



NEWS DIGEST

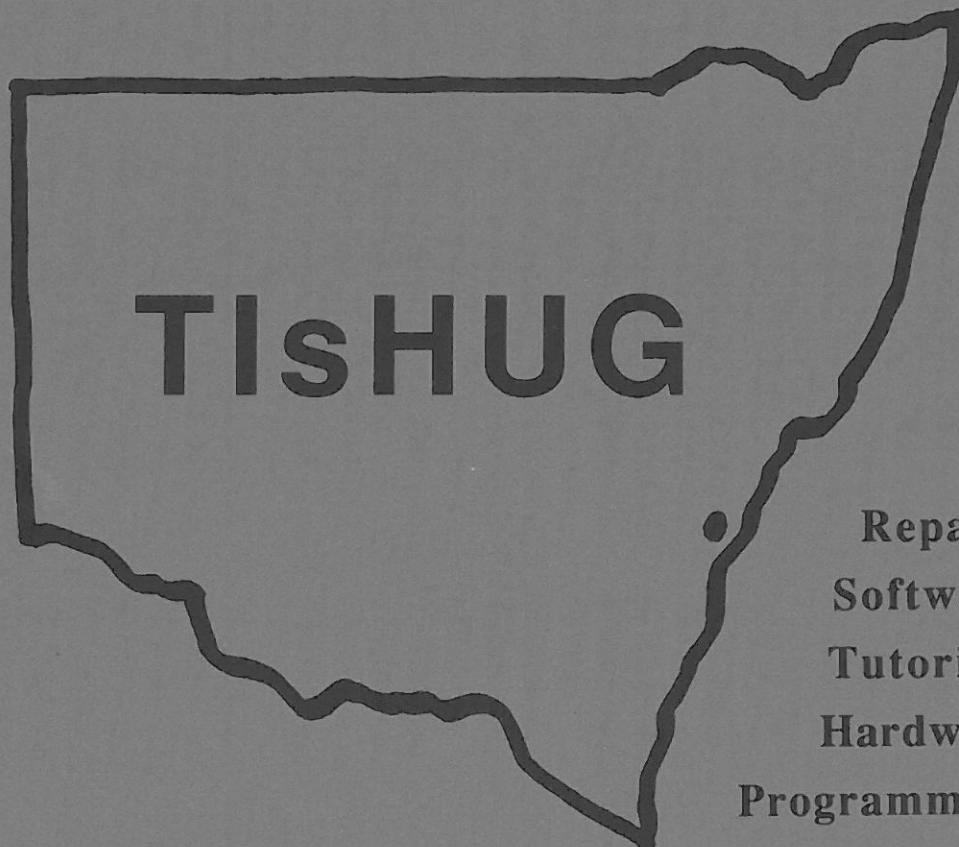
Focusing on the TI99/4A Home Computer

Volume 10, Number 8

September, 1991

Registered by Australia Post - Publication No. NBH5933

For all your TI99/4A



Repairs
Software
Tutorials
Hardware
Programming

Computing Needs

Sydney, New South Wales, Australia

\$3

September 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 7th of September at Ryde Infant School, Tucker Street, Ryde. At 9 am, before the main meeting, there will be a meeting of all those wishing to be involved in the train project followed at 12 pm by the beginners' Editor Assembler class for all those interested.

Printed by

The University of Wollongong
Printery Services.

Index

Title	Description	Author	Page No.
Attention: BBS members	Club news	Mudie, Ross	5
Beginning Forth #9	Software hints	Raguse, Earl	25
Cartoon documentation	Software review	O'Neill, Tim	12
Co-ordinators report	General news	Warburton, Dick	2
Comic show instructions	Software review	Kazmer, Ray	13
Communicators	BBS information	Mudie, Ross	14
Editor's comment	General interest	Relyea, Bob	1
Extended BASIC tips #10	Software hints	Swedlow, Jim	19
G corner	Software hints	Warren, Bob	9
G, an introduction	Software review	Warren, Bob	7
G, the HIRES language	Software hints	Krawczyk, Gene	7
Games information	General interest	Brown, Robert	10
Hyphenator version 2.0	Software review	Stith, Wayne	11
Multiplan exercises #7	Spreadsheet	Schlesinger, Herbert	20
Multiplan machinations	Spreadsheet	Harms, Bill	21
Multiple menus for BOOT	Software hints	Christensen, Col	6
Regional group reports	General interest		27
Rotating TIA instances	Software hints	Dorais, Lucie	15
Secretary's notebook	Club news	Phillips, Terry	3
TI-Base tutorial #13	Data base	Smoley, Martin	23
TI-Bits #9	Software hints	Swedlow, Jim	17
TIM installation	Hardware hints	Lawrence, Al	4
TIshUG shop report	Club news	Harrison, Percy	5
TIshUG software column	Club software	Schreiber, Rolf	5
Treasurer's report	Club news	Trott, Geoff	18
Younger set	Quiz	Maker, Vincent	26

Attention Newsletter Editors

All material appearing in this issue of the TND may be freely reproduced in other Newsletters provided that both the source and author are acknowledged. Articles are also available on disk as TI-Writer files for a nominal charge.

Editor's Comment

by Bob Relyea

The amount of activities at the club meeting seem to be more and more varied so that a user at any level can find something to catch his attention or to meet his need. I purchased some of the latest software for PAGE PRO and had fun loading it up and seeing what it could do. Being a Science teacher I was especially interested in the snippets of the human body because I can incorporate them into work at school for tests, worksheets and revision. They were really good! I would like to see us get in almost the full range of PAGE PRO gear. I hope you enjoy the articles in this month's magazine and find them to be interesting.

Co-ordinator's Report

by Dick Warburton

I have been giving the idea of a TI Faire in 1992 some thought and feel it really is time to give it a go. While we have been the largest TI99/4A User Group in Australia, and have achieved all manner of firsts for the TI99/4A, we have not yet hosted a TI Faire. Both the Brisbane and Melbourne TI99/4A groups have, but not us. Why should we go to all the trouble to run a faire for an outmoded computer like the TI99/4A? After all, anyone with a few spare dollars, and no experience, can go out and buy a super-duper, whizz-bang machine with superb graphics and digitized sound, games on CD ROM, and with massive capacity for memory and data storage. Why would anyone want to even look at a TI99/4A, when we compare it with what is on offer. How can it compete? Answer: it cannot, if we assess its value purely in terms of computing power. However, if we look at the TI99/4A from other angles, its real value becomes apparent. Let me explain what I mean.

First, like many other machines, the TI99/4A is a fun machine. It plays games well. Considering its memory limitations, its graphics are surprisingly good, even amazing when we consider a program like Rock Runner. While the graphics cannot compete with many later designs, they are adequate for the user to enjoy the game. I am getting fed up with games on other machines where there is so much attention paid to the visual display, that the games seem to become unnecessarily complicated by the actual graphics. I think that the point of the game is to play the game against self, the computer or a friend, not simply be dazzled by the complexity of the graphics. I still enjoy the simplicity of TI99/4A games like Munchman, or Rock Runner. They run so well, they are challenging, and they are cheap to buy. The challenge of a game is in its mastery, not in its graphics. There is a wealth of programs and games which I have not explored yet. There is still so much in games that I can learn to play on the TI99/4A. I was amused the other day, to see an advertisement on TV for a dedicated games machine, which proudly advised the customer that their machine now uses the power of genuine 16 bit chip technology; just 10 years after the humble TI99/4A. You can buy a dedicated games machine for about \$100, but then you buy each cartridge for about \$30. If you have a games machine you are stuck with the ongoing cost, the limited choice of games, and no flexibility to use the machine for other things. Similarly, educational type machines fall into the same category. The TI99/4A is cheaper and does the job better all round.

Some fond parents spend lots of money on their kids, perhaps sometimes to relieve their own guilt. One of the latest indulgences is the sound synthesiser. Every noisy kid should own one. I was fiddling the other night, and discovered a program which turns my TI99/4A into a synthesiser. The quality of the sounds surprised me. I then used a program which allowed my TI99/4A to actually sing to me, using the speech synthesiser. What a voice. You can have Pavarotti or Carreras, whichever I have my TI99/4A to sing to me. I suppose the point is that parents who invest in a cheap TI99/4A, which they encourage their children to master, could open a Pandora's box of discovery for the child. The beauty is that it is not restricted to one area, and it is not expensive. An intelligent child can master its secrets very quickly, and develop talents and abilities, which were unsuspected. I know one young Tier whose real intellect was not recognised until he was let loose on a TI99/4A. The TI99/4A developed his latent gifts.

What am I really saying? The TI99/4A is extremely versatile. It is cheap to buy. It actually requires some input from the operator. It needs an intelligent operator. It cannot compete with specialized computer equipment, but for the development of talent in children and providing families with endless challenge and fun at low cost, it is without peer. It can even challenge serious computer programmers. It will challenge hardware hackers to get more and more performance from this remarkable machine.

How many people outside TI99/4A community know about its virtues? When I recently visited Texas Instruments at Ryde, there was surprise at the range of things that have been developed for the TI99/4A since they abandoned the concept in 1983. Very few other computer users know of its existence. I find that other User Group members are really surprised that such a group as ours still exists. They are even more surprised at the range of things we do to help our members. Imagine if we were able to put the TI99/4A on show in a public place. Imagine that we have lots of members who contribute time and effort to demonstrate to outsiders and other TI99/4A users the range of things we can do with a TI99/4A. Think of the possibilities. A whole battery of TI99/4As set up by their proud owners, all doing different things. Our computer controlled train project, perhaps with an interface set up for household uses, alarms etc. Games machines, some simply running popular modules. Machines using RAMdisks and EPROMs with a variety of applications. Control of ham radio, using the BBS, simply being used as a word processor. We could invite people to sit down and try the simplicity of the TI99/4A. I have no doubt that many people would be surprised at the convenience of a TI99/4A with a RAMdisk, and more than surprised at its operating speed. People just do not know about the TI99/4A. Would you like to help to show off your equipment? Would you like to help people to understand more about computers? Would you like to surprise other computer users? You can, if we have a TI Faire!

There are many advantages for us going ahead with a TI Faire in 1991, eg

- 1) It will give us a sense of purpose for next year.
- 2) It will bring us together through the variety of tasks which will need to be accomplished to achieve the Faire.
- 3) It will highlight to all us again the extraordinary versatility of this computer.
- 4) It will attract other TI99/4A members from other states. We will get to know new people, and be exposed to an even wider range of ideas. We may be able to open our homes and billet TI99/4A users from other states.
- 5) Hopefully we will have the opportunity to meet a legendary TI99/4A user from the United States. Negotiations are proceeding for him to come to Australia.
- 6) Previous experience suggests that software developers like Asgard, will make their latest software products available to sell at the Faire. It is possible that new hardware will also be available. Certainly, we will do our best to make available the widest range of the latest TI99/4A gear.

The more I think about it, the more exciting and challenging it becomes. There are lots of problems if we are to succeed, but they can and will be solved. We need volunteers to help organize the details. We need a committee to take care of all the organization, e.g. we need to plan the venue, the date, the actual program. We need people to publicize our efforts, and we need to raise some extra money to ensure that we don't run short. I have already had some people approach me, offering to help. I cannot do it alone. I need your help. Please give it.

Members may not yet have realized that the Hunter Valley User Group has stopped operating as a group, although some of their members are still keen to continue using their TI99/4As. We would be privileged to welcome this bunch of talented people into TISHUG. I can see mutual advantages from such a merger. Hopefully we will see some of these members at our monthly meetings.

Do not forget the next meeting. See you there. ○

Secretary's Notebook

by Terry Phillips

Once again the weather turned out perfect and a large number of members turned out for the August meeting. Come to think of it, I cannot remember a meeting where the weather has been bad for quite some time. The more that attend meetings the more fun it is, so if you have not attended a meeting for a while why not come along and join in the host of activities that are available? Or why not come along and engage in idle chit-chat with the regulars present?

New members continue to be attracted to the group - a lot through referrals from TI - and it is a big welcome to the following:

Christoph Leitner - Rockhampton
Geoff Barnes - Blacktown
Mick Khan - Campsie
Kerry Bytheway - Liverpool

So far this year, 14 new members have signed up.

Things are not so rosey with the Hunter Valley Group. I received the following note from Brian Woods, secretary of that group.

"It was decided at the Annual General Meeting that due to a lack of input to the newsletter due in part to more and more of our members moving on to other computers, as well as a falling off of interest, that the Hunter Valley 99ers Users Group in its present form should be wound up.

"This letter is to inform you of that decision, and ask that you no longer include us on your exchange newsletters list. I take this opportunity to thank you for past exchanges. It has been of vital importance to our group, as it is with groups around the world that the TI community has always been prepared to share information concerning our orphan.

We at the Hunter Valley 99ers wish you well, and again, thanks for your newsletters."

It is always sad to see a group wind up - particularly one that was as active as the HV99ers. Any former members of that group are most welcome to make application to join us.

Jerry Keisler, a member from Paris Texas, forwarded the following comments on the July newsdigest:

1. Do you have the address of Thomas Opheys, Germany. I know some 386 owners that would like his TI99/4A emulator in lieu of buying a TI99/4A. (I personally do not have his address but I am sure Rolf will include it somewhere in this issue.)

2. TI-Bits No. 7 - The only way to fight lightning is total isolation. I too have surge protectors on the house, outlet feeding the computer and at the computer, but your best protection is unplugging everything, power, modem, outside controls etc. Lightning protection is like an insurance policy, however you can never protect 100% of lightning hits, 99% yes. Open switches, blown fuses etc mean nothing to lightning. After all, it travelled a mile or so of sky to reach ground, so what is a few inches? I use fast acting MOVs in the house and slow, brute force service entrance protectors on the service, the solid state kind, not air gap.

3. To Charge or Not to Charge - If you find batteries on sale, put them in the freezer until you use them. All batteries have a shelf life that the freezer will greatly extend. Yes batteries can be re-charged, however carbon-zinc and alkaline batteries deplete their cathode when operating. During use an oxide layer forms between the cathode and the electrolyte past increasing internal resistance. Charging or heating, in oven for short times, will remove the oxide. You can repeat charging or heating until the cathode is gone.

4. CQ .. CQ - I am told there is an amateur radio operator in Allen Texas who is using a TI99/4A to control his repeater. It even talks! I will try to find out more. I am also a HAM but have not linked computer to radio.

Thanks for that Jerry. Glad to hear you enjoy reading the newsdigest.

Kevin Cox, from Coffs Harbour, wrote to tell us all how he uses his computer.

"It all started in 1983 when we decided to buy the children a computer to further their education. Along with the computer we also purchased a new portable TV to act as a monitor plus a couple of games modules. After a period of time when the novelty of having a computer in the house had worn off it was left idle for a period of time, and that is when I decided to have a fiddle with it as I was not interested in the games (maybe because I was not very good at them).

"I started to read the instructional booklets which came with the machine, and knowing nothing at all about computers, I was amazed when things started happening on the screen. With basic programs for the Texas being very scarce I started entering other basic programs and ran into a lot of trouble with non TI basics.

"After constant visits to the newsagents I was able to purchase some magazines devoted to the TI. I then purchased a Mini-Memory module and began to explore Assembly Language but with very limited success as programs for the Mini-Memory were practically non-existent. I did however learn the basic concept of the language.

"I then advanced to a disk drive system which was really heaven compared to the cassette recorder. I then discovered FORTH language. At first I could not make head nor tail of it, but after repeated study it soon started coming together and I think it is one of the better languages for the Texas, being able to edit, compile, run and save the program without having to leave the original program.

"With other languages available for the Texas such as "C", Pilot, "G", Pascal and more, the introduction of utilities such as "XDP" and lately The Missing Link, the TI user has plenty to choose from, not to mention Ram Disks, 80 column cards, new keyboards and extra memory.

"Even though I own a PC clone, I continue to use the Texas. It has not the speed of other machines but is much more user friendly and with a good backup service in the TISHUG Group, the life of the Texas has gone beyond all expectations."

Thanks for that Kevin. It is always good to get comments from members.

The July Micropendium has just been received and you can check it out in the publications library. The list of contents in this issue includes:

Regena on BASIC - Rise and Fall Game.

Extended Basic - Robot Teacher.

BASIC/Assembly - Putting TI Artist instances on your screen.

The Art of Assembly - Some 'primitive' source code sections.

Micro Reviews - TI Casino and Harrison Software Music.

12 Mhz TI - Accelerator chip turns your TI into a very fast machine.

Just before I finish, can anyone tell me where member Peter Schubert has got to? His last newsdigest has been returned marked Left Address.

TIM Installation

by Al Lawrence, former HV99er

I have just finished putting (TIM) TI Image Maker & Son of a Board (SOB) from Oasis Pensive Abacutors, Canada in my console, with no major problems encountered, and it worked first go on an Amiga monitor. A set of comprehensive instructions, with diagrams are included and only one wire to solder from each board. Once the computer is disassembled and the motherboard uncovered, installing the SOB is a matter of removing GROM 0 and GROM 1 marked as CD2155 and CD2156 and taking the small 2" x 2" SOB carefully and plugging it into the empty socket next to the CD2157 chip. Next you have to solder the yellow wire attached on the SOB to the wide ground at the edge of the motherboard. Now re-assemble everything together with the console connected but not the PE box. If everything was OK you will see the new OPA Micro-Manager title screen on power-up. Now run a simple test programme to ensure you have removed the correct GROMS.

If everything worked, switch off, unplug the motherboard, remove the 40 pin Video chip TMS9918, firmly plug the TIM board into the socket, turn the board over and solder the blue wire to A13 of the system address bus. This is used by the V9958 chip in decoding the memory mapped I/O ports. Re-assemble as before, and even if you have not got the monitor cable or adaptor made you can still test that it works if you have a speech synthesizer. You will hear the computer say "READY TO START" by pressing any alpha key and then it will honk more speech - "DISK" "MODULE" - at you. If you have no speech then you will only hear a HONK each time you press a key. After it all works it is only a matter of re-assembling it all. You are no longer able have the top shield in place but this is no big deal. Also you have to cut a slot in the back of the case as the edge of 25 pin connector protrudes over the line of the casting.

There is, as I said, a very-easy-to-follow instruction set included and most hackers should find it simple. It works straight off on the Amiga Monitors and the pin outs are included but the Wang requires some mods either to the Monitor or by inverting the Sync signal output, Tony McGovern off the top of his head and fingers put a circuit out for me to try and after a Saturday afternoon well spent (downtown?) at F'Web Farm it all came together and I now owe Tony a BIG RED.

For further information contact Gary at:

Oasis Pensive Abacutors
432 Jarvis St.# 501-502
Toronto, Ont. M4Y-2H3
CANADA

Total cost is A\$205 if you get the right day for our dollar rate and much cheaper to send the total in Canadian currency.

The FunnelWeb environment? Well, it is still growing and Version 4.40 (final????) is now released. The main BIG difference is the 80 column Editor no longer requires the 40 column Editor on the disk. You can now shift between 40 and 80 column mode when the new hybrid 80 column Editor is in Show Directory mode by the simple use of Ctrl 4 or 8 keys. Disk Review is also improved as well as some minor bugs and tidying up in the Configuration. I know that I and a lot like me still reach for F'Web as it is simply the BEST for many of the tasks that in days gone by required heaps of \$\$\$ and much changing of disks, modules etc. With TIM and SOB boards from O.P.A. in my console and all the Quest and Horizon RAMdisks in the PE box, I am content to use the TI system as long as there is life there. It is still a very powerful setup. And when I can get using it with the Amiga for the Flight Simulators it will be great fun. With the Emulators for the TI and Amiga that enable ASCII files from the other machines to be loaded, transported and worked on, hopefully it will be a long, long time before any NEED for a Clone arises.

Best wishes to Geoff Phillips and I hope he gets the support he deserves in co-ordinating the new HV99 club with the smaller number of dedicated persons who are staying with the RR-99/4A RR?? Yes, for with all the additions it surely is a Vintage Rolls Royce now.

LONG LIVE THE TI!

continued from page 10

1. Load the game in its normal way.
2. Press FUNCTION-QUIT when the title screen appears.
3. Select Extended Basic and wait for prompt.
4. Type in the following:

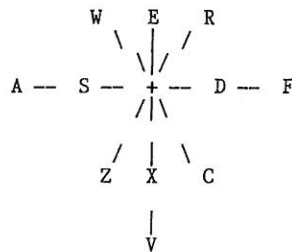
```
CALL LOAD(-18146,16,1) [ENTER]
CALL LINK("RUNNER") [ENTER]
```

You NOW have UNLIMITED misses!!

With Star Trek, the cheat mode does not let you choose any options, it just makes it impossible to die, i.e. you cannot lose any of your shield etc.. Also note that the computer will not let you know that you are in cheat mode once you press ***, it just continues in the normal way, except you cannot die!!!

It also came to my attention at the meeting about how few people knew how to play SPAD XIII (the best flight simulator available for the TI). For diagrams and more detailed information see TND November 1988 and TND February 1989, but for now, I will give you a brief tour!

Stick Movement



Bomb ----- B
Kill Prog. - K
Front view - U
Fire ----- Q
Move ----- M

Note: "A" is strong left, "F" strong right. "V" is strong down.

Rudder

```
Left --- ","
Right -- ". "
```

Views

Front ---- 1
Left ---- 2
Right ---- 3
Back ---- 4
Up ----- 5
Down ---- 6

Throttle

```
Down --- 7
Up ---- 8
Top ---- 9
Stop --- 0
```

Well that is the end of another article for this month. Please feel free to contact me on the BBS, username GAMES or write to the below address.

Mr Robert Brown
GAMES INFORMATION
141 Beecroft Road
Beecroft 2119
NSW Australia

COMING UP:

- * Maps of your favourite adventures!!
- * How to design your own TI Runner screens
- * Reviews of that messy game Microsurgeon, and the ultimate Pinball game for the TI!

STAY TUNED!!!

SAME TIME
SAME PLACE
SAME MAGAZINE
SAME AUTHOR

This month there is no shop news to report except to remind you that Rolf has released some new commercial software, including several modules, at the August meeting which I have added to the Commercial Software list below so if you would like to purchase any of them would you please get in touch with me as stocks are very limited.

We now have plenty of 8k and 32k memory chips in stock for the 32k Eprom ramcards as well as a supply of TI Artist Plus and Funnelwriter Eproms so hopefully those members building Eprom Ramdisks will take advantage of this and maximise the capacity of their Ramdisk.

PRICE LIST.

5.25 in. DSDD Disks (Boxes of ten)	\$6.00
5.25 in. HD Disks (Boxes of ten)	\$12.00
3.5 in. DSDD Disks (Boxes of ten)	\$12.00
5.25 in. DSDD Half Height Drive (New)	\$65.00
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
Eprom Ram PC Board	\$45.00
Eprom Ramdisk Basic Kit	\$35.00
Funnelweb Eprom Set (3 Eproms)	\$36.00
TI Artist Eprom Set (2 Eproms)	\$24.00
32K Static Ram IC (62256)	\$10.00
8K Static Ram IC (6264LP)	\$5.00
74LS08 IC (quad Schottky)	\$0.50
1K Resistor	\$0.05
Exchange Console	\$30.00
ROS Version 8.14	\$12.00

NOTE: ROS 8.14 must be purchased with first Eprom Set.

COMMERCIAL SOFTWARE.

Artoons SSSD	\$12.00
Character Set Graphic Design Cataloguer.....	\$6.00
Character Set Graphic Design I	\$12.00
Character Set Graphic Design II	\$10.00
Character Set Graphic Design III	\$14.00
Display Master	\$15.00
Genial Traveler (SSSD)	\$6.00
Microdex I (SSSD)	\$16.00
Microdex II (SSSD)	\$11.00
Nuts and Bolts #1 (DSSD)	\$6.00
Nuts and Bolts #1 (SSSD)	\$7.00
Page Pro 99 version 1.6	\$28.00
Page Pro Utilities	\$17.00
Page Pro Applications #1	\$2.00
Page Pro Line Fonts	\$9.00
Page Pro Medical Clipart	\$9.00
Picasso Publisher Version 2.0	\$14.00
Picasso Publisher Support Disks	\$6.00
Picasso Applications Disk	\$2.00
Rockrunner (SSSD)	\$15.00
Spell It! (DSDD version)	\$24.00
Spell It! (SSSD version)	\$27.00
The Missing Link (TML)	\$28.00
The Missing Link Companion Disk	\$2.00
TI Artist Plus	\$25.00
TI Base Vers 3.01 (SSSD)	\$25.00
TI Sort SSSD	\$15.00
Tris Module	\$25.00
Typewriter Module	\$25.00

From time to time we manage to get hold of unwanted TI Systems so if you need a particular piece of hardware for your own system please let me know and we will see if we can find it for you.

Jerry Keisler, one of our members in the US, wants to know Thomas Opheys' address. The address that I have for him is: Franz-Stockbauer-Weg 1, App. 88 W-8390 Passau, Germany

Jerry, if you do write to him, please mention that you first read about his TI99/4A emulator in the TND.

Software Releases for September 1991

DISK A212 is a re-release of 'G', a high level graphics language developed by Gene Krawczyk of ATICC, the Adelaide TI99/4A User Group. What does it do? This question is answered in depth by Bob Warren, in another article in this month's TND. Highly recommended!!

DISK A385 is called 'Cartoon Kit', a drawing aid by Tim O'Neil, designed to allow the budding artist to draw cartoon like characters. TI-Artist Plus (or V2.01) is required to run this disk. A full description on using this program is given elsewhere in this issue.

DISK 406 is 'Hyphenator' by Wayne Stith, a TI-Writer utility described fully elsewhere in this issue.

DISK 407 is the first of two disks of US baseball team logos by Deanna Sheridan. This disk contains 14 pictures of American League teams, in TI-Artist format. A suitable program is required to view or print out these images.

DISK 408 is the companion disk to A407 and contains 12 pictures of US National League teams, in TIA format.

Tigercub Software Releases

TCC-2 is the second disk in the 'Tigercub Collection'. The disk is menu driven and offers six options, which are: Handy Dandies #1 (13 programming routines and 14 useful DEF statements), Handy Dandies #2 (24 programming routines), Handy Dandies #3 (12 useful routines, including a perpetual calendar), Colour Programming Tutor, Music Programming Tutor, and Sprite Editor. This disk is just brimming with programming ideas and demonstrations, and is highly recommended to anyone wanting to learn to program in TI or Extended BASIC, or just looking for new ideas.

Commercial Software

Due to problems at Asgard (just ask Larry Saunders, he knows!), no new software has been received from them. I am currently waiting on software, as well as hardware such as the MIDI interface and the 80 column upgrade from OPA. Installation of this device is the subject of another article, from a member of the former HV99ers (the first of many contributions, I hope).

Attention: BBS Members

from Ross Mudie

How long is it since you changed your password on this BBS? Please change your password reasonably regularly and always if it is used at meetings etc. Ross Mudie, SYSOP, 3rd August 1991.

Packaging and postage: Surface Airmail

1 to 2 Disks -----	\$1.90	1.90
3 to 9 Disks -----	\$2.90	\$3.60
10 to 20 Disks -----	\$3.90	\$4.80
TI Artist Plus -----	\$3.00	\$3.70
Display Master -----	\$3.00	\$3.70
TI Base -----	\$3.00	\$3.70
TI Sort -----	\$3.00	\$3.70

Multiple Menus for BOOT

by Col Christensen, TIBUG

The program BOOT is now used widely by floppy drive and RAMdisk owners. Some hard drive users also find its new features a benefit for many of their applications. Its 24 menu options provide a wide selection of programs that can be quickly loaded by a simple keypress.

Now that RAMdisks of greater than 256Kb (even up to 1.5Mb) are being built, owners of these are finding that the provision of just 24 options is a little restrictive. We are never satisfied, are we? Just think though, without a want or need for something better, there would never be progress. We might still be making do with things like the "OLD CS1" command.

One way to increase the number of menu options is to have a second BOOT menu, with the second version renamed, so that if the first menu does not show the required program, pressing a particular option will load the second version of BOOT, with a further 24 options. If need be, more than two versions of BOOT could be chained together in this way.

One problem that raises its ugly head with that system is the method of configuring each menu. Once you configure the menu and agree to save the alterations back to disk, the whole boot program in memory, complete with alterations, gets saved back to disk as the two files BOOT/BOOU. How does it know to use this filename? Simple. The filename is contained in the code of the boot program. But what if you had renamed the files to COOT/COOU for your second menu program and configured it with 24 more menu options? When you save that configuration back to disk, it gets saved as BOOT/BOOU, overwriting the original files. This happened to me quite a number of times until I woke up to what was happening and I became quite adept at retyping the menu names and filenames to repair my first BOOT program. What I should have done was to use a sector editor to change the filename imbedded in the program from BOOT to COOT.

Anyway there had to be some other way to use more than one menu list with my 1Mbyte RAMdisk without having to go in with a sector editor and modify code in the multiple copies of BOOT/BOOU. Finally a method came to light when I was playing around with the "[" and "]" keypress facilities. The latter will save ("Put" - to quote the screen prompt) a list of your menu names and filenames to a DIS/VAR 80 file under a filename of your choice. The other keypress will retrieve ("Get") them from disk and instal them in the current copy of BOOT in memory. This is a terrific addition to BOOT and would have saved me much re-typing, had I known of it earlier.

You have probably guessed by now how you can have multiple menus, using just one copy of BOOT/BOOU on the RAMdisk. Yes, it is by having multiple menu lists. These load very quickly following the "[" keypress and in an instant the next menu is on the screen.

Now a few tips on how to set up a number of menu lists. First, though, configure your BOOT program with your most commonly used programs on its menu. Press "]" and save (Put) the menu list to "DSKn.1-CFG" where "n" is your RAMdisk number.

Remember: Your BOOT menu list is to be saved under DSKn.1-CFG, the second menu as DSKn.2-CFG and any further menus as DSKn.3-CFG, etc. Now you can configure the defaults in the BOOT program.

DO NOT PRESS <ENTER> AFTER TYPING THESE DEFAULTS:-
In turn enter your defaults for the following keypresses:

- "1" Directory
- "2" View a File
- "3" Run
- "X" Xrun
- "P" printer devicename

"D" delete a file
"T" track disk

Do this by pressing the key, typing the default and finish by pressing <BACK> to escape.

Press "]" and type "DSKn.2-CFG" for the "Put" default and press <BACK> to escape. DO NOT PRESS <ENTER>.

Next press the "F" key sequentially until your favourite foreground colour appears on the screen then press the "B" key to set the background colour.

Your defaults are now housed in the BOOT program in memory and all you have to do is to save it all back to the RAMdisk. The only way to do this is to go back to editing the menu list. Press FCTN <5> to get to the screen editing mode, then FCTN <9> and you will be asked which drive number to save BOOT to. Once you have typed your RAMdisk number, YOU CAN THEN PRESS THE <ENTER> KEY. Press <BACK> in answer to the next question about GRAM number.

After the above, you can set up all the other menus that you need. It would probably be best to load a copy of BOOT/BOOU from your original disk that lists a menu of "option 1" to "option 24". DO NOT PRESS <ENTER> UNTIL ADVISED. Press FCTN <5> and type in each of the program names and related filenames. Make one of the program options "FIRST MENU" and the related filename to load as DSKn.BOOT. Then press FCTN <9> twice if you want to do the other two screens of options. When you have finished editing, again press FCTN <9> twice to escape. What you want to do is to save (Put) the list of program names and filenames only to the RAMdisk by pressing "]". After the "Put:" prompt, type DSKn.2-CFG for the second menu (or DSKn.3-CFG for the third and so on) AND PRESS <ENTER>.

Repeat the above paragraph for any subsequent menus you need. Press FCTN <=> to quit and if auto power-up is ON, your first menu appears on the screen. If you want to use one of the other menu lists, simply press "[" and the default "Get: DSKn.2-CFG" appears. Press <ENTER> to load the second menu list. Slick, huh? (I am intrigued by that "huh" used by a number of our overseas counterparts). For any other list, type over the "2" with whichever menu list you want. Slick again, huh?

All subsequent menu lists will have an option called "FIRST MENU" for loading BOOT so it is an easy matter to return to it. To get from the second menu to any other menu even the first one, press "[" and change the "2" in "Put: DSKn.2-CFG" to whichever you want.

Editor's note: This version of BOOT was released as Disk A386 in August 1990, and is still available from Percy Harrison at meetings or by mail order. O

continued from page 22

```
180 DATA 400,77
190 FOR A=1 TO 4 ! rows
200 FOR B=1 TO 2 ! columns
210 READ AMT(A,B)
220 NEXT B :: NEXT A
230 ! end of input routine *****
240 !
250 DISPLAY AT(6,1):"ENTER DESIRED SYL
K FILE NAME DSK"
260 ACCEPT AT(7,4)BEEP SIZE(-15):FILE2
$ :: IF FILE2$="" THEN 260
270 OPEN #2:FILE2$,DISPLAY ,FIXED 128,
OUTPUT
280 DISPLAY AT(9,1):"Enter DESIRED nam
e for 2nd column." :: ACCEPT AT(10,1
0)SIZE(10)VALIDATE(UALPHA):NAME$ :: IF
NAME$="" THEN 280
290 R$=CHR$(13)&CHR$(10)
300 T$="ID;PMP"&R$&"F;W1 2 6"&R$&"F;DG
2G8"&R$&"B;Y"&STR$(A+2)&"X"&STR$(B)&R
$
310 T$=T$&"NN;N"&NAME$&"ER1:6C2"&R$
```

continued on page 19

G, an Introduction

by Bob Warren, ATICC

G can stand for many things, including GRAPHICS, GREAT, and GOOD FUN. The graphics language G is all of these things. It was developed by Gene Krawczyk, one of the original members of ATICC (in Adelaide, ed.), and he has made it available to the Club. It was written in Assembly Language, and so requires a disk drive and memory expansion to run.

G is a powerful graphics language with simple commands, which makes it ideal for children, but it is also sophisticated enough to provide a challenge for adults. Unlike TI-Artist, which only produces still pictures, G allows a form of animation, and can rapidly change screens. A screen can be STORED at any time, a new screen drawn, and then the old screen RESTORED when required. A screen can also be SAVED to, or LOADED from, a disk when needed.

I have edited Gene's instructions and commands to produce a beginner's version of G which everyone can use. Details of this version are given below. Gene also included some commands which allow an experienced Assembly Language programmer to execute Assembly routines, including sprites (I think). When I have sorted these out and written instructions for their use, I will present an Advanced G (not available at this time, ed.).

I hope there will be sufficient interest for a regular column in our bulletin to be devoted to G, both to answer queries and to exchange programming ideas. ○

G, the HIRES Language

by Gene Krawczyk, ATICC
Documentation by Bob Warren, ATICC

Loading and operating G

To load G from Extended BASIC insert the G system disk into drive #1 and select Extended BASIC. This action automatically loads G into memory, and the program will then attempt to load & run a file called 'GLOAD', if this file is present on the disk (in the same way that the system attempts to load & run a file called 'LOAD' everytime you select Extended BASIC). In this instance 'GLOAD' is a demonstration program, but any program can be called 'GLOAD', if you want it to automatically run everytime to boot up G. If 'GLOAD' is not on the disk in drive #1, the program will immediately go into menu mode, described below.

To load G with the Editor/Assembler module, select '5' from the E/A menu, then type in 'DSK1.GEE' (without the quotes) and press <enter>. The program will boot up by running the 'GLOAD' file described previously.

To escape at any time press FCTN <9>, which will return you to the menu mode. The screen displays the following message:

G a graphics programming language
(L)oad,(S)ave,(I)nit,(E)dit,(R)un.

Pressing any of these five keys will put you into that mode.

(L)oad

Expects input from the keyboard, e.g. DSK1.SAMPLE will load the file 'SAMPLE' into memory. These files are in DIS/VAR 80 format without any control characters such as carriage returns. If you load another file the old file will be lost. If the load is successful, the program will be tokenised and then run.

(S)ave

Will save the file in memory, e.g. DSK1.SAMPLE will save what ever is in memory to the disk drive. If there is nothing in memory then it will save an empty file to the disk. Note: loading an empty file into memory (e.g. 'DSK1.NEW') is a good way to clear the memory, since no command exists which can do this directly. You can also save to a printer or other peripheral by typing in the required device name, e.g. TP will save the file to the thermal printer.

(I)nit

If you select (Y)es to the prompt to initialise a file, the program will clear the memory pointers and restart. This is similar to a cold start and the program will try to auto boot a file called 'GLOAD' from drive #1.

(E)dit

Selecting 'E' will put you into the Editor. The key presses used to move the cursor around the screen are the same as the ones used in TI Writer. The main difference is that this editor does not window its screen. Instead, each line of 80 columns is displayed as two consecutive rows of forty columns.

The following key presses are used in the Editor:

To delete a line press FCTN <3>
To insert a line press FCTN <8>
To scroll the screen up press FCTN <4>
To scroll the screen down press FCTN <6>
To delete a character press FCTN <1>
To insert a character press FCTN <2>
To go back to the menu press FCTN <9>
To run the file in memory press FCTN <5>

(R)un

This command does the same as in TI BASIC.

Now for a few simple programs to get your feet wet. Note: anything in parentheses is the key press you have to make. From the menu mode, with the original disk in drive #1 press (L) to load a file. Type in: DSK1.GLOAD <enter>. This will load and automatically run a small demonstration. To stop the demonstration, press and hold FCTN <9>; this will take you back to the menu. To look at the program, press (E) to enter the editor.

All commands will be typed in uppercase on the left side of the screen. A small example will be shown first and then explanations.

```
SET 10 20 TO 100 150
```

This will draw a line starting at location 10 across and 20 down as the starting point. The line will then be plotted to position 100 across and 150 down. All co-ordinates are given as x y, the x is columns and the y is rows. The value of x can be from 0 to 255, and the value of y is 0 to 191. Notice also there are no commas or parentheses in this language to separate variables or commands. The separator is a space (the same as in FORTH).

```
SET 10 24 TO 134 180  
STOP
```

This will put a line on the screen and then it will just stop. When you press a key you will be taken back to the command mode. The actual command SET puts a dot on the screen and the command TO draws the line from the old position to the new co-ordinate. This allows you to have as many TOs as you like to draw shapes. So the command TO always draws from the last plotted point to the new one. If you just want to put a dot on the screen you would use:

```
SET 100 100
```

Description of Commands

ANGLE

Sets a new angle for the DRAW command and can be between 0 to 360 degrees. Other values can be used,

but if numbers greater than 360 are used, they are just reduced to a value between 0-360. The use of larger numbers may also slow down execution time. For example:

ANGLE 250

sets the angle to 250 degrees.

ARC

Allows you to draw circles and arcs, e.g.

ARC 100 100 25 25 0 360

draws a circle at co-ordinates x=100, y=100 with a diameter of 25 pixels.

ARRAYS

Only one dimensional arrays are allowed, in the format @(a), where 'a' can be any integer between 0 and 500, expressed as a decimal or hexadecimal number.

BCOLOR c

Same as COLOR, but changes the background color. The default is clear. Seems only to be used in TRACE 4.

BGND

Sets the background color. The same rules apply as for FGND, but only works on the background colors.

BOX x y a b

Draws a box on the screen whose top left hand corner has the co-ordinates of 'x' and 'y'. The box is 'a' amount wide and 'b' amount tall, e.g.

BOX 100 100 50 20

puts a box on the screen which is 50 pixels wide and 20 pixels tall, with the top left hand corner located at x=100 and y=100, remembering that the left hand corner of the screen has the co-ordinates x=0, y=0.

CLEAR

Clears an area of the screen.

CLS

Clears the graphic screen without first saving it. If you accidentally use this command, you would have to design this screen again.

COLOR c

Changes the foreground pen color to a new value. The value is from 0 to 15. Any value larger than this is truncated.

CONSTANTS

Constants can be integers, in decimal or hexadecimal format.

COS

Gives the cosine of a number.

DISPLAY

Prints data to the screen, but erases what ever is underneath.

DRAW

Draws a line from the last point plotted, at the angle specified in the ANGLE command.

FGND

Sets the foreground color. For example,

FGND 6 sets the foreground color of all "on" pixels to red.

FGND 6 0 255 sets the "on" pixels from 0 to 255 to red.

FILL x y

Fills in a area on the screen. This is not a very exact type of fill but it may do the job. It might have to be used a few times, however, to fill in awkward shapes. You may have problems with this command.

FOR NEXT STEP (optional)

Same as in TI BASIC, but only 10 levels of nesting are allowed, e.g.

FOR I=1 TO 1000 NEXT I

produces a delay.

FORMAT

Sets up the distance between characters for use in the print statement. The normal distance is 8, and in forty column mode the normal distance is 6.

GOSUB :LABEL

Same as GOTO except a return is stored so you can do subroutines. There are only forty levels of subroutines allowed in G.

GOTO :LABEL

Does a branch to the label in the file, e.g.

:SAM GOTO :SAM

would give the same result as 100 GOTO 100 in TI BASIC.

All labels must be preceded by a colon (:), and must not be more than six characters in length, seven counting the colon. Also the label must be the first word on a line, otherwise the search routine will not find your label. If you put a space in front of the label the word will also be missed.

IF THEN

Similar to the TI BASIC 'IF' expression. If true then do rest of line, else go to next line. You can not do this though:

IF A=10 THEN :LABEL

is not correct; you must do this instead:

IF A=10 THEN GOTO :LABEL

INVERT

Inverts an area of the screen.

KEY\$

KEY\$ A puts the key press into variable A. No key press means A=255.

LET

Assigns a value to a variable, e.g.

LET A=10

gives 'A' the value of 10, and

LET A=10*3+4/2+C*B

will give 'A' the value of the calculation. The value is worked out from left to right without precedence among the mathematical operators, as there is in TI BASIC. Also you cannot use parentheses to change the way you want the program to calculate the value.

As in TI BASIC the LET comand is optional, but if you want speed use it, as LET quickens the processing time. Without LET there is a lot of checking.

LINestyle

Changes the type of line being used.

LOADS

Loads a bit map screen from disk in GRAPHX format, e.g.

LOADS DSK1.SAMPLE

loads a screen called 'SAMPLE' from drive #1.

PATTERN

Alters the fill pattern.

PRINT x y a or
PRINT X Y "SKFFK"
Prints on the screen at location x y the value of 'a', which can be a variable or a number. You can have as many variables as you like in one print statement, e.g.

```
PRINT 10 10 A B C D 100
```

prints on the screen the value of a, b, c, d and the number 100. If you put a quote after the co-ordinates, then the characters after the quotes are displayed on the screen. The string of characters can be terminated by a quote, or by leaving the rest of the line blank.

REM

As in TI BASIC, it allows you to type in remarks in the program. When the interpreter comes to a REM statement it automatically goes to the next line.

Additionally, the program does not alter your file in any way, so any extra spaces you type in are left there.

RESTORE

Puts up the old screen.

RETURN

Returns from a GOSUB, the same as in TI BASIC.

RND

Gives a random number, e.g.

```
RND 255 A
```

generates a random number between 1 and 255 and puts it in A

SAVES

Saves a screen to disk in GRAPHX format.

SCREEN c

Alters the color of the screen to the value of c, which can be from 0 to 15.

SIN

Gives the sine of a number.

SIZE

Alters the size of the characters in the print statement.

SIZE a

Alters the size of the the characters in the print mode.

SIZE 0 default, same size as normal.

SIZE 1 double size characters.

These are the only two sizes available at the present time.

STOP

Waits for a key press, then returns to the main menu.

STORE

Saves a screen to memory, recalled by the RESTORE command.

TRACE a

Changes the drawing mode of the pen.

TRACE 0 (delete)

From now on all pen commands will erase.

TRACE 1 (draw)

This is the default drawing mode.

TRACE 2 (invert)

All pixels that are 'on' are now turned 'off' and all pixels that are 'off' are turned 'on'. A very useful mode.

TRACE 3 (special color mode)

Will not alter any pixels, but will just alter the

foreground color of the pixels. This will only be noticed if you have a pixel 'on'.

TRACE 4 (special color mode)

This mode will show up straight away, as you do not draw with pixels but with background colors. The background color needs to have been set up already with the BCOLOR command, so now all drawing will be 8 pixels wide by 1 pixel tall. A strange mode but useful for some types of work, particularly for clearing backgrounds.

VARIABLES

Variables are the single letters of the alphabet.

WRAPOFF

Does not allow screen wrap.

WRAPON

Allows screen wrap.

Summary of Commands

ANGLE A (0-360)

ARC A B C D E F

ARRAYS @(10) @(B) (range 0-500)

BCOLOR A (0-15)

BGND 11 0 2000

BOX 10 10 110 110

CLEAR 10 10 100 110

CLS

COLOR 6

COS 32 B

DISPLAY 10 10 "STRING"

DISPLAY A B VARIABLE VARIABLE

DRAW 50

FILL 10 10

FGND 11 100 200

FOR I=TO 1000 NEXT STEP

FORMAT 8

GOTO :LABEL

GOSUB :LABEL

IF A=10 THEN (continue rest of the line)

INVERT 10 10 110 110

KEY\$ A

LET A=100

LINestyle A

LOAD DSK1.PROGRAM (DIS/VAR 80 format)

LOADS DSK1.SCREEN (in GRAPHX format)

PATTERN

PRINT

REM

RESTORE

RETURN

RND 200 A

SAVES DSK1.SCREEN (in GRAPHX format)

SCREEN 3

SET A B TO A B

SIN 23 A

SIZE 0

STEP (optional)

STOP

STORE (saves screen to temporary memory)

THEN (optional)

TRACE A (A=0 to 4)

VARIABLES (single letters A-Z)

WRAPOFF

WRAPON

G Corner

by Bob Warren, ATICC

Last time I gave a list of the commands for G and a general guide to running G. However, the instructions for two of the commands, SIN and COS, were wrong. This time I intend to show the correct way of using SIN and COS, and to illustrate them in some simple, but effective programs. Those who are a little rusty on trigonometry can skip straight to the programs.

The problems with SIN and COS arise because the values of sin and cos vary between +1 and -1. Since G only
continued on page 16

Games Information

by Robert Brown

Welcome to article 2, of Series 2. I hoped you enjoyed the first article, I know I did!

At this stage, I would like to thank Rolf Schreiber for his nice comments about my articles (see page 5, Aug 1991 TND). I would also like to clear up a few points with the editor (see page 12, Aug 1991 TND). In my last article, I said "...the TI could no longer be used for that flash newsletter or that big spreadsheet." What I meant is that flash newsletter, with very high density graphics, along with scans, ie DTP (Desk Top Publishing of high quality). Bob, you should be very proud of the TND, as it is the BEST looking TI Magazine in the WORLD!. Now lets get on with this interesting, very informative article....

This month I hope to review some games, old and new ones. I would also like to hear from members, if they would like me to include solutions to some of the "Sierra" games available on other computers, along the lines of Police Quest, Space Quest etc... Speak up if you do!

The first game I will look at is JUNGLE HUNT. This game has been available for the TI for sometime now, on all medias i.e. Module, Cassette & Diskette.

The idea of the game is to rescue your so called GIRLFRIEND, who has been captured by the cannibals. To do this, you must firstly swing through the jungle on ropes, then swim through crocodile infested waters, not forgetting to come up for air! Then run and jump over falling boulders and then, if you have enough strength, fall off the cannibals that have captured your women!

You swing the trees and jump from the rope by pressing the fire button. When you eventually reach the water, you will see crocodiles. You can either swim around them, or you can kill them with your knife. BEWARE, you can only stab them when their mouth is closed! (Press the fire button to stab with the knife.) As I mentioned before, there is a bar showing your available air left, which decreases with your time under the water, so make sure you surface to replenish your supply!

The next stage is to survive a bombardment of rolling boulders, both small and big ones! By pressing the fire button, you can jump over them. To gain extra height, you can hold the joystick in the left position and also press the fire button!

Once past the boulders, you finally met the mean looking cannibals. Here you use the fire button to jump over their spears. When you finally rescue your girl, you get this message... "You finally leap up to your love for a tearful reunion, followed by a reward (you dirty minded people) of additional bonus points for any time remaining."

Overall, Jungle Hunt is an okay game, perhaps a little easy for the more advanced game players, but still one whole looking at!!!

Now, as mentioned in the last GAMES INFORMATION, I would do a review of APLINER, for all those ski mad people in the TI community. (For readers not in Australia, please note that it is the middle of winter here!!!)

This game was made way back in 1982 by the one and only Texas Instruments. They did an okay job I suppose! The game is better played with speech, as the game will warn you on falling objects etc..

Once you have finished loading the game, you have the option of choosing 1 or 2 players, and then typing their name into the computer. The next screen will show you a listing of the six mountains (with their heights) that you must climb! Press any key and you now find

yourself at the bottom of a mountain. On the screen their is some valuable information. Some of the info is your level, seconds allotted to your climb, along with the mountain name and height. The elevation reading will climb from zero, upwards when you begin the climb. You start with four men, using only one at a time, and the final three being represented by the boots at the bottom of the screen. Also the current score, along with the high score is show. When a two player game is being played, the high score shows the second person.

The object of the game is really simple... get to the TOP!!!! Eight directions are available to move in... use them ALL! There are 6 mountains to climb, that rotate through 3 levels, thus 18 levels in total.

One must beware of the falling objects. These include rockslides, avalanches and icefalls. The higher the level, the fast they fall, also going right for YOU, instead of being random. There are numerous fixed obstacles on the way up. Some when touched do nothing, other will make you fall back a few metres, or even worse still, right to the bottom! Note - watch out for the ABOMINABLE SNOWMAN at the top of Everest!!

There are 2 types of bonuses available in this game. The first is if you get to the top in the given time. The second being the gems that you CAN pick up on the way. These gems appear on the animals, and can be obtained by touching the animal, but BEWARE, if you hold down the fire button to long, you may get ZAPPED by the animal.

One interesting point to note is that you can wrap around the screen, meaning walk off the left hand side onto the right hand side, and also right to left. The sounds are okay, along with the speech, which adds enjoyment to the game... in other words, worth trying!!

After attending the TIsHUG meeting today, (the first for some 2 years) I came across the need to go over some of my earlier articles, mainly for the new members, especially Peter (I forgot your surname) and a gentleman there with his nice family (Sorry I do not know your name!)

The subject came up about TEST or CHEAT modes. These test modes were put into the games by the author (mainly Texas Instruments), to try the different levels, thus they were called test modes. Here I quote from my FIRST article (see page 7, March 1988 TND).

"...You may not know this but quite a few of the old TI modules have the hidden test modes. This allows users to 'CHEAT' in several ways, for example, change levels or lives or speed depending on the capabilities of the game.

"The games that I know that do this are... Alpiner, Chisolm Trail, TI Invaders (Disk & Tape versions only), MunchMan, Munch Mobile, Moon-Mine, Star Trek, Hopper and TI Runner.

"How do we do this sir?, I hear you ask. Well my son, it is simple. All the above games except TI Runner, Chisolm Trail and Hopper you can type '###' which is 'SHIFT-8-3-8' at the title screen. With Chisolm trail you must wait until it gives you the prompt LEVEL (1-9) then quick as a flash type ### and you are in. With Hopper, you only type SHIFT-8, then you get to choose the level you want, from 0-9.

With TI Runner, there are a few CHEAT modes available. To skip over a level press FUNCTION-5, only when the man is flashing, thus straight after the "LOADING LEVEL X" message. Another CHEAT mode is to go to any screen in TI Runner. To do this type SHIFT-3 (only when the man is flashing) and then the man will disappear. You can now type the level you want, ranging from 1 to 50! NOTE, if you want levels 1 to 9, you must put a leading zero, i.e. level 5 is 05. Way back in August 1988, I also published a command for getting UNLIMITED misses in TI Runner. To do this you must firstly have the BBS version of the game, i.e. assembly embedded into X-Basic - see below. continued on page 4

Hyphenator version 2.0

by Wayne Stith, VA USA

Original program and documentation copyright 1985 by Wayne L. Stith.

Send \$10.00 for each machine on which this program is to be used to:

Wayne L. Stith
715 Timken Drive
Richmond, VA 23229

If you wish me to supply the program on disk, send \$10.00 (\$15.00 if you want the source code as well).

This utility allows the user to place hyphens at will in a document which was created in word-wrap mode on the TI 99/4A and saved in a DIS/VAR 80 file. The user may also make minor corrections.

To load:

TI-Writer utility option: Accept the default name of UTIL1.

Editor/Assembler option 5: Press <enter> at the prompt.

Extended BASIC: Run the LOAD program.

Using the program:

After passing the first screens, enter as the source file the name of the document you wish to process. For the object file, enter the name of the file you want HYPHENATOR to create. Left and right cursors are active with FNCT <S> and <D>. You may change the device name to any drive your system supports. (Cassette files are not supported.) Your source document is not erased. You may use identical filenames if different device names are entered, e.g. DSK1.LETTER and DSK2.LETTER. Avoid using the function or control keys to enter a space at the end of the filenames. They will be read as invalid codes.

The next screen will prompt you for the margins and indentation. Type in the values you included in your original document. You may use the cursor to move back and forth on a line. Press <enter> to enter the value. Press <enter> three times to accept the default values. Be sure to use three numbers for each entry; e.g. a left margin of "5" should be entered as "005". The program will accept only numbers; thus you may not enter a "+". Absolute indentation should not be used. The indentation is the number of spaces to be indented from the left margin. You may set margins large enough to handle condensed print. In theory one could enter a margin value up to 999, but such a value is useless. One could also enter a left margin value which is greater than the right margin value. In both cases unpredictable results are likely.

As the program processes your document, each record is tested to determine whether it should be displayed. Blank lines are not displayed, but are sent immediately to the new file. Lines which contain a carriage return symbol without text and which are not part of a paragraph are also not displayed. HYPHENATOR recognizes the format commands used in TI-Writer (e.g. ".IN" and ".CE") and enters them into the new file without displaying them. At the end of the program the tab settings line is not displayed, but is included in the new file.

An entire paragraph is entered into the program's buffer. HYPHENATOR looks for the carriage return symbol to determine the end of a paragraph; thus only documents written in word-wrap mode can be properly processed. There is space in the buffer for 5280 characters, equivalent to one 66-line, 80-column, single spaced page.

The edit screen displays the current margins and indentation at the right. At the top up to five lines of text are displayed, enough to allow for a line length of 160 characters. The proper indentation is shown for the first line of the paragraph even though you may not have indented on the screen when creating the document. A cursored character appears in inverted colors to indicate the last character the TI-Writer formatter would print if it could split a word.

The program recognizes internal formatting commands which are not printed but only alter the final print-out. The ampersand (&) indicates to TI-Writer that the following word is to be underlined. The "@" symbol indicates that the following word is to be overstruck. ASCII codes 0-31, entered in special character mode, send other commands to the printer. All these commands are recognized as unprintable. Although they are shown on the screen, the line length is adjusted so that the cursored character indicates the correct length. HYPHENATOR, like TI-Writer, recognizes a double ampersand or a double "@" as a character to print instead of as a command to the printer. The line length is adjusted accordingly. In addition, if you forgot to add two spaces after a period at the end of a sentence, TI-Writer will automatically add the second space when printing. HYPHENATOR adjusts the line length to account for this. Adjustments are based on the margins and indentation you entered, and not on the length of the displayed text.

Options on the edit screen:

CTRL <1>: Restart paragraph. Selecting this option brings you back to the beginning of the paragraph.

CTRL <2>: Change margins and/or indentation. Select this option to change margins, for example, for long quotes that must be indented further from both margins than normal text. Enter the new values the same way you entered the original ones. After entering the new values, the currently displayed paragraph is restarted.

CTRL <3/4>: Change colors. You may set the screen and text colors to your preference. This feature is active on most screens.

CTRL <5>: Next paragraph. The paragraph currently displayed is written to the new file and the next paragraph is displayed.

CTRL <9>: Finished. This option automatically ends the program by copying the currently displayed paragraph into the new file. Any changes you have made to THIS paragraph are ignored. The rest of the source document is then copied into the new file without your intervention.

<ENTER>: Chop off the line. Use the arrows (see below) to place the cursor on the last character of the word or part of a word which you wish to see printed out on one line. Then press <enter>. What happens next depends on the character the cursor is on. If the cursor is on the last letter of a word or on a space, the text up to that point is sent to the buffer as long as you have not tried to exceed the margins. The next portion of text is displayed. If the cursor is on any letter of a word except the last letter, the screen is erased after the cursored letter. You will be asked whether you wish to hyphenate the word. Answering "Y" or "y" will place a hyphen on the screen as long as the margins are not exceeded. Press <enter> to display the next portion of text. If you answer "N" or "n", the computer will understand the cursored letter to be in a word which should start the next printed line and will display the next portion of text accordingly. In any case, the program will not send any line to the buffer if the margins have been exceeded.

ARROWS: Move cursor. Use FCTN <S,D,E,X> to move the cursor to any position within the top five lines of the screen. You will note that the originally cursored letter retains its inverted colors to mark the last character the TI formatter would print if it could split a word. Use this end-of-line marker as a guide.

Whenever the carriage return symbol is present in the displayed text, an "end of paragraph" message is displayed. You may see this message several times, depending on whether the symbol is before or after the cursored letter. The cursored letter may also be "out in the middle of nowhere" if the last line to be printed out has just a few characters. Continue processing the lines normally. When the paragraph has been processed, you will be asked whether to write the paragraph to disk. A negative response will bring you back to the beginning of the paragraph.

The last screen allows you to process another document or quit to the title screen.

Minor editing:

You may make minor corrections to your text by simply typing the correct letter. Thus, you may change "recieve" to "receive". You may not add letters. If you delete a letter with a space, you will have an extra space which the TI formatter will ignore, thus forcing you to calculate the correct line length instead of letting the computer do the work!

The object file is written in 80 columns. The tab settings of your source document are saved.

Suggestions:

When deciding where to "chop" off the displayed line, remember that the TI-Writer formatter rigorously prints as much of the line as it can. Be certain that you do not delete too many characters, although you may actually chop off the line anywhere up to and including the originally cursored character.

When hyphenating words which are to be underlined, avoid splitting a word which begins with an ampersand, such as "&moonbeam". The result will be "&moon" and "beam", with the ampersand no longer applying to "beam". If you wish to hyphenate the word, you must return to TI-Writer to add an ampersand to "beam", but this is unlikely to cause a problem, since the added ampersand will not affect the line length. Likewise, avoid splitting a word preceded by "@".

The carat (SHIFT 6) indicates that a space is required. HYPHENATOR considers the carat to be just like any other character.

Should you place the cursor on the last letter to be printed, or on the space after it? It does not matter. The program automatically adjusts the next screen display (and cursor position) accordingly.

The program reads the screen - rather than the input buffer - to create the new file. When you press <enter>, the screen is read no further than the character you cursored. Any editing you may have done in the text beyond that point is ignored.

If you place the cursor on a hyphen which already exists in the original document and then press <enter>, the program recognizes that the portion of the word which appeared after the hyphen is to be the first word on the next printed line. The screen display is adjusted accordingly. You do not need to hyphenate the hyphen!

Creating a document in TI-Writer:

Avoid creating a document in which any part of the text is in fixed mode. Since HYPHENATOR looks for a carriage return symbol to signal the end of a paragraph, it will stack up lines written in fixed mode until it can find the symbol.

Avoid creating long lines without any spaces. While this is unlikely, it can occur occasionally. HYPHENATOR adjusts for this situation by writing 80 characters to the new file. This may cause strange results when the document is printed out.

Avoid placing extra spaces in your document, whether by

accident or by spacing twice after a colon or semicolon. The TI formatter will ignore extra spaces, but HYPHENATOR will treat them as printable text.

Occasionally macro commands are created with the transliterate (.TL) feature. While the program will automatically send the command line to the new file, it will not check out the logic of the commands. You may have to account for this when checking the line length.

If you look at the finished document via TI-Writer, you will notice extra spaces at odd places. These are due to the fact that the program adds a space to every line it sends to the buffer for the new file. They do not affect the print-out. O

Cartoon Kit Documentation

Program by Tim O'Neil, USA, documentation author unknown

Many "companion" disks of one kind or another are available for use with Chris Faherty's TI Artist, one of the best graphics utilities for the TI-99/4A. Such disks often contain many outstanding selections of Fonts and Instances to use with TI Artist, but so far equal attention has not been given to using Slides in a creative way.

Well, Tim O'Neill of Wilmington, Delaware has come up with a "cartoon kit" that is useful and fun for adult and child alike. You too can draw cartoon characters, with a little help from Tim's Instances and Slides!

Incidentally, if you don't have TI Artist and want to know where you can get it and/or "companions" related to TI Artist, two places you can check are Asgard Software, P.O. Box 10306, Rockville, MD 20850, and Texaments, 53 Centre Street, Patchogue, NY 11772. (Editor's note: TISHUG members should first check with Percy Harrison, our shopkeeper)

To use Tim's "cartoon kit", just follow the following steps:

- (1) Load in TI Artist (Version 2.01 or later)
- (2) Go to "2. Enhancements". You will be spending most of your time in this section of TI Artist.
- (3) "For Instance," start with a body. (Remember: you can't get ahead if you ain't got nobody.) You have a choice here of loading BODY1, BODY2, ... BODY9 as an Instance.
- (4) Then load in MOUTH as Slides, choosing one. Then NOSE as Slides, then EYES, then EARS, then HAT, each time choosing what you like and then placing it where you want it.
- (5) All of the preceding is done in the "Enhancements" section of TI Artist (Version 2.01). To finish your picture, by drawing the outline of your cartoon character's face, go to "1. TI Artist", and fill in the missing parts, using the Zoom option.

(6) Finally, Store your picture on disk or print out a hard copy. It wasn't hard at all, was it! Incidentally, I think you may already know this, but I have met some TI Artist owners who do not: you can use FCTN <;> to toggle your speed. You can erase portions of your picture with the "Box" feature, if you toggle with FCTN <.> first as well.

Thanks go to Tim O'Neill (and to Chris Faherty, author of TI Artist) for approval to share these materials!

Editor's note: The results may not be World Class, but are good enough for beginners and children who like to create. Feel free to add to, or change, the collection of pre-drawn eyes, noses, ears (especially the ears!). If you come up with anything worthwhile, PLEASE share it with us. O

Comic Show Instructions

edited by Ray Kazmer, USA

Original Documentation Translated by:
J. FREDERICKS and KEN GILLILAND

English Version of COMIC-SHOW EDITOR
and expanded DOCS by: RAY KAZMER

CREATING A COMIC SHOW WITH THE CONVERTED EDITOR

All "animation" must FIRST be drawn with TI-ARTIST. COMIC SHOW EDITOR only arranges TI-ARTIST pictures ("P") into runnable assembly code. There is no "easy way" to make animated COMIC-SHOWS, but your rewards can be great, if you stick with it. Using the EDITOR, IS a snap!

Other files in the COMIC-SHOW package are COLOR, COMIC-3D, PIC-COMP1, PIC-COMP2, PIC-SHOW1, and PIC-SHOW2. Descriptions of what these other programs are supposed to do, were NOT included in the original German documentation. Since the EDITOR seems to work well by itself, all the other files (above) are IGNORED in the following instructions but ARE included on this disk, in case you wish to experiment. NOTE: Only the EDITOR itself was translated (99%) into English. We will be discussing ONLY the EDITOR, named "COMIC/2" on this disk.

Creating animated COMIC-SHOWS from TI-ARTIST pictures is a "two-step" process. ARTIST pictures must FIRST be "compressed" (into a "squeezed format") which can be "previewed" with the EDITOR. (VERY helpful when "de-bugging" your creation.) When satisfied with the results, you can then "compile" your "squeezed" files into a "stand-alone" COMIC-SHOW, which can then be RUN from OUTSIDE the COMIC SHOW EDITOR environment.

WARNING: The TEX-COMP "LOAD" menus contain HIDDEN assembly routines. NEVER RESEQUENCE THESE LOAD PROGRAMS! NEVER EDIT LINES NUMBERED LESS THAN 100 or you will destroy those assembly routines! ALSO NOTE: The "LOAD" on side 1 is different than "LOAD" on side 2. You should make back-up copies of each side of this disk and use THOSE to experiment.

LOADING THE COMIC-SHOW EDITOR

The EDITOR uses option 3 "LOAD RUN" of the EDITOR ASSEMBLER module. A "simulated" E/A MODULE LOADER is supplied, on this disk. After "E/A" loads, select OPT. 3, "LOAD RUN" from the E/A menu. For "FILE NAME?" type: DSK1.COMIC/2 and <ENTER>. The EDITOR auto-starts.

USING THE COMIC SHOW EDITOR

SPECIAL NOTE: During MOST operations, pressing FCN/9 will abort ALL previous commands and return you to the MAIN MENU. This key-press is NOT shown on-screen at any time, but IS intermittently active.

MAIN MENU

- PRESS: 1 BUILD CONTROL FILE
2 VIEW CONTROL FILE
3 SAVE FINAL VERSION
4 DISK DRIVES IN USE?
5 END

After loading the EDITOR, your first consideration must be option 4. Press your "4" key to tell the EDITOR how many drives you are using, from 1 to 5 drives. The number "recycles" to 1 after 5.

1. BUILD CONTROL FILE (COMPRESS TI-ARTIST PICTURES) - THE 1ST STEP

This "reminder" appears on your screen: "_P FILENAMES". Below that, a prompt for the drive number, which will contain your "DATA DISK" (the disk containing your PREDRAWN SERIES of TI-ARTIST "_P" PICTURES.) You must now INPUT the SEQUENCE of your "_P" pictures in a LOOP. Example:

```
DSK1.GARF1_P   My "GARFIELD AND ODIE" was comprised
DSK1.GARF2_P   of 6 TI ARTIST pictures. Their
DSK1.GARF3_P   filenames were entered EXACTLY as I
DSK1.GARF4_P   have shown here.
DSK1.GARF5_P
DSK1.GARF6_P
DSK1.GARF5_P
DSK1.GARF4_P
DSK1.GARF3_P
DSK1.GARF2_P
DSK1.GARF1_P
```

NOTE: After entering the LAST PICTURE in a SERIES, you must build your CONTROL FILE to "LOOP" back, through the series, in order to get a good "fluid motion" in your COMIC SHOW. The last filename entered (bottom of the CONTROL FILE you are writing) MUST MATCH the first filename, at the TOP of this list. If it does not, the first picture will be shown only ONCE (when the COMIC SHOW first runs) and WILL BE SKIPPED OVER AFTERWARDS!

One more "DSK1." prompt will appear at the bottom of this list. To go on with the process, merely press <ENTER> with a "blank" for a file-name (press <ENTER> twice) and the EDITOR moves into it's next phase.

It should be noted that the "colour" portions ("C") of your TI-ARTIST files, are NOT used. You can also shorten ARTIST filenames before you do this process, to make things easier. This example would also work:

```
DSK1.G1
DSK1.G2
DSK1.G3, etc.
```

The EDITOR does not need the "_P" tag to compress ARTIST pictures, however, if you intend to retain your pictures, in ARTIST format, to RELOAD into TI-ARTIST for corrections (and there WILL be corrections!) then the "_P" MUST be retained.

N.B. You must use at least 2 TI-ARTIST pictures, but NO MORE than 100!

After writing the CONTROL FILE list and pressing <ENTER> twice you are prompted for a "SHOW-NAME." This is a TEMPORARY TITLE, into which all of the pictures will be "compressed" and stored for later use. I used a SHOW-NAME of: DSK1.TEST1 (Keep it simple and easily remembered.) After entering a SHOW-NAME, a prompt follows, asking: "PAUSE BETWEEN READ/Writes? (J/N)" (The "J" stands for "ja" the only "German" still in the EDITOR. Sorry, I could not find THAT with ADVANCED DIAGNOSTICS!) The "PAUSE" will probably be necessary (to swap disks) if you have 15 or more pictures in your CONTROL FILE and are using only 1 SSSD drive.

After this prompt is answered, the EDITOR starts making "compressed" files, starting with the SHOW-NAME you entered. Example: TEST1, TEST2, TEST3, etc. Original ARTIST pictures are not changed, only "read."

A screen display is shown, above each "TEST" filename being created, which gives you the file size (in BYTES) of each new "TEST" file.

A second screen display shows a running TOTAL (in BYTES) of the size of the SHOW-NAME file you are creating.

A third screen display shows "SECTORS FREE" on your "WRITE DISK." If using only 1 SSSD drive and this number hits zero, the WRITE-DISK is full and the EDITOR will

stop with an ERROR MESSAGE. But, if using a a DSSD or DSSD drive and this number hits zero, the EDITOR continues creating the files. This number is based on SSSD and is NOT correct when using any other drive types! (Do not worry about it.)

If an error occurs in any READ/WRITE, you are given three choices:

1-CORRECT (an incorrect filename. Works, sometimes.) 2-REPEAT (last function. Error usually results.) 3-IGNORE AND GO ON (locked up my console!)

The BEST BET is to go SLOW and ensure you hit NO errors. If an error does occur, the FCTN/9 key IS active at this point and you may save a lot of time and grief for yourself, by merely starting over.

2. VIEW CONTROL FILE

This function "previews" a COMIC SHOW, using your "TEST" files. When prompted, type in the SHOW-NAME you used. (Example: my "DSK1.TEST1".) "VIEW" will NOT work on uncompressed ARTIST pictures, a CONTROL FILE MUST be made first. When your "SHOW" RUNS, 2 keys will be active:

"+" = SLOW Picture Sequence, and

"-" = SPEED Picture Sequence.

This will be the first time you will see your COMIC SHOW "in motion." If corrections are to be made, you must make them on the ORIGINAL TI- ARTIST pictures, with TI-ARTIST, then re-process those pictures with BUILD CONTROL FILE again, before "testing" your changes with "VIEW."

3. SAVE FINAL VERSION (COMPILE STAND-ALONE PROGRAM) - THE 2ND STEP

After you attain whatever degree of perfection your patience allows, you are given two choices, with this final step:

PRESS 1- VIEW-FILE LOADER

2- EA/5 LOADER

Pressing "1" results in a further compacted series of "PROGRAM" files which can ONLY be loaded via the VIEW CONTROL FILE function of COMIC SHOW EDITOR. (There seems to be NO logical reason for THIS choice!)

"2" results in a genuine "stand-alone" COMIC SHOW which can be loaded via option 5 of the E/A MODULE or many other loaders usually found in User Group Libraries or on BB's. (I used THIS choice to save GGARF1.)

You will be prompted for the SHOW-NAME (my DSK1.TEST1.) Then, you are prompted for a "SAVE-NAME." THIS IS YOUR FINAL SAVE AND THE FIRST TWO LETTERS OF THE FILENAME YOU SELECT NOW, MUST BE THE SAME! Example: my SAVE-NAME for "GARFIELD AND ODIE" was: DSK1.GGARF1, seen as the FIRST file of my creation, if you decide to catalog this disk.

The DOUBLE FIRST LETTER is the "identifier" COMIC SHOW EDITOR builds into it's stand-alone programs, the same way TI-ARTIST uses it's "_P" to identify a legitimate picture. If you forget to enter the doubled letter, the EDITOR will double the SECOND letter in a SAVE-NAME. (My GGARF1 would have been called "AARF1".) Even if you DO forget, there is no big problem in re-naming all the files later. But REMEMBER: The FIRST LETTER MUST BE A DOUBLE LETTER, or your COMIC SHOW will not run!

CLOSING NOTE

"GARFIELD AND ODIE" was my VERY FIRST ATTEMPT at using the COMIC SHOW EDITOR. If you enjoyed it, or found these DOCS were a help, or if you were inspired continued on page 24

The Communicators

from Ross Mudie

How many BBS members are having problems getting on the BBS because their modem or terminal program will not work? or they have forgotten their user number or password?

At the next TISHUG meeting at Ryde Infants school on Saturday 7th September 1991, help will be available to work out your modem interfacing problems. Bring along your computer, modem and cables, together with your terminal software. Do not forget your modem book and your RS232 book. Members will attempt to resolve your communication problems and get you on to the BBS.

If you not at present a member of the BBS but wish to join you may also bring your equipment along to get it set up. The cost of BBS access for TISHUG members is just \$5 for the year (plus of course what it costs you in phone calls). BBS membership can be arranged at the meeting and a demonstration provided.

The TEXPAC BBS has been operational since 9am Sunday 1st July 1984. Since the BBS commenced operation it has received over 30300 calls. The first SYSOP for the TISHUG, TEXPAC BBS, was Shane Andersen. Back in those days the club membership was well over 1000. The BBS was very difficult to get on because it was always very busy and the time limit was just 15 minutes. The hours of operation were initially just a few hours in the evenings and the weekends. In the early days the system was open access, however a number of "nuisance callers" made it necessary to allocate user numbers and passwords which restricted access and overcame the "nuisance caller" problem. All members are urged to periodically change their password to prevent possible unauthorised access.

I took over the SYSOP job in in September 1986 and immediately commenced the task of BBS program improvement. The BBS was very un-reliable back in those days due to the interaction of users and the system; it was very easy to cause the BBS program to crash by wrong operation. The program is now considered to be fairly reliable, however, another obscure program bug was discovered and fixed early in July 1991. (As well there is always another bug!) My son Peter also helps as the assistant SYSOP when I am unavailable.

The BBS consists of a TI99/4A console (NTSC because it is only connected to a monochrome monitor), TI expansion system and Dataplex multi-speed modem. The TI expansion system contains a RS232 card with modified ROM, 128K card (using 32K only), an AT disk controller and a Triple Tech card which provides the clock and printer buffer. The system has 3 half height disk drives being used in double density and a logging printer which assists in diagnosing system failures. The console, printer and PE box are all 110 volt powered via step-down transformers. The 240 volt input to the system and the telephone line both connect via a FaxGuard to reduce the chance of failure due to lightning surges. The major cause of system failure these days is interruption to the mains power. The program uses TI BASIC with the Editor/Assembler module providing the necessary linking utilities for the linked assembly program. This is a "hardware reliable" implementation on the computer.

The BBS is loaded with a selection of up to 40 basic, extended basic and implanted assembly (inside extended basic) programs in addition to information files. The programs and files are changed each 2 months. Members can send mail and files to each other and place news or adds for ALL users to read.

Your BBS operates 24 hours per day, 7 days per week on 02 456 4606. The system has a nominal limit of one hour per call but nobody really complains if calls go a little over. The modem allows operation at 2400, 1200, 1200/75 and 300 bauds. The system is ready for all its members to call, any time.

Fast Extended BASIC

Rotating TIA Instances

by Lucie Dorias, Ottawa Canada

After doing last month's pic, I decided to add one more TI-Artist utility to my repertoire. INSTAROT rotates an instance on its left or right side, a function that TI-Artist Plus still lacks; its VECTOR function does some sort of a rotating, but the results are far from good. Genial Computerware's GRAPHICS EXPANDER does a great instance job, in any direction, but you might not have it (or, like me last month, just have forgotten you have it!). JOYPAINT's ROTATE function is excellent, but it a hassle to use it with TIA instance (convert it to a pic, convert with JOYPAINT PAL, load JOYPAINT proper to rotate, then do the reverse trip back to your original instance). Do you have JOYPAINT anyway? And its PAL? My Extended Basic utility does exactly the same thing, although it takes a little more time.

Still unknown to many users, TIA (and TIA+) can flip a picture, i.e. reverse it horizontally (mirror) or vertically (up-side down); this function is available in both the MOVE and COPY mode of the ENHANCEMENT function. To understand how it works, I included a diagram at the end of this text; the original instance HOUSE_I is at the top, framed by a broken line. The big dots at the corners of the frame indicate the starting position of the cursor when you press ENTER (or the fire button) to "frame" the portion of the picture to be moved or copied.

If you place the cursor at the upper left corner, you copy or move the picture as is; but if you position it at the upper right corner, the picture will be mirrored (horizontal flip). If you start at the lower left corner, the picture is flipped vertically; and if you start at the lower right corner, you picture will be flipped both vertically and horizontally. The bottom part of the diagram shows you what can be done with my INSTAROT program (new instances HOUSE/L_I and HOUSE/R_I, and what you can do with the rotated pictures by placing the cursor at different positions. You will see that if you need both left and right rotations, flipping one rotated picture twice (1 - upper right, 2 - lower left) will give you the second rotation, so no need to spend too much time with INSTAROT.

Instances are DV/80 text files so they can be used outside the TIA environment. Below (left column) is the content of the HOUSE_I file, the instance used in the diagram; "rlcl" means character in row one/column one, etc. The first line of data gives the number of columns and rows, and each line after it contains one character definition; HOUSE_I having 5 col x 3 rows (15 char), the file has 15 lines of definition. These are expressed in eight decimal numbers, equivalent to the hex numbers we would use in a CALL CHAR statement. The file is read row by row, from upper left corner (rlcl) to lower right corner (r3c5 in the present case). To see the house, write a program with the CALL CHAR values in the right column...

```
5,3
0,8,28,28,28,20,50,50
0,0,0,7,13,8,26,23
1,0,0,255,255,255,255,255,127
224,192,192,224,224,240,240,240,248
0,32,112,112,112,80,200,200
34,97,97,65,65,62,8,8
50,48,50,23,23,23,23,23,23
127,127,127,64,79,73,79,64
248,253,253,9,233,40,232,8
136,132,132,4,4,248,32,32
255,0,0,0,0,0,0,0
255,0,0,0,0,0,0,0
255,0,0,0,0,0,0,0
255,0,0,0,0,0,0,0
255,0,0,0,0,0,0,0
255,0,0,0,0,0,0,0
--- total COLUMNS, ROWS in instance ---
rlc1 > CALL CHAR(A,"00081C1C143232")
rlc2 > CALL CHAR(B,"00000070D081A17")
rlc3 > CALL CHAR(C,"010000FFFFF7F")
rlc4 > CALL CHAR(D,"E0C0C0E0E0F0F8")
rlc5 > CALL CHAR(E,"002070707050C8C8")
r2c1 > CALL CHAR(F,"22616141413E0808")
r2c2 > CALL CHAR(G,"32303217171717")
r2c3 > CALL CHAR(H,"7F7F7F4040494F40")
r2c4 > CALL CHAR(I,"F8F8F8D09E928E808")
r2c5 > CALL CHAR(J,"8884840404F82020")
r3c1 > CALL CHAR(K,"FF00000000000000")
r3c2 > CALL CHAR(L,"FF00000000000000")
r3c3 > CALL CHAR(M,"FF00000000000000")
r3c4 > CALL CHAR(N,"FF00000000000000")
r3c5 > CALL CHAR(O,"FF00000000000000")
```

To keep it as fast as possible, INSTAROT does not convert the decimal values to hexadecimal, so it cannot show the instances on the screen; but it works! (I used it to draw the diagram.) It takes about 4.5 seconds to

rotate one character; so, before proceeding, TEX will compute the estimated time needed to rotate the instance. But beware that, due to TEX's memory space, your instance has to have 250 characters or less (see DIM); if you work with a bigger instance, just save it as two smaller ones, and rotate them separately.

Half of the program is devoted to housekeeping: initialising in lines 120 - 160, screen interface in 180 - 250, get row/column, calculate estimated time, and ask you to proceed, lines 270 - 320. The sprites defined at the beginning are an example of rotations of a frog face (what else!); since they can be magnified, three frogs (left-rotated, normal and right-rotated) are defined in line 140. Everywhere in this program, to save memory space, I used A\$, S and K each time I needed a temporary variable. For the pre-scan, note where I put the GOTO 180 (end of line 150). It works!

The rest of the INIT is easy, so is the screen interface. Just note that the program will add "_I" to your instance name if you do not write it (line 210); then, after opening the input file (instance to rotate), Tex will suggest a name for the rotated instance name, and adding "/L_I" or "/R_I" according to your answer R\$ (lines 230 -240); you can keep the new name or change it before the output file is opened.

Line 270 reads the first line of the instance file; it is a string with a comma, so we must LINPUT it, not INPUT, and then read the segments left and right of the comma to get the C and R numerical VALUES (line 270). T is the total of characters to be rotated; if there are too many, Tex will tell you and GOTO 510 to ask you for another. Otherwise, it will compute the time needed to rotate the instance and ask you to proceed or not (lines 300-320). An interesting point: even if the S value can have decimals, being a multiple of 4.5 the image will display it as a round number.

To rotate an instance, two things must be rotated: the place of each character into the new row/column grid, since a row will become a column (and vice-versa), and then the content of each character definition, i.e. each line of input instance file. My solution was to use a memory array Z\$ to hold the new lines of numbers resulting from the rotation at the correct rotated position, then saving the whole array to disk in one operation (line 500). Because of the rotation, the first line of the new file will now contain the values of "R,C", since the starting column total is now the row total, and vice-versa. A DV/80 file is always a sequential file, because each record is VARIABLE; it would be nice to use a relative file to save each new line into its new position right after it is processed, but TIA could not read the file. Therefore, the need for a memory array, which explains the 250-char limit of INSTAROT.

In line 350, AP (new Array Position) and SAP (Start-of-AP for each row) are initialised according to the R\$ rotation wanted and to the total number of chars and of rows in the instance to be rotated. AP is used to save the content of each character into Z\$ after it is rotated (line 460), then the next AP/SAP values will be computed, again according to the rotation wanted (lines 470-480).

Each line of the instance (i.e. one chardef) is LINPUTed (commas!), processed and saved to memory by the FOR-NEXT Z portion of the program (lines 360-490). Lines 370-390 is an X loop to convert the string (eight decimal numbers separated by commas) into numerical VALUES and then put them into the V() array to be processed; it also doubles as a loop to zero the N() array: after the rotation routine proper, N() will be filled with eight new decimal values, one for each row of the rotated char, and another X loop will transform N() into a string with commas for the new instance file (lines 450-460).

Lines 410-440, that lie between the two X loops, are the heart of the character rotation routine. I have written a very detailed explanation of how the routine works, with nice graphics and table, but it takes two

full pages! So I will just summarize as best as I can what the routine does, and I will keep the complete explanation for an Appendix to this month's tutorial on Volume four of my Complete Works, coming out, hopefully in June.

To rotate one chardef, Tex takes each V() decimal value (For Y=1 to 8) and dissects it into the "powers of two" values it contains, from 2⁷=128 to 2⁰=1 (For X=7 to 1 STEP -1). If the power of two value is found, it means the pixel is on, and the corresponding pixel in the rotated character as to be set on to, but in a new position in the chardef grid. Tex places the "power of two" value corresponding to that pixel position in the corresponding variable in the N() array; the total of each N() variable will in the end give us the decimal value of the new row. Simple, is it not? No matter if it is still all strange to you, just type the program (I will have a copy on disk at the meeting for those who want one. Bob), trying not to make mistakes, and watch the results!

```

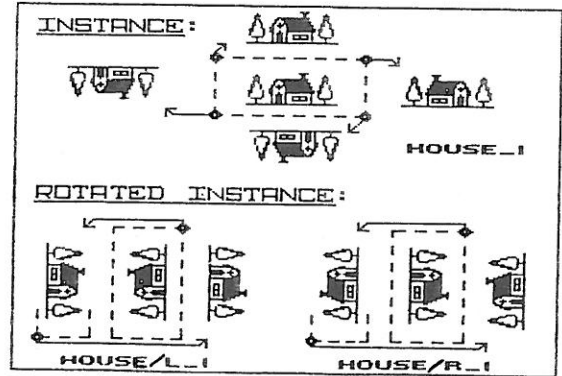
100 ! INSTAROT: Rotate a TIA Instance/ L. Dorais, OTTAWA
    UG, April, 1991
110 !
120 CALL CLEAR :: CALL COLOR(13,14,1):: CALL MAGNIFY(2)
130 DIM Z$(250):: F$="DSK1." :: L$=RPT$(CHR$(128),28)
140 CALL CHAR(123,"7C9EB77B7BB79E7C6699BDFDBE77E3C3E79
    EDDEED793E",128,"OOOOOFFFF")! FROGS, LINE
150 CALL SPRITE(#1,123,16,1,17,#2,124,14,1,121,#3,125,16
    ,1,225):: GOTO 180
160 A$,AP,C,K,N(),OP,P,R,R$,S,SAP,T,V(),X,Y,Z,ZZ :: CALL
    KEY :: !@P-
170 ! ----SCREEN DISPLAY/INPUT ---
180 DISPLAY AT(2,5):"<LEFT";TAB(19);"RIGHT >": :L$
190 DISPLAY AT(6,1):"INSTANCE TO ROTATE:" : "&SEG$(F$
    ,1,5): : "ROTATION WANTE D (L/R):"
200 DISPLAY AT(14,1):"ROTATED INSTANCE:" : " DSK": :L$
    :: A$=" I"
210 ACCEPT AT(8,5)SIZE(-12)BEEP:F$ :: IF POS(F$,A$,1)=0
    THEN F$=F$&A$
220 F$="DSK"&F$ :: OPEN #1:F$,INPUT :: F$=SEG$(F$,1,LEN
    (F$)-2)
230 ACCEPT AT(11,24)SIZE(1)VALIDATE("LR")BEEP:R$
240 DISPLAY AT(16,2):SEG$(F$,1,11)&"/"&R$&A$
250 ACCEPT AT(16,5)SIZE(-12):F$ :: F$="DSK"&F$ :: OPEN
    #2:F$,OUTPUT
260 ! --GET ROW/COLUMN, TELL TIME --
270 LINPUT #1:A$ :: P=POS(A$,"",1):: C=VAL(SEG$(A$,1,P
    -1)): R=VAL(SEG$(A$,P+1,2)): T=C*R
280 DISPLAY AT(20,1):R;"ROW X";C;"COL =";T;"CHAR": :L$
290 IF T>250 THEN DISPLAY AT(23,6)BEEP:"INSTANCE TOO
    BIG!" :: GOTO 510
300 S=4.5*T :: K=INT(S/60):: S=S-60*K ! TIME IN K MIN S
    SEC
310 DISPLAY AT(24,1):USING "EST TIME: ##M##S PROCEED? Y"
    :K,S
320 ACCEPT AT(24,28)SIZE(-1)BEEP:A$ :: IF A$="N" THEN
    510
330 ! -- ROTATE ONE LINE OF 8 CHARS --
340 DISPLAY AT(24,1):" NOW ROTATING CHAR:"
350 IF R$="L" THEN SAP,AP=T-R+1 ELSE SAP,AP=R
360 FOR Z=1 TO T :: DISPLAY AT(24,21):Z
370 LINPUT #1:A$ :: S=1 :: OP=0 :: ZZ=INT(Z/C)
380 FOR X=1 TO 8 :: N(X)=0 :: P=POS(A$,"",S):: IF P=0
    THEN P=LEN(A$)+1
390 V(X)=VAL(SEG$(A$,S,P-OP-1)): S=P+1 :: OP=P :: NEXT
    X ! GET DEC VALUES
400 ! -- ROTATE ONE CHARACTER--
410 FOR X=7 TO 0 STEP -1 :: FOR Y=1 TO 8 :: IF V(Y)<2 X
    THEN 440
420 IF R$="L" THEN N(X+1)=N(X+1)+2 (8-Y)ELSE N(8-X)=N(8
    -X)+2 (Y-1)
430 V(Y)=V(Y)-2 X
440 NEXT Y :: NEXT X :: A$=""
450 FOR X=1 TO 8 :: A$=A$&STR$(N(X)): IF X<8 THEN A$=
    A$&" "
460 NEXT X :: Z$(AP)=A$ ! BUILD DECIMAL STRING, PUT IN
    NEW ARRAY
470 IF R$="L" THEN AP=AP-R :: IF Z/C=ZZ THEN AP=SAP+ZZ
480 IF R$="R" THEN AP=AP+R :: IF Z/C=ZZ THEN AP,SAP=SAP
    -1
490 NEXT Z :: DISPLAY AT(24,1):"SAVING NEW INSTANCE TO
    DISK"

```

```

500 PRINT #2:STR$(R)&","&STR$(C):: FOR X=1 TO T :: PRINT
    #2:Z$(X):: NEXT X
510 CLOSE #1 :: CLOSE #2 ! END
520 DISPLAY AT(24,1)BEEP:"ROTATE ANOTHER INSTANCE?(Y/N)"
530 CALL KEY(O,K,S):: IF S=0 THEN 530 ELSE CALL CLEAR ::
    IF K<>78 THEN 180

```



continued from page 9

uses whole numbers, SIN and COS would only return the numbers -1, 0 or +1, which would not be of much use. In most programs sin and cos are usually multiplied by some other number anyway, so Gene set up the commands as: SIN A B, and COS C D, where A and C are the angles, B and D are the numbers multiplying sin and cos. B and D are also the variables to which the values are returned. For instance,

```
LET B=10 SIN 30 B
```

sets B equal to 5, since $\sin 30 = 0.5$, so $0.5 \cdot 10 = 5$.

Try this little program, it is as easy as ABC.

```

A=128 B=96
V=90 W=60 F=1 G=3
SCREEN 1
C=6
FGND C
FOR I=1 TO 5
A=A+1 C=C+1
FOR T=0 TO 360
LET M=V LET N=W
COS F*T M
SIN G*T N
SET A+M B+N
NEXT T
FGND C
NEXT I
STOP

```

Also try other values for the variables F and G and see what happens. In this program SIN and COS are immediately multiplied by the pixel locations on the screen, and so the results are always large whole numbers. In other situations where you may wish to use SIN and COS you may have to multiply them by a large number, say 100, do the calculation, and then divide by 100 to get the answer to the required accuracy.

You can use the same technique of using a variable in different functions to plot all sorts of patterns.

As an example of simple animation, try this little program.

```

REM MOVING HELLO
FOR I=8 TO 128 STEP 4
DISPLAY I 20 " HELLO"
REM NOTE THE SPACE
NEXT I
STOP

```

If anyone has any problems or suggestions (or even reads this column), I would be glad to hear from them.

TI-Bits Number 9

by Jim Swedlow, CA USA

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

THE TRITON TURBO XT CLONE A HANDS OFF FIRST PEEK

An XT clone that works with the 4A for under \$500? Too good to be true? Perhaps.

Let us take a look. This unit comes with 256K of RAM, one drive and a bridge box designed by Miller's Graphics (now MG) that allows you to use the 4A keyboard with the Turbo XT. The initial price is \$499. Your 4A will still need and use the PE box but both units can share the 4A's keyboard and your current monitor.

I called Triton and talked to one of their representatives. This unit looks like most XT clones. It has a 150 watt power supply (minimum is 135). The mother board is expandable to 640K of RAM (which means that you do not need to buy a board to expand memory). It has a legal BIOS.

The problem is the price to configure the unit to be competitive with most XT clones.

MONITOR: the unit is set up so that you can use your current CRT, be it a TV or a composite colour or monochrome monitor. However, you will soon be unhappy with this for IBM graphics. The resolution in text mode will limit you to 40 columns (TV's and colour monitors cannot display 80 columns). A monochrome monitor will do 80 columns but it will not do justice to graphics intensive programs like Flight Simulator. The bottom line is that an RGB monitor is a must to use the abilities of the system. COST: \$299

DISK DRIVES: the Triton unit comes with one drive. A one drive TI system is difficult. A one drive IBM system is worse. A second drive is required by some software and quite helpful for others. COST: \$99

MS DOS: Without it, your computer is useless. The 4A's DOS (Disk Operating System) is built in. The XT's is not. COST: \$69.95

MEMORY: the Triton comes with 256K which is now just barely adequate. An upgrade to 640K will allow you to run most software. COST: \$89

KEYBOARD: the 48 keys on your 4A pale when compared to a decent IBM clone keyboard. Trident sells one. COST: \$89

Now lets add all this up:

Turbo XT	\$ 499.00
DOS	69.95
640K chips	89.00
2nd Drive	99.00
RGB Monitor	299.00
Keyboard	89.00
Total	<u>\$1144.95</u>

While the add on costs are competitive with most clone dealers, the total is a bit high for a RGB colour system.

Before spending \$500 to \$1000 for a new computer, there are a number of things to consider.

First, do you really want a new machine? Does your 4A meet your needs? Are TI-Writer and Multiplan fast enough and big enough or are you constantly running out of memory? Do you have a need or desire for programs that you see in the stores but cannot buy for your TI?

If you are seriously considering upgrading, you must do your homework. There are many brands and endless options. The hard fact, however, is that a full system will cost \$1000 or more.

If you are leaning toward an IBM (brand name or clone), you will find "The IBM XT Clone Buyer's Guide", by Edwin Rutsch (Modular Information Systems, \$9.95) quite helpful. It should give you enough information to understand the options and to compare various machines.

When figuring the cost, include software. While not everything costs \$500, many items are in the \$50 to \$100 range. There is, however, a wide selection of public domain and fairware available. These range from excellent to junk, but the price is right.

Only you can decide what is best for you. The Triton Turbo XT clone may be what you are looking for or your old 4A just might do for a while longer.

Our computer will not last forever. It will, I suspect, become like 78 records - supported by only a few of the faithful. The bulk of current users will move on. The question really is not 'if' but 'when'.

CUSTOMIZING FUNNELWRITER - PART II

This month we will talk about modifying the User List. You will need a disk sector editor. DISK UTILITIES is recommended, but you can use any editor that will print sectors. This material applies to Version 3.3 of FUNNELWRITER.

If you load FUNNELWRITER from Extended Basic, there are two User Lists. The first one is the one you can access from the initial menu. This User List is embedded in the LOAD program. Customising it is fairly well documented. For Assembly Language programs, set K as 3 for Memory Image or Program files and 4 for Load and Run or Display Fixed 80 files. Do not resequence the LOAD program as it is an Assembly Language shell.

The other User List is accessed from the central FUNNELWRITER menu. If you keep pushing 5, number 2 will become User List. When you opt for that, FUNNELWRITER loads the User List defined in a file called UL. This option will allow FUNNELWRITER load up to 7 Assembly Language programs. You have to modify UL to include those programs you want. I currently have it load TURBO COPY, SPELL CHECK and DISK COPY (a broken track copier).

A word about disk space. My system has double sided, single density drives. This allows me to have a number of programs on the FUNNELWRITER disk and use it not only as a substitute for the TI Writer module but as a loader for various Assembly Language programs. Even with DSSD, however, I could not fit all of the SPELL CHECK files on the disk. I renamed UTIL1 to SPELL and put it on the FUNNELWRITER disk. After it loads, it asks you to push ENTER to load UTIL2. At that point I swap disks and run the Spell Check program.

Most, but not all, Assembly Language Programs will work within the FUNNELWRITER environment. You should make sure that the ones you opt for will function before modifying UL.

WARNING: If you are new to using a sector editor (or even if you are not) FIRST make a copy of your FUNNELWRITER disk. Do all of these modifications on the copy. Then, if something goes wrong, all you messed up was the copy. Keep the original working copy until you have fully tested the copy.

Now on to customising UL. First, print the file. It is two sectors long. Most of the first sector is Assembly Language code. At the end of the first sector you will see:

SCREEN ENTRIES-->>

Following that you will see the space for the first menu line at the bottom of the first sector and the remaining menu lines at the top of the second sector. The numbers 1 thru 8 (8 is BACK - to return you to the FUNNELWRITER menu) are already there. You need to use your sector editor to enter the program names and types. Here is an example:

3220 2054 5772 6279 2020 2020 2003
2 T ur bo

The first line is in HEX. HEX 20 is the character for SPACE. Note the 03 at the end of the line. This tells FUNNELWRITER that this is a Program File. Had it been Load and Run (DF80), the last HEX value would have been 04. It is easier to enter the program name in ASCII mode and the file type in HEX mode.

Next, you need to tell FUNNELWRITER the name of the file on the FUNNELWRITER disk. These are at the bottom on the second sector of UL. They follow this:

<<--FILE-NAMES-->>

The entry for TURBO COPY is:

0A44 534B 312E 5455 5242 4F20 2020
D SK 1. TU RB O

The 0A at the beginning is the length of the name in HEX (10 for those of you with ten fingers). The DSK1. is followed by 10 spaces (HEX 20). You can enter any file name you want. I did not show the remaining spaces above, but they are in UL.

QUOTE OF THE MONTH

"If debugging is the process of removing bugs then programming must be the process of putting them in."

COMPUTERISED COUPONS

My wife is, among many other things, a dedicated grocery coupon user and a limited computer user. I thought to bring these two together. Unfortunately, coupons do not lend themselves to computerisation. It would take more time than it was worth to input them. What was really needed, however, was a container to keep the little rascals sorted by type. We had tried (and abandoned as hopeless) commercial coupon holders, shoe boxes, 3x5 card boxes, envelopes and a host of other containers. The solution came from the computer world: disk boxes! We now have two 100's and a 50 - they work like a charm.

WORD OF THE MONTH

ELEGANT: An elegant program is one that is neatly and tightly written. It is artwork. Most good programs written for the 4A (and other small computers) must be elegant because we simply do not have the memory to be sloppy. When you are limited to 48K, you must conserve every byte. When you have 640K (which is becoming the PC standard), you do not.

ON CUSTOMER SUPPORT

Elegance also applies to retailers. I recently purchased a game from INFOCOM, a program from TEXAMENTS and some hardware from Pilgrims' Pride. I had problems with all three.

The INFOCOM disk was defective. Off in the mail it went and about two weeks later back came a good disk. No fuss, no muss and a reasonable turnaround time.

Pilgrims' Pride was out of one of the items I ordered and they send me a credit. For the wrong amount. Off in the mail went a protest and back came a nice response and a corrected credit.

The TEXAMENTS program did not operate correctly. I have just written to them so there has not been time for a response. I will tell you the rest of that story once I hear from them.

It is good to know that the folks we deal with stand behind their products. More important, by supporting the retailers who support the 4A we can help extend our orphan's life span.

Enjoy.

continued from page 25

SCR #61

```
0 ( CLINE INSTRUCTIONS EGR 10/88) FORGET IT : IT ; GRAPHICS2
1 : TYPTXT 61 BLOCK 15 5 DO DUP 64 I * + 60 I 3 - CLINE LOOP
2 61 BLOCK 64 15 * + 60 16 CLINE ;
3 TYPTXT KEY DROP TEXT 63 LOAD ;S
4
5 CLINE is the word in 64SUPPORT that does these tiny
6 characters. If 64SUPPORT is loaded, and the rules
7 for CLINE are followed, it can be used to print
8 text and labels on graphic displays. CLINE expects
9 on the stack, a memory address where the text is
10 located, the number of characters to print, and
11 the line number of the CRT to print it on, as will
12 demonstrated, the CLINE syntax is as follows.
13
14 Addr Cnt Line# CLINE
15 PRESS ANY KEY
```

SCR #62

```
0 \ CLINE GRAPHICS DEMO - DATA EGR 10/88
1 110 70
2 100 30
3 90 5
4 80 20
5 70 40
6 60 40
7 50 50
8 40 70
9 30 60
10 20 20
11 10 10
12 0 1
13
14
15
```

SCR #63

```
0 ( CLINE GRAPHING DEMO E RAGUSE 10/88 ) 33 CLOAD 2ROVER
1 CLS 8 12 AT ." PATIENCE, GRAPHICS LOADING"
2 : YAXIS 16 16 16 100 LINE ;
3 : XAXIS 16 100 240 100 LINE ;
4 : YSCALE 100 20 DO 15 I 17 I LINE 8 +LOOP ;
5 : XSCALE 240 16 DO I 99 I 101 LINE 20 +LOOP ;
6 : LABL 64 BLOCK 16 0 DO DUP 64 I * + 64 I CLINE LOOP ;
7 : AXES LABL YAXIS XAXIS YSCALE XSCALE ;
8 : SCALE 104 - ABS 8 10 */ 16 + SWAP 2 * 16 + SWAP ;
9 : POINT SCALE 2DUP DOT ;
10 : PLOT 2SWAP SCALE 2ROVER LINE ;
11 : MSG TEXT ." TO DO IT AGAIN ENTER <GRAF> ELSE DIR " CR ;
12 : GRAF SPLIT AXES 62 LOAD POINT
13 11 0 DO PLOT LOOP PAK MSG ;
14 CLS 7 12 AT ." FOR A CLINE GRAPHING DEMO " PAK GRAF
15
```

SCR 64

```
0 A DEMONSTRATION OF CLINE GRAPHICS
1 for printing tiny char on graphic projects
2 100
3 90
4 80
5 70
6 60
7 50
8 40
9 30
10 20
11 10
12 0
13 0 10 20 30 40 50 60 70 80 90 100 110
14 THE RISE AND FALL OF TI FORTH POPULARITY
15 PRESS ANY KEY
```

Treasurer's Report

by Geoff Trott

Income for July	\$1711.53
Payments in July	\$3462.18
Excess of expenses over income for July	\$1750.65

continued from page 26

To eliminate cell names, select the name command; put the name to be removed in the "define name" slot; Tab over to the "refer to" position and remove the reference. If, when you press <ENTER>, the "refer to" is blank, the name is no longer active.

XB tips Number 10

by Jim Swedlow, CA USA

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

IF THEN AGAIN

IF THEN is the most versatile command in extended basic. This month, another approach.

Often you will use a variable as a flag. You set it to one value at the beginning of a program and then change it if something happens. Then, using IF THEN, you can tell if that 'something' happened and branch your program's operation accordingly.

When doing this, remember two things: First, during prescan, all numeric variables are set to zero. Second, the IF THEN test considers zero as false and all other values as true.

The variable that you are using as a flag, then, does not need to be given an initial value. Further, any change to a non zero value would make it look like a true statement in an IF THEN test. If A is your flag you could do this:

```
IF A THEN 220 ELSE 300
```

If the 'something' had not happened then A would still be zero, the IF test would be false and control would pass to line 300. If the 'something' had happened, A would not be zero, the IF test would be true and execution would branch to line 220. This should help you simplify your IF THEN testing.

SPEED

You always hear how slow Basic and XB are when compared to assembly language. Alas, 'tis true. There are ways, however, to make your program run a bit faster. Consider this code:

```
10 FOR I=1 TO 200
20 B=3*I
30 IF B>999 THEN 50
40 NEXT I
50 END
```

When you RUN this, your 99/4A notes that there are two variables and assigns

memory space for each. Each time line 20 executes, the memory location for I is found and the value obtained and multiplied by three. Then the location of B is identified and this value stored there (the order of these steps may be a bit different but each step must happen).

Now look at this:

```
10 FOR I=1 TO 200
20 IF 3*I>999 THEN 40
30 NEXT I
40 END
```

Now there is only one variable and the number of steps is reduced. The result: running time drops from 3.80 seconds to 3.01 seconds and memory drops from 92 bytes to 69.

In this example, time dropped 21%. If there had been a PRINT command in the loop, running time would be 10 times higher and the time drop under 5%. The reason is that it takes a lot of time to print on your screen.

When would you want to make an intermediate value? When you will use it more than once.

DISK MENU PROGRAM - VERSION 2.0

Some time ago I promised an enhancement to the disk

menu program published in Extended Basic #5. Will, here it is. It will:

--Display up to 130 programs from your disk and run any program with a single key stroke.

--Disable FCTN QUIT (in line 160)

--Display or print a disk directory.

--Delete any unprotected file from your disk.

--Support systems with one to four drives.

This program requires memory expansion and at least one disk drive. Save it on any disk that will be used with XB under the name LOAD (so it will auto boot). If you do not want to key it in, it is in our library.

There a few modifications you may want to make to match this program to your system:

--If your printer's name is not PIO, substitute the correct name in line 330.

--If you do not have a printer, insert the following code into line 330 between the line number and the existing code:

```
330 GOTO 270 !
```

--If you have one drive, change line 380 to read:

```
380 CALL SCREEN(8) :: A=@
```

--If you have one, three or four drives, modify the VALIDATE statement in line 390 to match your system.

--If you want to put a final fail safe before deleting a file, insert the following code:

```
362 DISPLAY AT(21,@):"Press <Y> to confirm":"deleting ";
A$(A) : :
364 CALL KEY(3,C,B) :: IF B<@ THEN 364 ELSE IF C<>89
THEN 270
```

Just as I put the finishing touches on this column, I get the June ROM and find that Ramon has published his XLOADER, which does the same thing as the above DISK MENU program. I talked to Ramon and decided to publish my program because it is different. We did the same thing in different ways. I like his paging better than mine but I like my routine for deleting a file better. He gets up to 32 program on a screen while I get only 24, but Anyway, the point is that two different folks doing the same thing do it differently. So compare the two and learn (I see some things I will 'borrow' for Version 2.1)!

P. S. The real reason I published mine is that I got Ramon's the day before my deadline and I did not feel like coming up with something else! Oh well.

Enjoy!

o

continued from page 6

```
320 T$=T$&"C;Y6;X2"&";ESUM(R[-5]C:R[-2]C)"&R$
330 !
340 FOR COL=1 TO 2
350 T$=T$&"C;X"&STR$(COL)
360 FOR ROW=1 TO A-1
370 IF LEN(T$)>128 THEN CALL WRITE(T$,T1$):: T$=T1$
380 T$=T$&"C"
390 T$=T$&"Y"&STR$(ROW)
400 T$=T$&"K"&STR$(AMT(ROW,COL))&R$
410 NEXT ROW
420 NEXT COL
430 !
440 T$=T$&"C;Y"&STR$(ROW)&"K"&CHR$(34)&"&CHR$(34)&R$
```

continued on page 27

Multiplan Exercises #7

by Herbert Schlesinger, USA

You are not limited to formulas in cell protection. You can lock all cells which are not to be changed. This can be done by locking each cell individually as above, before we made a range out of the command, OR we may lock the entire sheet and then release the "input cells" individually or by ranges. Look here:

1. Move the pointer to "Home" (R1C1).
2. Select Lock; select Cells.
3. Type in the colon (:) and press the Lower Right key (FCTN 1)
4. Tab over to the status option.
5. Select "Locked" and press <ENTER>.

In the sheet we are working on we must unlock the input cells:

1. Place the pointer in R5C2 (the first increase rate)
2. Select the Lock and Cells options from the menus.
3. Define the range you want to unlock as R5C2:R8C3.
4. Press the tab key (CTRL A or CTRL 2)
5. Move the parentheses to Unlock by pressing the space bar or the letter "U" and then press <ENTER>.

You can now only change the cells which should be able to be changed.

There is another key which we have not discussed. CTRL 3 is the "Next Unlocked Cell" key. Its purpose is to make it easier to move around a spreadsheet which has many locked cells. This allows you to go from one unlocked cell to another, without bothering to "try" cells which are locked.

SORTING:

Bring up the file "Sorting".

You can sort the sheet anyway you wish by using the SORT command:

1. Place the pointer on the first name in the column to be sorted. (R3C1)
2. Select the Sort option from the menu, and you get:

SORT by column: 1 between rows: 2 and: 255 order:(><

3. Tab over to "between rows".
4. Type in 3 so that the label and the underline will not be included in the sort.
5. Press <ENTER>.

The > shows that the sorting is in ascending order. If we had wanted to go from Z to A, or from a lower to a higher number, we would have selected (<) in the command.

To re-sort by floors, for example, move the pointer to the top of the floor column, again beneath the title and underline, (R3C3). Select the Sort command which will then show:

SORT by column: 3 between rows: 2 and: 255 order:(><

Tab over and change "between rows" to 3 and then <ENTER>.

SORTING WITHIN A SORT:

Notice that the first names of the Smiths are not in alphabetical order. To correct this, sort by the "First names" and then again by the last names. Once a list is sorted, re-sorting by another criteria will not change the previous sort's order.

These sorting capabilities make Multiplan a good medium for mailing list management. Last names in C1, first names in C2, addresses in C4, etc will allow sorting by name, by city, by zips for bulk mailing. Adding names is no problem - add them at the end and re-sort.

USING NAMES FOR CELLS AND RANGES:

In the file "NAMES1" we had a formula:

$$R1C2 * R2C2 / (1 - (1 + R2C2)^{-R3C2})$$

Using named cells rather than cell references, the formula becomes:

$$\text{Principal} * \text{Interest} / (1 - (1 + \text{Interest})^{-\text{Term}})$$

A formula in words such as this is much easier to understand and there is less likelihood of entering a wrong cell reference.

NAMING A SINGLE CELL:

Naming a single cell; here the "principal" cell:

1. Pointer to R1C1 (where the word principal is located).
2. Select the Name command :

NAME: define name: Principal to refer to:R1C1

3. To use the name "Principal" (the default since the pointer is in that cell), press the TAB key (CTRL 2) and the pointer moves to the "refer to" option.

4. Press the right arrow key to move the pointer to R1C2 and the submenu changes:

NAME: define name: Principal refer to:R1C2

Now press <ENTER>, and the cell in R1C2 has the name "Principal".

Place the pointer on the word "Interest" (in cell R2C1) and repeat the above for the cell R2C2. And again for "Term" (in cell R3C1) so that R3C2 has the name "Term". Note that while the cells in Column 1 have the "Labels", the cells in Column 2 have the "Names".

Now put some figures in column 2. The cell named "Principal" should have 1000 entered into it. Place 0.1 in the cell named "Interest" and 12 in the cell named "Term"(R3C2). Thus to calculate the payment on the loan place the pointer in cell (R5C2) which is labeled payment and press the = key. Enter this formula:

$$\text{Principal} * \text{Interest} / (1 - (1 + \text{Interest})^{-\text{Term}})$$

Press <ENTER> and the result, the payment, appears in R5C2. As always on a properly constructed spreadsheet, change a figure in one cell and the others respond with a new result.

GROUPS OF CELLS:

Load the file "CHECKING".

A name can refer to a single cell, to a row, a column or even a range (group of cells). Let us set up a sheet for the keeping of a bank balance:

1. The "Beginning Balance" is typed into R1C2. Format the cell wider if it is necessary.
2. Deposits are entered into column 1 below the label."Deposits"
3. Checks are entered into column 2 below the "Check" label.
4. Ending Balance is shown at R1C4 beside its label.

The Formula we are going to use is this one:

$$\text{Beg_Bal} + \text{SUM}(\text{Deposits}) - \text{SUM}(\text{Checks})$$

continued on page 26

Multiplan Machinations

by Bill Harms, USA

EDITOR'S NOTE: This is an article, the first part of which appeared in the December, 1988 issue of the TND. The second part was never printed so I have decided to print the whole article together for those of you who use Multiplan and want to have a go at this aspect of its capabilities.

In this article I will introduce you to a method to transfer data from a basic program to Multiplan.

I use Multiplan to keep my budget and to estimate my income taxes. I have a spreadsheet with 18 columns: 12 months, Yearly Total, Year-To-Date, Weekly Average, Monthly Average and two for Taxes. Those last two have formulas to get various numbers from the spreadsheet. The rows include: Pay, Interest, Expenses, Loans and Other. You can really do "What If'ing" and "Why Not'ing" with Multiplan.

It was a bear, however, adding up all the darn monthly checkbook entries in separate categories (Meals Out, My Pay, My Wife's Pay, Groceries, Interest, etc).

Since Multiplan does not have a FOR/NEXT capability and cannot scan data until some criteria is met it was too time consuming to enter the 100 to 150 monthly entries directly from my checkbook into Multiplan (if there was enough RAM).

Now I use a nice little fast (I mean fast) Extended Basic program I wrote to get those Transactions added by category. Then I can use a SYLK creator to quickly and correctly prepare them for loading into my Multiplan spreadsheet.

SYLK (or Symbolic Link) files are a little known feature of Multiplan. They can be written to disk by a basic program and read by Multiplan.

In this article I will show you how this is done. This material is based on a program I got from TI, a series of articles in the May (and later) 1985 SUPER 99 MONTHLY (now called THE SMART PROGRAMMER) and the Multiplan manual.

This bare bones program is based on the one I received from TI in 1984. The disclaimer was bigger than the program! It writes a disk file with a one cell spreadsheet that can be read by Multiplan.

```
100 OPEN #1:"DSK1.SYLK",DISPLAY,
    OUTPUT,FIXED 128
110 CALL CLEAR
120 INPUT "ROW NUMBER: ";R$
130 INPUT "COLUMN NUMBER: ";C$
140 INPUT "CELL CONTENT: ";A$
150 FOR Q=1 TO 27-LEN(A$)
160 W$=W$&CHR$(0)
170 NEXT Q
180 X$=CHR$(34)&A$&CHR$(34) !
    Surrounds contents with quotes
190 Z$=CHR$(13)&CHR$(10) !
    Carriage Return and Line Feed
200 Y$="D;PMP"&Z$&"F;DGGB"&
    Z$&"B;Y"&R$&"X"&C$&Z$&
    "C;K"&X$&Z$&"W;N1;A1 1"&
    Z$&"E"&Z$&W$
```

This monster of a line has the symbolics needed for Multiplan to read the file. See page 205 of the Multiplan manual for explanations.

```
210 PRINT #1:Y$
220 CLOSE #1
230 END
```

If you enter and run this program, you will find a file on your disk called 'SYLK'. Before you can load this file, you must change it. It may seem a bit odd,

but the file must be written as DISPLAY, FIXED 128 and then changed to INTERNAL, FIXED 128 in the file header. In other words, the file must use DISPLAY notation but must look like an INTERNAL file! There are two ways to do this. You will find an application of Barry Traver's RAW (Read And Write) in SUPER 99 MONTHLY. Or you can use ADVANCED DIAGNOSTICS to change the last four hex characters of the first line of the file header to <0202>.

Once you have done this you can load your file. First, boot Multiplan. Press <T>ransfer and then <O>ptions. Next press <S>ymbolic and then <ENTER>. Now press <T>ransfer again and this time <L>oad your file.

Here is the Multiplan spreadsheet:

1
1 HARMS

This is what the data looks like on disk using Millers Graphics great ADVANCED DIAGNOSTICS:

```
Drive : 2   Track : 3
Side  : 1   Sector : 34
Byte  : 0   Display: Ascii
```

```
I D ; P M P * * F ; D G O G
B * * B ; Y 1 ; X 1 * * C ;
K " H A R M S " * * W ; N 1
; A 1 1 * * E * * * * *
```

Most of the **'s stand for CR/LF (Z\$ in line 190).

There are many ways you could input data besides the simple INPUT in line 140. You could read data in from DATA statements or from a disk file. That disk file could be created by most anything: TI Writer, RS232, another module or a Multiplan Print File.

You can create data in Basic and then "dump" it into a spreadsheet en masse' instead of just keyboarding it.

You could transmit the outputted SYLK file of your Multiplan spreadsheet to others via RS232. The DIF (Data interchange Format) used by Lotus 1-2-3 and Visicalc only accommodates the cell content, not the sheet parameters.

This is only a taste (BAD?) of what you can create to load data into Multiplan. It really opens Multiplan up to other software.

Now let us do a 2 row by 2 column spreadsheet.

The most significant change in this program is that it can create a file with many cells — a couple of columns of data.

After using this program, you could load the file into Multiplan and save it as a Normal file. Then you could eXternal Copy it or parts of it into another spreadsheet that has formulas and other data. You might load a month of information into a sheet that has many months of data plus formulas for calculating Year-to-Date, Average, Units per Time Period, etc.

As I mentioned above, the Basic program must write a Display Fixed 128 file. You must change the File Descriptor Record (FDR) to make it look like the file is Internal Fixed 128. Otherwise Multiplan cannot read it. The only trick is to find the FDR!

This is a very sad deal, but it is very easy to change it with a good sector editor such as Miller Graphic's ADVANCED DIAGNOSTICS. Change byte 12 of the FDR for your SYLK file from Hex 0002 to Hex 0202. The FDR's are in Sectors 2 thru 32.

Another way is to use Barry Traver's fine program RAW (Read And Write) which is written in Assembly

language. It will make the change right out of your Extended Basic program.

I said that the maximum width of a Multiplan cell (to view) is 27 characters. WRONG! It is 32, but the cell can hold 127 characters and you can format cells to be "Continuous". This allows TEXT to display/print right on over as far as it goes.

Each SYLK record (remember - files are made up of records) can include cell content (text, numeric value or formula), row and/or column numbers and many other symbols for you to describe windows, sheet bounds, formatting of sheet and individual cells, sheet links and more. In fact, you can include just about everything except Multiplan commands such as COPY and DELETE.

Your cell content data can be split into more than one record. You just keep on creating 128 character records of SYLK symbols and data until you run out of information that you want put into the SYLK file. Then you fill the last record in the SYLK file with nulls <CHR\$(0)> so that it is 128 characters long.

The first program had 27 nulls which made it a 73 byte record. I could not get it to load using a larger number of nulls (like 123). R. Mitchell's program as published in the May, June and August, 1985 SUPER 99 MONTHLY had the key.

You continue building a string (numerics are included via STR\$(xx)), using the & to concatenate each new symbol or data item onto the string as you go UNTIL the LENgth of the string exceeds 128. Then you print the first 128 characters of the string into your disk SYLK file. Right after printing you move any characters in your string beyond 128 to a temporary variable.

Then you move the remaining characters back into your main string and continue building until it has more than 128 characters again and you do another print to disk of another SYLK record.

There are two types of SYLK symbols: Record Type Descriptors (RTD) of 1 or 2 characters and Field Type Descriptors (FTD) which are preceded by a ";". These need to be surrounded by double quote marks (not mentioned in the Multiplan manual). CR/LF's are used to separate SYLK record types (careful - a Record Type or RTD is a SYLK symbol while a record is part of a file - records contain record types).

Lets look at a sample: "C;K". This is a RTD C which means that it is a data point. The FTD ;K means that the value of the data point follows. Once you set a row or column number with the RTD C and FTD of say ;Y (for row), all the remaining data points (;K or ;E) are put at that row. You only need give another "C;X" or "C;Y" when changing the row (;Y) or column (;X) of the data point.

One of the fastest ways to learn the correct formatting of SYLK records is to enter some stuff into Multiplan and save the sheet in Symbolic format. Then you can look at the file with a sector editor like ADVANCED DIAGNOSTICS.

The following program creates a SYLK file of 2 columns with 4 rows of values and a formula for the SUM of the 2nd column. The 2nd column is NAMEd per your choice.

NAMEing is a Multiplan technique. It is quite helpful since a name of a cell or a range of cells can be used in a formula (to wit) SALES-COST (SALES minus COST). You would have NAMEd some cells SALES and some others COST, so the cell with this formula yields the profit. It could be named PROFIT.

A normal formula might look like R4C6-R9C6. For a bunch of stuff you might use SUM(R1:40C3)+SUM(R90C4:5). Relative references (relative to the cell that has the formula) as in my program look like SUM(R[-5]C:R[-2]C).

The whole sheet looks like this when loaded into Multiplan:

column >> 1	2
1 18	15
2 1	75
r 3 33	199
o 4 400	77
w 5	-----
6	366

The value in R6C2 (366) is from the formula SUM(R[-5]C:R[-2]C).

Using Craig Miller's ADVANCED DIAGNOSTICS we can see the whole sheet!

```

I D ; P M P * * F ; W 1 2
6 * * F ; D G 2 G 8 * * B
; Y 7 ; X 3 * * N N ; N W H
H ; E R 1 : 6 C 2 * * C ; Y
6 ; X 2 ; E S U M ( R [ - 5
] C : R [ - 2 ] C ) * * C ;
X 1 C ; Y 1 ; K 1 8 * * C ;
Y 2 ; K 1 * * C ; Y 3 ; K 3
3 * * C ; Y 4 ; K 4 0 0 * *
C ; X 2 C ; Y 1 ; K 1 5 * *
C ; Y 2 ; K 7 5 * * C ; Y 3
; K 1 9 9 * * C ; Y 4 ; K 7
7 * * C ; Y 5 ; K " - - -
- - " * * W ; N 1 ; A 1 1
* * E * * * * * * * * * *

```

The 128 Character SYLK record ends after the fourth character in the 15th line. This is also the end of the SYLK file.

R. Mitchell's program reads a Display Fixed 80 file and writes the data into a SYLK file that Multiplan can read. It has one word processor line on each row. You could change it to read Display Variable 80 or any other type of file.

Multiplan has a 255 row and 64 column limit. I have found that about 23 columns and 40 rows of formulas, data and labels (text) is about all that the 4A's RAM can hold.

The program I wrote, which is based on R. Mitchell's, follows. These comments may help you understand it.

Line 300 adds sheet identification, windows, format and bounds.

Line 310 adds the name you inputted in line 280 and the range for that name (R1:6C2).

Line 320 adds the formula for R6C2.

Line 400 adds the cell content as numerics (here taken from the array AMT).

Lines 370, 450 and 470 work together to ensure that each record is 128 characters long.

Lines 510 thru 540 write the SYLK records to disk.

Now you can load your data into Multiplan to suit your needs. Instead of DATA statements you can use a file INPUT statement instead of the READ statement in line 210.

```

100 ! SYLK * 11/85
110 DISPLAY AT(1,5)ERASE ALL:"A SMALL
SPREADSHEET"
120 ! data input section *****
130 OPTION BASE 1
140 DIM AMT(4,2)! array for input data
*****
150 DATA 18,15
160 DATA 1,75
170 DATA 33,199

```

continued on page 6

TI-Base Tutorial #13

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.

```

                                MLIST/C
CLOSE ALL
PRINTER NONE
SET PRINTER=DSK2.MLIST
LOCAL TST C 2
LOCAL ANS N 2 0
LOCAL MV C 50
LOCAL AST C 2
REPLACE AST WITH "*"
USE DSK2.TNAMES
SET HEADING OFF
SET RECNUM OFF
*
WHILE .NOT. (EOF)
  REPLACE TST WITH SA
  IF TST <> "No"
    SET TALK OFF
    CLEAR
    WRITE 2,4, FN
    WRITE 4,4, LN
    WRITE 6,4, SA
    WRITE 8,4, CT
    WRITE 10,4, ST
    WRITE 12,4, ZP
    WRITE 22,2, "1=Mr. 2=Mrs.;"
    " 3=Miss. 4=Ms."
    WRITE 23,2, "ENTER SELECTION =>"
    READ 23,2, ANS
    SET TALK ON

```

After all this time I still have not covered all the tricky little tasks which TIB will automatically do for you. This month I will cover a couple more items I think are very interesting. What you see above is one CF. The first line contains MLIST and the last line contains Formatter. It is too big to type in through the TIB editor unless you break it down into smaller CFs and use DO xxxxx to combine their operation. I used FunnelWeb. If you have no need for the function it performs, you may consider it a waste of time, but if you need this type of utility it could be a real lifesaver.

```

DOCASE
CASE ANS = 1
  REPLACE MV WITH "1 Mr."
  PRINT MV
  BREAK
CASE ANS = 2
  REPLACE MV WITH "1 Mrs."
  PRINT MV
  BREAK
CASE ANS = 3
  REPLACE MV WITH "1 Miss."
  PRINT MV
  BREAK
CASE ANS = 4
  REPLACE MV WITH "1 Ms."
  PRINT MV
  BREAK
ENDCASE
IF (ANS<1) .OR. (ANS>4)
  REPLACE MV WITH "1 "
  PRINT MV
ENDIF
REPLACE MV WITH "2 " | FN
PRINT MV
REPLACE MV WITH "3 " | LN
PRINT MV

```

```

REPLACE MV WITH "4 " | SA
PRINT MV
REPLACE MV WITH "5|" " |TRIM(CT);
|", "|ST|" "|ZP
PRINT MV
PRINT AST
ENDIF
MOVE
ENDWHILE
*
SET HEADING ON
SET RECNUM ON
CLOSE ALL
PRINTER EPSON
SET PRINTER=PIO.CR.LF
RETURN Copyright Martin A. Smoley 1989
*
* CF to produce a TI Writer Mail list
* to be usable with TIW or FunnelWeb
* Formatter. Includes user screen input

```

Here is the task. I have a bunch of names and addresses in my TIB Database and I would like to send a bunch of those people one of those neat form letters, the kind that has individual addresses at the top and maybe the persons first name scattered around in the text so it looks like I wrote each letter. They live far enough apart that they probably will not compare letters and find me out. I know that FunnelWeb or TIW will write the letters for me, but I need to take the correct information out of the DataBase and put it into a file that they can work with.

```

Listing of TNAMES. 10/18/89
REC NM LN FN MI SA
CT
ST ZP PH XP GP ID
0002 1 Aardvark Grant E. 9995 State Rt. 84
Geneva OH
44014 1-465-9876 88/02 NOCO 0717851
0003 2 Aardvark Willard J. No Newsletter
OH 1-465-7689 88/09
NOCO 0717852
0005 3 Jones Jan W. 37285 Burgandy Lane
Mentor-on-the-Lake OH
44060 257-1029 88/08 NOCO 0820871
0000 4 Smoley Martin A. 6149 Bryson Drive
Mentor OH
44060 216-257-1661 89/02 NOCO 0713831
0004 5 Vivannovitch Aulga I. 111 E. 98th. St.
Cleveland OH
91023 541-5415 88/05 NOCO 0712881
0001 6 Whitman Raymond (Slim) B. 2574 East 254th.
Eastlake OH
44094 951-2345 88/09 NOCO 0921861

```

** MLIST **

```

1 Mr.
2 Grant
3 Aardvark
4 9995 State Rt. 84
5 Geneva, OH 44014
*
1 Ms.
2 Jan
3 Jones
4 37285 Burgandy Lane
5 Mentor-on-the-Lake, OH 44060
*
1 Mr.
2 Martin

```

3 Smoley
 4 6149 Bryson Drive
 5 Mentor, OH 44060

```
*****
.FI;AD;LM 0;RM 55;PL 20
*1* *2* *3*          TEST1
*4*
*5*
```

Dear *1* *3*

This is a test *2*, to show the use of FunnelWebs mailmerge. I know your address is *4* *5*, and I will be contacting you again. Until then *2*, please try to find some new and interesting uses for the TI 99/4A.

```
Mr. Grant Aardvark   FINISHED LETTER
9995 State Rt. 84
Geneva, OH 44014
```

Dear Mr. Aardvark

This is a test Grant, to show the use of FunnelWebs mailmerge. I know your address is 9995 State Rt. 84 Geneva, OH 44014, and I will be contacting you again. Until then Grant, please try to find some new and interesting uses for the TI 99/4A.

I have listed TNames at the bottom of 13.1.1 to show you the DataBase (DB) I am using. It is the same DB we normally use except for Jan Jones and Aulga Vivannovitch who have been created to represent female possibilities in your DB and Willard Aardvark who does not have a street address listed. These represent only a couple of the problems I am sure you will encounter in a large DB, but the techniques in handling these problems can be used to handle almost any problem with a little ingenuity. I will go through the MLIST/CF and try to explain what I am doing. The command PRINTER NONE does not exist. "So what's up?" We are going to print selected parts of the DB to a Disk file named MLIST as you see in the next line of the CF. We need a finished file in the form you see to the left of this page. FunnelWebs Mailmerge can handle blank spaces and even blank lines, as you see, but it cannot handle CR or LF symbols at all. When TIB prints to your printer, or to a disk, it automatically sends CRs and LFs. There is no way to stop it. "Oh Yeah!" I loaded TIB and type USE DSK1.PRINTER and then EDIT. I paged through the file until I found the name of a printer I thought I would never own. I changed all the data in all the fields to zeros except for the CR and LF fields, which I changed to 20. 20 is the Hex value for a space. Then I changed the name field to NONE and pressed FCTN 8 to record my new data. Do not be surprised if the record seems to disappear at that second. TIB will instantly resort the file and stick NONE in where it belongs. If you need reassurance, page back and forth through the file and you should find NONE. Then press FCTN 9 to get out and enter CLOSE to reclose that DB. Now when our CF sends the command PRINTER NONE, TIB will load none commands and a couple space characters into its control code area. At the end of the CF I reload my EPSON codes and SET my printer back to PIO.CR.LF.

The next few lines should be quite common by now. SET heading and recnum off is needed or you will find more junk in the disk file which FunnelWeb cannot handle. The While loop will move through the TNames DB one record at a time until it finds the EOF. Next I show you only one of the many ways that we can handle Willard J. Aardvark. You may have a DB which contains a varied group of people or whatever. In that group you may have a percentage of records that you use for different reasons and therefore wish to exclude records from mailing to mailing. You might have a field that contains, No Newsletter, or Not a member, or Not married, or Not a good sales lead, or No Lamb this year. Without going into an explanation of why you might be

keeping these weird records on people or sheep, I will give you an idea on how to test for distinguishable characteristics. If you place any of the previous phrases into a 2 space field, TIB will take only the first two places from left to right and throw the rest away. So, in all of these cases you would end up with "No". If you took the first three places, you would end up with some answers of "No" and some of "Not". You can then test for the difference between the answers. This technique will allow you a greater flexibility in distinguishing between certain types of data. In the case of TNames, if "No" is found, no action is taken and that record is passed over.

If "No" is not found, everything between IF and ENDIF is performed. Briefly, if "No" is not found, the First Name, Last Name, Street Address, etc. of the current record are placed on the screen. You are then asked to determine if this person is Mr., Mrs., Miss or Ms. Entering a 1,2,3 or 4 will cause TIB to place your selection in the MLIST file as item 1. If you enter a number above or below the range, TIB enters a circumflex after one in the MLIST file. If a 1 is called from TEST1, as you see, there must be a corresponding answer in the MLIST file. When this task is finished, TIB also places the FN, LN, SA and CT, ST, ZP fields into MLIST. TIB continues this process for each record until it hits the EOF marker. It then sets the system back to its original state and RETURNS to the Dot Prompt. For those of you who do not use mailmerge here is a quick run through. You create a letter like TEST1. The numbers between asterisks tell the Formatter that there is a word or phrase in a Mail List file which it should substitute at that point. If everything is correct, in the file, you will receive a letter with all the insertions in the right places. Note: FINISHED LETTER. When you enter the Formatter use DSK1.TEST1 and PIO.LF as usual. For Mailing List enter Yes, and for Pages enter All or "1", "1,2", "1,2,3" or "1,2,3,4" for particular names on the list. Copies: 1, Pause: N, and DSK1.MLIST for Mailing list. At that point you are on your way.

For Sale

The following items are surplus to my requirements and are available either individually or as a complete system for just \$500.

TI99/4A Beige console	50
TI Peripheral Expansion Box	150
TI Disk Controller Card	50
TI RS232 Serial/Parallel	75
PEB 1200 baud Modem Card	75
Corcomp Triple Tech Card	75
2 Mitsubishi Floppy Drives	150
TI Speech Synthesizer	30
6 Games and Education Modules	30
Navarone Module Widget	20
TI Dual Joysticks	10
Shugart Floppy Drive	10
	<u>725</u>

Notes:

- Console includes 32K memory and CPU reset switch.
 - PEB is in as new condition and includes external isolation transformer and silent fan.
 - Floppy drives are high quality units with head load solenoids.
 - Triple Tech card includes 64Kb printer buffer, two way printer switch box, real time clock and provision for speech synthesiser.
 - Modem card supports 1200 baud.
 - Modules include Parsec, Munchman, Car Wars, Dragon Mix, Alligator Mix and Reading Fun.
- Please contact Lou Amadio on 042-284906.

continued from page 14

enough, by my "translation" of the COMIC SHOW EDITOR to give it a shot, I certainly WOULD NOT mind finding a couple of dollars in my mailbox! (Yes, \$2.00, IN CASH would be OK!) Thank you, A BUNCH!

Beginning Forth - part 9

by Earl Raguse, UGOC, CA USA

BIT MAP GRAPHICS WITH TEXT

In the last lesson #8, I said that we would find a way to have text in the bit map graphics mode. If you have the 64SUPPORT Editor loaded, this time we will do that. Since, I think that this is such an important feature, I will explain in detail. In the future, I will go on to do a type of graphics which might be improved by using transcendental functions. Since resident Forth does not have this capability, we must use an extension known as Floating Point Arithmetic, available on the TI Forth System Disk, and which will be the a future lesson subject.

Screens #61 thru 64 are a simple example of the complex things that one could do with Forth. The heart of it all is on Screen #63. Screen #64 is the data for CLINE to label our graph. Screen #62 is the arbitrary data that is plotted on the graph, and Screen #61 is an explanation of the rules for using the word CLINE, which in turn uses itself to display the text on the CRT in the GRAPHICS2 mode.

The word CLINE uses the word SMASH which is defined in Assembly code on the TI System Disk Screen #65, COMPACT LIST. Screens #65 thru #67, are loaded by -64SUPPORT as the basis for making a 64 column screen. These "tiny" characters are only 4 pixels wide, but are the normal 8 pixels high including the space between lines. The CRT screen width is 256 pixels, which divided by 4, is 64 characters wide in tiny characters.

I originally tried to use CLINE by loading the above screens by themselves, but was not successful in making it work, unless the full 64SUPPORT Editor was loaded. I have not found the connection, but then I have not really studied it thoroughly yet. In the meantime, its not such a high price to pay to have the full 64 Editor loaded while I study the problem. If I figure it out, you will hear about it.

As Screen #61 states, CLINE expects the address, of the text to display, the number of characters, and the CRT line number to display them on. One does not have the option of tabbing to the right, all text is displayed starting from the left margin; you fix that by including spaces. The word BLOCK, used by TYPTXT, is the magic word here.

BLOCK removes a screen number from the stack, and loads that screen into editing buffer memory, if it is not already there, then puts the address on the stack. This provides an ideal way use a screen to input and store text, for display, to put it into memory, and to provide the address for CLINE. Each line of a screen is 64 characters, thus the nth line can be accessed, if one just adds an offset to the BLOCK address, thusly: BLOCK 64 n * +, where n is the line number. Then put the character count, and the line of the CRT screen, on which CLINE is to display the text, on the stack and execute CLINE. See line 2 of Screen 61 of TYPTXT. When multiple lines are to be displayed, one can use a loop as is done in TYPTXT of Screen 61.

Screen #62 is arbitrary data to be plotted. I chose the technique of LOADING it on the stack for use by PLOT, but it could have been treated the same way as the text using BLOCK. This is not quite as easy as simply LOADING all the data to the stack. The penalty here, if it is one, is that the last number on the screen ends up on the top of the stack. Therefore, my data appears to be in reverse order.

On to Screen #63, YAXIS and XAXIS just draw lines for the Y and X axes, using LINE as we have already discussed. The only decision here is where to put the lines. You must leave room for the scale numbers, titles etc. YSCALE and XSCALE plot a series of short cross lines on these axes for tick marks of the scale.

In drawing the axes and the scales, you must take into account that there are 8 pixels to the character (ie screen line) in the Y axis, so that one does not have full freedom of location of the ticks, if the ticks are to be opposite numbers to be printed by CLINE. In the X axis, the ticks should be on multiples of 4 pixels, in this case I used 20, in order to line up with the numbers, and yet give me a full scale close to 220 pixels. Zero of course is offset to the right to allow room for the Yaxis and numbers. Notice that in YSCALE, that the starting point is 20, instead of the 16 that YAXIS starts with. This offset of 4 is to put the ticks opposite the center of the numbers. I, of course, did not think of all that the first time I tried to set up the graph format.

The word LABL puts Screen #64 on the CRT ready for the axes to be superimposed. I did not think that way the first time, I always learn the hard way. If one prints the axes first, as seems natural, LABL just wipes them clean. It is not fundamental that LABL will wipe out the axes if they were printed first, but it just turns out to be easier to have the number of characters to print in the LABL loop constant at 64, therefore all lines, including lines 2 thru 12 of screen 64 are printed 64 wide to match the Xaxis scale as shown, thus printing blanks over everything to the right of the numbers on lines 2 thru 12. Please do not let the coincidence, of the 64 screen number matching the 64 characters per screen line and 64 characters to print on a line confuse you.

The most complex word is SCALE, which converts the numbers on the data screen to fit on the axes and scales as designed. First we must take into account that 0 0 is at the top left. That is done by subtracting 104 from the number then using the ABSolute value. Why 104 instead of the 100 for the top of the Y axis? Remember the 4 offset to make the ticks opposite the numbers on the Y scale?

Since the Y axis 10 scale ticks are on multiples of 8 pixels, we are required to scrunch the number in the ratio of 8/10 with our scaling operator */, then add an offset of 16 to account for the two lines at the top of the screen used for titles. The X axis is a little easier, here we just scale by multiplying by 2 (ie 20 pixels are 10 counts) and adding 16 because the zero axis is shifted to the right to leave room of the Y axis numbers.

All the above is relatively simple, if it were required that the program examine the data and set up a set of X and Y scales to best fit the data, it would be at least twice as complex, and several screens like 64 might have to be available for selection. I suppose it is possible to compute arbitrary scale numbers, put them in memory somewhere, like PAD, then print them to the screen with CLINE, but I do not want to think about that until it becomes necessary.

The word POINT just provides a starting place for LINE. Note that after scaling the numbers, it 2DUPS them to save them for the next LINE word, before plotting (DOT) the point.

The word PLOT first 2SWAPs to get two new numbers on top of the stack, SCALEs them, 2ROVERs them to save them for the next execution of LINE, then draws the line.

The word MSG, is I hope, obvious. GRAF is the word that does it all, it sets up the graphics mode with SPLIT, draws the graph basis with AXES, then loads the data to the stack, plots the first point, then executes a loop to plot all the remaining points, with lines drawn between. When through, it waits with PAK for you to press a key to display MSG.

That about uses up the space allowed, so I will continue next time with more graphics.

C U next time, May the FORTH be with U

continued on page 18

Jenny's Younger Set

Sorry about not having an article last month but nobody sent me anything to put in. This month we have a program sent in by Vincent Maker which should be of interest to you younger set. Let us see if you can type it in, and then get a perfect score. Remember gang all you have to do to get an article in the magazine is to save your program on cassette give it to me and I will do the rest. If you want to say something just put the letter on REM statements in a program and I can easily change it over to a word article.

JENNY

```

100 CALL CLEAR
110 REM BY VINCENT MAKER
120 PRINT "THIS IS A QUIZ PROGRAM TO TEST YOUR KNOWLEDGE
OF HISTORY."
130 INPUT "PRESS ENTER TO CONTINUE.":LKJ$
140 CALL CLEAR
150 PRINT "Q1. WHICH SPARTAN KINDLED THE FIRST INVASION
OF ATTICA IN THE PELOPONNESIAN WAR?"
160 PRINT
170 PRINT
180 PRINT "A) ARCHIDAMUS"
190 PRINT "B) LEONIDAS"
200 PRINT "C) CLEON"
210 PRINT "D) NICIAS"
220 PRINT
230 PRINT "PRESS YOUR GUESS"
240 CALL KEY(O,K,L)
250 IF L=0 THEN 240
260 IF K=65 THEN RIGHT=1 ELSE WRONG=1
270 IF K=65 THEN PRINT "RIGHT" ELSE PRINT "WRONG"
280 INPUT "PRESS ENTER TO GO ON":GHJ$
290 CALL CLEAR
300 PRINT "2. FOR HOW LONG WAS THE PEACE OF NICIAS?"
310 PRINT
320 PRINT "A) 500 YEARS"
330 PRINT "B) 50 YEARS"
340 PRINT "C) 25 YEARS<"
350 PRINT "D) WHO KNOWS"
360 PRINT "PRESS YOUR GUESS"
370 CALL KEY(O,J,K)
380 IF K=0 THEN 370
390 IF J=66 THEN RIGHT=RIGHT+1 ELSE WRONG=WRONG+1
400 IF K=66 THEN PRINT "RIGHT" ELSE PRINT "WRONG"
410 INPUT "PRESS ENTER TO GO ON":NM$
420 PRINT "3. WHICH CITY STATE WON THE BATTLE OF
MANTINEA?"
430 PRINT
440 PRINT "A) SPARTA"
450 PRINT "B) ATHENS"
460 PRINT "C) THEBES"
470 PRINT "D) WHO KNOWS"
480 PRINT "PRESS YOUR GUESS"
485 PRINT "PRESS YOUR GUESS"
490 CALL KEY(O,U,I)
500 IF I=0 THEN 490
510 IF U=65 THEN RIGHT=RIGHT+1 ELSE WRONG=WRONG+1
520 IF U=65 THEN PRINT "RIGHT" ELSE PRINT "WRONG"
530 INPUT "PRESS ENTER TO GO ON":FGH$
540 CALL CLEAR
550 PRINT "4. WHEN WAS THE PEACE OF NICIAS INTRODUCED?"
560 PRINT
570 PRINT "A) 234 BC"
580 PRINT "B) 422 BC"
590 PRINT "C) 421 BC"
600 PRINT "D) WHO KNOWS"
610 PRINT "PRESS YOUR GUESS"
620 CALL KEY(O,Y,U)
630 IF U=0 THEN 520
640 IF Y=67 THEN RIGHT=RIGHT+1 ELSE WRONG=WRONG+1
650 IF Y=52 THEN PRINT "RIGHT" ELSE PRINT "WRONG"
660 IF RIGHT=0 THEN A$="TERRIBLE 0/4"
670 IF RIGHT=1 THEN A$="POOR 1/4."
680 IF RIGHT=2 THEN A$="FAIR 2/4."
690 IF RIGHT=3 THEN A$="GOOD 3/4"
700 IF RIGHT=4 THEN A$="VERY GOOD 4/4."
710 INPUT "PRESS ENTER TO SEE HOW YOU WENT.":IOP$
720 CALL CLEAR
730 PRINT A$
740 END

```

continued from page 20

Since Multiplan does not allow blank spaces in cell names we must insert an underline character between Beg and Bal. Now we must name the cells and ranges we are going to use:

1. Place the pointer over the label "Deposits" (R3C1).
2. Select the Name command and we see:

NAME: define name: Deposits to refer to: R3C1

3. Press the Tab key (CTRL 2) to move the pointer to "refer to".
4. Press the down arrow key so that the submenu refers to row 4 of column 1, like this:

NAME: define name: Deposits to refer to: R4C1

5. Press the colon key : to begin a range and then the down arrow key ten times, or as many as you wish, so that we have:

NAME: define name: Deposits to refer to: R4C1:R15C1

6. Press <ENTER>.

Now the cells below the "Deposits" label are, collectively named "Deposits". To name the cells under the "Checks" label, place the pointer to R3C2 and follow the steps above. You may need more spaces for checks than for the deposits.

The single cell R1C2 needs to be named by placing the pointer in it's "label" cell R1C1, except type in the name "Beg_Bal" as that is the name we want to use in our formula.

To calculate the ending balance, place the pointer in R1C4, press the = key and enter our formula:

Beg_Bal+SUM(Deposits)-SUM(Checks)

Press <ENTER>. Any data put into the Deposit column or the check column will be calculated into the Ending Balance at R1C4. You will notice that after each entry there is a lapse of time while the sheet is being refigured. As mentioned on page 8, using the Options option to Recalc(No) will speed things up, but when you want the final result you must use the Recalc key (FCTN 8).

REVIEWING CELL NAMES:

When you have many cells named or even when you just forget the names you have used, the names can be reviewed by using the Name command. Select the Name option and a cell name will appear with the cell or cells it refers to. Use the right arrow key to "scroll" through the names on the sheet. When you have found the one you are looking for, or seen all of them press <ENTER> to return to the main menu. You may also JUMP to named cells by using the Goto command. Its submenu offers Name as the first option. If you select Name, <ENTER> you are offered a name which will respond to "scrolling" as above by using the right-arrow key. When the desired name appears, press <ENTER> and the pointer will jump to the named cell or the upper left cell in a range. If you wish you may type in the name of the cell you want and press <ENTER> without scrolling through a list of NAMES.

RULES FOR NAMING CELLS

1. Maximum name length is 31 characters.
2. Names must always begin with a letter, A-Z.
3. After the first character there may be digits, periods or the underline character.
4. No other punctuation is permitted.
5. Do not include combinations of characters resembling cell addresses (such as R2C5).

Defaults are offered in the displays as proposed entries, but if what is there is not what you want, either type in the proper response or point to it.

continued on page 18

Regional Group Reports

Meeting Summary For September

Banana Coast	08/09/91	Sawtell
Central Coast	14/09/91	Saratoga
Glebe	12/09/91	Glebe
Illawarra	09/09/91	Keiraville
Liverpool	13/09/91	
Northern Suburbs	26/09/91	
Sutherland	20/09/91	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 held 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, including Word Processing, Spreadsheets and hardware repairs. Last month we examined Bob's new Grade Standardiser program and software that Rolf brought to the meeting, including Comic Show Animator. 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 REPORT

The July Regional meeting again proved to be quite entertaining with a good roll up in attendance. As usual, we digressed slightly, with a large slice of the evening being devoted to motor mechanics.

Back to computers, and Derek Wilkinson produced a very handy utilities program which loaded via the Johnson Loader software. The disk included Typewriter, Printer Set-up and Disk Labler.

My 32K card had failed earlier in the week and Garry Wilson obliged with a replacement chip which was installed during the evening.

Rumour has it that Ron Stewart's Hard Disk Controller Card and Disk Drive are coming out of mothballs for our next meeting. Derek cannot wait to get his hands on them.

Regular meetings are held on the third Friday of each month at the home of Peter Young, 51 Jannali Avenue, Jannali at 7.30pm.

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.

SEPTEMBER MEETING - 7TH SEPTEMBER

The September meeting will involve demonstrations of the latest software as well as a whole host of other activities. Some of these will include:

- * 9am to 11.30am - Computerised model train group
- * 12noon to 2pm - Assembler Class. Bring your E/A book, note pad, pen and a formatted SSSD disk.
- * 12noon to 4pm - BBS Demonstration (as & when requested).
 - Latest games for young and old.
- * 2pm to 4pm - Modem interfacing clinic (bring your own system if you need assistance).
 - Latest software.
 - Technical advise and repair help. Bring hardware projects such as RamDisk etc. if you need help.
 - TISHUG Shop open.
 - Publications Library available.
 - Programming help for your TI99/4A.

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

October	08 September
November	13 October

These dates are all Sundays and there is no guarantee that they will make the magazine unless they are uploaded by 6:00pm, at the latest.

TISHUG Meetings for Sydney, 1991

September

Modem clinic. Bring your modem in if you are having problems using it to connect to the BBS. Bring in your system as well for a full check-out. Demonstrations of the latest software or hardware.

October

The third Buy, Swap and Sell day. Your last chance this year to get some money for Christmas presents or to get an early present for yourself.

November

The second all day tutorial session.

December

The Annual General Meeting followed by some festive eats and drinks. Make sure you attend and give your support to all the workers in the club. ○

continued from page 19

```

450 IF LEN(T$)>128 THEN CALL WRITE(T$,
T1$): T$=T1$
460 T$=T$&"W;N1;A1 1"&R$&"E"&R$
470 IF LEN(T$)>128 THEN CALL WRITE(T$,
T1$): T$=T1$
480 PRINT #2:T$&RPT$(CHR$(0),128-LEN(T
$))
490 CLOSE #2
500 END
510 SUB WRITE(T$,T1$)
520 PRINT #2:SEG$(T$,1,128)
530 T1$=SEG$(T$,129,LEN(T$)-128)
540 SUBEND
    
```

○