Mass Users of the Ninety-nine and Computer Hobbyists

MARCH 1995 Monthly Newsletter Version 14.03
WE'RE STILL ALIVE IN 95

RETURN TO
RA BISHOP

M.U.N.C.H.
C/O J. W. COX
905 EGDEBROOK DRIVE
BOYLSTON, MASS. 01505

NEXT MEETING: TUESDAY, MARCH 14th.

POSTMASTER: Forwarding and Address Correction Requested.

FIRST CLASS!!

R.A. BISHOP
16 FRENCH AVE
NORTHCOTE 3070.
VICTORIA AUSTRALIA

NEXT MEETING TUESDAY, Mar. 14, 1995 7:00 PM. Happy Spring!

MUNCH OFFICERS AND NUMBERS (all in 508 area unless noted)

PRESIDENT	W. C. Wyman	865-1213		
VICE-PRESIDENT	Open		MUNCH DUES:	
TREAS./EDITOR/CLK.	Jim Cox	869-2704	New Membership	$25.00
DEMO LEADER	Jack Sughrue	476-7630	Renewal	$15.00
Asst. Demo Leader	Lou Holmes	617-965-3584	Newsletter Sub.	$13.00
LIBRARIAN	Walt Nowak	413-436-7675		
Advanced Programmer	Dan Rogers	248-5502		

****

FEBRUARY MEETING. The February meeting had seven members in attendance. We demoed the DOM and tried our hand at some games. Corson brought up the idea of having a Fair at out meeting place. Everyone there was quite excited by the idea and we decided to see if the hall was available in late April. I contacted the Community House people and unfortunately the hall is used on weekend days for three different scout troops, evenings were the only available time and this does not seem workable. We should keep the idea alive and try for something in the Fall, then we would have more time to plan.

MARCH MEETING. It's pot luck for this month. We can look at the DOM and whatever else comes up.

RAFFLE. Every month we have a raffle to help defer the rental cost of our meeting hall. A typical raffle will have programs, blank disks, books, bumper stickers and all sorts of odds and ends of interest to the T.I. user. This month we have some Tandy Model 4 computers.

REPRINTS. Reprints are permitted as long as credit is given to M.U.N.C.H.

ARTICLES. I am always looking for articles for this newsletter, anything which interest you will probably interest other members of the T.I. community, so please share your ideas and opinions with all of us.

DISK LIBRARY. The disk library is at all meetings. We have copies of all disks in the library and they are available to members for just $1.00 for each disk unless otherwise specified. You can order them through the mail, please add $1.00 for the first disk and $.40 for each additional disk ordered to cover postage and handling.

DISK OF THE MONTH. This month's DOM #142 but I haven't had a chance to put it together yet.

ADVENTURE II. This is our fund-raiser for 1994/95. The cost to members is $4.00 add $2.00 for first class postage. The regular price is $6.95 plus postage. This is a two DSSD disk set, archived. There is also a special on The Adventure Compendium and Adventure II for members it is $8.00 plus $3.00 for first class postage.

FOR SALE. Former member Dennis Lavoie has two complete systems and lots of cartridges and software for sale. He wants to sell everything together. If you are interested contact Jim Cox at the number above and he will give you more information.

ALSAVE, by Tod Kaplan, ranks with FUNLWEB, DM1000, DSKU, BOOT and REDISKIT among the greatest tools ever given to the TI community - and from a programmer's viewpoint it ranks above them all. I should give equal credit to Barry Boone's SYSTEX, which does much the same thing.

Barry Traver has written, and collected, a diskfull of these routines which he has distributed under the name of XXB ((Extended Extended Basic). Karl Romstedt, in connection with his assembly classes, wrote a double diskfull of great routines which were also recently distributed with Barry's Genial Traveler. My 127 Screen Fonts offers 127 different screen character sets. In Tips #57 I published a neat routine by Gikow to detect the position of the Alpha key, and in Tips #59 I have a great boot tracking routine by Adrian Robinson. Bud Wright wrote a routine for Irwin Hott, to convert lower case to capitals so it could be accessed by speech. Someone released some code for quick flip-on "help" screens. I am sure there are others that I have not heard of - and I would like very much to hear of them!

When these assembly routines are merged in with ALSAVE, they effectively add new CALLs to the TI XBasic language - in fact, they can be accessed by a CALL to a subprogram which contains the CALL LINK. Even TI's built-in CALLs can be modified, as John Behnke did with his remarkably ingenious VDPUTILII (which I scrunched into BXB and Karl Romstedt speeded up into his assembly version BXBAL).

We need lots more of these little specific assembly routines. I will never be able to learn assembly, but I seem to be pretty good at putting ideas into the heads of those who can, so -

I wish there was a CALL LINK("COUNTN",N()) to sort numbers and a CALL LINK("COUNTS",M$(),FIELD) to sort strings on a specified field. I know that John Clulow's assembly sort, Peter Hoddie's Sort Experiment, and TI-Sort do the job, but they are rather large programs. A simple XBasic sort can be written in half-a-dozen lines, surely a simple assembly sort is possible? Come to think of it, I like to cram my records together separated by ASCII 0 to 31 rather than on fixed fields, so I would also like a CALL LINK("SORTA",M$(),A) to sort on the first occurrence of ASCII A!

I wish there was a CALL LINK("COLOR",SET,FORE,BACK) that was faster than the XBasic CALL COLOR, so I could speed up my music that is accompanied by changing kaleidoscopic colors. Maybe that's not possible?

I wish there was a CALL LINK("SOUND",ATTACK,DECAY,DURATION,VOLUME,NOTE1,NOTE2,NOTE3,NOISE) so that I could program attack and decay in music faster than I can through a loop in the volume control.

POS in Extended Basic is very useful to find the first occurrence of a specified character or string, but they failed to give us a CALL LINK("LASTPOS",M$,P$) to find the LAST position.

ACCEPT AT with a negative SIZE to accept a default of whatever is on the screen is very useful, but if you elect to key in something which is shorter than the default, you must erase the extra characters. This results in a lot of burps and honks out of the computer, especially when the previous input becomes the default. How about a CALL LINK("ACCEPT....? with SIZE and VALIDATION if possible, which would accept the default if the first keypress was ASCII 13 but would otherwise ignore the default? I have written such a routine in XBasic but it depends on CALL KEY and therefore results in errors if you type too fast.

Well, those are just a few ideas. If anyone responds to those, I am sure I can come up with more!

UNLIMITED BASIC

by Jim Peterson

Those who have learned something about programming in assembly, Forth, C, Pascal, Fortran, or what have you, like to put down Basic as being a primitive language fit only for beginners and children. The truth of it is, those languages are so hard to learn that few of their proponents become truly proficient in them, and their programmers have contributed relatively few programs to us. If Basic did not exist, the TI world would not have 5000 programs available - or even 500!

Basic is relatively easy to learn, easy to write, easy to modify, and has all the speed that is necessary for most purposes. After years of programming, I am still surprised that I can usually find some combination of Basic commands to acomplish almost anything I want to do.

TI Basic also has one invaluable feature which, I understand, is not available even in IBM Basic - it allows the use of subprograms with local variables which are distinct from variables of the same name in the main program. The great advantage of this is that libraries of subprograms can be assembled, and merged into programs as needed with no worry about conflicting values assigned to variable names. I took advantage of this feature to compile my three disks full of 348 Nuts & Bolts routines. These effectively add 348 more CALLs available to TI Basic - but, I must admit, some of them do execute much too slowly!

Assembly language is difficult to learn, and difficult to write, but its capabilities are almost unlimited, and it takes up little space in memory. And the wonderful thing is - assembly language routines can be called upon, from Extended Basic, to do whatever Basic cannot do or cannot do fast enough!

It is possible to create an entire programming environment in asembly, to be called on from XBasic as needed. SXB was the first of these, but I have had no experience with it. Riccio's STAR was nothing but a random collection of assembly routines which had already been published separately and were best kept separated. Curtis Alan Provance's EDP and Craig Sheehan's XDP are superbly written programming environments oriented toward graphics, and Norm Stillwell's Music Preprocessor is a similar tool for programming music. Unfortunately, no programmers - myself included, I'm sorry to say - seem to have taken advantage of them.

Triton had the opportunity to greatly increase the power of TI Extended Basic when they introduced their Super Extended Basic module. Unfortunately, most of their added CALLs are nothing that cannot be emulated in ordinary XBasic (as Art Byers has done) and some are worthless fluff. I asked Mike Dodd why they did not give us a 40-column screen and he replied, "We never thought of it - and there was room, too!"

But, if Triton had included something worthwhile, programs written using those commands could not be run by those using the original Extended Basic module - which is why I have never used any of them.

What we really need are small assembly routines which can be loaded into XBasic when needed and then LINKed to perform whatever needs to be done. These routines can be placed on the same disk as your XBasic program, and loaded into it by a CALL LOAD. However, far better, they can be merged invisibly into the XBasic program by using ALSAVE, for much faster loading, and can then be copied together with the main program.

Funnelweb's DISKREVIEW
A review by John Bulakowski
- Nutmeg TI-99er's

Version 4.2 of Funnelweb has a useful utility program called DISKREVIEW. It is a combination disk directory, file review, and program loader. With this program you can:

* Call up/print a directory of a disk in any drive
* Protect/unprotect files
* Delete/rename/view files
* Load and run any E/A or Extended Basic language programs

The last feature is particularly nifty because you don't have to know anything about the program structure that you're trying to run. All you do is place the cursor next to the program that has been listed by the directory feature, press R (for Run), press FCTN 6 (proc'd), and then a number (usually 1 to 3 in the case of assembly programs) of what the DISKREVIEW program suggests as appropriate. That's all. The selected program will then load and run. No more guesses as to trying E/A #1, 2, or 3. No more wondering what an "object" file is, or it's name. I have gotten into the habit of running most of my programs through DISKREVIEW. It's also quick to load and run, which makes it a real competitor to other XB loaders. This, by the way, leads me to the only problem that I have found to date using this.

It appears that this program 'seeds' the randomize statement in XB programs with the same number each time the XB program is loaded and run. To give a practical example of what this means to the user, let's run an XB program that would generate a random sequence of five, one digit numbers. The following is such a program:

```
1 OPEN #1:"PIO" :: RANDOMIZE :: FOR I=1 TO 5 :: NUMB=INT(RND*9) ::
PRINT #1:NUMB, :: NEXT I :: CLOSE #1::END
```

If this program was loaded and run through DISKREVIEW it produces the following sequence on my computer: 3,5,1,8,4. If it were loaded and run again through DISKREVIEW, the same exact sequence of numbers would be generated in lieu of the desired effect, namely a different set of numbers. By following this through, any XB program that utilizes a random number generator will always start the same each time it is loaded and run through DISKREVIEW. In the case, say, of a card game, this dooms one into playing the same hands every time. There are a couple of ways out of this problem. One is to alter the DISKREVIEW program. I'm not smart enough to do that. The other is to clear (FCTN 4) the XB program from running once it has been loaded through DISKREVIEW and then type in RUN (and press ENTER). This will clear the 'seed' number out and permit the program to run as advertised. For those of you that may have been using DISKREVIEW to load your XB programs and finding that perhaps the resulting game (or whatever) is always producing the same results, it may not be the fault of the XB program. Try the above suggestion and see what happens.

Notwithstanding the above, DISKREVIEW is a fine program. I recommend that if you haven't been using it, please do. You will be pleased with the results.

HANDY TIPS FOR THE TI99/4A COMPUTER

Here are a few tips for beginners (good for experienced programmers,too) from old newsletters.

**#1 If you have the speech synthesizer and the TEII cartridge here is a trick for debugging programs. All you have to do is enter your program, type LIST "SPEECH" and hit enter. The computer will read your listing back to you.

**2 If you want to disable the quit key (fctn +/=) type in CALL INIT :: CALL LOAD(-31806,16) and then enter. You must have Extended Basic.

**3 If you are going to save a program to tape and type OLD CS1 instead of SAVE CS1 don't panic. Press FCTN and E together then press (ENTER). This will take you out of the tape loop.

**4 You don't have to enter line number in TI BASIC or EXTENDED BASIC. Before you start enter NUM n(1),n(2); where n(1) is the starting line number and n(2) is the desired increment.

**5 In TI Basic you can edit a line with the edit command or with the FCTN key and either the E or X keys. To use edit, type EDIT n (n=line number). The other way is to enter the line number and press FCTN X or FCTN E. This is the only edit method recognized by Extended Basic.

**6 You can list programs to the screen in several ways. Try these; LIST, LIST n, LIST n-, LIST n-n.

**7 If you want or need to renumber the lines in a program either to make it neater or make room for new lines you don't have to renumber them individually. Just enter thecommand RESn.n for resequence (starting number, interval between lines).

**8 When entering a listing in Extended Basic and several lines are very similar, you can save time by typing in the first line and hitting (enter). Then press FCTN 8 (redo). Change the the line number and make the changes to the line as needed and hit (ENTER).

**9 Have you ever pressed ERASE by mistake and lost the whole line? Don't panic and DON'T hit (enter). Instead press FTNC ? and (enter). Your line will still be intact.

*10 In Extended Basic type in RUN CS1. Follow the instructions on the screen. It will load the program and then run it automatically.

*11 In Extended Basic you can use Rem or ! to put documentation in a program that the program will ignore.

*12 When you want to stop a listing on the screen in Extended Basic, just hit any key. To start the list again, strike any key.

*13 You can add comments after a GOSUB or GOTO. They won't interfere with the program and you don't need REM or !.

*14 With Extended basic and a disk system, save a program under the name LOAD. When you start with this disk in the drive #1, it will load and run that program.

*15 If you have the TEII cartridge and the Speech Synthesizer type in the program on page 37 of the TEII manual. Try entering strings of K's, Q's, U's, W's, J's, jor x's for different sound effects. Try mixing them for interesting sounds.

*16 If you have Extended Basic and 32K type this in as the last line of your program: CALL INIT :: CALL PEEK(2,A,B):: CALL LOAD(-31804,A,B)
This will return you to the title screen when the program is ended.

*17 When hooked up to a black and white tv use CALL SCREEN(15). This will disable the color generator and remove the vertical lines you may have seen.

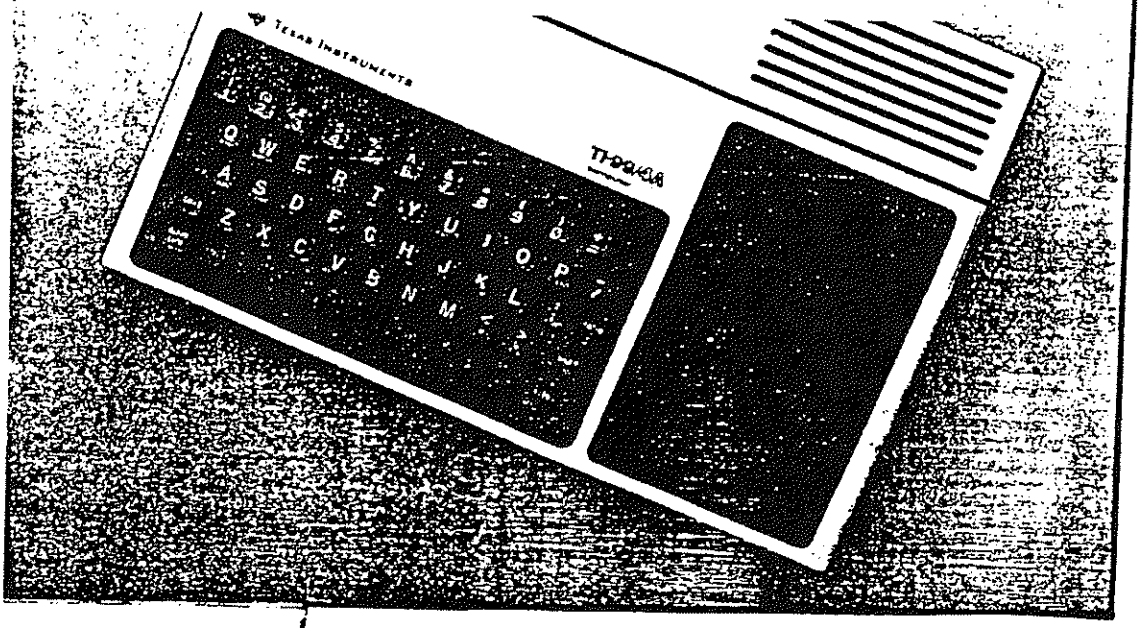
*18 To speed up loading Infocom games, don't use Extended Basic. Use Mini-memory or E/A instead. To use these, select the load and rum option and type DSK1.BOOT. When this is finished loading, press (ENTER) until you get the program name, then type START. On the Mini-memory, you'll get an error after BOOT loads, but keep pressing (ENTER) and proceed as above.

*19 If you want to disable the keyboard for any reason, type in CALL LOAD(-3257,2,128). You will have to turn off the console to regain control.

KNOW YOUR TI99/4A

Submitted by Guy Myers

No. 1



TI99/4A Keyboard
The keyboard is of a higher standard than on most home computers, though some users have commented that the 'bounce' on each key is too stiff. The number of keys is also rather limited, presumably to make room for the cartridge slot on the right-hand side. Most of the keys, therefore, double up — pressing 'CTRL' and 'E' will achieve the cursor-up function. The 'FCTN' key turns the top row into user-definable keys, and it is possible to insert a strip of plastic above this row, on which labels can be written.

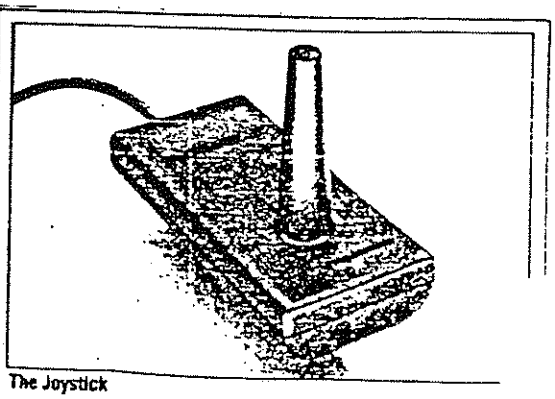
In terms of design and construction, Texas Instruments' T199/4A is one of the most professional home computers.

It uses a 16-bit microprocessor, the TMX9900, designed and made by Texas Instruments, who make semiconductors, calculators, microprocessors and minicomputers. TMS9900 was on of the first 16-bit Chips.

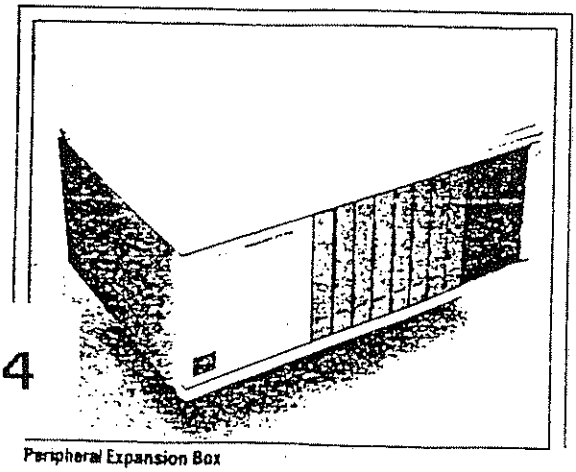
The T199/4A has a 48-key keyboard, which by the general standards of home computer keyboards is very good to type on. There is a sapce to the right of it that receives software cartridges, which Texas refer to as 'solid state software'. A similar connector on the right-hand edge of the case permits hardware expansion. The expansion modules, which are large plastic boxes, contain disk drive controllers, memory expansion and a serial (RS232) interface and are connected via and expansion box, a unit which is essential if you wish to extend th machine.

The screen display is in 16 colours with highresolution graphics, and there is also a sound generator capable of producing three independent notes or 'voices' at once.

The computer is designed for new users to computing, BASIC being the resident language and LOGO the most popular add-on language. In America it is used a lot in schools, and it competes with the Apple II for the position of top-selling educational micro in most States.



The Joystick



Peripheral Expansion Box

TI99/4A

PRICE
Usually £99

SIZE
380×260×70 mm

WEIGHT
1.8 Kg (4lbs)

CLOCK SPEED
1MHz

MEMORY
26 Kbytes ROM, 16K user RAM, 8K graphics RAM. There are an extra 256 bytes of 'scratchpad' RAM not normally available to the user. These are used for the internal registers of the 9900, most CPUs have them built in

VIDEO DISPLAY
Character display of 24 rows of 32 columns. There are 16 colours which can be used as foreground and background colours. No user graphics are available on the basic machine but individual 8 × 8 character cells can be defined with a sequence of 16 characters

INTERFACES
Cassette, joystick, video (not TV), a cartridge slot and a connector for the expansion bus

LANGUAGES SUPPLIED
BASIC

OTHER LANGUAGES
Extended BASIC, TI LOGO, UCSD (University of California at San Diego) PASCAL, TI FORTH, and Assembler

COMES WITH
Power supply adaptor, TV adaptor, cassette connector and manuals

KEYBOARD
Typewriter-style with 48 moving keys, including control and function keys. The numeric keys double as function keys, depending on the added software cartridges

DOCUMENTATION
There is one main manual with an addendum for the UK market, which describes how to connect up the computer and how to use the 'solid state command modules'. This introduction is very short and has many diagrams but no photographs. There is a detailed list of commands available in the BASIC, a section giving some example programs, and a short glossary at the end of the manual

FROM: HUNTER VALLEY 99er

WHATEVER HAPPENED TO THE FUN OF IT?

by Jim Peterson

Yes, whatever happened? Were you with us back in the early days, way back back in 1983? Do you remember the days before the BBS's and user group libraries could supply you with programs by the hundreds, when every program you could acquire was a prized possession to be run and used and marveled at, to be shared with your friends, (even though it might have a copyright notice on it!), when people actually keyed in programs from listings and brought them to meetings to get help in debugging them?

Do you remember when almost everyone was trying to learn to program, and helping each other? The late Earl Dodd was writing music programs, in his unique barbershop quartet style, and bringing them to me to add graphics. Paul Powers, once our user group president, liked to program advanced math theorems and to reprogram more efficiently the programs written by others - it was he who suggested to me the use of mergeable subprograms, which led to the Nuts & Bolts series, the only profitable thing I have ever done. It was teenage Brian Beery who pointed out a ridiculous error that was driving me bonkers - Brian might have become another J. Peter Hoddie if he hadn't taken up the guitar.

Do you remember when the IUG was the only source of public domain programs? I was fascinated by the program descriptions in the IUG catalog. I wanted to see every program, to see what some other programmer had been able to do. I swapped the IUG for every program I could, and bought as many more as I could afford. Often I was disappointed, but I never got over that curiosity. I still have it - I read the descriptions of new library acquisitions in user group newsletters, and I get the itch to see the program. Often I write and ask for it.

But, am I the only one left who has that curiosity? I spent hundreds of hours gleaning out the best from my library of several thousand PD programs, arranging them by category, improving them, filled nearly 300 disks (now 400), published a 13-page catalog listing them all, offered them for a copying fee less than most user groups charge their own members, cheaper than downloading them from GENIE - and in 1989, only 175 people in all the TI world were curious enough to send me an order!

Of course, many users have large libraries of programs that they never get around to even looking at. And, the potential uses of the computer have become so varied that many users have specialized in one field and have little interest in anything else. Some are mainly interested in increasing the speed and memory capacity of their machine, and have little time to make use of that speed and memory by actually running programs. Many others nowadays are hooked on graphics. To each his own. Personally, if I want to decorate my walls with pictures of nudes, I will buy a Playboy magazine and rip out pictures far better than any monitor screen will ever show or any dot matrix printer will ever produce!

Of course, even in the early days all was not sweetness and light. It seemed that everyone was out to make a buck, and those who made the buck were mostly those with questionable business ethics. When I first made contact with the rest of the TI world, I had already written about 90 programs, and I soon met people who wanted to form a business partnership with their one or two programs and my ninety. It was mainly to get them off my back that I decided to go into business for myself - a decision that I have regretted a thousand times.

Charlie LaFara started the International User Group as a nonprofit exchange of public domain software, and converted it into a business for his own profit. An entrepreneur in California acquired his programs and copied his catalog, leading to a lawsuit. Later on, a TI business in Florida called itself a "group" and sold my public domain programs, which led me to announce that "Tigercub Software is a one-man user group pretending to be a business, not a business pretending to be a user group!"

Of course, not everyone was a crook - most simply started out with unrealistic expectations, got in over their heads and faded away, leaving their creditors holding the bag. Emerald Publishing Co. extended credit to too many software advertisers who never paid up; then they got ridiculous and tried to publish a magazine with no advertising! Finally, they ripped off all their remaining subscribers, as did more than one other TI publication which never delivered or refunded subscriptions. There were exceptions, honest companies such as Random Access and another one-man operation in Texas whose name I wish I could remember.

I learned very early not to extend credit to anyone who decided to start up a software business. And I learned not to send an order to anyone for anything until I knew that someone else had actually received their order. There are not many TI old-timers who have not been ripped off at least once!

But, whatever happened to the fun of it? I remember demonstrating my programs at a local school, at a library, at computer fairs - the kids were fascinated! If only the Apple peddlers had not succeeded in brainwashing the educational system! Are Jack Sughrue and Eunice Spooner the only educators still using the TI in the classroom? In the days when I was exchanging my Tips From The Tigercub newsletter with nearly 200 user groups, I twice asked them to let me know of any schools in their area where the TI computer was being used - only two ever responded! If we had harnessed all that youthful energy and enthusiasm, our user groups might now have replacements for all those who are abandoning us for Big Blue.

Whatever happened to the fun of it? Whatever happened to the HOME computer? (remember, that was what the TI-99/4A was called!). They tell me that the COCO is the only home computer left, because there is no way to make it anything more than that. I may just take a look at it - maybe that's where I'll find the fun that I'm missing!

evaluate, manipulate and comprehend the numerals of other systems.

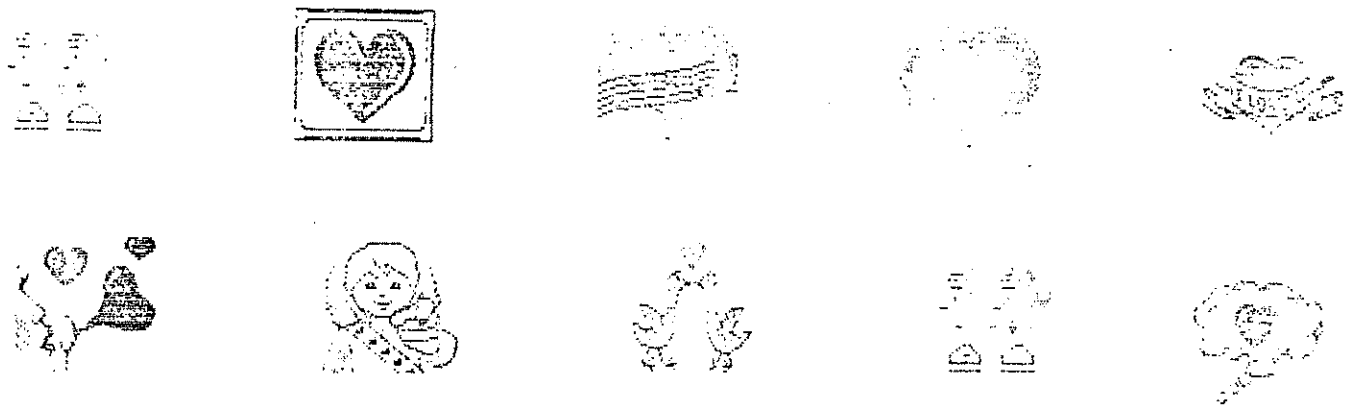
Our anchor of understanding for all systems is "1" and "0", "10" and "11", "10^{Y-1}" and "10^{Y+1}", where Y indicates the power of position, reading from right to left and beginning with 0,1,... These symbols define every notation system of counting numbers. We summarize what you already know for your convenience in recalling:

The universal unit counter, "1", is the only value in common with all systems. The universal "place-keeping 0" has the same meaning in all systems: "no value here but remember the position." Thus "10" always represents the first power of the base of the system, the counting number of the base itself.

So every prime number is equal to 11 in the system where base B=(P-1) and we might conclude that therefore (B+1)=P, a prime number. If B=2 then 11=3=prime. If B=4 then 11=5=prime. If B=6, then 11=7 which is prime. We discover that this is so only if B is an even number. If B=3 then 11=4, if B=5 then 11=6, but if B=10 (decimal system) then 11=11=prime, and for B=12 11=13 (a baker's dozen) which is prime. But if B=8 then 11=9=3x3 and if B=14 then 11=15=3x5. So we discover that in every system where the counting base is an even number, B+1=11 is not always a prime. It may be an odd composite number.

This gives us a little research problem for anyone who enjoys playing with numbers on a computer. It does not require a professional or amateur genius to discover differences between the factorizations of bases where B+1 is prime and those where B+1 is an odd composite number. Any one of us may discover or rediscover another key to the identification of primality.

If any member of a TI99/4A Users Group is interested in this line of research hobby we will continue sharing the results of previous work in the field.



```
11123456789009876543211 1
1                       1
1 PLAYING WITH NUMBERS  1
0        No 13          0
9  By Meredith Beyers   9
9                       9
99987654321001234567899 9
```

THE BASIC LANGUAGE OF THE UNIVERSE:
A View of NUMBERS as a key
to Human Understanding

Numbers not only provide a universal language of Counting, but also of Coding, Communication, Evaluation, Identification and Understanding. They provide a link between the so-called "material" universe of "Nature" and the mental activity of its top product: the individual human being whose consciousness struggles to survive and control through inventive construction.

In order to survive he must bring Order into apparent chaos wherever it touches him. He must gain control over himself and his environment or he will flounder and perish.

Civilized education offers him the organized knowledge of his predecessors, but it is unproductive unless he Uses it to generate momentum for his own self education. Only with this will he write textbooks instead of studying them. . .

And, believe it or not, the present and future Key to all this is NUMBERS. They contain all the little keys to mental and mechanistic controls of complexity to avoid chaos.

This has been the underlying goal and spur of all our efforts since our first essay in The Computer Voice (February, 1990): Title "Innumeracy".

Innumeracy is to Numeracy what illiteracy is to literacy. No one can make a comfortable place for himself in the culture of his native civilization without the literacy of being able to read and write as well as speak in the language of his fellow citizens, and no one can make a profitable place for himself in the growing world of merging cultures without acquiring the numeracy of being able to think and understand, compute and interpret the truly basic universal language of the human race: the Counting Numbers.

Whatever the spoken or written names, words or symbols for the numerals, the NUMBERS are, and will always remain, the same. We inherit a universal means for writing and understanding the numerals used for identification, organizational coding and evaluation: a positional notation system adaptable to any mode of counting or representation.

The alphabets of all written and spoken languages are now interchangable with numbers. Even in our small personal and home computers we can process words as well as compute with numbers and create music as well as illustrative art.

Thus we find our use for identification numbers growing while our understanding of number systems dissipates in confusion. Why? Because we have thus far failed to master the complexities of the tools we use to simplify everything else.

Most of us grew up in the educational environment of a widely standardized decimal system. Today we find children growing up with educational and personal computers based on binary, octal and hexadecimal coding, but programmed to teach decimal arithmetic. So we are all dependent on conversion to base "10" numerals in order to