

---

---

# ISHUG

---

---

# NEWS DIGEST

---

---

Focusing on the TI99/4A Home Computer

---

---

Volume 10, Number 2

March, 1991

---

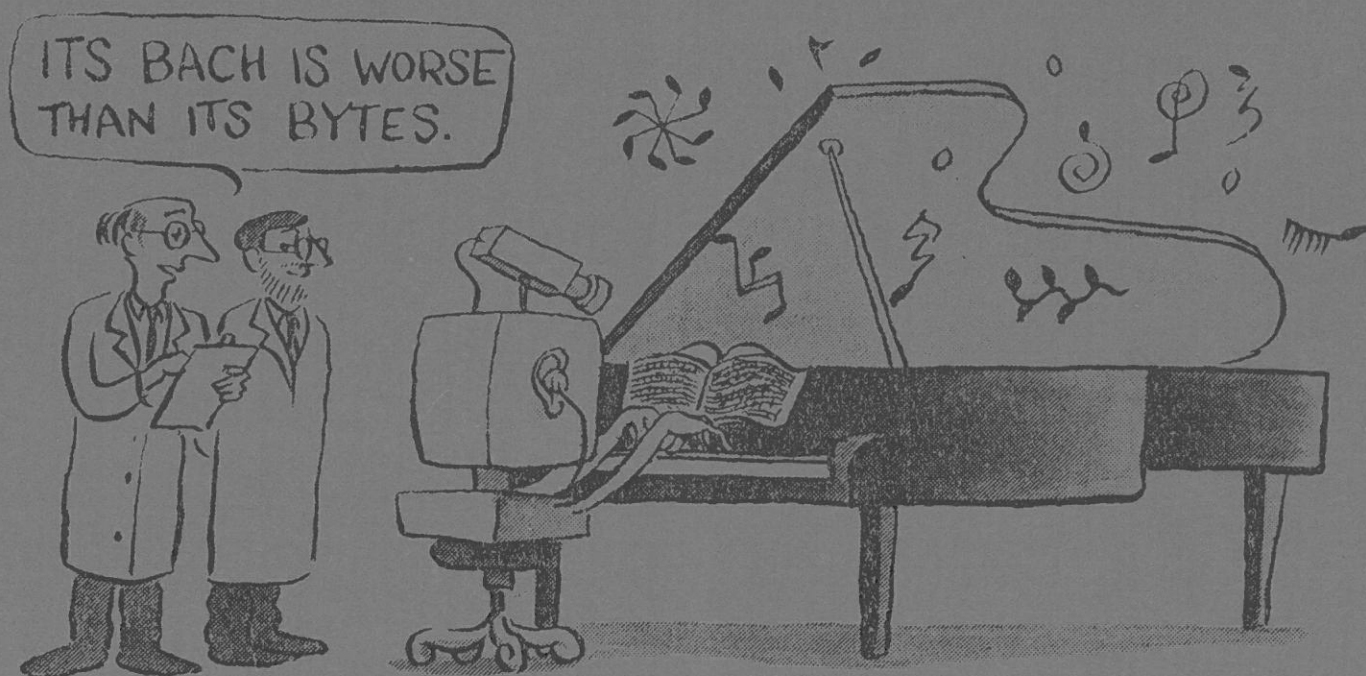
---

Registered by Australia Post - Publication No. NBH5933

---

---

## Harrison Music software demonstration



## MIDI music interface demonstration

---

P.O. Box 214, Redfern, New South Wales, Australia, 2016

\$3

---

TIshUG News Digest

Index

March 1991

All correspondence to:

P.O. Box 214  
Redfern, NSW 2016  
Australia

**The Board**

**Co-ordinator**

Dick Warburton (02) 918 8132

**Secretary**

Terry Phillips (02) 797 6313

**Treasurer**

Geoff Trott (042) 29 6629

**Directors**

Rolf Schreiber (042) 84 2980

Russell Welham (043) 92 4000

**Sub-committees**

**News Digest Editor**

Bob Relyea (046) 57 1253

**BBS Sysop**

Ross Mudie (02) 456 2122

BBS telephone number (02) 456 4606

**Merchandising**

Percy Harrison (02) 808 3181

**Publications Library**

Warren Welham (043) 92 4000

**Software library**

Rolf Schreiber (042) 84 2980

**Technical co-ordinator**

Lou Amadio (042) 28 4906

**Regional Group Contacts**

**Central Coast**

Russell Welham (043) 92 4000

**Coffs Harbour**

Kevin Cox (066) 53 2649

**Glebe**

Mike Slattery (02) 692 0559

**Illawarra**

Lou Amadio (042) 28 4906

**Liverpool**

Larry Saunders (02) 644 7377

**Northern Suburbs**

Dennis Norman (02) 452 3920

**Sutherland**

Peter Young (02) 528 8775

**Membership and Subscriptions**

Annual Family Dues \$30.00  
Associate membership \$10.00  
Overseas Airmail Dues A\$60.00  
Overseas Surface Mail Dues A\$45.00

**TIshUG Sydney Meeting**

The next meeting will start at 2.00 pm on 2nd of March at Ryde Infant School, Tucker Street, Ryde. At 12 pm, before the meeting, there will be a beginners Editor Assembler class for all those interested.

Printed by  
The University of Wollongong  
Printery

Title	Description	Author	Page No.
Archiving	Software review	Arnold, Tom	21
Assembly class	Link-it #19	Mudie, Ross	7
Beginning Forth #3	Software hints	Raguse, Earl	16
Co-ordinators report	General news	Warburton, Dick	2
Communicators	BBS information	Mudie, Ross	2
CSGD software overview	Software review	Amadio, Lou	6
Editor's comment	General interest	Relyea, Bob	1
Handy tips for the TI99/4A	Software hints	Spirit of 99	11
Look at assembler language #1	Software hints	Green, Art	22
Multiplan exercises	Spreadsheet	Schlesinger, Herbert	17
Newsletter update	General interest	Relyea, Bob	22
Regional group reports	General interest		23
Root finder	Software hints	Dorias, Lucie	20
Secretary's notebook	Club news	Phillips, Terry	3
Spellbreaker part 3	Adventure hints	Scorpio	12
Techo time	RAMdisk, consoles	Trott, Geoff	5
TI*MES utility catalogue	Software review	Shaw, Stephen	13
TI-Base tutorial #8	Data base	Smoley, Martin	19
TI-Bits #3	Software hints	Swedlow, Jim	8
Tips from the tigercub #61	Software hints	Peterson, Jim	9
TIshUG shop report	Club news	Harrison, Percy	3
TIshUG software column	Club software	Schreiber, Rolf	4
Treasurer's report	Club news	Trott, Geoff	23
Younger set	Games	Ball, Eric	11

**TIshUG Fairware Author of the Month**

The Fairware Author for this month is Ross Mudie for his tutorial series on linking assembler language routines to Extended BASIC, TI-opoly game, upgrades and fixes to the BBS software and many other contributions to TIshUG such as Joytalk and the wire I/O interface. The shop will have his tutorial disks and TI-opoly for sale in the near future. All donations collected at the meeting and sent in will be presented to him.

**Editor's Comment**

by Bob Relyea

The first meeting of the year seemed to be better attended than in previous years when we had the AGM first up. I like to get right into computing the first month so this suits me better than the previous system.

Over the holiday I did a bit more work with the 'Fish' program that I was working on with the help of Ross Mudie. I only need to do one more 'major' thing to it before it is completed. Does anybody know how to update records using an Extended Basic program? At present the program can update records manually but I want it to be automatic. I continue to use my computer to run off and update class lists for school as well as to standardize sets of class marks for the HSC students. I am looking forward to the activities at the March meeting. See you there.

# Co-ordinator's Report

by Dick Warburton

I thought this month we might talk about software, generally. There are significant moves in the PC world to reduce the copying of programs, particularly by business interests. I suspect that if programs were reasonably priced there would be less copying. The prices of well known PC software never cease to amaze me considering that their research and development costs have been met many times over. They are exorbitantly priced. The TI world has much less software, and is far less expensive, simply because there is only a small demand for the software, and the TI is not seen as a business machine. When PC software was introduced the prices of the original business machines were very high. Even the early PC's cost at least three or four times what they cost today, yet their performance could not be compared with today's machines. It seems to me that PC commercial software producers want it both ways. They seem to want to provide software at a cost which is out of proportion to the cost of machines today, and also cash in on the the large numbers of home computers being sold.

Some of the best software about is public domain software. It is sold and distributed with very little profit. TI users are very conscious of the freeware available, some of which is outstanding. We use Funlwriter all the time, or DM1000, or Disk Utilities. Unlike the business PC world, we are seeing the emergence of some excellent low priced programs for the TI, through Asgard. In fact as the TI continues to evolve, new software continues to emerge to take advantage of its capabilities. If this continues as the TI keeps developing, we will gradually find ourselves in possession of a computer which will meet most of the needs that any home computer user might have at very low cost. As my TI is set up now, with ramdisks and multiple drives, I find it is a more convenient and user friendly machine than any other I come to use at work or at home. Despite its slow CPU speed, I use it much faster than any other machine I have tried. Loading from ramdisk is very fast, and the proposed development of ramdisks with eproms will make our software capability even stronger. Working with files is easy if we work on the ramdisk. A TI with a ramdisk is a most useful machine, and it will process the bulk of available software both efficiently and effectively.

The software presently available for the TI can meet most of our needs at low cost. We can use freeware or commercial programs. I believe that many computer users have far more software than they can ever hope to use effectively. It takes time and effort to become proficient with any useful program. The more features a program has the more complex it is to learn. I still have so much to learn about using Funlwriter or even Disk Utilities without getting heaps more software. I would hope that 1991 will be a year where TISHUG members become proficient with their chosen software. I can do almost everything I want to do by using the following programs.

1. Funlwriter
2. Disk Utilities or DM1000
3. Ramdisk Operating Software
4. Multiplan
5. Prbase or TIbase
6. Mass Transfer or Telco
7. TI-Artist+
8. Pro Page
9. Utility Programs

I would like to become really proficient in the use of these programs, and I am proposing that if there are others like myself who still feel they have something to learn about software, that we devote a set time each month to developing our skills using our basic software. There are other newer programs which need practice too, eg, Spell It! I would like to see set tutorials each month in the basic areas. There are newer members who have not yet enjoyed the experience of using a ramdisk,

or even an expanded system. It would seem sensible for the club to provide some help in low cost expansion systems, particularly with multifunction cards, and ramdisks. There is no comparison between a TI with and without a ramdisk. A ramdisk turns a TI into a powerful and useful computer which can meet our needs well. Where does this all lead?

Well we can make sure that 1991 is the year where TI users get the best possible use out of their computers, by firstly acquiring proficiency with basic software, and expanding their machines to take advantage of the newer developments with ramdisks and eproms. There are many other possibilities eg hard disks, 80 column cards etc. but the key to improved computer functioning lies in using the ramdisk. To my knowledge, no other computer on the market today uses a battery backed up ramdisk. I suspect a ramdisk system could work extremely well on a laptop. A TI with a ramdisk is as user friendly as any system I have seen, and as effective. The newer software seems to be being written with the ramdisk in mind. You can expand your TI for a fraction of what it will cost to get comparable or better performance. New disk drives are now cheap. We can build our own multifunction cards or ramdisks. Memory prices have come down so Ramdisk memory is now affordable. Eproms are cheaper still. An effective expanded TI is now well within the reach of most members. Remember that we can get quality software very cheaply in comparison to PC software, and we can run it very efficiently if we take the trouble to expand our systems appropriately, and learn to use the programs proficiently.

I intend to do both things this year. I am determined to become more proficient using my software, and I intend to add a number of eproms to my ramdisk. If there is sufficient interest in new ramdisks, a new order for ramdisk boards will be given. Kits will be made up. If you are interested in expanding your system or in becoming more proficient in using software, see me at the next meeting to work out details of how we will do it.

See you at the next meeting,  
Dick Warburton

o

## The Communicators

by Ross Mudie

The usage of the BBS is slowly decreasing. The BBS is receiving around 100 calls a month from about 20 users, which includes SYSOP remote maintenance calls. Membership renewal time is coming up for most members thus all BBS members are urged to renew their TISHUG and BBS membership to allow the continuance of this club service.

The BBS has had some down time in recent months with power failures due to storms and telephone cable failure. The junction cable linking Berowra telephone exchange to the outside world was damaged by earth works isolating the area for an over night period.

There is very little "new" information coming in to be placed on the BBS, however many of the BBS members are new to the TI99/4A, thus the repeated "old" information and programs are new to these members. It is quite surprising how much we all forget which makes reviewing these old files (just like reading the old TND's) quite interesting.

I have been reading in some of the overseas user group magazines about a module which allows a TI99/4A with CORCOMP disk controller to format disks to IBM compatible format. Text files can be saved in the IBM format for use on an IBM computer. For most users the requirement of the special module and Corcomp disk controller make this impractical. Another way to transfer text files between a TI99/4A and an IBM, or in fact other computer types is to link the RS232 ports

continued on page 8

# Secretary's Notebook

by Terry Phillips

The first meeting of 1991 was held on a fine sunny day, perhaps even a little on the warm side, but it certainly bought out a large number of members. I counted about 50 in attendance at one stage during the afternoon. It was good to see a number of younger members also in attendance, although I must say a number of them were looking fairly bored as the afternoon progressed. I had a quick talk with Dick Warburton on this aspect and he and I agreed that the only way to hold the younger members interest would be to establish the long awaited GAMES ROOM, and this in fact will become a reality from the March Meeting onwards. That month will see the ROCKRUNNER competition in full swing with a very generous prize, from Larry Saunders, being donated to the winner. So all you young members come along and try your hand at Rockrunner. If you have not as yet played it then it will not take long to learn but a great deal of time to master. We hope to have another computer set up for games play as well.

Since my last column we have attracted a further 8 new members, 6 locals and 2 from overseas. So it is a big welcome to the following:

Curtis MAVITY - Mesa, Arizona  
John SENTER - Northridge, California  
Peter GARTNER - Bondi Junction  
Jim LAVELLE - Maryborough  
Geoff PHILLIPS - Warners Bay  
Lauren LUDEKE - Riverstone  
Neil WARBY - Epping  
Michelle PAULL - Edensor Park

In addition, former member David ANDREWS, has come back to the fold and rejoined the group after an absence of nearly 12 months. Welcome back to you also David.

A former member from Victoria, Andrew Szemere has written advising he is selling his TI equipment and all the following is offered for \$500 the lot. Andrew says everything has been well cared for and is in excellent condition. If interested he can be contacted on (03) 803-6587 between 7pm and 10pm week nights only.

Here is the list:

Beige console with power pack, UHF modulator, cassette cable, TI joystick, PE Box with Corcomp 32K card, Corcomp RS232 card, TI disk controller and one SSSD full height drive, Navarone cartridge expander, DM2, TI Writer, Editor Assembler, TE2, Extended Basic, Parsec, Microsurgeon, TI Invaders, Music Maker, Return to Pirates Isle, plus lots of program and utilities disks and books, magazines etc. Everything is here that you need for a complete system.

Recently a collection for his fairware contributions was taken up for software author Art Green, probably more familiarly known as "RAG SOFTWARE". The following response has been received from Art:

"I want to thank the members of TIsHUG for their fairware contributions for my programs. The money you sent will soon be returning to Australia through an indirect route. I, like a lot of other people have recently discovered that Australia produces some very good wines, several of which I drank over the holidays.

"To update your program library, enclosed are copies of the latest versions of my programs. You should note that there have been two minor bugs fixed in Multiplan. Bill Chavanne of the US has worked MP over with his TITAX package (US income taxes) so that there probably are no more bugs.

"Also enclosed is a disk containing several articles that I wrote for the Ottawa TI Group. You can pick and choose and republish those you think may be interesting to your group.

"I found your newsletter to be excellent. The Ottawa User Group is, like a lot of others, slowly decreasing in size and our newsletter has suffered. In the copy of your newsletter that you sent me I noticed mention of a problem using the Personal Record Keeping module with a RAM Disk. This problem may be conflicting subroutine names. PRK has quite a few "funny" subroutine names. Names like >04, >05 and so on.

"Finally I would like to wish a HAPPY NEW YEAR to all members of TIsHUG."

Thanks Art for that nice letter and it is great to see that authors do appreciate receiving contributions. Remember we usually have a fairware author selected each month and you can make your donation at the meeting or send it through the mail.

By the way, the disks that Art mentioned in his letter have been passed onto Rolf who will no doubt make them available at future meetings.

That is it for this month. See you at the next meeting on 2nd of March. o

## TIsHUG Shop with Percy Harrison

This month I am very disappointed to report that the Coloured Monitors which we had on order with Wang will not be forthcoming. We have been advised that, because of the serious downturn in the economy, Wang's source for these Monitors has dried-up and, consequently we will not receive any more Monitors from them. I have therefore arranged with our Treasurer to refund all monies paid on outstanding Monitor orders except in those cases where I have been requested to do otherwise. If you have not yet received your refund please let me know so that I can follow-up with Geoff.

We still have a limited supply of new and slightly used 5 1/2 inch DSDD Half Height Disk Drives at \$65.00 and \$50.00 respectively, both come with a 12 month guarantee, all ideally suitable for expanding your TI System.

Another new stock line is a 8.5, 17 volt Power Pack priced at \$25.00 and very suitable for powering up Disk Drives and other low voltage hardware.

We have the opportunity of purchasing a number of Drive Cases suitable for housing one Half Height 5.25 in. Disk Drive. These will retail at \$35.00 each, anyone interested in purchasing these should let me know so that we can determine whether we should proceed with a bulk purchase.

In addition to the club software listed in the various issues of the TIsHUG TND News the following shop items are available, but when ordering by mail please ensure that you remit sufficient funds to cover the cost of packaging and postage:

5.25 in. DSDD Disks (Boxes of ten) .....	\$6.50
5.25 in. HD Disks (Boxes of ten) .....	\$16.00
3.5 in. DSDD Disks (Boxes of ten) .....	\$12.00
3.5 in. SSDD Disk Drives .....	\$20.00
5.25 in. DSDD Half Height Drive (New) .....	\$65.00
5.25 in. DSDD Half Height Drive (Used) .....	\$50.00
12 Volt AC Transformer .....	\$3.50
13 Volt Arlec Transformer .....	\$12.00
8.5, 17 Volt Transformer .....	\$25.00
60 VA Transformer .....	\$20.00
MFC Printed Circuit Board .....	\$30.00
MFC Kit (Disk Controller) .....	\$102.50
32K Kit for MFC .....	\$26.50
PIO/RS232 (single port) Kit for MFC .....	\$42.50
Combined 32K and PIO/RS232 Kit .....	\$60.00
Music Kit with PCB .....	\$65.00
32K Memory PC Board .....	\$7.00
TI Joystick .....	\$12.00

continued on page 23

# TISHUG Software

## Column by Rolf Schreiber

My column this month will be briefer than I intended, mainly because of time limitations, and not due to any lack of software. The hot weather over the past month resulted in my taking a break from computing (very unusual for me, just ask my wife!). I need someone to help me review new software, which means becoming familiar with the software and then writing an article about it for the TND. So, if you are interested in helping the club, as well as honing your journalistic skills, here is the chance that you have been waiting for. Do not be afraid that you will be asked to review a complicated software package straight away. There are plenty of simple BASIC games or utility programs in our library that you can practice on. In fact, the best software to start reviewing is the software that personally interests you.

Take the time to peruse the utility software listing from Stephen Shaw's UK based User Group Library. We already have quite a few of those disks in our library and I am always willing to get any disks requested by members.

### Software releases for March

DISK A439 is called Multiplan Exercises. This DSSD disk contains all the files referred to in the Multiplan tutorial, currently running in the TND. The software and tutorial were all written by Herbert Schlesinger, a dedicated TI99/4A User in the USA. A two disk SSSD version is available for \$3.00.

DISK A444 is the seventh disk in the TI Base tutorial series from Martin Smoley. The disk is SSSD and contains the necessary sample data base and setup files required to work through the tutorial on the computer.

DISK A450 is the latest version of Funnelweb (V4.31). This version has an upgraded DiskReview. It is available as a DSSD disk, or in SSSD format.

DISK A451 is the latest version of Art Green's Multiplan Upgrade. This version (V4.02), which comes on a SSSD disk, corrects the bug when cataloguing a disk with the Transfer Load option. A GRAM based version is also available for GRAM devices such as the Gram Kracker or the P-Gram card.

### Commercial Software Releases

Character Set & Graphics Design I (CSGD I). This is a three disk (all SSSD format) set of programs which make extensive use of your printer's graphics capabilities. The programs are all menu driven and allow you to configure your system for a single or double drive setup, as well as configure your printer for a PIO or RS232 interface. The programs enable you to create your own new character sets, or modify existing ones. These character sets can then be used in printing out letterheads, greeting cards, signs and messages, as well as any TI-Writer text files on an Epson compatible dot matrix printer. You can also create pictures or save screens from BASIC or Extended BASIC programs to disk, for later printing or manipulation, such as a mirror image, double size, or changing into a negative. The CSGD I package comes with extensive on-disk documentation, as well as printed supplementary notes. Included with the software are 10 fonts, 10 small pictures and 3 large pictures, and the price is \$12.00 from the shop.

Character Set & Graphics Design II (CSGD II) is the second program package in the CSGD software series created by David Rose. There is some overlap in the features of CSGD II, compared to CSGD I, but the two program packages essentially compliment each other. The primary feature of this software package is a

powerful banner maker. With it you can create true graphics banners using any of the CSGD fonts and small graphics found in the CSGD series. Included on this two SSSD disk set are 8 fonts and 65 small pictures, as well as a documentation file. Printed documentation is also included in the price, which is \$10.00.

Character Set & Graphics Design III (CSGD III) is a continuation of the printer utilities of CSGD I. The package comes as a three disk (all SSSD) set and offers four new, powerful, and advanced graphic printing capabilities. CSGD III can be used to design and produce custom stationery and letterheads, signs and messages and graphic labels. Using the Docuprinter software included on the disk, it is possible to print documents created with TI-Writer (or any of its clones) in any one of six bit-mapped character sets. Included are 25 fonts, 40 small graphics and label borders, as well as documentation files and printed documentation. The price from the shop is \$14.00.

CSGD Cataloguer is an extremely useful utility if you use CSGD graphics and fonts. The program is menu driven and allows you to print out (on a dot matrix printer) a catalogue of the actual small graphics on a disk. Up to ninety pictures fit on a page, complete with filenames for easy identification later on. A similar print out is possible for CSGD fonts. The software can also create a cross reference of all your CSGD graphics and fonts. As an added bonus, TI-Artist Pictures and Instances may be similarly cross referenced. I have personally used this program to print out listings on my new NX1000 printer, and I am impressed with the program's capabilities. It is available from the shop for \$8.00 (SSSD disk and instructions).

The Missing Link (TML) has previously been previewed in the TND, and a more comprehensive review is planned for the future. This software package comes on a single SSSD disk with a 36 page manual, and includes a paper saver utility and a comprehensive demonstration. It was designed to give Extended BASIC programmers easy access to all the capabilities of the TI99/4A computer, which previously meant knowing how to program in assembly language. For those of you unsure as to TML's capabilities, there is a demonstration only disk available from the shop for only \$2.00. Just ask for disk A400. The price for the TML package is \$28.00.

TML Companion Disk #1 is available for \$2.00. This disk requires the TML software and will not run without it. The disk contains a series of demonstration files, collected by Stephen Shaw.

Spell It! is a new spell checker from Asgard Software. It was thoroughly reviewed in last month's TND. The software is available on a DSSD disk for \$22.00, including a comprehensive manual. The SSSD version comes on three disks and costs \$24.00. A more extensive version for a hard disk system can be ordered for \$45.00. This version contains a dictionary of about 200,000 words.

Page Pro 99 V1.6 comes as a package comprising two SSSD disks, two manuals and an addendum. The software allows the user to produce pages of text combined with graphics (commonly called desktop publishing). A vast range of support software is already available for Page Pro 99, with more to come. The package will be the subject of a more comprehensive review in the future, as soon as I can get someone to do it. The price is \$28.00 from the shop.

Page Pro Applications #1 is available from the shop for \$2.00. The SSSD disk contains a variety of utilities designed to enhance the capabilities of Page Pro.

Page Pro Utilities is a SSSD disk containing a collection of related programs. A menu allows easy

continued on page 12

# Techo Time

with Geoff Trott

## Errata for the EPROM system

There was at least one mistake in last month's article (we only put these in to see if anyone actually reads the articles). In the table on page 6 at the bottom, the first line for EPROM E10 should have connected to pin 27 of E10. The other confusion was about EXJ, which is a set of 16 holes labelled EXJ on the top left of the board between the 74LS154 and the memory ICs.

Lou has decided to take a break this month (he could not think of anything new to say) and he asked me to step in and say a few words. I have been quite busy over the Christmas/New Year season looking at miniPE systems, super cartridge batteries and now some consoles. Firstly, the miniPE system.

## MiniPE fun and games

Bob Relyea, our esteemed editor, has a miniPE system just like the one I have, with a single RS232 motherboard, disk controller board and a RAMdisk board. He up-graded the diodes in his console and had a 32K extended memory also in the console. He was having problems with losing the ROS in the RAMdisk which I had one go at and then he was having problems with having to wait for the disks to start to work properly. All in all it was not very reliable and so not satisfactory. For a variety of reasons he decided that it would be better if the 32K memory was put on the miniPE motherboard (as mine was) and if the disk controller were to be powered externally. This was after we had done a few swaps of boards where his disk controller and RAMdisk worked fine on my system but mis-behaved on his system.

So I ended up with his console and miniPE system for a month or so to do all this. As it turned out, I could get his console to mis-behave after running for a while and the disks would go off-line so I knew that the problem was not with his home installation or power source. The first thing was to take the 32K memory from his console and install it on the miniPE board. No real problems here. While everything was apart I inspected the power supply and noticed that the soldering around the new diodes was rather suspect so I cleaned that up. I then measured the current drawn by the disk controller board and found it to be 420 ma which was much more than I expected. The whole console uses about 1 ampere, so that the disk controller was increasing the current drawn from the 5 volt supply by 50%. The RAMdisk was drawing about 110 ma and had its own supply any way. It was easy to install a 7805 regulator chip with a few capacitors (holes already there) and heatsink and to remove the link which connects the 5 volts to the console supply. Of course there was not much room for a heatsink but with a little creative bending of a reasonable sized one from Dick Smith it all ran not too hot. I bought a plug pack from Dick Smith which was rated at 500 ma at 9 volts. I was actually after one which was in their catalogue at 600 ma and was going to power both the disk controller and the RAMdisk from the one plug pack but there were none in stock here at that time.

Then there was the simple matter of installing the latest in my electronic solutions to stop losing the ROS from the miniPE RAMdisk and then returning it all to Bob. At the last meeting he was still smiling (he is always smiling) and the system was still running properly. By the way, I have taken to always doing a CALL AF (in either BASIC) before turning off and a CALL AO after turning on (DiskReview in Funnelweb 4.31 will do this also). A bit of a nuisance I suppose but not as much of a nuisance as losing your ROS. The miniPE seems to be quite temperamental in that regard, but I know of two that are running well (fingers crossed, touch wood).

## Battery backup of Cartridge Memory

I have a Super Space cartridge and a Maximem/Supercart built by Lou both with battery backup. Unfortunately they both discharge their batteries far too quickly for my wallet if they are left plugged into a console which is turned off. This is because to enter their low power mode the chip select line must be held high. To do this requires a resistor between that pin and the battery supply and that the output that drives that pin does not draw any current when the power is off. This is where the problem occurs, as the output that drives that line comes from a chip which does not behave like that. Consequently current flows from the battery through the resistor and the chip select line is held low. The memory chip does not go into its low power mode and the battery discharges at a fast rate.

The fix I used was to put an npn transistor in series with the chip select line (cut the track and bridge the cut with the transistor) with the collector to the memory chip side and its pull up resistor and the emitter to the console side. The base is connected through a 10k resistor to the 5 volt supply (not the battery). This means that when the 5 volt supply is active, the transistor is on and all signals pass straight through, while when the 5 volt is at 0 volts, the transistor is off and no current will flow through it so that the chip select line at the chip will be held high by the resistor and the memory chip will enter its low power state. The new battery has lasted for many months, thank goodness. I did try some other methods like capacitor coupling but these were not as successful.

## Consoles and more consoles

At the meeting in February I was given 7 consoles to have a look at. Three of them had been looked at by the console repair group, one was from the Central Coast group where a member is not having much fun with his computing and the other three were exchanges made at the meeting. At this stage I have fixed four of them (the easy ones) and I am pondering the other three. I thought I would give you a run down on each.

The first had the symptoms of a cyan (normal background colour) blank screen with no noise but otherwise dead. The console tester runs and indicates no problems. On further investigation, the system "stops" because of an interrupt which is not able to be cleared. That is, the interrupt line into the 9900 (pin 32) is low, which is determined by the INTREQ signal from the 9901, which is determined by either an interrupt from the 9929 video processor or an external interrupt. The blank cyan screen is there because at a particular point in the start-up sequence, the screen is blanked and interrupts are enabled. At that time the interrupt processing routine is entered and appears to be unable to identify the source of the interrupt (which appears to be the video processor) and so just keeps looping. Because the console tester does not use interrupts it runs without errors. At this stage I have checked the 9900, 9901 and 9929 which all work well in another system.

The second one had a black screen and a low frequency noise. The console tester did not run so after taking everything apart and removing the GROMs and sound chip I looked at the 9900 pins. The clock was present but there was no signal on pin 7 (IAQ). This indicated a dead 9900 and sure enough it was not good when tested and a new 9900 made it as good as new. One success!

The third one has a rather peculiar screen which is trying to be correct but the colour bars are all over the place. Selecting BASIC and all looks OK until you try to type something in, when two characters appear and when ENTER is pressed the whole line repeats and an error message appears. The console tester runs without showing any problems and the Diagnostic module also runs without errors although the screen displays are all over the place, particularly in the multi-colour mode. This

is a very strange problem and I have not had time to look any further.

The fourth console had a black screen and low frequency noise but the console tester ran and indicated a problem with the high byte ROM. I put in a correctly programmed EPROM and all OK.

The fifth console has a black screen and low frequency noise and the console tester will not run. Some of the address line look strange but as I do not have any 9900 chips left I have not progressed any further on this one. I will remove the 9900 and check it next.

The sixth console was quite dead with no noise and the console tester produced no lights. Checked the power supplies and the +5 volt supply was not present. There were two transistors and one resistor blown and replacing these means that all is now working.

The last console was a black and silver one which had a black screen and high pitched tones. The console tester did not work so I took it to pieces and started by taking out the GROMs and sound chip. Still no action with the console tester but the processor signals all seemed to be fine. I put the system GROM back in and all of a sudden everything started to work! Amazing, what did I do? The console tester showed that the vacant GROMs were generating a checksum of >C0000 which indicated that one of the bits on the data lines (the most significant one) was not being pulled low with the rest. There is a transistor and a few resistors which is connected to the common point of a set of 1.5k resistors in a sip and takes the common point to +5 volts when data on the 8 bit bus is going out of the processor and to 0 volts when data is coming in to the processor from the 8 bit bus. Looking at these signals when data was being read from the GROMs, I could see the problem (the line was held high) but there are 3 chips involved at that point. Doing some inspired guess work I took one of them out. Not that one! Then there were two. The next one out also was OK. Checked the resistors and no problems there. Take the last one out and it was fine too! Time to tear out the last few remaining hairs! Then I started looking with the ohmmeter and noticed that there was a difference in the resistor value on that line. Looking around I just happened to notice what looked like a biological growth under the cartridge port socket on the board. Scrubbing around the socket removed the problem electrically but I could still see stuff under the socket. Taking the socket out allowed all the mess to be cleaned up. There was a bit of corrosive damage but nothing too bad and a wash with soapy water followed by alcohol and it looked much better. I would say that it was the residual of some milk based drink which was spilt some time ago. Flexing the board allowed it to work for me and allowed me to find the problem. Everything was re-soldered into place and it was all working again. During this repair my console tester stopped working. It was a RAM memory problem.

I will tell you the finish to this saga when I know more. Watch this space and you will be the first to know (well almost). Happy hunting. o

## Overview of CSGD I, II and III

by Lou Amadio

"CSGD" stands for Character Set and Graphics Design. The programs were written by Dave Rose and are marketed by Texaments. Three sets of disks, (CSGD I, II and III) are available through the club shop.

Although written as stand-alone packages, all three programs compliment each other. In all, the graphic libraries include 10 programs, 80 character styles, 57 large picture files and over 300 small graphics, in other words, excellent value for money.

Read the supplied instructions carefully (some are

on disk) as this will allow you to get more effective use out of the programs.

Required hardware consists of a console with 32K memory, Extended BASIC, disk controller, disk drive, RS232 or PIO interface and, of course, an Epson compatible graphics printer.

The printer must be set to accommodate 8 bit graphics. This may or may not be switchable on your printer.

All of the programs are loaded from Extended BASIC using the auto load option. As the programs are also written in Extended BASIC (except for some parts of CSGD III), loading and running seems to be a little slow.

The system would improve significantly if run from a RAMdisk.

### CSGD I

The main purpose of this program is to "Print Messages", with a certain number of enhancements. These include: normal or compressed print, up-side down printing (for greeting cards), centering, quadrant selection, character size and spacing, line spacing control and inclusion of graphics and text on the same line.

Examples of the type and quality of available graphics are included in the Getting Started guide which comes with the program disks. A number of graphics and fonts are included on the disk plus instructions on how to create a custom letterhead with the Print Messages option.

### CSGD II

This program supports the design and production of banners. As in CSGD I, the program allows control of character spacing, up-side down printing and inclusion of a graphic and text on the same line. In addition, there is support for message centering (bearing in mind that the banner is printed sideways) and normal or negative printing. There is also an option for defining blocks (with patterns) for elongated printing and other special effects.

Numerous character and graphic patterns are included with the disks plus instructions on how to use them.

### CSGD III

This program set is the third in the series from Dave Rose. This is a revised and improved version of the other CSGD programs, although it does not supercede the other two. Improvements include the use of assembly language subroutines to speed up some of the slower operations such as printing and manipulation of graphics.

CSGD III supports the following:

- Labels - using fonts, graphics and frames.
- Letter heads - Custom designs for business or personal use.
- Messages - revised and improved version originally included with CSGD I.
- Document Printer - Prints DV80 files in a number of different font styles.

The program loads automatically with Extended BASIC and should be configured for printer type prior to use (using the "Configure System" option).

Print out the manual and review the instructions prior to using the program. The instructions which accompany the disks include examples of available fonts and graphics. c

# Assembly Class

## Link-It #19, by Ross Mudie

The next assembly class will be held at 12 noon, prior to the meeting on 2nd March 1991. The class is working slowly towards a CALL LINK("ACCEPT") routine. The following article explores the byte which conveys the number of parameters (arguments) in the CALL LINK and argument identifiers. As usual, class members should bring their Editor/Assembler manual for reference purposes, plus a note pad and pencil.

### LINK-IT 19 by Ross Mudie, 5th February, 1990

When linking between extended BASIC and assembly there are 4 very convenient utilities which may be used to pass values and strings between the two environments. These are:

STRASG -Pass a string from assembly to Extended BASIC.  
STRREF -Pass a string from Extended BASIC to assembly.  
NUMASG -Pass a value from assembly to Extended BASIC.  
NUMREF -Pass a value from Extended BASIC to assembly.

Unlike subprograms in extended BASIC, which must have a perfect match between the number and type of arguments in the SUB PROGRAM CALL and the SUBPROGRAM, a linked assembly program can test for:

- The number of arguments in the CALL LINK.
- The type of each argument in the CALL LINK.

The reasons for wanting this information is to allow programming versatility and bullet-proofing. By allowing a variable number of arguments your assembly link can use just the number of parameters required from the Extended BASIC program. If the type of variable is tested, then the variable may be numeric or string to suit the requirements of the extended BASIC program. For example a CALL LINK ("ACCEPT",x) routine could allow x to be either string or numeric.

If CALL LINK("ACCEPT",N) was used, the ACCEPT routine could allow a numeric input only and return a numeric value.

If CALL LINK("ACCEPT",S\$) was used then the routine could return a string.

The information on page 278 of the e/a manual gives details which apply to linking with TI BASIC. The address >8312 is applicable to extended BASIC and contains the number of arguments in the link list as a BYTE sized value. The argument identifiers are from >8300 to >830F under extended BASIC. >8300 is for the first argument, >8301 is for the second argument, etc. These are BYTE sized values. The values are 0 for a numeric expression, 1 for a string expression, 2 for a numeric variable, 3 for a string variable, 4 for a numeric array and 5 for a string array.

In the example CALL LINK("ACCEPT",x) the presence of a numeric variable in x would result in a byte value of 2 in >8300 and a 1 in >8312.

Another example of using the argument identifiers would be to prevent the assembly program from trying to pass a value or string back to extended BASIC if it found a 0 or 1 in the appropriate identifier. (If an attempt was made to pass information back to an argument which was an expression, rather than a variable, then the extended BASIC program would terminate in an error).

The following programs allow the subject of this file to be explored. The extended BASIC program loads the assembly program and links into it. If the number and type of variables in the CALL LINK argument list are changed the assembly program will report what it sees. The number of arguments may be changed from zero to 16. The loading method using ON ERROR to reload the linked

assembly overcomes the un-necessary reloading delay when RUN is used rather than RUN 170.

The assembly program reports on the screen the number of arguments in the CALL LINK and the type of argument. Assembly subroutine structure (BL) is used for the SCROLL and PRINT functions.

### EXTENDED BASIC PROGRAM

```
100 ! SAVE DSK1.LOAD
110 ! LINK-IT 19 Ross Mudie, 5th February 1991.
120 ON ERROR 190
130 CALL PEEK(16376,D,E,F):: IF D=83 AND E=84 AND F=65
    THEN 170
140 PRINT "Loading Assembly..."
150 CALL INIT
160 CALL LOAD("DSK1.O")
170 CALL LINK("START",1,"B",C,G$,H()),I$(),J$(),K,L$,M)
180 END
190 PRINT "Reloading Assembly..."
200 RETURN 150
```

### ASSEMBLY PROGRAM

```
        IDT 'ASYCLASS'   S=S O=0
        DEF START       LINK-IT19
        VSBW EQU >2020   Ross Mudie
        XMLLNK EQU >2018 5th February 1991.

SAVRTN BSS 2
WS      BSS 32

NUMBER TEXT ' 0 1 2 3 4 5 6 7 8 910111213141516'

ARGTXT BSS 2
TEXT ' Arguments in CALL LINK'
EVEN

ARGNO TEXT ' Argument '
ARGNOV BSS 2
BUFFER BSS 19

TYPE0 TEXT ' numeric expression'
TYPE1 TEXT ' string expression '
TYPE2 TEXT ' numeric variable '
TYPE3 TEXT ' string variable '
TYPE4 TEXT ' numeric array '
TYPE5 TEXT ' string array '
EVEN

LOOKUP DATA TYPE0,TYPE1,TYPE2,TYPE3,TYPE4,TYPE5

SCROLL BLWP @XMLLNK   Scroll subroutine. Place number
DATA >26             lines to scroll on R10 then
DEC R10              BL @SCROLL.
JNE SCROLL
RT

PRINT MOVB *R4+,R1    Subroutine to print on screen
AI R1,>6000           which adds the hex 60 offset
BLWP @VSBW           for extended BASIC.
INC R0               The VDP start location must be
DEC R2               placed in R, the start address
JNE PRINT            of the text in CPU RAM in R4 &
RT                  number of bytes to write in R2

*
START MOV R11,@SAVRTN Save the return address, first
        LWPI WS      Load program register work space

        MOVB @>8312,R3 Get the number of arguments
        SRL R3,8      Make it into a word value

        SLA R3,1      Multiply by 2
        MOV @NUMBER(R3),@ARGTXT Get text for number
        SRL R3,1      Divide by 2

        LI R10,2      Number of scrolls
        BL @SCROLL

        LI R0,737     Start position on screen
        LI R4,ARGTXT  Start of text in CPU RAM
        LI R2,26      Number of bytes to print on screen
        BL @PRINT    continued on page 10
```



# TI-Bits Number 3

by Jim Swedlow, CA USA

[This article originally appeared in the User Group of Orange County, California ROM]

## THE INPUT-OUTPUT BUFFER

When you send data to your printer or to disk, your TI stores information in the input-output buffer. Generally it will keep data until it sees the end of a record.

To illustrate, consider this program that demonstrates the graphics abilities of Epson and compatible printers:

```
10 OPEN #1:"PIO.CR"
20 PRINT #1:CHR$(27);"L";CHR$(127);
   CHR$(0)
30 FOR I=1 TO 127
40 PRINT #1:CHR$(I)
50 NEXT I
60 PRINT #1:CHR$(13)
70 CLOSE #1
```

Line 10 opens your printer and tells your 4A NOT to send a carriage return and a line feed every 80 characters. Line 20 puts your printer in graphics mode and tells it to expect 127 graphics characters. The loop in lines 30 through 50 send the entire range of graphics characters. Line 60 sends a carriage return to clear the printers buffer.

Since there is no print separator after the CHR\$(I) in line 40, each character is taken as a record and sent to the printer. If you added a semi-colon after the CHR\$(I), all 127 graphics codes would be held in the input-output buffer until line 60 executed.

The difference is speed. Without the semi-colon, it took about 10.2 seconds for this program to run. When the print separator is added, run time dropped to 7.5 seconds.

## TRACKBALL TIP

The following article was on a disk someone sent me and was written by Rick Kellogg. I do not know where he can be reached.

A few months ago I purchased a trackball for my computer. I planned to use it with a few of the games that I had and with 'TI ARTIST'. Being somewhat lazy, I soon grew tired of switching the trackball and joystick plugs. Then an idea came to me. I pulled out my old (ahem!) ATARI 2600 and took the "Y" adaptor. Lo-and-behold, when this adaptor was hooked up to the TI joystick port, I could plug in both my TI joysticks and the trackball at the same time. Not only was this convenient, but at any time I could switch from joystick 1 to the trackball and not even lose a life in a game. For those of you who use ATARI compatible joysticks, all you have to do is plug your TI "Y" adaptor into the ATARI "Y" adaptor and you are all set. You 'TI ARTIST' fans will love it.

## THE SMART PROGRAMMER

This is probably old news to most of you, but if you have not heard, Richard Mitchell of Bytemaster Computer Services has taken over Miller Graphics THE SMART PROGRAMMER. As a result, Bytemaster no longer publishes SUPER 99 MONTHLY. The new publication, however, continues many of the traditions of both the old SMART PROGRAMMER and SUPER 99 MONTHLY.

Craig Miller has contributed to both of the new SMART PROGRAMMER issues released thus far. While it can get a bit techie sometimes, it remains an excellent publication. Further, it meets the needs of all TI owners, from novice to expert. The cost is \$15 (\$18 for first class). Write to:

Bytemaster Computer Services  
171 Mustang Street  
Sulphur LA 70663

## QUOTES OF THE MONTH

The best laid schemes o' mice and men  
Gang aft a-gley  
---Robert Burns 1759-1796

But it does move!  
---Attributed to Galileo Galilei  
1564-1642

## ON SUBPROGRAMS

SUBprograms, if you remember, use different variables from the main program. As a refresher, consider this:

```
10 A=3 :: CALL TEST :: PRINT A
20 SUB TEST :: A=10 :: SUBEND
```

If you run this, you will get the number 3 on your screen because the variable A in line 10 is a different variable from that in SUB TEST.

I wondered about how DATA strings and DEFINitions worked in SUBprograms. So I ran some experiments.

It turns out that a XB program can read a DATA statement anywhere. It works if the DATA statement is inside a SUBprogram and the READ command is in the main program or vice-versa. In other words, for purposes of READING DATA, the special rules about SUBprograms do not apply.

No so with DEFINitions. If you DEFINE A in the main program, it does NOT carry over into the SUB program. For example:

```
10 DEF A=10 :: C=A :: PRINT C
20 CALL TEST(C) :: PRINT C
30 SUB TEST(C) :: C=A :: SUBEND
```

This little program will first print 10 from line 10 where C is set equal to A, which is DEFINed to be 10. It will then print 0 as inside SUBprogram TEST, A is not DEFINed so it is zero. In the same manner, if you DEFINE something inside a SUBprogram, that DEFINition does not operate outside of that SUB.

Enjoy!

continued from page 2

together. This too is in many cases inconvenient due to the need to have the two computers in the same place at the same time. The method I use to get files written at home on the TI99/4A onto the IBM at work is to send the file as mail to myself on the BBS. I then log on to the BBS with a modem on the IBM and read the file. This is easiest way to transfer files from the TI99/4A at home to the IBM at work. All BBS members are welcome to use the BBS in this way. There is ample disk space on the mail disk at most times for this type of use, especially if members delete their mail files once successful downloading has been confirmed.

Yet another service provided by the BBS is getting those last minute files to the TND editor. Files for the club newsletter can be prepared off line with your favourite text editor in D/V 80 format, then sent as mail to the EDITOR. The BBS allows the file to be uploaded at a time convenient to the sender and read at a time convenient to the newsletter editor (Please note the cut-off dates on the back cover of each TND. ED).

New members are welcome on the BBS, just contact SYSOP Ross Mudie at TISHUG meetings or phone (02) 456 2122. The BBS number is (02) 456 4606 and it is available 24 hours, 7 days a week.

# Tips from the Tigercub #61

by Jim Peterson, Tigercub Software, USA

156 Collingwood Ave.  
Columbus, OH 43213

My stock of Tigercub Software catalogs is depleted and it would not pay me to reprint it. Therefore I have released all copyrighted Tigercub programs, except the Nuts & Bolts Disks, for free distribution providing that no price or copying fee is charged. All of my Tigercub programs have been added to my TI-PD library and are cataloged, by category, in Supplement #8.

My three Nuts & Bolts disks, each containing 100 or more subprograms, have been reduced to \$5.00. If I run out of printed documentation, it will be supplied on disk.

My TI-PD library now consists of 419 disks of fairware (by author's permission only) and public domain, all arranged by category and as full as possible, provided with loaders by full program name rather than filename, Basic programs converted to XBasic, etc. The price is just \$1.50 per disk(!), post paid if at least eight are ordered. TI-PD catalog #3 listing all titles and authors, is available for \$1 which is deductible from the first purchase.

This little program will not do any of the fancy things that the sophisticated poster programs do, but it may do a few things they do not.

First key in this fontmaker.

```
100 DISPLAY AT(3,1)ERASE ALL:"Filename? DSK" :: ACCEPT
  AT(3,14)BEEP:F$
110 OPEN #1:"DSK"&F$,OUTPUT
120 FOR J=32 TO 126 :: CALL CHARPAT(J,C$):: CALL HEX_BIN
  (C$,B$):: FOR K=1 TO 64
130 IF SEG$(B$,K,1)="0" THEN CH$=CH$&CHR$(32)ELSE CH$=CH
  $&CHR$(42)
140 NEXT K :: PRINT #1:CH$ :: CH$="" :: NEXT J :: CLOSE
  #1 :: STOP
150 SUB HEX_BIN(H$,B$):: HX$="0123456789ABCDEF" :: BN$=""
  0000X0001X0010X0011X0100X0101X0110X0111X1000X1001X1010X1
  011X1100X1101X1110X1111"
160 FOR J=LEN(H$)TO 1 STEP -1 :: X$=SEG$(H$,J,1)
170 X=POS(HX$,X$,1)-1 :: T$=SEG$(BN$,X*5+1,4)&T$ :: NEXT
  J :: B$=T$ :: T$="" :: SUBEND
```

This program reads the hex code of each character from ASCII 32 to 126, converts it to a 64-byte binary string of 0's and 1's, then changes each 0 to the blank ASCII 32 and each 1 to a printable character, and saves the result to a file of patterns to print characters 8 spaces wide by 8 spaces high.

The 42 in line 130 creates characters composed of asterisks. Change it to J and the characters will be composed of themselves - the A will be made up of A's, etc. Or, check your printer manual and substitute one of the special graphic symbols in ASCII 224 - 255.

The character patterns are designed from the hex codes in memory, so you can first merge in a reidentified char set such as a CHARAL file or one of the fonts in my Nuts & Bolts disks or in my 127 Screen Fonts disk. Create as many fonts as you want, then key in this poster maker program.

```
100 OPEN #1:"PIO",VARIABLE 136 :: PRINT #1:CHR$(27)&"@";
110 DIM CH$(94):: Q,H=1 :: W,SP=8 :: DB$,SU$="N" :: D$,E
  $="Y" :: GOTO 150
120 F$,CH$( ),J,Q$,M$,FLAG,OUT$,A$,S,SS,PC$,H,T$,L,A,X,K,
  T,X$( ),SK,ST,DD
130 CALL KEY :: CALL SOUND
140 !@P-
150 DISPLAY AT(3,4)ERASE ALL:"QUICK & DIRTY POSTERS" ::
  DISPLAY AT(5,7):"by Jim Peterson"
160 DISPLAY AT(12,1):"Font file? DSK" :: ACCEPT AT(12,15)
  BEEP:F$ :: ON ERROR 170 :: GOTO 180
```

```
170 GOSUB 680 :: RETURN 160
180 OPEN #2:"DSK"&F$,INPUT :: FOR J=1 TO 94 :: LINPUT #2
  :CH$(J):: NEXT J :: CLOSE #2 :: GOTO 190
190 DISPLAY AT(3,1)ERASE ALL:"Load download font? Y/N N"
  :: ACCEPT AT(3,25)SIZE(-1)VALIDATE("YN")BEEP:Q$ :: IF Q$
  ="N" THEN 230
200 ON ERROR 210 :: DISPLAY AT(3,1)ERASE ALL:"Filename?
  DSK" :: ACCEPT AT(3,14):F$ :: OPEN #2:"DSK"&F$,INPUT ::
  GOTO 220
210 GOSUB 680 :: RETURN 190
220 LINPUT #2:M$ :: PRINT #1:M$ :: IF EOF(2)<>1 THEN 220
  ELSE CLOSE #2
230 IF FLAG=1 THEN 260 :: FLAG=1
240 ON ERROR 250 :: DISPLAY AT(3,1)ERASE ALL:"Output fil
  e? DSK" :: ACCEPT AT(3,17):OUT$: :: GOSUB 670 :: GOTO 260
250 GOSUB 680 :: RETURN 240
260 DISPLAY AT(3,1)ERASE ALL:"(1) PICA:"(2) ELITE:"(3)
  CONDENSED":STR$(Q):: ACCEPT AT(6,1)SIZE(-1)VALIDATE("12
  3"):Q
270 IF Q=1 THEN S=80 :: A$=CHR$(18):: GOSUB 640 :: GOTO
  300
280 IF Q=2 THEN S=96 :: A$=CHR$(27)&"B"&CHR$(2):: GOSUB
  640 :: GOTO 300
290 S=136 :: A$=CHR$(15):: GOSUB 640
300 DISPLAY AT(3,1):"Char width 1, 6, 7 or 8? "&STR$(W):
  : ACCEPT AT(3,26)SIZE(-1)VALIDATE("1678")BEEP:W :: SS=IN
  T(S/W)
310 DISPLAY AT(3,1)ERASE ALL:"double width? "&DB$
320 ACCEPT AT(3,15)SIZE(-1)VALIDATE("YN")BEEP:DB$
330 IF DB$="Y" THEN SS=INT(SS/2):: S=S/2 :: A$=CHR$(27)&
  "W"&CHR$(1):: GOSUB 640 ELSE A$=CHR$(27)&"W"&CHR$(0)::
  GOSUB 640
340 DISPLAY AT(3,1)ERASE ALL:"Double-strike? "&D$ :: ACC
  EPT AT(3,16)SIZE(-1)VALIDATE ("YN")BEEP:D$
350 IF D$="Y" THEN A$=CHR$(27)&"G" :: GOSUB 640 ELSE A$=
  CHR$(27)&"H" :: GOSUB 640
360 IF Q>1 THEN E$="N" :: GOTO 380 ELSE DISPLAY AT(3,1)
  ERASE ALL:"Emphasize? "&E$ :: ACCEPT AT(3,12)SIZE(-1)VAL
  IDATE("YN")BEEP:E$
370 IF E$="Y" THEN A$=CHR$(27)&"E" :: GOSUB 640 ELSE A$=
  CHR$(27)&"F" :: GOSUB 640
380 IF DB$="Y" OR E$="Y" THEN 410
390 DISPLAY AT(3,1)ERASE ALL:"Superscript? "&SU$ :: ACCE
  PT AT(3,14)SIZE(-1)VALIDATE("YN")BEEP:SU$
400 IF SU$="Y" THEN A$=CHR$(27)&"S"&CHR$(0):: GOSUB 640
  ELSE A$=CHR$(27)&"T" :: GOSUB 640
410 IF W=1 THEN 430 :: DISPLAY AT(3,1)ERASE ALL:"Spacing
  ? "&STR$(SP)&" /72"
420 ACCEPT AT(3,10)SIZE(-3)VALIDATE(DIGIT):SP :: IF SP>
  127 THEN 420 ELSE A$=CHR$(27)&"A"&CHR$(SP):: GOSUB 640
430 PRINT #3:PC$:: PC$="" :: IF W=1 THEN 450
440 DISPLAY AT(3,1)ERASE ALL:"Multiplied height? "&STR$(
  H):: ACCEPT AT(3,20)SIZE(-1) VALIDATE(DIGIT):H
450 DISPLAY AT(12,1)ERASE ALL:"MAXIMUM LENGTH";SS;"LET
  TERS" :: LINPUT T$ :: L=LEN(T$):: IF L>SS THEN 450
460 IF W>1 THEN 470 :: T$=RP T$(" ",(SS-L)/2)&T$
  :: PRINT #1:T$ :: GOTO 510
470 FOR J=1 TO LEN(T$):: A=ASC(SEG$(T$,J,1))-31 :: FOR
  K =1 TO 57 STEP 8 :: X=X+1 :: X$(X)=X$(X)&SEG$(CH$(A),
  K,W) :: NEXT K :: X=0 :: NEXT J
480 T=(S-L*W)/2
490 FOR J=1 TO 8 :: X$(J)=RPT$(" ",T)&X$(J):: NEXT J
500 FOR J=1 TO 8 :: FOR K=1 TO H :: PRINT #1:X$(J)::
  NEXT K :: NEXT J
510 DISPLAY AT(3,1)ERASE ALL:"OK? Y/N Y": :: ACCEPT AT(3,
  9)SIZE(-1)VALIDATE("YN")BEEP:Q$ :: IF Q$="N" THEN 540
520 IF W=1 THEN PRINT #3:T$ :: SP=8 :: GOTO 600
530 FOR J=1 TO 8 :: FOR K=1 TO H :: PRINT #3:X$(J)::
  NEXT K :: X$(J)=" " :: NEXT J :: GOTO 600
540 FOR J=1 TO 8 :: X$(J)=" " :: NEXT J
550 DISPLAY AT(3,1)ERASE ALL:"(R)edo last line?":"(S)
  tart over?":"Choice? R/SR" :: ACCEPT AT(5,13)SIZE(-1)
  VALIDATE("RS")BEEP:Q$
560 sF Q$="S" THEN 590 :: GOSUB 650
570 CLOSE #3 :: OPEN #3:"DSK "&OUT$,INPUT
580 LINPUT #3:M$ :: PRINT #1:M$ :: IF EOF(3)<>1 THEN
  580 ELSE CLOSE #3 :: GOSUB 670 :: GOTO 620
590 CLOSE #3:DELETE :: GOSUB 670 :: GOTO 620
600 DISPLAY AT(3,1)ERASE ALL:"Skip how many lines? " ::
  ACCEPT AT(3,22)VALIDATE(DIGIT)BEEP:SK :: FOR J=1
  TO SK*8/SP :: PRINT #1 :: PRINT #3 " " :: NEXT J
610 DISPLAY AT(3,1)ERASE ALL:"More? Y" :: ACCEPT AT(3,7)
  SIZE(-1)VALIDATE("YN")BEEP:Q $ :: IF Q$="N" THEN CLOSE
  #3 :: STOP
```

```

620 DISPLAY AT(3,1)ERASE ALL : "Load new font? N" ::
ACCEPT AT(3,16)SIZE(-1)VALIDATE("YN")BEEP:Q$ :: IF Q$=
"Y" THEN PRINT #1:CHR$(27)&"@" :: GOTO 150
630 DISPLAY AT(3,1)ERASE ALL : "Change codes? N" ::
ACCEPT AT(3,15)SIZE(-1)VALIDATE("YN")BEEP:Q$ :: IF Q$=
"N" THEN 450 ELSE 260
640 PRINT #1:A$:: PC$=PC$&A$ :: RETURN
650 DISPLAY AT(3,1)ERASE ALL BEEP:"Set printer
to top of page":"and press Enter"
660 CALL KEY(O,K,ST):: IF ST=0 THEN 660 ELSE RETURN
670 OPEN #3:"DSK"&OUT$,VARIABLE 136,APPEND :: RETURN
680 CALL SOUND(1000,110,0,-4,0):: DISPLAY AT(23,1):
"CANNOT OPEN THAT FILE!" :: FOR DD=1 TO 100 :: NEXT DD
:: RETURN

```

This program asks you for one of your font files. Next it allows you the option of downloading special characters to your printer, if you have such a file on disk. Then you are asked for an output filename; this is necessary because the program rapidly uses up available string storage memory.

Then you are taken through the various printer options. You also have a character width choice of 1, 6, 7, 8. The normal screen font uses only 5 of the 8 pixels of width, so you can select a width of 6 or 7 to get more letters on a line. If your font file used a wider char set, be sure to allow for spacing. If you select 1, you will print a line in the normal printer font.

You are also asked for the line spacing, in 1/72" increments. Characters are normally 8 lines high, but you have the option to print each line multiple times for tall characters or, with closer line spacing, for denser print. Try 3/72" with superscript multiplied by 3, or 5/72" with a solid block graphic character with triple printing.

Finally, you are shown the maximum number of characters according to your options, from 5 double-width 8-wide to 22 compressed 6-wide; you input a line and see it printed. It will be automatically centered.

If you are satisfied with it, the line is saved to disk, you specify the number of lines (8/72" spacing) to skip, and you are taken thru the options (including a new font) for the next line. The previous selections become the default options, so you can skip through quickly.

If the line is not satisfactory, you have the option of advancing the paper to the next page and reprinting the poster up to that point from the disk file and then continuing. Now, here is the neat part. When you have finished your poster, you can print as many copies as you want.

Just key in this program -

```

100 OPEN #1:"PIO",VARIABLE 136 :: PRINT #1:CHR$(27)&"@"
110 DISPLAY AT(12,1)ERASE ALL:"Filename? DSK" :: ACCEPT
AT(12,14)BEEP:F$ :: OPEN #2: "DSK"&F$,INPUT
120 DISPLAY AT(12,1)ERASE ALL:"Load a download font? Y/N"
:: ACCEPT AT(12,27)SIZE(-1)VALIDATE("YN"):Q$ :: IF Q
$="N" THEN 150
130 DISPLAY AT(12,1)ERASE ALL:"Filename? DSK" :: ACCEPT
AT(12,14)BEEP:F$:: OPEN #3: "DSK"&F$,INPUT
140 LINPUT #3:M$ :: PRINT #1:M$ :: IF EOF(3)<>1 THEN 140
ELSE CLOSE #3
150 DISPLAY AT(12,1)ERASE ALL:"How many copies?" :: ACCE
PT AT(12,18)VALIDATE(DIGIT):N :: FOR J=1 TO N
160 DISPLAY AT(12,1)ERASE ALL BEEP:"position paper, pres
s Enter"
170 CALL KEY(O,K,S):: IF S=0 THEN 170 ELSE CALL CLEAR
180 LINPUT #2:M$ :: PRINT #1:M$ :: IF EOF(2)<>1 THEN 180

```

You will have to reposition the paper after each one.

The poster maker program was written for my Gemini 10X and I have not tried to offer options for other printers, since I do not have them available for

testing. However, I think that these are the essential changes for the Epson standard.

```

260 DISPLAY AT(3,1)ERASE ALL:"(1) PICA":"(2) ELITE":"(3)
COMPRESSED PICA":"(4) COMPRESSED ELITE":STR$(Q):: ACCEP
T AT(7,1)SIZE(-1)VALIDATE("1234"):Q
270 IF Q=1 THEN S=80 :: A$=CHR$(18):: GOSUB 640 :: GOTO
300
280 IF Q=2 THEN S=96 :: A$=CHR$(27)&CHR$(77):: GOSUB 640
:: GOTO 300
290 IF Q=3 THEN S=132 :: A$=CHR$(15):: GOSUB 640 ELSE S=
160 :: A$=CHR$(15):: GOSUB 640
670 OPEN #3:"DSK"&OUT$,VARIABLE 160,APPEND :: RETURN

```

And these changes should make compressed elite available on the Gemini SG10 in Star mode.

```

260 DISPLAY AT(3,1)ERASE ALL:"(1) PICA":"(2) ELITE":"(3)
COMPRESSED PICA":"(4) COMPRESSED ELITE":STR$(Q):: ACCEP
T AT(7,1)SIZE(-1)VALIDATE("1234"):Q
270 IF Q=1 THEN S=80 :: A$=CHR$(18):: GOSUB 640 :: GOTO
300
280 IF Q=2 THEN S=96 :: A$=CHR$(27)&"B"&CHR$(2):: GOSUB
640 :: GOTO 300
290 IF Q=3 THEN S=136 :: A$=CHR$(15):: GOSUB 640 ELSE S=
160 :: A$=CHR$(27)&"B"&CHR$(4):: GOSUB 640
670 OPEN #3:"DSK"&OUT$,VARIABLE 160,APPEND :: RETURN

```

Other modifications should be fairly easy. The variable S contains the maximum number of characters per line. In lines 310-400, the option is turned on if it is selected, turned off if it is not.

Almost out of memory,

Jim Peterson

continued from page 7

```

MOV R3,R3          Check for no arguments
JEQ END           If none, exit early

LI R5,>8300        First Argument Identifier
CLR R8            Up counter for argument number
LI R10,2          Initial number of scrolls
LOOP2 BL @SCROLL

CLR R6            To ensure last bit will be clear
MOVB *R5+,R6     Put current ARGID value in R6
SRL R6,7         Byte to word and multiply by 2
MOV @LOOKUP(R6),R6 Get the address for text
LI R7,BUFFER     Destination for text
LI R2,19         Number of bytes in text
LOOP1 MOVB *R6+,*R7+ Move text byte by byte
DEC R2          Decrement loopl counter
JNE LOOP1       Finished yet?

INC R8           Up counter for argument numbers
SLA R8,1        Multiply by 2 for word offset
MOV @NUMBER(R8),@ARGNOV Get number text
SRL R8,1        Divide by 2 back to original number

LI R0,736       Start position on screen
LI R4,ARGNO     Start of text in CPU RAM
LI R2,31        Number of bytes to write on screen
BL @PRINT

LI R10,1        Ready for scroll 1 line
DEC R3          Finished all arguments yet?
JNE LOOP2      No, do the next

LI R10,1        For scroll 1 line
BL @SCROLL

END CLR R0
MOVB R0,@>837C Clear STATUS byte
LWPI >83E0      Load the GPL register Work Space
MOV @SAVRTN,R11 Restore saved return address
RT             Return to calling program

END

```

## Handy Tips for the TI99/4A

Here are a few tips for beginners (good for experienced programmers, too!) from old newsletters. Re-typed from the Spirit of 99 Newsletter of May, 1990 by Bob Relyea.

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

2. If you want to disable the Quit key (Fctn +) type in CALL INIT :: CALL LOAD(-31806,16) and then enter. To re-enable the Quit key use CALL LOAD(-31806,0). You must have Extended Basic.

3. If you are going to save a program to tape and type OLD CS1 do not panic. Press FCTN and E together and then press (enter). This will take you out of the tape loop.

4. You do not have to enter a line number in TI Basic or Extended Basic. Before you start enter NUM n(1),n(2) where n(1) is the starting number and n(2) is the desired increment.

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

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

7. If you want or need to renumber the lines in a program either to make it neater or make room for new lines, you do not have to renumber them individually. Just enter the command RES n(1), n(2) for Resequence, where n(1) is the beginning number you desire and n(2) is the increment.

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

9. Have you ever pressed ERASE by mistake and lost the whole line? Do not panic and do not hit (enter). Instead press FCTN ? and (enter). Your line will still be intact.

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

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

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

13. You can add comments after a GOSUB or GOTO. They will not interfere with the program and you do not need a REM or !.

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

15. If you have the TE II cartridge and the speech synthesizer type in the program on page 37 of the TE II manual. Try entering strings of K's, Q's, U's, W's, J's or X's for different sound effects. Try mixing them for interesting sounds.

16. If you have Extended Basic and 32K type this

in as the last line of your program: CALL INIT :: CALL PEEK(2,A,B) :: CALL LOAD(-31804,A,B). This will return you to the title screen when the program is ended.

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

18. To speed up loading Infocome games, do not use Extended Basic. Use Mini-Memory or E/A instead. To use these, select the 'load and run' option and type DSK1.BOOT. When this is finished loading, press (enter) until you get the program name, and then type start. On Mini-Memory you will get an error after BOOT loads, but keep pressing (enter) and proceed as above.

The following are a few tips taken from the newsletter LA 99ers TOPICS.

1. Jim Peterson reminds us of a Danny Michael hint:

"... to avoid those lock-ups and other foul-ups that occur when you CALL INIT after you have already CALLED INIT ... use:"

```
CALL PEEK(8198,A) :: IF A<>170 THEN CALL INIT
```

2. From the Rocky Mountain 99er, April, '90 comes:

"... and from the HV 99ers comes this neat tip ... you can eliminate the need to use the arrow keys to go to the actual filename when you load or save a file. The Editor does not care if there are a couple of spaces after a "LF" or "SF" command. So you type in:

```
LF (space)(space)
```

You will find the cursor positioned on the actual filename when it appears on the screen."

3. John Dunning of the EAR 99ers reminds us that:

"When you type in a line number in the command mode and press Fctn +X, that a line is brought up on the screen. If you continue to hold the two keys down the next line appears, and the next, and so on. But did you know that once you have started to bring the lines up you can let go of the Fctn key and just hold the 'X' and the lines will continue to scroll up the screen. Handy is you need to have your right hand free to follow a print-out, etc. It also works with the 'E', 'S' and 'D' keys. ○

## Jenny's Younger Set

Dear Jenny,

My name is Eric. I have been having trouble with my programs. One of my problems has been with "Donkey Kong". I cannot find out how to get the maid in level 2.

Also I do not know how to get past the monster at the top of the mountain in the "Bigfoot" game. I hope that you can find the answers to my questions.

Yours sincerely, Eric Ball.

Dear Eric,

I can help a bit with Bigfoot, but I have not played Donkey Kong and could not find anyone to help on this one either. To capture the abominable snowman in Bigfoot, you first must collect all the gold on the ledges. To do this you throw ropes up to the ledges and climb up these dodging the snowballs coming down. You have to pick up a box first before you can pick up the lumps of gold and when you have picked up all the gold you need to get up to the ledge on which Bigfoot stands when a black rope will appear going upwards. When you start climbing this rope it lowers a cage over the top of Bigfoot and you have him! You just pull down on this rope and Bigfoot is carried away. The game then continues with the next level, which is a bit harder. For your other question of how to get the maid in level 2 of Donkey Kong, I appeal to anyone who can help us to write in and explain how to play this game. Please write in! ○

# Spellbreaker part 3

Copyright 1985 Infocom  
This walk through is by Scorpia, Copyright 1984.

I hope you feel in the mood for a dip, because it is time to do some more splashing around (if nothing else, you are certainly going to be the cleanest Enchanter in the land!). Blorple "WATER". Learn the Snavig spell. To make your life a bit easier, stuff everything but the fish into the zipper and close it. Hmm, a bit dark in here, but not to worry, the Grues will not getcha (they come a little later on...EVIL grin). Save the game.

Go south into the ocean. SPLASH! You dropped the cube again! And that grouper is back, too. As a safety precaution, drop the fish, then Snavig the grouper. How about that, it worked! Now you are a grouper, too! Unfortunately, since you are now a fish, you can not hold the zipper, and it sinks out of sight. Oh well.

Still, there may yet be hope. Swim down until you reach the grouper's nest. Aha! There is the "WATER" cube and the zipper, and another cube sitting in the nest. Of course, you can not do very much right now. So, just wait until you turn back into yourself again. Now, quickly get everything and head for the surface. WHEW! You just made it!

After getting a few lungfuls of air, blorple "WATER". Do not bother drying off, because you are going to be wet again pretty soon. Open the zipper and get your burin and spell book. Write "LIGHT" on the cube from the nest. Head north to the Oubliette.

Learn the Liskon spell, then put your book in the zipper and close it. Liskon yourself. Ooof, what a strange feeling! Now, enter the outflow pipe. You will be swept along west with the flow of water. As this happens, you will pass by a cubical object...yep, it is another cube! Get that and keep going west until you come to a spot where you can climb out.

Go up. Sonovagun! You are back in the Ruins Room! Now, at last, you can put the "WATER" cube away. Write "CHANGE" on the cube you just found. Get your spell book and learn Tinsot, then blorple "LIGHT". Go west, and you are on the side of a volcano. Pretty hot around here! And sizzling lava bits are flying about all around you. One lands practically at your feet, but it is too hot to touch. Tinsot the fragment, then take it. Put that and the "LIGHT" cube in the zipper.

Blorple "CHANGE". You are in a bizarre room where the exits keep changing on you. Go north to the Bare Room. Here you find a superbly-crafted compass rose. Get that, and blorple "CHANGE" again. This time, go west to the Carving Room. Examining the wall you see a carving there that closely resembles the compass. Could it be...? Put the compass in the carving. Sure enough, a small opening just appeared in the north wall!

Do not go running off just yet. First take the compass! Without that, you will not get very far. Now, through the hole and into the next room. Here you see eight walls, each with a direction rune written on it. The trick is to touch the compass to one of the runes, thereby creating a new opening. You can only go in a particular direction once. If you check the compass, you will see that each time you go in one of the directions, that arm on the compass turns to lead. Also, some of the runes on the wall are in lead, too. Do not touch those.

Ok, let's move along here. Touch the compass to the NorthWest rune. Go through the hole that appears. Touch the compass to the West rune, and go through the hole. Now, touch the NorthEast rune and go through that opening. Ah! Here is a room with an opening already created...but it is plugged up!

No matter, a quick Rezrov, and that obstacle is gone. Go through the hole on the west wall. Well, well. Here is another cube (what else?). Write "VOID" on that cube (by the way, you can drop the compass, as you do not need it any more). Blorple "VOID", which takes you to a very strange room indeed!

Go south, and you will find yourself on a plain, next to a large, green-eyed rock. Here is a good place to save, as you are about to play a little game of tag with that brown-eyed rock yonder, the one with the cube on its back. You can not walk over to it; the surface

is much too slippery. Even if you could, that would not help much, as the rock will move away from you. So you are going to need some form of transportation. Get the lava fragment and offer it to the green-eyed rock. After it digests this little tidbit, it will allow you to climb on its back, and will take you wherever you want to go.

If you have played this part of the game before, you may be wondering if it really is possible to catch up to the brown-eyed rock. Well, it IS possible, and, in fact, actually quite easy (honest!).

The first thing to do is map out the area completely. You will see that the plain is composed of 15 locations. They almost form a square, except for the upper left-hand corner, which comes down on a diagonal. This is what allows you to catch up to the brown-eyed rock. It is best to make a large diagram of the area, and use two small objects (such as a dime and a penny) to represent yourself and the brown rock, moving each object on the diagram as you and the rock move in the game.

OK, now the fun begins. Tell the green-eyed rock to move west. The brown-eyed rock will move randomly either north or south. Do not worry about that; it does not matter in the least (just keep track of it on your diagram). Next you move to the northeast. The brown-eyed rock will move randomly again, either north or south. Move east. Once you are in this position, the brown-eyed rock is effectively trapped. Start going south. The brown-eyed rock will also go south (usually) until it reaches the bottom of the diagram. At that point, it will go west (most of the time). However, when it does so, it will be right below you, and you can move south onto the same square! If the rock happens to go north instead of west, no problem; you can catch it by going east.

All right, now just jump over to the brown-eyed rock, take the cube from its back, and climb down. Write "BLACK" on the cube, learn Snavig, then blorple "BLACK". Hmm, certainly an appropriate name for THIS cube...you are in total darkness. Trial and error will show you the only way out is down, which takes you to a dark cave, where your burin is not doing such a great job of providing light.

As a matter of fact, you \* do not \* want any light here, so put the burin in the zipper and close it. Now, go down again. Shudder! You are in a Grue cave, and there are plenty of them lurking in the darkness. You do not want to be noticed, so snavig grue. Whew! Now you can go investigate that strange pool of light.

Here you see a pillar with a cube on top. Fortunately, it is not hard to reach. First go down into the pool, then climb the pillar and take the cube. Now climb down, and go up (make sure you DO NOT say "climb up", or you will be back up the pillar again!). Go up again and you are away from the Grues. After that, just hang around until you are yourself again, then get your burin and write "FIRE" on the cube.

"Spellbreaker" is copyrighted 1985 by Infocom.

This walkthru is copyrighted 1985 by Scorpia, all rights reserved.

continued from page 4

access to the various options, including a line font editor, Page Pro Stripper (removes blank space from around a Page Pro picture), Page Pro Enlarger (enlarges or reduces Page Pro pictures in either or both directions), and Page Pro Flipper (flips pictures). There are also options to catalogue the disk, and to load Page Pro. Included on the disk are samples of HIRESS pictures which show what the software is capable of. The price of the package is \$17.00.

Page Pro Templates #1 to #12 mentioned in last month's TND are priced at \$8.00 for each volume (comprising 1 to 3 SSSD disks).

Rock Runner, the new game from Asgard should be available at the meeting. The price will be \$15.00. ◦

# TI\*MES Utility Catalogue

by Stephen Shaw, England

>ASSEMBLY UTILITIES BY COOK AND JOHNSON: TWO DISKS of utilities as source code to incorporate into your own assembly programs, such as BLWPCLR, BLWPGCHAR, or DISPLAYAT. ("ASSEMBLY MADE EASY").

>BEAXS by Paolo Bagnaresi of Italy. "Better Editor Assembler" for XB loading. TI's EdAs amended for loading with XB module, with one or two utility files added.

>C99 by Clint Pulley. THE language! FIVE DISKS PLEASE which contain the compiler and a number of library and demonstration files, as well as comprehensive documentation. You will also need a book on C. This is a fairly complete implementation of small C. You prepare your c source code with say the TI Writer Editor, then compile using this package, then assemble using TI's Editor Assembler. Then you have a machine code program ready to run! This is version 4.0, sometimes referred to as REL 4.

>C99 REL4 UPDATE: If you already have Vn 2.1, you need only send for this one update disk which contains all the amended files to upgrade your present version.

>C99 TUTOR. From D L Mahler and the Boston Computer Society, this is not so much a tutorial as a large number of sample files, includes brief instructions on use and HOW TO use c99.

>C BITS. FROM MIKE CAVANAGH, a disk containing a complete-ish library for C programs for use with RAG's RAGLIB and Linker Librarian, with a modified control file for same; modified C optimiser, modified C string library file, and a logo-type graphics program- C code-needs two inputs, try for example 5,81

>CATLIB Vn 1.5 by Marty Kröll. A disk catalogue program with a capacity of 123 disks/900 files per data file. Small data files can be merged. Printouts can be of disks, files or standard format, in 1 2 or 3 columns. Printer control characters are replaced, so printouts stay neat. Global search on disk/file names.

>CATCOM by Marty Kröll. This companion disk to CATLIB above allows you to add extra data such as file extensions and file descriptors, to make your disk data base more useful.

>C COPY Vn 1.8 by C Winter. A rapid disk copier. Program is in German but docs are in English. Heed the warnings- the extra speed is at a cost to user friendliness! Some additional features as well as straight copying.

>CO LIST (PLUS TXB). The program by Tony McGovern which will prepare program LISTS 28 columns wide- just like they appear on screen. Ideal for print outs that someone has to key in! Together with a fast action bagatelle type game which has lots of options.

>COMIC 1:  
Animation editor with ENGLISH documentation, allows you to chain up to 100 TI Artist pics together for SMOOTH animation effects in a machine code environment. Includes ExBas loader. Includes sample animations: Two TI Artist instances come to life- a Ghost and Pluto. And if you enjoyed Ray Kazmers picture of Garfield and Odie on page 42 of TI\*MES issue 16, take a look at Ray's animated version!

>COMIC 2:  
Three more animated samples, with XB loader. Bouncing balls, rotating wire frame box, and stretchy lettering.

>COMIC SHOW Vn 4.0 (1988): This is the latest version of COMIC, with English docs by the author, and 3 additional samples of animation. (PPYJAMA demo will not

load with Funlweb- use the load menu supplied on this disk)

>CREATIVE FILING SYSTEM by Mark Beck, THREE DISKS PLEASE, Version 7. Either the best or second best database program for the TI (opinions vary- some put PRBASE first). CFS is full of useful utilities and has math capability. Lots of docs on disk.

>DATABASE1: A commercial database from SPC Software, who have not advertised for ages, and is not apparently on sale anywhere. I have been unable to contact them and assume they have ceased interest in their program. TWO DISKS required. A simple database with several utilities. Perhaps best thought of as a "list processor". Satisfies the majority of my simple database requirements. Full docs on disk.

>DEMO DISK containing: A machine code program giving you key-press access to speech extracted from some TI Modules; an animated space picture; three programs similar to mini-men LINES, called LINES, TRIANGLES and RECTANGLES, and a logo-type program called ROSE, which must be loaded from Funlweb Load Option 3-good inputs for Rose are 13 then 313. AND ALSO a program to print to printer an 8x10 pic of the Mona Lisa. LIGHTSHOW- feed music into tape socket; SCRATCH music maker(music?); and MUSIC DEMO with themes from three modules- Midnite Mason, Cerberus, and Demon Driver.

>DISK UTILITIES Vn 4.12 by John Birdwell. A very complete disk utility which now allows you to change the name of a file when copying ( think carefully about this!), check free space on destination disk BEFORE copying starts, sector edit changes in inverse to make them stand out... etc etc.

You may check a disk for bad sectors (non destructive) and mark out bad sectors from the BitMap without reinitialising the whole disk. Add comments to files, add date to disks. Excellent printout showing location of FDRs and each file segment. Phew! Excellent program.

Plus Q4CAT, which will read 4 disks and print the catalogues in 4 columns of condensed print.

>DSKU/M..The same as the above, but modified for owners of Myarc disk controllers who do not have either an 80 track eeprom or a Geneve.

>DM1000. VN 3.5 Possibly the widest used disk manager and certainly a classic in the TI world. From Canada, this program can deal with any disk controller, and will catalogue, initialise, and rename disks, list,copy, move, delete, protect, unprotect, and view files. Although DM1000 is included on the FUNLWEB disks, you need to order THIS item to obtain the DM1000 documentation.

- \*\*\* The Ottawa group have advised June 88 that versions of DM1000 over Vn 3.5 may not be reliable, and you should not under ANY circumstances use Vn 4.0. You will need this 3.5 disk for the docs, AND will find Vn 3.5 on the Funlweb 4.1 disk. They expect to release a fully tested revision of the program around end 89 (previously end 88!)

>DM1000 SOURCE CODE TO VERSION 3.5. TWO DISKS REQUIRED.

>DUMPIT. A disk which will show you how to dump modules composed only of GROMS ( no ROMS) to disk. We already have them all on disk- this is how it was done. Disk contains DISKO, BUGOUT, and instructions.

>DUTCH DEMO... various graphics routines in machine code, including SKY which is already on our DEMO disk... there is also a fast flashing multi colour mode display, a LINES program which is nothing like the TI Lines program but has lots (LOTS) of graphics made up of lines... and a receding square tunnel to boggle you... lots of source code including for dsrlnk, gplnk and kscan.

>ENHANCED DISPLAY PACKAGE Vn 2.2 from Paragon Computing, programmer C A Provan. TRUE Freeware! IBM style- useful documentation on disk, after that you get what you pay for! Immediately usable and good demo program. This disk contains a program which places machine code utilities into memory for your XB programs to use by means of CALL LINKS.

There is a clock WITH ALARM!, windows, and display commands are amended for both 32 and 40 column screens. There are routines to save and load screen displays, PEEKV, GTEXT, and a much extended and useful CHRSET. Disk contains 15 pages of docs and a good demo program.

IMPORTANT: EDP supplied by me is configured for 50 cycle mains: the clock/alarm runs quite accurately ON 50 CYCLE MAINS. It will NOT run accurately on US 60 Hz mains. The electricity supply in the UK is required to maintain a short term frequency accuracy of 2%. This clock has been measured as better than .5%, well within supply constraints.

>EDP Vn 2.4US: Please request by THIS full name! A slightly unbugged version but with US clock-just runs slowly on UK consoles! The author has kept no record of amendments and nothing too drastic seems to have changed.

>EUROPA. An Extended Basic suite occupying 343 sectors. For each country in Europe, shows the position, gives the car plate (eg GB), the flag, area, population, population density, money, capital... slow but may have some educational value?

>EUROWRITER or TI Writer Vn 2.0, yes, TI had a second version! Now widely used on the continent, as THIS version is FULLY multilingual, with foreign character sets on screen, and command abbreviations in the foreign language. REQUIRES the TI WRITER MODULE, will not function with any XB loader. The really bad news is that if a European user sends you a text file, you will not be able to load it with your Vn 1, as TI handily made them incompatible ( although a European CAN load our Vn 1 files, and is therefore unaware of the problems!).

>FRACTAL EXPLORER Version 3. TWO DISKS REQUIRED. This package enables you to explore the odd graphic world of "real" numbers. Screens are designed in multicolour mode, and CAN BE PRINTED in grey shades or outline on an Epson printer. Screen design takes a little while, but screens can be saved for fast reloading later. Supplied with docs and sample screens. Odd program! but fun.

NB: The source code supplied is commented and can be of real assistance if you are learning TI machine code!

>FUNLWEB Vn.4.21. (AVPC and HRD and Geneve compatible). THREE DISKS! If you do not own this package, wake up and get it now! Undoubtedly the most widely owned and most heavily used program for the TI99/4A.

It started out as an XB loader for TI Writer, but has gone several parsecs beyond that by now. In the Editor alone, you now have an "end of line" warning beep, a ruler at the bottom, a "show directory" which you can print, and the ability to view one file while another resides in memory.

Capable of loading almost ANYTHING in machine code, complete with Assembler, Formatter, and Disk Manager, DISK REVIEW (an enhanced cataloguer), and Sector Editor and.... you cannot live without this one! Compatible with Myarc XB too.

By Tony and Will McGovern of Funnelweb Farm in Australia. Windowed CONFIGURE program for personalising program, Editor has faster MOVE/COPY, with better error handling. Can hold two tab settings at once- switch with command ST. Assembler editor places comment field in lower case.

Disks include a much modified version of DM1000 Vn 3.5 - which is more reliable than the original Vn 3.5.

NB: FUNLWEB 80 column version now available- modified ED and Disk Review files with extra docs. PLEASE SPECIFY 80 COLUMN VERSION WHEN ORDERING. Also requires THREE DISKS as above.

>G -HIGH RES LANGUAGE. Undated, no address, but from Adelaide Australia.

Another language! which is used for bit map graphics, using a basic like structure. Commands available are:

SET, STOP, Variables A to Z, Constants Dec or Hex integers, one array only, single dimension; LET, CLS, GOTO/LABEL, GOSUB/LABEL, RETURN, REM, COLOR, BCOLOR, SCREEN, BOX, FOR/NEXT/STEP, ANGLE(degrees), DRAW, IF/THEN, LOADS, LOAD, SAVES, KEYS, FORMAT, FGND, BGND, TRACE, FILL, PRINT, RND, SIZE, PATTERN, RESTORE, ARC, SIN, COS, WRAPON, WRAPOFF, STORE, CLEAR, INVERT, LINSTYLE, DISPLAY.

Docs are perhaps too brief but there is a sample program to inspect.

>GRAPHIC LABEL MAKER by S J McWatty. Modified by R J Bailey. A FULL disk of small graphics to be used with a program that lets you type in what you want on a label, with optional centering of text and optional addition of a small graphic on the left hand side. Seems to use labels about 3 1/2 by 7/8ths of an inch. Includes a graphic editor by R Felton so you can create your own label graphics.

>GLM GRAPHICS: THREE DISKS of ready to use graphics for the above label maker.

>GLM COMPRESSED GRAPHICS: TWO DISKS of archived and compressed graphics for the above label maker, complete with Archive to uncompress and unpack the files. Includes many graphics from the above disks, but also many more. Have ten to fifteen blank initialised disks to hand before you start to unpack! and REMEMBER that the disk system will only tolerate 126 files per disk, regardless of disk format. Use of MOOPY on UTIL16 is recommended on the final disks.

>GPL MANUAL by Art Green. TWO DISKS.

>GPL DISASSEMBLER by Art Green. REQUIRES ability to run a machine code program irrespective of module, eg gramcracker, widget, etc.

>GPL ASSEMBLER by Art Green, TWO DISKS. REQUIRES GRAMCRACKER or similar device.

>HBMPRINT. This disk is a utility to let you print out data saved with the Home Budget Management module. Also for Household Money Management, the UK version, which we have on disk.

>INVENTORY MANAGEMENT: TI PRODUCT PHD5024- formerly sold for US\$59! - REQUIRES the PRK or STATS modules OR library disk MODUTIL. NO Docs- not sold by TI-UK. First create blank IF70 and IF80 files using the file INIT. INVENTORY is a blank PRK file. CONVERT1 transfers data from the IF80 file to the PRK file. CONVERT2 converts from the PRK file to the IF80 file. UPDATE transfers from IF70 (Main) to IF80 (Stock) file. REPORT1 and REPORT2 provide printouts. Copy master disk and use copy! The Basic files provide a good lesson in how to use the "hidden" calls of the PRK module.

>JAPANESE.... Don Shorock SAMPLE DISK... Don publishes a lengthy catalogue of "multiple choice" programs, largely language based. This one is for JAPANESE and includes the Kanata text. If you are interested in using your TI with a multiple choice quiz to learn languages, OR have an interest in Japanese text, why not have a look at this one.

>JBM103. A bitmap graphics utility for TI-Ex Bas. Allows you to draw bit map pics in ExBas using easy CALL LINK commands. Set single pixels, draw lines and circles. Screen dump to Epson printer via PIO. Switch between 32 column mode and bit-map mode. Pictures can be saved to disk and reloaded. Compatible with TI ARTIST ( and hence also MAX/RLE and GRAPHX when their pictures are saved as TI Artist format!).

>BALLS. A JBM103 demo disk containing 3 ExBas programs written for JBM103 which produce fractal globes, together with two sample pics, plus three

animation sequences produced with Comic Show 4.0, and sample Comic Show command file.

>JP GRAPHICS (Vn 3.1) by J P Morin. Ed As or XB graphics program. A remarkable bit map graphics program written in FORTH for fast speed but using single key presses to produce complex designs. NOT AN EASY PROGRAM TO USE but very powerful. Includes a LOGO mode with turtle graphics - the FORTH split-screen is used to allow you to instruct the turtle with LOGO like commands, eg PD 5 FD 12 RT etc etc. Knowledge of Forth not required. Forth not required - a complete program. Occupies two disks: one for the program and one for the documentation.

>LINKER by RA GREEN. VN 3. This program changes DF80 object code to memory image format- with options on locating the image, a compact output, and a clever way to resolve unresolved references. A library file is supplied with common references and the program incorporates just what it needs into the image. Also see the next disk below.

Due to lack of room, the alternative TI Writer and loader from Johnson and Ballman of Florida has been removed from this disk, and as there are better alternatives, pruned from the library.

>LINKER LIBRARIAN by Tom Bentley. Perhaps essential for the above! This program allows you to build up your own libraries quite easily, and a c99 library file is included- when you use LINKER to create a memory image of a C99 program, the LINKER will search the library for the c99 library entries the program needs- for instance, PRINTF would be AUTOMATICALLY loaded. Neat.

(See also C Bits in an earlier Util file).

>LISP 99. Version 2.0. Another language for the TI. Documentation is thin. Quite a lot of commands available. Basically an implementation of MacLisp but with the Property List commands removed. The suggested book is readily orderable.

>LOGOUTIL by E P Rebel. TWO DISKS REQUIRED. The two utilities are also on UTIL-22 but here have docs and source code. Allow you to make a LOGO2 disk file auto-start, and to dump to printer character and tile definitions. Plus REFDEF for MM and EA only, which gives you details of redef tables for DF80 files.

>MAC-LABELS- a disk of label utilities by Mike and Ed Machonis, primarily for 1 7/16 inch labels. Lots of formats and designs. PLUS a program to print function strips- your own design or ready files for popular modules and programs. You can even print a disk directory to a label... 340 sectors used!

>MASSCOPY: (Vn 3.25) Fast disk copier. ExBas+32k. 1 to 3 drives in any format. Can copy to drives 2 3 in one go. Said to be able to copy to different formats. With 2 drives copies a disk in under 3 minutes. Can be used with one drive alone. Copies entire disk: files NOT selectable. Also able to utilise Foundation Myarc 128k cards. Some glitches but copies well. By Steven Lawless. \*This disk ALSO contains: Hi Res sketching program for mini-memory, and some programs for your printer: mazes and pin-ups. Also a Foundation 128k transfer utility. (Upon request can be supplied as Myarc RAM-Disk)

>MC-1A : Machine code utilities for Extended Basic : SOURCE CODE. Utilities include: TI Disassembler, DSRLINK, Screen graphics dump, centralising text on screen, print LOGO Procedures to printer (especially for LOGO One owners!), Quicksort, save and recall screen data, horizontal scroll, access to VDP, and a Statistics utility.

>MC-1B : OBJECT code for the above, ready to be CALL LOADED from ExBas. PLUS Documentation. Plus Demonstration programs. PLUS a routine to print Gothic characters on your printer.

>MC-2 :Source code, object code (for ExBas) and documentation for: Cursorflip, makes the cursor behave like a that on a professional word pro when you back space!; MM>DISK- A utility to save the contents of MM to disk and then reload them (works with DATA as well as m/c programs); A disk directory utility and disk label print utility; and a utility to create a file of strings in the 32k ram. Also a tiny routine to give you access to the BIG character set (as on the title screen), with demo. OTHER useful utilities too : no more room ....

>MC-3 : Mainly a data base program for EDITOR ASSEMBLER or MINIMEMORY modules only. The machine code allows you to make use of the system memory as a data storage area, and by means of CALL LINKS you may transfer a data file between disk and ram, transfer a data record between ram and your Basic program, sort data( bubblesort or quicksort). Up to 300 records can be stored, and are placed on disk as DV80 files.

Also on the disk is: DEFTABLE, an Basic/XB program for EA,MM or XB, which lists the first/last addresses in each memory segment and also the DEF table- name, address, address contents. ALSO a program to create an auto-load type of routine for machine code programs. It loads with Opt 5 of EA and allows you to select from a screen menu M/c programs on the disk for either Opt 3 OR Opt 5. NOT ONLY BUT ALSO... a game for TI Writer...

>MORSE code generator and keyer by Ross Mudie- allows text to be typed on screen while generating CW at 5 to 40 wpm- 16 line screen buffer- audio output via TV and also keys the cassette remote for transmitter operation. Auto repeat CQ. Hardware instructions in file DISKMORSE.

>MUDIE 1. Perhaps more of a tutor, but a useful utility- a CALL LINK(K.. to enhance CALL KEY, a utility to read CorComps clock, and TIs speech. Has commented source code.

>MUDIE 2. 40 column utilities. With commented source code. Read and learn.

>MUDIE 3- disk 86/3, which follows on from the earlier pair of disks we have. This disk contains object code for an enhanced 40 column mode for ExBas program using CALL LINKS- added since 86/2 are sounds and colours. There is also a short routine to detect (and warn of) alpha lock key position, a routine to load machine code into high ram from exbas, and a routine to add POKEV,PEEKV and POKER to ExBas, allowing you to inspect and change VDP Ram and to change VDP registers.

> MULTIPLAN V 4.00 ROM VN by Art Green. REQUIRES MULTIPLAN MODULE. You also need a Multiplan manual OR any textbook- Multiplan on the TI is virtually identical to all other (early) versions. This disk is ESSENTIAL for all TI Multiplan users- Art has thoroughly rewritten the INTERPRETER file including use of PAD for a REAL increase in speed.

> MULTIPLAN V 4.00 GRAM VN by Art Green, as above, but configured for Millers GramKracker or any device able to handle GK files.

>MULTIPLAN SYLK CONVERTORS...by Peter Walker (UK). The programs on this disk enable you to transfer data between a Multiplan Sylk file and a TI Writer DV80 file. Once you have set up your spreadsheet with dummy data, transfer it to DV80 format to see how it looks - now you can work out how to transfer data from your own programs into a suitable DV80 file which you can transfer back to Multiplan Sylk format.

A separate version is included for owners of Mechatronic's version of Extended Basic. Runs from XB. Also on this disk is TIMP PRINT by JB Mathis, Version 1.6, which contains various Multiplan sheets with filled cells to transfer to an empty cell in your own sheet to send printer control codes.

>MYARC XB UTIL: The files on this disk enable you to use Myarc XB ONLY to load Graphx pics in XB, dump a

continued on page 21



# Beginning Forth - part 3

by Earl Raguse, UGOC, CA USA

## EDITOR CHANGES

You have discovered by now, I am sure, that the Editor is not very friendly when it comes to moving the cursor around. There is fix for that which I will give you this month. Why did I not give it to you last month? Because you will now appreciate it much more than if you had it all along.

The lack of auto repeating keys was the one thing about Forth that bugged me, and I had vowed that, as soon as I understood enough about Forth, I fully intended to modify the Editor to make the cursor repeat. Well, as they say (I do not know why) "All things come to those who wait". I guess I waited long enough because the September 1984 issue of "THE SMART PROGRAMMER" carried Pete Korner's (LA 99ers) revised Editor. The article had Screens #34 thru #38 and a #41. After much study I concluded that the changes occurred on only 3 screens including the new #41 which is included herewith. The other changes are as follows:

Screen #34

Change line 1 to read: (SCREEN EDITOR 09JUL82 LCT)  
O CLOAD RKEY

Screen #38

xdd 29 LOAD to line 0 Change line 2 to read: : VED  
BOX SWAP CLS LISTL ! .CUR BEGIN RKEY CASE

That is it, except for Screen #41, which just happens to be vacant on the original Forth Systems disk. I recommend that you copy Screens #34 thru #38 from your Forth System disk, make the above edits, add Screen #41, LOAD them (ie 34 LOAD, never mind the "not unique" jazz) and see if it works. If it does you can safely save your screens to your previously made Working Forth disk. Now boot it, FORGET BOX to purge the dictionary of EDITOR, then go through the process of loading -EDITOR etc. as I described last month. Now do a BSAVE as before. Its a JOY! Thank you Pete Korner!

## STACK MANIPULATION

A major feature of Forth is that it uses the stack concept for passing values from one part of the program to another. To the uninitiated, (No, I guess I mean BASIC oriented persons) this is sometimes a difficult concept, but it is one of the things that makes Forth fast.

Forth does not have to find a variable and then fetch its value, the proper value is always there in the same place, on top of the stack ready to be used. The only problem is that you, the programmer, are the one responsible to see that the value needed is on top of the Parameter Stack or Stack as it is usually called. Forth does have another stack called the Return Stack, where it stores, among other things, the address it will return to after executing a word. When this is the stack in question, it will always be referred to as the Return Stack or RStk if no confusion could result.

This really is not too difficult a concept to master as we will shortly learn. For those who have not had the opportunity to read STARTING FORTH, the Stack is operated as a LIFO stack. If you can possibly beg borrow or steal that book, please read it, the cartoons really help a lot to understand stacks and Forth operations in general. LIFO means Last In First Out.

One way to envision a LIFO stack is the familiar spring supported plate dispensing stack found in most fast food dinners and restaurants. The last plate put on must be the first one taken off. There is another similarity which may seem strange to those familiar with the way memory usually works. The value on top of the

stack is not merely Copied, but is Removed. Forth provides a way to compensate for this as well as to access other than the top number on the stack.

Now does one put a number on the stack? Just enter one via the keyboard and press ENTER. For a while, I will replace ENTER with <E>, then I will gradually stop doing it altogether, but you know you have to do it. If you want to see the number on the stack, enter .S <E> (Print Stack) and Forth will print the stack to the CRT without disturbing the stack. The right-most value is the top of the stack. Try entering at least three numbers. The Forth word for print is . (dot). If you enter . . . <E> the top three numbers on the stack will be printed to the CRT. Now if you enter .S <E> the top three numbers on stack will be gone, because you used them.

To clear the garbage from the CRT, do CLS <E>. Isn't that easy?! Now, lets learn how to manipulate the stack. First do ABORT <E> then .S <E>, and verify the stack is empty. Enter the numbers 1 2 3 4 just like that, one <E> will do for all, or you may hit <E> after each one. Enter .S to verify that they are there and in that order. Now enter DUP .S and find the the stack has 1 2 3 4 4 on it. Enter DROP .S to find that the last 4 is gone. Now enter DROP again then .S to see 1 2 3. Now enter ROT (Rotate) .S to see 2 3 1. Try ROT ROT .S to see 1 2 3 again. Get the idea? ROT lets you access the third number down in the stack. Now enter OVER .S to see 1 2 3 2. OVER copies (does not remove) the second value down in the stack onto the top. See, that is not so tough. Later we will write some new words to further manipulate the numbers on the stack when the occasion demands or it seems convenient for us. That is what is nice about Forth, we can change it to be more convenient for ourselves.

Now that we can put numbers on the stack, we need to do something with them, why else put them there. Forth uses what is called Post Fix Notation. This is what most people call Reverse Polish Notation or RPN. The "Polish" refers to a very learned Polish mathematician, Jon Zukasiewicz, whose name no one could pronounce. He first used a particular notation to explain logical principles, the method had great merit and became known as Polish Notation, but some, the Italian Mathematician Giuseppe Peano in particular, found it awkward and reversed the order of the symbols, thus creating Reverse Polish Notation RPN which has proven to be a very durable expression. The method found favour among computer designers because it was easier for the computer to store two numbers, then to do the required operation on them. The only problem with it is that humans are not taught to think that way. Its not very hard to learn however.

## INTEGER ARITHMETIC

Let's try some arithmetic. First, we put two numbers on the stack, then say what we want done with them. Enter 2 2 \* . to see 4 on the CRT. 10 2 / . should put 5 on the CRT. 3 4 + . should produce 7, and 8 3 - . should give you 5. Now try 8 3 / . to get 2. Whoa, what happened here? Everybody knows that is not the right answer, or is it? It is, when one does integer arithmetic.

Forth does Integer Arithmetic, like any other computer language which does not incorporate a special subroutine to evaluate fractional remainders when two numbers are not exact multiples of one another. Computers do not inherently do decimal arithmetic, with decimal points and all. Many languages, TI Forth and BASIC among them, provide special programming to handle fractional results with great precision. The TI 99/4A is outstanding in that it has a built in routine which is accurate to about 10 decimal places. Some of the more popular home computers are lucky to produce results accurate to 6 places. Forth does have access to the 99/4As Floating Point Arithmetic (FPA) capability, but it must be separately loaded as we discussed earlier. It does arithmetic very slowly compared to Integer.

continued on page 18

# Multiplan Exercises #1

by Herbert Schlesinger, USA

The purpose of this paper is to help the infrequent MULTIPLAN user become familiar with the possibilities of MULTIPLAN.

At all times if the "?" question mark key or FCTN 7 (HELP) is pressed, HELP screens will appear applicable to the screen or command you are in at that time. More later on this.

You will need the following files on a disk in order to be able to follow through the tutorials in this series.

	PAGE
-PRINTME	Title and disclaimer
GRADES	Entering Alpha, values & Averaging 5
GRADES1	Intermediate screen . . . . . 8
GRADES2	Resulting screen
TAXES	Formatting . . . . . 10
TAXES1	Formulas . . . . . 12
TAXES2	Ending Screen
SALES	Functions . . . . . 11
SALES1	Ending Screen
PRINT_SS	Printout illustration . . . . . 14&31
COMMISSION	Ranges and Copying . . . . . 15
COMMISSION1	Resulting screen
TENYEAR	Drawing Ranges and Copying . . . . . 17
TENYEAR1	Windows & Borders . . . . . 24&27
TENYEAR2	Ending Screen
NAMES	Some FUNCTIONS are illustrated . . 20
NAMES1	End function screen ; using NAMES . 29
INSERT	Using INSERT & DELETE . . . . . 22
INSERT1	The insert sheet at the end
MOVECOL	Columns and Ranges . . . . . 23&24
MOVECOL1	What we end up with
SORTING	Sorting by Alpha or numbers . . . . 28
CHECKING	Balance for Deposits and checks . . 30
CHILDREN'S--MEN'S--WOMEN'S	Supporting sheets . 33
HERB'S STORE	Dependant sheet . . . . . 33

As a general rule as you go through these exercise you will start with a named file and will end with the screen looking like the screen brought up by the named file plus an additional number: such as CELLS and CELLS1 or something like that.

It is necessary to have the following configuration for your system:

- Console
- PE Box
- 32K expansion card
- The Multiplan cartridge and disk

## MULTIPLAN EXERCISES

### ENTERING MULTIPLAN:

Put the MULTIPLAN screen on your monitor as follows:

1) Place the MULTIPLAN Cartridge in the console; the MP disk in drive 1 and select "2 Multiplan" from the options on the title screen. When the disk stops turning the MULTIPLAN screen will instruct you to press <ENTER>. A screen will form which has certain features:

There is a series of numbers down the left margin; also numbers across the top. These are the ROW and COLUMN numbers. Actually there are more columns and rows than you see on your screen. Using the ARROW keys (FCTN S D E X) you can scroll to 63 columns and to 255 rows.

In the upper left corner is a highlighted #1 about which more later. (Page 25 & 26)

Just below Row number 21 is a COMMAND LINE. This says: Alpha (highlighted) Blank Copy Delete etc. As the space bar (also the TAB key on the Geneve) is used the

highlight (EDIT CURSOR) moves from one of these commands to the next; or, if the first letter (the capital) is pressed, that command is entered. Beneath this command line are the instructions:

"Select option or type command letter.  
The bottom line looks like this:

R1C1 (see note below) 100% Free Multiplan: TEMP  
(also page 7 & 22)

"CONTROL =" brings back the command line and cancels a command if you made an error or changed your mind.

The cell at R1C1 (ROW1 COLUMN1) is highlighted with the CELL POINTER and this lower left corner will always tell you which cell is the active cell. As you use the arrow keys notice that this indicator changes. The next item tells how much space is left in the Multiplan memory. As the amount indicated gets closer to 0% the entire operation becomes slower since the program has more work to do. The last of the notations is the name of the file on which you are working. If the file is unnamed it is called TEMPORARY. The space between the cell notation and the % left will show the contents of the highlighted cell; figures, a formula or alpha if anything is in that cell.

2) The Cell Pointer is moved by using the "ARROW" keys, FCTN E S D or X, or by using the COMMAND "Goto". As the right arrow key (FCTN D) is pressed the cell pointer moves to the right; when it reaches the right border of the screen it moves to the next column and the leftmost column disappears from the screen. It is not lost, just hidden from view. As the left arrow key is pressed, the "pointer" moves back until it reaches "C1" (column 1). If the down arrow key is pressed we move from R1 (row 1) down the "page" and on reaching the bottom of the screen continue as far as row 255 if we wish. Of course the top rows scroll off the screen as we add more rows to the bottom.

When we wish to enter a command, there are two choices:

A) Each press of the spacebar (or TAB key on the Geneve) will cause the MENU POINTER to move to the next option. When the desired option is highlighted press <ENTER>. Or,

B) Type in the first letter of the option you want and it is there immediately.

Suppose we want to move the CELL pointer using the Goto command. Press the space bar six times to Goto and then press <ENTER>; or just press G. The bottom of the screen will then show the Goto options:

GOTO: Name Row-col Window  
Select option or type command letter 'o'.

The menu pointer highlights the Name option.

To get to any cell from anywhere in the sheet, move the pointer to Row-col; press <ENTER>, or type "R" and the screen will look like this:

GOTO row: 1 column: 1  
Enter a numberxx

Now if you want to go to, say, R15C12; merely enter 15 at highlight 1 and then PRESS CTRL A (TAB KEY) which will shift the pointer over to the column field, Thus:

GOTO row: 15 column 1  
Enter a number

Type 12 at the highlight and press the <ENTER> key, the cell pointer moves to that position and the indicator at the bottom left of the screen reads R15C12.

All the commands work on the same principle: whatever your highlight is on in the command line, upon pressing <ENTER> you are in that command situation. Or just press a letter.

#### DATA:

There are three types of data used in the spreadsheet:

- a) ALPHA - Letters or numbers NOT used in calculations: names, dates, addresses etc.
- b) NUMBERS - Numbers used in calculations such as quantities, scores, dollar amounts.
- c) FORMULA - Which perform the calculations as summing, multiplying, dividing, or other combinations.

#### LET'S TRY THIS OUT:

!!!! Put the "GRADES" file on the screen. To put any file on the screen we must use a certain procedure:

From the command line press TRANSFER. The options then are:

TRANSFER: Load Save Clear Delete Options Rename  
Select option or type command letter

Select LOAD and enter the Filename (do not enter the Dsk#, just the file name) in Capitals or press an arrow key to see the choices (Directory of files on the disk). Move the highlight with the ARROW keys and then Press <ENTER>. When the file "GRADES" has been loaded there is a list of names and grades; add a name or two to those already there:

#### ENTERING DATA:

Move the cell pointer to the cell you want data entered in.

Select "ALPHA" from the command line by pressing "A" or by just pressing <ENTER> since that option is already highlighted. A prompt at the bottom of the screen invites you to enter the desired information - in this case a name, say James.

After typing the name, using Upper or Lower case or a mixture, you can enter it into the cell area (the highlighted cell) by either of two methods:

- 1) Press <ENTER> which enters your data AND brings back the entire command line. OR,
- 2) press the right arrow (FCTN D) key or any other arrow key you wish. In this case the data is entered into the highlighted cell and the highlight moves in accordance with the key pressed. The Command line then looks like this:

#### ALPHA/VALUE:

#### ENTERING FORMULAS:

If the next thing you type starts with a number, this command line will shift to "VALUE", but if a letter is entered, it shifts to ALPHA.

Getting back to our example: enter a few names and give each some scores in columns 2 and 3.

Here is how we do it: Using the arrow keys place the cell pointer in the CELL you want to enter something in; Press <ENTER> (or "A") to get Alpha and enter a name (James); Press the down arrow and you are in position to enter the next name. The first is in the cell you wanted it in. To enter a score place the Cell Pointer where you want it and, after entering an amount, press an arrow to go in the direction desired.

Next we would like to know what the average grade is for each student so we will use a formula: AVERAGE=Sum of grades/number of grades (in this case 2). Using the arrow keys place the cell pointer at RIC4 and enter = (the equal sign). The bottom of the screen will show:

#### VALUE: enter a formula

Enter the formula: (RIC2+RIC3)/2 and press <ENTER>. The result immediately appears in the highlighted cell (RIC4). Do this for each student, but note that the formula must state the correct cells - R2 for the second row, R3 for the third, etc.

Notice that the contents of each cell is shown at the left lower portion between the cell name and the % of space left.

TO END A SESSION; and save what you have been doing, if you wish to save it, follow this procedure: Press <ENTER> to bring up the command line if it is not already there, OR press "CTRL =" to accomplish this whenever you want the command line; Press Transfer and at the option offered press "S" or move the highlight to Save and Press <ENTER>. Now you are offered:

TRANSFER SAVE filename: GRADES or whatever file name is present, if any.

#### Enter a filename

Type in a name or use the default you find there and then press <ENTER>. If you choose a filename already on the disk you will be asked: "Overwrite existing file?". If you wish to overwrite press "Y", otherwise press any other key and you are back on the command line. If an unused filename is entered the warning is not given.

#### TO LEAVE MULTIPLAN:

Highlight Quit or press "Q" from the command line which will bring up - Enter Y to confirm. So, if you wish to leave press "Y". Pressing any other key will return you to the command line.

-----0-----

continued from page 16

If you want a more accurate answer to the problem of 8/3, then write 8 3 /MOD, and Forth will leave both the Quotient and the Remainder (ie the MODulo) on the stack. Try 10 7 /MOD . . to get 1 3 on the CRT. Integer arithmetic does not usually cause problems when simple arithmetic is all that needs to be done; by techniques to be discussed later, one can get answers as accurate as one needs. However, when one needs transcendental functions like logarithms, or trigonometric functions, etc is usually best to use FPA. Later, I will show you how to get around this if you need speed and will settle for less accuracy.

#### LOGIC OPERATIONS

Screen #60 shows some logic demonstration words. Rather than print some examples here, (there are too many combinations) I have written Screen #60 to allow you to test each of the logical operators. Please enter and LOAD this screen, the screen number you use is not critical, since it is not referred to by any other word.

To test the words, put combinations of 1's and 0's (True and False) on the stack (actually any none zero value is the same as a 1) and test each word out after rereading the TIFM Chapter 2 page 3. The thing to learn here is the order in which the numbers are compared. When you can predict the answer for any logic operation on any two numbers on the stack, you have got it made.

The operation NAND (Not AND) is not a resident Forth word so I created one. Actually this is called "defining", and next time I will cover how to define new words in Forth. In fact, we will be rewriting parts of Forth to suit our selves.

That is about all we can cover this time, C U next time, may the FORTH be with U.

# TI-Base Tutorial #8

by Martin Smoley, North Coast 99ers USA

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

\*\*\* More on Printer Controls \*\*\*

Last month I thought I covered printer controls, but then I tried to use them. I made a real mess of things. At that point I decided that instead of trying to remember another set of printer commands, I would change the TIB commands to the type I use the rest of the time. I normally use directly inserted codes which can be found on page 146 of the original TI-Writer manual. I placed these characters in the right hand column of last months FunnelWeb TI-Base Control Code sheet. After firing up TI-Base I typed USE DSK5.PRINTER, which is where my PRINTER DB is located, and then MODIFY STRUCTURE. This brought up the PRINTER STRUCTURE SCREEN, which I then modified to look like the one below.

CREATED	CHANGED	FIELD	DESCRIPTOR	TYPE	WIDTH	DEC
		1	NAME	C	010	
		2	FF	X	002	
		3	LF	X	002	
		4	CR	X	002	
		5	G	X	004	
		6	UL	X	006	
		7	e	X	002	
		8	f	X	004	
		9	4	X	004	
		10	E	X	004	
		11	SPS	X	006	
		12	SBS	X	006	
		13	HT	X	002	
		14	ST	X	020	
		15	Drft	X	026	
		16	BLANK	X	030	

000 1 PRINTER 00006/00007

You can type over the name and press FCTN 8 to save your changes. Do not change anything in the width column or you will lose the data in the file. Next I type EDIT <E>, to get into the file and make one change. When the EDIT Screen came up I pressed FCTN 5 to get to the EPSON record and I pressed <E> until the cursor was in the ST field. I then entered 1B4428 followed by all zeros, and FCTN 8 to save my changes. This changed my ST (Set Tab) field to set one tab at 40 columns. 1B44 means Set Tab, and 28 is Hex for 40. The zeros are null bits and do not do anything. I then typed CLOSE ALL <E>, to get out of the PRINTER DB. I had previously removed the asterisk (\*) from the beginning of the PRINTER EPSON line in my SETUP CF, so I typed DO DSK5.SETUP. The SETUP file ran and I saw the PRINTER flash on the screen so I new that my new commands were in place. NOTE: I have placed a copy of the PRINTER database at the bottom of this page with my modification to the ST field. I chopped off the BLANK field to make it all fit on one line. The BLANK field contains 30 zeros. You cannot print out your PRINTER DB because of the special X-type field designation. To print this out I place a copy of the two PRINTER files on a disk in drive #1 and entered these commands. COPY DSK1.PRINTER/D DSK2.C\_PRNTR/D GO <E>, then COPY DSK1.PRINTER/S DSK2.C\_PRNTR/S GO <E>. If you have a two drive system, you must also have a copy of the TIB system in Drive #1, or you can make the copy directly from your TIB system disk. These commands will cause TIB to COPY the printer files to two new files. You can now enter USE DSK2.C\_PRNTR <E>, and the MODIFY STRUCTURE <E>. At this point change all of the X-type

designations to C, for Character, and FCTN 8 to save the changes, and FCTN 9 to get out of that mode. At that point I entered TOP <E>, then PRINT (f) and PRINT ALL. My printout was roughly what you see at the bottom of this page except for added line because of field lengths. With this printout you can visualize the complete PRINTER database and see where your own particular printer fits in. Remember to modify the original PRINTER database if needed. The C\_PRNTR database is completely useless except for a printout.

I really hope this helps some of you get into the swing of printer controls. I spent a lot of time and messed up a lot of data in order to produce this tutorial.

Hopefully you now have modified your printer control database and have put it in use by removing the asterisk from the SETUP file on your system disk. Now let's put the PRINTER COMMANDS to some use. I had a specific question at one time that I will answer in the same Command File. The question was, "How can I print more than one label across? I got some labels cheap that are two across."

## PRINTER DATABASE

```

REC NAME      FF LF CR G   UL   e f 4   E
                SPS  SBS  HT ST          Drft  BLANK
0006 DIABO    OC OA OD 0000 000000 00 0000 0000 0000
000000 000000 09 00000000 000000000000
00000000000000000000000000 00000000000000
0000 EPSON    OC OA OD 1B47 1B2D01 0E 000F 1B34 1B45
1B5300 1B5301 09 1B442800 000000000000
1B481B2D3014121B351B461B54 00000000000000
0003 MX-80    OC OA OD 1B47 000000 0E 000F 0000 1B45
000000 000000 09 1B440A0A 0A0A0A0A0A00
1B4814121B460000000000000000 000000000000
0002 NEC      OC OA OD 0000 000000 0E 1B51 0000 1B21
000000 000000 09 1B283031 302C30320000
1B220F1B4E000000000000000000 000000000000
0004 OKIDATA  OC OA OD 1B48 1B43 1F 001D 0000 1B54
001B4A 001B4C 09 00000000 000000000000
00000000000000000000000000 000000000000
0005 PROPRINTER OC OA OD 0000 000000 00 0000 0000 0000
000000 000000 09 00000000 000000000000
00000000000000000000000000 000000000000
0001 TI-850   OC OA OD 1B47 000000 0E 000F 0000 1B45
000000 000000 09 1B440A0A 0A0A0A0A0A00
1B4814121B4600000000000000 000000000000
    
```

```

* Command file 2WLBS/C
* Copyright Martin Smoley 1989
*
* SET TALK OFF
* TRACE ON
SET RECNUM OFF
SET HEADING OFF
SET PAGE=000
CLEAR
COLOR WHITE,DARK-BLUE
LOCAL XPDT C 38
LOCAL NAME C 38
LOCAL ADDR C 38
LOCAL CTSTZ C 38
LOCAL TEMP C 38
USE TNames
TOP
PRINT (ST),(G)
WHILE .NOT. (EOF)
  REPLACE XPDT WITH "          " ;
  | " Exp. Date " | XP ;
  REPLACE NAME WITH TRIM(FN) | " " ;
  | MI | " " | LN
  REPLACE ADDR WITH SA
  REPLACE CTSTZ WITH TRIM(CT) | " " ;
  | ST | " " | ZP
MOVE
IF (EOF)
  PRINT (G),XPDT,(Drft)
  PRINT (CR),(LF)
  PRINT (E),NAME,(Drft),(G)
  PRINT (4),ADDRS
    
```

```

PRINT CTSTZ,(Drft)
PRINT (CR),(LF)
ELSE
REPLACE TEMP WITH "      ";
| "      Exp. Date " | XP
PRINT (G),XPDT,(HT),TEMP,(Drft)
PRINT (CR),(LF)
REPLACE TEMP WITH TRIM(FN) | " ";
| MI | " " | LN
PRINT (E),NAME,(HT),TEMP,(Drft),(G)
REPLACE TEMP WITH SA
PRINT (4),ADDRS,(HT),TEMP
REPLACE TEMP WITH TRIM(CT) | " ";
| ST | " " | ZP
PRINT CTSTZ,(HT),TEMP,(Drft)
PRINT (CR),(LF)
MOVE
ENDIF
ENDWHILE
CLOSE ALL
DO DSK5.SETUP
RETURN
*
* CF to print two across labels

```

The Command File on this page is the working result of my effort for this month. It is probably not the most efficient as far as programming is concerned, but it does print labels two across and it also demonstrates the use of the printer controls in the Command File mode. If you wish to type this CF in and use it, it is too large for the TIB Editor, so it must be entered with FunnelWeb in the non-wordwrap mode. Lines 4 and 5 do not execute because of the asterisk in the first column. If you run into a lot of bugs, you can find many of them by watching the program lines scroll up the screen. When the CF works well, remove the asterisk from that line to turn the screen junk off. If your problems are massive, as mine were, remove the asterisk from the TRACE ON line and all lines which are executed will also be sent to the printer, along with line numbers. The printout will help you find your problems. When the CF runs well remove the TRACE line completely. The line PRINT (ST,(G)) is the first line to issue printer controls. It sets the printer tab at 40 columns and turns double strike on. Note: This line can be typed at the Dot Prompt and executed provided a DB is in use at the time. I am using TNames which contains 5 records. "WHILE .NOT. (EOF)", is roughly the same as saying to TIB, if you have any data right, now load the respective fields into XPDT, NAME, ADDR and CTSTZ, then MOVE to the next record. The IF statement says, IF you have turned up the EOF marker with that MOVE, then print out the fields you have and jump to the end or ENDIF. This will print one label. If the MOVE has not brought us to the EOF marker, TIB will jump to the ELSE statement. The ELSE statement is the same as saying, we must have another record so proceed with the statements right after the ELSE, which will produce two across labels. TEMP is reused for each line so that portion works like this. REPLACE TEMP WITH " Exp. Date" XP, is filling TEMP with items from the second record because of the MOVE statement. Therefore, PRINT (G),XPDT,(HT),TEMP,(Drft) says this; set double strike on, print whatever is in XPDT (which we gathered from the previous data record), Horizontal Tab to column 40, print TEMP (which contains data from the second record, and last, change the printer back to Draft mode. This will print across the page side by side with a tab of 40. The next line prints a Carriage Return and a Line Feed. I like the PRINT (CR),(LF), it eliminates the need to initialize space for BLNK, which is what I previously used for a line feed. This process continues until the two labels are printed, another MOVE is executed and the whole process starts over. When the EOF is reached the WHILE statement no longer executes, and in my case the SETUP file is run. I have started using my SETUP CF at the end for other Command Files. I turn on the RECNUM etc. and it DISPLAYs the STATUS so I can see exactly what is happening. This way I always return to a system I am familiar with, and it is easy to do. You may not want to use my symbols for control codes, (G) double strike, (f) condensed etc., but you should have enough information at this time to set up

whatever you want and be able to use whatever you do set up. I am beginning to like this type of printer control, because it is available at all times. I do not need to jump to FunnelWeb to insert special characters. I have not tried this yet, but I believe that the control codes for your personal system can be greatly expanded by dedicating the whole PRINTER database to one printer. For example, you would not have DIABLO, EPSON, MX-80, etc. You would have EPSON1, EPSON2, EPSON3 etc. and fill the whole database with EPSON commands. With the PRINTER database on the PRGDISK you could then SELECT an unused slot within a running CF and execute PRINTER EPSONn (n=1-0). At that point TIB would load another 15 printer commands which you could use. Interesting thought isn't it. See you next month. o

## Root Finder

by Lucie Dorais, Ottawa Users Group

Lucie Dorais published this very neatly programmed routine in the news letter of the Ottawa User Group. It will find any root, from square root to 9th root, of any number, by a brute force method, and show you its calculations on the screen.

(After typing in "run" the usual introduction appears at the top of the screen and then after a brief delay a line, a number 3 and a couple of letters appear underneath with the cursor on the 3. I eventually identified this as a cube root symbol as you do not need the number 2 if it is a square root. Anyway you type on top of the 3 the root that you require and press 'enter'. The cursor then jumps to another place and you are then to type in the number that you want a root taken of. The computer immediately jumps into action and calculates the desired root and shows you what it is doing on the screen. Neat! After performing the calculation a menu appears at the bottom of the screen which gives you 4 choices - [A]nother, [C]hange, [P]rint, and [Q]uit. If you choose 'A' then it will calculate the same root of Another number. If you choose 'C' then it will give you the opportunity of Changing the root as well. 'P' will give you a print-out which is interesting as the printer forms a root symbol with the number that you are taking the root under the symbol and then it prints the "=" with the answer after it. For those of you with an RS232 you will have to change line 120 to read PR\$="RS232.BA=4800(or whatever)". I cannot say that it is faster than the pocket calculator that I have but it is certainly more interesting to watch and you can have a print-out as well. ED)

```

90 CALL CHARPAT(121,CH$):: CALL CHAR(33,CH$)
100 DISPLAY AT(1,1)ERASE ALL:" ROOTS B! Lucie Dorais":
    To Find any root from cube root to 9th root"
110 !
120 ON WARNING NEXT :: PR$="PIO"
130 L$=RPT$(" ",28):: E$=RPT$(" ",168):: S$=RPT$(" ",8)
140 CALL CHAR(120,"00000000002050F",122,"018182C2C4646
    830",123,"0B0101",125,"FF")
150 DISPLAY AT(5,9):"x3y"&RPT$(" ",10):S&"{Z" :: GOSUB
    280
160 ACCEPT AT(6,2)VALIDATE(NUMERIC)BEEP:N :: IF R>2 THEN
    180
170 IF R=1 THEN AV=N :: GOTO 220 ELSE AV=SQR(N):: GOTO
    220
180 LO=0 :: HI=SQR(N)
190 AV=(LO+HI)/2 :: T=AV^R
200 IF OAV=AV THEN 220 ELSE OAV=AV :: DISPLAY AT(12,8):
    AV
210 IF N<T THEN HI=AV :: GOTO 190 ELSE IF N>T THEN LO=AV
    :: GOTO 190
220 AN=AV :: DISPLAY AT(12,8)BEEP:"=";AN
230 DISPLAY AT(22,1):L$:" [A]nother [C]hange [P]rint
    [Q]uit"
240 CALL KEY(0,K,S):: IF S=0 THEN 240 ELSE K=POS("ACPQ",
    CHR$(K),1)
250 IF K=0 THEN 240 ELSE ON K GOTO 260,260,270,290
260 DISPLAY AT(7,12):E$:E$:E$ :: IF K=2 THEN GOSUB 280
    :: GOTO 160 ELSE 160

```

continued on page 21

# Archiving

by Tom Arnold, Hamilton, Ontario Canada

Archiving - what in the world is that? A little imitating maybe? Well you are going to have to live with it whether you like it or not. However, it really is not all that difficult. To begin with, archiving is the squeezing together of files so that they can be transmitted faster or fit on a disk more compactly. There are many good reasons for doing this. The most important is when using bulletin board services. Time costs money, especially if you are hooked up to The Source or CompuServe. By compacting files we can transfer more data in less time and so save money. (Not applicable to TEXPAC which supports DIS/VAR 80 files only).

Another use is for storage on disks. If you send disks to friends by mail you will be able to get more on your disk, thus saving postage. As I am sure you will come across archived files I thought you would like an explanation of how they work.

The first Archiving program that I saw was Barry Traver's Archiver which was written in Extended Basic, slow but effective. He soon came up with an assembly language version which was much faster. Several other versions are around but the best and most recent is Barry Boone's Archiver II V2.3. If you do not have this one get it from our library. It has several features the others do not have, including a Cataloguer option, Archived File Catalogue and Return to Funnelweb. Most importantly it archives in two stages so that you can make your files smaller still.

The best way to find out how this program works is to take a disk with half a dozen files and programs on it and archive it. Follow these steps:

1) Load up the Archiver II V2.2 with the E/A module (or FunnelWeb) and select #5 - DSK1.ARC1.

2) Select the option PACK FILES and then name the Disk Drive you wish to pack. Answer "yes" to "packing all files". Send the output to another fresh disk in drive #2. Make sure the disk is blank so that you have room for all the files. Give the packed file a logical name. Usually at this stage the files are given the name as such: NAME\_ARC the \_ARC indicating that the files have been packed. The packed files will be in the form DISPLAY FIXED 128.

3) Now select the COMPRESS option. Name the output file as such: NAME\_ARC the \_ARC indicating compressed files. When you come across these file name you will know you will have to do a two step procedure to unpack your files. Compressed files have the structure INTERNAL FIXED 128. Answer "All: to pack all the files. The file names will automatically be written to your disk. The compress option only asks for the respective disk drives, make sure you have enough space on your disks.

4) Now that you have packed the files, try unpacking them. you should land up with the same files you started with.

Handy hint: The intermediate files (the packed files) are not really needed and are used only temporarily. I used the RAM Disk to write to so that I saved time and do not have to worry about changing disks. For example, if unpacking I decompress files to the Ram Disk from disk drive #1. Then I unpack from the Ram Disk to disk drive #2.

The space saving using this double packing is quite significant but varies with the types of files you are unpacking. Program files do not shrink as well as text files. One example I tried went from 352 sectors to 233 where another of 66 only went to 54 sectors (it was two program files).

So next time you come across a file name with the suffix \_ARC or \_ARK then get out that Archiver and do not forget to send Barry Boone a donation. His address is:

Barry Boone  
Box 1233  
Sands Springs,OK  
U.S.A. 74063

continued from page 20

```
270 OPEN #1:PR$ :: PRINT #1:S$&" "S$&" "&
STR$(R)&"/" :: PRINT #1:S$&"/";N;TAB(26);"=";AN;"
:: CLOSE #1 :: GOTO 240
280 ACCEPT AT(5,10)VALIDATE("123456789")SIZE(-1)BEEP:R
:: RETURN
290 END
```

Later on Lucie found and published this mathematical method which is faster (instant answers! ED) but not nearly as much fun to watch.

```
90 CALL CHARPAT(121,CH$):: CALL CHAR(33,CH$)
100 DISPLAY AT(1,1)ERASE ALL:" ROOTS B! Lucie Dorais:"
To Find any rootfrom cube root to 9th root"
110 !REVISED VERSION USING THE FORMULA AN=NA(1/R)
120 ON WARNING NEXT :: PR$="PTO"
130 L$=RPT$(" ",28):: E$=RPT$(" ",168):: S$=RPT$(" ",8)
140 CALL CHAR(120,"00000000002050F",122,"018182C2C4646
830",123,"0B0101",125,"FF")
150 DISPLAY AT(5,9):"x3y"&RPT$(" ",10):S$&"Z" :: GOSUB
280
160 ACCEPT AT(6,2)VALIDATE(NUMERIC)BEEP:N :: IF R>2 THEN
180
170 IF R=1 THEN AN=N :: GOTO 220 ELSE AN=SQR(N):: GOTO
220
180 AN=NA(1/R)
220 DISPLAY AT(12,8)BEEP:"=";AN
230 DISPLAY AT(22,1):L$:" [A]nother [C]hange [P]rint
[Q]uit"
240 CALL KEY(0,K,S):: IF S=0 THEN 240 ELSE K=POS("ACPQ",
CHR$(K),1)
250 IF K=0 THEN 240 ELSE ON K GOTO 260,260,270,290
260 DISPLAY AT(7,12):E$:E$:E$ :: IF K=2 THEN GOSUB 280
:: GOTO 160 ELSE 160
270 OPEN #1:PR$ :: PRINT #1:S$&" "S$&" "&
STR$(R)&"/" :: PRINT #1:S$&"/";N;TAB(26);"=";AN;"
:: CLOSE #1 :: GOTO 240
280 ACCEPT AT(5,10)VALIDATE("123456789")SIZE(-1)BEEP:R
:: RETURN
290 END
```

continued from page 15

Myarc XB screen to printer, and copy files. Now you can dump your Myarc XB bit map graphics!

>JOE NOLLAN: An interesting disk of LOADers and MENU makers of one sort and another, and the source code to make your own "title screen" come up too! Joe has tackled a major problem- when switching from one XB program to another, resetting ALL the characters!!! Best menu/load suite I have seen.

>NUTS AND BOLTS DEMO DISK. An auto-loading auto-repeat demo disk of some of the utilities on Jim Petersons Nuts and Bolts disks.

>PAUL-1 by Paul Scheidemantle, primarily a disk labelling program (226 sectors) for 1 and 1.5 inch labels (shops sell 1.5 inch labels here as 1 7/16, paul is including the gap between labels). Many formats and styles including one two or three columns of file names - with one column you can add comments. Also added to this disk is a 17 sector utility to chart the usable characters in all your TI Artist fonts.

>PAUL-2 by the same author, two GRAPHX posters- GRAPHX needed to produce the required result, two A4 posters! Jokey text which illustrates how GRAPHX can indeed be used for an A4 output.

# A Look at Assembler

by Art Green, Ottawa Users Group

As I see it, programming is the ultimate computer game. It requires the intellectual activity of the toughest "adventure" games and the skill of the action games. It also requires practice, as does any good game. There are several programming games that you can play. If BASIC programming could be equated to checkers, Assembler Language programming would be equivalent to chess. Although Assembler Language is more difficult than BASIC, it is also more fun.

Like any game, there are certain skills and tricks that help you win. My intention in this series of articles is to show you some of these tricks and skills. I hope the articles will be of benefit to both the novice and the experienced programmer. So, dig out the rule book (the Editor Assembler Manual), as we will be reading the rules very carefully to try to make them work for us, not against us.

The following articles will have coding examples written for use with a good macro-assembler (if there was not one available for the TI99/4A I would be tempted to write one). Also, in many of the examples we will evaluate the "cost" of coding the problem. One of the goals of the Assembler Language game is to minimize the cost.

There are three costs to be considered for each piece of code. The first two are related to the machine. The first is the "storage cost" which is the number of words of memory the section of code requires. The second is the "time cost" of running the code.

Calculating the storage cost is easy, you just count the number of words the instructions and data use. The time cost takes a bit more thought. We would like to be able to actually count the number of machine cycles needed to execute the code, but that is very tricky to do. However, on most microprocessors, you can estimate the time cost by counting the number of words of storage that must be fetched or stored during execution of the code. Multiply and divide instructions are the exception to this rule.

Let us try an example. Consider the instructions:

```
MOV @A,@B      Move word from A to B
MOVB @A,@B     Move byte from A to B
```

Both of these instructions have a "storage cost" of 3 words (each instruction occupies 3 words of memory) and a time cost of 5 words. Even though the second instruction moves only one byte, we all know the TI99/4A is a 16 bit machine and storage is always fetched or stored a word at a time even though only one of the bytes is needed. The time cost of the instructions is broken down as 3 words to fetch the instruction itself, one word to fetch A and one word to store B.

Now, a little more difficult example:

```
MOV *R1,*R2
```

This instruction has a storage cost of 1 word and a time cost of (quick now) yes, you guessed it, 5 words. The time cost is broken down as:

```
1 word to fetch the instruction
1 word to fetch the address in R1
1 word to fetch the address in R2
1 word to fetch the value pointed to by R1
1 word to store the value where R2 points.
```

You have to remember, the "registers" on the 9900 microprocessor are not special (like a lot of microprocessors) they are simply a method of addressing memory.

The third cost of code is more subjective than the first two. This third cost is the effort you have to

exert to write the code and to type it into the computer. This cost can be minimized by practice and by judicious use of the tools available. One of the best tools you can use is a macro. Most of the examples in upcoming articles will use and show the use of macros.

We have established the ground rules now for future articles. Next time we will look at "switches".

"It is a mistake to allow your computer to realize that you are in a hurry, or have a deadline to meet."

## Newsletter update

by Bob Relyca

HUNTER VALLEY 99ers, Christmas, 1990: President's Pen; Secretary's Report; Computing - The Future? by Garry Christensen; Random Bytes (#1,2,&3) by Bob Carmany; Far Out by Dick Schaydel; Nibblebytes by Al Lawrence; An Enhanced Kscan Routine by Tony McGovern; Beating Around The Bush by Ron Kleinschafer; Little Tex by Tony McGovern; Diskreview out of Funnelweb 4.31 by Al Lawrence; New-Age/99 by Jack Sughrue #9; Be Of Good Cheer #10, Comprodine, Part Two; Cross-Word Maker; Handy Tips for the TI99/4A; The Texas Instruments TI99/4A Home Computer, A Short History; A TI99/4A Refresher Chart.

MICROPENDIUM, November, 1990: Chicago Faire; Regena On Basic; Extended Basic, Computers, the Pope and Caesar; Characters and Colours in Assembly; My-Basic, Tips On Using Subdirectories; TI-Base, Command File Editors; More Bulletin Boards, Places To Phone With Your Computer; Reviews - Asgard Mouse, Artist Printshop & PagePro Headline Maker; Rave Expansion Box; Products From Asgard; a new source for a BBS, and a new software company debuts; User Notes.

SPIRIT of 99, December, 1990: Newsletter Rates; TI World News, November 1990 by Jim Peterson; Hardware; I Like Brain Games by Jim Peterson; TechTalk by Mike Maksimik; Music-Pro Version 1.4 by Jim Peterson, a review; Style a Line, A Tinygram by Ed Machonis; Print-A-Doc, a print utility by Ed Machonis; Heavy Duty Disk Sleeve by Allan Cox; TI-Base Tutorial 18.1.1,2,&3 by Martin Smoley;

LA99ers TOPICS, December, 1990: Ramblin' Thoughts From The President; My Subprograms Listed by Earl Raguse; Music Subprogram Listings by Earl Raguse; Speak To Me (using the TEII) by Chick De Marti; The Cracker Barrel by Chick De Marti; Hark The Herald Angels Sing, A Christmas Music Program by Vincent Maker; Personal Auditor (Home Accounting System) by Bill Gaskill; New/Age Special by Jack Sughrue; TI-Base Version 3 Index by Jerry Keisler; Kids Korner; Software Price List; Tips Version 1.6/ER, a review by Fred Moore; TI Print Shop, a routine with Tips V.1.6.

TI FOCUS, December, 1990: News and Views by Tom Arnold; Modem Prices Dropping and Attention Corcomp Owners by John Van Weelie; Air Taxi by Don Shorock, reviewed by Jim Peterson; The IBM/TI Connection (another TI super goodie) by Randy Packham; Jim Peterson TIPD Disks by Tom Arnold; Super Extended Basic, a module review by Tom Arnold; More TI-Base by Rick Lilley; Club Page by Tor Hansen; Basic Module Maintenance by Dave Ratcliffe TICO Topics.

THE PUG PERIPHERAL, December, 1990: TI Express by Patrick F. Powell, a crossword puzzle related to the TI; TI Artist Plus, Using Copy and Move by Jerry Keisler; New/Age99 #10 by Jack Sughrue; All About Character Sets by Andy Frueh; The Kiddie Corner by Sue Harper; From The Librarian.

UGOC ROM, December, 1990: From the President by Ben Hatheway; Graphic Sig by Bill Nelson; TI Bits #33 by Jim Swedlow; UGOC Hall Of Fame by Bill Nelson; The Doomsayer by Earl Raguse; Membership Corner by Bill Nelson; Hello There, Again by N. Armstrong, editor; In My Humble Opinion by Bill Nelson; Newsletter Tidbits #13 by Earl Raguse.

continued on page 23

# Regional Group Reports

## Meeting Summary For MARCH

Banana Coast	10/03/91	Sawtell
Carlingford	20/03/91	Carlingford
Central Coast	09/03/91	Saratoga
Glebe	07/03/91	Glebe
Illawarra	11/03/91	Keiraville
Liverpool	08/03/91	
Northern Suburbs	28/03/91	
Sutherland	15/03/91	Jannali

### BANANA COAST Regional Group (Coffs Harbour area)

Regular meetings are held in the Sawtell Tennis Club on the second Sunday of the month at 2 pm sharp. For information on meetings of the Banana Coast group, contact Kevin Cox at 7 Dewing Close, Bayldon, telephone (066)53 2649, or John Ryan of Mullaway via the BBS, user name SARA, or telephone (066)54 1451.

### CENTRAL COAST Regional Group

Regular meetings are normally held on the second Saturday of each month, 6.30pm at the home of John Goulton, 34 Mimosa Ave., Saratoga, (043)69 3990. Contact Russell Welham (043)92 4000.

### GLEBE Regional Group

Regular meetings are normally on the Thursday evening following the first Saturday of the month, at 8pm at 43 Boyce St, Glebe. Contact Mike Slattery, (02)692 0559.

### ILLAWARRA Regional Group

Regular meetings are normally on the second Monday of each month, except January, at 7.30pm, Keiraville Public School, Gipps Rd, Keiraville, opposite the Keiraville shopping centre. A variety of activities accompany our meetings. Contact Lou Amadio on (042)28 4906 for more information.

### LIVERPOOL Regional Group

Regular meeting date is the Friday following the Tishug Sydney meeting at 7.30 pm. Contact Larry Saunders (02) 6447377 (home) or (02) 7598441 (work) for more information.

### NORTHERN SUBURBS Regional Group

Regular meetings are held on the fourth Thursday of the month. If you want any information please ring Dennis Norman on (02)452 3920, or Dick Warburton on (02)918 8132.

Come and join in our fun. Dick Warburton.

### SUTHERLAND REGIONAL GROUP

Regular monthly meetings are held at the home of Peter Young, 51 Jannali Ave., Jannali on the third Friday of each month at 7.30pm. unless otherwise advised. (02) 528 8775.

The format of each meeting is quite informal, with topics ranging from software reviews to hardware modifications with a fair sprinkling of purely social chatter in between.

Future meeting dates are as follows:

FEB 15/02/91

MAR 15/03/91

APR 19/04/91

Peter Young Regional Co-ordinator

### TISHUG in Sydney

Monthly meetings start promptly at 2pm (except for full day tutorials) on the first Saturday of the month that is not part of a long weekend. They are held at the RYDE INFANTS SCHOOL, Tucker Street (Post Office end), Ryde. Regular items include news from the directors, the publications library, the shop, and demonstrations of monthly software.

### MARCH MEETING - 2ND MARCH

The MARCH meeting will emphasise music. The club

has excellent music software from Harrison Music for sale and the demonstration this month will be the MIDI music interface. This is of industry standard, and should enhance the already excellent music capabilities of our computer.

\*\*\*\*\*

The cut-off dates for submitting articles to the Editor for the TND are:

April	10 March
May	7 April
June	12 May

These dates are all Sundays and there is no guarantee that they will make the magazine unless they are uploaded by 9:00pm, at the latest.

\*\*\*\*\*

## Treasurer's Report

by Geoff Trott

Income for January	\$239.37
Payments in January	\$1671.07
Excess of expenses over income for January	\$1431.70

continued from page 3

### COMMERCIAL SOFTWARE.

TI Sort SSSD .....	\$15.00
Display Master .....	\$15.00
TI Artist Plus .....	\$25.00
Picasso Publisher Version 2.0 .....	\$14.00
Picasso Publisher Support Disks .....	\$14.00
Picasso Applications Disk .....	\$2.00
Nuts and Bolts #1 (DSSD) .....	\$6.00
Nuts and Bolts #1 (SSSD) .....	\$7.00
Genial Traveler (SSSD) .....	\$6.00
The Missing Link .....	\$28.00
Missing Link Applications Disk .....	\$2.00
Page Pro Version 1.6 .....	\$28.00
Page Pro Utilities .....	\$17.00
Spell It! .....	\$22.00
CSGD Cataloguer - Introduction .....	\$8.00

Packaging and postage charges:

	Surface	Airmail
Up to 2 Disks -----	\$1.80	\$2.30
3 to 9 Disks -----	\$2.40	\$2.70
10 to 15 Disks -----	\$3.10	\$4.30
TI Artist Plus -----	\$2.40	\$2.70
Display Master -----	\$2.40	\$2.70
TI Base -----	\$2.40	\$2.70
TI Sort -----	\$2.40	\$2.70
5.25 inch half-height drive (1.25 Kg) -----		refer to your local post office

Bye for now.

continued from page 22

LEHIGH 99'ER COMPUTER GROUP, October, 1990  
President's Message; Allentown TIBBS by Brad Snyder; C99 V.4.0/ROS 8.14 Pix; a new HFDC Card by Chris Pratt; Asgard's YAPP; Hard Card Follies; Triton Closes Shop; New/Age99 # 9 by Jack Sughrue; Editorial; Why Should You Learn To Program? by Jim Peterson.

December, 1990: President's Message; Computer Desk by David Strachan; In Our Midst; Idea Wanted!; The Animator by Brad Snyder; NASA Frequencies by Henry Badon; Mass-Transfer, a Review by Ted A. Stringfellow; A Little About Fast Term by Ed Hall; Review Of Telco 2.3 by Frank Garvin.