TISHUG

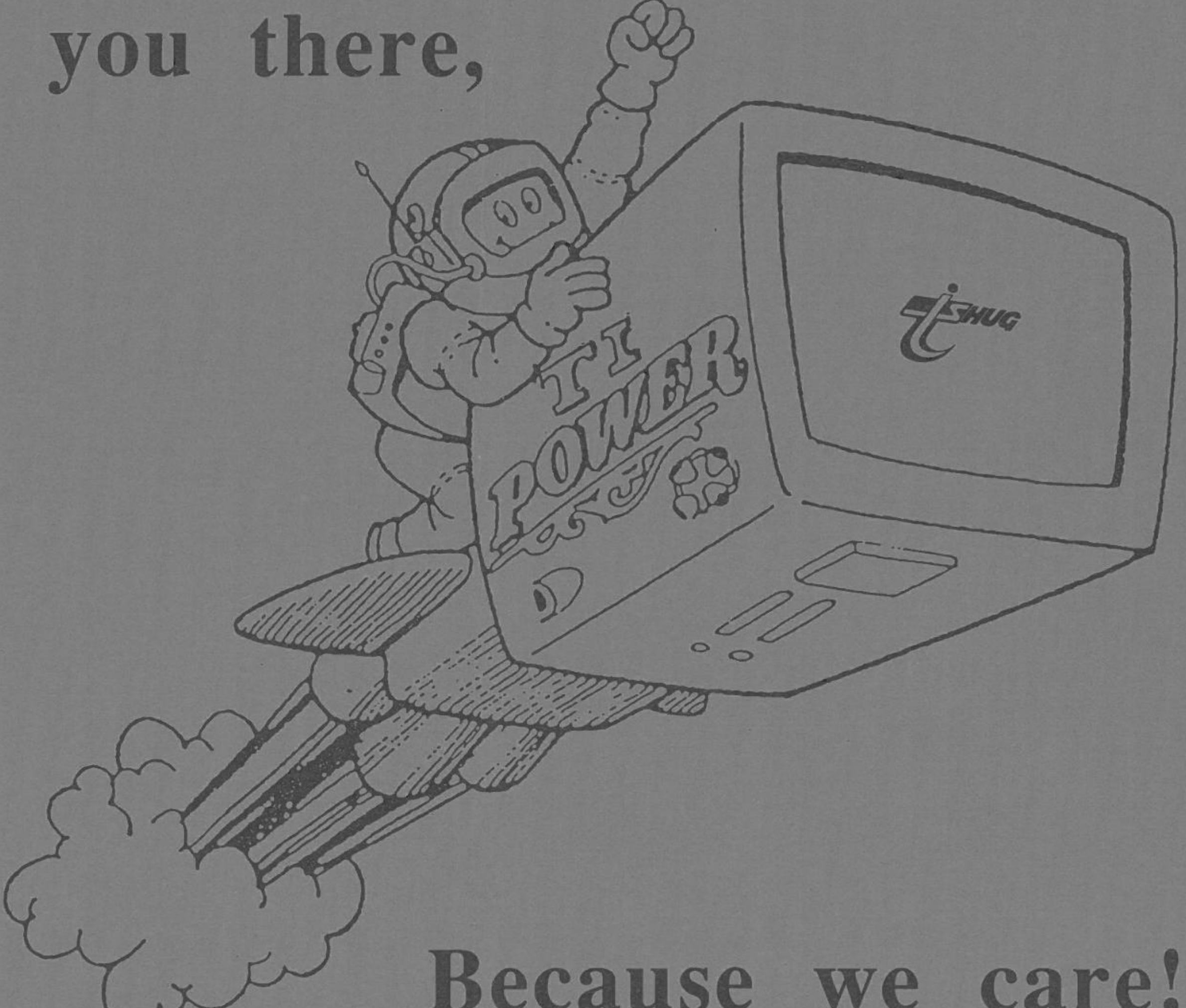
NEWS DIGEST

Focusing on the TI99/4A Home Computer

Volume 10, Number 7 August, 1991

Registered by Australia Post - Publication No. NBH5933

See you there,



Because we care!

Sydney, New South Wales, Australia $3

TIsHUG (Australia) Ltd
A.C.N. 003 374 383

TIsHUG News Digest

August 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 3rd of August 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.

TIsHUG News Digest ISSN 0819-1984

Index

Cover courtesy TIUP Perth

STOP PRESS

TIsHUG is now the official TI99/4A repair and service facility for Texas Instruments in Australia.

Editor's Comment

by Bob Relyea

For the second straight time I was surprised (disappointed?) by the lack of gear for sale at the Buy SWAP and SELL DAY. Some of us were pondering as to what it all meant. There were plenty of other activities, though, and there will be this month as well. Be sure to have a look at Ross' article to get some ideas as to what will be on.

I was able to get quite a few articles done this month about Word Processing as well as using TI-Writer to do graphics. There will be more on these topics in the coming months. Hope you enjoy them! o

Co-ordinator's Report

by Dick Warburton

TIsHUG (Australia) Ltd, now has the permission of Texas Instruments to repair and service TI99/4A computers and peripherals Australia wide. As many members know already, we offer a console exchange service to club members for only $30. We also have been able to help members with a variety of other problems and even repair some of the more exotic TI99/4A peripherals. As well as repairing consoles etc., we can help members to add colour monitors, upgrade to a better system through multifunction cards and memory expansions. Low cost methods of expansion are now available for those willing to do a little for themselves with club help. At present the club shop is reasonably well stocked to meet the needs of TI99/4A users. At present we are waiting on new software from Asgard as well as some new hardware. We are particularly interested in developing the use of the 80 column card in 1991, particularly if we can source some more low cost monitors. We are presently looking at the possibility of using CGA type monitors, if the resolution is adequate.

Projects presently under way include the famous 64K EPROM RAMdisk. Some are completed already. The shop is now stocked with memory chips and Funlwriter EPROM kits. We still have a few RAMdisk kits left, so get in before they all disappear. A further development of the EPROM RAMdisk is the distinct possibility that we will be able to add 32K EPROMs to RAMdisks with 8K memory chips. There are some limitations in that presently it cannot address more than 256K of memory. Craig Sheehan and Peter Mudie are working together on the problem and are confident they can get a working card up and going very soon. I wonder whether we could take this a little further and develop a module with 128k of EPROM programs for users with more limited systems. Judging from our success with EPROMs, I cannot see why not.

By the way, for those of you with troublesome clock cards, the problem appears to be solved. Peter Mudie seems to have found a simple solution. This should be written up soon.

Another project is the effort to build a computer controlled train system. The group is "in training" each month at Ryde School at 9 am. We are presently aiming at completing a suitable power supply and an interface for the computer. If you are interested, come along and join in the effort. If you would like some training in basic electronics or in practical soldering etc., come along. There will be scope to learn some programming skills as well.

Sydney monthly meetings will be formalized again each month, starting at 2 pm sharp. The usual services will be available:
the shop;
printer ribbon re-inking;
the library;
technical help from Geoff; and
hopefully more computers available for games for the young at heart.

If you have some favourite games, bring them along and try them out with the others. If you can bring a console along together with a television and some modules, do so. We will find a way to set you up. The more computers working the better.

We intend to hold a TI-Faire some time next year. This will entail quite a lot of work and we need some volunteers to help us out. Hopefully, we will attract some of the latest and best software and hardware from overseas. Hopefully we will have plenty to show for our own efforts. Keep those RAMdisks flowing off the assembly lines and get with the EPROMs. We need help to organize the program and a range of other things. A committee will be needed to begin fairly soon. If you like the idea of meeting a few famous TI99/4A names, trying the latest hardware and software, put your name down to help us out. We are also considering the possibility of "bringing out a famous TI99er" for the Faire and looking after her/him for the period.

Do not forget that there are other meetings available for members between monthly meetings. For example:
the Liverpool meeting at Larry Saunders' place.
the Glebe meeting (now Strathfield) at Mike Slattery's (34 Arthur Street, Strathfield West); and
the Sutherland Regional meeting at Janalli.

These meetings are listed on the back page of TND but others are not:
the console repair group at Cyril Bohlsen's place (4 Madelaine Ave, Northmead phone 6395847), now on the second Friday after the meeting;
the word processing group at Percy Harrison's home (3 Storey Street, Ryde) on the second Monday of the month; and
the technical group which meets at Ryde Infant's School on the third Monday evening of each month, for those who want help with any technical problem in making or fixing things. We help each other.

TND is your monthly magazine, and we need more of your expertise in there. It does not matter if you are learning and want help. Write down your experiences, your tips, your programs, your frustrations, anything humorous and send them off to the editor. Let us try to rely less on overseas articles. Some of you could make significant contributions if you chose to do so. If you have queries about anything related to computers, send them in. If you have a complaint, write it down and send it in. Let us use TND more fully. It is our lifeline for our members.

Well, 'bye for now. I will see you at the next meeting.

Dick Warburton. ○

Insights

by Ron Albright, USA

As the tenth anniversary of my joining the "Computer Age" approaches, I have been given to much reminiscing. I remember my first computer with almost (operative word "almost") the same sense of excitement that I did my first kiss or my first..er..well, car. It was June, 1981, the early days of the bloody "Home Computer Price Wars".

It was a terrific time for the new computer buyers. The "Home Computer" was the watchword and the major players in this low-end market were Texas Instruments, Commodore, Timex-Sinclair and Atari. I have purposely excluded IBM and Apple whose products were well outside the price range of the casual enthusiast. The major products were the TI99/4A, the Commodore 64 and VIC-20, the Timex-Sinclair Z80 and the Atari400 and later 800.

Like millions of other parents with the idea, I was going to get something to help educate the kiddies on what was surely to be the "Wave of the Future", something called "computer literacy". What I really was after was a computer for myself. I was seduced by the idea that a computer would somehaow make my life easier and more fun.

After some teeth gnashing and second guessing, I bought the TI machine. At a "mere" $329, the 16K silver-and-black console had it "all". I was not at all sure what it "all" was, but the ads featuring Bill Cosby confirmed that the TI99/4A did everything and that was good enough for me. I plunked the money, bought a couple of cartridges (one for the kids; Munchman, a PacMan clone) and one for me (Tax and Investment Record Keeping). I was told I neede a disk drive, so I dropped another $500 or so for that.

I was ready for an adventure that has yet to end. After convincing my wife that it was "for the kids", I proceeded to lock the door to my office and began learning about "computing".

I bring this all up now because I think we all owe so much to these pioneering little machines. I know I do. Probably many of you started with one machine and have since moved up and on to another, more powerful model. The little TI99/4A (for which I, over the course of the next four years, must have spent over $2500 expanding it with modems, RS232 interfaces, printers, memory expansion and software) taught me 90 percent of what I now know about computers. When I upgraded to an IBM clone in 1986, the knowledge I had come by using the little TI99/4A stood me well in this new genre of machines. I thought I would recite a few lessons carried on with me from the TI99/4A to show you just what these "Classic Computers" have taught us.

1. Computers seldom die; they just get unplugged. The very same console I bought in 1981 is still running. Plug it in and it lights up and computes just as well as it did when I popped open the box. Sure, I have replace the keyboard (the keys started sticking and displaying 20 "A"s when I only wanted one), but nary a chip, solder connection or diode has faulted.

2. Computers are not terribly fragile. I learned early on not to be intimidated by computers. Since I could get a replacement for the TI99/4A for $25 when they went out of production, I became very brave when it came to "popping the hood" and taking a look inside. I installed cards, hot-wired cables and modems and replaced keyboards without a bead of sweat. When I got my new, shiny IBM, I never lost the hacker's mentality. When others, new to computers, got their IBMs, they hired "experts" to install boards for them and paid exorbitant prices for cables and memory upgrades. A whole new industry arose. Since I had cut my teeth on a relic, I simply kept doing on my $2500 monster what I was doing on my $25 bargain basement TI99/4A. I would hesitate to guess how much money I have saved by being a "pioneer" (you have seen the pioneers; they are the ones you pass on the trail of computing lying at the side of the road with arrows in their back and empty billfolds). I owe it all to the TI99/4A.

3. Basics are basics; concepts are concepts. The (now) rudimentary editor that I used with the TI99/4A (TI-Writer) taught me all the basics of word processing. I mean, folks, there are only so many ways you can cut and paste, block copy or move, set margins, etc. When I stepped over to WordPerfect (4.1, then 4.2, then 5.0 and now, 5.1), what I carried with me from "TI-dom" allowed me to start a full leap ahead of other new PC computerists. Ditto for spreadsheets (the TI99/4A had a version of Multiplan), data bases and telecommunications (how many ways are there to build a null modem cable?). Power and speed may change but concepts do not.

With these tenets in mind, I think I have a solution to the problem of computers in the schools. We all know there are too few and we all know it is due to cost constraints. In most US schools, there are 20 to 30 students per computer. If the students are lucky enough, they get an hour or two per week on the keyboard. We know that is not enough, but cannot afford to improve it.

Why do we not try this? Given that
A. computing is computing, regardless of the brand name stamped on the cover; that
B. basic software concepts are constants (how to do the functions are not, but are not related to the computer, themselves); and that
C. computers are hardy devices that will, with a modicum of care, last probably a decade; then
why do we not start a program to revive the "orphan" (defined as out of production) computers of the world. Let us start PTA-driven drives to ferret out those closeted classics and move them into the schools where they belong. Let us make PTA membership dues payable by donating a computer (any computer) to the school. The owners will get a tax write-off and the schools will get a warm CPU.

Let us get the children out of in the neighbourhoods, not selling Christmas wrapping paper or candy, but asking people to go into their attics ans resuscitate their old computers for their local schools. Let us plug these up all over the classrooms and turn the kids loose on them. Instead of "Computer Labs", a sequestered, hallowed hall where the school's computers lay enshrined (and unused), let us have them lined up along the walls of every classroom, plugged in and ready to roll. Lay a few software packages (probably cartridges and an occasional cassette tape), and Voila!

Purists will argue that this will never work. A teacher cannot be expected to know the idiosyncracies od a dozen different brands of computers and can never teach the class to use such a potpourri of OSs.

I say, correct! Why try? I tell you what I think. If you have the computers around, have them plugged in and give the students time to use them, the teacher can sit back and watch. The children will become experts in no time. They will teach each other.

Ever try to learn a Nintendo game? How to jump, shoot, dodge, find secret doors, etc. I never can. But give one to one of my 9 year old twins and, within 30 minutes, they are on board number 12 and racking up points in the high six figures.

Children approach computers differently. We want to know "why", they only need to know "how". They find out by trial and error and move on..and on..and on. Computers are not barriers to them. They are fun and they are tools.

How many computer classes have ever taken a computer apart? I dare say, probably a few. With the little orphans, they can whip out a phillips head screw driver and go at it. Find out what makes them tick. What is the harm?

I think it will work. I think it is an idea so simple that it has been overlooked for much too long. There is even a national clearinghouse organisation that will facilitate just such an exchange. The National CRISTINA Foundation (1-800-CRISTINA) will help you match your donations with worthy causes.

With millions of TI99/4As, VIC-20s, Coleco ADAMs and Sinclairs stored in the attics and closets of America, there are plenty to get the classrooms up to speed in no time. I think it is time we started. I have a TI99/4A I will donate today. I will keep the other two for the twins. What about you?

Secretary's Notebook

by Terry Phillips

Due to technical difficulties, I was unable to produce my regular column for the July Newsdigest. The problem was a busted Television, which was finally repaired at a cost of $95. I am still having problems with my modem, which prevents me from uploading to the BBS. Ross has come to my rescue and loads my file onto the system from the disk I send him. Thanks Ross.

The last meeting was the buy, swap and sell afternoon, and it was pretty well attended, although there were not a great lot of items on sale. The club picked up two fully expanded systems, one with a printer, for $500 each. Also at that meeting the model train technical group had their first get together and are now working on building up a power supply for the proposed model train layout.

There are 4 new members to welcome to the club:

John Ashton - Baulkham Hills
Jenny Finikiotis - Bondi Junction
Larry Reid - Ipswich
John Dixon - Carina

Memberships that expired in April, May and June have nearly all been renewed with only 23 now

continued on page 21

Letters to the Editor

Dear fellow TI99ers,

I am very pleased that I became an overseas member of TIsHUG. The Newsdigest is a high quality spring of information and I have a great admiration for the activities you are able to support. I feel that the dues I have paid for the Newsdigest are not the right value for what I receive and so I must do something to make up the difference.

I decided to adapt a program that I wrote for my son some time ago; the most difficult part was to translate the comments into English. This program was to run quickly in a Forth context so it is written in Assembler. Its aim is to solve a system of linear equations of the first degree. It was successful from the first and the work-mates of my son, who is in charge of the laboratory, were most impressed with my program. It is easy to be the smart one in the town! The laboratory is equipped with about ten TI99/4As and Geneves that are used to drive experiments and to interpret data. This model of computer is good for that job as the information about the internal workings of the computer are in the public domain, it is easy to program which helps as there are no commercial programs available for this job and Forth is available which the students know well. Many tools exist to help: debugger; disassembler; disk sector access utilities; and Funnelweb operating system. With one or more RAMdisks, as Dick Warburton said in the March issue, the TI99/4A is a very good machine which is not expensive and may well have the best quality to price ratio.

I am the owner of the original program and you may use it as you want, as well as everything on the floppy I have enclosed. It is Freeware. I know that such a program may not be very useful for all TIsHUG members, but there may be a few who would find a use for it.

I have also worked on direct access to my disk controller (a Myarc one) and was able to write a program to read a track and to read and write sectors. When I know how IBM and Macintosh files are configured, I shall write a program to convert data from one format to the other. Some years ago, when I only had MaxiMem and four pages at address >4000 for CRU addresses >1400, >1402 and >1600, >1602, I wrote a fast cassette transfer program. A year later I constructed my first Horizon RAMdisk and as I had no disk drive or controller I saved and loaded the 718 sectors in 15 minutes on tape. The basis of such a program was published in "99er Magazine" in 1985. At the present time it is of not much interest to me but I may send you a listing.

In September I boosted my hardware with a second disk drive. The first one is a 5.25 inch half height, made in Japan and sold by BASF as model 6129, 48 tpi, double sided. The new one is also made in Japan by Epson. It is the SMD-280 model and takes 3.5 inch floppy disks with 80 tracks. I use a Myarc controller and put in the 80 track EPROM and set the internal switches so that the new drive is DSK2 for 40 tracks and DSK4 for 80 tracks! This is possible by using two jumpers (or setting two switches) on the drive for both DS1 and DS3 (DS0 corresponds to DSK1). It is necessary to use Disk Utilities as DM1000 cannot handle 80 tracks, which is a price that may be paid for its swiftness. This solution forbids copying between DSK2 and DSK4. For me this was a discovery that Lou Amadio had mentioned long ago.

It is possible that I may be interested in buying software, floppy disks or hardware from the TIsHUG shop. Is it possible for an overseas member to do this? If it is would you please inform me of the price for air mail for a floppy over the price given in the TND.

I want to thank the person who sent me the TNDs (1 to 8) from last year (Rolf). I was very happy and grateful to receive them. I was only enrolled as a member from 1st November 1990 so I am in debt for those 8 issues. I propose that I send you something to cover the cost of those the next time I send you a cheque.

I enclose with this letter some mailing coupons and I thank you for the time you have spent to read and decode my English.

Yours sincerely, Pierre Garoche.

[I have edited Pierre's letter and tried to make it more readable. I hope that I have not lost Pierre's meaning in the process. Although some sentences were a bit obscure, he has done a better job than I would be able to do writing in French. I trust that other members will be inspired to write something for the club by the efforts of one of our newest members. Geoff Trott] o

Dear Bob,

Almost a decade ago, I bought a TI99/4A Home Computer. Over the years it grew and expanded far beyond the expectations of myself or, for that matter, the expectations of the designers at TI.

Today, with mixed feelings, I sold my TI99/4A system to the club.

Firstly, I feel sorrow at the loss of a long-time friend (christened "Camielle" years ago by my wife in a moment of jealousy). It has been a reliable and good system, giving me and my family much pleasure and service over the years. It has been of great use to my children in their school work, as well as developing and maintaining their understanding of computers, which is essential in today's world.

Secondly, I am pleased that it will be used to good purpose and will not be consigned to the back of the cupboard or, worse, to the shed where it would deteriorate, as neglected and unused electronic things do.

My reason for selling is, to some extent, lack of time, but mainly, a lack of space. The children grow but the house does not.

I also have mixed emotions about Texas Instruments' abandonment of the TI99/4A. At the time it went on the market, it was an innovative concept with advanced features. There appeared to be some limitations, such as addressable memory, but I feel that the greatest limitation was TI's marketing. I believe that "they can make but can't sell".

The good part is that the user groups around the world picked up the ball (or should I say "picked up the orphan"?) and ran with it. One of the Front Row Forwards in the game was the Sydney TIsHUG. The members of the clubs have banded together, maintained the enthusiasm that the machine deserves and developed the TI99/4A, as I said before, beyond expectation, overcoming its perceived limitations on the way.

The advantages of this are many-fold. There is, for instance, more input into the development, more ideas, imagination of what can be done, than is available from one department of a company. Hardware costs are less because the design and development is done by enthusiasts who usually are not even interested in recouping the expenses. This also applies to software. It is a paradox that, with most things, you get more out of something than you put in.

There is the pleasure of the challenge, the enjoyment of the work, the pride of the accomplishment, the companionship of sharing ideas and the satisfaction of contributing. In short, were it not for the abandonment, neither the TI99/4A or the user groups would be what they are today and both are great!

continued on page 10

TIsHUG Software Column by Rolf Schreiber

I would like to take this opportunity to welcome back Craig Sheehan and Robert Brown after a long absence. Robert had contributed enormously to the Club in the past with his regular Games Info columns in the TND, while Craig's software and hardware contributions (including the EPROM RAMdisk) have been incalculably useful.

Elsewhere in this month's magazine is an article by my good friend Lou Amadio concerning the involvement of interested Club members with the Software Library. I will bring along a number of disks of software each month, for people to review and return the following month. The reviews should be on disk as Dis/Var 80 files, suitable for processing with TI-Writer (or FWB). If the software is commercial, you will still be expected to buy it, and not to pirate it. I like to think that our members are both mature and honest enough to realize that there is no such thing as a free lunch, and that it is not smart to kill the goose that lays the golden eggs. In other words, keep your end of the bargain!

Software releases for August 1991

All disks are SSSD and will be available from the TIsHUG shop next meeting (or by mail order) for $2.00 each.

DISK A405 is called 1000 Words and was completely written in assembly language by Norman Rokke. This program is more fully described in another article in this issue. Suffice it to say that 1000 Words is a utility which allows the incorporation of graphics in text, when printed through the TI-Writer Formatter.

DISK A436 is a fairware offering from Charles Earl, the author of Press. Charles needed sophisticated debugging software, which he had to write himself, during the development of Press. Hotbug is the result of Charles' need for a better debugger and is the answer to every software hacker's dream. Included on the disk are full instructions and three versions to load into different areas of memory, including the Supercart module, and a similar set of versions which can be used with a remote terminal through an RS232; ideal for debugging programs which take up all the available memory in the TI99/4A.

DISK A437 is a fun program called Nasty. It was written by Chris Lang and it turns the computer into a machine with a mind of its own. Some of the words used by the program as replies to input from the keyboard are not suitable for small children, so read the README file before you let your kids loose on this one! As an added bonus I have included another game written by Chris, called Segregation. This is a game which requires keen powers of observation; the object of the game is clearly explained in accompanying instruction files.

DISK A438 is called More Games. The author of most of these games is John Behnke, and the disk comes from the Boston Computer Society's software library, where it is identified as disk BCS 49. The programs are a mixture of assembly language and Extended BASIC games, and include two versions of bowling; a monopoly game (an American version, nowhere near as good as Ross Mudie's version of TI99-opoly); a version of Pacman called Buzzard Bait; a game of chance called RAT/RACE; a game called BOXING and a game called MISSILE. The disk lacks a menu to automatically load the programs, and there is not enough room on the disk to install BOOT.

Tigercub Software Release

Tigercub Collection #1 comprises a collection of games which should entertain kids from 9 to 90. When Extended BASIC is selected, the Tigercub Quickloader menu comes up with 13 selections. Number 13, Motorcycle Jump, was not found on the disk, but the other 12 selections more than made up for the missing game. These were:

1) Alley Craps: play craps (a card game) against the computer.
2) Whitewater Run: guide your raft down Whitewater Canyon whilst avoiding obstacles.
3) Scrum: 511 different puzzles-in-one based on a grid of 9 squares.
4) Haunted Graveyard: find you way through a graveyard in the dark without getting killed.
5) Mechanical Aptitude: match shapes that fit together.
6) Four in a Row: play against another person or the computer.
7) Gameloader: Assembly loader for Bomber, Cat & Mouse and I'm Lost (all assembly games in E/A 5 format).
8) High Jump: Beat the high jump record. Speech option is available.
9) Mr Kroaker: Catch butterflies. Requires joystick and has speech available.
10) Leaper: Similar to Sir Prancelot. Requires joystick and comes with instructions.
11) Left/Right: a co-ordination game from Sweden. Dual joysticks are essential.
12) Mazzo: move a ball through a maze. A great game from Tony Constantinidis (a former TIsHUG member).

Commercial Software

Tris, the exciting game from the USSR is now available for the TI99/4A in module form. This version of Tetris closely follows the PC version, and packs a lot of features into 8K of program code. Tris is reviewed elsewhere this issue, and is available in very limited numbers for $25.00.

Typewriter 99 is another new piece of software which comes as a plug in module with printed instructions. It was reviewed by Lou Amadio in the TND in May this year and is available in strictly limited numbers for $25.00.

Page Pro Medical Clipart is a set of four DSDD disks of high resolution pictures of human anatomy and related medical topics. These pictures could be useful to anyone studying medicine or anatomy, just interested in art. They are available as a set for $10.00, or singly for $3.00. Page Pro 99 is required to view or print out these pictures. I will have some pictures on show at the next meeting.

Page Pro Line Fonts is a package of 23 new line fonts, including 2 pattern fonts that can be combined to create hundreds of different patterns, 2 line element fonts and 19 line graphics fonts that contain foreign language and other special characters compatible with the series of small fonts in Page Pro Fonts #1 and #2. The package includes reference sheets containing all the patterns that can be created with the pattern reference fonts. The price is $9.00.

Software Review

TRIS, by Lou Amadio

TRIS is the TI99/4A implementation of that famous game TETRIS. TETRIS was conceived by a Russian programmer and has been adapted to run on most contemporary computers.

TRIS was designed by Jim Reiss and Chris Bobbitt of Asgard Software, Rockville USA.

TRIS is remarkable in that it combines strategy and arcade style action in the same game. The object of the game is to manipulate falling shapes in order to form a solid row. When this happens, the row disappears. As the game progresses, the shapes fall faster and faster.

The game is easy to play (at least in the early stages) but difficult to master. Strategy plays an important role in achieving success with this game.

continued on page 15

TIsHUG Shop with Percy Harrison

Hi, firstly I would like to apologise for my absence from the June meeting and for omitting my shop article from the July issue of the TND magazine, both of which were due to my wife dragging me off to the Cook Islands and Western Samoa for two glorious weeks. The break from the hustle and bustle of the big smoke was much appreciated, especially as all we did was to laze about in the tropical sundrenched beaches and enjoy the island sightseeing. It is the first time ever that I can remember getting a winter suntan.

I would also like to thank Alf Culloden and Terry Phillips for taking on the shop in my absence and was more than pleased to see that the days takings were not affected by the absence of our software disks. Thanks Alf and Terry for a job well done.

Unfortunately there is no new hardware to offer you this month but I would suggest that the members who indicated that they would be buying the Eprom Ramdisk and kit, but have not yet done so, do so without delay as we are rapidly running low on stocks and those that miss out may have to wait quite a while before another PC Board run is organised. Unfortunately I was not given the list of the members who indicated that they wanted the Eprom Ramdisk so sales have been on a first come first serve basis and, therefore, some members are going to miss out on this first run. Sorry about this but we must sell these cards and kits as quickly as possible so as to recover the large monetary outlay we had to make for this stock.

I am somewhat disappointed that, to date, there has not been a single member advise me of their requirements for programmed EPROM Chips. In order to meet your requirements and to keep our workload and costs to a minimum it is necessary to know in advance just how many and what programmed chips are required. This way we can meet your needs at the least cost to the club and ultimately to you our members so please give a little thought to what you want to include on your Ramdisk and let me know.

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 1 (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
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

Packaging and postage charges:

	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
5.25 inch half-height drive (1.25 Kg)	refer to your local post office	

o

John Birdwell
In Memorium

by Rolf Schreiber

John Birdwell died on 27 December 1990, after losing his fight against liver cancer. John was best known as the author of Disk Utilities, and was selected as our fairware author of the month in March 1990. I received the latest version of Disk Utilities, Vn 4.2, directly from him in response to the contributions that we collected on his behalf that month. In keeping with his expressed wishes we did not release the documentation that came on the disk, since John wanted to restrict the documentation to those Users who had personally sent him a substantial fairware contribution, ie ten dollars or more.

A John Birdwell Memorial Fund has been set up in the USA in memory of this talented programmer. The trustees of the fund intend to award an annual prize to the individual (or organisation) who has made the most significant contribution to the TI99/4A and Geneve community that year, as an acknowledgement of service to the Users of this community.

Since we are a part of this community, it is only fair that we should give something in return for all that we have taken in the past. As a mark of respect to John and his achievements, we are going to release Disk Utilities Vn 4.2 as I originally received it, with complete documentation. I am asking all of you who regularly use this program to purchase this disk for $11.00. Ten dollars will be sent to the John Birdwell Memorial Fund from the sale of each disk, as our contribution to this worthwhile cause. The name of everyone who contributes will be sent along with the donation, to show that there are people in the TI99/4A community in Australia who also care. o

1000 Words

a TI-Writer Utility, by Norman Rokke, OH USA

OVERVIEW

1000 WORDS is a utility program for use with TI-Writer. It converts picture files from TI-Artist (files with the _P extension) to display variable 80 files. The files created by 1000 WORDS can then be printed from the Text Formatter of TI-Writer. This gives you the capability of using TI-Writer to print documents which contain both text and graphics. Since the documents are printed using the Formatter, the text portions are printed using the normal text of your printer, not graphics which look like text as is the case with other programs for incorporating graphics with text in a document. The result is that text is printed just as rapidly as normal from TI-Writer. You do not have the selection of many different fonts and the ability to intermix graphics and text on the same line as with other programs however. Nevertheless, if you are satisfied to have documents containing sections of text alternating with sections of graphics, 1000 WORDS gives you the capability. In addition, if you are already familiar with TI-Artist and TI-Writer, you know essentially everything you need to know to produce documents such as those described above.

SOFTWARE REQUIREMENTS

TI-Artist (from INSCEBOT INC.) - This software is needed to create the picture files which are required by 1000 WORDS.

TI-Writer - The Text Formatter of TI-Writer is needed to print the files created by 1000 WORDS. You may use either the original Command Module version or one of the Extended BASIC loader versions (FUNNELWEB, BA-Writer etc.).

PRINTER REQUIREMENTS

Below is a list of printer codes used in the files produced by 1000 WORDS. Please check to make sure that these codes do the same things with your printer. The codes are for the TI 99/4 Impact Printer (same as Epson MX80). As such, they will probably work with most Epson printers and compatible printers.

ASCII code	Description
9	Moves printhead to next set tab position.
17	Enable printer.
19	Disable printer
27,75,n	Sets line spacing to n/72 inch.
27,68,n,0	Sets horizontal tab at column n.
27,75,n1,n2	Normal density graphics mode. (480 dot columns per line)

Note to users of the TI 99/4 Impact Printer. In order to allow selection and disabling of the printer, a DIP switch inside the printer must be changed. When this is done, the printer is not ready to accept data when the power switch is turned on. Whenever you want to use the printer you must turn it on and then send it ASCII code 17 to make it ready to accept data. There are two ways that this can be done. The first is by means of a short program from BASIC or Extended BASIC. Such a program is provided on this disk under the filename PON. You will have to change the printer filename in the OPEN statement if your printer is not connected to the parallel port. The second way to send the code is from TI-Writer. The file PRON on this disk will do this. Simply load it into the Text Editor and use PrintFile (PF).

RUNNING 1000 WORDS

1000 WORDS can be run from option 5 of Editor/Assembler or from option 3 of TI-Writer. Simply enter the filename DSKn.WORDS and press ENTER. When you see the title screen, press ENTER and you will see the main menu.

1. CREATE OVERLAP FILE

2. CREATE FORMATTER FILE

3. QUIT

CREATE OVERLAP FILE

This option is available to help you create the TI-Artist files needed to produce a TI-Writer file which will print graphics the entire width of the page. This requires two picture files from TI-Artist (actually slightly less than two complete files as shown below). There is a small area of overlap between the two files (shaded area in drawing below).

```
|--------------------|----|-----------------------|
|                    |////|                       |
|                    |////|                       |
|                    |////|                       |
|                    |////|                       |
|                    |////|                       |
|                    |////|                       |
|                    |////|                       |
|                    |////|                       |
\___________  ____________/                       |
            \/         \____________  ____________/
                                    \/
       FilenameL_P              FilenameR_P
```

You should create the file defining the left half of the page first. When you have that completed, you can use the CREATE OVERLAP FILE option to produce a TI-Artist picture file which has the overlap area defined at the left edge of the screen. This should help you maintain vertical alignment between the left and right portions of the graphics. The background colour for the overlap area in the right half of the graphics is red. This serves to define the area of this file which will be included when printing is done. The overlap area is defined by the data in the file defining the left half of the page.

TRY IT OUT- On the disk is a file DEMOL_P. This is a TI-Artist picture file which contains some geometric shapes.

Let us create the overlap file from this file. Load 1000 WORDS and select option 1 from the main menu. The prompt

ENTER TI-ARTIST filename
(omit L_P extension)

is displayed. Change the drive number if necessary and enter the filename DEMO. Note that the extension L_P is not necessary. Press ENTER and the program begins. When the program is finished you will be returned to the main menu if there is no error. In case of an error you will get a message indicating the type of problem and then be returned to the main menu after pressing ENTER.

If things worked properly, you have created a file DEMOR_P (and DEMOR_C as well). Leave 1000 WORDS and load TI-Artist. Load the file DEMOR (in TI-Artist you can leave off only the _P extension. Use TI-Artist to draw a rectangle and a circle. Try to keep the height of these figures about the same as that of the geometric figure in the overlap area on the left of the screen. When you have finished save the picture using the filename DEMOR. While you are in TI-Artist you may wish to look at the file DEMOL which contains the other geometric figures.

CREATE FORMATTER FILE

This option is used to create the file which will print the graphics when printed by the Formatter of TI-Writer. After selecting this option you are presented with two choices.

1 FOR SINGLE WIDTH PICTURE
2 DOUBLE WIDTH PICTURE

TRY IT OUT- If you created DEMOR_P as described above you are ready to use option 2. When you select option 2, DW appears on the screen to the right of CREATE FORMATTER FILE to indicate that you chose double width picture. You also will see the following prompt.

ENTER TI-ARTIST filename
(omit L_P extension)

The program expects two files which have the same filename and the extensions L_P and R_P. You need only enter the filename. The extensions will be added for you. Change the drive number if necessary and enter the filename DEMO. Press ENTER and you will see the following prompt.

ENTER filename for TI-Writer
file to be created

You are being asked to provide the name for the DV80 file which will be created by the program. Change the drive number if necessary and enter DEMOPT2 and press ENTER. The following prompt appears

ENTER filename for TI-Writer
file to follow DSK1.DEMOPT2
(Press ENTER for none)

If the graphics which will be printed by the file you are creating represent the end of your document press ENTER. However, if text or additional graphics are to follow, you must enter the filename and the drive number that the file will be in when you print from the Formatter. Type DSK1.DEMOPT3 (or DSK2.DEMOPT3 if you have more than one drive) and press ENTER. The program begins converting the files and when finished returns you to the main menu.

Option 1 to create a Formatter file from a single width picture is very similar to option 2 described above. Only one TI-Artist file is required so the filename of the file less the _P extension is entered in response to the first prompt. The second and third prompts for filenames are handled in the same manner as described above.

TRY IT OUT- Select option 1 from the CREATE FORMATTER FILE menu. Enter DSK1.DEMO (or DSK2.DEMO if you have more than one drive) to the first prompt and press ENTER. Enter DSK1.DEMOPT4 (or DSK2.DEMOPT4 if you have more than one drive) for the second prompt and press ENTER. Press enter when the third prompt appears since no text or graphics will follow DSKn.DEMOPT4.

PRINTING FROM THE FORMATTER

Now let us see how the files created by 1000 WORDS can be printed from the Formatter of TI-Writer. In order to mix files containing text and files which will print graphics it is necessary to use the Include File Format Command (.IF filename). As a demonstration let us look at creating a document which contains text followed by the double width picture you created earlier, then some more text and finally the single width picture created earlier.

TRY IT OUT- The four sections of the document are in four separate files - DEMOPT1,DEMOPT2,DEMOPT3, and DEMOPT4. DEMOPT2 and DEMOPT4 were created earlier by 1000 WORDS. DEMOPT1 and DEMOPT3 are already on the disk. If you load DEMOPT1 into the Editor of TI-Writer, you will notice that the last line in the file is .IF DSK1.DEMOPT2 (if you have more than one drive change this to DSK2.DEMOPT2). When everything else in DEMOPT1 has been printed the Formatter is instructed to print what is in DEMOPT2 (the double width graphics).

When DEMOPT2 was created you specified that it would be followed by DSK1.DEMOPT3 (or DSK2.DEMOPT3) so .IF DSKn.DEMOPT3 (n=1 or 2) is the last line in DEMOPT2 and when printing of it is finished, DEMOPT3 will be printed. DEMOPT3 contains .IF DSKn.DEMOPT4 as its last line so the file containing the single width graphics will be printed next. We indicated that no TI-Writer file would follow DEMOPT4 so printing stops after it is printed.

To print the document load the Formatter and enter DSKn.DEMOPT1 in response to the prompt

ENTER INPUT FILENAME:

Answer the other prompts as always and you should obtain a document graphics and text intermixed.

There is a second method for printing the combination of text files and graphics files. This involves creating a file which consists of Include File commands for all of the files to be included in the document. To duplicate the document produced earlier this file would look as follows:

```
0001  .IF DSKn.DEMOPT1
0002  .IF DSKn.DEMOPT2
0003  .IF DSKn.DEMOPT3
0004  .IF DSKn.DEMOPT4
```

Use the Editor to create the above file and save it using the filename PRINTDEMO. When using this method the Include File commands are not needed in the files to be printed so DEMOPT1,DEMOPT2, and DEMOPT3 will need to be changed. You can do this easily for DEMOPT1 and DEMOPT3 by loading the files into the Editor and deleting the last line and then saving the file. To change DEMOPT2 you will need to use 1000 WORDS to create it. Load 1000 WORDS, select CREATE FORMATTER FILE from the first menu and select DOUBLE WIDTH PICTURE from the second menu. Respond to the first prompt with DSKn.DEMO.

Respond DSKn.DEMOPT2 to the second prompt. Press ENTER when prompted for the filename for the file to follow DSKn.DEMOPT2. After the program is finished you can load the Formatter and print the file DSKn.PRINTDEMO. You should get the same document that you got earlier. You can use either method. The choice is yours based on which you feel more comfortable with.

GENERAL COMMENTS

You should be aware that the DV80 files created by 1000 WORDS may be quite large. A file for a double width picture that fills the entire height of the screen will produce a file 114 sectors long. Make sure you have sufficient space on the disk before creating the file.

The files created by 1000 WORDS are in DV80 format and can be loaded into the Editor of TI-Writer. However, do not expect them to make any sense. You will not see the picture displayed. Also, do not save the file from the Editor. Doing so will alter some of the data in the file so that it will not print properly from the Formatter. If you need to change anything in a picture, you must do this using TI-Artist and then run 1000 WORDS again to create a new DV80 file.

All formatting commands of TI-writer can be used in the text files which come before or after graphics files produced by 1000 WORDS. However, in a text file which comes before a graphics file, the .NF command (No Fill) should be given before the .IF command. If a text file follows a graphics file, the margin settings are to be used.

If your printer is connected to the RS232 port, it is advisable to add the file specification .TW to the printer device name at the Formatter prompt.

The files created by 1000 WORDS alter the page length (.PL-number of lines printed per page). In a multipage document you may find that you need to change the page length again in the file which follows printing of graphics. Do this on a relative basis. That is, if you need 4 more lines on the page use the command .PL +4. If you need 5 fewer lines use .PL -5. On a page which follows graphics, you will need to reset the page length to a value for the number of lines assuming only text will be printed.

continued on page 10

TI-Writer Graphics

by Anne Dhein, IL USA

What follows is an abridgement of an article about TI-Writer Graphics taken from the TI-Writer Supplement by the Chicago Users Group. The part that I cut out explains how graphics can actually be created from TI-Artist Instances (that is, how it works) and it is quite lengthly. If any user has need of the remainder of the article, see me (Bob) at a club meeting and I will arrange a copy. I found this material to be fascinating as I had not imagined the capabilities of using TI-Writer for graphics as well as for text. This article shows how they can be used together through the formatter using .TL commands. Graphics can be done directly from the editor of TI-Writer/Funlweb but it is a long processes so this article shows how it can short-cutted. The only thing that I was not able to get to work was the 'centering' feature. If anybody works out why, let me know. The original article starts in the next paragraph.

I use TI-Artist to design a picture and an Extended Basic program (included at the end of this article) to convert the instance to a transliterate file that can be used to dump the graphics to a printer from the TI-Writer formatter.

TI-Artist is a generalised drawing program which has the capability of saving pictures on disk in a DV/80 format. The graphics data for this picture, called an INSTANCE, can be looked at with the TI-Writer. The Instance files were designed in such a way that they would be easy for someone to use in his own Extended Basic programs. Instead of using them in programs, we want to use them as the basis of a TI-Writer transliterate file that will reproduce the same picture from the TI-Writer formatter.

The program listing following this article was written by David Dhein. It was enhanced by Paul Berg of Trio+ Software so that graphics could be centered (!?), and so it could be used with a ProWriter as well as the Epsom Printer. Type it into (or see me to get a disk copy ED) Extended Basic and save the file on disk with the name CONVERT.

To use, you must first prepare your picture with the TI-Artist Program, and save it as an Instance. Load CONVERT into Extended Basic and RUN. When asked for the Instance file name, type in the name of the Instance, but omit the "_I" from the end of it. This is the name that the transliterate file will be called. You will then be asked for the type of compatible printers, and with the ProWriter and compatibles. If you wish, the program can centre your graphics on the page; otherwise, the design will start at the left margin.

Your transliterate file is now ready to be prepared. Depending on how large it is, you may have to wait quite a while for it to be converted. When it is complete, you will have a file that you can run through the formatter of TI-Writer. It will produce the same graphics that you prepared in TI-Artist, although, perhaps, smaller than you expected - one screen-width will cover about one-half of a page width. When you use the formatter, use the same file name that you typed in the conversion program. When asked for the printer device name, use .CR on the end instead of .LF which is the normal default. This works on our TI Impact printer, on the Gemini 10 and 15's and on ProWriters run on a serial interface, but if you are using a parallel interface (PIO) you may have to experiment some.

If you are a long-time user of TI-Writer, you may have already realized that when .CR is used as part of the formatter's printer parameters, a text file prepared in the editor would no longer print properly. With the carriage suppressed, each suceeding line is typed right over the top of the first. Since one of the beauties of CONVERT was to be the fact that graphics could be printed right along with text without having to run a page through the printer twice, we had to think of a way around this problem. The solution was to add a line feed character to each text line. This does have limitations: graphics line spacing and normal line spacing do not easily mix in long files, and no good way has yet been found to use a line space of 8 lines per inch in a longer file.

If your text is not too long you can ad line feed characters right from the editor. Press Control U which will give you a flashing underline cursor. Now type a Shift J after each line of text. Press Control U again to get the regular cursor back. Do not add line fed characters to the transliterate file - it is complete as it is.

If you have a long file which is to be printed 6 lines to the inch, prepare your text file on the editor as usual. You can experiment with format commands - some work perfectly with the .CR parameter, others are tricky. When the document is ready (minus the drawing), run it through the formatter, only instead of printing it on paper, print it on a disk. Use, for example, DSK1.MYFILE as the device name. Be sure to use a different file name from the one you already have if you do not want to lose the original file. The new file will print through the formatter when the .CR is used as part of the printer device name.

Now you can insert your drawing into your text document at any point using the MERGE technique described on page 73 of the TI Writer manual. If your files are too long to be merged into one file, you can break the text file into two or parts (be sure to use a different file name for each section). Use the Include File command (page 109 in the TI-Writer manual) to print all the parts, one after the other.

The worse that members of our group have done with graphics through the TI Writer has just scratched the surface of what could be done. We worked with Epsom and ProWriter but there are other printers out there too. Using the Special Character mode you could undoubtedly produce a graphics from the editor as well ... it has been done on a limited basis. Since so little is written about the Transliterate Command or the Special Character mode in the TI-Writer manual, one can only wonder what TI could have done with them if they stayed in the 99/4A computer business. If anyone out there can add to the program - for instance, making it work with a different type of printer - or can shed more light on the mysteries of TI-Writer graphics, we would like to hear from you.

```
1 GOTO 100 :: A$,C$,N$,N1$,NAME$ :: COMMANDS,EPSON,A,FD
  ,I,J,K,L,N,P,SD,X,Y :: DIM B(10):: DIM C(10):: !@P-
100 DISPLAY AT(1,4)ERASE ALL:"TI-ARTIST TO TI-WRITER"
105 DISPLAY AT(2,4):"  CONVERSION PROGRAM"
110 DISPLAY AT(5,1):"INSTANCE file name:"
120 ACCEPT AT(5,21)SIZE(8):NAME$
130 DISPLAY AT(7,3):"The file is on drive 1"
140 ACCEPT AT(7,24)SIZE(-1)VALIDATE(DIGIT):FD
150 DISPLAY AT(8,1):"Which drive for new file? 1"
160 ACCEPT AT(8,27)SIZE(-1)VALIDATE(DIGIT):SD
170 DISPLAY AT(10,1):"Select Printer: 1"
172 DISPLAY AT(11,1):"1. EPSON"
174 DISPLAY AT(12,1):"2. PROWRITER"
180 ACCEPT AT(10,17)SIZE(-1)VALIDATE(DIGIT):P
190 A$="DSK"&STR$(SD)&"."&NAME$
200 NAME$="DSK"&STR$(FD)&"."&NAME$&"
210 DISPLAY AT(18,8):"Working...."
220 OPEN #1:NAME$,INPUT
230 OPEN #2:A$,OUTPUT
240 INPUT #1:X,Y
241 DISPLAY ERASE ALL AT(1,1):"OUTPUT CENTERED? Y" ::
    ACCEPT VALIDATE("YyNn")SIZE(-1)AT(1,18):C$
250 IF X*Y>25 THEN DISPLAY AT(20,4):"This may take
    awhile." :: DISPLAY AT(21,4): "Please be patient..."
260 PRINT #2:".TL 92:10" !\=CHR$(10)=LINE FEED
270 IF P=1 THEN PRINT #2:".TL 61:27,65,8" :: PRINT #2:"
    COMMANDS
275 ! = IS 8/72 LINE SPACE  > IS 12/72 LINE SPACE
280 PRINT #2:".TL 62:27,65" :: PRINT #2&CHR$(27)&"T16"
    :: PRINT #2:"\" !PROWRITER COMMANDS
285 ! > IS 6 LINES TO INCH  2ND LINE IS CUSTOM LINE
    SPACE AT 16/144
```

continued on page 17

Transliteration and Foreign Languages

by Bob Relyea

There are a couple of very powerful tools available through TI-Writer that are almost ignored in the TI-Writer manual. One of them is called Transliteration. The sheer size of the word is enough to put most people off so they often fail to take it any further. 'Transliterate' in the TI-sense means to go beyond (hence -'trans') the usual assignment of characters (hence - 'literate') to something else of your choosing on the keyboard. In other words, virtually any character on the keyboard can be substituted for any other by giving the computer certain instructions before passing it through the Formatter. These instructions are called Transliterates. To be able to follow this article you will need a copy of the ASCII Codes in front of you. They can be found in various places such as page 145 of the TI-Writer manual and page II-137 of the Green and Gold Users Book. You can also refer to page 196 of the Extended Basic manual but this one does not include information for lower case letters, so forget it unless that is all you have available.

Let us run through an example or two. You may remember from your word processing work that the overstrike @ and the ampersand (underline) & do not print as such when passed through the formatter. They have been sort of automatically transliterated, in a sense, already by the software package. If you run across a document quite often with a lot of these 'pesky' symbols included (like Assembly Source Codes - sorry Ross) and you want to pass it through the Formatter without them performing their usual task then you either have to type two of them in succession or use transliterates. The latter choice is the one that I want to concentrate on.

The Transliterate form as given in the manual is:

```
TRANSLITERATE = .TL n1:n2,...nz
```

This means we can assign to the character, n1, any other character, n2, or any other characters, n2, n3, n4, etc. I want to look at the singular first. Suppose we want to transliterate (hereinafter called TL) so that a single @ is printed at a particular part of your text. You choose a character on the keyboard that normally does not get much use or that you are unlikely to use in the text you are currently developing. We will use | as this is not often used. We want it to 'become' @, so we look up the ASCII number for the | and the @ and find them to be 124 and 64 respectively. So we substitute 124 (|)) with 64 (@) with a TL instruction. Everytime the Formatter comes across the | it prints a @ instead. This way, only one of the @ gets printed. The TL command would look like this:

```
.TL 124: 64
```

If, for some reason you want to use the | later in the text, then you re-assign it to its original value by:

```
.TL 124:124
```

You would never want to re-assign most of the characters on the keyboard because they are in constant use but this system is very powerful for the lesser-used characters as it opens up a new world of Word Processing. There are other articles in this issue which show that this is the basis for doing graphics with the TI-Writer.

Now I want to look at assigning a character with several commands. I say commands, because the TL commands can be printer commands as well, such as backspacing. I find that feature is useful for printing those accent marks and non-English language-using symbols such as up-side-down question marks that you see in the Spanish language. I learned Spanish during the four years that I lived/taught in Perú, South America and have occasion to write in Spanish still. Notice the accent marks that I am able to put on the 'u' of Peru with the TL function. Let us consider the following sentence in Spanish:

¿En qué año nació usted?

This is a question asking a person in which year he/she was born. In Spanish you have to put a question mark at the beginning of a sentence as well as at the end. Many words carry accent marks as well. And then, there is the tilde (~) that, together with the letter n forms a separate letter in the Spanish alphabet (ñ). The way I go about getting the above sentence is to use the following TL commands:

```
.TL 35:110,8,126
.TL 43:111,8,39
.TL 33:99,8,39
.TL 124:101,8,39
```

On the Editor the above sentence in Spanish looks like the following:

```
!En qu| a#o naci+ usted?
```

I will explain the word nació. To get the accent over the 'o' I had the symbol + (ASCII 43) to replace the 'o' (ASCII 111) and then instruct the printer to backspace (8 - see page 145 of the TI-Writer manual) and then to print the apostrophe ('). Neat, huh? Try and work the others out for yourself. This enables you to 'fix up' any sentence the way you like. For an up-side-down question mark you could use a lower case 'i' which would be close enough.

Try experimenting with the TL function and see what you can come up with. Then write me an article and I will put it in the TND. o

continued from page 19

TI may have pulled the plug on production of the TI99/4A back in 1983 but TISHUG is still going strong in 1991; and why not, it is a great little computer. Come along to the meeting and enjoy the benefits of your computer's user group.

Next meeting is at Ryde Infants School, Argyle St, Ryde on Saturday 3rd August 1991.

-- ALL VERY WELCOME -- o

continued from page 8

1000 WORDS is offered to the TI community of users as a SHAREWARE product. If you use it and find that it is useful please send $10.00 to the author at the address below. I would also appreciate your comments (good or bad) about 1000 WORDS. I created the program for my own purposes originally, but I thought other TI users might find it useful and hope that you do.

Norman Rokke
Apt B204
231 Woodridge Dr
Wintersville, OH 43952 o

continued from page 4

This is, I feel, a fitting time to express my appreciation to everyone around the world who has contributed to the enjoyment that I have gained from my TI99/4A. The names are too many to list (I do not even know the names of some) who assist, directly and indirectly, with hardware and software, assist in the running and administration of the user group, News Digest, the Shop, the meetings, tutorials, BBS, etc., or just shared their time and interests with me.

Thank you from Harold Payne. o

Lurking Horror part 3

This walk through is by Scorpia, Copyright 1987.

Ok, go back down through the trap door, into the Cinderblock Tunnel, and from there past Ancient/Dead Storage to the temporary basement (make sure you have your flashlight on until you reach the basement).

Go up and get the flask that is in the Temporary Lab, then return to the basement and make your way to the elevator. What you need now is the urchin. He shows up in different places at different times, so he may not be around at the moment, and you will have to go looking for him.

When you catch up to the little darling, show him your (living) severed hand. Whoosh! That scared the heck out of him, and he takes off, dropping a pair of boltcutters as he vamooses. Grab those, and return to the elevator if you are not there now.

Drop the flask, axe and cutters. Head west to the stairs beyond the Aero basement, and this time go down to the sub-basement. Squeeze northwest through the crack into the Tomb of the Unknown Tool. There is not much here except a locked hatch in the floor. But with your master key, that is not a problem. Unlock the padlock, open the hatch and (hehe) down the hatch!

You are in the muggy Steam Tunnel, with little scurrying sounds coming out of the darkness. Ick! Rats! Go east once to the steam valve, and give it a good whack with your crowbar to loosen it up. CLANG!! Now just wait for the rats to show up. They have to be physically present in the room with you.

And here they come! Rats...LOTS of rats! They look pretty hungry, too. Hurry! Open the steam valve! HISSSS! That took care of them, all right! Now you can get on with your business without being disturbed.

Continue east to the end of the tunnel, where years of damp have weakened the south wall. Pry the bricks with the crowbar. Aha! There is something on the other side. As a matter of fact, it happens to be the elevator shaft. Unfortunately, the rest of the bricks resist your crowbar, and the restraining rod just will not budge. Oh well.

Return to the stairs up. Hmmm, looks like one rat was killed by the steam, or perhaps trampled by his frantic brethren. A close examination shows a strange tattoo of some kind on the beast; obviously, this was no ordinary rat. Good thing it is dead, and also good that you do not need to drag the corpse around with you.

Once out of the steam tunnel, go back to the elevator, making sure you pick up the padlock along the way. Pry open the doors with your invaluable crowbar. The elevator should be just above you at the lobby level. If it is not, go upstairs and summon it to the lobby, then go back down to the basement.

Stick the crowbar in the doors to keep them open (shades of Star Wars!). Now you have a good view of the elevator machinery, and one part of it looks just like a hook (which, in fact, is what it is).

Climb down into the shaft. Well, what is this? A length of greasy chain! And there is the restraining rod in the wall! Bet you know what is coming now! Pick up the chain, and wrap it around the rod, then padlock it in place. Hoist yourself out of the shaft, making sure you still hold the other end of the chain. Put that end on the hook.

We are making progress here. Clamber up the stairs to the second floor and press the up elevator button. Of course, the elevator cannot move while the doors are still forced open, so back down you go (with all this running up and down, you are doing a pretty good imitation of an elevator yourself!). Remove the crowbar and wait.

Before long, you will hear a rending screech from the shaft. Goodbye, restraining rod! Once the elevator stops moving, force the doors one more time. Pick up the flask and cutters, and enter the shaft. Wow, look at that hole! More than large enough for you to fit through with all your gear!

All right! Through the hole, into the Steam Tunnel, and follow it all the way along to the west end and the Muddy Tunnel. Pretty icky, huh? Do not worry, it gets worse! (grin) From the Muddy Tunnel, go down to the Large Chamber. What a weird place this is! Whatever could it be? Well, it happens to be some sort of incubator for ersatz urchins, many of whom are even now emerging from their slots in the wall.

A close look shows you that all of these pseudo-urchins are connected to a main wire running into a hole in the floor. That is the answer! Cut the wire with the bolt cutters! Aha! the urchins collapse, and the wire shrivels away to nothing. Down you go again into the Wet Tunnels.

This is a maze, but no need to worry! Your trusty hand will show you the way through! Bet you were wondering what that hand was really for, eh? So just head in whatever direction the hand indicates, until you reach the slime curtain. Nasty, nasty stuff, that slime. Anything that touches it will become coated with slime, and turn into slime itself. GAK! But, you have a way of dealing with this foul obstacle!

Open up the flask. A cold mist begins boiling out of it. The mist alone will not help, however if you look inside, you will see the flask actually contains liquid. Pour the liquid on the curtain. That did the trick! The entire curtain of slime freezes up and cracks, revealing a door. Now we are getting somewhere! Unlock the door with the key and enter (it will lock itself behind you, but you will have other things to worry about soon enough).

You have reached the Inner Lair! In one corner lies a large, pulsating mass, the stuff of nightmares. The hand jumps off your shoulder into the knee-deep water. Is it frightened? Or is there something down there? Reach in with your own hand. Aha! A power line.

Just as you pull it up, the door opens again, and in walks...the hacker! How he ever found you is a mystery, but he very much appreciates the danger that thing in the corner represents. While he maunders on about it, start hacking (heehee) at the power line with your axe.

Oh no! The hacker is heading for the mass...it grabs him... he is been absorbed! Too late for him...just keep at your work until the power line is severed. Now, open the metal box. Inside are all manner of connections, including one that leads to the mass, and one that has coax cable. Unplug the coax cable.

Look! Something is happening in the corner! The mass is splitting open! The hacker is coming out...but he appears to be a very different person (?) now. You cannot let him get to you! Quick, plug the power line into the socket. ZZZAAAPPP!!

That gave the mass a real jolt! But it is not dead! It is transforming itself into something....something terrible! And you have the feeling that your axe (or any other normal weapon) would be useless against this thing.

Wait! The stone is getting warm! In fact, it is becoming almost too hot to handle! When it is red-hot, throw it at the creature! Implosion! The creature is gone and the stone falls to the floor. WHEW!!!

At least it is over...or is it? You pick up the stone, and notice a small crack in it. You hear sloshing behind you, but it is only the hacker, somewhat dazed, but his old self again. Then something in the

continued on page 12

Games Information

by Robert Brown

Welcome to Games Information, Series II. Yes I am back! The last article of Series I was written in the June 1990 TND, and since that time, I have completed the HSC, and am attending University part-time and working full-time in the computer industry.

Well what has happened in those 14 months. Firstly, club membership has fallen dramatically (Actually it has levelled off due somewhat to some U.S. users joining our club. ED) including my writing partner, Stephen Judd. I have also noticed that usage on the BBS has just about fallen away, which seems a bit of a waste, as the club just bought a DS/DD Disk Controller card, thus enabling the BBS to hold more programs, mail and interesting news. Ross tells me that the programs only get updated bimonthly now as there is no new software around, and the interest is falling. A lot of members have gone to OTHER machines, for varied reasons and have left the TI field totally - my Dad is a classical example. I too use PC's at work, and quite like them (I still prefer a Macintosh though), but the TI994/A will always be an influence on my life. The TI was the first computer I ever used, way back in 1981, when we purchased one going cheap. I remember spending hours in front of the machine, just typing in a game from a magazine, and saving it to cassette. Times changed, and we upgraded to a twin floppy machine, with printer, modem etc. I still write programs in basic, with links to assembly and I thank the TI for getting me started on program writing as it is the basis for my career in the computing industry. I am now writing programs in Cobol, for our Mainframe at work. Times HAVE changed, and the TI cannot be used for that flash newsletter (What is wrong with ours? ED) or that big spreadsheet, but for the average user, the TI is an easy and quite powerful machine for whatever the average person needs.

The next question which has to be addressed is- what am I going to write about in the coming Series of articles? Hopefully someone can answer this for me?? My plans are to give you a solution for some adventures, such as Mystery Fun House and other Infocom Games. I also plan to publish some maps of the most popular adventures available on the TI. These maps outline the rooms in the adventure and helps you to move around more easily.

At this moment, I will take this opportunity, to review the last series of articles, outlining what they obtained and date and page number for quick & easy reference.

In November 1989, I did a review of Spad XIII (a flight simulator). In my opinion, this is the best flight simulator available for the TI at this very moment. If you have not brought a copy, another flight simulator to think about is 4A Flyer. Outlined below is a quick summary of the commands of 4A Flyer.

Increase Power	-> 1
Decrease Power	-> 2
Raise Flaps	-> I
Lower Flaps	-> M
Raise/Lower	
Landing gear	-> G or ";"
Enter/Exit	
Combat Mode	-> C or N
Begin Landing	-> L of F
Select Weather	-> W
Fire at Enemy	-> Fire Button, Q, V,

** NOTE **

Do not forget that pulling the joy-stick down (or pressing "X" on the keyboard) will push the nose, not the reserve!

No	Month Year	Topic	Page
1	March 1988	Cheat Mode Hitch Hikers Guide Part 1.	7 7/8
2	April 1988	Meteor Belt	4
3	May 1988	Slymoids Hitch Hiker Guide Part 2.	11 13
4	June 1988	TI Runner Hitch Hikers Guide Part 3.	13 18
5	July 1988	Tennis Sneggit	13 17
6	August 1988	TI Runner (Unlimited Miss) Star Trek Sewermania Computer War	13 13 13/18 18
7	Sept 1988	Midnite Mason Submarine Commander	13 13/18
8	Nov 1988	Spad XIII Part 1.	13/22
9	February 1989	Spad XIII Part 2. Legends 1.1	19/24 24
10	May 1989	Zork I	17/22
11	June 1989	Zork II	15/20
12	Sept 1989	Zork III	17/18
13	Dec 1989	Witness Karate Challenge	13 13
14	March 1990	Adventure Overview	15
15	May 1990	Spell Breaker Part 1.	14/16
16	June 1990	Spell Breaker Part 2.	14/21

Well that wraps up the first article in the second series of GAMES INFORMATION! As usual, if you need help, or have any questions on this article, please feel free to contract me on the BBS, username GAMES to write to me, address below.

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

COMING UP:

- * Reviews of Night Mission
- * Microsurgeon
- * Alpiner (get ready for the Ski season!)
- * Miner 2049er
- * Defender
- * Micro Pinball
- * Board Maker (lets you make your own TI Runner Screens)
- * Hints and Solutions for all the Scott Adams and Infocom adventures.
- * PLUS maps of some of the mazes.

o

continued from page 11

stone moves, the crack widens, and a small black creature flits out into the night. Perhaps it really is not over, after all.....

The Lurking Horror is copyrighted 1987 by Infocom, Inc.

This walkthru is copyrighted 1987 by Scorpia, all rights reserved.

o

```
100 ON BREAK NEXT

200 ! SAVE DSK2.STDMARK
210 CALL CLEAR
220 CALL SCREEN(6):: FOR S=0
 TO 12 :: CALL COLOR(S,16,1)
:: NEXT S
230 DISPLAY AT(4,6):"*******
************":"      *
        *":"    *     GRADE
      *":"     *
    *"
240 DISPLAY AT(8,6):"*  STAN
DARDISER  *":"      *
        *":"    *
          *"
250 DISPLAY AT(11,6):"*
           *"
260 DISPLAY AT(12,6):"******
*************"
270 DISPLAY AT(15,1):"This P
rogram calculates the Mean a
nd Standard Deviation of a s
et of marks, and then"
280 DISPLAY AT(18,1):"standa
rdises them to a mean and  s
tandard  deviation of your c
hoice."

290 !Default file name
300 DIM SN$(80),M$(80),Y2(80
),M(80)
310 DISPLAY AT(24,1):"PRESS
ANY KEY TO CONTINUE..."
320 CALL KEY(0,K,S):: IF S=0
 THEN 320
330 DISPLAY AT(1,1)ERASE ALL
:"  GRADE STANDARDISING MENU
": :"1. Create NEW File": :"
2. Standardise EXISTING Data
": :"3. Load OLD File"
340 DISPLAY AT(9,1)BEEP:"4.
Edit/View EXISTING Data": :"
5. Save EXISTING Data":
:"6. Print a Hard Copy of
      Existing Data"
350 DISPLAY AT(16,1)BEEP:"7.
 Print EXISTING Data on
 Screen": :"8. 'Quick' Stand
arisation": :"9. End"
360 DISPLAY AT(24,1):"PRESS
NUMBER OF CHOICE"
370 CALL KEY(0,KK,S):: IF KK
<49 OR KK>57 THEN 370 ELSE K
K=KK-48 :: CALL CLEAR
380 ON KK GOTO 460,3110,1400
,2250,1150,1950,1730,2470,40
0

390 !
400 DISPLAY AT(10,2):"     DO
 YOU WISH TO SAVE
   YOUR DATA?"
410 DISPLAY AT(15,2):"
PRESS S TO SAVE,
  ANY OTHER KEY TO END"
420 CALL KEY(5,K,S):: IF S=0
 THEN 420
430 IF K=83 OR K=115 THEN 11
80 ELSE 440
440 END

450 !
460 !CREATE NEW FILE
470 !
480 FOR I=1 TO N ! PURGE EXI
STING ARRAY CONTENTS FIRST
490 M(I),N=0 :: SN$(I)="" ::
 SQL=0 :: T=0
500 NEXT I
510 INPUT "FILE NAME? ":FN$
520 CALL CLEAR
530 INPUT "HOW MANY MARKS DO
 YOU WISH TO ENTER? ":N
540 CALL CLEAR
550 FOR I=1 TO N ! EDIT EXIS
TING ENTRY POINT
560 DISPLAY AT(1,1):"ENTRY#"
;I;"OF";N;"ENTRIES"
570 DISPLAY AT(6,1):"Student
's                      Name
? ";SN$(I)
580 DISPLAY AT(9,4):"Mark? "
;M$(I)
590 ON WARNING NEXT
600 ACCEPT AT(7,11)VALIDATE(
UALPHA)BEEP SIZE(-15):SN$(I)
610 ON WARNING NEXT
620 ACCEPT AT(9,11)VALIDATE(
NUMERIC)SIZE(-5)BEEP:M(I)
630 DISPLAY AT(15,1):"PRESS
E to Edit,  any other key to
  continue."
640 CALL KEY(5,K,S):: IF S=0
 THEN 640 ELSE IF K=69 THEN
DISPLAY AT(7,11):SN$(I):: GO
TO 600
650 IF K=13 THEN 660
660 CALL CLEAR
670 T=T+M(I)
680 SQL=M(I)*M(I)+SQL
690 AVE=INT(T/I)
700 NEXT I
710 CALL CLEAR
720 DISPLAY AT(1,3):"TOTAL M
ARKS  = ";T
730 DISPLAY AT(3,3):"NO. OF
MARKS = ";I-1
740 DISPLAY AT(5,3):"AVERAGE
      = ";AVE

750 !
760 ! STDEVIATION
770 !
780 Z1=SQL
790 Z2=T*T/(I-1)
800 Z3=I-2
810 STD=(Z1-Z2)/Z3
820 NSTD=SQR(STD)
830 DISPLAY AT(7,3):"STDEV
      = ";INT(SQR(STD))
840 DISPLAY AT(17,3):"PRESS
ANY KEY TO CONTINUE"
850 CALL KEY(0,K,S):: IF S=0
 THEN 850
860 CALL CLEAR :: PRINT :: G
OTO 870
870 DISPLAY AT(5,2):"   WE W
ILL NOW STANDARDISE
 YOUR MARKS!"
880 FOR J=1 TO 350 :: NEXT J
890 DISPLAY AT(15,2):"ENTER
THE DESIRED MEAN - ":ME$
900 DISPLAY AT(18,2):"ENTER
THE DESIRED STDEV- ":D$
910 ACCEPT AT(15,26)VALIDATE
(NUMERIC)SIZE(-3):ME
920 ACCEPT AT(18,26)VALIDATE
(NUMERIC)SIZE(-3)BEEP:D
930 CALL CLEAR
940 L=3
950 FOR I=1 TO N :: L=L+1
960 Y=M(I)-AVE
970 Y1=Y/NSTD
980 Y2(I)=Y1*D+ME
990 Y2(I)=INT((Y2(I)*100)/10
0)
1000 DISPLAY AT(1,2):"NAME
          OLD MARK STD"
1010 DISPLAY AT(L,1):SN$(I)
1020 DISPLAY AT(L,19):M(I)
1030 DISPLAY AT(L,25):Y2(I)
1040 IF I/18=INT(I/18)THEN G
OTO 1050 ELSE 1090
1050 DISPLAY AT(23,1):"PRESS
 'ENTER' TO CONTINUE..."
1060 L=3
1070 CALL KEY(0,K,S):: IF S=
0 THEN 1070
1080 CALL CLEAR
1090 NEXT I
1100 DISPLAY AT(23,4):"PRESS
 ANY KEY TO RETURN"
1110 DISPLAY AT(24,12):"TO M
ENU"
1120 CALL KEY(0,K,S):: IF S=
0 THEN 1120
1130 GOTO 330

1140 !
1150 ! SAVE ROUTINE
1160 !
1170 CALL CLEAR
1180 IF N>0 THEN 1220
1190 IF N=0 THEN DISPLAY AT(
10,1):"          * WARNING *
               NO DATA PRESENT
" :: DISPLAY AT(15,1):"PRESS
 M TO RETURN TO MENU,   ANY
OTHER KEY TO CONTINUE"
1200 CALL KEY(0,K,S):: IF S=
0 THEN 1200 ELSE IF K=77 THE
N 1210 ELSE 1220
1210 CALL CLEAR :: GOTO 330
1220 CALL CLEAR :: DISPLAY A
T(1,1):"SAVE ROUTINE": :"ENT
ER FILE NAME ->";FN$ ::
ACCEPT AT(3,19)BEEP SIZE(-10
)VALIDATE(UALPHA,NUMERIC," "
):FN$
1230 ON ERROR 1320
1240 OPEN #1:"DSK2."&FN$,INP
UT ,INTERNAL,FIXED 80
1250 DISPLAY AT(10,1):"  OVE
RWRITE EXISTING FILE?
      (Y/N)"
1260 CALL KEY(0,K,S):: IF S=
0 THEN 1260 ELSE IF K=89 THE
N 1270 ELSE 1280
1270 CLOSE #1 :: GOTO 1320
1280 CALL CLEAR
1290 CLOSE #1
1300 DISPLAY AT(1,1):"SAVE R
OUTINE": :"ENTER FILE NAME -
>";FN$ :: ACCEPT AT(3,19)BEE
P SIZE(-10)VALIDATE(UALPHA,N
UMERIC):FN$
1310 ON ERROR 3620
1320 OPEN #2:"DSK2."&FN$,OUT
PUT,INTERNAL,FIXED 80
1330 FOR I=1 TO N
1340 DISPLAY AT(24,1):"SAVIN
G";I
1350 PRINT #2:SN$(I),M(I),Y2
(I),N,ME,D,FN$
1360 NEXT I
1370 CLOSE #2
1380 GOTO 330

1390 !
1400 ! LOAD ROUTINE
1410 !
1420 FOR I=1 TO N :: !PURGE
EXISTING ARRAY CONTENTS FIRS
T
1430 SN$(I)="" :: RN$="" ::
M(I)=0 :: Y2(I)=0 :: ME=0 ::
 D=0 :: SQL=0 :: AVE=0 :: T=
0
1440 NEXT I
1450 I=0
1460 ON ERROR 3650
1470 DISPLAY AT(1,1):"LOAD R
OUTINE": :"ENTER FILE NAME -
>";FN$ :: ACCEPT AT(3,19)BEE
P SIZE(-10)VALIDATE(UALPHA,N
UMERIC):FN$
1480 OPEN #1:"DSK2."&FN$,INP
UT ,INTERNAL,FIXED 80
1490 I=I+1 :: N=I :: DISPLAY
 AT(24,1):"LOADING";I
1500 INPUT #1:SN$(I),M(I),Y2
(I),N,ME,D,FN$
1510 IF EOF(1)THEN 1520 ELSE
 1490
```

```
1520 CLOSE #1
1530 SQL=0
1540 FOR I=1 TO N
1550 CALL CLEAR
1560 T=T+M(I)
1570 SQL=M(I)*M(I)+SQL
1580 AVE=INT(T/I)
1590 NEXT I
1600 Z1=SQL
1610 Z2=T*T/(I-1)
1620 Z3=I-2
1630 STD=(Z1-Z2)/Z3
1640 NSTD=SQR(STD)
1650 FOR I=1 TO N
1660 Y=M(I)-AVE
1670 Y1=Y/NSTD
1680 Y2(I)=Y1*D+ME
1690 Y2(I)=INT((Y2(I)*100)/1
00)
1700 NEXT I
1710 GOTO 330

1720 !
1730 ! ROUTINE TO PRINT DATA
 ON SCREEN
1740 !
1750 CALL CLEAR
1760 IF N=0 THEN DISPLAY AT(
10,1)BEEP:"* NO DATA IN MEMO
RY *": : :"PRESS ANY KEY FOR
 MENU" ELSE 1780
1770 CALL KEY(0,K,S):: IF S=
0 THEN 1770 ELSE 330
1780 CALL CLEAR
1790 L=2
1800 FOR I=1 TO N :: L=L+1
1810 DISPLAY AT(1,2):"NAME
         OLD MARK STD"
1820 DISPLAY AT(L,2):SN$(I)
1830 DISPLAY AT(L,19):M(I)
1840 IF Y2(I)=0 THEN DISPLAY
 AT(L,25)SIZE(1):"" :: GOTO
1860
1850 DISPLAY AT(L,25):Y2(I)
1860 IF I/20=INT(I/20)THEN G
OTO 1870 ELSE 1910
1870 INPUT "PRESS 'ENTER' TO
 CONTINUE":B$
1880 L=2
1890 IF B$="ENTER" THEN 1900
1900 CALL CLEAR
1910 NEXT I
1920 DISPLAY AT(22,2):"    PR
ESS ANY KEY TO RETURN
      TO MENU"
1930 CALL KEY(0,K,S):: IF S=
0 THEN 1930
1940 GOTO 330

1950 !
1960 ! ROUTINE TO PRINT A HA
RD COPY
1970 !
1980 CALL CLEAR
1990 IF N=0 THEN DISPLAY AT(
10,1)BEEP:"* NO DATA IN MEMO
RY *": : :"PRESS ANY KEY FOR
 MENU" ELSE 2010
2000 CALL KEY(0,K,S):: IF S=
0 THEN 2000 ELSE 330
2010 DISPLAY AT(12,5):"MAKE
SURE YOUR PRINTER"
2020 DISPLAY AT(14,13):"IS O
N"
2030 DISPLAY AT(20,1):"
PRESS P TO PROCEED
M TO RETURN TO MENU"
2040 CALL KEY(0,K,S):: IF S=
0 THEN 2040
2050 IF K=77 OR K=109 THEN 3
30
2060 IF K=112 OR K=80 THEN 2
070
2070 DISPLAY AT(10,1)ERASE A
LL:"     PRINTING IN PROGRES
S"
2080 OPEN #3:"RS232.BA=4800.
DA=8.PA=N"
2090 PRINT
2100 PRINT #3:TAB(7);"FILE N
AME: ";TAB(19);FN$
2110 PRINT #3:"        NAME
                       OLD M
ARK             STD MARK"
2120 PRINT #3:"-------------
----------------------------
------------------------"
2130 FOR I=1 TO N
2140 PRINT #3:TAB(7);SN$(I);
TAB(39);M(I);TAB(59);Y2(I)
2150 NEXT I
2160 PRINT #3:"-------------
----------------------------
------------------------"
2170 PRINT #3:TAB(7);"NUMBER
 OF STUDENTS =";TAB(27);I-1;
TAB(33);"MEAN";TAB(39);AVE;T
AB(59);ME
2180 PRINT #3:TAB(34);"STD";
TAB(39);INT(SQR(STD));TAB(59
);D
2190 PRINT #3:TAB(32);"TOTAL
";TAB(39);T
2200 CLOSE #3
2210 GOTO 330

2220 !
2230 ! EDIT/VIEW ROUTINE
2240 !
2250 AVE=0 :: SQL=0 :: T=0 :
: NSTD=0
2260 I=1
2270 IF I>N THEN I=1
2280 DISPLAY AT(1,1):"Entry#
";I;"of";N;"Entries"
2290 DISPLAY AT(6,1):"Studen
t's                     Nam
e?  ";SN$(I)
2300 DISPLAY AT(9,4):"Mark?
";M(I)
2310 IF KK=4 THEN DISPLAY AT
(23,1)BEEP:"Press E to Edit,
 SPACE for  next entry or M
for Menu" ELSE 2350
2320 CALL KEY(0,K,S):: IF S=
0 THEN 2320 ELSE IF K=69 THE
N DISPLAY AT(23,1): : :
:: GOTO 2350 ELSE IF K=32 TH
EN 2330 ELSE IF K=77 THEN 23
80
2330 I=I+1 :: IF I>N THEN I=
1
2340 GOTO 2270
2350 ACCEPT AT(7,11)VALIDATE
(UALPHA)BEEP SIZE(-15):SN$(I
)
2360 ACCEPT AT(9,11)VALIDATE
(NUMERIC)SIZE(-5)BEEP:M(I)
2370 I=I+1 :: GOTO 2270
2380 FOR I=1 TO N
2390 Y2(I)=0
2400 NEXT I
2410 CALL CLEAR
2420 DISPLAY AT(10,10):"* WA
RNING *"
2430 DISPLAY AT(12,10):"
MARKS WILL HAVE TO BE
   RE-STANDARDISED"
2440 DISPLAY AT(16,10):"
RETURN TO MENU AND
   USE OPTION(2)"
2450 DISPLAY AT(21,10):"
PRESS ANY KEY TO RETURN
        TO MENU"
2460 CALL KEY(0,K,S):: IF S=
0 THEN 2460 ELSE 330
2470 !
2480 ! ROUTINE FOR QUICK
       STANDARDISATION
2490 !
2500 IF N>0 THEN 2510 ELSE 2
590
2510 DISPLAY AT(10,1):"
   * WARNING *"
2520 DISPLAY AT(13,1):"  EXI
STING DATA WILL ERASED"
2530 DISPLAY AT(15,1):"
 PRESS M FOR MENU        ANY
 OTHER KEY  TO PROCEED"
:: CALL KEY(0,K,S):: IF S=0
THEN 2530
2540 IF K=77 OR K=109 THEN 3
30 ELSE 2550
2550 CALL CLEAR
2560 FOR I=1 TO N
2570 SN$(I)="" :: FN$="" ::
M(I)=0 :: Y2(I)=0 :: ME=0 ::
 D=0 :: T=0 :: AVE=0 ::
SQL=0
2580 NEXT I
2590 INPUT "FILE NAME? ":FN$
2600 CALL CLEAR
2610 INPUT "HOW MANY MARKS D
O YOU WISH TO ENTER? ":N
2620 FOR I=1 TO N
2630 CALL CLEAR
2640 PRINT :: PRINT
2650 PRINT "ENTER MARK ";I
2660 INPUT M(I)
2670 T=T+M(I)
2680 SQL=M(I)*M(I)+SQL
2690 AVE=INT(T/I)
2700 NEXT I
2710 CALL CLEAR
2720 DISPLAY AT(5,4):"TOTAL
MARKS=  ";T
2730 DISPLAY AT(7,4):"NO. OF
 MARKS= ";I-1
2740 DISPLAY AT(9,4):"AVERAG
E =     ";AVE
2750 Z1=SQL
2760 Z2=T*T/(I-1)
2770 Z3=I-2
2780 STD=(Z1-Z2)/Z3
2790 DISPLAY AT(11,4):"STAND
. DEV. = ";INT(SQR(STD))
2800 NSTD=SQR(STD)
2810 DISPLAY AT(24,1):"PRESS
 ANY KEY TO CONTINUE..."
2820 CALL KEY(0,K,S):: IF S=
0 THEN 2820
2830 CALL CLEAR
2840 DISPLAY AT(13,4):"WE WI
LL NOW STANDARDISE"
2850 DISPLAY AT(14,10):"YOUR
 MARKS!"
2860 FOR K=1 TO 300 :: NEXT
K
2870 CALL CLEAR
2880 DISPLAY AT(13,2):"ENTER
 THE DESIRED MEAN ":ME$
2890 DISPLAY AT(15,2):"ENTER
 THE DESIRED STD- ":D$
2900 ACCEPT AT(13,26)VALIDAT
E(NUMERIC)SIZE(-3):ME
2910 ACCEPT AT(15,26)VALIDAT
E(NUMERIC)SIZE(-3)BEEP:D
2920 CALL CLEAR
2930 L=2
2940 FOR I=1 TO N :: L=L+1
2950 Y=M(I)-AVE
2960 Y1=Y/NSTD
2970 Y2(I)=Y1*D+ME
2980 Y2(I)=INT((Y2(I)*100)/1
00)
2990 DISPLAY AT(1,2):"      O
LD MARK     STD"
3000 DISPLAY AT(L,9):M(I)
3010 DISPLAY AT(L,19):Y2(I)
3020 IF I/20=INT(I/20)THEN G
OTO 3030 ELSE 3070
```

```
3030 DISPLAY AT(24,1):"PRESS
 'ENTER' TO CONTINUE..."
3040 L=2
3050 CALL KEY(0,K,S):: IF S=
0 THEN 3050
3060 CALL CLEAR
3070 NEXT I
3080 DISPLAY AT(24,1):"PRESS
 ANY KEY TO CONTINUE..."
3090 CALL KEY(0,K,S):: IF S=
0 THEN 3600
3100 GOTO 330

3110 !
3120 ! STANDARDISATION (DIRE
CT FROM NUMBER 2)
3130 !
3140 AVE=0 :: T=0 :: SQL=0
3150 FOR I=1 TO N
3160 T=T+M(I)
3170 SQL=M(I)*M(I)+SQL
3180 AVE=INT(T/I)
3190 NEXT I
3200 CALL CLEAR
3210 PRINT :: PRINT
3220 DISPLAY AT(5,4):"TOTAL
MARKS=  ";T
3230 DISPLAY AT(7,4):"NO. OF
 MARKS= ";I-1
3240 DISPLAY AT(9,4):"AVERAG
E =     ";AVE
3250 Z1=SQL
3260 Z2=T*T/(I-1)
3270 Z3=I-2
3280 STD=(Z1-Z2)/Z3
3290 DISPLAY AT(11,4):"STAND
. DEV. = ";INT(SQR(STD))
3300 NSTD=SQR(STD)
3310 DISPLAY AT(24,1):"PRESS
 ANY KEY TO CONTINUE..."
3320 CALL KEY(0,K,S):: IF S=
0 THEN 3320
3330 CALL CLEAR
3340 DISPLAY AT(13,4):"WE WI
LL NOW STANDARDISE"
3350 DISPLAY AT(14,10):"YOUR
 MARKS!"
3360 FOR K=1 TO 200 :: NEXT
K
3370 CALL CLEAR
3380 DISPLAY AT(13,2):"ENTER
 THE DESIRED MEAN ";ME$
3390 DISPLAY AT(15,2):"ENTER
 THE DESIRED STD- ":D$
3400 ACCEPT AT(13,26)VALIDAT
E(NUMERIC)SIZE(-3):ME
3410 ACCEPT AT(15,26)VALIDAT
E(NUMERIC)SIZE(-3)BEEP:D
3420 CALL CLEAR
3430 L=3
3440 FOR I=1 TO N :: L=L+1
3450 Y=M(I)-AVE
3460 Y1=Y/NSTD
3470 Y2(I)=Y1*D+ME
3480 Y2(I)=INT((Y2(I)*100)/1
00)
3490 DISPLAY AT(1,2):" NAME
          OLD MARK STD"
3500 DISPLAY AT(L,2):SN$(I)
3510 DISPLAY AT(L,19):M(I)
3520 DISPLAY AT(L,25):Y2(I)
3530 IF I/19=INT(I/19)THEN G
OTO 3540 ELSE 3580
3540 DISPLAY AT(24,1):"PRESS
 'ENTER' TO CONTINUE..."
3550 L=3
3560 CALL KEY(0,K,S):: IF S=
0 THEN 3560
3570 CALL CLEAR
3580 NEXT I
3590 DISPLAY AT(24,1):"PRESS
 ANY KEY TO CONTINUE..."
3600 CALL KEY(0,K,S):: IF S=
0 THEN 3600
3610 GOTO 330
3620 CALL CLEAR
3630 CLOSE #2
3640 RETURN 1300
3650 CALL CLEAR
3660 DISPLAY AT(2,1):"ERROR
IN ATTEMPTING TO LOAD"
3670 DISPLAY AT(5,1):"PLEASE
 CHECK THE FOLLOWING
     LIST"
3680 DISPLAY AT(11,1):"1. No
 disk in drive"
3690 DISPLAY AT(13,1):"2. Di
sk in incorrect drive"
3700 DISPLAY AT(15,1):"3. In
valid file name"
3710 DISPLAY AT(19,1):"PRESS
 ANY KEY TO CONTINUE..." ::
CALL KEY(0,K,S):: IF S=0 THE
N 3710
3720 FOR I=1 TO N :: FN$=""
:: NEXT I :: I=0
3730 CALL CLEAR
3740 DISPLAY AT(1,1):"LOAD R
OUTINE": :"ENTER FILE NAME -
>";FN$ :: ACCEPT AT(3,19)BEE
P SIZE(-10)VALIDATE(UALPHA,N
UMERIC):FN$
3750 RETURN
```

o

continued from page 21

```
100 ! SAVE DSK1.LOAD
110 CALL CLEAR
120 CALL INIT
130 CALL LOAD("DSK1.O")
140 CALL LINK("KEYIN",10,6,KEY)
150 DISPLAY AT(14,6):KEY
160 GOTO 140

* S=S  O=O
       IDT   'KEYIN'
       DEF   KEYIN
XMLLNK EQU   >2018
KSCAN  EQU   >201C
NUMREF EQU   >200C
NUMASG EQU   >2008
SOUND  EQU   >8400
FAC    EQU   >834A
CFI    EQU   >12B8
CIF    EQU   >20
VSBW   EQU   >2020

SAVRTN BSS   2
WS     BSS   >20                32
ROW    BSS   2
COL    BSS   2
SCRPOS BSS   2
B5     BYTE  5
FF     BYTE  >FF
       EVEN

KEYIN  MOV   R11,@SAVRTN
       LWPI  WS

       CLR   R0
       LI    R1,1
       BLWP  @NUMREF
       BLWP  @XMLLNK
       DATA  CFI
       DEC   @FAC
       MOV   @FAC,@ROW

       INC   R1
       BLWP  @NUMREF
       BLWP  @XMLLNK
       DATA  CFI
       DEC   @FAC
       MOV   @FAC,@COL

       MOV   2@ROW,R3
       SLA   R3,5         Mpy by 32

       A     @COL,R3
       MOV   R3,@SCRPOS

       MOVB  @B5,@>8374

SCAN   BLWP  @KSCAN    Result in >8375

       MOV   @SCRPOS,R0
       MOVB  @>8375,R1
       AI    R1,>6000
       BLWP  @VSBW

       CB    @>8375,@FF
       JEQ   SCAN

       MOVB  @>8375,R2
       SRL   R2,8
       MOV   R2,@FAC
       BLWP  @XMLLNK
       DATA  CIF

       CLR   R0
       LI    R1,3
       BLWP  @NUMASG

       CLR   R0
       MOVB  R0,@>837C
       LWPI  >83E0
       MOV   @SAVRTN,R11
       B     *R11
       END
```

o

continued from page 5

Playing the Game

TRIS for the TI99/4A only comes as a module (a disk version is available for the Geneve).

Before playing the game you must input:

1) The Level (0 to 9) that you wish to start playing.

2) The Height (0 to 9) that the blocks fall.

3) Shape Preview (Y/N) prior to dropping.

4) Alternate Control Keys (Y/N) allows use of J, L, K and SPACE instead of S, D SPACE and ENTER. The documentation explains the key assignments of your choice.

As the blocks fall, you use the appropriate keys to move left or right, rotate and finally drop the shape in order to fill the current row. Fast thinking and dextrous fingers will help you move up in levels.

This is a top game and demonstrates once again that you do not necessarily need complexity to make an intriguing game. Highly recommended . o

Reviewing Software

by Lou Amadio

One of the advantages of reviewing software is that you get to sample some interesting programs without actually having to pay for them. You cannot keep the programs, unfortunately, but at least you are in a better position to decide on what is worth buying.

How would you like the opportunity to try some software free of charge? I am talking about software that is currently in our club library. Rolf, our software librarian, is in the process of categorising the many hundreds of programs in the library and he needs your help to do this.

All you have to do is pick up some library disks from Rolf at the monthly meeting, run the programs and categorise the individual files on the disks and return them the following month, hopefully ready to tackle another lot of disks.

Library disks could contain a number of files which run in any of the supported languages for the TI99/4A. Some programs on the disks will have associated data files.

This is what we need to know about the files on the club disks:

1) Determine the operating language and run the programs to ensure that they operate correctly.

2) Indicate whether 32K memory expansion is necessary.

3) Record what peripheral devices are required - eg Speech Synthesizer, Terminal Emulator module, etc.

4) If a printer is supported does the program interface with a serial or parallel system.

5) Rate the program from 0 to 10. Try to rate the program on how well it achieves its intended goal. Very briefly indicate your impressions of the program.

6) Indicate a category that best describes the program. Use the following as a guide:

GAMES - Adventure Arcade Simulation
Strategy Chance Word

LANGUAGES - BASIC Extended BASIC GPL
Assembler LOGO Forth
UCSD Pascal XDP XXB
Pilot C99 Gee

EDUCATION - Pre-school Primary High School

UTILITIES - Data Base Spreadsheet Word Processing
Finance Disk Utility Cassette Utility
Terminal Emulator Miscellaneous

MUSIC/SPEECH - Playing Composing Examples
Analysis TE II

GRAPHICS/PICTURES - Graphx TI Artist Paint'n Print
Picasso Sketch It Draw a Bit
Pix Pro RLE YAPP
GIF Page Pro Mac Flix
XHi Myart Joypaint
TIPS Other

TEXT - Documentation Instructions Articles

COMMERCIAL - Indicated by copyright notice.

This is your opportunity to try out some new software and, at the same time, help Rolf get the software library in order.

1000 Words Exit Bug

by Rolf Schreiber

While I was reviewing the 1000 Words TI-Writer utility, I had trouble quitting the program after loading it from the RAMdisk menu. The same problem would crop up if I loaded it through the 'BOOT' menu loader, or the Funnelweb loader. The program just seemed to get lost whenever I selected the option to exit. I knew from previous experience that this is a common problem with programs which are loaded and run under conditions which differ from standard or expected conditions, ie the conditions which exist when memory image files are loaded through option 5 of the Editor/Assembler module. If the program exited by returned to the calling program, when it had been loaded in an environment which differed from the E/A module, then it would most certainly get lost; ie the code to search for would be 0460 0070.

I used Disk Utilities to search for any occurrences of this hex string and found it in the 11th sector of the file 'WORDS', at bytes >EA to >ED (234-237 in decimal). To correct the problem I needed the program to exit by returning to the title screen, ie the necessary patch was 0420 0000. I made the substitution and wrote the changes back to the disk. When I tried exiting the amended 1000 Words program, it did so without a hitch, so I knew that I had fixed the problem.

continued from page 20

Assembler Listing

```
*       Interrupt demonstration
*
* Called by ROM interrupt routine
* with GPL workspace at >83E0
* R11 = return address
* R12 = address of this routine
* R13 = >9800 GROM read data
* R14 = system flags
* R15 = >8C02 VDP write address
*
* This program is self relocatable
*
USRINT MOV  @T(R12),R3    get tick counter
       CLR  R2            set R2,R3 for divide
       INC  R3            count ticks
       CI   R3,60         1.2 seconds?
       JL   TOK           jump no
       CLR  R3            wrap from 60 to 0
       SWPB @C(R12)       and swap screen chars
TOK    MOV  R3,@T(R12)    save new tick counter
       DIV  @F(R12),R2    divide into 4 quadrants
       DEC  R2            R2 has quadrant 0 to 3
       JLT  Q0            jump if quad 0
       JEQ  Q1            jump if quad 1
       DEC  R2
       JEQ  Q2            jump if quad 2
Q3     LI   R0,551        left side going up
       SLA  R3,5          Q3=551-32*rem
       JMP  QS
Q2     LI   R0,598        bottom going left
QS     S    R3,R0         Q2=598-rem
       JMP  OUT
Q1     LI   R0,119        right side going down
       SLA  R3,5          Q1=119+32*rem
       JMP  QA
Q0     LI   R0,72         top going left
QA     A    R3,R0         Q0=72+rem
OUT    ORI  R0,>4000      VDP write bit
       SWPB R0
       MOVB R0,*R15       set VDP addr for write
       SWPB R0
       MOVB R0,*R15
       NOP
       MOVB @C(R12),@>8C00 char to VDP
       B    *R11          return to ROM routine
F      DATA 15            # chars per quadrant
T      DATA 0             tick counter
C      DATA >8A80         '* ' chars for display
       END
```

TI-Bits Number 8

by Jim Swedlow, CA USA

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

FAIRWARE REVIEW: DISK UTILITIES
by John Birdwell

You may have a favourite disk editor - one that you know and love (?) - one that meets your needs. Mine has been Miller Graphics' Advanced Diagnostics. At least until now. John Birdwell's DISK UTILITIES has jumped to the top of my list. It is easily the best sector editor I have used.

It is what a sector editor should be. You can dump a file to your printer in HEX and ASCII. DISK UTILITIES will follow the file on the disk even if it is fractured. The file dump is like Disk+Aid with HEX on the left and ASCII on the right. The print out can be in condensed print. You can also print a sector or a group of sectors.

You can compare two files or disks. Any sectors that do not match will be dumped to your printer. It can also give you a detailed file report.

DISK UTILITIES supports a string search. You can search a disk, any part of the disk or within a file. The string can be in HEX or ASCII.

The sector editor gives you a full screen editor. The various controls are easy to remember. Pressing CTRL H and CTRL A, for example, switches the screen display between HEX and ASCII. CTRL W will write the sector back to disk. Unlike Advanced Diagnostics, DISK UTILITIES keeps track of the current sector for writing sectors. You can, however, write to any sector on any disk.

Another nice feature is the Disk Report. This prints a disk catalog with two new features. First, the catalog includes each file's sector numbers. Invaluable if have it before you blow a disk directory. Also, DISK UTILITIES hides a short file description in the file header and prints it out as part of the catalog.

This program is a sector editor only. It does not have the ability to look into your 4A's memory that Disk+Aid has nor the extensive documentation and diagnostic features of Advanced Diagnostics. But it does have all the features one needs in a disk editor.

There is more, but this should give you an idea of what DISK UTILITIES can do. Without doubt, it warrants your serious consideration. DISK UTILITIES should be in our library by the time you read this. If you like it, send John the $10 he asks for. It is well worth the price.

John's address is:

John Birdwell
7052 Springhill Circle
Eden Praire, MN 55344

QUOTE OF THE MONTH

"It is better to know some of the questions than all of the answers."

---James Thurber (1894-1961)

CUSTOMIZING FUNNELWRITER

It has been said that FUNNELWRITER may be the most significant program written for the TI. One could argue this point but not easily dismiss it.

I have been working on getting FUNNELWRITER to support the utilities that I normally use. This is the first of a series on customizing FUNNELWRITER.

The first thing I wanted to do was to enable FUNNELWRITER to load FAST-TERM. When you press 5 on the main menu, one of the options that comes up for number 2 is MODEM. I could not find, however, what file name was needed. After a bit of searching (using DISK UTILITIES), I found it: MD.

FAST-TERM comes with two files named UTIL1 and UTIL2. You must rename them (using DM1000) to MD and ME and then copy the files to your FUNNELWRITER disk. Change the names before copying because there already is a UTIL1 on the FUNNELWRITER disk and you do not want to overwrite it.

When you switch item 2 to DISK EDIT, FUNNELWRITER loads Disk Patch, or Disco. This is a bare bones disk sector editor. I wanted to load DISK UTILITIES so I removed Disco from my FUNNELWRITER disk, renamed the two DISK UTILITIES Files (UTIL1 and UTIL2) to DP and DQ and copied them.

I did all of this renaming and copying on back-up copies. My originals are safe and unmodified. Always keep a master copy of important programs.

The next subject is customizing the user list.

ON GETTING FAIRWARE

On a shelf high above my computer is a large disk box with my masters. One of my prized possessions in that box is an original DISK MANAGER 1000 (V3.3) from the Ottawa TI Users Group. I have a copy from our library but somehow it is not the same.

Our library has many fine fairware programs and I (among others) have often urged you to support fairware authors. One way is to request a program directly from the author (and then send some support).

Just a thought.

Enjoy. o

continued from page 9

```
290 FOR K=1 TO Y
300 FOR L=1 TO X
310 IF P=1 THEN INPUT #1:C(7),C(6),C(5),C(4),C(3),C(2),
    C(1),C(0):: GOTO 330
320 INPUT #1:C(0),C(1),C(2),C(3),C(4),C(5),C(6),C(7)
330 FOR I=7 TO 0 STEP -1
340 A=C(I)
350 FOR J=7 TO 0 STEP -1
360 IF 2^J>A THEN 390
370 A=A-2^J
380 B(J)=B(J)+2^I
390 NEXT J
400 NEXT I
410 A$=STR$(B(0))
420 B(0)=0
430 FOR I=1 TO 7
440 A$=STR$(B(I))&","&A$
450 B(I)=0
460 NEXT I
470 IF P=1 THEN PRINT #2:".TL "&SEG$(STR$(127-L),1,3)&
    ":27,75,8,0,"&A$ :: GOTO 90
480 PRINT #2:".TL "&SEG$(STR$(127-L),1,3)&":27,83,48,
    48,48,56,"&A$
490 NEXT L
491 IF C$="Yy" THEN PRINT #2:".CE"
500 N$="" :: FOR N=1 TO X :: N$=N$&CHR$(127-N):: NEXT N
    :: PRINT #2:N$&"\"
510 NEXT K
520 FOR N=1 TO X :: N$=".TL " :: N1$=SEG$(STR$(127-N),1
   ,3):: N$=N$&N1$&":"&SN1: PRINT #2:N$ :: NEXT N
530 PRINT #2:".TL 92.92"
540 IF P=1 THEN PRINT #2:".TL 61.61"
550 PRINT #2:">"
560 PRINT #2:".TL 62.62"
570 CLOSE #1
580 CLOSE #2
585 !@P+
590 END
```

o

XB tips Number 9

by Jim Swedlow, CA USA

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

PRODUCT REVIEW - NEATLIST

NEATLIST will take any program in memory and list it to any legal device (printer, disk, etc.) with the same speed as the LIST command. It has, however, several advantages:

--Each line of the listed program can be up to 255 characters wide. You can, therefore, use your printers powers (condensed print, expanded print, etc.).

--You can send function codes to your printer enabling you to access its features.

--NEATLIST can print out a list of variables used in the program and, if you opt, a list of line numbers where each variable is used.

--It is an assembly language program that loads thru XB.

The list is nicely designed. Consider this program:

```
10 CALL CLEAR :: PRINT :: DIM A(5)
20 FOR I=1 TO 5 :: A(I)=I :: NEXT I
```

NEATLIST would produce this:

```
10 CALL CLEAR :: PRINT :: DIM A(5)
20 FOR I=1 TO 5 :: A(I)=I :: NEXT I
```

NEATLIST is available from our library for $2 on your disk. The program, source code and documentation fills a single sided single density disk. It is distributed under the freeware concept, pay the author only if you like the product. I did and I did. Danny Michael, the programmer, asks for $10. A reasonable market price would be $20 to $30!

If you have memory expansion, a printer and a disk drive, you will find NEATLIST quite useful for debugging and pre scan (see below).

If you like NEATLIST, pay the programmer the $10 he asked for. He earned it!

CALL PEEK

This is from the NEATLIST documentation. CALL PEEK(8198,A) will let you know if a CALL INIT has been performed. If A returns as 170 then it has, any other value indicates that it has not.

GEMINI 10X FUNCTION CODES

The documentation that came with my GEMINI 10X was very helpfull when I was learning but is a bit difficult as a reference document. The listing in this issue of the ROM covers most of the instructions you will normally use (excluding downloading characters and graphics). I am told that these function codes also apply to other printers, such as the Epson.

ASCII CHART

I was asked how I learned XB's tokens for last month's list of ASCII codes. This little program did it:

```
10 OPEN #1:"DSK1.TEST",OUTPUT,
   VARIABLE 163,DISPLAY
20 FOR I=1 TO 254
30 PRINT#1:CHR$(0)&CHR$(I)&
   CHR$(I)&CHR$(0)
40 NEXT I :: PRINT #1:CHR$(255):
   CHR$(255) :: CLOSE #1
```

After entering this program, enter the following commands:

```
>RUN
>NEW
>MERGE DSK1.TEST
>LIST "PIO"
```

The MERGE command can take 5 to 10 minutes. You will need to delete some lines to get a good listing as some of the lines will send your printer function codes. When you are done, compare your program list to the ASCII chart.

LIBRARY

In our users group library, you will find the programs that have appeared in this column thus far. Included are: LOADMAKER, PRINTER, ORACLE, CONVERTER, ASCHART, PAYMENT and LOAD (a disk menu program for those with memory expansion).

PRE SCAN

You load your program, enter RUN and then . . . nothing. Finally your program starts to execute. On a short program this wait is not noticable but on a long one it can seem endless. Why the delay?

Your 99/4A is going thru your program, line by line, and alloting memory space. It is noting each variable used, each subprogram CALLed, the first DATA line, DEF statements, DIM statements, etc.

Futhermore, it is making an unduplicated list. Suppose you use the variable A 123 times in your program. The first time your 99/4A notes it and makes memory space. The other ONE HUNDRED AND TWENTY TWO TIMES it checks, notes that it already knows about this variable and moves on. Even at the speed of the 4A, this takes time.

XB has some tools to control pre scan. !@P- turns it off and !@P+ turns it on. The following items must be in the range of the pre scan:

--At least one use of each variable.

--At least one use of each CALL statement. For example, if you use CALL CLEAR five times, the first use must be within the pre scan.

--All DEF, SUB, SUBEND, DIM and OPTION BASE statements.

--The first DATA statement in the program.

!@P+ must be on a line by itself while !@P- can be at the end of a multi-statement line.

There is also a short cut -- pre scan does look at CALL statements but it does not check validity. Therefore you can do something like this:

```
10 DATA 2,3,4
20 OPTION BASE 1 :: DIM A(17)
   :: GOTO 30 :: CALL HCHAR ::
   :: CALL SPRITE :: CALL SAY
   :: R,S,T,U,V=W :: A$=B$
   :: !@P-
30 ! Program continues
```

Note that the code after GOTO 30 in line 20 will never be executed so it does not need to meet syntax requirements.

You should not activate pre scan until your program is fully debugged. If you forget something, you will get a SYNTAX ERROR.

With a bit of work, you will cut the pre scan time down significantly !

Techo Time

with Geoff Trott

RAMdisk for MiniPE system

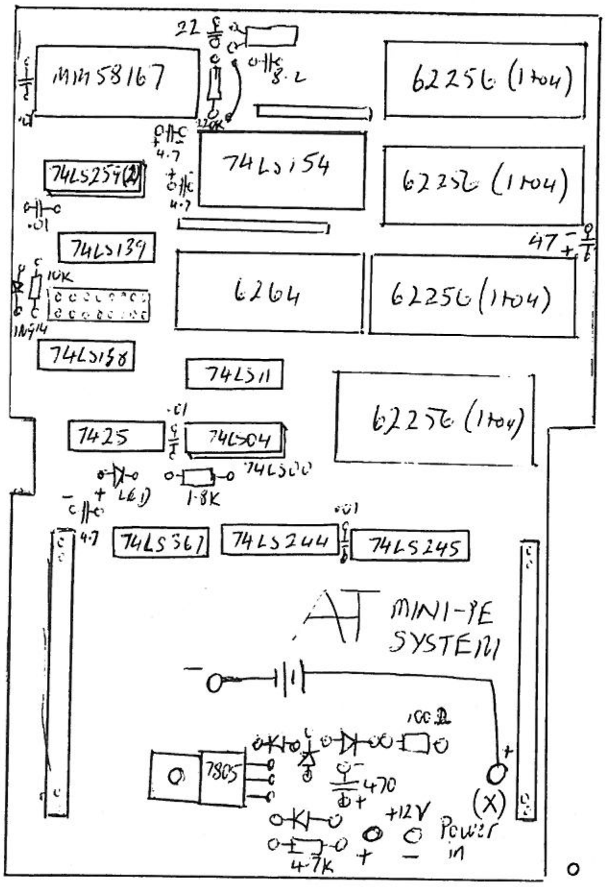
The shop has printed circuit boards in stock for RAMdisks which can be added to Peter Schubert's MiniPE system. These can also have up to 4 EPROMs added to them (with more than that requiring major surgery). They can also provide a clock for a time of day display on the RAMdisk menu. For people wanting to build them, here is some useful information. The parts list is:

Resistors
1x100 ohms
1x1k ohms
1x1.8k ohms
7x2.2k ohms in 8 pin sip
9x2.2k ohms in 10 pin sip
1x4.7k ohms
1x10k ohms
**1x220k ohms
Capacitors
**1x8.2pf ceramic
**1x22pf ceramic
4x0.01uf bypass
3x4.7uf tantalum 6.3v
1x47uf tantalum 6.3v
1x470uf electrolytic 16v
Diodes
1x1N914 small signal diode
4x1N4004 rectifier diode
1xLED any colour
Integrated Circuits
**1xMM58167 clock chip
1x6264 8K by 8 bit static RAM
(4 to 16)x62265 32K by 8 bit static RAM
1x74LS00 quad 2 input NAND
**1x74LS04 hex inverter
1x74LS11 triple 3 input AND
**1x7425
1x74LS138 3 to 8 line decoder
1x74LS139 dual 2 to 4 line decoder
1x74LS154 4 to 16 line decoder
1x74LS244 8 bit buffer
1x74LS245 8 bit bi-directional buffer
2x74LS259 8 bit addressable latch
1x74LS367 6 bit buffer
1x7805 5 volt regulator
Miscellaneous
**1xCrystal 32kHz
Batteries and holder as required 3.6v
2x20 pin in-line socket/pins, 0.1" centres
1x16 pins in two lines at 0.1" centres
**For the clock only. These are not needed for the RAMdisk.

As far as construction goes, consult the overlay diagram for the placement of parts. It would be wise to use sockets but they must be very narrow as the ICs, particularly the RAM chips, are very close together. I would not recommend mounting rechargeable batteries on the printed circuit board as I have seen too many circuit boards where corrosion has started. If you want to use NiCad batteries, mount them off the board, otherwise use a lithium battery. If you decide to put the clock on board, leave pin 23 of the 58167 chip out of its hole in the printed circuit board and wire it instead to the junction of the LED and the 1.8k ohm resistor, to achieve better power down of the clock. If you have the clock on board, you may also experience problems with loss of ROS and I have a little circuit board which can be installed to help this problem. More than 4 62256 ICs are installed, they must be put on the top of the first 4 ICs. This can only be done before all the ICs are installed in the board. Pin 20 of the RAM chips above the first level are connected individually to the 74LS154 by wires. There is a potential problem with the choice of manufacturer of the 74LS164 IC. This chip must not cause current drain through its outputs when power is off and the battery is supplying the RAM and clock ICs. Be careful of Hitachi made 74LS154s.

An external power supply is recommended, so cut the track underneath the board on the right hand side leading from the end pin of the connector (underneath the (X) on the overlay diagram. There are two links on the board, one for the clock circuitry near that IC and the other to provide either battery supply to pin 28 of the 6264 or +5 if an EPROM is used in its place for the ROS (not recommended by me for 32K RAM chips).

The second 74LS259 is mounted on top of the first one with pins 1, 2, 3, 8, 13, 15 and 16 of the two ICs connected together and pins 6, 7, 9, 10, 11 and 12 of the top one are not used. Pins 4, 5 and 14 of the top one are connected by wire to holes at a small slant angle and in line with the respective pins. The 74LS04 is mounted on top of the 74LS00 with only pins 7 and 14 of the two ICs connected together. Pin 6 is connected at a slant to a hole in line with pin 6 while pin 5 is connected to a hole in line with pin 7. No other pins of the 74LS04 are used.



continued from page 27

provided by setting up a computer and calling the BBS in the normal way. The TISHUG BBS is only available to members, however you can join at the meeting if you like what you see. Providing that I can be prised away from the host of things that I am involved in at the meeting, new members can be validated into the BBS on the spot.

In summary, here is what to expect at the TISHUG meeting on Saturday 3rd August:

9.00am to 11.30am - Train set group.
12noon to 2.00pm - Assembler class.
12noon to 4.00pm - BBS demonstration.
2.00pm to 2.30pm - Formal meeting, whats on info.
2.30pm to 4.00pm - Latest GAMES (3 computers).
- RamDisk Help group.
- Demonstration of RAMBO and PGRAM.
- Latest Software.
- Technical advise and repair help.
- TISHUG SHOP open.
- Publications library available.

continued on page 10

A Look at Interrupts

by Art Green, Ottawa Users Group

In an earlier article on sprites, interrupts were mentioned. This article looks at interrupts in some more detail. A short program is given that demonstrates how they can be used.

First, what is an interrupt?

To answer this question let us look at an analogous situation. You are sitting reading this article; the phone rings; you stop reading, remembering where you are and answer the phone; then you hang up the phone and return to your reading just where you left off. You were "interrupted" in your reading.

The same sequence of events happens in your TI99/4A. The CPU <you> is executing a BASIC program <reading this article>; an interrupt from the VDP occurs <the phone rings>; the CPU stops executing the BASIC program, saving its current status and position and executes the VDP interrupt service routine <answers the phone>; then completing the interrupt routine it returns to the BASIC program at exactly the point where it left off <you return to reading>.

Thus, an interrupt is a signal that some important event has happened. The TI99/4A has two types of interrupts or event signals.

The first type comes from the input/output devices. These interrupts are used by the routines on the device ROMs.

The second type comes from the VDP. The VDP signals an interrupt to the CPU every 1/50th of a second (power line frequency). This interrupt can and is used as a timer in the TI99/4A. There are at least two functions in BASIC that require a timer: sound processing (CALL SOUND specifies a duration for the note), and sprite movement (a speed is specified).

The VDP interrupt service routine in the console ROM does the timing for these two functions. It turns the sound generator off or moves a sprite when it is time to do so. The ROM interrupt service routine will also execute a user interrupt routine if one is present.

An Assembler Language programmer can make use of this feature. The presence of a user interrupt routine is indicated by placing the routine's memory address at location >83C4 hexadecimal or -31804 decimal.

The following BASIC program demonstrates the use of a user interrupt routine. The program can be run using TI BASIC with either the Mini Memory or Editor Assembler module, or can be run using Extended BASIC. Using Extended BASIC or the Editor Assembler module requires the memory expansion unit or card.

Let us have a quick look at the program.

180 XM is the address of the user interrupt routine if the extended memory is present.

190 MM is the address of the user interrupt routine if the Mini Memory is present.

220-240 the presence of the extended memory is determined by poking a value into it, then PEEKing to see if that value is there. If the extended memory is not there PEEK will return a zero.

300-400 reads the DATA statements and pokes the machine language user interrupt routine into memory.

440 pokes the address of the user interrupt routine into address >83C4, thus making it active.

460-470 is an idle look so you can see the display created by the interrupt routine.

540-630 these DATA statements are the machine language interrupt routine. Each DATA statement has 10 bytes of program and a check sum. If you mis-type one of the values you will get a message at statement 490 when the program is run.

The display created by the interrupt routine consists of a box drawn with asterisks. One asterisk is placed on the screen every interrupt (i.e. every 1/50th of a second). There are 60 asterisks in the box so that the box is completely drawn in 1.2 seconds. After drawing the box it is erased in the next 1.2 seconds by writing blanks instead of asterisks.

You will note that the display continues even after the idle loop at lines 460 and 470 completes. And in fact, it will continue even while doing other things in BASIC. Try listing the program, or entering a new one.

Finally, for those who understand assembler language, a list of the assembler source program for the interrupt routine follows the BASIC listing.

BASIC Listing

```
100 REM INTERRUPT DEMO
110 REM
120 REM MACHINE LANGUAGE
130 REM ROUTINE LOADED AT
140 REM >2600 EXTENDED MEM
150 REM >7200 MINI MEMORY
160 REM
170 CALL INIT
180 XM=9728
190 MM=29184
200 LAD=XM
210 REM TEST FOR EXT MEM
220 CALL LOAD(XM,170)
230 CALL PEEK(XM,X)
240 IF X=170 THEN 27
250    REM MUST BE MINI MEM
260    LAD=MM
270 A=LAD
280 REM LOAD ML ROUTINE
290 CALL CLEAR
300 FOR D=540 TO 630 STEP 10
310    CHECK=0
320    FOR N=1 TO 10
330       READ X
340       CALL LOAD(A,X)
350       CHECK=CHECK+X
360       A=A+1
370    NEXT N
380    READ X
390    IF CHECK<>X THEN 490
400 NEXT D
410 REM POKE INTERRUPT
420 REM ROUTINE ADDRESS
430 REM INTO >83C4
440 CALL LOAD(-31804,LAD/256)
450 REM JUST IDLE AWAY TIME
460 FOR N=1 TO 10000
470 NEXT N
480 STOP
490 PRINT "ERROR IN DATA STATEMENT ";D
500 STOP
510 REM EACH DATA STATEMENT
520 REM HAS 10 DATA BYTES
530 REM AND A CHECK SUM
540 DATA 192,236,000,092,004,194,005,131,002,131,987
550 DATA 000,060,026,003,004,195,006,236,000,094,624
560 DATA 203,003,000,092,060,172,000,090,006,002,628
570 DATA 017,015,019,010,006,002,019,004,002,000,94
580 DATA 002,039,010,083,016,002,002,000,002,086,242
590 DATA 096,003,016,007,002,000,000,119,010,083,336
600 DATA 016,002,002,000,000,072,160,003,002,096,353
610 DATA 064,000,006,192,215,192,006,192,215,192,1274
620 DATA 016,000,216,044,000,094,140,000,004,091,605
630 DATA 000,015,000,000,138,128,000,000,000,000,281
```

continued on page 16

Grade Standardisation

by Bob Relyea

Back in the April, 1989 issue of the TND I put a couple of articles in about standardising student's marks. There was a Multiplan version and an Extended Basic version. The Extended Basic version was a very 'skeletal' program containing just the bare necessities, as it were. You may recall that I have been working on a Fish program for one of the club members and have made some advances with it. In the process of working on the Fish program I made consultations with Ross Mudie (who has not?) on a couple occasions about some points of programming that I did not understand. With the knowledge that I gained I went back and thoroughly re-wrote the Grade Standardising program which is able to do far more than the previous one. I have included the program at the end of this article. Instead of typing it all in, if you have need of it see me at a club meeting and I will arrange a copy.

Standardising Grades is not something that everybody has need of. The reason, therefore, that I have included the article in the TND is to provide some clues for those who are seeking of various ways of setting out a program. The main part of the program is to 'juggle' grades and re-assign them to a particular average and standard deviation of your choice (see the April, '89 article for more details). This procedure involves an algorithm, which is just a series of calculating steps that lead to a answer of some sort. So, this program could be applied to any algorithm. For anybody who has little programming experience, it may, due to its size, appear a bit daunting. Believe me, I am not a crash-hot programmer either and most of what results is the result of pure 'slog'. The result was very rewarding, however, and the program does about everything I could hope for.

One thing about the program that beginners could take note of is that it is completely menu-driven. This is easy to set up. All you have to do is get the listing right and then feed each choice off to an appropriate line number. Each step from there is a little sub program that does a particular task and then goes back to the main menu.

I have made the mistake in the past of accidentally saving over an existing file that I did not want to over-write. I consulted Ross about this and he gave me an outline of steps that I might use to remedy the matter. If you are interested in having the computer prompt you with a 'Over-Write Existing File?' message then have a look at the way I set it up. The basis of it is to actually ask the computer to 'load' a file by using Input in the Open statement. This means that the file name that you have offered for 'saving' is searched for on the disk. If found, instead of using the usual Print statement which follows the Open statement, the next line will instead supply the prompt 'Over-Write Existing File (Y/N)?' You then have a line for the 'Y' and one for the 'N' response. The first line following both the 'Y' and the 'N' response will have to Close the original file. If, on the other hand, the file is not found then there would normally be an error message. To allow for this you put an 'On Error [line number]' line just before the Open statement which directs proceeding off to another Open statement (this time with 'output') and you are in business. This is just one of the many things that I learned in the many hours that I spent on the program.

One way to learn from somebody else's programs is to go to the menu and follow each choice and carefully note how each routine is done. Play around with the program yourself by changing a few things here and there to see what the result is. My program could have been written more efficiently by using multi-line commands and I am sure that the seasoned programmers could improve on it here and there but it works OK for me and that is all that counts at this stage. Hope you learn something from it.

program listing on page 13

Assembly Language Class

by Ross Mudie

The assembler class at the last meeting ran a little differently to the usual. The class project was to come up with an assembly program linked from extended basic. The function of the program is to accept a single key press and then return the result to extended basic. The little extras required are: To provide a cursor at the required row and column; to beep; to get the character off the screen at the cursor position; to return the character off the screen if just the enter key is pressed; and of course ensure that the old character was left on the screen when enter is pressed.

The program was taken up to the stage of showing the key pressed at the required row and column. (The counting of columns is like VCHAR or HCHAR which start with column 1 at the very left of the screen). The program then returns the value for the key pressed in the variable KEY. In extended basic the program is linked using CALL LINK("KEYIN",ROW,COLUMN,KEY).

The program developed so far is not complete and it contains some unnecessary steps which were included to make it easy to understand where the class was up to.

Items still to be added are: BEEP, getting the character off the screen and returning it if ENTER is pressed and flashing a cursor or the character off the screen position. The extended basic test program and the assembly, so far, follow. The idea is to have a go yourself to see if you can write the rest of the source code and make it work. Later the code will be worked on to make it more compact by removing the unnecessary steps.

Members are welcome to attend the assembly language class at 12noon on the next meeting day, Saturday 3rd August 1991. All you need is your E/A manual, a pen and some paper for taking notes. Subdued cheering is permitted whenever the assembler comes up with 0000 errors or a section of code actually works! If you bring a SSSD formatted disk you can take home a copy of the group's handywork at the end of the class session.

Here are the extended basic and assembly programs. Type them in and assemble the assembly program; they do work! Then see if you can write the rest of the code, or come along to the class and learn how to do it.

The assembly source file is purposely not commented yet. If you want try to work out how it works and comment it, this will also be handled as a group project in the class. By the way, if you run the program do not despair at the blank cyan screen, just press a key...

program listing on page 15

continued from page 3

outstanding. Your Directors have resolved to contact most of these people with a view to trying to entice them back to the club. The feeling is that most will probably renew.

I have just received the June issue of Micropendium and it has, as usual, some interesting articles. One article on the TI Image Maker makes for very interesting reading. Harry Brashear, the reviewer, gives it straight A's. At a cost of $179 (US) you can obtain the TIM boards, 2 disks of software and have 80 column capacity for your TI99/4A. OPA, 432 Jarvis Street, Suite 502, Toronto, Ontario, Canada are the suppliers. Naturally you will need an appropriate monitor to get the most out of your 80 column card. Your club is purchasing one, so watch for it being demonstrated at a forthcoming meeting.

That is all for this month. See you at the next meeting.

Multiplan Exercises #6

by Herbert Schlesinger, USA

WINDOWS:

Bring up the file "TENYEAR1" on the screen. (Use transfer and Load commands, remember?) For this exercise it will be best if the Options from the main menu calls for Recalc(Yes)No so that changes will show automatically. If you have set this for No, go back and change it.

Move the pointer to R5C2, ABC Co's increase rate, and change the 15% figure to 12% (.12). The sheet will show the resulting changes at once all across the board. The rental and the totals for each column from 1984 to 1994 will adjust themselves. To see this we must scroll the screen to the right - but when we do this we lose the indentifying labels on the left edge of our sheet. To help us with this problem we shall use a Window to Freeze the labels so that they do not scroll.

FREEZING TITLES:

The Window-Split-Titles command will allow the titles in column 1 to be frozen on the screen. Here is how:

1. Move pointer to R1C1.
2. Select Window command which gives these options:

Split Border Close Link

Select Split

3. You then see:

Horizontal Vertical Titles

Select Titles

4. Now the screen shows:

# of rows: 0 # of columns: 0

5. Press the Tab (CTRL A) key to select 0 rows since we are going to freeze a column. Type 1 for number of columns.

6. Press <ENTER>. Notice that a #2 is hylited indicating window #2.

Now when you press FCTN 1 (the END key) to get to the other end of the projection, column 1 will remain in place with the labels so you can see what each figure means. Some of the Long Titles will be shortened or truncated.

If you wish to freeze a row of titles follow the same procedure, but remember to position the cell pointer BELOW the row of titles (actually place it on the first row you want to be free to scroll.) To unfreeze the titles, select Window and Close options and press the <ENTER> key.

SPLITTING THE SCREEN:

Rather than scrolling to reach from one end of sheet to the other, there is another way - Split the screen. Here is how:

1. Pointer to the first row or where you want the screen to split. Try row four (R4C4).
2. Select Window from the menu: there are four options:

Split Border Close Link

Select Split

3. Then we have:

WINDOW SPLIT: Horizontal Vertical Titles

Select Vertical.

4. Resulting in these options:

WINDOW SPLIT VERTICAL at column: 4 linked: Yes(No)

Press <ENTER>. Column 4 is the default because we put our pointer there before we started this operation.

The screen will split into two windows each with its number in its upper left corner. The pointer is active in the window with the hylited number.

We can scroll around in window #2 and examine the figures while the Titles and the first years figures are also on the screen. Now if you change ABC Co's rate, the change at the far end of the projection can be seen as it changes. Use CTRL 6 (change window key) to change the pointer from one window to the other. If there are more than two windows, pressing this key causes the active window to be rotated. In order to view more of the sheet at once, Close the existing window by selecting the commands Window Close and then pressing either 1 or 2 since we have only two windows open at this time. Now we are back where we started. Next move the pointer to the line below Total which in this case is row 11. Now select the Window command, Split, Horizontal and then press <ENTER>. The screen will split at row 11, window #2 will be hylited, but it is empty as you see.

To put the later years in window #2, move the cell pointer to R3C9. The easiest way is to use the Goto command (Page 5). But, here it is again:

From the command line, press the space bar six times and press <ENTER>, or press "G". The bottom of the screen will then show the Goto options:

GOTO:Name Row-Col Window

Select the option or type command letter

Press "R".

GOTO row: 1 column: 1
Enter a number

Enter a 3, then tab to the column and enter a 9.

Press <ENTER>

The columns you asked for are now in window #2. Scroll as you wish in this window. Now if you switch to window #1 and change a rate of increase or the 84 rental figure, you can watch the changes all across the years.

WINDOW BORDERS:

Things can be made clearer if you outline the windows with borders. Realize that these take room and you may find that you are better off without them. However, to place the borders:

1. Move the pointer to window #1 (use the Fctn 6 Key)
2. Select the Window and Border commands.
3. Multiplan will ask:

WINDOW change border in window number: 1

Press <ENTER> to border window 1

4. Move the cell pointer to window #2.(Use Fctn 6).
5. Select as 2 above specifying #2
6. Press <ENTER>

When you tire of the borders repeat the above. Notice that the commands are to change the border: if it is not there you get a border and if it is there it goes away.

continued on page 24

Beginning Forth - part 8

by Earl Raguse, UGOC, CA USA

GRAPHICS

We are going to take up an entirely new subject - graphics. Forth gives us bit map mode graphics, a capability we did not have in XBASIC. In the bit map mode, the CRT screen is 256 pixels wide by 192 pixels high, in theory, at least, my CRT does not show them all. The 0 0 coordinate is in the upper left corner, instead of the lower left corner where mathematicians usually put it. We are able to plot a DOT (pixel), of a specified color, at any specified set of coordinates on the CRT. There is also the word LINE which plots a solid line of dots on a line between two specified sets of coordinates. This is all in addition to the graphics capabilities that you have grown to expect in XBASIC.

Before we get too far into this thing, you should read the TIFM Chapter 6 Graphics. There are three graphics modes available, but the standard GRAPHICS mode is like XB and will be only briefly discussed here with a small sprite demonstration. There is also MULTICOLOR mode which I do not find very useful, it will not be discussed here, you may try it yourself, the TIFM is fairly clear.

First I must point out that all is not rosy however, there is no text capability in the bit map mode. Hold on though, next time, I shall show you how we get around that however, but we must have the 64SUPPORT Editor loaded. Even though I do not like to work with it, it is not all that bad when one gets used to it. In addition, you must have -GRAPH loaded from your TI FORTH (backup) System Disk.

Go back to lesson #2, and if you did not elect then to load -64SUPPORT and -GRAPH, please make a new working disk for graphics which also includes these. Since a near future lesson will include Floating Point Arithmetic, I suggest you load -FLOAT also, further be sure to load my UFW's, because they will be needed for almost every thing I write. Then BSAVE the lot as explained before.

Also this time I am including another set of stack manipulators, Screen #33, which are needed for the graphics screens herein. You may BSAVE them also if you wish, but if not, they will be CLOADed (Conditional Loaded) by the graphics screen if they are not already in the dictionary, but you must number the Screen #33, or change the CLOAD statement. Observe how IT is redefined at the end of the screen instead of immediately as usual. This insures that, once in the dictionary, they will stay until you reboot, or deliberately FORGET them.

To invoke the Bit Map Graphics mode, you must specify GRAPHICS2, SPLIT, or SPLIT2. The latter two provide a combination of bit map and text modes, and the best way to find out what they are is to try them, see TIFM Chapter 6 page 20 and 21. The -64SUPPORT Editor uses SPLIT. Be sure to read about DRAW, UNDRAW, and DTOG, I do not use them, so try them yourself. I will use DMODE in the default state of zero, or the DRAW mode.

DOT is quite simple, you just put a pair of numbers (within CRT limits) on the stack and enter GRAPHICS2 DOT. Did it work? Ok, but how do you get rid of that black screen, and get the cursor back? Enter TEXT blindly, you will not see it but you will get the TEXT mode screen back with the cursor. I think it would be very tedious to do anything useful with DOT by itself, but LINE is more impressive.

Try this: GRAPHICS2 5 5 250 180 LINE. Aha! Eureaka! and all that kind of stuff, is not that neat? Enter TEXT or just simply TX, the UFW abbreviation. The trick is to include the word TEXT in the graphic word definition, after a suitable delay, use WAIT or KEY. See the example screens.

Lets now experiment with SCREEN and DCOLOR. SCREEN will accept either a HEX or DECIMAL number on the stack. The Color Table is on page 6. Try HEX 8 SCREEN. You should get a red CRT, and no cursor. SCREEN puts you in a graphics mode where there is no text! Just enter TEXT or TX and you will get the cursor and the original CRT color back.

The TIFM says specifically that you must use HEX to change DCOLOR, and I have found that to be true. The TIFM is rather vague about DCOLOR and there is no example. I had initially assumed that DCOLOR was like SCREEN, but it did not work, so I read some more. It turns out that DCOLOR is a variable, and thus you must store (!) a value in DCOLOR, two HEX digits, the first is the foreground color and the second is the background color. Enter HEX 40 DCOLOR !. Now, repeat the LINE experiment above, you should get a blue line, do not forget to set BASE to DECIMAL or put the coordinates in HEX. Recall the UFW's define DEC for DECIMAL.

Enough of working in the immediate mode, let us do a test screen. Screen #57 demonstrates all we have talked about above, and more. Notice that it begins with 33 CLOAD 2ROVER, which loads the stack manipulators that were mentioned before. Each LINE word requires 2 pairs of X Y coordinates on the stack. If we wish to draw lines from one point to another, and then on to another ect, we would find ourselves entering the last number pair as part of the next 2 pairs. Since this happens very frequently, it seemed like I should define a word 2ROVER to do this for me. That is what Forth is all about!

The word 2ROVER, on Screen #33, duplicates the top two numbers on the stack and inserts them under the fourth number down, so they will be there for the next line to be drawn. Since this resembles the operation of OVER in reverse, but for two numbers, I called it 2ROVER, (Double Reverse Over). Note that 2ROVER in turn uses 2DUP, and 2SWAP, words we will see a lot of later when we work with Double Precision numbers. I do not want to get side tracked today about how these work, in the near future, however, I will discuss them in detail. If you can figure out what they are and what they do, feel free to use them.

Screen #49 defines the word XBOX which draws a box with an X in it. The word TEST tests it after setting things up for GRAPHICS2. This box is white on black, the normal default setting for GRAPHICS2.

The word SCRN turns the CRT white. BLUE, DKRED, GREEN, and BLACK define their respective colors on a white screen, making use of SCRN. The word WH/BK restores colors to default white on black. The words GRN, BLCK, RED and BLU, call the color change words and execute XBOX. You will notice that after executing one of the color words, the line colors remain the same but on a black screen if you execute TEST by itself. To get back to default white on black, you must execute WH/BK TEST.

```
SCR #33
 0  ( STACK MANIPULATORS EGR 12/87) FORGET IT
 1  : PICK ( n1 -- n2 )  2 * SP@ + @ ;
 2  : ROLL ( nk ... n1 k -- nk-1 ... n1 nk )
 3       DUP 1 = IF DROP ELSE DUP 1 DO SWAP
 4       R> R> ROT >R >R >R LOOP 1 DO
 5       R> R> R> ROT ROT >R >R SWAP LOOP THEN ;
 6  : NIP ( n1 n2 -- n2 ) SWAP DROP ;
 7  : TUCK ( n1 n2 -- n2 n1 n2 ) SWAP OVER ;
 8  : 2DUP ( n1 n2 -- n1 n2 n1 n2 ) OVER OVER ;
 9  : 2DROP ( n1 n2 -- ) DROP DROP ;
10  : 2SWAP ( n1 n2 n3 n4 -- n3 n4 n1 n2 )
11       ROT >R ROT R> ;
12  : 2OVER ( n1 n2 n3 -- n1 n2 n3 n1 n2 )
13       >R 2DUP R> ROT ROT ;
14  : 2ROVER ( n1 n2 n3 n4 -- n3 n4 n1 n2 n3 n4 )
15       2DUP >R >R 2SWAP R> R> ;  : IT ;
```

```
SCR #49
( GRAPHICS DEMO #1 EGR 11/88 REV 10/89)  DEC
  FORGET IT : IT ; 33 CLOAD 2ROVER
: M CLS 7 12 AT ." TRY RED GRN BLK BLU or DIR" CR ;
: XBOX 30 10 225 180 2ROVER LINE 225 10 2ROVER LINE
       30 10 2ROVER LINE 30 180 2ROVER LINE 225 180
       LINE 225 10 30 180 LINE 600 MS TEXT ;
: TEST GRAPHICS2 XBOX ; HEX .
: SCRN GRAPHICS2 OF SCREEN ; \ white screen
: WH/BK FO DCOLOR ! 1 SCREEN ;
: BLUE  40 DCOLOR ! SCRN ;
: DKRED 60 DCOLOR ! SCRN ;
: GREEN CO DCOLOR ! SCRN ;
: BLACK 10 DCOLOR ! SCRN ; DEC
: GRN GREEN XBOX M ; : BLK  BLACK XBOX M ;
: RED DKRED XBOX M ; : BLU  BLUE  XBOX M ;   M
```

Screens #58 thru #60 are a little demonstration of the GRAPHICS mode with Sprites. Load Screen #60 first. The demo allows you interact with sprite instructions, while the sprites are running, you could do this in XBASIC, maybe not so easy, but you could. I do not explain, I think its pretty clear if you read the TIFM.

Well, I think that is enough for this time, next time we will continue with some more elaborate examples, including the promised text in the graphics mode.

C U later, may the FORTH be with U.

```
SCR 58
( SPRITES EGR REV 12 85) HEX
  CLS B 9 AT ." LOADING SPRITES"
: GO GRAPHICS 2000 SSDT ; DECIMAL
: SPTPAT 26 0 DO I 65 + CHARPAT I SPCHAR LOOP ;
: BLDSPT 26 0 DO 120 90 15 I I SPRITE 1 MAGNIFY
  20 RND 20 RND - 20 RND 20 RND - I MOTION LOOP ;
: M MAGNIFY HOME ;
: G #MOTION HOME ;
: DOSP  1 1 GO SPTPAT 27 G BLDSPT 1 M ;
: SPSP  DELALL TEXT 59 LOAD ;
: QTSP  DELALL TEXT 60 LOAD ;
: ?STOP ?TERMINAL IF QTSP ENDIF ;
: GSPR BLDSPT ?STOP MYSELF ;
: GOSP  DOSP GSPR  ;
: 59L 59 LOAD ; : 60L 60 LOAD ;

SCR 59
( SPRITES EGR REV 12 85 )
58 CLOAD 60L
8  4 AT ." THIS IS DOSPRITE"
5  6 AT ." FOR FUN ENTER <DOSP> THEN "
5  7 AT ." TRY <3 M>, <O M>, <20 G>, "
5  8 AT ." OR <O G>, THEN <26 G>  "
5  9 AT ." OR ANY COMBINATION OF <n G>"
5 10 AT ." OR <n M>, <SPSP> ENDS IT"
5 13 AT ." !!
 DO NOT START TILL YOU !!"
5 14 AT ."    !! KNOW HOW TO STOP!!"
5 19 AT ."     BORED? TRY <60L>."
QUIT

SCR 60
( SPRITES EGR 10 22 85)
  FORGET IT  : IT ;
  58 LOAD  CLS
 11 10 AT ." THIS IS GOSPRITE"
   9 12 AT ." ENTER <GOSP> TO START "
   9 14 AT ." HOLD <FCTN 4> TO QUIT "
   7 18 AT ." HOLD FCTN 4 TILL IT STOPS"
 10 20 AT ." BORED?  TRY <59L> "

QUIT
```

o

continued from page 22

LINKING:

When you are scrolling in one window the other is not affected UNLESS the windows are Linked. Linking causes the windows to have the same movement in both at the same time; if you move vertically in a horizontally split situation, or horizontally when the windows are split vertically, the relative position of the data remains the same:

WINDOW SPLIT HORIZONTAL at row: XXXX linked: Yes(No)

The "at row" is the pointers present position. The "linked" is always NO. To Link windows, specify the row, press TAB (CTRL 2) and then select Yes and press <ENTER>.

Whenever you call for the window command the option to Link appears on the submenu:

WINDOW LINK window number: 1 with window number: 2
linked:(Yes)No

Use the TAB key to select the windows to link and Yes to have them link. Repeat the process to unlink by pressing No.

-----------0-----------

PROTECTING CELLS

On this spreadsheet which is "TENYEAR1" most of the input are formulas; only columns two and three use Input Data, data which we must type in to the screen. All the information in columns four thru 13 are provided by the formulas we have entered. To make sure that we do not erase a formula by mistake, or some one else using the spreadsheet does not make that error, we can use the Lock command to protect the formulas:

LOCK Cells Formulas
Select Formulas. You are then asked:

Enter Y to confirm

So enter a Y.

After locking the formulas the pointer can move about as before, but if you attempt to erase or change a "formula" cell you will receive the warning:

Locked cell may not be changed

To UNLOCK all the formula cells move the pointer to the Home position,(R1C1) with CTRL 1. Use the Lock command from the menu and then the Cells option:

LOCK CELLS: R1C1   status:Locked(Unlocked)

Indicate the entire spreadsheet by pressing (:) to make a range and then the End key (FCTN1). Then tab (CTRL2) to the status section and select "unlocked". Press <ENTER> and the entire spreadsheet is unlocked. Move to any cell and change it as you wish. o

Treasurer's Report

by Geoff Trott

The renewal of membership bulge is now past with a pleasing result this year. We have only lost about 20 members, with some of those still expected to re-join. I have sent out 140 membership cards so far this year, which is very good. We have over 20 members overseas and these compensate a little for the loss of local members.

Income for June	$2062.65
Payments in June	$3418.74
Excess of expenses over income for June	$1356.09

o

TI-Base Tutorial #12

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.

At this time, work (the way I make money to support my TI) is taking up most of my time. I will try and write a couple example programs (CFs) to keep you going until I can get back to my TI on a regular basis. I am sorry if the articles are a little skimpy, but it is the best I can do right now.

```
*                    DOTM
SET TALK OFF
 CLEAR
SET RECNUM OFF
SET HEADING OFF
LOCAL TEMP C 19
LOCAL TIME C 11
SELECT 5
CLOSE
USE DSK2.DT'TM
  WRITE 12,10,"TURN YOUR PRINTER ON"
  WRITE 20,10,"ENTER THE TIME"
  WRITE 22,4,"TIME EXAMPLE >12:49 P.M.<"
  WRITE 23,4,"             >          <"
  READSTRING 23,18,TIME
 IF TIME <> "           "
     TOP
    DELETE RECORD
     PACK
    APPEND BLANK
     REPLACE 5.DT WITH .DATE.
     REPLACE 5.TM WITH TIME
  ENDIF
    BOTTOM
    MOVE -1
      SET RECNUM OFF
      SET HEADING OFF
 REPLACE TEMP WITH "   SYSTEM LAST RUN "
 PRINT (Drft),(E),TEMP,DT,TM
    MOVE
 REPLACE TEMP WITH "SYSTEM CURRENT RUN "
 PRINT TEMP, DT,TM,(LF)
 PRINT (Drft)
      SET TALK ON
      SET HEADING ON
      SET RECNUM ON
CLOSE
SELECT 1
CLEAR
RETURN  Copyright Martin A. Smoley 1989
*
* Save current TIME  DATE to  DT'TM
```

In this issue I have 3 CFs that do almost the same thing, but not quite. Their difference is what makes them interesting. All three of the CFs ask you for the time. They then print out the last time and date the CF was run and the current time and date which you just entered. I created it because I was printing out several copies of the same report in one evening and I could not tell the updates from the first printout. Having the time at the top of each printout solved my problem. This CF could be used for the last time you balanced your checkbook or paid your bills, etc. Create a very simple database named DT'TM or DT'TM2 depending on the DB you find in the USE statement in the CF you wish to use. The DB contains 2 fields. The first is named DT, type = D, with a width of 8. The second is named TM, type = C, with a width of 12. The CFs will use this DB to store the time and date for retrieval the next time the CF is run. The first CF (DOTM) is the one I use. It will allow you to set the length of the DT'TM Db by appending as many records as you wish. In other words, you could keep the last two times and dates the Db was run or the last ten if you wish. The CF will eliminate the oldest record in the file and append the newest record to the end of the file. The interesting part of this CF is the use of TOP, BOTTOM, DELETE RECORD and PACK to hold the DB at a pre-determined size. MOVE -1 and MOVE are used to locate the records to be printed.

```
*                          DOTM2
      SET TALK OFF
     CLEAR
LOCAL TEMP C 19
LOCAL TIME C 11
      SET RECNUM OFF
      SET HEADING OFF
  SELECT 5
  CLOSE
  USE DSK2.DT'TM2
     BOTTOM
  WRITE 12,10,"TURN YOUR PRINTER ON"
  WRITE 20,10,"ENTER THE TIME"
  WRITE 22,4,"TIME EXAMPLE >12:49 P.M.<"
  WRITE 23,4,"             >          <"
  READSTRING 23,18,TIME
 IF TIME <> "           "
    APPEND BLANK
     REPLACE 5.DT WITH .DATE.
     REPLACE 5.TM WITH TIME
  ENDIF
   MOVE -1
 REPLACE TEMP WITH "   SYSTEM LAST RUN "
 PRINT (Drft),(E),TEMP,DT,TM
   MOVE
 REPLACE TEMP WITH "SYSTEM CURRENT RUN "
 PRINT TEMP, DT,TM,(LF)
 PRINT (Drft)
      SET TALK ON
      SET HEADING ON
      SET RECNUM ON
CLOSE
CLEAR
  SELECT 1
RETURN
*
* DOTM2  Copyright Martin A. Smoley 1989
*
* Save current TIME & DATE to  DT'TM2
```

Another CF would be needed to print out any extra times and dates in the DB. The CFs in this article will only print out the last record and the current time and date. The next CF (DOTM2), works almost the same as the first except for the records kept. DOTM2 works with your last two entries, but it keeps all of the previous entries. This CF would be great if you wanted to keep a complete record of the time and date a certain set of CFs was used, but you must remember that you have a self generating DB in the system. This means that each time you use the system DT'TM2 will get larger and therefore you will have less and less disk space for other types of TI-Base use.

```
*                          DOTM3
     SET TALK OFF
    CLEAR
LOCAL TEMP C 19
LOCAL TIME C 11
LOCAL TMTMP C 12
LOCAL DTTMP D 8
     SET RECNUM OFF
     SET HEADING OFF
SELECT 5
CLOSE
USE DSK2.DT'TM
    BOTTOM
  WRITE 12,10,"TURN YOUR PRINTER ON"
  WRITE 20,10,"ENTER THE TIME"
  WRITE 22,4,"TIME EXAMPLE >12:49 P.M.<"
  WRITE 23,4,"             >          <"
  READSTRING 23,18,TIME
 IF TIME <> "           "
```

```
      REPLACE DTTMP WITH 5.DT
      REPLACE TMTMP WITH 5.TM
      TOP
      REPLACE 5.DT WITH DTTMP
      REPLACE 5.TM WITH TMTMP
      BOTTOM
      REPLACE 5.DT WITH .DATE.
      REPLACE 5.TM WITH TIME
   ENDIF
    MOVE -1
  REPLACE TEMP WITH "   SYSTEM LAST RUN "
  PRINT (Drft),(E),TEMP,DT,TM
    MOVE
  REPLACE TEMP WITH "SYSTEM CURRENT RUN "
  PRINT TEMP, DT,TM,(LF)
  PRINT (Drft)
       SET TALK ON
       SET HEADING ON
       SET RECNUM ON
CLOSE
CLEAR
  SELECT 1
RETURN  Copyright Martin A. Smoley 1989
*
* Save current TIME & DATE to  DT'TM
```

DOTM3 is almost identical to DOTM. It only saves two records and reads and prints in the same manner. The real difference is that it holds data in its own variable space while it moves through the DB DT'TM to replace old data with current data. It created this CF to get away from the PACK command. In certain instances this algorithm will be faster. More important it does not place any system messages on your screen. This allows you to hold messages or menu selections in place on the screen without having those annoying system messages that scroll the screen up and throw away the top line on the screen. I am sure that most of you will find this problem minute, but the idea may help you somewhere else in your programming endeavor. Another idea which you may need from time to time is the selection of a unique number. Neither human selection or the random generation of a computer should be trusted with this task. Unique code numbers are the truest when you have the computer extract pieces of the date, the time and at least two letters of a persons name. Some companies use time, date, zipcode and names. Take a look at some of your junk mail for numbers that might follow this pattern. I mentioned the unique number uses because parts of these time, date CFs could be used to generate a unique number that you could then relate to a person or companies name in a mailing list. This number could then be used to relate two or more DBs together to gather mathematical data, as I have shown in the past. It could also be used as an access code or for other information. In other words, any of these three CFs could be converted to ask you for the time and your access code. It could then save a record of who used the system, with the time and date. There are many many uses for any one idea. You may need to slightly modify a particular CF for a new job, but its easier than writing a new one.

TI Sort

I must put in a couple of plugs for Inscebot. They have created some great software for the TI and I think that TI Sort will be close behind TI Artist and TI Base. I use it more and more as time passes. It is fast, accurate, and very versatile. If you work with any amount of data, I think you should pick up a copy of TI Sort for your collection of utilities.

TI-Base Ver. 2.03

I previously received version 2.02 for testing. I had not had much time to play with it when version 2.03 arrived at my door. Version 2.02 corrected a bunch of minor problems and version 2.03 corrects several more. With the latest versions of TI-Base you also have the ability to load from a hard drive and use a PATH function to find the TI-Base main program files. You may never notice some of the problems that are constantly being corrected, but they are being corrected anyway. I bring this up because I feel that the TI community is getting more support from people like Dennis Faherty than you can comprehend, at a very small cost. Please try to support the efforts of our last major software suppliers. o

Newsletter update

by Bob Relyea

TIBUG (Brisbane), May, 1991: Editorial, TI-Base Tutorial No. 4, "I Remember" by Col Christensen, Newsletter Digest, What's New, Tips From The Tigercub No.29, List of Hardware and Software Suppliers, Trading Post, TI to IBM Cable by Steve Burns, various advertisements, TML Graphics Programs by S. Shaw, Module Library, Multiplan Exercises No. 1 by Herbert Schlesinger, Why Learn To Program? by Jim Peterson, Review of "Yet Another Paint Program".

HUNTER VALLEY 99ers, March, 1991: President's Report, Cosmetics and Skin Cancer, Tribute to John Birdwell (MICROpendium), Far Out (in the bush) by Dick Schaydel, Getting Even?, Assembler Executing ... by Bob Carmany, Random Bytes by Bob Carmany, Multiplan by Peter Smith, New/Age Special by Jack Sughrue, New-Age/99 No. 12 (Mickey Revisited) & No. 13 (the VCR Connection) by Jack Sughrue, Computer Knowledge Test, New Products, What's New?? by Joe Wright, Joystick Adaptor (complete with diagram) by Paul Mulvaney).

TI UP TItBITS, April, 1991: Hear Ye (Editorial), Putting It All Together by Jim Peterson, Secretary's Report, XB Tips Nos. 1 to 4 by Jim Swedlow, DV/80 File Converter, Multiplan Data, Anniversary Reminder by Steve Wilkinson, A Real Clock for the TI99/4A complete with diagram) by John Willforth, Programs regarding the clock to type in, meeting notices, Notes On LASER Printing With The TI-99/4A by Lary Fairbanks, Hardware Review.

LA TOPICS, May, 1991: President's Thoughts, LA Users Group Market Place (a list which comprises half of the magazine), sale of Modules, The Cracker Barrel by Chick De Marti, TI-Writer Tips.

TI FOCUS, May, 1991: News and Views by Tom Arnold, You Don't Have to Have It All by Jim Peterson, Tiny Tim from OPA by Randy Packham, Club Page by Tor Hansen, Simple Programs by Jim Peterson, Music Pro Version 1.4 by Jim Peterson, Language Drills by Don Shorock (a review), The Home Computer by Jim Peterson, Stackware on Geneve.

SPIRIT of 99, May, 1991: Announcements, Gemini 10X Printer Problems by Allan Cox, Letter to the Editor, Tournament Solitaire (a review by Jim Peterson), Reformatting by Jim Peterson, Tips From The Tigercub No. 64 by Jim Peterson, Programming Music the Easy Way by Jim Peterson, TIPS 180 (long article) by Ron Wolcott.

THE OTTAWA NEWSLETTER, May, 1991: Editor's Notes, President's Message, Fast Extended Basic by Lucie Dorais, DIJIT AVPC 80 Column Card (long, comprehensive article) by Jan Alexandersson.

UGOC ROM, May, 1991: Slate of Officers Nominated, Happy Birthday UGOC (10th anniversary) by Earl Raguse, TI still cares a little bit, Membership Corner, You, You, You and You by Earl Raguse, Subprogram List by Earl Raguse, A Review of BITMAP DRAW by Stan Corbin.

THE BOSTON COMPUTER SOCIETY, April/May,1991: Listen by Justin Dowling, The National "TI-ECHO" (the NETWORK and how to connect with it) by Kevin Coleman, Intro to the UCSD P-System by Ron Williams, Why Should You Learn To Program by Jim Peterson.

TIdbits, May, 1991: President's Bit and In The News by Gary Cox, Most of the rest of the newsletter is a review of software and hardware such as recent ASGARD Products. They also plan on going ahead with their version of a versatile 80 column card. Telecommunications Part One by Richard Lumpkin, TI-Base by Mike Curtis, Editor's Bit by Marshal Ellis. o

Regional Group Reports

Meeting Summary For August

Banana Coast	11/08/91	Sawtell
Central Coast	10/08/91	Saratoga
Glebe	08/08/91	Glebe
Illawarra	12/08/91	Keiraville
Liverpool	09/08/91	
Northern Suburbs	22/08/91	
Sutherland	16/08/91	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. 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

Our last meeting concentrated on the features of two software products, Page Pro and its related program, Postermaker.

An interesting bug was discovered with Page Pro, which related to the saving of LG fonts as pictures. Each time the picture was retrieved, the text was corrupted with additional characters from the font file.

Postermaker turned out to be a very user friendly print program for Page Pro pictures. Just point the arrow and select your options.

Some time was also spent in trying to adapt the dip switch settings on a recently acquired modem to meet our needs. No luck just yet, but hopefully better news by next meeting.

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

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.

AUGUST MEETING - 3rd AUGUST

The August meeting will involve demonstrations ofthe latest software as well as a whole host of other activities. See Ross Mudie's article elsewhere in this issue for more details.

****

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

September	11 August
October	08 September

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.

****

PARKING ACCESS FOR THE TISHUG MEETINGS AT RYDE
by Ross Mudie, 3rd August 1991

A number of us attending the meeting at Ryde Infants found that a fence had been erected between where the cars are usually parked and the school building where the meeting is held. The school is leasing part of its grounds to a nearby shopping centre for staff parking. As a result I explored a better method of access to the school. The school has a rear access via a parking area off ARGYLE ST. The school entry is not signed as the school, however there are some good indicators to the school entry. If you are approaching in Argyle St from Blaxland Rd, the entry is on the left, immediately after a large Safety House sign. If approaching in Argyle St from Princes St direction, the entry is on the right in a crossing area. From both directions watch out for the steel posts with red and white bands. The car park which you will enter from between these banded steel posts is adjacent to the school assembly hall.

SOME OF THE THINGS TO SEE AND DO AT THE NEXT MEETING
by Ross Mudie, 3rd August 1991

There are so many things going on at the TISHUG monthly meetings, it is hard to keep up with it all. At the meeting today I was very busy all day with a very interesting range of subjects including the train set project, assembly class, looking at a NTSC console with a minor problem, demonstrating a program that I am in the process of writing for a member and providing some extended basic programming concept ideas for another member. I also observed many other members busily engaged in demonstrations of all types and providing each other with help in a spirit of co-operation.

The train set group is exploring the computerisation of a small HO gauge model train set. This is an ideal project to be involved in if you are interested in model trains or controlling other electronic peripheral devices. The train set will be controlled via the WIRE I/O interface. The WIRE I/O is ideal for controlling robotic devices or anything where you need to interface from a number of wires into and out of the computer.

If you have a device that you want to control, come along to the meeting and learn. The train group will meet at 9am through to 11.30am on the next meeting.

The assembly class is currently working on assembly linked from extended basic. This small group is working with guidance to write small linked routines and, believe it or not, it is starting to make sense! The assembly group will meet from 12noon to 2pm at the next meeting.

There was a demonstration of the club's Bulletin Board System (BBS) at today's meeting. Only 2 people availed of this demonstration, but it will be available again next meeting from 12noon. The BBS demonstration is

continued on page 19