITE) that is compatible with an earlier form of "RAW" that I made available. There's lots of room left, so we should be able to add some new commands to future versions of XXB

without affecting present commands.

The 25 new commands can be placed in the following categories: (1) Disk Access, (2) Character Sets, (3) Text Mode, (4) Graphics Mode, (5) Peeks and Pokes, and (6) Miscellaneous (wouldn't you know!).

Let's take them up, category by category.

#### (1) DISK ACCESS

READ and WRITE ("RAW" stands for Read And Write) allow you to read or write a single sector on a disk. The format is as follows:

```
CALL LINK("READ",DRIVE,SECTOR,A$,B$),
```

 where DRIVE is the drive number, SECTOR is the sector number (remember: the first sector is sector zero!), and A\$ and B\$ are two 128-byte strings that contain the information from the sector.

The reason we need two strings is that one string just isn't quite big enough: a sector contains 256 bytes of information, and the maximum length of a string is 255 bytes!

```
CALL LINK("WRITE",DRIVE,SECTOR,A$,B$)
```

 is similar, where DRIVE is drive number, SECTOR is sector number, and A\$ and B\$ are the two 128-byte strings of information to be written on the disk.

Many have put my "RAW" routines to good use, including John Johnson (with his PRBUTL, a very nice PRBASE utility) and Rich Mitchell (who knows as much about Microsoft Multiplan (tm) as anyone I know!). My ARCHIVER program on this disk uses READ and WRITE routines.

#### (2) CHARACTER SETS

Peter's CALL LINK("DSPLY", R, C, M\$) is very similar in operation to XB's DISPLAY AT, but DISPLAY AT works in a meaningful way only in normal graphics mode, while Peter's CALL LINK("DSPLY", R, C, M\$) can be used in either graphics mode or text mode, provided that CALL LINK("INIT", L, R, M) has been properly set.

His CALL LINK("ACCEPT", R, C, L, V\$, M\$) is an amazing facility, because it offers something not even available in Myarc's Extended BASIC IV (or whatever they call it): a multi-line ACCEPT AT! And you can set the Length and the Validation string (features we're used to from Extended BASIC), as well as suggest a default Message.

See the file called ACCEPT/S on this disk for more comments on ACCEPT, DSPLY, and INIT, which work not only in text mode, but also in graphics mode.

Many CALLs accessed while in text mode create a screen "glitch," so CALL LINK("KEYR", R\$) is provided to avoid that problem. (One irritation that comes with working with text mode from Extended BASIC is that apparently all CALLs to user-defined subprograms will cause such glitches.)

CALL LINK("CLRTXT") will clear the screen in text mode. It operates in a similar way to CALL CLEAR in graphics mode.

Finally, CALL LINK("MODE28") takes you safely back to graphics mode in Extended BASIC.

#### (4) GRAPHICS MODE

--- -----

Peter Hoddie's CALL LINK("INIT", LEFTMARG, RIGHTMARG, MODE) also has its uses in graphics mode, where the mode should normally be set to 32 (although 31 and 33 are also useful - see the

CALL LINK("NEWSET") gives you new character set, with large capital letters and true lower case.

CALL LINK("CHRSET") -- when used along with CALL CHARSET -- restore the old character set. (CALL CHARSET in Extended BASIC does not restore the lower case letters nor clear character sets 13 and 14, so CALL LINK("CHRSET") makes that simple to do. This can be important if you do any program chaining, which does not automatically restore character definitions!)

#### (3) TEXT MODE

--- -----

Text (40-column) mode is not normally available in Extended BASIC, unless you have, for example, Oak Tree Systems' "DEP" (Display Enhancement Package), a useful utility which is a sort of XXB itself. Our XXB gives you some features, however, which are not available in the DEP, especially thanks to J. Peter Hoddie, who gave us some really nice routines: ACCEPT, DSPLY, and INIT.

CALL LINK("MODE40") puts you into text mode. Warning: text mode is an area in which the Extended BASIC programmer must walk with care. Two things to remember: (1) You must use CALL LINK("MODE28") in order to return safely to graphics mode, and (2) any kind of error in text mode can throw you into never-never land!

Peter's CALL LINK("INIT", LEFTMARG, RIGHTMARG, MODE) lets you set your left and right margins as well as the mode. For text mode, MODE must be 40, but some creative use of LEFTMARG and RIGHTMARG can allow some interesting possibilities for ACCEPT and DSPLY.

CALL LINK("TXTCOL", F, B) sets the foreground and background colors for text mode.

-----  
One disadvantage of accessing a/l routines from XB is the necessity of repeatedly typing in CALL LINK. Well, if one is prepared to sacrifice some of the speed of XXB, that can be avoided by MERGEing in XXBSUPPORT with your XB program.

The result is that no longer will you have to do a CALL LINK("BYE"): a CALL BYE will be sufficient. Likewise a CALL LINK("DSPLY",R,C,M\$) can now be done with a simple CALL DSPLY(R,C,M\$). And so on.

Be sure to read the REMs in the XBBSUPPORT file. One fact that is noted is that this is not a useful approach to use in text (40-column) mode (because of the screen "glitches" caused, as noted elsewhere), but it is passable elsewhere (again, with some sacrifice in speed).

#### XXBSUBPROG

-----

The XXBSUBPROG file provides some additional support of a different kind. The new commands in XXB can be combined with user-defined subprograms in Extended BASIC, to allow yet more new facilities.

One example is the ability to display a string either vertically or diagonally with ease and speed, though this is ordinarily somewhat laborious in Extended BASIC. (See STRINGDEMO on the disk.) Another is the ability to put up a rectangular graphic display

quickly at any location. (BOX/DEMO is an example of this.)

DON'T MISS two subprograms that make good use of the PEEKC and POKEC routines. SUB SAVE\_SCREEN(A\$()) and SUB PUT\_SCREEN(A\$()) allow you to save an entire screen into four 192-byte strings. Those strings can be put on the screen again with ease. No demo is provided, so it's up to you to try out these XXBSUBPROG on your own.

Once you've saved the screen into four strings, you do do anything you want with them, e.g., manipulate them or store them on disk. The strings, however, contain no information on character redefinitions (or sprites, for that matter).

XXBSUBPROG is just the start of a library of useful subprograms which put to work the 25 new commands in the present version of XXB. With your help, we can expand this library (as well as add new a/l subroutines to XXB)!

I hope you enjoy XXB as much as I enjoyed putting it together for you!

STRINGDEMO program on this disk).

His CALL LINK("DSPLY",R,C,M\$) and CALL LINK("ACCEPT",R,C,L,V\$,M\$) grant us some special uses also in graphics mode (see the BOX/DEMO program for one example).

My CALL LINK("KEYR",R\$) is a good replacement for the following common code:

```
100 CALL KEY(0,K,S) :: IF S=0 THEN 100
ELSE R$=CHR$(K)
```

Thus it has some use in graphics mode as well as in text mode, for which it was originally written.

(continued in XXB/DOCS2)

XXB/DOCS2

#### (5) PEEKS AND POKES

-----

My PEEKs and POKEs are different from usual in that they deal entirely with strings rather than values, but this is much more memory-efficient for Extended BASIC.

CALL LINK("POKEC",L,S\$) and CALL LINK("POKEV",L,S\$) poke strings into CPU RAM and VDP RAM respectively.

Likewise, CALL LINK("PEEKC",L,S,S\$) and CALL LINK("PEEKV",L,S,S\$) are able to peek into a certain Location for a String of designated Size, again for either CPU RAM or VDP RAM.

So far as VDP RAM is concerned, there is no compensation for the >60 (96 decimal) offset for the screen, but that should not be a handicap, since we have other utilities to deal with the screen (e.g., ACCEPT, DSPLY, etc.).

#### (6) MISCELLANEOUS

-----

CALL LINK("NOQUIT") disables the "Quit" key, while CALL LINK("OKQUIT") re-enables the "Quit" key again.

CALL LINK("NEW") prepares the way for a New program, without clearing the screen, while CALL LINK("NEWCLR") prepares the way for a New program and clears the screen.

The next three CALL LINKs came as a result of information provided by Craig Miller in the excellent manual that accompanies Night Mission.

A CALL LINK("RUN") re-runs the program, but without the pre-scan that normally takes place when that is done. CALL LINK("RELOAD") really is equivalent to RUN "DSK1.LOAD". And CALL LINK("BYE") closes all files and returns one to the title screen.

CALL LINK("RETURN") can be used as a test to see whether XXB has been loaded into memory.