



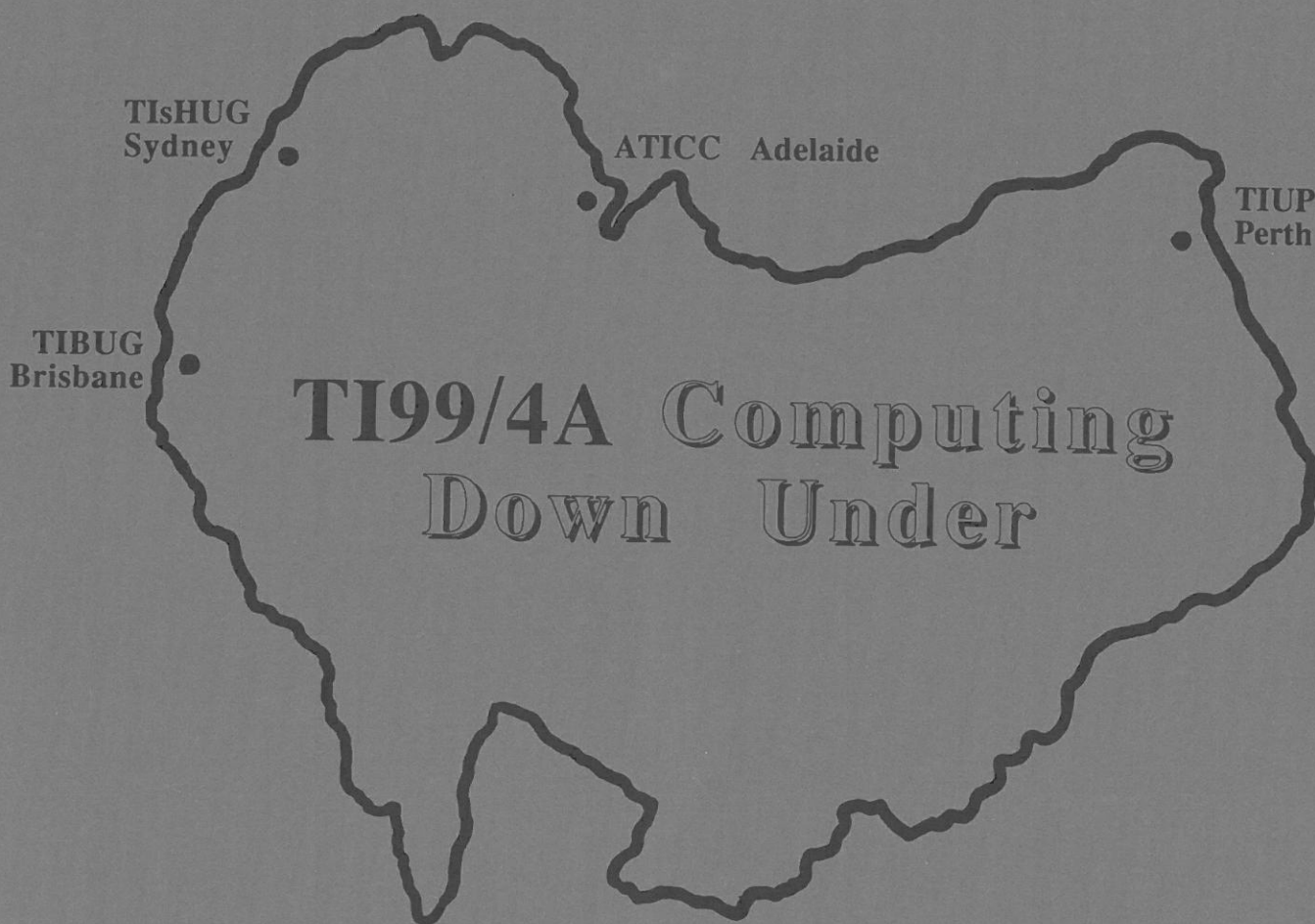
NEWS DIGEST

Focusing on the TI99/4A Home Computer

Volume 10, Number 9

October, 1991

Registered by Australia Post - Publication No. NBH5933



Sydney, New South Wales, Australia

\$3

TiSHUG News Digest

October 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 5th of October at Ryde Infant School, Tucker Street, Ryde. At 9 am, before the main meeting, there will be a meeting of all those involved in the model train project followed at 12 pm by the beginners' Editor Assembler class for all those interested.

Printed by

The University of Wollongong
Printery Services.

Index

Title	Description	Author	Page No.
Beginning Forth #10	Software hints	Raguse, Earl	19
Clock modifications	Hardware hints	Mudie, Peter	12
Co-ordinators report	General news	Warburton, Dick	2
Editor's comment	General interest	Relyea, Bob	1
Extended BASIC problem	Software hints	Relyea, Bob	15
Extended BASIC tips	Software hints	Relyea, Bob	7
Extended BASIC tips #11	Software hints	Swedlow, Jim	10
Letter to editor	Page Pro	Ball, Michael	3
Link-It #20	Software hints	Mudie, Ross	17
Look at CS1	String searching	Green, R.A.	18
Look at GPLLNK	Software hints	Green, R.A.	18
Multiplan exercises #8	Spreadsheet	Schlesinger, Herber	20
Program to type in	Itche worm		13
Program to type in	Typing	Harris, Daniel	14
Regional group reports	General interest		23
Secretary's notebook	Club news	Phillips, Terry	3
Software help	Software hints	Trott, Geoff	8
Telecommunications with TIW	Software review		6
Terminal emulator II tips	Software hints	Harris, Daniel	23
TI-Base tutorial #14	Data base	Smoley, Martin	21
TI-Bits #10	Software hints	Swedlow, Jim	9
TI99/4A world news	General interest	Peterson, Jim	16
Tips from the tigercub #62	Software hints	Peterson, Jim	11
TiSHUG shop report	Club news	Harrison, Percy	4
TiSHUG software column	Club software	Schreiber, Rolf	5
Treasurer's report	Club news	Trott, Geoff	3

Stop Press: \$ 40 off TIM 80 column card

We are currently taking orders for the TIM 80 column upgrade for \$165, a saving of \$40 off the normal price. Send the money with your order to Percy Harrison in the TiSHUG Shop before 14th October. The TIM 80 column upgrade is compatible with RGB analog monitors like the Wang monitors and Al Lawrence will be demonstrating his 80 column system at the October meeting.

Editor's Comment

by Bob Relyea

I cannot believe how fast this year is going by. It seems like one month (year?) rolls right into another one so fast that one can hardly keep up with all the developments. Software is coming to hand faster than I have a chance to review it. If anybody out there wants to review a bit of software then let Rolf or I know and we will keep you busy. You do not have to be prompted by us, however, if you have something around the house that has been reviewed then just write up an article about it (it does not have to be long) and we will see that it gets into the magazine. At least our 'problem' is a good one, in that, it is far better than not having new software. I see from one of Jim Peterson's articles that Asgard has reorganised their company to allow for faster customer service. When you consider the high quality of their software, that is certainly good news. See you all at the Buy Swap & Sell Day. o

Co-ordinator's Report

by Dick Warburton

We are approaching the end of 1991, and this year we will have our Annual General Meeting in December. We plan to start the AGM fairly early in the afternoon, so as to leave some time to do our other activities. We would like you to bring your family, and join in a Club get together after the meeting. Games machines will be up and running for the young people and there is plenty of room for kids to play in the school grounds. There will be a small charge to reduce our costs, and we hope it will be a fun time for family groups. Further information will be available at the next meeting.

This month's meeting will give some of you the opportunity to buy or sell some TI99/4A gear. I know that I always look forward to these days, usually to see if I can find a bargain. Rolf usually beats me to it. For those of you who want to expand your systems, this is the time to do it. You can usually pick up a good deal if you look around. There are sure to be some Peripheral Expansion Boxes and RAMdisks on offer.

An expanded TI99/4A is a most useful computer. I still get more fun from the TI99/4A than from other computers, but mainly because it is expanded with RAMdisks, a double density Multi-Function Card, a modem, and I use a colour monitor. It is a very useful machine. However I could not see myself struggling with tape recorders again, or being limited to using TI BASIC only.

If you are thinking about expansion, have a talk to Percy or myself, to see if the Club can help you in any way. There are different options available. If you want to expand your understanding of computers, perhaps with programming, at an elementary or advanced level, or would like to know more about how they work, talk to me about it. We can probably organize a group to suit you. Russell is organizing the next Tutorial Day in October, and he would like to know what you want. If you are interested in specific software, eg data bases or publishing programs, or adventure programs, I am sure that we can help you.

For those of you who want a little extra at club meetings, I am there from around 9 am. There is a train project being developed, there is an assembly class, and quite a few members are there to give you a hand with any problems you might have. Come early if you feel like it, and we may be able to form more Special Interest Groups. Last meeting, I was very pleased to see six machines up and running. The more we have, the more active people become. Let's aim at having a day where as many as possible of our members bring in their systems.

I was delighted to see Jim Mable's system available for display last meeting. Jim has successfully adapted a CGA colour monitor for TI99/4A use. Geoff tells me that it should not be too hard to adapt his monitor interface to drive a CGA monitor. They are getting cheaper, and some members could benefit from such an adaptation. If you come earlier on Saturdays we can help discuss such issues.

By the way, have you ever written an article for the TND? We need your help. What do you use your computer for? What programs do you find really helpful, at home or at work? Have you written any programs yourself? What games do you play? Do you have any queries about software or hardware? Do you have any points of discussion to raise, or any observations to make? Do you have any thoughts about the future for the club? Is there something you would like the directors to do for the club? Do you have any suggestions about how we can meet our members' needs more fully? Put your thoughts and ideas on disk, and send them to the Editor. This is your magazine, and we would like to know what you think. We still have a spark or two left, so let's see what we can do.

Don't forget that we are looking at running a TI Faire in Sydney in November 1992. A committee is being formed to organize the running of it. It is not yet finalized, and if you want to participate, contact me. You will be very welcome to join this group if you want to help. I suspect that those members who contribute the time and effort in this venture, will gain far more than they give. There will be the fun of working together and seeing the planning come to fruition. There will be a widening of ideas, and some new experiences. I am really looking forward to it already. I know we will all gain more by contributing to a common project, so come and join in the action. See you at the next meeting.

continued from page 22

and I would like a printout of all my Deposits. Since I know the Deposits do not have check numbers I told TIB to print everything that did not have a check number. DISPLAY CHKNO,DATE ;FOR DATE<"01/01/89" would display the original BALANCE and all records with no date. PRINT ALL ;FOR ((DATE>"02/29/89")) .AND.(DATE<"04/00/89")) would print only records dated March, (03). "The (;FOR) command is a very powerful tool."

```
* FORTSTI
CLOSE ALL
USE CHKBK
PRINT (Drft),(f)
PRINT ALL ;FOR ((DATE>"02/29/89").AND.;
              (DATE<"04/00/89"))
CLOSE ALL
PRINT (Drft),(E)
RETURN          ** This Works **
```

Here is a tip that is so valuable you should all give me money, but you can have the tip for free. Try to never input commands at the command line. Always create a Command File (CF), no matter how small or insignificant. I use MODIFY COMMAND FORTSTI to create FORTSTI which tests the print line you see in my text just prior to that. I tried the command four or five times before I got the result I wanted. By the time I changed the line three times I already forgot my old mistakes and started making them over again. For this problem I use another trick. I leave the old command line in the CF, but I place an asterisk at the beginning of the line so TIB will ignore it. Then I retype the new line below that for another try. At a later date I can type LIST DSK2.FORTSTI/C <E>, to get a printout of FORTSTI and see exactly which attempt worked and which did not. You do not have to issue printer commands in these little CFs, but you should always open and especially CLOSE the Db you wish to USE. First, this habit will allow you to see the name of the Db you USED with the command and more important, it will CLOSE the database for you. Many times I have gotten discouraged and changed disks or removed the disk and turned the system off while a Db was still open. There are also many other demons that will tear up your database if you leave it open while you are performing TIB high disk access commands such as COPY. Because I do not remember which commands are dangerous, I try to keep all my Dbs closed unless I am performing a specific task and once again I immediately close them. You will also notice that I leave myself notes inside the CF such as, This Works, Not Yet, This One's Junk, etc. One final comment. It took me seventeen hours to complete this tutorial from the first idea to the finished product. o

continued from page 10

Chris Bobbitt has announced that Harry Brashear has joined Asgard Software in the capacity of limited partner, with responsible for almost all filling of orders. He expects that this will eliminate the long delays that customers have experienced in dealing with Asgard, and that he will now have time to resume publication of Reflections/Asgard News and to concentrate more on new product development. Chris will continue to handle all customer service. o

Secretary's Notebook

by Terry Phillips

Spring is well and truly here and what a glorious day it turned out to be for the September meeting. Members in attendance numbered about 50 and while there was no formal meeting, poor Dick being struck down with a "bug" and losing a good part of his vocals, those that were there were entertained by a variety of demonstrations and the usual idle chat that seems to be part and parcel of meetings.

Coming meetings promise a smorgasbord. October will see another Buy, Swap and Sell day, and this will be held on Saturday the 5th, which by the way is the Labour Day long weekend. This may give the opportunity for our out of town members to get down to Sydney and perhaps pick up a few bargains.

November will be a full day tutorial meeting and club stalwart, Russell Welham, has been given the task of organising the various events. You may get a call from Russell asking to volunteer your time in an area of expertise.

December is of course Annual General Meeting time and with the formalities out of the way it will be time for a Christmas party, hopefully commencing about 4pm. This will take the form of a BBQ in the school grounds and the cost will be \$4 per family. For this small fee, food and drink will be provided. Why not make this a family day and bring them all along for a great festive afternoon.

Speaking of the AGM, it will be held on Saturday 7 December, and while full details and nomination forms will appear in the November TND, this preliminary advice is to let you know that applications for a position on the board of directors will close with the Secretary 3 weeks before that date, i.e. Saturday 16 November, 1991.

A big welcome to Stuart Moss of Cambridge Gardens. Stuart is the only new member to welcome this month. It is however a welcome back to Joe Bottemanne, Kerry Harrison and John Vandermeij. All three have "revived" their memberships.

The user list for the BBS for the month of August shows that there were a total of 113 calls, from 20 users, for a total occupancy of just over 20 hours. In an effort to stimulate interest in the BBS - and there has been some at meetings thanks to Peter Mudie and his demonstrations - watch out for a great new contest centering around the BBS. Member, Ian Mullins, will be the driving force behind this competition. Thanks Ian.

In the August Micropendium, there was an article by our own Daniel Harris on "RUN CSI". Good on you Daniel. Great to see locals getting a mention in overseas magazines.

Australia Post has advised, that due to a review of postal facilities in the Redfern area, and co-inciding with the opening of a new facility at Strawberry Hills, post office boxes currently at Redfern are to be re-located to Strawberry Hills. Our new box number address will be PO Box 1089 Strawberry Hills NSW 2012, but please DO NOT start using it yet. Further advice will appear in the TND when the actual move is made.

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

Treasurer's Report

by Geoff Trott

Income for August	\$1555.20
Payments in August	<u>\$588.10</u>
Excess of income over expenses for August	\$967.10

Letter to the Editor

Rolf has been at me for a long time now to submit an article for the news digest. So Rolf now you have got me to sit at the key board and prattle away!

Like probably a lot of TI99/4Aers, I seem to have spurts of usage of my computer. The most useful programme, and therefore, the one that is responsible for taking me back to the key board more often than not is Page Pro 99.

I am a member of the Shoalhaven Marine Rescue Association Ltd, and I am also the training officer for them. This work involves setting out training schedules, originating training manuals, and then presenting them in a form that is readable (i.e. not in my hand scribble!!). This is where Page Pro 99 has helped me no end.

In the training field it is always a good idea to break up information into small areas. There is nothing more boring than to have to wade through page after page of the written word. This is not the case when using Page Pro. The information can easily be broken up with the use of pictures, patterns, and large or small different style fonts.

The club has a good number of disks with different pictures and fonts. I have made use of the TIAP programme to draw my own pictures that I use to highlight some areas. I have used TIAP and Page Pro 99 together to redesign the Association's letterhead. This I am pleased to say came out very well. I have also used these programmes to make up a presentation to the Local Rescue sub-committee (this committee is made up of representatives of the state and federal police, local council, RTA, department of defence, SES, fire brigades, ambulance, and other volunteer rescue organisations). I needed to have this presentation looking as "professional" as I could. The results I obtained from my \$500 old computer was, I believe, equal to what the other organisations were putting out on their mega-buck machines.

There are always some things that we would like to see incorporated in the programmes that we use, and I am no different. These additions are no doubt easy for us to say, as we are not the ones that have to do the programming. What I have on my wish list is for Page Pro 99 to have a "tab" setting facility. A lot of my use involves writing beside a picture, and having to either enter and then arrow across, or arrow down and then back interrupts the flow; and once one is flowing one should not be interrupted. I would also like a means of viewing the entire page layout prior to printing. I do not need to see the words or view the pictures, the layout would be enough to avoid the paper wastage of printing out and then finding that it does not look quite right. I think I remember seeing the sort of thing I wish for on the "missing link" disk. That is only a small wish list to write, but a large one to programme!!

I also got asked by Rolf to print out and comment on the Medical Clipart disks now available from the shop. I have printed out most of them, the last few will have to await a new printer ribbon. Those that I have looked at are great. The detail is good, provided you use double density (this is why I ran out of ink). They are, in most cases, very large pictures. I tried using the Page Pro utilities reduction programme. This reduces the picture to a quarter of the original size. The result was not acceptable on those pictures that had writing on there as well. The other pictures were fine. The Page Pro 99 reduction programme on the utilities disk would be better if there were a choice of reduction sizes (another wish!!!). I did not try the "squeeze" horizontal and then vertical programmes; this may work as they only reduce by half?

continued on page 15

TISHUG Shop with Percy Harrison

My apologies to those members who asked for the Tris and Typewriter Modules which Rolf said would be available at the August meeting but at the time of writing this (a day before the September meeting) I still have not received either of these two modules and do not know when they will be available.

Over the last two months we have moved quite a bit of TI Hardware mainly to new members who have decided to dig their consoles out of the cupboard, get it running and expand the system to give them increased memory and Floppy Disk facilities. I would like to thank these members for their patronage and wish them many years of happy computing - welcome to the TI Club.

From time to time we manage to get hold of unwanted TI Hardware at very reasonable prices so if any member is planning to expand their system please check with me to see if I have the items that you are looking for. Also we keep a small stock of resistors, capacitors, regulators IC's etc at very competitive prices.

For those members who do not have Disk Drives we have a small supply of TI Games Modules which will plug into the port of your console and will give you many hours of enjoyment. These are available at about \$8.00 each including the instruction book so contact me to see what is currently available.

I am very disappointed with the response for Eprom Ram Card and Kits as many members who originally indicated that they would purchase these have not done so and consequently the club is left with about 13 cards and kits in stock - a dubious luxury that your club cannot afford particularly as the sales value of our total stock holdings now exceeds \$14000.00. If you want the club to continue to develop hardware to improve the performance of the TI Computer the please do the right thing by the club and honor any commitment you make concerning purchases as the quantity of items put into stock is based on the commitment made by the members prior to the club placing orders for the goods.

PRICE LIST

5.25 in. DSDD Disks (Boxes of ten)	\$6.00
5.25 in. HD Disks (Boxes of ten)	\$12.00
3.5 in. DSDD Disks (Boxes of ten)	\$12.00
5.25 in. DSDD Half Height Drive (New)	\$65.00
12 Volt AC Transformer	\$3.50
13 Volt Arlec Transformer	\$12.00
8.5, 17 Volt Transformer	\$25.00
60 VA Transformer	\$20.00
MFC Printed Circuit Board	\$30.00
MFC Kit (Disk Controller)	\$102.50
32K Kit for MFC	\$26.50
PIO/RS232 (single port) Kit for MFC	\$42.50
Combined 32K and PIO/RS232 Kit	\$60.00
Music Kit with PCB	\$65.00
32K Memory PC Board	\$7.00
Eprom Ram PC Board	\$45.00
Eprom Ramdisk Basic Kit	\$35.00
Funnelweb Eprom Set (3 Eproms)	\$36.00
TI Artist Eprom Set (2 Eproms)	\$24.00
32K Static Ram IC (62256)	\$10.00
8K Static Ram IC (6264LP)	\$5.00
74LS08 IC (quad Schottky)	\$0.50
1K Resistor	\$0.05
Exchange Console	\$30.00
ROS Version 8.14	\$12.00

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

COMMERCIAL SOFTWARE.

Artoons SSSD	\$12.00
Character Set & Graphic Design Cataloguer....	\$6.00
Character Set & Graphic Design I	\$12.00
Character Set & Graphic Design II	\$10.00
Character Set & Graphic Design III	\$14.00

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

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

continued from page 7

article by going through a few of them. The three options that I like the most are:

1. VALIDATE
2. BEEP
3. SIZE

You will notice on line 600 of my program that I used all three. Taking them in order, we will do VALIDATE first. Validate allows you to tell the computer to only accept certain characters following the prompt, such as just numbers (NUMERIC) or just letters (UALPHA). On line 600 I used UALPHA because I just wanted students names and I wanted to minimise an accidental mistake by typing in a number when I was tired and not paying attention. The BEEP option allows you to have the 'beep' sound when the cursor is placed on the screen as a form of alerting you. The SIZE option allows you limit the amount of characters that you can type in. In line 600, I used 15 so that I could type names in up to 15 characters long. The negative sign means that the input field is not blanked before data is inputted. You can also include an ERASE ALL option with ACCEPT which serves the same function as CALL CLEAR. If ACCEPT is used with DISPLAY, then it is better to use ERASE ALL with DISPLAY otherwise the statement you used with DISPLAY will be scrubbed! An example with these options would be:

```
10 DISPLAY AT(3,4)ERASE ALL:"LENGTH?":L$
20 ACCEPT AT(3,11)VALIDATE(NUMERIC)BEEP SIZE(-5):L$
```

The best way to master these principles is to type them in and experiment by changing the numbers I used to see what happens. Mastering the DISPLAY and ACCEPT commands will do a lot towards making your programs (no matter how short and simple) look more professional.

Next Month - Making a Menu.

TiSHUG Software

Column by Rolf Schreiber

I would like to point out something that some (most?) people seem to have missed reading about, on page six of August's TND, under the heading "John Birdwell - In Memorium". We have released a memorial edition of John Birdwell's "Disk Utilities Vn 4.2", complete with all documentation, for \$11.00 (DSSD disk), or \$12.00 for two SSSD disks, available through the TiSHUG shop or by mail order from Percy Harrison. Ten dollars from each disk sold will be sent to the John Birdwell Memorial Fund in the USA. I think the project is worthwhile, since all the proceeds will ultimately benefit the greater TI99/4A community. I have already contributed; what about you?

I received fifty disks of public domain software from Jim Peterson last month. A lack of space prevents me from listing all the disks, but I have catalogued them and will make the list available on disk at the next meeting.

The latest version of Funnelweb, available since last month, is V4.40 (up from V4.31), while the latest version of TI-Base that has just come in is V3.02 (up from V3.01). Each disk is available from the shop for \$2.00, and you must be a registered TI-Base owner to receive the V3.02 upgrade.

No answer yet from Thomas Opheys about the TI99/4A emulator for PCs, but finally good news about some hardware that we have been waiting a long time for. The MIDI interface has finally arrived from Mike Maksimik. However, please note that the current version (V2.3) does not allow all the features promised. Version 3.0 (expected shortly) will be available as a free upgrade to anyone who buys V2.3, and will allow both playback and recording of MIDI instruments. Gary Wilson will be happy to explain the features of the MIDI interface to interested people at the next meeting. The cost of MIDI MASTER 99 is \$75.00 each, or \$80.00 each if we order a minimum of seven units.

I received a phone call from Garry Christensen from TIBUG, the Brisbane TI99/4A User Group, a few days ago. He has been able to arrange for a substantial discount on the new TIM (TI Image Maker) 80 column upgrade from Gary Bowser of OPA in Canada. We need to order a minimum of about ten units to get \$40 knocked off the price of each unit.

Read last month's article by Al Lawrence about how easy it is to install TIM into the console, and how good the results are. TIM comes with an upgraded GROM chip set (called SOB, which sells separately for about \$45) and the complete package will only cost \$165.00 instead of the usual \$205.00. I am told that this is a limited offer, so do not take too long to decide if you want one!

TIM is compatible with RGB analog colour monitors, such as the Wang colour monitors owned by about sixty of our Club members, and it only requires a simple cable and a small modification to the monitor for it to work. So the upgrade to 80 column word processing is only going to cost those people another \$165.00. I know of people who bought other computers costing thousands of dollars, just for the convenience of 80 columns! I am definitely going to order one myself and both Geoff and Lou are thinking very hard about it, so we only need about seven others to make this one a goer.

Software Releases for October 1991

DISK A184 is a re-release of Basic Builder, a utility by Paolo Bagnaresi, which converts a DIS/VAR 80 listing of a BASIC (or XB) program back into a runnable program in a single operation. The disk comes with very extensive documentation files and the source code is available on request. Highly recommended.

DISK A453 is 'The Nutcracker Suite' (Opus 71A) from Harrison Music. To quote Bruce Harrison's own words "They're all there, from the sparkling Overture through to the beautiful Waltz of the Flowers. These pieces include special instrument sounds such as Celeste, Harp, Flute, Basses Horn, and Clarinet, as well as the Piano. Our best seller. Geneve compatible. SSSD. \$5.00" Due to his generosity, Bruce Harrison is making his software available to us as FREWARE, so the price to our Club members (provided that they live outside USA) is our normal price of \$2.00. If you like music, then this disk is a must!

DISK A462 is called REDISKTI and is a modified version of the REDISKIT track copier that was originally written by James Schroeder of the Chicago User Group. The modifications were made by Mike Dodd, and now allow those of you with only a TI disk controller card to copy disks very quickly by cloning them. The program will clone an entire SSSD disk in about 32 seconds, including formatting the destination disk, while DSSD disks take about twice as long. Note that a dual drive setup is required for this program. A second version for CorComp disk controllers is included on the disk.

Commercial Software

Some new software is available from Notung Software, a software company recently started by Ken Gilliland and Ray Kazmer. Ken and Ray are both very talented programmers and all the programs marketed by Notung Software are written by themselves. The quality of the software is excellent from what I have seen so far. Below is a list of their software that is available from the TiSHUG shop:

TI-Casino (DSSD disk).....	\$20.00
Son of the Disk of Dinosaurs, two disks.....	\$16.00
Star Trek: The Next Generation Calendar (DSDD).....	\$14.00
Fonts and Borders I (for TI-Artist).....	\$10.00
Fonts and Borders II (for TI-Artist).....	\$10.00
Fonts and Borders III, two disks (for TIA).....	\$12.00
The Ring Companion, two disks.....	\$12.00
Filmlib for TI-Base.....	\$10.00
BABA Brewery Beer Labels.....	\$10.00

Details about this software:

TI-Casino is really eight games in one. The software was designed to simulate a Casino gambling experience without losing all your money and the shirt off your back. For the first time it is possible to play multiple games with the same money, and even to cash in at the end of a session. The games all require a joystick and include Acey Ducey, Baccarat, Blackjack, Craps, Draw Poker, Keno, Roulette, and Slot machine. Very effective use of graphics (in brilliant colours), and sound makes TI-Casino stand out from other, similar games of chance.

TI-Casino is also available on two SSSD disks and comes with a comprehensive 48 page manual which contains full details on each game including its history, playing strategies and the full story behind the creation of this entertaining software package. If Kerry Packer owned a TI99/4A, this is the game he would be playing all the time.

Son of the Disk of Dinosaurs is a follow up package to Disk of Dinosaurs that Ken Gilliland created for Asgard Software. The first disk contains sixteen instances of dinosaurs that can be used with TI-Artist to create realistic pre-historic scenes. The second disk, called the program disk is menu driven and comes up with the following selections from Extended BASIC:

- A. Dinosaur Facts
- B. Animated Cartoon
- C. Paleontology Quiz

The first option allows you to read about most of the dinosaurs included on the instances disk and why

they became extinct, using a custom designed 40 column text file viewer. The information can also be read with any TIW program, or printed out.

The second option alone is almost worth the price. The animated cartoon is delightful to watch and shows the capabilities of our computer. The sequence contains 62 frames and was processed with the Comic Show editor, which featured as one of our software releases several months ago (DISK A249).

The paleontology quiz teaches you about dinosaurs. The object of the game is to build a complete dinosaur skeleton by answering all ten questions correctly. Each time a correct answer is given, another bone appears. Different question and answer modules will be available in the future. A twelve page manual accompanies this two disk set.

Star Trek: The Next Generation Calendar will print 13 full size pages of Star Trek characters and their corresponding months, and a cover. The calendar covers the months from July 1991 to June 1992 and each picture has been scanned from photographs and ported over to the TI99/4A using some fairly elaborate techniques.

The program comes with a stand alone print routine compatible with most Epson type dot matrix printers, but the pictures can also be loaded through Pix-Pro, using the McPix feature. Individual months, or the whole year, can be easily printed and the results look very professional. An 8 page manual comes with the software which is available on one DSDD, or several SSSD disks.

Fonts and Borders, Volumes I, II, III are three volumes of TI-Artist compatible fonts and instances. Included in Volume I are 7 TIA fonts and over 15 border/design instances. Volume II contains 6 larger borders, the Artist font, a brick font and a very special instance alphabet. Volume III comes on two disks and contains completely new artwork for use with TI-Artist Plus! Each software package comes with a manual showing all the artwork. Buy all three packages for \$30.00, a saving of \$2.00 off the usual price.

The Ring Companion is a two disk set of music examples, artwork and informative text about Richard Wagner's "Der Ring des Nibelungen". The program is menu driven from Extended BASIC, while the artwork can be viewed or printed out using TI-Artist. An 8 page manual explains all about using this software package.

Filmlib for TI-Base is a complete video librarian that not only tells you on which tape (and where on the tape) the movie you are looking for is, but also up to three of the stars, the director, the producer and even who wrote the music score. You can also input the year of release, the running time, and give each film a rating as well as a synopsis. The DO MENU directive can not only print out a movie list, but can search out actors, directors, or even all the movies from a particular year or belonging to a particular category. If you own TI-Base and do not use it, or know what to use it for, then this program could change all that. A 12 page manual helps to get you started.

BABA Brewery Beer Labels: You do not have to be a beer drinker to buy this program, but it sure helps. The disk contains artwork for people who like to brew their own special brews (you cannot get more special brews than "Woolly Stout" or "Scotch Fleece"!) and even has the recipes as well. Four B&W labels and 3 five-colour labels (using a colour printer) are included and you need TI-Artist for viewing or printing them out, while the Extended BASIC load program allows you to easily look at the recipes. The 8 page manual shows all the artwork and explains how to get the best out of the program.

Other commercial software includes:

XB:BUG (programmer's utility).....\$22.00
XBasher (programming aid).....\$15.00
Picture-It, (TIA_I to XB, TIW conversion).....\$15.00

Now for some details about this software:

XB:BUG is a tool for development of programs in TI Extended BASIC. It comes in various versions to allow loading into different areas of memory. You can jump into XB:BUG at any time by simultaneously pressing the CTRL and SHIFT keys. Its great value is in helping you track down errors in the programs you are writing (or modifying). It allows you to look at just about everything that is going on in your program while it is running, without disturbing it in any way. It can show information on graphics, variables, subprograms, files, data, and the program in general. XB:BUG comes on a SSSD disk with a 16 page manual.

XBasher's function is to reduce the size of an Extended BASIC program. It accomplishes this by shortening variable names, combining lines together, removing REMs and using many other less obvious techniques. XBasher has been designed with many options, so that you can decide which of its many features to employ on a particular program. XBasher works on files saved in merge format (DIS/VAR 163), obtained when you save a program to disk with the command:

```
'SAVE DSKn.filename, MERGE'
```

This software comes on a SSSD disk with an eight page manual.

Picture-It allows you to convert TI-Artist instances into runnable Extended BASIC code, or into a TI-Writer file for printing out through the formatter. You can also convert small pictures into movable sprites for use in your own programs.

Other options include viewing TIA instances on the screen, merging two instances into one and creation of banners from instances. Included with the program are plenty of examples of instances to play with. A second disk (DSDD) contains 26 fonts from A to Z, suitable for creating letterheads with TI-Artist. Documentation consists of three pages of text.

This software will be available for sale at the next meeting, but be warned that only one copy of each item has been purchased. Anyone buying this software will not be disappointed, since it represents good value. o

Telecommunications with TI-Writer

from TI-Writer Supplement, Chicago Users' Group, USA

Ever want to use TI-Writer for sending information via a modem? The Magnetic Users Group, North Andover, Ma. has discovered a way to do it.

SENDING PARTY: - Compose text as usual with the Editor of TI-Writer. When it is time to save it to disk, use PRINT FILE with no control characters:

```
PF <ENTER> C DSK1.README <ENTER>
```

Exit the EDITOR section of TI-Writer and enter the FORMATTER section.

```
FILENAME = DSK1.README  
DEVICE NAME = RS232.LF  
USE MAIL LIST = N  
WHAT PAGES = (ALL)  
NUMBER OF COPIES = 1  
PAUSE AT END OF PAGE = N
```

CHECK - to be sure that the RECEIVING PARTY is ready before you toggle the sending modem on. When everything is ready, hit the final <ENTER>.

RECEIVING PARTY: - Enter the EDITOR section of TI-Writer and prepare to LOAD files:

```
LF <ENTER> RS232.LF continued on page 8
```

Extended BASIC Tips

by Bob Relyca

Each month to the end of the year, at least, I plan on putting in articles of varying lengths about a particular aspect of Extended Basic programming. This month will be a short one about program titles. There are two ways of going about it:

1. Using the PRINT statement, or
2. Using the Display AT statement

I used to use the print statement several times in succession until the whole title page appeared but during the programming of the Fish program that I did last year it was pointed out that the DISPLAY AT statement was also used for this and had at least two distinct advantages-

1. It did not scroll up like the PRINT statement does, and
2. It comes up on the screen a bit faster.

The comparison of the two methods are as follows:

```
230 DISPLAY AT(4,6):"*****"; * *;" * *
GRADE *;" * *;"
240 DISPLAY AT(8,6):"* STANDARDISER *;" * *;" * *
*;"
250 DISPLAY AT(11,6):"* *;"
260 DISPLAY AT(12,6):"*****"
```

```
230 PRINT "*****"
240 PRINT "* *;"
250 PRINT "* GRADE *;"
260 PRINT "* *;"
270 PRINT "* STANDARDISER *;"
280 PRINT "* *;"
290 PRINT "*****"
```

You will have to respace some of the '*' in the DISPLAY AT example above as I 'scrunched' them to fit onto the size column that I have to work with.

In each case you have to remember that there are 32 columns available on the screen so you have to do a bit of counting to get it right. Try each of the above methods if you are new at programming and see what you can do.

PLACING THE CURSOR USING 'ACCEPT AT'

There are several ways of being able to input information in a running Extended Basic program. The two most common ones are:

1. The Input command,
2. A combination of the Display At command and the Accept At command.

The most common of the two would, undoubtedly, be the first one. A typical example of using the Input command in a program would be:

```
10 INPUT "ENTER THE LENGTH:";L
20 INPUT "ENTER THE WIDTH:";W
30 INPUT "ENTER THE HEIGHT:";H
40 PRINT "VOLUME =",L*W*H
```

This is a mini-program in its own right that you put information into and get an immediate result- the volume. The purpose of the ':L' part of line 10, for example, is that once you have inputted your data following the prompt, you are telling the computer to name it 'L'. Everytime this letter (L) appears in the program, such as in the L*W*H the computer will replace it with the number that you inputted. If you want to input words (called a string) then you would have to add

the '\$' after the letter. You can add to the above program by using units, such as cm for each dimension, and cubic cm for the volume. How close the cursor appears to the ':' depends on how close to the ':' you placed the '\$'. You do have some flexibility with the printing of the word 'Volume' and the answer to L*W*H by using TAB, but beyond that there is not too much you can do. If you key in this program and have a 'play' with it you will notice that the prompts executed by the INPUT statement are always at the bottom of the screen.

The second way of inputting information would be by using the DISPLAY AT command in conjunction with ACCEPT AT. Have you ever noticed while running a program how the cursor can appear to be placed almost anywhere by the programmer and that the prompts are located anywhere on the screen? Also, at times you can only input certain types of characters or the computer gives its famous honk? This is all accomplished by this second method. Some people are put off by its appearance but it is not really too hard and it is certainly more professional-looking as well as more flexible. If you have had a look at the Grade Standardiser program that I had published in the August TND (pages 13 & 14), then you see examples of its use. A good example of what I am about to explain is found on lines 570 to 620 of the section on Creating a New File.

First, let me briefly explain how to use each command, starting with the DISPLAY AT first. In your Extended Basic manual you will notice a lot of information written after DISPLAY AT at the top of the page under the section 'format'. My use of it here will be kept to basics. If you want to have any statement (whether it is to be used ultimately as a prompt or not) appear any place you like then you must type in:

```
10 DISPLAY AT(row,column):"statement"
```

The numbers you give for (row,column) will ensure that the beginning of your statement will start at the (row,column) you specify. An example of this in my program would be lines 270 & 280. If you want to use the statement as part of prompt then you must 'label' it like we did above with the INPUT statement. It would then look like this:

```
10 DISPLAY AT(row,column):"LENGTH?";L$
```

As with the INPUT statement, the L\$ serves as a label for use later if you want to do something with the information inputted, like including it in a formula for calculation. If you are going to use DISPLAY AT as an input statement then you must use ACCEPT AT as well. This command is very interesting because there is a lot you can do with it. As with the DISPLAY command, all the brackets, etc written after ACCEPT under the section 'format' in the Extended Basic manual just about turn you off. So again, I will try to go slow and explain the basics. First of all ACCEPT will allow you to place the cursor any place you like. To use the example above, you would want to place the cursor a bit after the 'LENGTH?', so what you must do is to count how many characters the word and question mark make up. In the case of LENGTH? there would be 7, so you add this onto the COLUMN number you used and then add on one or two more so that the cursor would appear just a bit after the prompt 'LENGTH?'. An example would be:

```
10 DISPLAY AT(3,4):"LENGTH?";L$
20 ACCEPT AT(3,11):L$
```

This will place the cursor right next to the '\$'. If you wish to have it a bit further to the right then instead of using 11, use 12 or 13. Now, how did I arrive at the number 11? Well, the column number is 4 and there are 7 characters in 'LENGTH?'. Adding 4 + 7 gives 11. Remember that the 'L' of LENGTH? starts on column 4 so if you do your sums properly, the '?' will land on column 10. This means that the cursor will be right next to it on column 11. Basically, that is how you go about it. There are a few nice options that you can include with ACCEPT so I will conclude this months
continued on page 4

Software Help

by Geoff Trott

John Hagart wrote to me from far north Queensland to ask for help with some programs he was typing in from the TND. The first one was called "Barbie Boutique" by Lucie Dorias and appeared on page 13 of the July 1991 TND. When John typed it in, it gave a DATA error on line 560. One of the problems with finding out what is wrong with an Extended BASIC program is that there can be multiple statements on the one line separated by "::". An error in a line could be generated by any of the statements on the line. In this case, the line was:

```
560 READ DR(X),DC(X),SZ(X):: X=X+1 :: IF X<16 THEN 560
ELSE CALL SPRITE(#1,60,16,CR,CC):: CALL
CHAR(68,A$,71,B$)
```

The first three statements on the line form a loop which reads from data statements the values for the three variables DR, DC and SZ until the index X is incremented to 16. A DATA error is saying that there were not enough data in the DATA statements. This means that something has gone wrong when typing in the DATA statements. Some idea of how serious a problem it is can be obtained by finding the value of X when the error occurs. Run the program until the error occurs and then enter the statement:

```
PRINT X;DR(X);DC(X);SZ(X)
```

If the printed value of X is 15, only one or two values are missing. Probably a comma is missing. If X is less than 15, a whole line or two could be missing. Comparing the values of the variables printed with last values in the last DATA statement gives a bit more information. I have typed this program in and managed to get it to work (although I have not worked out how to play the game) so what is printed does not have errors in it, even though it looks a bit strange in places.

John managed to get past this error and then ran into a SYNTAX ERROR on line 720. This program turns off the pre-scan, which means that it may not find some errors until the program is actually run. On the end of line 170 is "!@P-" which turns off the pre-scan, while at the start of line 820 is "!@P+" which turns the pre-scan back on again. If you get an error like SYNTAX ERROR in a program like this, you should remove these two and allow the full error check to take place. I do not know what the problem is here but I would guess it is in the number of parentheses or a space in the wrong place. This line contains two statements which could be put on two separate lines to isolate which is in error. For example:

```
720 GOSUB 760 :: CALL SPRITE(#N,C,UC(AR),DR(N),DC(N))
```

could be changed to:

```
720 GOSUB 760
721 CALL SPRITE(#N,C,UC(AR),DR(N),DC(N))
```

The error John received could also be as a result of the previous error not putting the correct DATA in the variables UC, DR or DC.

The second program was printed on page 28 of the July 1989 TND and is a game called Mini Mancala. This is a well known game which was enhanced by Wade Bowmer. This program gave John an error message of BAD VALUE in line 130. Line 130 is:

```
130 FOR I=0 TO 7 :: READ A$,B$ ::
CALL CHAR(I+48,B$,I+96,A$,I+104,A$,I+112,A$,I+120,A$)
:: NEXT I
```

This is a very similar problem. The READ statement is reading in DATA statements (at the end of the program) which are used to set up character sets. There should be 16 string variables in the DATA statements which should look like character patterns. If the

values of I, A\$ and B\$ are printed when the error stops the program, it should be easy to identify what has gone wrong. I imagine that a comma has been missed as there are two in a row at one point. When I type in the program, I missed a comma and received the same error message.

Errata

There actually is an error in the listing of the above program as printed. It is in the last line of the program, which is a DATA statement. You will notice that the line does not start in the correct column and has an odd number of double quotes. It looks like:

```
500 DATA "'gf hon","a2e i2m","bcd jkl",,,
~, "q2u y2}","rst z{!"," >"
```

but should have looked like:

```
500 DATA "'gf hon","a2e i2m","bcd jkl",,,
"pwv x ~","q2u y2}","rst z{!"," >"
```

The reason for the problem is that the character between the x and the ~ is not a space but the character (ASCII code 127). This caused the printer to delete the line up to that point so that the printout showed just the rest of the line. When typing in that line of the program, you must replace this space by FCTN[V], which will put the character in the correct place.

Other errors found by John Hagart are in the April 1991 TND on page 10 in the left hand column, line 210 should read:

```
210 ANG=RND*360 :: IF ANG<12 OR ANG >350 THEN 210 ELSE
IF (ANG>160)AND(ANG<210) THEN 210
```

In the right column of the same page, line 220 should read:

```
220 X=LA*X*(1+X)^(-BETA)
```

In the next month's issue, May 1991, John found another two errors. On page 8, in the left column in the middle of the page, line 140 should read:

```
140 G=INT(RND*140)+60
```

On the same page in the right column, line 160 should read:

```
160 FOR X=TOP TO BOTTOM STEP 1/VS :: FOR Y=LEFT TO RIGHT
STEP 1/HS :: TEMP=INT(((X/32)^2)*Y/32)
```

All of these are problems with the formatter where * followed by two digits does not print and ^ is the defined space character and so prints as a space. These can be avoided with care or by using transliterates (as I have done here), but it does make printing program listings through the formatter a bit of a problem.

It is nice to hear from someone who types programs in. I hope everyone who has typed in a program and failed to get it to run will follow John's lead and write in to ask for help, or bring the problem to a meeting. This is one of the best ways to learn to program. O

continued from page 6

When the sending party is ready to send, wait until you hear the squeal of his modem, then toggle the receiving modem on and hit the ENTER key. You will not see anything on your screen, but the lights on your expansion box will flicker. Then, if everything has been done correctly, the file will suddenly appear. Then SAVE FILE to your own disk in the usual manner.

If you do not get your timing correct, you may lose all or part of one line. You can recover most of it with "OOPS!" (CTRL1), but it is easier if you make sure your text starts with one or two blank lines. O

TI-Bits Number 10

by Jim Swedlow, CA USA

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

ALPHABET SOUP

We are constantly bombarded with acronyms. This list is provided as a public service to help you sound like you know what you are talking about!

AI - "Artificial Intelligence" - trying to make computers think like people. A science in its infancy.

ALGOL - "ALGOritmic Language" - a programming language.

ANSI - "American National Standards Institute"

APL - "A Programming Language" - an interactive programming language that is well suited for handling complex operations on arrays.

ASCII - "American Standard Code for Information Interchange" - and you thought that the II was version two! Pronounced "ask-key".

BASIC - "Beginners All purpose Symbolic Instruction Code" - some suggest that the acronym came after the name!

BBS - "Bulletin Board System"

BIOS - "Basic Input Output System" - the part of CP/M or MS-DOS that allows the CPU to communicate with the keyboard, screen, printer, etc.

C - a programming language developed at Bell Labs. Its predecessors were B (1970) and BCPL (1967).

CMOS - "Complementary Metal Oxide Semiconductor" - a type of IC noted for its low power consumption and resistance to damage. Often used in portable computers. IC's of this type usually have the letter C in their name.

CP/M - "Control Program for Micro-computers" - a family of operating systems that would have been the standard for business had IBM not used PC-DOS (see MS-DOS).

CPU - "Central Processing Unit" - the part of the computer where arithmetic and logical operations are performed and instructions are decoded and executed.

CRT - "Cathode Ray Tube" - the screen on your TV or monitor.

EOF - "End Of File"

IC - "Integrated Circuit" - a chip with many miniature transistors and other devices.

ISO - "International Standards Organ- izaton"

LISP - "LISt Processor" - a programming language often used for AI applications.

MODEM - "MODulator-DEModulator" - a device that encodes and decodes data for transmission over telephone lines, coaxial cable, fibre optics, microwaves, etc.

MS-DOS - "MicroSoft Disk Operating System" - the operating system for computers that use the 8086 or 8088 microprocessor family. MS-DOS is sold by IBM as PC-DOS for the IBM PC.

PROM - "Programmable Read Only Memory" - a chip that can be programmed once but not revised. EPROM [Erasable PROM] chips can be erased and reprogrammed.

TTL - "Transistor-Transistor Logic" - a high speed IC that is often used for input-output devices (a TTL monitor, etc).

WYSIWYG - "What You See Is What You Get" - brought to its current potential by the Mac, this means that your item appears on your CRT exactly as it will look when it is printed.

QUOTES OF THE MONTH

"The optimist proclaims that we live in the best of all possible worlds; the pessimist fears that this is true."

---James Branch Cabell

"There are moments when everything goes well; do not be frightened, it will not last."

---Jules Renard

"It is not possible to make things fool proof; fools are too ingenious."

---Edsel Murphy

COMPILED, ASSEMBLED AND INTERPRETED Or, Why BASIC is slower than Assembly

A computer language is what you use to tell your computer what to do. It is a common vocabulary. If you have done any programming, you know that your computer believes this language literally.

Your computer do not speak BASIC or Assembly Language. It speaks Machine Language (which is code that the CPU can execute directly).

When a BASIC program is running (also called during 'run time') something called a BASIC interpreter acts as a middle man between the program and the CPU. As each line executes, the interpreter reads the instructions and translates them to Machine Language. This takes time.

In Assembly Language, you write a source program using the Editor and then use the Assembler to assemble it into Machine Language. That is why the module is called Editor/Assembler. When you run an assembled program, execution is much faster as there is no need for an interpreter.

A compiled program is a hybrid of these two. You write your source program in a higher format. The 'higher' a language is the closer it is to English. The 'lower' it is, the closer it is to Machine Language. BASIC is a high high level language while Assembly is low level.

The language called C looks somewhat like BASIC but compiles into Machine Language.

Now you know.

WORD OF THE MONTH

ALGORITHM: An algorithm is a sequence of instructions that tell how to solve a particular problem. An algorithm must be specified exactly, so there can be no doubt about what to do next, and it must have a finite number of steps. A computer program is an algorithm. Some programs are so complicated that there is no algorithm to solve them.

THE COMPUTER SHOPPER

This is a thick magazine. Most of it is computer related ads - mainly for XT clones. It does, however, have a TI column - and a good one. You can usually find this it at computer related stores (try Software, etc in Westminster Mall). Subscriptions are about \$15 (call 1-800-327-9926 during business hours).
o

XB tips Number 11

by Jim Swedlow, CA USA

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

PRINT SEPARATORS

There are any number of ways to display text on your screen. In this and the next two items, I will cover some ideas that might help you. All of these will also work in BASIC except where noted.

Suppose that you want to display menu choices at screen lines 5, 7, 9 and 11. You could use four DISPLAY commands:

```
DISPLAY AT(5,1):"FIRST LINE"  
DISPLAY AT(7,1):"SECOND LINE"  
DISPLAY AT(9,1):"THIRD LINE"  
DISPLAY AT(11,1):"FOURTH LINE"
```

Another way to do this is to combine your DISPLAY's into one command:

```
DISPLAY AT(5,1):"FIRST LINE": :  
"SECOND LINE": : "THIRD LINE": :  
"FOURTH LINE"
```

Note that the print separator between lines is : . I have seen (and used) : "" . The double quote was inserted to tell the computer to display a blank line. This is not necessary, : : works just as well and uses less memory. In XB you must type a space between the colons or your 4A will think that you are inputting a line separator.

This also works with PRINT statements in both BASIC and XB.

PROMPTS FOR INPUT

You have probably used a prompt with the INPUT command like:

```
INPUT "INPUT NAME":N$
```

The limitations here are frustrating. You cannot use a string variable, if you want to leave blank line then you must use a lot of spaces, etc, etc. Sometimes I use a PRINT statement for the prompt and then follow with INPUT. By accident I discovered that INPUT respects any trailing print separator on the preceding PRINT command:

```
PRINT A$;TAB(20);  
INPUT B$
```

The INPUT ? will appear at the 20th column.

PRINTING STRINGS

If you are printing two strings, you would normally use a semi-colon <;> between them. When you do this, your 99/4A looks at the current location on the screen and the length of the next print item. If it is longer than the space left on the current line, the string starts at the beginning of the next line.

This can result in annoying blanks in your right hand margin. There is another way:

```
PRINT A$&B$
```

Regardless of the length of A\$, B\$ will start in the next empty column. This may or may not give you the result you wanted, but it does give you options.

DISK LABEL

This month's program is a revision of one in our library under the same name. Essentially, it reads a

disk directory and prints a disk directory in a size that will print on a standard 3 1/2" x 1" computer fanfold label. This upgrade has some features not available in the one in our library: it will read a disk in any drive; it will print more than one label per disk; it prints the file type as well as the file name; and, it prints all files on the disk.

The program requires a GEMINI 10X or compatible printer.

Enjoy!

```
100 ! DISK*LABEL  
110 ! VERSION XB.2.0  
120 ! BY JIM SWEDLOW  
130 ! BASED ON A PROGRAM BY P.C.B. AND W.A.R.  
140 ! 3 AUG 85  
150 ON WARNING NEXT :: B,@=1 :: CALL CLEAR :: DIM F$(129)  
,T$(129),Y$(4):: FOR I=0 TO 14 :: CALL COLOR(I,16,@)::  
NEXT I :: CALL SCREEN(5):: GOTO 160 :: CALL KEY :: A,C,C  
$,D$,E$,J :: !@P-  
160 Y$(@)="DF" :: Y$(2)="DV" :: Y$(3)="IF" :: Y$(4)="IV"  
:: E$=CHR$(27):: D$="DSK1." :: OPEN #2:"PIO"  
170 DISPLAY AT(7,9):"DISK LABELER": : "Check the positi  
on of the label before starting.": : " Labels/Dis  
k: 1": " Drive: DSK1 ": " Comment:"  
180 DISPLAY AT(20,@)BEEP:"PRESS": "<P>Print, <C>hange or <  
Q>uit"  
190 CALL KEY(3,I,J):: IF J=0 THEN 190 ELSE IF I=81 THEN  
CALL CLEAR :: CLOSE #2 :: STOP ELSE IF I=80 THEN 210 ELS  
E IF I<>67 THEN 190 ELSE DISPLAY AT(20,@): : :  
200 I=@ :: : ACCEPT AT(14,20)SIZE(-2)VALIDATE(DIGIT)BEEP:B  
:: B=MAX(B,@):: ACCEPT AT(15,20)SIZE(-@)VALIDATE("12")B  
EEP:I :: D$="DSK"&STR$(I)&".": : ACCEPT AT(17,6)SIZE(-19  
)BEEP:C$ :: GOTO 180  
210 C=0 :: DISPLAY AT(20,@):"Initializing": : : OPEN #@  
:D$,INPUT ,RELATIVE,INTERNAL :: INPUT #@:F$(0),I,I,I ::  
T$(0)="FREE "&STR$(I)  
220 DISPLAY AT(21,@):F$(C);"";T$(C):: INPUT #@:F$(C+@)  
,I,J,J :: IF F$(C+@)<>"" THEN I=ABS(I):: C=C+@ :: IF I=5  
THEN T$(C)="Prog" :: GOTO 220 ELSE T$(C)=Y$(I)&STR$(J):  
: GOTO 220  
230 CLOSE #@ :: DISPLAY AT(20,@):"Printing": : : FOR A=  
@ TO B :: J=0  
240 PRINT #2:E$;"G";E$;"B";CHR$(2);F$(0);" ";C$;" ";T$(0  
);CHR$(15);E$;"S";CHR$(0);E$  
;"3";CHR$(12): :  
250 FOR I=J+@ TO J+8 :: PRINT #2:F$(I);TAB(12);T$(I);TAB  
(18);F$(I+8);TAB(29);T$(I+8);TAB(35);F$(I+16);TAB(46);T$(  
I+16):: NEXT I :: PRINT #2:E$;"@" :: IF C>J+24 THEN J=J  
+24 :: GOTO 240  
260 NEXT A :: FOR A=@ TO C :: T$(A),F$(A)="" :: NEXT A :  
: GOTO 180  
O
```

continued from page 16

use only a composite synch line. The article states that a monitor with a pixel size of 0.42mm is acceptable but a smaller size is definitely better. The monitor should do a screen size of 640 x 480 pixels, which is fairly standard on most VGA systems. Paying more for a higher size would be a waste of money, because TIM can only do 512 x 424. Among the monitors said to be compatible are the Commodore AMIGA 1080 and 1084, MAGNAVOX 8CM515, Thompson 4120, and SONY XBR series (KV1311CR) - the article does not definitely state that all of these completely meet the requirements.

The same article reports a compatibility problem between O.P.A.'s S.O.B. (Son of a Board) and the MYARC floppy disk controller.

TEXAMENTS (53 Center St., Patchogue NY 11772) is now buying, selling and trading used TI-99/4A and Geneve hardware, software, resource material and accessories. They will make a formal offer within two days if you mail them a list of items you want to sell. You can get a listing of items available for sale by sending them a self-addressed postpaid envelope, or by calling their BBS (516) 475-6463. Used hardware is sold with a 30-day warranty, and non-defective items can be returned within 15 days for a 20% restocking fee. continued on page 2

Tips from the Tigercub #62

by Jim Peterson, Tigercub Software, USA

Tigercub Software
156 Collingwood Ave.
Columbus, OH 43213
Dec. 1990

My stock of Tigercub Software catalogues is depleted and it would not pay me to reprint it. Therefore I have released all copyrighted Tigercub programs, except the Nuts & Bolts Disks, for free distribution providing that no price or copying fee is charged. All of my Tigercub programs have been added to my TI-PD library and are catalogued, by category, in TI-PD catalogue #4.

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

My TI-PD library now consists of 452 disks of fairware (by author's permission only) and public domain, all arranged by category and as full as possible, provided with loaders by full program name rather than filename, Basic programs converted to XBasic, etc. The price is just \$1.50 per disk(!), post paid if at least eight are ordered. TI-PD catalogue #3 listing all titles and authors, is available for \$1 which is deductible from the first purchase.

According to Charles Good, running a program containing CALL SAY on a beige console without the speech synthesizer attached will cause a lockup. On a black and silver console, there is no lockup but program execution can be greatly delayed. To avoid that, CALL PEEK(-28672,@) at the beginning of the program and add IF @=96 before each CALL SAY (remember that, IF causes program execution to skip to next program line if not true!), or IF @<>96 THEN to skip over the CALL SAYS.

In Tips #60 I presented a routine to find the lowest power of 7 which contains six 7s in sequence. My version took 24 minutes to find the answer on my TI-99/4A. Several users tried this on a Geneve. The NUTI News of the Nittany UG, Oct 1990 reports that on a 9640 (MDOS 0.97H) with TI XBasic loaded through GPL (speed 5) it ran in 11 min. 33.86 seconds, and with MYARC Advanced Basic V2.99A loaded through GPL it ran in 4 min. 58.62 seconds!

Now, from the TI*MES of England, here is a method using a level of math beyond my comprehension that will solve the problem on an ordinary TI in 6 minutes and 17 seconds!

```
100 ! FASTER WAY John Seager
110 CALL CLEAR :: DIM ELEM(26):: ELEM(0)=7 :: POWER,SS=0
:: DISPLAY AT(1,1):"7 TO THE POWER OF"
120 ELM=SS :: SS,CARRY=0 :: POWER=POWER+1
130 DIS$=STR$(ELEM(ELM)):: FOR I=ELM-1 TO 0 STEP -1 :: D
IS$=DIS$&RPT$("0",10-LEN(STR$(ELEM(I))))&STR$(ELEM(I))::
NEXT I
140 DISPLAY AT(1,19):STR$(POWER);"=" :: DIS$
150 FOR I=6 TO LEN(DIS$)STEP 6 :: IF SEG$(DIS$,I,1)<>"7"
THEN 190
160 FOR J=I-5 TO I :: IF SEG$(DIS$,J,6)<>"777777" THEN 1
80 ELSE DISPLAY AT(24,1):"ANY KEY TO CONTINUE"
170 CALL KEY(O,K,S):: IF S=0 THEN 170 :: DISPLAY AT(24,1
):: J=I
180 NEXT J
190 NEXT I
200 ELEM(SS)=ELEM(SS)*7+CARRY :: IF ELEM(SS+1)=0 AND ELE
M(SS)<1.E+10 THEN 120
210 CARRY=INT(ELEM(SS)/1.E+10):: ELEM(SS)=ELEM(SS)-CARRY
*1.E+10
220 SS=SS+1 :: GOTO 200
```

And if you think that is fast, the Autumn '90 edition of TI*MES contains a Minimemory program to solve the program in 2 SECONDS! And an assembly version that will search to the 10,000 power and find 52 strings of six 7's in an hour and a half!

Here is a puzzler for you. Can you figure out why that 1000-microsecond CALL SOUND is cut short?

```
100 CALL CLEAR
110 DISPLAY AT(12,1):"Filename? DSK" :: ACCEPT AT(12,14)
BEEP:F$
120 ON ERROR 130 :: OPEN #1:"DSK"&F$ :: STOP
130 GOSUB 140 :: RETURN 110
140 CALL SOUND(1000,110,0,-4,0):: DISPLAY AT(24,1):"CAN'
T OPEN FILE" :: RETURN
```

I recently programmed a diskfull of gospel songs, and in each one I used this formula to set up an array containing the frequencies for 3 octaves:

```
DIM N(36) :: F=110 :: FOR J=1 TO 36 :: N(J)=INT(F*.059
463094^(J-1)+.5):: NEXT J
```

At the end of each selection I put-

```
CALL INIT :: CALL LOAD(-31961,149)
```

I do not remember where I learned that one, but it clears the screen, sets all colours and characters to default, deletes sprites, and looks for a LOAD program on DSK1.

The LOAx program has a routine to play each song one after another, but one song crashed with a BAD VALUE error even though it had previously been OK. I found that this was the only song that actually used N(1). The value should have been 110 but it had somehow changed to 24263 which the program line multiplied by 2, therefore out of range.

I found that the routine was correctly giving N(1) a value of 110 the first time but after the CALL LOAD it always had the 24263 value. Substituting other values for 110, I found that any value was being multiplied by 220.5727273, rounded off.

Further experimentation revealed that the problem was being caused by the ^ (exponentiation sign, shift 6 on your keyboard, in case someone prints this through the Formatter!). So I wrote this little routine to experiment with:

```
100 FOR J=1 TO 10 :: PRINT 2^J :: NEXT J :: CALL INIT
:: CALL LOAD(-31961,149)
```

I saved that as DSK1.TEST and then wrote another one 100 RUN "DSK1.TEST", saved that as DSK1.LOAD, and then entered RUN "DSK1.TEST".

It printed out the proper values time after time, so I changed the 2^J to read 2^(J-1). The first time around, the first value was 1 as it should be - the computer will consider any number to the power of 0 to have a value of 1. But, the next time around, the first value was FO.57000101 !

That was not even a valid numerical representation, so I changed the formula to 2^(J-1)*2, expecting it to crash. Instead, it gave me a value of 441.140002 !

Further experimentation showed that 2^(J-1)+1 gave a value shown as 1<1.570001. Changing the +1 to +10 gave 1=0.570001 and to +100 gave 2<0.570001 ! So, poking a value of 149 into -31961 will cause any number taken to the power of zero to have a value of 220.5727273, which will be represented on screen in some apparently undocumented format - it is not even radix 100. I wonder if the fellows who built this computer could explain that!

This one requires the TEII module and the Speech Synthesizer. Want to make the computer so mad it will fuss and fume and cuss and mutter? Run this program and answer the prompt with 1.

```
100 CALL CLEAR
110 OPEN #1:"SPEECH",OUTPUT
120 INPUT X
130 PRINT #1:"/"&STR$(X)&
"&STR$(X*.17)
```

```

140 PRINT #1:"THIS IS THE SECRET METHOD OF MAKING THE CO
MPUTER SPEAK IN A WHISPER"
150 GOTO 120

```

Want to make it whisper to you? Answer the prompt with 0 or -10. Why did I get an INPUT ERROR when the strings in this routine got too long?

```

100 CALL CLEAR :: X=1
110 X=X*2 :: A$=RPT$("A",X):: B$=RPT$("B",X):: C$=RPT$("C",X):: D$=RPT$("D",X):: PRINT A$:B$:C$:D$
120 OPEN #1:"DSK1.TEST",VARIABLE 254,OUTPUT :: PRINT #1:A$:B$:C$:D$ :: CLOSE #1
130 OPEN #1:"DSK1.TEST",INPUT :: INPUT #1:A$,B$,C$,D$ :: PRINT A$:B$:C$:D$ :: CLOSE#1 :: GOTO 110

```

Thanks to Irwin Hott for the answer to that one. I do not think it is in the books anywhere, but the TI will not input multiple records in a single INPUT if the total number of bytes is too high - less than 154 for two records to less than 144 for six records.

I still think computers should be fun, so here is a quickie for the kids, or for the kid in you -

```

100 PRINT TAB(9);"QUICK DRAW": : : " How good a gunslinger are":"you?": : " Can you outdraw":"Deadeye Joe?": :
110 PRINT " Watch the countdown from 1":"to 10.": : " Wait for the gun...": : " Then hit any key FAST!! - ": : " - and HOLD IT DOWN": :
120 PRINT " I got down to 20 once - can":"you beat that?": : " Press any key to start"
130 CALL KEY(O,K,ST):: IF ST=0 THEN 130
140 CALL CLEAR :: S@=300 :: CALL CHAR(58,"009F91919191919F"):: CALL CHAR(42,"0000FCFE171F0707")
150 CALL KEY(O,K,ST):: IF ST=-1 THEN 150
160 CALL CLEAR :: FOR M=1 TO 10 :: CALL HCHAR(12,16,M+48):: FOR N=1 TO 100
170 NEXT N :: CALL KEY(O,F,X):: IF F=70 THEN 330
180 NEXT M :: CALL CLEAR :: FOR J=1 TO 500
190 NEXT J :: IF F=70 THEN 330
200 CALL KEY(O,X,ST):: IF ST<>0 THEN 330
210 CALL HCHAR(12,16,42):: FOR D=1 TO S@
220 NEXT D :: CALL KEY(O,Z,X):: IF X=0 THEN 240
230 GOTO 270
240 CALL CLEAR :: PRINT :: PRINT "YOU'RE DEAD!"
250 FOR D=1 TO 200
260 NEXT D :: GOTO 160
270 PRINT "OUCH!" :: IF S@<51 THEN 290
280 S@=S@-50 :: GOTO 320
290 IF S@<31 THEN 310
300 S@=S@-5 :: GOTO 320
310 S@=S@-1
320 PRINT S@ :: GOTO 250
330 PRINT "YOU CHEATED!" :: GOTO 150

```

I always wondered about those recipe programs. Does the cook lug the computer out to the kitchen to read the screen, or use a printer to make a hardcopy of a file that was keyed in from a hardcopy in the first place? Anyway, some of those programs do convert quantities for different servings, so here is a little program to do that. It provides input and output in fractions instead of decimals, because that is the way recipes are written.

```

100 DISPLAY AT(3,6)ERASE ALL:"RECIPE CONVERTER"
110 DISPLAY AT(6,1):"Enter fractional quantities separated by a space from whole quantities."
120 DISPLAY AT(9,1):"For instance, to enter three and one-half, type 3 1/2"
130 DISPLAY AT(12,1):"Results will be rounded to the nearest 8th."
140 DISPLAY AT(24,7):"press any key" :: DISPLAY AT(24,7):"PRESS ANY KEY" :: CALL KEY(O,K,S):: IF S=0 THEN 140
150 DISPLAY AT(12,1)ERASE ALL:"TURN PRINTER ON!"
160 OPEN #1:"PIO" :: PRINT #1:CHR$(27);"@ " :: CALL CLEAR
170 DISPLAY AT(5,1):"Name of recipe?" :: ACCEPT AT(7,1):M$ :: PRINT #1:M$;"", ""
180 DISPLAY AT(3,1)ERASE ALL:"Recipe is for how many servings?" :: ACCEPT AT(4,1)VALIDATE(DIGIT)BEEP:R
190 DISPLAY AT(6,1):"You want to cook how many servings?" :: ACCEPT AT(7,11)VALIDATE(NUMERIC):S :: X=S/R

```

```

200 DISPLAY AT(10,1):"Name of ingredient? (just enter if finished)" :: ACCEPT AT(13,1)BEEP:A$ :: IF A$="" THEN STOP
210 DISPLAY AT(15,1):"Unit of measure?" :: ACCEPT AT(17,1)BEEP:M$
220 ON ERROR 310 :: DISPLAY AT(19,1):"Quantity in recipe?" :: ACCEPT AT(21,1)BEEP:AX$ :: A=VAL(AX$)
230 Q=X*A :: J=INT(Q):: P=Q-J :: IF P=0 THEN X$=STR$(J):Y$="" :: GOTO 290
240 IF J=0 AND P<=.0625 THEN X$="" :: Y$="less than 1/16" :: GOTO 290 ELSE IF P<=.0625 THEN X$=STR$(J):Y$="" :: GOTO 290
250 IF P>.9375 THEN X$=STR$(J+1):Y$="" :: GOTO 290
260 DATA .8125,7/8,.6875,3/4,.5625,5/8,.4375,1/2,.3125,3/8,.1875,1/4,.0625,1/8
270 RESTORE 260
280 READ M,N$ :: IF P>M THEN Y$=N$ :: X$=STR$(J)ELSE 280
290 IF J<1 THEN X$=""
300 PRINT #1:A$&" "&X$&" "&Y$&" "&M$ :: GOTO 200
310 P=POS(AX$," ",1):: Q=POS(AX$,"/",1):: IF Q=0 THEN 340
320 ON ERROR 340 :: IF P=0 THEN A=0 ELSE A=VAL(SEG$(AX$,1,P-1))
330 B=VAL(SEG$(AX$,P+1,Q-1-P)):: C=VAL(SEG$(AX$,Q+1,255)):: A=A+B/C :: RETURN 230
340 DISPLAY AT(24,1):"OOPS! TRY AGAIN" :: CALL SOUND(1,10,0,-4,0):: RETURN 220

```

And here is an oldie - a utility to get the bugs out of your programs.

```

100 ! MOSQUITO #2 by Jim Peterson from a PEEK by Crag Miller
110 CALL CLEAR :: CALL SPRITE(#1,42,2,100,100)
115 DISPLAY AT(22,1):"Don't let the mosquito get":"out of the TV!":"Press any key -QUICK!"
120 RANDOMIZE :: CALL PEEK(-31808,A,B):: CALL MOTION(#1,A-128,B-128):: CALL KEY(O,K,S):: IF S=0 THEN 120
130 CALL CLEAR :: CALL COLOR(1,2,8):: CALL SCREEN(2):: CALL CHAR(32,"FF888888FF888888"):: GOTO 120

```

Long live the TI-99/4A!

Jim Peterson
The Tigercub

Clock Modifications

by Peter Mudie

Many club members, myself included, have experienced problems with the Real Time Clock kit that came out in the Club Shop in 1989. An example of the problem would be resetting on power up and down.

There is a very simple modification that can be made to prevent the loss of time. It only requires a LED and a 2.2K resistor.

Before beginning, disconnect the battery backup. The first step is to bend pin 23 of the MM58167 clock chip out. If you have not yet built the kit or you have the chip on an IC socket (it pays to use IC sockets!), it is a simple matter of carefully bending the pin out. If the chip has been soldered in then the whole chip has to be desoldered and removed from the board and pin 23 then bent out. It is not advisable to bend the pin out with the chip in place.

The next step is to get the 2.2K resistor and solder it on top of the IC from pin 12 (ground) to pin 23 which is bent out. After that the LED is soldered with the cathode (bar end) to pin 23 and the anode (positive end) to the wire feeding +5V to the board.

The final Step is to double check what you have done. If it looks right, it is time to reassemble it and test it.

Neither the author or TISHUG will not take any responsibility for any damage to any TI99/4A or clock due to the use of information from this article.

```

10 REM*****
20 REM***      ***
30 REM***      ITCHE      ***
40 REM***      ***
50 REM*****
60 GOSUB 110
70 GOSUB 470
80 GOSUB 1560
90 GOSUB 2080
100 END
110 REM
120 REM***INSTRUCTIONS***
130 REM
140 GOSUB 430
150 PRINT "THIS IS THE GAME
OF ITCHE. ITCHE IS A SPECIA
L WORM WHO LIKES TO SOLVE MA
ZES. YOU"
160 PRINT "ARE TO CREATE A M
AZE FOR THEITCHE TO SOLVE."
170 PRINT
180 PRINT "BY USING THE FOLL
OWING INSTRUCTIONS, YOU
WILL BUILDTHE MAZE WALLS. U
PON COMMAND"
190 PRINT "(G), THE ITCHE WI
LL WIND ITSWAY THROUGH THE M
AZE."
200 FOR I=1 TO 9
210 PRINT
220 NEXT I
230 GOSUB 10
240 GOSUB 430
250 PRINT "THE FOLLOWING COM
MANDS WILL AID YOU:"
260 PRINT
270 PRINT "G)O, TO HAVE THE
ITCHE WORM FIND THE EXIT TO
YOUR MAZE."
280 PRINT
290 PRINT "Q)UIT, TO END THE
GAME."
300 PRINT
310 PRINT "C)LEAR, TO START
THE MAZE OVER."
320 PRINT
330 PRINT "P)LOT, M)OVE, AND
E)RASE FOLLOWED BY DIREC
TION KEYS:"
340 PRINT
350 PRINT TAB(13);"A=UP"
360 PRINT "      K=LEFT
      L=RIGHT"
370 PRINT TAB(12);"Z=DOWN"
380 FOR I=1 TO 3
390 PRINT
400 NEXT I
410 INPUT "PRESS ENTER WHEN
READY TO CONTINUE: ":ANS$
420 RETURN
430 CALL CLEAR
440 PRINT TAB(9);"*** ITCHE
***"
450 PRINT
460 RETURN
470 REM
480 REM***SETUP***
490 CALL SCREEN(15)
500 CALL CLEAR
510 X2=1
520 Y2=1
530 CALL CHAR(120,"FFFFFFFF
FFFFFFFF")
540 CALL CHAR(128,"FFFFFFFF
FFFFFFFF")
550 CALL CHAR(136,"FFFFFFFF
FFFFFFFF")
560 CALL CHAR(144,"FFFFFFFF
FFFFFFFF")
570 CALL CHAR(152,"FFFFFFFF
FFFFFFFF")
580 CALL COLOR(12,15,15)
590 CALL COLOR(13,5,15)
600 CALL COLOR(14,13,15)
610 CALL COLOR(15,11,15)
620 CALL COLOR(16,7,15)
630 CALL HCHAR(1,3,136,29)
640 CALL HCHAR(22,3,136,29)
650 CALL VCHAR(3,3,136,19)
660 CALL VCHAR(3,31,136,19)
670 IX=2
680 IY=5
690 CALL HCHAR(IX,IY,144)
700 CLR=120
710 FOR I=1 TO 27
720 CALL HCHAR(23,I+3,ASC(SE
G$("P)LOT M)OVE E)RASE C)LR
G)O",I,1)))
730 NEXT I
740 FOR I=1 TO 25
750 CALL HCHAR(24,I+4,ASC(SE
G$("A=UP Z=DN K=LF L=RT Q)UI
T",I,1)))
760 NEXT I
770 CALL KEY(3,ANS,STAT)
780 IF STAT=0 THEN 770
790 ANS$=CHR$(ANS)
800 IF ANS$="C" THEN 490
810 IF ANS$="P" THEN 880
820 IF ANS$="M" THEN 910
830 IF ANS$="E" THEN 950
840 IF ANS$="Q" THEN 980
850 IF ANS$="G" THEN 1540
860 CALL SOUND(200,1760,5)
870 GOTO 770
880 REM***PLOT ROUTINE***
890 CLR=128
900 GOTO 1110
910 REM***MOVE ROUTINE***
920 TMP=200
930 C2=120
940 GOTO 1110
950 REM***ERASE ROUTINE***
960 CLR=120
970 GOTO 1110
980 REM***QUIT ROUTINE***
990 CALL HCHAR(23,1,32,32)
1000 CALL HCHAR(24,1,32,32)
1010 FOR I=1 TO 24
1020 CALL HCHAR(23,I+4,ASC(S
EG$("DO YOU WISH TO QUIT Y/N
:",I,1)))
1030 NEXT I
1040 CALL KEY(3,ANS,STAT)
1050 IF STAT=0 THEN 1040
1060 ANS$=CHR$(ANS)
1070 IF ANS$="Y" THEN 1540
1080 IF ANS$<>"N" THEN 1040
1090 CALL HCHAR(23,1,32,32)
1100 GOTO 710
1110 REM***DIRECTIONS***
1120 CALL KEY(3,ANS,STAT)
1130 IF STAT=0 THEN 1120
1140 X2=IX
1150 Y2=IY
1160 ANS$=CHR$(ANS)
1170 IF ANS$="A" THEN 1230
1180 IF ANS$="Z" THEN 1280
1190 IF ANS$="K" THEN 1330
1200 IF ANS$="L" THEN 1380
1210 TMP=0
1220 GOTO 800
1230 REM***UP***
1240 X2=IX-1
1250 IF X2>1 THEN 1430
1260 X2=2
1270 GOTO 1420
1280 REM***DOWN***
1290 X2=IX+1
1300 IF X2<22 THEN 1430
1310 X2=21
1320 GOTO 1420
1330 REM***LEFT***
1340 Y2=IY-1
1350 IF Y2>3 THEN 1430
1360 Y2=4
1370 GOTO 1420
1380 REM***RIGHT***
1390 Y2=IY+1
1400 IF Y2<31 THEN 1430
1410 Y2=30
1420 CALL SOUND(200,1760,5)
1430 IF TMP<>200 THEN 1450
1440 CLR=C2
1450 CALL GCHAR(IX,IY,CK)
1460 AUX=CK
1470 CALL HCHAR(IX,IY,CLR)
1480 CALL GCHAR(X2,Y2,CK)
1490 C2=CK
1500 CALL HCHAR(X2,Y2,AUX)
1510 IX=X2
1520 IY=Y2
1530 GOTO 1110
1540 REM***GO ROUTINE***
1550 RETURN
1560 REM
1570 REM***PLAY***
1580 REM
1590 IF ANS$="Y" THEN 2070
1600 CALL HCHAR(23,1,32,32)
1610 CALL HCHAR(24,1,32,32)
1620 FOR I=1 TO 19
1630 CALL HCHAR(23,I+6,ASC(S
EG$("<ITCHE NOW SOLVING>",I,
1)))
1640 NEXT I
1650 X2=2
1660 Y2=4
1670 CALL HCHAR(X2,Y2,152)
1680 IF TMP<>200 THEN 1700
1690 CLR=C2
1700 CALL HCHAR(IX,IY,CLR)
1710 X3=1
1720 Y3=0
1730 DI=2
1740 IF (X2+X3=2)*(Y2+Y3=3)T
HEN 2070
1750 IF (X2+X3=2)*(Y2+Y3=31)
THEN 2070
1760 CALL GCHAR(X2+X3,Y2+Y3,
CK1)
1770 IF (X2+X3>1)*(X2+X3<22)
*(Y2+Y3>3)*(Y2+Y3<31)*((CK1=
32)+(CK1=120))THEN 1970
1780 DI=DI-1
1790 IF DI>=1 THEN 1810
1800 DI=4
1810 IF DI<>1 THEN 1850
1820 X3=0
1830 Y3=1
1840 GOTO 1740
1850 IF DI<>2 THEN 1890
1860 X3=1
1870 Y3=0
1880 GOTO 1740
1890 IF DI<>3 THEN 1930
1900 X3=0
1910 Y3=-1
1920 GOTO 1740
1930 IF DI<>4 THEN 1970
1940 X3=-1
1950 Y3=0
1960 GOTO 1740
1970 CALL HCHAR(X2,Y2,120)
1980 X2=X2+X3
1990 Y2=Y2+Y3
2000 CALL HCHAR(X2,Y2,152)
2010 CL=CL+1
2020 DI=DI+1
2030 IF DI<=4 THEN 2050
2040 DI=1
2050 CALL SOUND(100,1760,10)
2060 GOTO 1810
2070 RETURN
2080 REM
2090 REM***END***
2100 CALL HCHAR(X2,Y2,120)
2110 X2=X2+X3
2120 Y2=Y2+Y3 continued on page 15

```

```

1 LINPUT "PRINTDEVICE?":PRIN
TDEVICE$
2 DIM LINE$(30)
100 REM*****
110 REM**TYPING WITH*****
120 REM**COLOUR-SELECTABLE*
130 REM**GRAPHICS by*****
140 REM**D.N.Harris*****
150 REM**26th February*****
160 REM**1985*1/10/1986****
170 REM**ctrl characters **
180 REM**used to clear,****
190 REM**to blank,*****
200 REM**to write*****
210 REM**to end*****
220 REM**to erase*****
230 REM**as usual ctrl z***
240 REM**stops the program*
250 REM**without beeping***
260 REM**which is a sneaky*
270 REM**trick for late ***
280 REM**night computing***
290 REM**for the forbidden**
300 REM**the program runs**
310 REM**with the capital**
320 REM**alpha lock DOWN***
325 PRINT "LIST 100 TO 325 F
OR
INSTRUCTIONS"
326 PRINT "IF YOU KNOW WHAT
YOU ARE DOING JUST TYPE C
ON"
327 BREAK
330 INPUT "WHAT COLOUR?":C
335 FOR SET=2 TO 12
340 CALL COLOR(SET,C,16):: C
ALL SCREEN(16)
345 NEXT SET :: CALL COLOR(1
,16,14)
350 REM KEYBOARD CRAYON
360 CALL CLEAR
370 PRINT "**** KEYBOARD CRA
YON ****"
380 PRINT
390 PRINT "USE YOUR KEYBOARD
"
400 PRINT "TO DRAW LINES ON
THE SCREEN."
410 PRINT
420 PRINT "PRESS ^E TO DRAW
UP"
430 PRINT "PRESS ^X TO DRAW
DOWN"
440 PRINT "1,Z,+,AND . GIVE
DIAGONALS BUT
IT WILL NOT ERASE
DIAGONALLY"
450 PRINT "PRESS ^D TO DRAW
RIGHT"
460 PRINT "PRESS ^S TO DRAW
LEFT"
470 PRINT "***NOTE WELL THE
ARROW KEYS NEED THE CONTROL
BUTTON PRESSED OR THEY JUS
T TYPE THE LETTERS!***"
475 INPUT "press enter to
continue":E$
480 PRINT "TO ERASE OR MOVE
WITHOUT DRAWING A LINE ON
THE SCREEN~~HIT ^B(BLANK MO
DE)"
490 PRINT "^^ IS SHORT FOR TH
E CTRL KEY"
500 PRINT "TO START WRITING
AGAIN,HIT ^W(WRITE MODE)"
510 PRINT "TYPING WILL BE
MADE TO START IF YOU TAP
CTRL T"
520 PRINT "TO CLEAR THE SCRE
EN AT ANY TIME~HIT ^C"
530 PRINT
540 PRINT
550 PRINT "HIT ENTER TO STAR
T"
580 INPUT ENTER$
581 INPUT "CHARACTER":KK
570 CALL CLEAR
580 W$="FFFFFFFFFFFFFFF"
590 CALL CHAR(KK,W$)
600 R=12
610 C=16
620 WRITE=1
630 CALL KEY(0,KEY,STAT)
640 IF KEY<1 THEN 630
660 IF KEY<>131 THEN 680
670 CALL CLEAR :: CALL SCREE
N(16)
680 IF KEY<>130 THEN 700
690 WRITE=0
700 IF KEY<>151 THEN 720
710 WRITE=1
720 IF WRITE=1 THEN 740
730 CALL HCHAR(R,C,32)
740 IF KEY=132 THEN 900
745 IF KEY=144 THEN 746 ELSE
747
746 GOSUB 10000
747 IF KEY=146 THEN GOSUB 30
000
748 IF KEY=140 THEN GOSUB 20
000
750 IF KEY=133 THEN 850
751 IF KEY=137 THEN GOSUB 31
000
752 IF KEY=143 THEN GOSUB 32
000
760 IF KEY=152 THEN 950
770 IF KEY=147 THEN 1000
775 IF KEY=148 THEN 1450
780 IF KEY=154 THEN 2000
785 IF KEY=138 THEN GOSUB 40
000
790 IF KEY=49 THEN 1050
795 IF KEY=135 THEN GOSUB 15
000
800 IF KEY=90 THEN 1150
810 IF KEY=61 THEN 1250
820 IF KEY=46 THEN 1350
830 CALL VCHAR(R,C,KK)
840 GOTO 630
850 REM^UP
860 R=R-1
870 IF R>1 THEN 890
880 R=1
890 GOTO 830
900 REM^RIGHT
910 C=C+1
920 IF C<32 THEN 940
930 C=32
940 GOTO 830
950 REM^DOWN
960 R=R+1
970 IF R<24 THEN 990
980 R=24
990 GOTO 830
1000 REM^LEFT
1010 C=C-1
1020 IF C>1 THEN 1040
1030 C=1
1040 GOTO 830
1050 REM UP LEFT ~~~~
1060 IF R=1 THEN 1070 ELSE 1
080
1070 R=24
1080 IF C=1 THEN 1090 ELSE 1
100
1090 C=32
1100 REM
1110 CALL HCHAR(R,C,KK)
1120 R=R-1
1130 C=C-1
1140 GOTO 830
1150 REM DOWN LEFT
1160 IF R=24 THEN 1170 ELSE
1180
1170 R=1
1180 IF C=1 THEN 1190 ELSE 1
200
1190 C=32
1200 REM
1210 CALL HCHAR(R,C,KK)
1220 R=R+1
1230 C=C-1
1240 GOTO 630
1250 REM UP RIGHT
1260 IF R=1 THEN 1270 ELSE 1
280
1270 R=24
1280 IF C=32 THEN 1290 ELSE
1300
1290 C=1
1300 REM
1310 CALL HCHAR(R,C,KK)
1320 R=R-1
1330 C=C+1
1340 GOTO 630
1350 REM DOWN RIGHT~~~
1360 IF R=24 THEN 1370 ELSE
1380
1370 R=1
1380 IF C=32 THEN 1390 ELSE
1400
1390 C=1
1400 REM
1410 CALL HCHAR(R,C,KK)
1420 R=R+1
1430 C=C+1
1440 GOTO 630
1450 J=32
1451 CALL KEY(0,J,S)
1452 IF J=151 THEN 630
1453 IF J=8 THEN C=C-.25 ::
S=0
1454 IF J=9 THEN C=C+.25 ::
S=0
1455 IF J=10 THEN R=R+.25 ::
S=0
1456 IF J=11 THEN R=R-.25 ::
S=0
1457 IF J=138 THEN GOSUB 400
0
1460 IF C<1 THEN C=1
1461 IF J>11 OR J<8 THEN 147
3
1462 IF R<1 THEN R=1
1463 IF R>24 THEN R=24
1464 CALL HCHAR(R,C,32)
1465 IF J>11 THEN S=1
1473 IF S<1 THEN 1451
1480 IF J>30 THEN CALL HCHAR
(R,C,J):: C=C+1
1486 IF C>32 THEN 1487 ELSE
1501
1487 R=R+1 :: C=1
1501 GOTO 1451
2000 END
4000 REM ***JOYSTICK MODULE*
4100 X=0 :: Y=0
4110 REM
4120 CALL JOYST(1,X,Y)
4130 CALL KEY(1,K,S)
4140 IF K=18 THEN 4320
4145 IF K=11 THEN 1450 :: IF
K<>5 THEN 4150
4146 CALL HCHAR(R,C,32)
4150 C=C+X/4
4160 REM COLUMNS 1 TO 32
4170 R=R-Y/4
4180 REM ROW 1 TO 24
4190 REM **COLUMNS**
4200 IF C>32 THEN 4240 ELSE
4210
4210 IF C<1 THEN 4220 ELSE 4
260
4220 C=1
4230 GOTO 4260
4240 C=32
4250 REM **ROWS**
4260 IF R>24 THEN 4290 ELSE
4270
4270 IF R<1 THEN 4280 ELSE 4
300
4280 R=1
4285 GOTO 4300
4290 R=24
4300 CALL HCHAR(R,C,KK)

```

Extended BASIC Problem

by Bob Relyea

I put an article in the TND last month containing my Grade Standardiser program, as well as a description on 'how I did it' with certain aspects of the programming. I have also included another tutorial in another part of this issue explaining another aspect of the programming.

The part that I wanted to dwell on in this issue is an unusual problem that I ran across while making up the program which may be of use to users who have an occasion to write a long Extended Basic program. That program, incidentally, was the longest one that I have ever tried so I never ran across this particular problem before.

Everything was going along pretty well (with only a minor hiccup) until one day I noticed that a couple of subroutines would not work. Upon investigation I discovered that certain lines on the subroutines could not be called up. Whenever I typed the line number and then used the up or down arrow key in an attempt to get the line to appear, a beautiful mosaic of colours would appear on the screen along with an impromptu symphony - certainly nothing that I had remembered programming! I re-typed the lines and I made every other attempt that I could think of to get several 'disappearing lines' to stay in the program. Finally, in desperation, I called Ross (poor 'ol Ross will one day tire of hearing my voice on the other end of the receiver!) and after having talked about the problem for a while with little success, Ross then thought of something that he had run across several times himself in the past. He asked me how large the program had gotten to. My response was "forty-five sectors, I think". He said, "that is what the program is!" You may have noticed while cataloguing a disk that for a small Extended Basic program the listing is program, but for a large one it is Int/Var 254. So there is some kind of a change that takes place at around the 46 sector length. There appears to be some kind of defect in the system at this point and Ross said that although he had discovered the problem he has never had time to explore it any further.

He proceeded to give me several suggestions to overcome the problem including a use of the Merge option but none of them really worked. The best way to avoid the problem is (when you see the 46 sector mark approaching) to get the program over the 46 sector mark as quickly as possible by typing in 'rubbish lines' and then continuing with your programming. It was too late for me, however, so I had to adopt another routine.

What I ended up doing was converting the program to a D/V 80 file by using the LIST command and then I called it up with the editor, typed in the 'disappearing lines' and then added several rubbish lines to ensure that the program would be well-and-truly over the 46 sector mark. The rubbish lines would be made by using the REM statement and holding down one of the keys until the line was full. I repeated this for about six REM statements. Then I used a program that was in a recent issue of the TND to convert the D/V 80 file back into a running program and from then on I was 'home and hosed'. Later on when the program got even larger I was able to take out the rubbish lines so you would not have noticed them in the program that was published.

Ross said that he was 'caught' more than once by this defect in the system but nobody that I have spoken to can remember seeing it documented anywhere. So, if any of you can recall having a similar problem then what has been written above is possibly the reason. I trust that this will be of some use to some user. ○

continued from page 3

So that is what I have been doing with my computer. Putting it to use in my spare time, and enjoying the challenge of producing something that others have to spend a lot of money to equal.

To the committee, keep up the great work for us. It is only through your efforts that the likes of me and the great majority of the members have had the ability to remain TI99/4A users.

Michael Ball
Nowra NSW.
26 AUG 1991

○

continued from page 14

```
4310 GOTO 4120
4320 RETURN
10000 OPEN #1:PRINTDEVICE$
10010 FOR Z1=1 TO 24
10020 Z1$=""
10030 FOR Z2=1 TO 32
10040 CALL GCHAR(Z1,Z2,Z3)
10050 Z1$=Z1$&CHR$(Z3)
10060 NEXT Z2
10070 PRINT #1:TAB(5);CHR$(1
4)&Z1$
10071 LINE$(Z1)=Z1$
10080 NEXT Z1
10090 CLOSE #1
10095 KEY=32
10100 RETURN
15000 REM GRAPHIC OUTLINE AR
OUND CURSOR
15010 RAD=11
15020 IF C=RAD<1 OR C+RAD>32
THEN RAD=RAD-1 :: GOTO 1502
0
15030 IF R=RAD<1 OR R+RAD>24
THEN RAD=RAD-1 :: GOTO 1503
0
15040 FOR I=1 TO 360 STEP 30
15050 X=(RAD*COS(PI*I/180))
15080 Y=(RAD*SIN(PI*I/180))
15070 CALL HCHAR(R+X,C+Y,KK)
15080 NEXT I
15090 RETURN
20000 REM LOAD
20010 OPEN #3:"CS1",INPUT ,F
IXED,INTERNAL
20020 FOR LOAD=1 TO 24
```

```
20030 INPUT #3:LINE$(LOAD)
20040 NEXT LOAD
20050 CLOSE #3
20054 GOSUB 20055
20055 REM SCREEN PRINT
20060 FOR LINE=1 TO 24
20065 FOR LETTER=1 TO 32
20070 CH$=SEG$(LINE$(LINE),L
ETTER,1)
20075 CALL HCHAR(LINE,LETTER
,ASC(CH$)): NEXT LETTER
20080 NEXT LINE
20090 KEY=32
20100 RETURN
30000 REM RECORD
30010 OPEN #2:"CS1",OUTPUT,I
NTERNAL,FIXED
30020 FOR RECORD=1 TO 24
30030 PRINT #2:LINE$(RECORD)
30040 NEXT RECORD
30050 CLOSE #2
30060 KEY=32
30070 RETURN
31000 REM MODEM INPUT
31010 OPEN #9:"RS232/2",INPU
T ,DISPLAY ,VARIABLE 80
31020 FOR IN=1 TO 24
31030 INPUT #9:LINE$(IN)
31040 NEXT IN
31050 CLOSE #9
31060 GOSUB 20055
31070 REM
31080 REM
31090 KEY=32
31100 RETURN
```

```
32000 REM MODEM OUTPUT
32010 OPEN #10:"RS232/2",OUT
PUT,DISPLAY ,VARIABLE 80
32020 FOR OUT=1 TO 24
32030 PRINT #10:LINE$(OUT)
32040 NEXT OUT
32050 CLOSE #10
32060 KEY=32
32070 RETURN
```

continued from page 13

```
2130 FOR I=1 TO 10
2140 CALL HCHAR(X2,Y2,152)
2150 CALL HCHAR(X2,Y2,120)
2160 CALL SOUND(100,1760,10)
2170 NEXT I
2180 IF ANS$="Y" THEN 2190 E
LSE 2230
2190 CALL HCHAR(23,1,32,32)
2200 CALL HCHAR(24,1,32,32)
2210 PRINT "ITCHE SAYS YOU S
POILED HIS FUN"
2220 GOTO 2290
2230 IF Y2<>31 THEN 2270
2240 PRINT "ITCHE HAS SOLVED
THE MAZE"
2250 PRINT "HE DID IT IN ";C
L;" CLICKS"
2260 GOTO 2290
2270 PRINT "ITCHE CANNOT SOL
VE YOUR MAZE"
2280 PRINT "HE IS STUCK AT T
HE BEGINNING"
2290 RETURN
```

○

TI99/4A World News

by Jim Peterson, Tigercub Software, USA

According to the "Confidential File" written by Christopher Pratt, Tony McGovern is delaying his move to the Amiga in order to make further improvements to Funnelweb. He plans a complete rewrite of the Editor, and also wants to add hard drive support. Tony has given up hope on the Myarc HFDC and has turned to the new ESD H/F Controller card. Shane Truffer, the CEO/President of Electronics Systems Development Corporation, has promised to provide Tony with the card and a hard disk and any technical information he may need.

According to the same source, there is now a Digitizer for the TI, which ties together your VCR, TI and printer and produces a nice picture. The equipment is reported to be available from RFW Enterprises, 111 Oakridge St., Chicobee MA 01020, phone (413) 593-3274, and the "Imagewise Portrait Print" software can be obtained from Circuit Cellar Inc., 4 Park St. Suite 12, Vernon CT 06066, phone (203) 875-2751.

The West Penn 99'ers newsletter has reprinted a message from the Unix TI conference, by Thomas Opheys of the University of Passau in West Germany, regarding his project of writing a TI-99/4A emulator for the IBM PC! He states that the entire range of hardware will be supported - keyboard, VDP in all modes, sound processor (in only one channel because the PC has only one tone generator, but SoundBlaster owners will get all four), 32k card, MiniMemory, GROM/ROM modules like Extended Basic, Editor/Assembler, and peripheral cards such as the P-code card. The RS232 and Disk Controller cannot be emulated correctly on the hardware level, so the DSR ROMs will be patched to handle calls to those cards by DOS. Therefore, programs which access those cards directly, such as Telco, Turbo-Copy, TapidCopy, etc., will not work. Emulation of the Speech Synthesizer is not considered possible. Mr. Opheys states that the emulator is not yet finished.

Another BBS message reprinted in the same newsletter reports a flyer from McCann Software concerning a new Geneve program called HQ-Stacks, said to be similar to Hypercard and the like.

MS Express Software (P.O. Box 498, Richmond OH 43944) has released Adventure Hints - Series II by Lynn Gardner. The price is \$9.95 plus \$1 SH and sales tax for Ohio purchasers.

Mike Maksimik and Chris Bobbitt of Asgard Software have had a complete falling out. According to messages posted by Mike on the Chicago BBS, his Midi Master will not be sold by Asgard Software. It has been renamed MIDI Master 99 and will be sold by Crystal Software, 635 Mackinaw Ave., Calumet City IL 60409-4014. The current version 2.2, which includes interface, software, two connecting cables and documentation, is available now for \$45 (make checks payable to Michael Maksimik). This version allows loading of disk-based music files, and compiling of symbolic music files, for playing on the MIDI interface. It will NOT record keyboard-based music; that part is under development and will be released, hopefully toward the end of May, as version 3.0 (the Cakewalk compatible version). Mike states that those who purchase version 2.2 will be entitled to version 3.0 at no charge if they send in the registration card.

A message on Delphi's TI Forum reported that a "Trojan Horse" virus had been discovered in Telco, which causes the disk drive to wipe files off the drive while downloading.

CaDD Electronics (81 Prescott Road, Raymond NH 03077, (603)895-0119) has announced the availability of RICH GKXB by Richard Lynn Gilbertson, for \$24.95 plus \$2 PPM. This enhanced version of Extended Basic requires the use of a gram emulating device which supports the GK

file header, such as the GRAM Kracker, Gramulator, and Geneve 9640. Users of the GRAM Karte may convert the GK file type to the GRAM Karte files with the "Universal GRAM File Converter" also available from CaDD.

Among other features, more than 40 new commands have been added to Extended Basic, as well as more than 150 combinations of commands executed with a single CALL for greater speed in execution, and even conditional combined commands in which the second part will only execute if the first part is true.

Rodger Merritt has developed a method of converting TIPS graphics into a form which can be printed from the TI-Writer or Funnelweb Editor (but not the Formatter), and a program called PIXEASE which makes it quite easy to combine these graphics with text in any combination, and even in colour. He offers a flipped disk of 140 graphics for \$5 (plus PPM, I presume) and has 30 such disks, available in several combinations and with a discount for the entire collection archived. The address is Comproline, 1949 Evergreen Ave., Fullerton CA 92635.

Mike McCann's HQ_STACKS for the Geneve is such an advanced program that I have no clear idea what it is all about, but it is said to offer a graphical user interface screen, browser-stackware editing environment, MDOS command line interface utility, built-in F7 pop-up windows help system, and HQ_Stacks artwork resource program. Demo disks are available.

Mike Maksimik is now shipping his MIDI Master 99 Version 2.2. This has been publicized as having been developed using a CASIO keyboard costing \$80, and is stated to be compatible with any keyboard having a MIDI interface. Unfortunately, one of the first buyers has reported great difficulty finding an inexpensive keyboard which has such an interface. It seems that they are usually only found on the expensive professional models.

I sent a letter to JP Software, addressed to 2390 El Camino Real #107, Palo Alto CA 94306. The letter was postmarked at Columbus OH on 28 September 1990. On 15 July 1991 I received it back, stamped "Addressee no longer at this address/ return to sender" and "Returned to sender/ addressee unknown". Two users have informed me that they sent an order to JP Software and their check was cashed but merchandise was never received.

In June, Shane Truffer of ESD posted a BBS message stating that the ESD Hard Drive Controller Card would support IDE drives instead of MFM, and that he had sent refund checks to those who had previously ordered in advance. He stated that ESD would ship a 40MB IDE drive with the controller card for a price, to be announced soon, in the \$225-\$250 range; that the IDE drives have faster access times than all the MFM type drives; and that the controller card is directly on the hard drive, so that all that is needed is a simple interface to the TI. This is said to insure compatibility and to eliminate the problem that ESD has had with availability of WD MFM controller chips.

According to an unsigned article in the Kawartha Kronicle, Gary Bowser of O.P.A. has acknowledged that his RAMBO is not compatible with Super Extended Basic nor with the Speech Synthesizer, and the writer found that it was also incompatible with TI Extended Basic. He also found serious shortcomings in the RECALLIT +2 program - the two-column printout is too wide for a standard printer, there is no way to input a country name and no line to print it on a label, and the "remarks" field cannot be printed in the label format.

An article in the OSHAWA TI UG newsletter reports that O.P.A.'s TI Image Maker (TIM) in its present version is incompatible with many monitors. It requires an RGB analog monitor - not RGB digital (TTL). It works well on a 15.75 kilohertz scan, but most inexpensive RGB monitors are 31.5 kilohertz. They also use separate horizontal and vertical scan lines, but TIM is set up to

continued on page 10

Look at GPLLNK

by R.A. Green, Ottawa, Canada

The Operating System of the TI99/4A consists of code in ROM and in GROM. The code in ROM is assembler language. The code in GROM is TI's proprietary Graphics Programming Language (GPL). The ROM code has three main functions: interrupt processing, floating point arithmetic and GPL code interpretation. The GROM code has everything else!

There are, in all this GPL code, several very useful routines that can be used by Assembler language programs. The Editor Assembler and the Mini Memory modules provide a means, called GPLLNK, to access these routines in GROM. The Extended BASIC and TI-Writer modules do not provide a link to GPL.

I have developed a GPL link routine that will work for all modules. The Assembler source listing is shown below. The code for this routine is a bit tricky, so a few notes for those who want to understand the code may be in order.

1. The workspace registers are already loaded with some necessary values when RAGLNK is called.
2. The first, and only the first, time RAGLNK is called, it searches all GROMS until the hexadecimal value 0FFF is found.
3. The GPL operation code >0F is a call to an assembler language routine. The byte following the >0F, in our case, >FF gives the table number and entry number in that table. Table number 15 (>0F) begins at >8300 in the console CPU RAM, and entry 15 in this table is at address >831E.
4. A GPL CALL stacks the current GROM address then branches to the routine to be called. A GPL RETURN unstacks a GROM address then resumes execution at that address. RAGLNK stacks the GROM address of the >0FFF instruction, then goes to the GPL interpreter to begin execution of the GROM subroutine. When the GPL subroutine does a RETURN, the >0FFF instruction is executed, causing GPL to exit to the assembler language routine whose address is at >831E. This brings it back to RAGLNK who returns to his caller.

Assembler Source

```
*TITLE:      GPLLNK Subroutine
*AUTHOR:     R. A. Green
*FUNCTION:    Provides access to the GPL routines, no
*             matter which cartridge you are using.
*LINKAGE:    Same as described for GPLLNK in E/A or MM
*             manuals except that the GPL STATUS byte
*             need not be reset before calling:
*             BLWP @RAGLNK
*             DATA GPL-routine-address
*NOTES:       This routine depends upon finding the value
*             >0FFF somewhere in GROM. This value occurs
*             at least 3 times in the console GROMS in my
*             machines. >0F is the GPL opcode to call an
*             assembler routine.
*
DEF RAGLNK
RAGLNK DATA WSP,$+2      Linkage/Transfer Vector
MOV R0,R0                 Do we have an address of >0FFF?
JNE STACK                Jump yes
* Find an occurrence of >0FFF somewhere in GROM
MOV B R0,*R3              Set the GROM address to zero
MOV B R0,*R3
JMP $+4
SRCH1 INC R0              Increment our GROM address
MOV B *R4,R1             Get next GROM byte
SRCH2 CI R1,>0F00         Is it the start of our value?
JNE SRCH1                Jump no, keep looking
MOV B *R4,R1             Get the byte after >0F
CI R1,>FF00               Do we have >0FFF?
JEQ STACK                Jump yes, EUREKA!
INC R0                   Bump our GROM address past >0F
JMP SRCH2                And keep looking.
* Notice that the above loop will not end if we
* do not find an occurrence of >0FFF
```

```
* Put our GROM address on the GPL subroutine stack.
STACK INCT *R7            Bump GPL stack ptr at >8373
MOV B *R7,@REG2+1       Get stack ptr into >83xx
MOV R0,*R2              Our GROM address to the stack
MOV *R6,R9              Save contents of >831E
MOV R5,*R6              Put address of BACK into entry
* >F of table >F.
* Get GPL routine address from CALLER
MOV B R10,@>837C        Reset GPL STATUS byte
MOV *R14+,R8            Fetch the GROM address
LWPI >83E0              Switch to the GPL workspace
MOV @REG8,R6            R6=next GROM address to interpret
B @>0060                Go to GPL interpreter
* Hopefully GPL will come back here
BACK LWPI WSP           Switch back to our workspace
MOV R9,*R6              Restore value in >831E
RTWP                    Return to calling program
* Our workspace registers loaded with interesting stuff
WSP DATA 0             R0=our special GROM address
DATA 0                  R1 LSB is zero
REG2 DATA >8300        R2=GPL subroutine stack address
DATA >9C02              R3=GROM write address address
DATA >9800              R4=GROM read data address
DATA BACK              R5=address for GPL to come back to
DATA >831E              R6=address of entry >F of table >F
DATA >8373              R7=pointer to GPL subroutine stack
REG8 BSS 2              R8=GROM addr of GPL routine
BSS 2                   R9=saved contents of >831E
DATA 0                  R10=zero
BSS 10                  R11-R15
END
```

Look at CS1

by R.A. Green, Ottawa, Canada

Have you ever noticed how good it feels when you stop beating your head against a brick wall? Or how obvious a solution to a problem is once you have found it?

There has been quite a bit of interest lately on bulletin boards and in Computer Shopper about loading assembler language programs from cassette. Loading assembler language programs from cassette would be very useful for those that have 32K built into the console of an otherwise unexpanded system. The problem is that the convention for loading multi-segment programs requires that the last character of the name be incremented each time a segment is loaded. That is, a three segment program, TOMB, loaded from disk causes loading of three files: "DSK1.TOMB", "DSK1.TOMC" and "DSK1.TOMD". Editor Assembler Option 5 and TI-Writer Option 3 both know how to do this. What happens when the program is on CS1? Both Editor Assembler and TI-Writer proceed as usual and try to load three files: "CS1", "CS2" and "CS3".

There is the problem! Too bad. The solution is a special loader. Right? Well I thought so, and wrote one, but it is the wrong answer.

Let us back up a minute. As we all know, and are willing to tell anyone, TI did a good job designing the TI99/4A. One of the things they did really well is the support of I/O devices so that new ones could be added easily. One of the things that allows this is the way "device/file" names are constructed. A name is in two parts. The first part is the device name proper: "DSK1", "DSK2", "RS232/1", "RS232/2", "CS1", "CS2", and so on. The second part is a series of options separated from the device name and from each other by periods. In the name, "DSK1.TOMB", "DSK1" is the device name and "TOMB" is the option. It is only the disk ROM that considers this option to be a filename in the disk directory.

The standard way to do an I/O operation is to scan up to the first period to isolate the device name; find that device on some ROM/GROM; then call the ROM/GROM routine which will process the options and do the I/O.

What do you suppose would happen if you specified an option to a device that did not expect options?

continued on page 17

Beginning Forth - part 10

by Earl Raguse, UGOC, CA USA

STILL MORE GRAPHICS

This lesson includes example programs for a few more of the graphics words available in Forth. I hope you are studiously reading Chapter 6 of the TIFM. I find that this is one of the best written, in spite of my complaint about DCOLOR. After this lesson, we will have exercised a lot of Forth graphics words, but there are still Forth graphics words we will not get to; ie Sprite words SPRGET, SPRPUT, SPTDIST SPTDISTXY, SPRPAT, SPRCOL, SPCHAR, COINC, COINCALL, JOYST, multicolour words MULTI, MCHAR, MINIT, and bit map words DRAW, UNDRAW, and DTOG, and also BEEP, HONK, GCHAR. I think that the TIFM is pretty clear on how they work, and you have had a pretty good grounding in graphics, so go to it. Screen #63 is based on a graphics program by Ed York of the CIN/DAY UG which was printed in the Smart Programmer, this is what got me started in Forth graphics. I copied it and it worked! I immediately began to analyze what Ed had done to get that rather startling effect from just straight lines. As soon as I understood what the program was doing, I started trying to enhance it.

To envision what is going on here, imagine a yard stick on the floor, perpendicular to and touching a wall. Now take the wall end and slide it up the wall until it is vertical. Very simple, now do mirror images of this on the other walls and on the ceiling and also in the adjacent room, and you have it. I do not remember whether the original did the ceiling or not, but I know I added the colour to it. I can remember how amazed I was, the first time I saw this effect in a Forth demo by Bruce Carson. I was truly impressed. I did not know at the time that he had gotten it from the Smart Programmer and that I would soon be doing it also.

The display comprises eight stand alone segments. The segments are drawn by loops labeled A1 through A8. A1, for example, draws a line from pixel coordinates X=0, Y=191 to X=128, Y=191; This repeats 16 times, ending at X=128 (0 I 8 * +), Y=191 to X=128, Y=63 (191 I 8 * -). The others follow in the same pattern. A0 sets up GRAPHICS2 mode, sets DCOLOR (via D) to white, and the SCREEN to black, all in HEX. GO puts it all together in the white on black version, GC does it for the colour version. The rest of the screen is words in the immediate mode so that when loaded, the program automatically executes. Until you are sure of what is going on, I think you should not type in the last six words, so that they do not automatically execute, unless you are of the "go for broke" type.

You must have my UFW's from lesson #4 loaded, because these programs use FG, AT, MS, WAIT, DEC etc. You must also have defined IT on Screen #3. I will not keep repeating this. So if Forth says WAIT ? or something similar, its just saying that it has not been informed of these very useful words. Do it NOW. If you do not have DIR you can just leave it out, see the comments on that below.

The best way to test to see if you have any typing errors is to execute AO A1 TEXT, if that works try AO A2 TEXT and so on. A0 must be included each time, or leave off TEXT and type blindly.

I think this is a pretty powerful display for one Forth screen. Admittedly, its rather crowded, but quite readable except perhaps for all the HEX's and DEC's.

An example which illustrates several new graphics words while having a little fun, is WORMY STUFF on Screens #64 and #65. Note that these are all in HEX. If you have any trouble and Forth breaks with an error message, you will still be in Hex and Forth will not understand your decimal instructions until you enter DECIMAL.

What happens here, is that the CODEIT loop redefines 12 of the non-printing characters 17 through

28 (HEX 11 to 1C) using the word CHAR and the data loaded on the stack from Screen #65 (HEX 41), see WORM1. Then the SHIFT loop reads the patterns at 11-1C with I CHARPAT and copies them into the 12 characters located at 61-6C, with I 50 + CHAR. EMITTT prints a set of these characters to the CRT and PAINT uses EMITTT to paint the whole CRT screen with them.

So far so good, we are now ready to go. WORM calls RAND to get a random number from 1-12 which is added to 60 to get the address of one the modified characters at 61-6C, reads its pattern to the stack with CHARPAT and CHARs it into a random character position in the originally modified set of 12 characters at 11-1C, thus modifying the screen pattern everywhere that character is located.

Notice that WORM ends with the word COLR which, in turn, uses COLR1 and COLR2 to randomly change the foreground colour of character sets 2 and 3, which, by more than mere coincidence, are the 12 modified characters at 11 through 1C. Because the colours for COLR1 and COLR2 are randomly selected, the character sets may have the same colour. Note the delay of 100 MS in COLR; Forth is so fast that we need a delay, or there will be only a blur.

K1 and K2 are just messages. WORM1 and WORM2 sort of orchestrate the whole thing. If you have not yet installed (and why not?) a disk DIRectory, leave out the word DIR in WORM2. It also may be a good idea to put ABORT in this place until you are certain that everything works, or you will do a lot of chasing back and forth.

The word STOP? uses the Forth word ?TERMINAL to check if you are pressing FNCT 4 and if so will ABORT after restoring the TEXT mode. You may wish to change ABORT to DIR, after you are

```
SCR #63
0 ( GRAPHICS FROM ED YORK CIN DAY UG ) FG IT : IT ;
1 ( MODIFIED BY E RAGUSE 12 24 84 ) HEX : D DCOLOR ! ;
2 : AO GRAPHICS2 FO D 0 DMODE ! 1 SCREEN ; DEC
3 : A1 17 0 DO 0 I 8 * + 191 128 191 I 8 * - LINE LOOP ;
4 : A2 16 0 DO 255 I 8 * - 191 128 191 I 8 * - LINE LOOP ;
5 : A3 17 0 DO 128 I 8 * + 191 255 191 I 8 * - LINE LOOP ;
6 : A4 17 0 DO 128 I 8 * - 191 0 191 I 8 * - LINE LOOP ;
7 : A5 17 0 DO 0 I 8 * + 0 128 0 I 8 * + LINE LOOP ;
8 : A6 16 0 DO 255 I 8 * - 0 128 0 I 8 * + LINE LOOP ;
9 : A7 17 0 DO 128 I 8 * + 0 255 0 I 8 * + LINE LOOP ;
10 : A8 17 0 DO 128 I 8 * - 0 0 0 I 8 * + LINE LOOP ;
11 : A9 20 0 DO 254 190 I 10 * - 0 0 I 10 * + LINE LOOP ;
12 : GO AO A1 A2 A3 A4 A5 A6 A7 A8 1 WAIT TEXT ; HEX
13 : GC AO 60 D A1 30 D A2 50 D A3 70 D A4 90 D A5
14 : BO D A6 DO D A7 20 D A8 1 WAIT TEXT ; TEXT CLS
15 : C B AT ." GRAPHICS DEMO" 600 MS GO GC FO D DEC DIR
```

```
SCR #64
0 ( WORMY STUFF EGR 1/3/85 ) FORGET IT : IT ; HEX
1 : GR GRAPHICS ; RANDOMIZE : RAND OC RND 1+ ;
2 : STOP? ?TERMINAL IF TEXT ABORT THEN ;
3 : COLR1 OD RND 2 + 1 2 COLOR ;
4 : COLR2 OD RND 2 + 1 3 COLOR ;
5 : CODEIT 1D 11 DO I CHAR LOOP ;
6 : EMITTT 1D 11 DO I EMIT LOOP ;
7 : SHIFT 1D 11 DO I CHARPAT I 50 + CHAR LOOP ;
8 : PAINT 40 0 DO EMITTT LOOP ;
9 : COLR 100 MS COLR1 COLR2 ;
10 : WORM RAND 60 + CHARPAT RAND 10 + CHAR COLR ;
11 : K1 CLS OA OC AT ." THIS IS WORMY STUFF " ;
12 : K2 O6 OE AT ." HOLD <FCIN 4> TO QUIT EARLY " CR CR ;
13 : WORM1 K1 K2 41 LOAD GR CODEIT SHIFT 1 SCREEN PAINT ;
14 : WORM2 WORM1 40 0 DO WORM STOP? LOOP TEXT DIR ;
15 : DEC TEXT WORM2
```

through testing this screen.

I had intended to include a graphics screen that was based on things like square root, sines and cosines, but on second thought, that would take a lot of explaining of Floating Point Arithmetic, and that would certainly make this article much too long, thus it will have to wait. Until then,

continued on page 23

Multiplan Exercises

by Herbert Schlesinger, USA

PRINTING:

It is a good idea to save copies of your worksheet on paper, especially those printed out with the formulas. In case of the loss of the disk you would have a record to reconstruct your work. Multiplan allows you to place a copy of the worksheet into a wordprocessing environment for editing or for inclusion in another document.

1. Bring up the sheet to be printed on the screen. (See page 5).

2. Align the paper so a perforation is just above the print head and TURN ON the printer.

3. Select the Print command (press "P" or space over and press <ENTER>). We now have:

PRINT: Printer File Margins Options

Select Printer. If the sheet is too wide for the carriage of the printer, the work will be printed in sections for later assembly.

If what is printed out is not to your liking (apparent double spacing or some other flaw) you will have to change some of the defaults in this section of Multiplan. After making such changes, if the sheet is Transfer Saved, these new designations will be the new defaults for that file.

Select Print, then Margin to get:

PRINT MARGINS: left: 5 top:6 print width:70
print length:54 page length:66

If you are printing on wider paper, or using Condensed type, change the "print width" accordingly (Condensed used 132 for the 10X printer). Press <ENTER> when you have finished your changes and then the Printer command as above.

The "Left" option lets you set a wider left margin if desired. "Print width" is the maximum characters on a single line. The "print length" is the maximum number of lines on a single page and the "page length" determines when the program moves to a new page.

Print Options allow modification of other characteristics of the printed worksheet. When you select Print Options, this:

PRINT OPTIONS: area: setup:
formulas: Yes(No) row-col numbers: Yes(No)

"Area" option allows you to print a portion of the work sheet. Just show a range in the space provided; if more room is needed, Multiplan automatically provides the extra space. The "formulas" option, if YES, prints out the formulas for any cell rather than the result of the calculation. The columns will be widened out to 20 spaces and all text appears in quotation marks. "Row-col numbers" is just that. If Yes is selected the row numbers and the column numbers are printed out with your worksheet. Notice that both of these last two are defaulted to NO.

"Setup" allows special codes to be inserted so that signals to the printer may be placed here. The printer manual will provide the code for, say, condensed print (for the 10X it is "Esc B 3" or CHR\$(15)). Entered here, along with PIO if you have a parallel printer will do the job.

To use the PRINT on file command, merely enter a drive and then the filename you wish use to preserve the sheet. This saves a D/V 80 file, allows editing and also permits the file to be incorporated into a word processor document.

Bring " MEN'S " to the screen, and when you have dealt with that, then " WOMEN'S " and after that, " CHILDREN'S "

Spreadsheets can be of two classes: "supporting spreadsheets" and "dependant spreadsheets". Supporting sheets are independent spreadsheets which are complete in themselves for the information they contain, such as a department, accounts payable, accounts receivable, or whatever. But these sheets can be linked to a dependant sheet which takes figures or information from the supporting sheet(s) and co-relates the information. As an example we will set up four sheets linked together: " HERB'S STORE " for Herb's Store; " MENS "; " WOMEN'S "; and " CHILDRENS " as departments of the store. These departmental sheets will be the supporting sheets and HERB'S STORE is the dependant sheet which brings data from each department together for analysis.

These sheets have been set up under those names on the disk and in this case they should be printed out using the Printer options mentioned previously so that they are at hand for reference; since only one of them can be on the screen at a time, the others in hard copy will be handy.

Any data which will be transferred to the dependant sheet must be named (use the Name option from the main menu) so that the dependant sheet can easily call for that information. Understand that when the dependant sheet is "loaded" the supporting sheets are also checked to be available for the dependant sheet even though you can not reference them in that way. In this case the items which must be named are " Gross Profit on Sales " and " Total Expense ". To do this place the pointer in cell R7C2 (on MEN'S dept sheet) and, using the Name command, type in Sales as the name for this cell. Move the pointer to R15C2 and name that cell Expenses. Do the same for the other two department sheets.

Now bring up the dependant sheet HERB'S STORE and this is what you have:

	1	2	3	4	5	6
1			HERB'S STORE			
2			Department Summary			
3						
4		Mens	Womens	Children	Combined	
5						
6	Sales					
7	Expenses					
8	-----					
9	Profit					
10						
11	Percent					

Format the sheet so that the numbers: R6C2:R9C5 are set for currency (\$), and R11C2:R11C4 are formatted for percent (%) display. Labels in R4 should be centered, and R1C3:R2C4 are formatted as continuous (Cont).

Row 9 will have the formula R[-3]C-R[-2]C copied into columns 2,3,4 and 5. Row 11 will calculate the percentage of the entire profit by dividing the departments profit by the combined profit in R9C5. Use R[-2]C/R9C5. The "combined column" uses the formula SUM(RC[-3]:RC[-1]) for each of the two cells involved. The profit cell in C5 was figured as the others. When all of this is done, the totals show \$0.00 and the percent shows an error (#DIV/0!). This is because we have divided by zero (R9C5), BUT as soon as some figures are entered in the cells, this will correct itself.

That should get anyone started with MULTIPLAN. There are so many uses for a spreadsheet: Mailing lists; Checking accounts; Tax forms; Investment records - the list is endless. If you have questions you could send me a letter, but your best bet is to ask someone in your Group. Usually there is at least one person who will know. As a last resort, READ THE MANUAL. It is a lot to plough through, but most of the answers are there. 0

TI-Base Tutorial #14

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 its free.

Welcome to a session of total insanity with Martin S. I discovered a bunch of things lately that changed my thinking about all the tutorials I previously produced. I went several months without using TI-Base. A few days ago I thought I would rev up my system and jump back into writing the Tutorials. What a surprise! The system decided it did not like me. I gave it all kinds of different commands, but it would not do what I wanted it to do. At the same time I was having all these problems I was leafing through my old notes and letters and I found that many other people had the same problems at one time or another. To cut this story short, I came to the conclusion that many of my explanations of TI-Base functions were inadequate. They were only useful if you already knew what you were doing. While my memory was giving me problems, due to a lack of use, I had great difficulty using my own Tutorials to re-educate myself on the intricate functions of this language. If I have this much trouble interpreting my own work, I can imagine that there are a lot of people out there saying to themselves, "This guy is crazy. I do not understand a word of this junk". So I am going to try to regroup my thoughts and try again. It may not be any better than before, but this time I will try to include more of the antics I go through to accomplish a task in hopes that you will pick up some ideas along the way.

For this months article I decided that I wanted to start working with more numbers, and possibly, with the use of numbers from more than one Db at a time. I decided to work with utility bills for 1989 and a chequebook- type Db. I do not keep track of my bills or my chequebook on my computer, so I thought I could do this project as a beginner. I expect to make a lot of mistakes in planning and will probably change the Dbs several times in the next couple months. I started with a pencil and paper, as usual, and listed some utilities with dates and chequenumbers. I came up with the BILLS Db you see at the bottom of this page. I am not happy with it, but it is a start and it contains dates, payments and chequenumbers. This will give us something to play with. I created this Db to the specifications you see in the bottom right corner of this page. When TIB asked if I wanted to enter data, I answered yes, and just held the ENTER key until I had twelve blank records. Then I pressed FCTN 9 to get out of APPEND and typed EDIT <E>. Using EDIT to enter your data into blank fields is helpful, because you can page up to see numbers in the previous record. If you use this idea, you must remember to use FCTN 8 to record the data as you enter it. Unfortunately there are some things you will learn the hard way. I filled in a bunch of dates, payments and chequenumbers in what I hope is a logical manner for now.

```
* NUMBER CHKBK Db   CHKNUM/C
LOCAL NUMBER N 4
REPLACE NUMBER WITH 245
USE CHKBK
WHILE (NUMBER<330)
  APPEND BLANK
  REPLACE CHKNO WITH NUMBER
  REPLACE NUMBER WITH NUMBER + 1
ENDWHILE
CLOSE ALL
RETURN Copyright Martin A. Smoley 1989
```

```
E MOVE NUMBERS TO CHKBK FLCKBKP/C
SELECT 2
USE CHKBK
SELECT 1
USE BILLS
WHILE .NOT. (EOF)
  SELECT 2
  FIND 1.PCHK
  IF .NOT. (EOF)
    REPLACE 2.DATE WITH 1.MODATE
    REPLACE 2.PAY'DEBIT WITH 1.PHONE
    REPLACE REMARKS WITH "Phone ,"
  ENDIF
  SELECT 1
  MOVE
ENDWHILE
CLOSE ALL
RETURN Copyright Martin A. Smoley 1989
```

Listing of BILLS database	REC	MODATE	PHONE	GAS	ELECTRIC	SEWER	WATER	TAX	PCHK	GCHK	ECHK
SCHK WCHK TCHK	0000	01/15/89	56.16	110.41	68.88	.00	43.14	627.48	248	249	250
		252	253								
	0001	02/15/89	111.32	97.71	58.24	46.20	.00	.00	255	256	260
		261									
	0002	03/15/89	86.02	99.64	53.42	.00	.00	.00	265	266	267
	0003	04/15/89	44.81	69.83	54.39	.00	.00	.00	270	271	272
	0004	05/15/89	77.50	50.64	52.45	46.20	43.14	.00	275	276	277
		280	281								
	0005	06/15/89	63.34	26.53	58.10	.00	.00	627.48	285	686	287
		290									
	0006	07/15/89	89.27	24.69	59.48	.00	.00	.00	293	294	295
	0007	08/15/89	49.77	28.73	50.89	.00	.00	.00	300	301	303
	0008	09/15/89	81.01	22.99	54.49	.00	.00	.00	305	306	310
	0009	10/15/89	46.82	30.39	43.76	.00	50.64	.00	311	312	315
		320									
	0010	11/15/89	59.35	48.40	46.81	46.20	.00	.00	325	326	327
		329									
	0011	12/15/89	.00	.00	.00	.00	.00	.00			

CREATED 11/16/89 CHANGED 11/19/89
FIELD DESCRIPTOR TYPE WIDTH DEC

1	MODATE	D	008	
2	PHONE	N	007	02
3	GAS	N	007	02
4	ELECTRIC	N	007	02
5	SEWER	N	007	02
6	WATER	N	007	02
7	TAX	N	007	02
8	PCHK	C	004	
9	GCHK	C	004	
10	ECHK	C	004	
11	SCHK	C	004	
12	WCHK	C	004	
13	TCHK	C	004	
.SNAP				004
000	1 BILLS	00000/00012		

After I filled BILLS, I decided that a chequebook -type Db would also be interesting. I then proceeded to CREATE CHKBK. The STRUCTURE of BILLS is located in the bottom left corner of tutorial 12. After the CREATE screen, when TIB asked if I wanted to enter data, I answered no. This left the DB empty. First of all I hate to enter data, and second, this is a programming tutorial, so I wanted TIB to move the data for me. I used MODIFY COMMAND CHKNUM to whip up the little CF in the middle of the previous page. The top line of this CF is a comment line. The next line LOCAL NUMBER N 4, tells TIB to allocate space that I will use to store a N)number, with a length of 4 and label the space NUMBER. You should create the LOCALs you need at the beginning of the CF. If they are all together and up front, they are easier to examine when you are having problems. Note: It is unwise to create locals within a WHILE loop. REPLACE NUMBER WITH 245 puts the number 245 into NUMBER. I chose 245 because it seemed like a good place to start numbering the chequebook. If you wanted the first check number to be 1, you could use 1 instead of 245. USE CHKBK opens the chequebook Db. WHILE (NUMBER<330) is a loop that will go around and around as long as the value in NUMBER is less than 330. As long as NUMBER is

less than 330, all the commands you have placed between the WHILE and ENDWHILE will be carried out, if possible. The first command, APPEND BLANK, sticks one whole empty record on the end of CHKBK. Now that we have some space, let us store some data there. The next line takes our NUMBER, which is 245, and places a copy of it into CHKNO. CHKNO is my abbreviation for check number. The next line takes our number out of NUMBER (245), adds 1 to it (=246), and sticks that back into NUMBER. Without this line the check numbers would not increment by 1, and we would never reach 330 to get out of this loop. So this little CF takes the Db named CHKBK and fills it with numbers from 245 through 329. A short version of CHKBK is on the right side of this page. I added the deposit items at the top by hand, the check numbers were put in with CHKNUM and the other data was moved to CHKBK by modified versions of the CF named FLCKBKP, short for Fill ChecK BooK Phone. At this point there are many blank spaces in both of our Dbs, but that does not matter now. I want to press on with more refresher info and also some new commands.

```

CREATED 11/19/89 CHANGED 11/20/89
FIELD  DESCRIPTOR  TYPE  WIDTH  DEC
1      CHKNO       C      004
2      DATE        D      008
3      PAY'DEBIT   N      009    02
4      RET         C      003
5      FEE'CHRG   N      008    02
6      DEP'CREDIT N      009    02
7      BALANCE    N      010    02
8      REMARKS    C      040
000 1 CHKBK      00084/00108eof

```

REC	CHKNO	DATE	PAY'DEBIT	RET	FEE'CHRG	DEP'CREDIT	BALANCE	REMARKS
0085		12/31/88	.00		.00	.00	2349.23	
0086		01/01/89	.00		.00	389.23	.00	Pay ,
0087		01/16/89	.00		.00	421.56	.00	Pay ,
0088		02/01/89	.00		.00	265.98	.00	Pay ,
0089		02/16/89	.00		.00	398.12	.00	Pay ,
0090		03/01/89	.00		.00	267.61	.00	Pay ,
0091		03/16/89	.00		.00	404.63	.00	Pay ,
0092		04/01/89	.00		.00	376.39	.00	Pay ,
0093		04/16/89	.00		.00	267.22	.00	Pay ,
0094		05/01/89	.00		.00	449.99	.00	Pay ,
0095		05/16/89	.00		.00	329.94	.00	Pay ,
0096		06/01/89	.00		.00	502.34	.00	Pay ,
0097		06/16/89	.00		.00	359.33	.00	Pay ,
0000	245							
0001	246							
0002	247							
0003	248	01/15/89	56.16					Phone ,
0004	249	01/15/89	110.41					Gas ,
0005	250	01/15/89	68.88					Electric
0006	251							
0007	252	01/15/89	43.14					Water ,
0008	253	01/15/89	627.48					Tax ,
0009	254							
0010	255	02/15/89	111.32					Phone ,
0011	256	02/15/89	97.71					Gas ,
0012	257							
0013	258							
0014	259							
0015	260	02/15/89	58.24					Electric
0016	261	02/15/89	46.20					Sewer ,
0017	262							
0018	263							
0019	264							
0020	265	03/15/89	86.02					Phone ,
0021	266	03/15/89	99.64					Gas ,
0022	267	03/15/89	53.42					Electric
0023	268							
0024	269							
0025	270	04/15/89	44.81					Phone ,
0026	271	04/15/89	69.83					Gas ,
0027	272	04/15/89	54.39					Electric
0028	273							
0029	274							
0030	275	05/15/89	77.50					Phone ,
0031	276	05/15/89	50.64					Gas ,
0032	277	05/15/89	52.45					Electric
0033	278							
0034	279							
0035	280	05/15/89	46.20					Sewer ,
0036	281	05/15/89	43.14					Water ,
0037	282							
0038	283							
0039	284							
0040	285	06/15/89	63.34					Phone ,
0041	286	06/15/89	26.53					Gas ,
0042	287	06/15/89	58.10					Electric
0043	288							
0044	289							
0045	290	06/15/89	627.48					Tax ,
0046	291							

The CF named FLCKBKP is small but very powerful. If you can grasp the logic in its functions you will see your way to figuring out the use of TI-Base. "All right." The top line of FLCKBKP is for comments only. SELECT 2 tells TIB to switch to workspace 2 and open (USE) CHKBK in that workspace. Next TIB is told to go to workspace 1 (SELECT 1) and open BILLS in that area. We now have 2 Dbs open, CHKBK in work area 2 and BILLS in area 1. You will have trouble remembering where things are found, but TIB will not and when you forget and make mistakes, TIB will ask you what the heck you are doing. SELECT 1, USE BILLS and WHILE .NOT. (EOF) must be tied together in your mind. We want to leaf through BILLS one record at a time and do everything inside the WHILE loop until the End Of File (EOF) of BILLS is encountered. Note: At this point we are automatically looking at record number one in BILLS. SELECT 2 tells TIB to forget about BILLS (in 1) and start looking at CHKBK which is located in work area 2. TIB must be looking at CHKBK to use the FIND command in this situation. FIND 1.PCHK is a very powerful little statement. It says, go to work area 1 and find the value stored under PCHK. TIB does not care which record it is currently pointing at, that is up to you to remember. As I just said, we are looking at record one. Therefore, if you look across record one, just below the heading PCHK, you will find 248. That is the same thing TIB will come up with. CHKBK must be SORTed ON CHKNO. We are now looking at CHKBK (in 2), and the FIND command tells TIB to look at whatever field is sorted. TIB is looking through the CHKNO field for the number 248, which it just got from BILLS. If it finds a match for 248, it will not do anything, it just stops there with the record of the first Db aligned with the record in the second DB. This alignment will allow us to move data from one record to a corresponding record. If TIB searches the entire file and cannot find a match, it will stop at the End Of the File and there will be an (EOF) true. TIB does not tell you this, you must test for it. So the logic is this, if we hit the EOF then we did not get a match and that means we do not want to do anything. However, if we did not hit the EOF, or IF to CHKBK. In this case we REPLACE 2.DATE WITH 1.MODATE, REPLACE 2.PAY'DEBIT WITH 1.PHONE and REPLACE REMARKS WITH "Phone ,". Note: I have future plans that require the equal length of the first item in the REMARKS field. Also note that remarks have a length of 40, but I chopped off the extra length for this printout. As we pass out of the IF statement at ENDIF, TIB is working in area 2. We must switch back to area 1, where BILLS is and MOVE to the next record. When the ENDWHILE is encountered its job is to loop TIB back to the WHILE. The WHILE then tests to see if we have hit the (EOF) in BILLS. Note: If you did not RESELECT 1 just before the ENDWHILE the WHILE would look for the (EOF) in CHKBK. It does not care or know which Db it is working on. You must realize these things and tell TIB exactly what you want it to do. If all goes well, FLCKBKP will match up one of the PCHK field in BILLS with a CHKNO field in CHKBK and copy the date, amount and "Phone ," into CHKBK. This CF only covers the Phone data. I have my data disk in drive 2 and TI-Base in drive 1, so I typed COPY DSK2.FLCKBKP/C DSK2.FLCKBKG/C GO <E>, then COPY FLCKBKP/C FLCKBKE/C GO <E>, etc. until I had a copy with the name changed for Gas, Electric, Sewer, Water and Tax. Then I used MODIFY COMMAND to edit each CF. I changed the first letter in the FIND statement field (PCHK) to represent Gas, or Electric, or Sewer, etc. I also changed the last fieldname in the middle REPLACE statement (PHONE), to GAS, or ELECTRIC, etc. Last I changed the REMARKS entry from "Phone ," to "Gas ," or "Electric", etc. to correspond to the CF I was working on. You need six different CFs to handle all the xCHK fields in BILLS. It is really not that hard to do, plus you will make a lot of mistakes and by the time you correct everything you will have a lot of experience working with CFs.

Here is something new. PRINT ALL ;FOR CHKNO=" ". The part that I haven't discussed before is (;FOR CHKNO=" "). This is basically a filter. It filters out everything you do not want printed. In this case I have opened CHKBK (USE CHKBK),

continued on page 2

7.4

Regional Group Reports

Meeting Summary For October

Banana Coast	13/10/91	Sawtell
Central Coast	12/10/91	Saratoga
Glebe	10/10/91	Glebe
Illawarra	14/10/91	Keiraville
Liverpool	11/10/91	
Northern Suburbs	24/10/91	
Sutherland	18/10/91	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

The August meeting was well attended with a wide range of topics being covered. Herbert Shade brought along his recently constructed Ramdisk which is fast approaching completion. It is hard to believe that Herbert was working with a basic console only six months ago, as his system is now fully expanded. We also looked at some of the latest Picture files for the Page Pro software. In the week prior to the meeting I had received telephone calls from two potential new members, one of whom was looking for a replacement console. Percy Harrison was able to assist in this regard. I wonder how many more T.I.'s are out there, locked away in cupboards? 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.

The October meeting will be the 3rd Buy Swap and Sell day of the year. Come along with any gear that you want to sell or swap and also, be on the lookout for a bargain.

* 12noon to 2pm - Assembler Class. Bring your E/A book, note pad, pen and a formatted SSSD disk.

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

November	13 October
December	10 November

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

TISHUG Meetings for Sydney, 1991

October

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

November

The second all day tutorial session. Contact Russell Welham as soon as possible if you would like a particular tutorial session to run. He would also like to hear from volunteers willing to give a session on any topic.

December

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

Terminal Emulator II Tip

by Daniel Harris

Regarding the use of the TE II module and its mystery buttons; I just found one that will stop the sending of text instantaneously! It is CTRL <7> on the keyboard overlay that comes with the TE II package, the 'PAGE' control, which takes care of that screen just where you want to cut it. No more waiting for that CTRL <S> to stop the flow of text. There must be Users out there who will value this little tip. Of course, the TE II enables a screen of speech to be sent, enables saving to cassette or disk, as well as printing, has fancy keys for sending data from an external device I have yet to fathom, and can be used to send all upper case text. If I press CTRL <6>, which I just did - no miracle! Well, now I cannot type anything in upper case! What happens now? A CTRL <6> supposedly enables both upper and lower case. Hmmmmmm

continued from page 19

SCR #65

```

0 ( WORMY DATA EGR 1/3/85) HEX
1 9818 3CFF FF3C 1818
2 1818 1818 OF07 0000
3 183C 7EE7 C300 0000
4 1838 70E3 C70F 1C18
5 1838 78FC DC0C 1C18
6 1838 70E0 E070 3818
7 181C OEC7 E370 3818
8 181C OECF FF78 3000
9 0000 00C3 FF7E 3C18
10 0000 FFFF FF00 0000
11 001C 3EFF F370 3818
12 181C OEOF OF0E 1C18 DEC
13
14
15

```

C U next time, May the FORTH be with U.

74
 152 A453
 152 BT 14.2