

CLEVELAND AREA 99-4A USERS GROUPS NEWSLETTER

AUGUST/SEPTEMBER, 1986

OFFICERS	SOLOM	NORTHCOAST	GOLDEN CRESCENT	TI-CHIPS
PRESIDENT	STEVE WEINKAMER (1-274-2544)	RON MINADEO (943-2791)	CHUCK MARENO (324-4388)	RICH POLIVKA (238-3971)
VICE PRESIDENT	WALT RYDER (921-8223)	MARTIN SMOLEY (257-1661)	DICK BURGER (967-1317)	RUSS SHIMANDLE 1-887-5330
TREASURER	FRANK JENKINS (283-8526)	JIM HEKEEL (286-3179)	DALE SCHIEFERSTEIN (988-5150)	DAN KELLER (225-0658)
SECRETARY	RUSS OGDEN (1-626-2622)	JOHN BLACKMER (257-2672)	HAROLD SKIPWORTH (631-4529)	JAN FEDOR (524-6192)
LIBRARIANS	KIM JONES (1-274-8019) LADD PATA (662-9253)	DEANNA SHERIDAN (333-5986)	BOB BARTO (1-282-7037)	MARK McCAULEY (235-8888)

BULLETIN BOARD----- (24 HOURS) (216) (944-1072)

REMEMBER, this is a double issue and none will be printed in September.

To all who responded for our request for indexing newsletters: Marty Smoley has agreed to coordinate this project. He has your names and what you are willing to do and will be in contact with you soon. This probably won't get into full swing until September when everyone is back to a regular schedule. Thanks for your offer and you WILL BE CALLED UPON FOR ASSISTANCE.

Didn't anyone like Jim Hekeel's suggestion for a programming contest. Again, this would be a fall and winter project...not something you had to start on today.

A call for help again...We need some people to help with library cataloging. The same person (DEANNA) has been acting as chief librarian for all 4 clubs and also as the puter-together for the newsletter and it is getting to be too much for one person to handle. We have volunteers to take the Utilities and Music/Graphics sections and would like people for Basic Games, Extended Basic Games, Education, Home/Financial sections. We will be getting a large number of programs from Youngstown in a month or so and would appreciate any help in this area you can provide. See your club's librarian who will help Deanna coordinate this.

A new FREE BBS service went into service on July 16 called Cleveland Free-net, the brainchild of Dr. Thomas M. Grunder, an assistant professor at Case Western Reserve University's School of Medicine. This is an outgrowth of St. Silicon's and hopefully the network will expand to enable users to do their banking electronically, register for college courses, and receive the latest news, weather and sports reports, among many other uses. The system is available for lease to any community for \$1 per year. It is to have 15 phone lines (not yet installed). This is an extensive database and the PUBLIC SQUARE area includes computer 8165. We are in the process of contacting Dr. Grunder to see if we can add a TI 816 and this would give us exposure we could find no other way. The phone number is 216/368-3888. It can be used by any computer with a modem.

Somewhere I read that C.Regena is now marketing her own software. About time! Wish they would let her do more advanced items for Compute!. She is certainly capable of same, but they don't seem to realize that the people who are still using their TI's have E/A's, disk drives, J2K, modems, etc., etc. Her address is C.Regena, P.O. Box 1502, Cedar City, Ut 84720.

The July Bytesonger (Lexington, KY) reports that Disk Only Software has announced an agreement with Paul Charlton, author of FAST-TERM, to develop a "revolutionary" new terminal emulator program. The company is in search of new ideas to include and wishes to involve the TI community. If an idea suggested to the developer is unique and beneficial, a development letter will be issued with possibilities of remuneration of acknowledgement, depending upon the extent of work required. DOS asks that you send all ideas and suggestions to: Disk Only Software, P.O. Box 4170, Rockville, MD 20850 and include a SASE.

The same newsletter states Computer Shopper is undergoing radical surgery with the August, 1986, issue. A better grade paper will be used, the covers will be glossy, the size will be reduced to 9.5 by 12.5 inches, and it will be bound into square flat spines. In the "I thought everyone already knew department...", Computer Shopper will list your user group and BBS if you will send in a post card with the pertinent info. No Charge. Northcoast is the ONLY local group who has done this.

Found out why Craig Miller, Super 99 Monthly and Terrie Masters all left Compuserve for Genie. They were upset because Compuserve allowed 3 fairware track copiers to be placed in its library for downloading. These track copiers are billed by their authors as being able to copy ANY protected program. They are in our library. Some will only work with the TI-Controller and there is a CorComp version of one. We trust you will use them in a responsible manner.

Version 2.0 of PRBASE has been released and since I paid my dues on this one, the update have been ordered. Mr. Warren states that version 2.0 includes double sided disk handling

(710) records, 132 column tabular reports, storage and entry of printer control characters, 1 or 2 across mailing labels, selective indexing (for subsorting). R allows you to type in an information line at the second line of your report in addition to the original report features. You can save a screen in Edit or Add by pressing PROCD without entering each field. CREATE has been almost totally rewritten. You can initialize data disks with the program and the Report/Label Design section has been totally reworked to allow menu selection and editing of old reports. It will also attempt to build new reports for you. CREATE also allows you to set up and save printer codes, and easily allows you to change things like database name or output device.

Before we look inside, the following articles did not make the newsletter, but have been put on a general information disk in the library for your perusal...New Chip, the 9999/BS; Modular Programming Techniques; Special TI-Writer Printing Commands; Myarc's description of the new computer;

How to Speed Up things; Different ways to clear the screen; What is a Nibble; How to center headings and footings in TI-Writer; Using a Plotter; Have you used "POS" lately; Taking Advantage of Xmodem; and an Introduction to GPL Code. Since these didn't make the newsletter, the ones that did should be goodies.

Look inside for...A review by Steve Weinkamer of Multiplan Made Easy; Want to Start a Telecom SIG?; Learn how to use a BBS at Elmo Iacobucci's; add DM1000 to your GRAM KRACKER; automate your call-ups with Fast-Term; learn to input the same program in 3 languages..Extended Basic, c99 and Forth; Build a Mini-PE box from a disk drive enclosure; Use the Replace String command to avoid retyping repetitive words; Why Buy a Gram Kracker?; How to center expanded, condensed, etc., in TI-Writer; Compare 4A/Talk, Fast-Term and TE-II; Review of Compu's TI-Collection - Volume 2; Foolproof your TI Basic Programs; Use assembly language to speed up speech routines; Disk Drive Troubleshooting; learn about the incredible RLE graphics.

EXECUTIVE NOTES SOLO

Our last meeting, held on July 12, was attended by only 8 members. This is a rather dismal turnout, representing only about one third of the current membership number. This means that for two consecutive months that our turnout has totaled 17 attendees. In light of this, I will be sending letters in the next month or so to all of our friends that we have missed recently. (I realize that summertime is not the best time to spend at the computer, so some forgiveness is in order.) If you have not been paying close attention to the T.I. world lately, you've been missing out on a great deal of activity. My letter will explain the details more fully.

It was decided that due to the fact that our treasury has not grown at all lately, and that membership is still tapering off, that as of January 1, 1987, dues will go up from \$12.00 to \$15.00 per year. This breaks down to a sum of \$1.25 per month. Still one of the best bargains around!

Discussion also covered recruitment of new members. Several members mentioned the fact that fellow employees have T.I. systems, but seemingly have no interest in joining a user's group. Therefore, I will put together a letter of invitation to perspective group members and make it available to anyone who knows someone who fits that category. If you know someone who is a "closet" 99/4A user, please let us know so we can send a letter to them. We also discussed the possibility of putting notices on local cable T.V. and in the free listings of community newspapers. Mark Woodward has promised to look into this.

Our program last month was presented by yours truly, and covered several utilities available for use with T.I. Writer. Items covered were the two column, condensed printing format of this newsletter; one of several sort routines for names, lists, or what have you; and a review of the transliteration commands. There a quite a few

neat tricks that can be performed with T.I. Writer, and hopefully everyone learned a few from my demo. September's presentation will be given by Bruce Young.

Please note once again that there will be no meeting in August, and our next get together will be on September 13.

THE QUICKIE REVIEW CORNER

This month I will deviate from the usual program review and give a short book report instead. (Just like back in school.) For all of you Multiplan users, there is a great tutorial called Multiplan Made Easy, by Walter A. Ettl. (Osborne McGraw Hill publishers, 2600 Tenth Street, Berkeley, California 94710. I got my copy at Walden's at Randall Mall.)

The book is a very complete, easy to follow guide to the workings of Multiplan. Although it is written around IBM PC DOS 2.1, it is easily used with the 99/4A. The book starts out with the basics of the Multiplan screen, cell formatting, data entry, and the various commands. It then proceeds with formulas, worksheet printing and organization, and mathematical functions. Throughout the book, there are many sample spreadsheets to work on along the way, and at the end of each chapter there are exercises to do on your own.

One of the things that I like best about the book is that it goes into great depth with each explanation of Multiplan's functions and commands, and gives fine examples of each one. To me this makes it a better reference guide in many respects than the T.I. version that came with the program. The appendix contains a complete list of the commands, functions, arguments, operators, and error messages encountered. A good investment for \$12.95. Highly recommended. Rated A+.

See all of you September 13.

Steve Weinkamer

EXECUTIVE NOTES - NORTHCOAST

The July 19 meeting was attended by the dedicated members who would rather be at a meeting than basking in the sun. We also had guests whom we hope will join. We still need someone to take minutes for one or two more meetings.

I demonstrated PC-KEYS which is a resident program using function keys to catalog programs, list, load, etc. You can also create your own functions and save them for future use. I also demonstrated the track copiers which will copy most protected programs.

Earl Blewitt demonstrated Business Graphs 99 which creates bar and pie charts which can be printed, or slides could be made from the screen for projection. Data can be stored for future use for all except the pie charts. The program was user friendly and a good business aid.

Next month Music Synthesizer and Monopoly will be demonstrated.

Keep cool. See you at the August 16 meeting, Euclid Square Mall, 1:30 p.m.

RON MINADEO

EXECUTIVE NOTES - TI CHIPS

In July, we had our annual picnic in place of the regular meeting. Unfortunately, attendance was very poor. This may have been due to the wrong date being printed in last month's newsletter. To the best of our knowledge, all members were notified of the error. In any case, we were disappointed in the small number who did show.

The August meeting will be on the 16th of the month.

As President of the TI Chips group, I regret to inform everyone of my resignation. As many of you know, I am only 18, and I will be leaving for college this fall. It will therefore be impossible for me to attend the meetings. Current Vice President, Russ Shimandle, will be taking my place. I want to say that I enjoyed my short term as president of the group. The one thing that was a constant disappointment to me was the lack of participation. Everyone should try to do something to keep the club going. I wish you all the best of luck and happy computing!

RICH POLIVKA

ASSEMBLY NOTES

Tom Nellis

There will not be a meeting the month of August. The next meeting will be Sept. 13th at 7:30 PM. July's meeting was canceled at the last moment due to a scheduling conflict on my part. If I inconvenienced some of you I'm sorry.

If you have the tape tutorial, review the second installment. This is the point that we will begin with in September.

HOW ABOUT A TELE-COM SIG?

First does everyone know what a SIG is? SIG stands for Special Interest Group. A Tele-com SIG would serve several purposes: 1. Educational: As a group we could practice using some of the special terminal emulator programs such as MASS TRANSFER, TE4th, and tricks of the trade. Learn which

modems would suit our purpose the best, etc.

The second major benefit would be that as a group we could join and access the databases such as Comuserve and Genie. If you are like me and have stayed away from this aspect of your computer due to the cost, this might be just the answer. We would share the cost of the time spent on these databases and also the programs and information downloaded. There is also a lot of good information available on TI-oriented BBS around the country and we could split the long distance costs on these also. I know 3 people who have 1200-baud modems who would be willing to host such a group.

Thirdly, you would not even HAVE to own a modem to be involved and could still benefit from the programs and information taken from these boards and BBS. If you are interested, give me a call at 333-5986 and we will see if we can get something going for the Fall.

DEANNA

LEARN HOW TO USE A BBS (BULLETIN BOARD)

In response to a request by members at the June Northcoast meeting a seminar will be held at Elmo Iacobucci's, 2161 Pine Ridge Dr., Wickliffe, on Thursday, August 21, at 7:30p.m. Members of all the groups are invited to attend as the new TIBBS board will be demonstrated. Also, this would be a great opportunity for you to learn all about a BBS or input what you would like to see on the board if you are a regular user.

For those who are unfamiliar with BBS', they are simply another way to exchange information. You will probably need a complete system with disk drive as all of the terminal emulators except the TEII are on disk. That, an RS232 interface, a modem and a telephone and you are in business.

If you are a casual user, a 300-baud will work fine and be much cheaper. Just now the prices are starting to fall on the 1200-baud. Some local boards already only accept 1200 baud and if you join one of the commercial data bases, you get 4 times the information for your money as with the 300 baud.

The heart and soul of a BBS is the message board. If you are looking for a certain piece of hardware or software, post a message; want to sell something, heard of a good buy, announce a meeting...the possibilities are endless. I know of one person who plays chess. Since getting my modem late this past winter, I have become one of the more regular users of the Northcoast BBS. What fun to have mail left for you! I try to post tidbits from other clubs newsletters as soon as I get them and those using the bulletin board get this information 2 to 3 weeks before the newsletter is published.

Another part some people like on a Bulletin Board is the ability to download programs. We are trying to put on programs which have not yet hit the library and at the moment only members from Northcoast have access to them via the board. However, there has been talk of a small registration fee for the board which would allow others downloading privileges also. MOST of the clubs' newsletter is on disk and could be put on the BB for downloading almost a week before you get it in the mail. This meeting would be a good time to express your opinion on this matter and give Northcoast's Board a direction in handling this proposition.

Some people may not realize that you are not restricted

to only calling TI Boards. You can call Atari, IBM, Commodore, etc. However, you probably won't find any information you can use for the TI, nor would it do you any good to download programs.

With the placement of the TIBBS BBS in public domain, you will probably see a proliferation of BBS' run by individual members of groups as it was originally designed to run with a minium configuration of 32K memory and a single-sided drive. In fact a scenario in another newsletter envisioned that every member in the club would have his own BBS and that is the way we would communicate exclusively. So, if you think you would like to start your own BBS, come to the meeting at Elmo's.

There is a service which is not yet available in the Cleveland area, where for \$25 per month, flat rate, you can have unlimited calls to BBS systems throughout the country. You are limited to 1 hour at a time on-line, but can call as many times as you wish anywhere. When and if we hear this becomes available here, will let you know.

The following were taken from recent newsletters. If there is enough interest, will publish more as we come across them.

COLORADO

RM 99ERS BBS	303/423-2654	PAUL CONSTANTINE
ADS UP BBS	303/444-6129	DR. RON
THE STAR BOARD BBS	303/455-3113	MIKE HOLMES
TIBBS	303/574-5762	
VILLA-TI	303-574-2567	300/1200

FLORIDA

T.I.T.S.	305/889-3687	BLANE RICHARD
LOWE WOLF	305/894-9641	WILLIAM BYRD
COMM-LINK	305/682-1526	RICK GREGORY
NIGHT TRAIN	305/877-6546	DENNIS NEUBAUER

CHEERS 305/682-3701 ERIC MINDOCK

GEORGIA

ATLANTA	404/366-1914	CHARLES DUPREE
ATLANTA	404/991-6350	JIM FAIRCHILD

KENTUCKY

LXINGTON	606/266-9552	BERRY MILLER
----------	--------------	--------------

MASSACHUSETTS

BOSTON	617/331-4181	
BOSTON	617/335-8475	

OHIO

TICOMM (TOLEDO)	419/385-7484	BUD MILLS (24 HRS)
YOUNGSTOWN	216/755-8220	ED LUPTAK
TIBBS	614/451-0880	
TIABS	614/442-1852	M-TH 8p-6a. F 8p-M 8a
CINCINNATI	513/474-4620	JACK PORTER

TEXAS

LEAGUE CITY (JUG)	713/337-4128	
-------------------	--------------	--

WASHINGTON (All say not be TI)

PUGET SOUND 99ERS	206/784-4142	
TIBBS NW	206/641-5884	ED DURFEE
ABERDEEN	206/533-0941	
TELETRAVEL	206/742-9034	
BREMERTON	206/377-1845	
LYNWOOD	206/775-3660	MIKE ALLEN
QUEEN ANNE COMPUTER		
SHOPPE (QACS)	206/283-0953	
KJR RADIO BBS	206/878-0158	

DEANNA

ADDING DM1000 TO THE GRAN KRACKER BY KEVIN KAPUS, FRONT RANGER, JULY, 1986

The basic idea behind this article came from the K-Town 99er newsletter. I have made it easier.

If you wish to have DM1000, Extended Basic and the Editor Assembler on the main menu screen, first use load module and load E/A; then load E/A-MOVER from the GK disk. Move the E/A to Gram 7 and follow the directions on the screen. Next, load in Extended Basic and IBCALLS, then use the save module and save what you have just loaded. The module you have just saved will contain both XB and E/A so that you only have to load in one item to get both modules.

Now, on to adding DM1000:

First, load DM1000 from XB or E/A. As soon as it is loaded, hit the reset on the Gran Kracker, then go into the Gran Kracker and select option five, Edit Memory. Turn off the loader and the write protect, select Grams 1 and 2, enter the start address of A000, the end address of BAF2 and the destination of g2056. Use function 2 of the editor to move DM1000 from the 32K to the Gran Kracker.

With the loader and write protect off, select g2000, then function 9 to get into the memory window. Then type in

the following hexadecimal code into the specific memory addresses.

```
>2000 AA 01 01 00 00 00 20 10 00 00 00
>200C 00 00 00 00 00 00 20 26 11 44 49 53
>2018 4B 20 4D 41 4E 41 47 45 52 20 31 30
>2024 30 30 31 3A F2 BF 1D 00 20 56 BF 00
>2030 A0 00 BE BF 01 00 9F 31 00 08 AB F0
>203C 20 4E BF 4A 09 00 06 00 18 BF 4A 0B
>2048 00 06 00 4A 0F F0 78 78 78 78 78
>2054 7B 78 04 60 C2 74 0F 80 00 AB 0F 89
```

(NOTE: When copying these, I had a terrible time distinguishing "8's from "B's" and hope I got them right. Deanna)

Turn the write protect bank on, push reset and the space bar. You should now have DM1000, XB and E/A on your main menu screen. If you press DM1000, it takes approximately 2.3 seconds to load, much faster than loading it off a disk. Also, it saves ear and tear on your drives. By the way, be sure to save Grams 1 and 2. If you should have a problem that wipes out these Grams, you would be able to reload DM1000 without having to make any modifications.

X.

DISK DRIVE TROUBLESHOOTING
BY JIM MEKEEL, NORTHCOAST 99ERS

Q: My disk drive is not reading every time. How do I check it out?

A: Plug in the Disk Manager 2 module and run the comprehensive test using a blank disk since this test is destructive. This test checks the disk, controller and drive. One loop takes 15 minutes for single-sided and 28 minutes for double-sided drives. Each test loop runs six test that reads and writes to all sectors on the disk.

Q: Does this test check for head alignment?

A: No, it doesn't. If the drive passes all the above tests but still does not read correctly, then the most likely problem is the mis-alignment of the head(s). To test for this, write to a disk using a known good drive. Then take the disk, place it in the suspected bad drive and attempt to read from it. If the head continues to search for the file (searching is detected by hearing the head move back and forth without stopping), then you most likely have an alignment problem.

Q: Can I realign the head myself?

A: Yes, but you need to have an oscilloscope and the service manual for the drive. The procedure is usually simple and takes a short amount of time, but each manufacturer has a different set of instructions. The service manuals tell you that a test disk by the manufacturer must be used, but you can substitute a full disk that was written by a known good drive.

Q: What should I do if the drive does not pass the tests made by the Disk Manager 2 module?

A: The very first thing to do is to clean the heads. There are commercial head cleaning kits available, but I prefer the more direct route of cleaning the heads with a Q-tip and alcohol. To do this, you must turn off the power, disconnect the cables and remove the drive. Usually the drive has some internal covers that must also be removed. The head(s) are on hinged arms that are spring loaded. These arms can be gently lifted to expose the surface of the head. Using a Q-tip dipped in rubbing alcohol, gently wipe the head clean.

While you are in the drive, check the lubrication on all the moving parts. If the parts are plastic or nylon, do not use any lubrication as it might actually cause the parts to become "gummy". If the parts are all metal, then there should be a light film of oil on the surfaces. Use a light weight oil in very small amounts.

If you have a degausser (demagnetizer) for your tape deck, this can also be used the same way on the disk drive heads.

Reassemble the drive, install it in the system and run the tests again.

Q: If this doesn't solve the problem, what do I do next?

A: Pray that you win the lottery. Seriously, if you have reached this point and still have a problem, I suggest you check all the connections. If these prove to be all right, then you should substitute a known good drive for the suspect drive. If the new drive fails, then the problem is in the connections, disk controller card or the power supply. If the new drive works, then the problem is in the suspect drive. Take the bad drive to a Pro or obtain a service manual and troubleshoot the problem yourself.

GREAT GRAPHICS...WITH RLE
By Dave Talan, Northcoast 99ers

There is a new program which allows the TI to display excellent graphics resolution. No, the program isn't like TI Artist, or any other drawing program. As a matter of fact...its not for drawing, just for displaying. The program is "TI-RLE." RLE stands for "Run Length Encoded" graphics. RLE's are files located in various locations throughout Compu-Serve which contain encoded representations of high resolution images displayable on the screen of the user's computer. RLE is mostly intended for users operating with Compu-Serve's "VIDTEX" terminal emulator. However, thanks to Travis Matford, this is available for the TI!

His program, "TI-RLE" is used in conjunction with these RLE files which can be downloaded from Compu-Serve. If you are a member of Compu-Serve, these files can be found in the IFORUM Data Libraries, the PICSI 5, ARTFORUM, and the CB simulator area.

TI-RLE was developed with Paul Charlton's Fast-Term terminal emulator as the emulator in capturing files and is

currently suggested for use in downloading RLE files. For specific download instructions consult the Fast Term documentation or find more info on Compu-Serve by typing: 60 TINEWS at the main function prompt. The RLE files are simply incredible! Any thing from Khadafy to NC Fields and Mae West. These can be printed out or they can be viewed on the screen. The program requires one disk drive, the E/A cartridge, and a printer (optional).

The feature that impressed me was the fact that these files could be saved to be incorporated for GRAPHX, which in turn could be converted to TI-ARTIST...and so on!

So, as one can see this is a very versatile program for the graphics lover. Furthermore...there are always new pictures being developed, maybe some day on the TI. This program is Public Domain and can be downloaded from Compu-Serve or you can get it from the North Coast library.

For more information on this fine program contact Dave Talan at (216)/333-5829.

(Library note: 2 full disks of these graphics have been requested from one of our sister clubs and should be here by the next meeting).

SEE EXAMPLES ELSEWHERE IN NEWSLETTER!

Foolproofing TI BASIC programs usually involves compensating for erroneous input, controlling graphics, checking for incorrect values for computations, and anticipating I/O errors.

INPUT problems:

Using an INPUT statement is a reasonable and common method of accepting information from a user while a program is running. It has some limitations in that there is no way to control what information is being entered until it is too late. If some incorrect information is entered, a warning will be displayed and any graphics or text on the screen will be scrolled upward, thus producing a tremendous mess. A more tedious, but much more controllable method is to use the CALL KEY statement.

The CALL KEY statement allows the return of only one key at a time. Therefore, the program can check to see if the input is acceptable. Here is an example:

```
10 PRINT "do you want to continue?"
20 CALL SOUND(140,1400,4)
30 CALL KEY(5,N,S)
40 IF S<1 THEN 30
50 IF (N=89)+(N=121) THEN 100
60 IF (N<>78)*(N<>110) THEN 20
70 CALL CLEAR
80 PRINT "Thank you for your time!"
90 STOP
100 "Now we can continue..."
110 $$$the rest of the program$$$
```

LINE 20 produces the standard beeping tone like an INPUT statement.

LINE 30 scans the whole keyboard (unit 5) and returns the key number in the variable "N" and the keyboard status in the variable "S"

LINE 40 will transfer control back to line 30 if a "new" key is not being pressed (same key as before or no key at all).

LINE 50 checks for the "Y" (key 89) or the "y" (key 121) using a boolean "OR"

LINE 60 checks for the "N" (key 78) and the "n" (key 110) using a boolean "AND"

LINES 70-90 will be reached if the answer is no.

LINES 100 to the end of the program will be reached if the answer is yes.

This program will ONLY allow the letters N and Y (upper or lowercase) to be entered. All other entries will branch back to the CALL SOUND, indicating the input was ignored.

Using the POS function can allow quick scan for several options. An example would be:

```
10 A$="ABWGHULNSQTVZ"
20 CALL KEY(T,N,S)
30 IF S=0 THEN 20
40 ON POS(A$,CHR$(N),1) GOTO
```

In line 40 the "+1" allows a line number to be assigned for the possibility of a selection not in the list of options. In this case, it nearly loops back to the CALL statement for a (presumably) correct input.

This way, some odd string of possible options can easily be chosen from. The ON GOTO statement is quite useful in this application since the line numbers can be ordered irrespective of the alphabetic or other order the choices might be in.

Checking for out of bounds numbers as relates to computations or graphics is similar to checking for CALL KEY values:

```
10 CALL CLEAR
20 ROW=12
30 COL=16
40 CALL HCHAR(ROW,COL,42)
50 CALL JOYST(1,H,V)
60 CALL HCHAR(ROW,COL,32)
70 ROW=ROW-V/4
80 IF ROW<24 THEN 100
90 ROW=1
100 IF ROW>0 THEN 120
110 ROW=24
120 COL=COL+H/4
130 IF COL<32 THEN 150
140 COL=1
150 IF COL>0 THEN 40
160 COL=32
GOTO 40
```

In this example, joysticks are used as input to control an object moving around the screen. The IF THEN statements check for the vertical and horizontal boundaries BEFORE the object actually gets there. This way one cannot accidentally move off the screen. An alternative to resetting the variables to the values at the other ends of the screen would be to reset them at the previous position so the object "stops" at the edge of the screen. In either case, an error has been avoided.

A similar method would be used to check for maxima and minima when using limited mathematical calculations.

When it comes to I/O, things get a little more difficult. One way to avoid trouble is to OPEN files using parameters which virtually all devices can use. Since printers and cassetts are probably the most limited, finding common ground among them should help.

```
10 A$="CS1CS2RS232RS232/1RS232/ZPIODSK1DSK2DSK3"
20 INPUT "Device name: ":D$
30 IF POS(D$,".",1)>0 THEN 60
40 C$=D$
50 GOTO 70
60 C$=SE6$(D$,1,POS(D$,".",1)-1)
70 IF POS(A$,C$,1)>0 THEN 100
80 PRINT "Not a valid device.":"Try again."
90 GOTO 20
```

```
100 OPEN #1:D$,DISPLAY,OUTPUT,FIXED 80
110 $$$use the file now$$$
```

The string A\$ in line 10 contains a list of allowable device names.

LINE 30 finds out whether the name contains other parameters of a file name.

LINES 40 and 60 assign the device name portion of the string.

LINE 70 searches a\$ for the device name, and branches if found.

LINE 100 opens a file with generally acceptable format to most devices.

FOOLPROOFING YOUR EXTENDED BASIC PROGRAMS

Foolproofing TI Extended BASIC programs is somewhat easier than in TI BASIC. A number of statements allow for even the most inept of users to proceed error-free.

One of the most effective ways to foolproof programs is to use the VALIDATE and SIZE clauses in the ACCEPT statement. VALIDATE allows only specified characters to be entered at all. SIZE fixes the width of input and allows a "default value" to appear in the field so one only need press ENTER to use it (as is so common in many cartridges).

The ON BREAK NEXT statement allows the program to make the program unstoppable except through normal channels allowed in the program (this excludes QUIT, unfortunately). Also, the ON ERROR statement allows one to either ignore or deal with normally "fatal" errors such as string-number mismatches, I/O errors, bad values, etc., while the ON WARNING statement allows control of too-big numbers and input errors which are not "fatal", but are annoying anyway, and can thus be avoided.

The following program demonstrates many of these features.

```
100 ! CASH REGISTER PROGRAM
110 ! BY RICK LEWIS
120 !
130 ! VERSION 1.2
140 CALL CLEAR :: CALL COLOR(0,7,7):: CALL SCREEN(6)::
    FOR X=1 TO 9 :: CALL COLOR(X,2,15):: NEXT X ::
    A$="#####"
150 ON WARNING NEXT :: ON BREAK NEXT
160 PRINT "TAX RATE=";:: ACCEPT AT(24,22)VALIDATE
    (NUMERIC)BEEP:TAXRATE :: CALL CLEAR
170 CALL CHAR(100,"FF"):: CALL HCHAR(1,1,100,32)::
    CALL HCHAR(5,1,100,32)
180 DISPLAY AT(2,8):"ADDING MACHINE":TAB(10);"RICK
    LEWIS"
190 DISPLAY AT(9,1):"NUM QTY EACH EXTENDED" :: CALL
    HCHAR(8,1,100,32)
200 FOR X=1 TO 10
210 DISPLAY AT(6,1):"QUANTITY OF ITEMS"
220 DISPLAY AT(9+X,1)SIZE(3):X
230 ON ERROR 400
240 ACCEPT AT(9+X,6)SIZE(-3)BEEP VALIDATE(DIGIT):Q$ ::
```

```
IF Q$="" THEN 240 ELSE IF Q$="0" THEN 300 ELSE
Q=VAL(Q$)
250 DISPLAY AT(6,1):"COST PER UNIT"
260 ON ERROR 410
270 ACCEPT AT(9+X,11)SIZE(-7)VALIDATE(NUMERIC)BEEP:C$
:: IF C$="" THEN 270 ELSE C=VAL(C$)
280 DISPLAY AT(6,1):
290 DISPLAY AT(9+X,1):USING "## ###.##"
###.##":X,Q,C,Q$C :: TOT=TOT+Q$C :: DISPLAY
AS(22,16):USING "SUM "%A$:TOT :: NEXT X
300 DISPLAY AT(6,1): : : : : : : : : : : : : : :
310 TAX=TOT*TAXRATE/100 :: CALL HCHAR(21,23,100,8)::
    DISPLAY AT(23,16):"TAX" :: DISPLAY
    AT(23,21):USING A$:TAX
320 DISPLAY AT(6,1):USING "TAXRATE=###.##":TAXRATE
330 DISPLAY AT(24,15):"TOTAL" :: DISPLAY AT(24,21):
    USING A$:TAX+TOT
340 DISPLAY AT(22,1)SIZE(7):"YES, NO" :: DISPLAY AT
    (23,1)SIZE(7):"OR REDO"
350 DISPLAY AT(24,1)SIZE(10):"ANDTHER? "Y" :: ACCEPT
    AT(24,10)SIZE(-1)VALIDATE("YNR")BEEP:Y$
360 IF Y$="" THEN 350
370 IF Y$="N" THEN CALL CLEAR :: STOP
380 IF Y$="R" THEN TOT=0 :: DISPLAY AT(23,10): : : : :
    GOTO 200
390 TOT=0 :: DISPLAY AT(10,1): : : : : : : : : : : : :
    : : : : : : : : : : : : : : : : : : : : : : : : : : :
400 RETURN 230
410 RETURN 260
```

Explanation of Cash Register Program

To operate the program, one must first enter the tax rate in %. The program will only accept valid numeric data due to line 150.

LINE 190 starts the main loop allowing up to 10 pieces of data.

LINE 220 directs the computer to the appropriate line number in case of error. In this case it merely comes back, ignoring the error and waits for another (correct) input for the next line.

LINE 230 allows only DIGIT input, assuming no fractional quantities. Entering quantity of zero tells the computer the last data has been entered, and it skips to the totaling section (line 230 goes to line 290).

LINE 260 accepts cost information.

LINE 280 displays the information in a specific format, but all decimals are retained internally for tax and total computations.

LINE 290 erases the unused portion of the screen in case there was previous information there.

LINE 340 waits for a "Y", a "N", or an "R" (capitals only). "Y" will clear the workspace and start over on a new problem. "N" will clear the display and stop the program. "R" will allow recalculation of the problem using the previous values as default values.

LINE 380 is a faster way to clear a portion of the screen than using the CALL HCHAR statement.

TURBO SPEECH
By Stephen Shaw

Excerpted from the TI99/4A Exchange TIMES of Great Britain, Issue #6, Autumn, 1984, via CALL SOUNDS, June, 1986.

This information will give you speech in TI Basic with the Minimemory or if you have XB+32K, will give you speech faster than CALL SAY which slows programs down no end. (Should also work with TI Basic and Ed Assem or Super Space or Super Card.-ed). For this information, I am indebted to Neil Lawson who has been delving.

Program framework for timing purposes:

```
20 CALL INIT
30 S=-27648
100 FOR I=1000 TO 100 STEP -100
PRINT I/10 (COUNTDOWN MOD BY ED)
104 NEXT I
110 PRINT "START"
120 FOR X=1 TO 20
130 REM INSERT TEST ROUTINE
140 FOR Y=1 TO 30
150 PRINT ">";
160 NEXT Y
170 NEXT X
180 PRINT "END....."
```

This standard routine sets up a framework to test the new routine and gives a basic time reference. Note: times quoted are for my system. Yours may be different, but the ratios should be similar.

Running the above program with the loop in line 140 running 30 times as shown, takes 10.7 seconds from "START" to "END". Change line 140 to loop just 20 times and the timing is 12.7 seconds. Now we can insert 2 possibilities. The first is available only in XB:

```
130 CALL SAY("THAT IS CORRECT")
```

Run the program again. If line 140 is looped 20 times, the time is 44 seconds. If line 140 is looped 30 times, time is 50 seconds, a 6 second difference.

The time for speech is constant, it adds about 21 seconds to the program. Now for something different, (also works with minimemory)!!!!:

```
130 CALL LOAD(S,70,"",S,65,"",S,72,"",S,70,"",S,64,"",S,80)
```

If you now run the program, it says the same thing as many times, but look at the timing! If line 140 loops 20 times, it takes 26.3 seconds, looped 30 times takes 26.5 seconds. We know that looping 140 an extra 10 times adds 6 seconds. Where have those 6 seconds gone?

The CALL SAY routine holds everything up until it finished speaking, but using the CALL LOAD equivalent, while the computer is speaking, it gets on with the next chorus. The "dead time" is used and soaks up those 6 seconds. Thus, using the CALL LOAD equivalent, the computer speaks faster and also permits your program to run more quickly if there is work for it to do between speech outputs. That's the clever demonstration! (Impressed?) Now for the theory!!

References: Ed/Assem Manual: pages 351,355,422,427. (erata: the reference in para. 1 note, page 355, should be to section 22.1.4, not as printed in the manual).

Address -27648 is the SPEECH WRITE address. We keep feeding it with bytes, and in due course the computer speaks!

The bytes to load to that address are found out as follows: First, decide what you want to say from the standard vocabulary. Then look in the table (pp 422-427) for the address of that word or phrase. "THAT IS CORRECT" is given as 6B16. That is Hexadecimal, not a decimal number. The four numbers are reversed and become 6186. Now we offset them by Hex 40 and feed them in. As we are dealing with decimals with our CALL LOAD, that means we ADD decimal 64 to each digit in turn:

(6+64)	(1+64)	(8+64)	(6+64)
70	65	72	70

If the numbers were Hex A-F, these have a decimal value as follows: A=10,B=11,C=12,D=13,E=14,F=15.

Now we must indicate end of word by loading a zero again offset: Thus, 0+64=64. Finally, we must instruct the computer to speak by loading Hex 50 (decimal 80). Thus have loaded, in order, 70,65,72,64 and 80.

Check back to the listing. NOTE the way CALL LOAD has been used: a single command to load the same address with several different values.

To assist your experimentation, here are some HEX addresses from the Manual. Remember to reverse them, translate to decimal and offset!

TEXAS INSTRUMENTS...6696	THAT IS RIGHT..68FE
WHAT WAS THAT.....77E9	READY TO START.56B3
YOU WIN.....7DDB	AGAIN.....17A5
ANSWER.....1913	CHECK.....1DB2
CHOICE.....1DA2	COMMAND.....1F1A
ELSE.....28B6	GOODBYE.....3148
HELP.....3571	HURRY.....3757
INSTRUCTIONS.....39BD	I.....3793
I WIN.....37CF	JOYSTICK.....3AED
NAME.....47C0	NICE TRY.....49A5

This is not only a useful programming aid in its own right, but also by demonstrating a part of the Editor Assembler manual's sometimes complex instructions, it should assist you when you are ready to move on to Forth or Assembly language proper.

FOR SALE

GORILLA/BANANNA dot matrix printer - \$60 or best offer.
Rich Polivka 238-3971

SEMINI 10X PRINTER with cable and paper - \$200 or best offer.
DISK DRIVE, DISK CONTROLLER CARD, RS232 CABLE, EXPANSION BOX
\$275 or best offer. Together or separate. Joe Lorenz 842-3476.

EXTENDED BASIC - \$40

3 INFOCON GAMES -

Deadline, Sourcerer, Hitchhikers Guide - \$10 each

BEST OF 99ER - 10.00

NYCOVE FORTH - \$15.00

MARK X SIGNALMAN 300-BAUD MODEM, AUTO ANSWER, W/CABLE

Larry Prikockis - 256-1516 or leave message on
NorthCoast's BBS

TELECOM SIG
 Choosing a Terminal Emulator Program
 From the CALL SOUNDS, June, 1986

COMPARING 4A/TALK, FAST-TERM and TE-II

We are fortunate in having 2 excellent terminal emulator programs: The fairware FAST-TERM and commercial 4A/TALK.

The prices are comparable. Paul Charlton, the author of Fast-Term asks \$15, but you supply the disk and must print out your own instructions. 4A/Talk costs \$19.95, comes with a clear and professionally printed booklet, and, of course, the disk is supplied. 4A/Talk has some rather sophisticated protection on the disk, but the publisher, DataBioTics, will replace the disk if anything goes wrong. Either program will do the job for you; however, a little comparison cannot hurt.

What about TE-II?? If you have only cassette storage and a stand-alone RS232, this module is the way you are forced to go. It will let you log on to BBS's and information services, exchange information with other TI computers, but its capacities are limited. It will, however, access the speech synthesizer which no other emulator is able to do. (7)

FEATURE	TERM	TALK	TE-II
Capture buffer dump to disk	9.5K	8K	no
Printer spooler, print while reading	8K	8K	no
Disk Directory	yes	yes	no
HELP screen	no	yes	no

Use without smart modem	yes	yes	yes
Keyboard dial w/smart modem	(1) yes	yes	no
Auto dial w/smart modem from list	no	yes	no
Send message prepared in advance	(2) yes	yes	no
ASCII text capture	(3) yes	yes	yes
TE II protocol	yes	yes	yes
XMODEM protocol	yes	yes	no
Configure default file	(4) yes	yes	no
Change configure while running	(5) yes	yes	no
On screen call timer	(6) yes	no	no
Speak the screen content	no	no	yes

(1) FT would only dial out on Full Duplex, 4A/T on both half and full.

(2) My personal preference was for FT's way of handling this feature.

(3) TE-II will only screen dump, 1 screen at a time.

(4) With FT you can set up a separate configure file for each host or BBS simply by giving in a different file name. 4A/Talk could only use a file named CONFIGURE, and you had to call up the configure screen and redo it if a BBS called for different defaults. FT is better here, by far.

(5) 4A/Talk recalls a whole screen of defaults, FT can change one default at a time with a combo of key presses.

(6) FT's on-screen timer stops, for obvious reasons, while reading from or writing to disk. Timer is a guide more than an accurate clock.

(7) Bill Wright, Sysop of TURBO TECHIE BBS in New Jersey, tells us he has four or five "regulars" accessing his BBS with TE-II and Cassette.

BOOK REVIEW: COMPUTE'S TI COLLECTION - VOLUME 2
 BY DAN BOYCE, MADHUG, JULY, 1986

This is COMPUTE'S most recent for the TI. I have had it for a couple of months and am still working my way through it. All the programs and articles are new and have NOT appeared in COMPUTE magazine. Here is what it covers:

CHAPTER 1: Applications and Techniques

- MitiCalc
- Sketchpad
- Memo: The Simple Word Processor for the TI
- TI Screen Dump
- I/O Through the Joystick Port
- Record Blocking Techniques
- AI with TI (Artificial Intelligence)

CHAPTER 2: Fun and Games

- Labyrinth
- Spitfire
- BOG'L
- Speed Demon
- Blackjack
- Flood Waters
- Macro/Micro Maze
- Mad Hatter Ladder

CHAPTER 3: Inside the TI

- TI Memory Organization
- The Heart of the TI 99/4A: The TMS9900

Microprocessor

- TI Character Graphics
- TI 99/4A Character Definitions

CHAPTER 4: It's Educational

- Oh, So Simple
- Happy Face Arithmetic
- Spelling Tutor

CHAPTER 5: Assembly Language

- TI 99/4A Memory and Assembly Language
- Tips
- TI FastSearch
- Expand TI BASIC with Mini Memory
- TI File Management

This book leans more heavily to the technical and assembly language material. The program article with which I am playing is the one to use the Mini Memory to load assembly language utility programs from disk in BASIC. The best news is that COMPUTE has this available ON DISK for \$12.95 plus \$2 for shipping and handling. This is one I plan to get as I am too busy (lazy) to type in the programs. Maybe we will get lucky and COMPUTE will put all their other books on disk. (ED Note. COMPUTE is really keeping this existence of this book a secret. I have most of the books they have printed, subscribe to their magazine, and did not know this was out. However, Waldens will order for you and get it within a week or so and not have to pay shipping charges.)

WHY GRAM KRACKER?
BY HOWIE ROSENBERG
FROM THE BREAD BOARD, MAY, 1986

Some of you may be wondering why there is so much excitement in the 99/4A world based on Millers Graphics delivering the first Gram Krackers last week. Of course, those who own Gram Krackers can now save all of their modules to disk and never worry again about desirable modules disappearing from the marketplace. Considering that most modules are still available, I don't feel that backing up modules or even acquiring all those that you do not presently own is sufficient justification for buying a Gram Kracker. Yet I feel that Gram Kracker is one of the most significant TI-994/A products to ever come to market. The reasons are well understood by old timers, programmers and such, but may escape some novices and users who have never been concerned about the innards of the machine. This note is addressed to these folk in the hope of imparting some of the excitement that we old timers feel. The next two paragraphs are somewhat technical and could be skipped if the reader is so inclined. I suggest that you read them to really understand our excitement.

The TI-994/A computer has a 16-bit address bus. If you convert 16 binary bits to the equivalent decimal number, you will discover that the central processor unit (CPU) which is the brains of any computer, can access 64000 locations. As it turns out much of this memory does not exist in the TI console at all but rather in the peripheral expansion box (32k) and in the GROM port (8K). In fact, there are only 256 Bytes of CPU addressable RAM in the console. The 16K memory that comes with the bare console is not directly addressable by the CPU, but is a "mapped device". Most computers make use of mapping to expand memory capabilities beyond the capability of CPU address space. The mapped devices generally communicate with the CPU by

having several common addresses. One (or several) addresses tell the mapped device what address in it's space you want to communicate with. Other common addresses are used to transport data back and forth. If a device is RAM, it can both be written to and read from. If a device is ROM, it can only be read from. The TI machine appears somewhat like an octopus with arms extended to devices in all directions. In particular, the concept of GROM which is one of the mapped devices in our machine has had much to do with the machine's history, and uniqueness in the computer world.

Grom is sequentially accessed memory which has some unique characteristics relative to other areas of our machine. First, the device itself..the GROM chip is proprietary, and basically was TI's method of keeping control of their software. A good history of the early days and the consequences of TI's decision to keep the machine closed is found in "The Orphan Chronicles" (Ron Albright, published by Millers Graphics). The use of GROM in modules was intended to stop piracy and control software. The use of GROM in the console was a memory space saver, and enabled coding a large part of the operating system and console basic in GPL. Graphics Programming Language (GPL) is an assembly type language. It's characteristics are somewhat different than 9900 assembly in that it's a byte oriented language (8 bits=1 byte) as opposed to 9900 code which has both BYTE and Word (2 byte) instructions. The language appears to make widespread use of Macros (canned routines which can be CALLED in any program) and supposedly is easier to program in than 9900 code. TI never came out with a GRAM version of GROM., i.e., GRAM can be reprogrammed just as can RAM as opposed to the ROM and GROM which cannot be reprogrammed. There is one reference in the Editor Assembly manual which indicates that TI intended to release GRAM at some future time.

IN computer jargon, our machine has been described as having closed

architecture. Architecture refers to the way the electronic building blocks, CPU, RAM, ROM, and other logic elements are interconnected to form a computer. A machine with open architecture with the same addressing capability as ours would have 65K bytes of RAM addressable by CPU. Any ROM, i.e., the Operating System Code, which is the make-up routines in the machine, would be dumped to the RAM on power-up. It would be simple to change the code. It also would be simple to download any module, so TI gave us a closed machine. The GRAM KRACKER has opened it up! It is now possible to change the operating system which resides in ROM 0 because GRAM KRACKER contains a GRAM 0 which can at the flip of a switch be substituted for the console GROM. As more programmers become familiar with GPL code, more changes will become available to the TI community. We already have code to bypass the color bar screen, auto start a menu option, slash the zeros in both console and Extended Basic, among others. Similarly by the flip of another switch, GRAM 1 and 2 can be used to modify console Basic. The remaining memory in GRAM KRACKER 56K worth simulate the GROM that is in command modules and ROM which is used in a number of modules. The point is that not only can modules be copied, they can be altered. Already a number of changes have been suggested primarily by Millers Graphics, but several programmers have already entered into what will become a major overhaul. Extended Basic can now be changed to choose your own default colors, add some new commands such as CALL CLOCK, and CALL CAT among others. While writing new GPL code which is also now possible or modifying old code DP system, Basic, or modules requires some programming expertise, installing the changes in GRAM KRACKER using its editor is extremely simple. In a short time such changes as Saving Video Chess games on disk and printing out game scores, printing out sheet music from Music Maker and a variety of others will appear. Exciting times are ahead for TIers.

MINI PE BOX

FOR ALL OF YOU HARDWARE HACKERS WHO HAVE BUILT YOUR OWN POWER SUPPLIES FOR DISK DRIVES, INSTALLED QUIET FANS, IN YOUR PE BOX, HAVE 32K INSIDE YOUR COMPUTER AND 8K IN YOUR E/A, YOU MIGHT LIKE TO TRY WHAT ANDY CHERNIVSKY OF CHICAGO DID, AND THAT IS, BUILD HIS OWN PE BOX. THE FOLLOWING IS CONDENSED FROM THE TECH CORNER OF THE CHICAGO TIMES NEWSLETTER OF FEBRUARY, 1986, FROM AN ARTICLE WRITTEN BY ANDY.

With an extra TI disk controller and 32K expansion card, Andy developed the following: I was looking for an enclosure that could house one full-height drive and two TI cards. I noticed that the cards were almost the same size as a drive and wondered if they would fit into a standard enclosure, but to no avail. I decided to take the cards out of the protective covers (handle carefully!). Eureka!!!. The cards are the same size as a drive and 3 will stack in the same space as a half height drive.

A half-height drive and 2 cards, plus a cable interface card would fit comfortably in a single full-height enclosure with power supply. I decided 2 card slots would be enough. This would give me a fully-expanded system with only 2 slots.

The cards require a 60-position card edge connector. A normal surface mount plug of that type caused the card to stick up too high for the top of the enclosure to fit. I found another type of 60-position plug that would snap into ribbon cable and allow me to mount it from the under side through a rectangular

hole I had cut in the bottom. These connectors cost \$10 each and were mounted right against each other and against the left side of the enclosure. This left just the correct spacing for the cards to the enclosure side and top and to each other. The cards have 2 voltage regulators mounted on them. I mounted a small heat sink to each one to help dissipate the heat. In their TI clamshells they are clamped to the card by the clamshell corner screws and the cover itself acts as a heat sink.

To the right of the 60-pin position plugs, I mounted a 44-position card edge connector which would hold the interface board to go from the 44-position peripheral port on the console to the 60-position plugs for the cards. For the plug into the console, I used a 44-pin (position) card edge connector mated to a gutted 50-pin business telephone type connector. The 2 were joined with epoxy and the age old source of expansion irritation (that huge plug) was neatly solved. For the cable from console to box I used 40-conductor shielded, stranded computer cable. The shielding is very important as unshielded cable will cause strange patterns on the monitor. 40 wires are quite enough as you really only need 35 to do the job. I ran the cable out the front of the box, between the front feet. This way the box sits on the right side of the monitor about a foot behind the console with the cable going straight out and plugging right into the console neat as can be.

The bottom of the interface board plus into the 44-pin card edge connector which solders onto the cable which goes

to the console. At the top of the interface board is a 60-pin plug set. One side is soldered to the interface board, and the other snaps onto the 60-wire ribbon cable (more expensive plugs). The ribbon cable then goes down parallel to the interface board and snaps onto the 2 60-position plugs mounted in the case bottom.

The boards require that 3 unregulated voltages be supplied to them. Measuring my TI PE Box I got +13 (to be regulated to +5 volts on the individual cards), +22 (+12 after regulation) and -22 (-12 after regulation). When I measured my drive enclosure before the regulators I found +10 and +19 which is plenty to do the job. To get the negative voltage I added a small transformer, rectified and capacitor to the enclosure behind the transformer that was already there.

Now to the hard part. The interface board. I hand-wired the board on a piece of Radio Shack 44-position card edge finger PC board. It took 2 all-night shifts and a few daylight hours besides. The interface is basically just running the wires from the console and the cards to each other, to ground or to +5 volts with one exception; the data bus wires. They have to go through a buffer that protects some of the more sensitive chips in the console. The buffer is mounted on the interface board and the +5 volts it needs is supplied by the 5-volt regulated supply for the drive. It cost about \$60 and 30 hours exclusive of the TI cards, the drive and the enclosure. The finished product works perfectly and makes the TI an easily portable computer.

NORTHWEST DMID 99'ER NEWS, APRIL, 1986

There are some word processor programs that have a feature to reassign a key to spell an entire word or phrase. TI-Writer has a way to do this, by using one of the less commonly used characters, such as "(" (FCTN F), or ")" (FCTN 6). For example, if anytime you need to spell the word "computer" in your text, just use a "(" with your TI-Writer. After you have finished the entire letter use the "ReplaceString" command. You will be asked to use a "/". Insert the word you want to replace (old string), then you must use "/" again. Now type the word you want replaced (new string) and remember to use another "/" after typing the new string.

EXAMPLE: (NOTE: WILL NOT WORK IN "WORD WRAP" MODE)
"/(/computer", then when you press enter, you will note that the cursor will be at the first use of the "(" following your present cursor location; A= replace all instances of "(" with "computer"; Y= replace only the one "(" at the present location. Go to the next "("; N= do not replace this "(" but go to the next instance, or S= stop and go out of this command condition.

After replacements are completed, return to edit mode. It is important to remember that when you enter this command mode, it begins where the cursor is located when you first press FCTN9 (command mode). So, if you wish to replace all uses of the "(" with the word chosen, make certain your cursor is at the beginning of the text.

AUTOMATING YOUR CALL-UPS WITH FAST-TERM

By Walt Howe

Downloaded from a BBS by Dave Talan

Getting tired of dialing numbers and entering passwords for the BBS you call regularly? If you have an autodial/autoanswer modem and Paul Charlton's firmware terminal emulator program, Fast-Term, you can eliminate those repetitive callups and logons forever. All you have to do is create an ASCII text file for each number you frequently call listing the modems dial code (ATDT) with the complete phone number, and the text answer to all the questions the BBS asks you at logon. You can use TI-Writer, the E/A or MINIMEM editor, or even a short basic program to create the text file. Put each item on a separate line, and if using TI-Writer, be sure to eliminate all those control codes such as "cr", which will usually foul up the logon.

Here is an example of the text file to automatically call Gary Blydenburg's PRO99 BBS:

<u>TEXT</u>	<u>REMARKS</u>
ATDT6645988	Add 1 if not in his area
N	"Are you using TE2?"
###	Your user number
PASSWORD	Your password

Save this file (without remarks) to disk using an appropriate title, such as "PRO99". Then, when you want to call the BBS, load Fast-Term and whatever defaults you normally use. Press FCTN N and type "dsk0.PRO99", then press FCTN comma to send the file. When it asks if you want to "Send one line at a time?", type Y. Then wait for the answer and the "CONNECT" indication. At each prompt, press the space bar to provide the next line of your text file. When you have entered all prompts for which you have answers, press FCTN 4 to exit the ASCII send mode and proceed as you always would. Create a file for each of those BBS you call regularly, even for CompuServe and The Source, and you won't have to keep looking up numbers and passwords and endlessly typing them.

FORTH SCR #5

by WESLEY R RICHARDSON
BLUEGRASS 99 COMPUTER SOCIETY, INC.

Problem solving can be fun or frustrating, a cause for defeat or a learning experience. Programming a computer is like problem solving because there is an objective to reach, and once it is attained, the problem solver feels an exhilaration from reaching the goal. This month TWELVE is a program which gives you a problem to solve. Versions have been included in Extended BASIC, c99, and FORTH. I would like to thank Jon Keller for his time and efforts in converting the XB version to c99.

Instructions for TWELVE are given in each version, but there are some slight differences in the way data is entered. In XB, press ENTER after you have pressed the letter(s) of your choice. In FORTH and c99, simply press the letter(s). In all

versions, when you do not wish to add any more weights on a side press ENTER. In all of the versions, consider the weights on each side of the balance to all be concentrated at one point as far as the distance to the fulcrum point is concerned.

The c99 version should be created using the Editor Assembler Editor, and use the filename TWELVE;C to save the program in DIS/VAR 80 format. You will need the new version of c99, with the files PRINTF GRF1RF RANDOM;C and CONIO in disk drive 1 to be able to compile the C code. Use E/A option 5 to load C99C from disk 1. Answer Y to the first two questions, and then give DSK2.TWELVE;C as the input filename and DSK2.TWELVE;S as the output filename. After compiling with no errors, load the E/A Assembler to assemble TWELVE;S with no options and the object file TWELVE;O. To run the program, use E/A #3 and load CSUP GRF1 PRINTF and TWELVE;O and then give START as the program name to start the program.

The FORTH version should be entered using -EDITOR. To run the program, load -TEXT -GRAPH and -GRAPH1 and then do 35 LOAD. After leaving the program, MON will return you to the TI title screen.

As far as solving TWELVE, it can be done in three weighings for any block heavy or light.

```
100 REM TWELVE EXTENDED BASIC
110 REM WESLEY R RICHARDSON
120 REM BLUEGRASS 99 COMPUTER SOCIETY INC
130 REM JUNE 1986
140 CALL CLEAR
150 PRINT TAB(9);"T W E L V E"
160 PRINT
170 PRINT "THERE ARE TWELVE BLOCKS"
180 PRINT "MARKED A THROUGH L. ELEVEN"
190 PRINT "OF THESE HAVE THE SAME"
200 PRINT "WEIGHT. ONE BLOCK IS EITHER"
210 PRINT "HEAVIER OR LIGHTER THAN THE"
220 PRINT "OTHERS. THE TASK IS TO USE"
230 PRINT "THREE BALANCE WEIGHINGS TO"
240 PRINT "DETERMINE THE ODD ONE, AND"
250 PRINT "IF IT IS HEAVY OR LIGHT."
260 PRINT "THE ARROW OVER THE BALANCE"
270 PRINT "INDICATES THE HEAVIER SIDE."
280 PRINT
290 PRINT "PRESS ANY KEY"
300 PRINT
310 CALL CHAR(91,"FFFF")
320 CALL CHAR(92,"FFFF183C66C3B1")
330 CALL CHAR(93,"10387CFE10101010")
340 CALL CHAR(94,"F0E0E09008040201")
350 CALL CHAR(95,"0F07070910204080")
```


...TWELVE

SCR #35

```

0 ( TWELVE #1 FORTH -TEXT -GRAPH -GRAPH1 )
1 ( WESLEY R RICHARDSON JUNE 1986 )
2 ( BLUEGRASS 99 COMPUTER SOCIETY INC )
3 BASE->R HEX : IT ; : GXY GOTOXY ;
4 : INST 9 6 GXY ." T W E L V E" CR CR
5 ." THERE ARE TWELVE BLOCKS" CR
6 ." MARKED A THROUGH L. ELEVEN" CR
7 ." OF THESE HAVE THE SAME" CR
8 ." WEIGHT. ONE BLOCK IS EITHER" CR
9 ." HEAVIER OR LIGHTER THAN THE" CR
10 ." OTHERS. THE TASK IS TO USE" CR
11 ." THREE BALANCE WEIGHINGS TO" CR
12 ." DETERMINE THE ODD ONE, AND" CR
13 ." IF IT IS HEAVY OR LIGHT." CR
14 ." THE ARROW OVER THE BALANCE" CR
15 ." INDICATES THE HEAVIER SIDE." CR -->
    
```

SCR #36

```

0 ( TWELVE #2 )
1 CR ." PRESS ANY KEY " KEY DROP ;
2 : VBL VARIABLE ;
3 0 VBL TRS 0 VBL COR 0 VBL ODD 0 VBL BAL
4 0 VBL WGT 0 VBL SID 0 VBL ALP 0 VBL ANS
5 0 VBL HL
6 : INIT GRAPHICS RANDOMIZE
7 FFFF 0000 0000 0000 5B CHAR ( 91 [ ] )
8 FFFF 183C 66C3 8100 5C CHAR ( 92 \ )
9 1038 7CFE 1010 1010 5D CHAR ( 93 ] )
10 F0E0 E090 0804 0201 5E CHAR ( 94 ^ )
11 0F07 0709 1020 4080 5F CHAR ( 95 _ ) ;
12 DECIMAL
13 : PICK 24 RND 1+ DUP 12 > IF 25 - ENDIF
14 ODD ! CLS ;
15 -->
    
```

SCR #37

```

0 ( TWELVE #3 )
1 : AL GXY ." A B C D E F G H I J K L" ;
2 : BEAM 2 BAL @ 4 * 3 + 25 91 HCHAR ;
3 : CTR 14 BAL @ 4 * 3 + 1 92 HCHAR ;
4 : PLCE ALP @ 2 * SID @ 14 * + BAL @ 4
5 * 2 + ;
6 : ACPT PLCE GXY KEY ;
7 : AVAIL DUP 65 - 2 * 3 + 1 GCHAR DUP 32
8 = IF DROP 0 ELSE ANS ! DROP 1 ENDIF ;
9 : CHK DUP 13 = IF DROP 1 ELSE DUP 77 <
10 IF DUP 64 > IF AVAIL ENDIF ENDIF
11 ENDIF 1 = ;
12 : ADEL PLCE 1 ANS @ HCHAR ANS @ 65 - 2
13 * 3 + 1 1 32 HCHAR ;
14 -->
15
    
```

SCR #38

```

0 ( TWELVE #4 )
1 : CWGT WGT @ 2 - SID @ 4 * + WGT ! ANS
2 @ 64 - ODD @ ABS = IF WGT @ ODD @ SID
3 @ 2 * 1 - * ODD @ ABS / + WGT ! ENDIF ;
4 : LALP 7 1 DO I ALP ! 13 ANS ! BEGIN
5 ACPT CHK UNTIL ANS @ 13 = 0= IF ADEL
6 CWGT ELSE LEAVE ENDIF LOOP ;
7 : LSID 2 0 DO I SID ! LALP LOOP ;
8 : APOS 14 BAL @ 4 * 2 + 1 ;
9 : ARRW WGT @ 0 = IF APOS 93 HCHAR ENDIF
10 WGT @ 0 < IF APOS 94 HCHAR ENDIF WGT
11 @ 0 > IF APOS 95 HCHAR ENDIF ;
12 : DBAL 3 1 AL BEAM CTR LSID ARRW ;
13 : RESP 2 17 GXY ." ODD BLOCK (A-L) ?"
14 KEY ;
15 -->
    
```

SCR #39

```

0 ( TWELVE #5 )
1 : CHK2 DUP 77 < IF DUP 64 > IF ANS ! 1
2 ENDIF ENDIF 1 = ;
3 : ANSWR BEGIN RESP CHK2 UNTIL ;
4 : RHL 2 18 GXY ." HEAVY OR LIGHT (H/L)?"
5 KEY DUP 72 = IF HL ! 1 ENDIF DUP 76 =
6 IF HL ! 1 ENDIF 1 = ;
7 : AHL BEGIN RHL UNTIL ;
8 : RIGHT 2 19 GXY ." *** CORRECT !! ***"
9 1 COR +! ;
10 : WRONG 2 19 GXY ." *** SORRY ***" ;
11 : CHL HL @ 74 - ODD @ * 0 < IF RIGHT
12 ELSE WRONG ENDIF ;
13 -->
14
15
    
```

SCR #40

```

0 ( TWELVE #6 )
1 : CANS ANS @ ODD @ ABS 64 + = IF CHL
2 ELSE WRONG ENDIF 1 TRS +! 2 21 GXY
3 ." CORRECT=" COR @ . ." TRIES="
4 TRS @ . ;
5 : LBAL 4 1 DO I BAL ! 0 WGT ! DBAL LOOP
6 ANSWR AHL CANS ;
7 : DONE ." type TWELVE to restart "
8 CR CR ." type FORGET IT to end" ;
9 : FINI 3 23 GXY ." PLAY AGAIN (Y/N) ?"
10 KEY 89 = 0= IF TEXT DONE QUIT ENDIF ;
11 : TWELVE CLS INST INIT BEGIN PICK LBAL
12 FINI AGAIN ;
13 R->BASE
14 TWELVE
15
    
```

In a previous column we discussed centering of information within a document using the standard font of the printer at 10 characters per inch. There are, however, many dot-matrix machines with capability for other pitch settings.

Conventionally, you are able to set six pitches on dot printers. These include pica, elite (12 characters per inch), condensed (17), expanded condensed, (8.5), expanded elite (6), and expanded pica (5). You can also alter fonts, or shapes of the characters and change language character sets.

Using the different pitch settings, especially expanded ones, in conjunction with the emphasized mode or double-strike, gives a lot of possible varieties for the headers you use in documents.

A real problem crops up, however, when one tries to center all the variants. In general, you will find that parts of your WP system are not smart enough to see that you have changed the size of the characters you are printing. If you try to center characters with the normal margins of your system and you are using different width characters, the system gets confused.

The areas that cause problems are once again the magic words: the margin settings. When you use the command to center (.CE), that command assumes that you want conventional width characters centered inbetween your normal margin setting.

The formula used to center a phrase in a document goes something like this. For this explanation, I will use the word "phrase" to name the item that is to be centered. There are four parts to the formula by the time the phrase is printed.

First, divide the number of characters in the phrase by 2 to determine the center of the phrase (we can call the formula $cip/2=a$).

Second, find the number of characters from margin to margin and divide this quantity by 2 to determine the center of the line to print ($(rm-lm)/2=b$).

Third, subtract a from b to determine how many spaces to add to the left side of the line in order to center the phrase ($b-a=iss$).

The 3 foregoing items are done by the Formatter in response to the centering command (.CE) At this time, the Formatter assumes that it has centered the phrase between the margins. But one more thing still has to be done; we still have to space over for the margin setting at the left side of the document. How this is done is a function of the equipment and user choice.

The left margin of your printed document can be established by the formatter only (printer left margin set to 1), by the printer only (the formatter left margin set at 0), or by some combination of the two. In any case, the offsetting of the printed characters away from the left edge of the paper is accomplished by the formatter, printer, or both counting characters from the leftmost column to the first printed column, then starting the actual print ($lss+lm=lpp$).

The formula above is a so-called universal formula. It applies to any pitch of print on any medium and with any width of margins. I have used it for setting centering of lines on a typewriter, among other things. The only

stipulation is that the pitch of print must be uniform throughout the line to be centered.

Notice I said in all cases that we are counting characters, not measuring space with a ruler. Because of this fact, any change in character size means the centering is done incorrectly so far as the linear space measure is concerned.

Given a document on paper that 8 1/2 inches wide, and using 1 inch margins, you will have the following approximate margin settings related to each character width setting.

Characters/inch	.LM	.RM
17	17	127
12	12	90
10	10	74
8.5	9	64
6	6	45
5	5	38

Some of these are approximations. Since you are not going to have any data sitting exactly at the left margins when you are centering, this is not a problem. By the way, I can hear you are thinking "127?? But TI-Writer can only handle 80 columns." WRONG!! TI-Writer can print up to 255 columns in one line if you handle it properly! However, this is a subject for another column.

If we are going to be switching character sizes within a document, the centering of those characters will depend upon their sizes. The margin settings that are correct when you are centering your normal text will not be correct for centering double-sized text.

You have to play with the margin commands to the Formatter (.LM and .RM) to reduce the number of characters in the line appropriately. You can adjust the centering if it's off one or two characters by changing the numbers in the margin set commands as needed.

To accomplish this, all you need to use is an algorithm for every line that used sized letters with the centering. Here is a sample of one I used when I was testing this hypothesis. I have had to add the one character to the left of each line, obviously, because I would rather print the algorithm than to have it execute while we are printing this article.

```
.CO 6 CHAR/INCH CENTERED HEADER
.CO Set temp margins for center
.LM 6;RM 45
.CO Set 12 CHAR/INCH, EXP modes
.TL 92:27,66,2,27,87,1
\
.CE
Print and cntr data on this line
.CO
.CO Restore char size modes
.TL 92:27,66,1,27,87,0
\
.CO Restore document margins
.LM 10;RM 75
```

By choosing the proper margins and code sequences for the transliterate commands (.TL), I can center any phrase on a line properly. By the way, the .TL sequences I show are for

a Gemini 10X printer and may not work with all brands. In each case, a 27 in the .TL command is the ASCII character code for ESC, and starts off a command sequence.

Basically, the .TL command causes the character string that follows the colon (:) to be sent to the printer in place of the character code that precedes the colon, each time the character whose ASCII code is the one before the colon is found in the text. 92 is the ASCII code for a backslash. Notice how I redefine the backslash and reuse it for each of the .TL commands.

Once again, let me stress that you may have to play individually with the left and right margin sets for each line of sized characters to make the printed phrase center exactly where you want it. The values I show are for starting points only and not to be taken as absolute.

There are 2 ways of setting the left margin. I prefer to set the margins in the formatter and leave the printer margin at 1. Some printers do not have settable margins, and by leaving the margin control with the formatter, I don't have to be concerned about this problem.

If you choose, you can set all your document and formatter left margins at 0 and adjust the printer's left margin. The algorithm above will still work if you do this, but you will have to delete the .LM commands, substitute, margin numeric values accordingly. The fact that this involves one more .TL command is why I went the way I did; there is less in-line programming involved.

I have added the routine below to my TI-Writer working master disk. That way I can read in the centering file at the appropriate place (using the "after line number" parameter in the LF command for the Editor) and save myself a lot of typing and memory work. Also by using lines that are already coded with the proper settings, I can delete all but the ones I need and not have to remember the numeric values for the margin sets. That takes about the same number of keystrokes as typing the numbers in, but makes the computer do the memory work for me.

The imperfection with this, is that the accuracy is plus or minus one character. This has to do with whether the phrase has an even or odd number of characters, whether the margins are even or odd on both sides or some combination of the two. I point this out because, at 5 characters per inch, the offset could be very noticeable. If so, you may want to adjust the right or left margin setting to adjust the center determination appropriately. The values in the table above were the ones I developed as I tested the centering using the routine.

Once you read this document into the one you are creating, you must do 2 things. First, select the character size and margin settings that match the ones you are using currently (alter if necessary) and copy those lines to the

restore part of the routine (near the bottom). The lines you do not need to set up your centering and character widths should be deleted from the top are of the routine. Finally, enter the phrase you want to size and center in the line indicated. If you wish to have more than one line, you can do that also; just indicate in the .CE command how many lines are to be centered. Because these must be consecutive lines, you cannot have them in different character sizes.

Finally, each time the .TL command executes due to sending the selected code, it causes the printer to space one line. This will leave one blank line above and one blank line below your centered information. (This may be printer dependent, so check on it) If you can't tolerate this, you must then send whatever code your printer uses to suppress spacing, do the .TL sequence, and then restore spacing both above and below the centered line.

The filename I have used for this document file on my TI-Writer working master disk is CENTERS. It assumes 8 1/2 inch paper and use of one-inch margins on each side of the page. My base margin settings are the restore/10 character per inch settings used in this sample.

```
.CO CENTER SIZED PRINTING
.CO $$$5 CHAR/INCH$$$
.LM 5;RM 39
.TL 92:27,66,1,27,87,1
.CO $$$6 CHAR/INCH$$$
.LM 6;RM 45
.TL 92:27,66,2,27,87,1
.CO $$$8.5 CHAR/INCH$$$
.LM 9;RM 56
.TL 92:27,66,3,27,87,1
.CO $$ CHAR/INCH$$$
.LM 10;RM 75
.TL 92:27,66,1,27,87,0
.CO 12 CHAR/INCH$$$
.LM 12;RM 90
.TL 92:27,66,2,27,87,0
.CO $$ CHAR/INCH$$$
.LM 17;RM 128
.TL 92:27,66,3,27,87,0
\
```

Print data on this line

```
.CO RESTORE CHAR SIZE
.CO $$ CHAR/INCH$$$
.LM 10;RM 75
.TL 92:27,66,1,27,87,0
\
```

Happy Centering!



Mickey & Minnie



(C) 1980
L.A. Olyn
Committee



Catherine
Deneuve



!!NEWSLETTER SUBSCRIPTION ONLY!!

___ Send me the next 12 issues of the CLEVELAND AREA 99/4 USER GROUPS NEWSLETTER. I am enclosing a check or money order for \$8.00. PLEASE PRINT

Name _____ Mail to: TI User Groups Newsletter
Address _____ C/o Frank Jenkins, 19809 Gladstone Rd, Cleveland, OH 44122
City, State _____ ZIP _____ This subscription is not necessary for fully-paid members of the User Groups.

___ Send me a membership form for the following group: ___Solon: ___ TI Chips: ___Golden Crescent: ___Northcoast

CLEVELAND AREA 99/4A USERS GROUPS
P.O. BOX 23283
EUCLID, OHIO 44123

CHECK YOUR EXPIRATION DATE.
THIS MAY BE YOUR LAST ISSUE!

!! TIME DATED MATERIAL !!