T.I.-Bings

from New-JUG/North

JANUARY 1988 Volume 6 Number 1

NEXT MEETING: January 19th, Dumont H.S. Faculty Lounge, 7-9:30 P.M.

Motto: We are a family enjoying the unspeakable peace and freedom of being orphans. (Paraphrased from George Bernard Shaw) NOT A BIDEN QUOTE!!!!!!

HAPPY NEW YEAR

GHAMTON, N
PM
JAN 11

New Jersey UG/North
P.O. Box 84
Dumont, NJ 07628

Dallas TI Computer UG*
PO Box 29863
Dallas, TX 75229

User Groups: Please Reciprocate!

TI-Dings from NewJug/North
January, 1988
Happy New Year to all!

>DEFAULTS<

The opinions expressed in these newsletters are solely of the author(s) and do not reflect those of the membership as a group. Reviews and comments are made with the intentions of informing members of products and events dedicated to the TI community.

Letters from subscribers are welcome, either in the form of criticism of the NL's, information, or otherwise.

This newsletter is published monthly (when possible) and other use of its title will be sorely resented.

Copying of any articles or its parts MAY be made without author's permission as long as byline is included.

EDITOR'S RAMBLINGS:

By the 'Farmer in the Dell'
Henry Hein

According to Bill Staedeli the club is still very much alive! Though few showed their faces at the last meeting the group voted a new set of officers. Good to see enthusiasm is still there! Notably, the club felt that the traveling to other meeting sites would be a great inconvenience to most members, the idea of merging with other user groups was out of the question. Feedback from members on the NL appeared to be most favorable. I guess I can hang in here but the disadvantage is the distance. The mail service is slow, and as some of you may have learned, I'm hard to get on the phone. Best after 10 PM or before 8 AM.

Yes, I'm up with the chickens, though I don't have any, but I love the early morning quietude. Besides, I'm on a dual rated electrical system and warm up the house before the high daytime rates come on, and fire up the wood stove. So I'm up at 5 AM or thereabouts, to dry the laundry, make the coffee, warm the house, shower, et al., doing chores quietly so not to wake my sleeping beauty. Sounds like a lot, but routines make things easy. Enough of this!

My barn is almost finished and soon I'll be able to utilize more basement space for various activities.

We, my wife and I, spent four days (Christmas Eve et seq.) visiting ol' Jersey. I tried calling some of you, reached some, either you were out shopping or away. Nevertheless, I hope you have had a great time, as I did.

May your NEW YEAR be greater!!

Special thanks for Frank Lees and others for your wonderful wishes.

I heard that the weather, post Christmas week was weird all around the the metro area. Here, too. We had ranges in temperature from one day to the next of 50 degrees. One day 43+, the next 13 below. Lots of snow, really 8+ inches in an overnight storm, and in the metro area some severe ice storms causing havoc during rush hours. Thanks to satellite communications (TV) I can get all the info from NY and NJ. Believe me, it was worth it. Wonder how they handle it in the NORTH countries.

On the way home from my Christmas journey I made a scheduled stop at the computer flea market held in Suffern's Holiday Inn on the 27th. I met Frank Orlando and John Bonito, and non-TI friends. John suggested we include some IBM software and hardware news for and open the base of club of computer users. There's one like this in my area but no TI users. Parenting the latter group is a good assemblage. Commodore, Apple, Atari, and IBM are all represented. They also have their own BBS. Believe it or not, their library, though extensive, is heavily weighted toward IBM software in public domain. What I've seen of these programs, written by amateurs and pros alike, was absolutely amazing. Too lengthy to report on them here. If it weren't for sentimental reasons, I'm tempted to archive my TI. I'M NOT ABOUT TO DO THAT FOR A LONG WHILE!!

I'm reviving use of my Apple IIC. I recently found some programs that rival the low prices of JOYPAINT, FONTWRITER II, ARTIST, PRINTER'S APPRENTICE, and others. Though just as complicated as above, I'll keep on tinkering with graphics, writing, and data basing.

The computer fair, by the way, had many IBM and Apple PUBDOM programs to do
>>>>>>>>>>>>>>>>>PAGE 2

the things I wish our little 48K machine can't, that is, without the extra effort needed. I'm sorely tempted now. But, on the other hand, there are things they can't do as readily as ours, and for that matter, as cheaply! It's really a mixed bag! OUR 'DOS' is the easiest to handle, our XBASIC is so easy to access, edit, and work with than the others. FOR SURE! But, the memory!!, oh so lacking!

N E W S B Y T E S:
New FUNLWEB

Steve, Jr. told me that he got the latest copy of the above, version IV. It just goes to prove that perseverence by Tony and Will McGovern toward perfection is worth reward. I thought it was perfect enough with the 3.1.4 version. I finally got the 3.4.4 sent to me by Ellen Kramer to work but could not download the documentation and the loader due to some disk imperfection. Tony and Will always provide some interesting reports on what was updated but, in this case, I'm in the dark except that that version provided a distinctive white marker for a carriage return symbol. An improved screen character set is on the 3.4.5 version.

Complete Filing System v.7

Frank Filice bought the program and if you have any q's feel free to call after his office hours. I heard it does number crunching, but I don't know for sure. I also wonder if it can store data in DV 80. Remember what I wrote last month? MICROPENDIUM's review doesn't mention either number crunching or type of file storage. The reviewer (Bill Gaskill) did write that line graphs can be developed from numerical data entered in each record. He did rate it second best to PRBASE

RE: NOVEMBER TI-DINGS

Still missing! Postal trace has not found the masters (galley proofs) mailed to Steve. How could the PO lose a large 9 1/2 X 12 inch envelope with six double sided sheets with a return address on the first (and envelope) is beyond my comprehension.

RE: COVER PAGE:

The current CP was done with the APPLE IIC and a program called MULTISCRIBE. The best feature here is that the program allows to mix sizes and fonts on the same line. Thought I'd try it out for size. Otherwise it is pretty neat, isn't it? P.S. It's a word processor, besides, in its own right. Fonts are not easy to make, and, ike other font making programs, it IS a tedious chore. And, unlike some software made for TI, there is no way to convert fonts of one graphics program made for APPLE to another. MULTISCRIBE does allow the user to create little pics that you might substitute for a letter and embellish text. The program recognizes each pic you draw as a separate letter (pictogram). A sample file of these pictograms is included for use or you can make another. For example, when hitting the letter "A" you can get an apple and an "k" will draw a key. It's called the MICHELANGELO FONT. Interesting, yes, and mixing sizes or fonts can be done. That is really something! This doesn't encourage me to archive the TI, though. It just adds to variety. Programs made for IBM, even some in pubdom can be found doing these marvelous things, too. But Thanks again to TI's DV80 we can still do the 'font tricks' of good programs made for us.

Again, on the APPLEs.. Finally, at long last, (we've had it for three years or more) someone wrote a MULTIPRINT program for the APPLE and featured in Feb. COMPUTE magazine. Another mag had a version last summer. Both limited to to a max 3-column spread, however. We do one better!, and a sideways program, besides, with Tom Freeman's disk mentioned last month and several issues before.

NOTE: COMPUTER SHOPPER magazine still carries its TI column. Other than MICROPENDIUM, which covers the most news, there appears to be nobody else except club newsletters. More than half of them cover ample news and hard copy programs, programming tips, jokes, etc. I'm particularly impressed by the graphics they portray. Some are done professionally in printing houses and/or with the help of LASER printers. Wouldn't it be nice to have one? Yes, they can be accessed by BASIC and XBASIC commands, TIW, but perhaps not by

>>>>>>>>>>>>>>>>>PAGE 3

current graphics programs which have embedded codes like TI ARTIST, GRAPHX, etc. The nearest thing to a LASER is a 24 pin dot matrix. There are lots of them around and at reasonable prices.

FLASH:

John Bonito sent me a lengthy bulletin my copy machine can't back up. It contains some interesting info re: RYTE DATA products and others. Awhile ado I mentioned a kit to make your TI Disk Controller into DD. It is a kit of materials you MUST solder to the present controller card. He cautions that it is ONLY compatible with the TI controller card, and will not recognize disks formatted or filled with CorComp, MYARC or other techniques.

John said that MICROPENDIUM feedback writers complain about R/D's poor business practices. He also mentioned that the National 99 UG was another firm cited. Walt Macieski ordered $50 worth of programs from them and hasn't heard anything from them.

Walt, they were defunct for a long time. Did they cash your check? If so, you have grounds for fraud. I believe the K-Town UG has their complete library now. Ed.

John also said that the 80-column card offered by R/D (above) can ONLY be utilized with TIW when calling the PF command. Also, according to John, the purchase of a PC connection may bring disappointment. To save anything you must have an IBM formatted disk. The latter doesn't format disks for IBM use.

Complaining of MICROPENDIUM hard copy he finds that some of the programs don't run and waste too much of his time trying to debug them. Corrections often appear, if they do, months later.

Material regarding user groups fading everywhere, particularly of orphaned computers, from John highlights the demise of the Amateur Computer Group of New Jersey. This group does not seem to have a center of interest anymore, that is, in special interest groups (SIGs) such as FORTH, ASSEMBLER, techie add ons, etc.

For that matter, come to think of it, neither do individual TI clubs to any great extent. There used to be a national FORTH SIG in Florida and now no longer active. The defunct National 99ers used to have an E/A SIG which disappeared almost 4 years ago. The ROM UG is still alive and kicking but without the fanfare and magnitude of last year. The same is true for many others. One group that is still the most enthusiastic is in AUSTRALIA and from what I hear puts out a tremendous NL. The Chicago TImes NL is an inspired journal for users and can be subscribed to. For subscription info write to Tony and Will McGovern (see FUNLWRITER). For Chicago TImes UG write to PO Box 578341, Chicago IL 60657 for membership. Subscribers must me members, according to their stated policy. This latter group has a lot of SIGs, including PASCAL.

B E W A R E:
or the Adventures of Henry

In early December I replied to an ad in the Binghamton Press-Bulletin (NY) re: a free computer offer and to make some extra change. It sounded too good. I was called on the phone in reply to my query by someone in Chicago to go to a motel in Binghamton to be briefed. I asked what kind of computer would I be getting if I agreed to the terms of the offer. Reply, an IBM. The meeting was 24 hours and 50 miles away with an impending snowstorm on the way. I asked for an appointment at another time or place and the woman suggested next month in Long Island. She definitely didn't know the Geography of NY, another 200 miles? No way! I told her I'll be in Bingotown tomorrow night on dogsled if I had to.

Yep, the snow came but the highway was sanded, salted, and prepared for the worst. It was blinding snow but melting as it fell. No traffic, no problem, and I got there with one hour headway, just in case.

Finally, the company rep came and gave a spiel to about 20 enthusiastic folk. Here was the offer. We would hook up our computers to a separate phone in the house, punch in phone numbers from a local directory, enough for a computer to do a day's work of calling up people about a certain product, telephone poll, etc. Really nice, one hour/day to punch in the numbers, and once or twice a week ship the data to their central office in

>>>>>>>>>>>>>>>>>>>>PAGE 4

Chicago. Yes, we earn about $700/month doing this. Here's the catch. I must BUY an APPLE II E with 2 DD's (approx. $1400) and the software/peripheral package for, get this, $6400. Plus, I had to pay them a license fee of $240/yr for the privilege to use their equipment should I want to do my own business, or theirs, too. The software is really what turned me off. All it does is turn on a cassette recorder to send a voice message, and receive a voice message, and that I would have to listen to it and punch in the response data into the computer the names and addresses of those who responded to the recorded message favorably. Who needs all that? Even a TI can do it, and with home made programs.

Here is another kicker! If you didn't feel up to continuing to work for them, you can sell back your computer for half of what you paid for it. What about the $6400 in the other package.

How do you pay for the instruments? Well, take out a business loan, of course. The company didn't finance it.

It was quite a spiel. Going into business you can write off depreciation on your income tax. Write off the interest, too.

All I have to say is beware of CYBERTRONIXX, Inc. Yes, you CAN make money on it and come out ahead. But they advertised a FREE COMPUTER, and was led to believe an IBM. I didn't bite the bytes, nor do I recommend it. FREE means FREE and I'm asking the AG office to look into it. The company is based at 1171 Tower Road, Shaumburg, IL 60173. Their 'expertise' is market research. I wonder!

How about you Chicagoans? Want to check them out?

That was a tale to tell and I couldn't resist it. It's interesting to note the scheming that goes on for the almighty buck. Who, in the long run, really gains from it?

P.S. The snow didn't start to stick until I got back home. The snow job didn't stick either, except in my mind,

for YOUR edification.

SUPPORT LIST:

In the past I've been remiss in listing addresses of TI support and now that I've got a little file together with the help of Joe White of K-Town 99ers. Here they are with my comments:

I'm sure they have listings of their products available and sending SASE's would speed things up in getting back to you.

GENIAL COMPUTERWARE, (Boston branch, use this address for ordering all products except Genial TRAVeler), P.O. Box 183, Grafton, MA 01519.

GENIAL COMPUTERWARE, (Philadelphia branch, use for ordering Genial TRAVeler), 835 Green Valley Dr, Philadelphia, PA 19128, 215-483-1379. A HOT ITEM. Also, from either the above you can order the latest modem data PC TRANSFER program mentioned several months ago.

Mike Dodd, 116 Richards Dr. Oliver Springs, TN 37840, I believe put out an update of PRBASE and a few other popular programs. Ask for a listing with a SASE to make it easy. Author of PC TRANSFER, also available from above.

J.Peter Hoddie, 12 Paul Revere Road, Lexington, MA 02173, author of many useful programs sold commercially and MANY for the price of a disk through the Boston Computer Society. SASE.

BYTEMASTER Computer Services, 171 Mustang St. Sulphur, LA 70663-6724.

DIJIT SYSTEMS, 4345 Hortensia St, San Diego CA 92103, I believe make a RGB adaptor and an 80 column card for the TI among other things.

RAVE 99, 112 Rambling Rd. Vernon CT 06066. Users rave about their IBM or AT style keyboards.

Los Angeles 99ers, PO Box 3547, Gardena CA 90247-7247. A really great UG and NL. Worth joining or subscribing. I think it was $15/yr

DATABIOTICS, P.O. Box 1194, Palos, Verdes Estates, CA 90274. Has a cartridge/cassette variation of TI Writer among many other things worth looking into.

>>>>>>>>>>>>>>>>>>PAGE 5

MYARC, Inc., PO Box 140, Basking Ridge, NJ 07920-1014, 201-766-1701. You know who and what they do!

TIGERCUB SOFTWARE, 156 Collingwood Ave, Columbus, OH 43213. $1 gets you a catalog deductible on first purchase. Well worth it. Jim Peterson deserves our support!

ASGARD Software, P.O.Box 10306, Rockville, MD 20850, If you don't know about them yet, it's really time you did!

Chicago TI UG, P.O.Box 578341, Chicago, IL 60657. Like LA, a GREAT group to join or subscribe!

HORIZON COMPUTERS: P.O.Box 554, Walbridge, OH 43465. Another RAM disk expert! Ask Frank Filice.

Bud Mills, 166 Dartmouth Dr., Toledo OH 43614. I don't know myself but will try to find out what he offers. Do the same! SASE

MICROPENDIUM, P.O. BOX 1343, ROUND ROCK, TX 78680. YOU KNOW WHAT!

DISK ONLY SOFTWARE, P.O.Box 244, Lorton, VA 22079.

Great Lakes Software, 804 E. Grand River Ave, Howell, MI 48843

For others like McCann Software (The Printer's Apprentice, etc. et al), RYTE DATA, and others, go through our old NLs and compile them with these. Why not try them in your databases? or TIW?

REMINDER: Members dues are due! They have been reduced in anticipation of the club's foundering by the end of the year or before. I just hope that it'll pay for the NLs printing and mailing costs for that time period even though we APPEAR to have an large surplus due to sales of excess equipment. I don't like to say it but it has to be said. Pay up or lose out on the NL by the end of January. Feel free to write or call me at RD #1 Box 343A, Otego NY 13825, 607-988-7789 for information or call any of the officers or previous officers for help on any TI problem. Regards for now, Henry.
<<<<<<<<<<<<<<<<<<<<<EOF>>>>>>>>>>>>>>>>>>

QUESTIONNAIRE for the NewJug/North Members:
R.S.U.P. RD #1 Box 343 A, Otego, NY 13825

1. Would YOU like to write an article, collection of data, jokes, graphics, or in any way contribute a listing of a favorite program/project?
2. What questions would you like answered re your computer and its capabilities? or program capabilities?
3. Do you have any answers for people who pose above questions?
4. What problems have you encountered and solved yourself? Would you share with us your findings?
5. Remember our old motto: "How can I help?" It's supposed to have been a mutual endeavor! So let's reexamine it and lend me a hand to put out a better newsletter. How can I help?! Let's hear it!

To start 1) call up TIW or Funlweb's TIW, set margins .LM 0;RM 39;IN 4;CE 2 (for title and byline). Make it easier, set your tab (command line) L over 0 and R over 33. Now you'll be able to see all your text as you type. When finished just save file to disk and mail it off to me, above. I'll edit it, clean it up, publish it, and return your disk. That's Help!
Thanking you ahead of time and I can hardly wait for your replies!

Henry

TI-DINGS JANUARY 1988 CONTINUED:

Editor's Note:

The Gleanings of OCT 87 MICROPENDIUM were received too late for last month's NL. The November Gleanings just arrived for publication. H H

Gleanings from Micropendium
BY Frank Filice
Vol.4 No.9,October,1987

Gleanings from Micropendium
by Frank Filice

Vol.4 No.10,November, 1987

>>>>>>>>>>>>>>>>>>PAGE 7

LET'S TALK RAM DISK
by John F. Willforth

We have a problem as TI-99/4A users these days that most market viewers and many TI owners would not have believed possible just a few months ago. There are four major vendors of RAM DISKS in the U.S.A.. There is also a variety of features and sizes in these units, some of which are not found in units being produced for Atari, Commodore, Apple, or the P.C. lines of computers. The biggest problem facing the user now is "which to buy ?".

The purpose of this article is to provide some thoughts and facts to help you decide. The next several paragraphs are not intended to promote any one of the RAM DISKS mentioned, and may contain erroneous information, hopefully by omission rather than commission.

First a RAM DISK is by definition, a software/firmware supported RAM circuit board emulating a DISK. i.e. a circuit card, that when plugged into your PEB, will allow you to store and retrieve disk type files to/from the unit with the same ease as you would to your physical disk drive (DSK1 for example). Because RAM is a non-mechanical device, it is not subject to the delays of positioning a read/write head over a cylinder (TRACK), and waiting for the diskette to now rotate to the desired sector, and then read/write data from/to the spinning disk in serial (like cassette) form. These three mechanical limitations are the main reason that disks are slow. Yes, disks are about as much faster than cassettes as RAM DISKS are faster than disks! If you buy one, YOU WILL enjoy that kind of improvement, no matter which brand you buy.

The major two types of RAM DISKS are those using DYNAMIC RAM (MYARC, CORCOMP) and STATIC RAM (HORIZON, MIKE BALLMAN enhanced HORIZON [sold by Bud Mills]).

* DYNAMIC RAM is less expensive, larger capacity, but requires more support circuitry, draws more power, and is more cumbersome to support if the power is lost (like turning off the PEB).
* STATIC RAM is lower power and thus easy to support during power outages. They are more expensive, take more space on a board, and thus for the amount memory needed, more expensive than DYNAMIC RAM.

I would like to talk about additional features. The first one that I am most asked about, is the spooling features. All but the HORIZON and the enhanced HORIZON, have the spooling feature. At this printing, all that have spooling do it in a different manner, but just as effectively. Some of you may ask, "What is SPOOLING?". Well to make it simple, spooling is storing data that is to go to a device (printer modem, etc.) in memory space, and releasing it as it can be used by the receiving device. Remember the TI sits there sending to the printer until all the file is sent. Then it is able to accept your next command or continue instructions. A spooler accepts this information as if it were the printer, modem etc., and at a much higher rate than any of those peripherals could, and in most cases will accept the entire file to be processed in a few seconds verses several minutes. The TI-99/4A will then ASSUME that all that it had to do was done and come back to you for further use, when in fact the job is still being completed by the spooler at a pace that the printer, modem etc. can handle it. Pretty neat! Huh?

Another feature is partitioning, or multiple disks being assigned within a single RAM DISK CARD. What this means is that if you have a single drive on your system (DSK1 for example), you may call a portion of a RAM DISK DSK2, or DSK3, DSK4, etc. Now you have one physical, and up to who knows how many other disks which are part of the RAM DISK.

Still another feature is built in COMMANDS, each disk mentioned above has it's own set. For example, you can type "CALL DM" in BASIC COMMAND MODE, and a file called DM1 will be booted from the disk, followed by DM2. Many commands dealing with memory are also incorporated.

Features such as CLOCK, (Time Of Day), Analog-To-Digital, etc are now coming available on the RAM DISKs.

You may need more information to order your RAM DISK than I've provided here. Next month's article will get more specific on each RAM DISK but if you believe the ads, maybe you can understand them a little better now. and your ready to jump in. Good luck!

From Pittsburg UG

DEBUGGING

by Jim Peterson

When you have finished writing a program, the next thing you should do is to run it. And, very probably, it will crash!

Don't be discouraged. It happens to the very best of programmers, very often.

So, the next thing to do is to debug it. And you are lucky that you are using a computer that helps you to debug better than some that cost ten times as much.

There are really three types of bugs. The first type will prevent the program from running at all - it will crash with an error message. The second type will allow the program to run, but will give the wrong results.

And the third type, which is not really a bug but might be mistaken for one, results from trying to run a perfectly good program with the wrong hardware, or with faulty hardware. As for instance, trying to run a Basic program, which uses character sets 15 and 16, in Extended Basic.

First, let's consider the first type. The smart little TI computer makes three separate checks to be sure your program is correct. First, when you key in a program line and hit the Enter key, it looks to see if there is anything it can't understand - such as a misspelled command or an unmatched quotation mark. If so, it will tell you so, most likely by SYNTAX ERROR, and refuse to accept the line.

Next, when you tell it to RUN the program, it first takes a quick look through the entire program, to find any combination of commands that it will not be able to perform. This is when it may crash with an error message telling you, for instance, that you have a NEXT without a matching FOR, or vice versa.

And finally, while it is actually running and comes to something that it just can't do, it will crash and give you an error message - probably because a variable has been given a value that cannot be used, such as a CALL HCHAR(R,C,32) when R happens to equal 0.

The TI has a wide variety of error messages to tell you when you did something wrong, what you did wrong, and where you did it wrong. But, it can be fooled! For instance, try to enter this program line (note the missing quotation mark).

100 PRINT "Program must be saved in:"merge format."

And, sometimes you may be told that you have a STRING-NUMBER MISMATCH when there is no string involved, because the computer has tried to read a garbled statement as a string.

Also, the line number given in the error message is the line where the computer found it impossible to run the program; that line may actually be correct but the variables at that point may contain bad values due to an error in some previous line.

If the error occurs in a program line which consists of several statements, and you cannot spot the error, you may have to break the line into individual single-statement lines. This is the easiest way to do that - Be sure the line numbers are sequenced far enough apart. Bring the problem line to the screen, put a ! just before the first ::, and enter it. Bring it back to the screen with FCTN 8, retype the line number 1 higher, use FCTN 1 to delete the first statement and the ! and ::, put a ! before the first ::, and continue. Then, when you have solved the bug, just delete the ! from the original line and delete all the temporary lines.

Pages 212-215 of your Extended Basic manual list almost all the error codes, and almost all the causes of each one - it will pay you to consult these pages rather than guessing what is wrong.

You may create some really bad bugs when you try to modify a program that was written by someone else - especially if you add any new variable names or CALLs to the program. Your new variable might be one that is already being used in the program for something else, perhaps in a subscripted array. I have noticed that programmers rarely use @ in a variable name, so I always tack it onto the end of any variable that I add to a program.

Also, the program that you are modifying may have ON ERROR routines, or a prescan, already built in. The ON ERROR routine was intended to take care of a different problem than the one you create, so it could lead you far astray - you had better delete that ON ERROR statement until you are through modifying.

The prescan had better be the subject of another lesson, but if the program has an odd-looking command !@P- up near the front somewhere, it has a prescan built in. And if so, if you add a new variable name or use a CALL that isn't in the program, you will get a SYNTAX ERROR even though there is no error. One way to solve this is to insert a line with !@P+ just before the problem line, and another with !@P- right after it.

When a program runs, even though it crashes or is stopped by FCTN 4 or a BREAK, the values assigned by the program to variables up to that point will remain in memory until you RUN again, or make a change to the program, or clear the memory with NEW. This can be very useful. For instance, if the program crashes with BAD VALUE IN 680, and you bring line 680 to the screen and find it reads
CALL HCHAR(R,C,CH)
just type PRINT R;C;CH and you will get the values of R, C and CH at the time of the crash. You will find that R is less than 1 or more than 24, or C is less than 1 or more than 32, or CH is out of range.

In Extended Basic, you can even enter and run a multi-statement line in immediate mode (that is, without a line number), if no reference is made to a line number. So, you can dump the current contents of an array to the screen by
FOR J=1 TO 100::PRINT A(J);::NEXT J - or you can even open a disk file or a printer to dump it to.

You can also test a program by assigning a value to a variable from the immediate mode. If you BREAK a program, enter A=100 and then enter CON, the program will continue from where it stopped but A will have a value of 100.

You can temporarily stop a

program at any time with FCTN 4, of course (the manual says SHIFT C, but it was written for the old 99/4), and restart it from that point with CON. Or you can insert a temporary line at any point, such as 971 BREAK if you want a break after line 970. Or, you can put a line at the beginning of the program listing the line numbers before which you want breaks to occur, such as 1 BREAK 960,970,980 Note that in this case the program breaks just BEFORE those listed line numbers. You can also use BREAK followed by one or more line numbers as a command in the immediate mode.

The problem with using BREAK and CON is that BREAK upsets your screen display format, resets redefined characters and colors to the default, and deletes sprites. So, it is sometimes better to trace the assignment of values to your variables by adding a temporary line to DISPLAY AT their values on some unused part of the screen. If you want to trace them through several statements, it will be better to GOSUB to a DISPLAY AT. And if you need to slow up the resulting display, just add a CALL KEY routine to the subroutine.

Sometimes, your program will appear to be not flowing through the sequence of lines you intended (perhaps because it dropped out of an IF statement to the next line!) and you will want to trace the line number flow. This can be done with TRACE, either as a command from the immediate mode or as a program statement, which will cause each line number to print to the screen as it is executed. If used as a command, it will trace everything from the beginning of the program, so it is usually better to insert a temporary line with TRACE at the point where you really want to start. Once you have implemented TRACE, the only way to get rid of it is with UNTRACE.

TRACE has its limitations because it can't tell you what is going on within a multi-statement line, and it will certainly mess up any screen display. Sometimes it is better to insert temporary program lines to display line numbers. I use CALL TRACE( ) with the line number between the parentheses, and a subprogram after everything else

```
30000 SUB TRACE(X)::DISPLAY
 AT(24,1):X :: SUBEND
```

Some programmers use ON ERROR combined with CALL ERR as a debugging tool, but I can't tell you much about that because I have never used it. ON ERROR can give more trouble than help if not used very carefully, and I cannot see that CALL ERR gives any information not available by other means.

Sometimes you can debug a line by simply retyping it. It is only very rarely that the computer is actually interpreting a line differently than it appears on the screen, but retyping may result in correcting a typo error that you just could not see. In fact, most bugs turn out to be very simple errors.

When you are debugging a string-handling routine, don't take it for granted that a string is really as it appears on the screen - it may have invisible characters at one or both ends. Try PRINT LEN(M$) to see if it contains more characters than are showing; or PRINT "*"&M$&"*" to see if any blanks appear between the asterisks and the string.

There is no standard way to debug a program. Each problem presents a challenge to figure out what is going wrong, to devise a test to find out what is really happening.

Don't debug by experimenting, by changing variable values just to see what will happen, etc. Even if you succeed, you will not have learned what was wrong so you will not have learned anything - and if your program contains lines that you didn't understand when you wrote them, you will have real problems if you ever try to modify the program. (Believe me, I speak from experience!)

Someday Somebody Else's (remember him) article, picture, joke, bit of wisdom, hint(s), or anything else, might fill in these blank columns!!

SPRITES, PART 1 - by Jim Peterson

The sprites of TI Extended Basic are mostly used in fast-action arcade-type games, but they have other uses as well.

Up to 28 sprites can be placed on the screen at one time, but there is one very serious limitation - if more than 4 of them are in a line horizontally, only the 4 lowest-numbered ones will be visible. That is why, if you have numerous sprites moving about the screen, one of them will occasionally disappear and reappear, or a horizontal slice of a magnified sprite will become transparent.

A sprite is placed on the screen by the statement
CALL SPRITE(#N,ASC,COL,DOTROW,DOTCOL)

N is the sprite number, between 1 and 28, and it must be preceded by the # sign. ASC is the ASCII code of the character that you wish the sprite to have. It must be between 32 and 143 - the ASCII characters 33 through 126 are the keyboard characters, the others will be blank unless you redefine them. COL is the color you wish the sprite to have, using the same color codes, 1 to 16, as are used for CALL SCREEN or CALL COLOR.

DOTROW and DOTCOLUMN are the dot row and dot column at which you wish the sprite to appear. You know that the monitor screen consists of 24 rows and 32 columns. Using HCHAR or VCHAR, you can place a character on any one of those 768 spaces (PRINT and DISPLAY start at column 3 of the graphics screen). Each of those spaces consists of a grid of 8 x 8 dots, totaling 64. By turning various of those dots off (blank) or on (colored), a character is displayed on the screen. Therefore the screen is 8 x 32 or 256 dotcolumns wide and the visible screen is 8 X 24 or 192 dotrows deep. Actually dotrow can be anything up to 256; dotrows 193 through 256 are hidden below the bottom of the screen, and sprites can be hidden there.

The upper left hand corner of your sprite will be at whatever dotrow and dotcolumn you specify.

To convert an graphics screen (HCHAR) position into dotrow and dotcolumn, use DOTROW=ROW*8-7 and DOTCOL=COL*8-7; to convert a PRINT/DISPLAY position, you must use DOTCOL=(COL+2)*8-7.

So, CALL SPRITE(#1,42,16,89,121) will place sprite #1, in the form of the asterisk (ASCII 42), colored white (16) in the middle of the screen. If you want, you can give it motion when you create it, by giving it a row-velocity and a column-velocity. These velocities can be from -128 to 127. A positive row velocity moves the sprite down, negative moves it up; a positive column velocity moves it right, negative moves it left.

Velocity 0 is a standstill, and speed increases from 1 upwards and from -1 downwards.

So, CALL SPRITE(#1,42,16,89,121,5,5) will place that white asterisk in the middle of the screen and start it movin slowly at a 45 degree angle downward to right (since the values 5 and 5 are positive and equal). It will continue moving at that direction and speed until you tell it to do otherwise, all by itself and without program control. When it reaches the right edge of the screen, it will "wrap around" and appear at the left. When it reaches the bottom, it will disappear briefly while it passes through those hidden dotrows, and "wrap around" to appear at the top.

If you want to change the pattern of the sprite, there are three ways to do so. You can CALL SPRITE again with the same sprite number but a different ASCII character - but if the existing sprite is not in the position of the dotrow and dotcolumn you specify, it will disappear and reappear in the new position. Or you can reidentify a character by CALL CHAR, and any sprite having that character will change accordingly, without affecting its color, position or movement. Or you can use CALL PATTERN(#N,ASC) to change the pattern of sprite #N to the pattern of the specified ASCII character, without affecting color, position or motion.

There are also two ways to change the color of a sprite. CALL SPRITE with the same sprite number and ASCII but a different color code will recreate the sprite with the new color, but in whatever position is specified. CALL COLOR(#N,COLOR) will recolor sprite #N to the specified color code without affecting its pattern, position or motion.

If you want to change the position of a sprite, CALL LOCATE(#N,DOTROW,DOTCOL) will make it disappear at its old location and appear at the new location. The pattern and color will be unchanged, and if it was in motion the same motion

TI-D1Nes
NewJug/North
Jan 88
p 11

will continue from the new position.

To change the motion of a moving sprite, or to start a stationary sprite into motion or vice versa, use CALL MOTION(#N,RV,CV) - RV and CV being the same row velocity and column velocity optionally used in CALL SPRITE. CALL MAGNIFY will change the size of your sprite. You do not specify a sprite number with this CALL, because it affects all sprites that are on the screen or are subsequently placed on the screen. CALL MAGNIFY(2) enlarges the sprite 4 times so that it fills 4 of the graphic screen spaces, 256 dot spaces. CALL MAGNIFY(3) causes the sprite to consist of 4 characters, occupying 4 graphic screen positions. The upper left of these characters will be the ASCII specified in the CALL SPRITE or CALL PATTERN, provided that the ASCII is evenly divisible by 4 - otherwise, it will be the next smaller ASCII evenly divisible by 4. The next higher ASCII will be in lower left, the next in upper right, the next in lower right. In other words, if you use CALL MAGNIFY(3) and CALL SPRITE(#1,64,2,10,10) you will get a sprite looking like this - @B
AC
- and if you CALL SPRITE(#1,65,2,10,10) you will get exactly the same thing, because the computer will substitute the next lower number, 64, which is evenly divisible by 4.

Naturally, you will not have much use for sprites consisting of four characters, unless you redefine them into a single pattern, and in that case you must remember that they will appear in that upper left/lower left/upper right/lower right sequence. Fortunately, there are sprite editor programs to take care of this for you.

CALL MAGNIFY(4) will enlarge that 4-character sprite so that it fills 16 graphic screen positions. Note that magnification options 2 and 4 actually enlarge each dot to fill 4 dot positions, so that the sprites have a more angular, blocky appearance.

And finally, CALL MAGNIFY(1) will return magnified sprites to their normal single-space size.

Programming with sprite motion is unlike any other programming, because you do not control the program execution step-by-step. When you set a sprite in motion, it continues in motion while the program goes on to do whatever it is supposed to do next. When you want to control the sprite again, you must catch up with it and find out where it is. There are three ways to do this.

CALL COINC(ALL,C) will give a value of -1 to C if any two sprites on the screen are overlapping, even slightly, or 0 if they are not. CALL COINC(#1,#2,TOL,C) will give C a value of -1 if the upper left hand corners of sprites #1 and #2 are within TOL dotrows and dotcolumns of each other. TOL may be any number you want, depending on whether you want to catch them only when they are right on top of each other, or just getting close. If not within tolerance, C will equal 0.

CALL COINC(#1,DOTROW,DOTCOL,TOL,C) will give C a value of -1 if the upper left corner of sprite #1 is within TOL dotrows and dotcolumns of the specified DOTROW and DOTCOL.

CALL COINC is not foolproof. If you give the sprites a fast motion, a coincidence may not be caught. And when you alternate your CALL COINC with other statements such as CALL JOYST, a coincidence will be missed if the program is executing some other statement at the time.

CALL POSITION(#N,DOTROW,DOTCOL) will give the dotrow and dotcolumn that the upper left corner of the sprite is occupying at the instant it is called. This one again is not foolproof because the sprite will have moved from that position before another statement can be executed to do anything with the information.

CALL DISTANCE(#1,#2,D) or CALL DISTANCE(#1,DOTROW,DOTCOL,D) will give to D a value depending on the distance between the two sprites, or between the sprite and the location. The value, as I understand it, is the square root of the total of the squares of the difference between the dotrows added to the squares of the differences between the dot columns. I'm not sure how useful all that is, and I have rarely seen this CALL used by programmers.

Finally CALL DELSPRITE(#N) will delete sprite #1 from the screen and CALL DELSPRITE(ALL) will delete them all.

Those are just the basics of sprite programming. What can be done depends solely on your ingenuity.

SPRITES, PART 2 - by Jim Peterson - Coming soon

New Sug/North
Jan 88
p. 12