



NEWS DIGEST

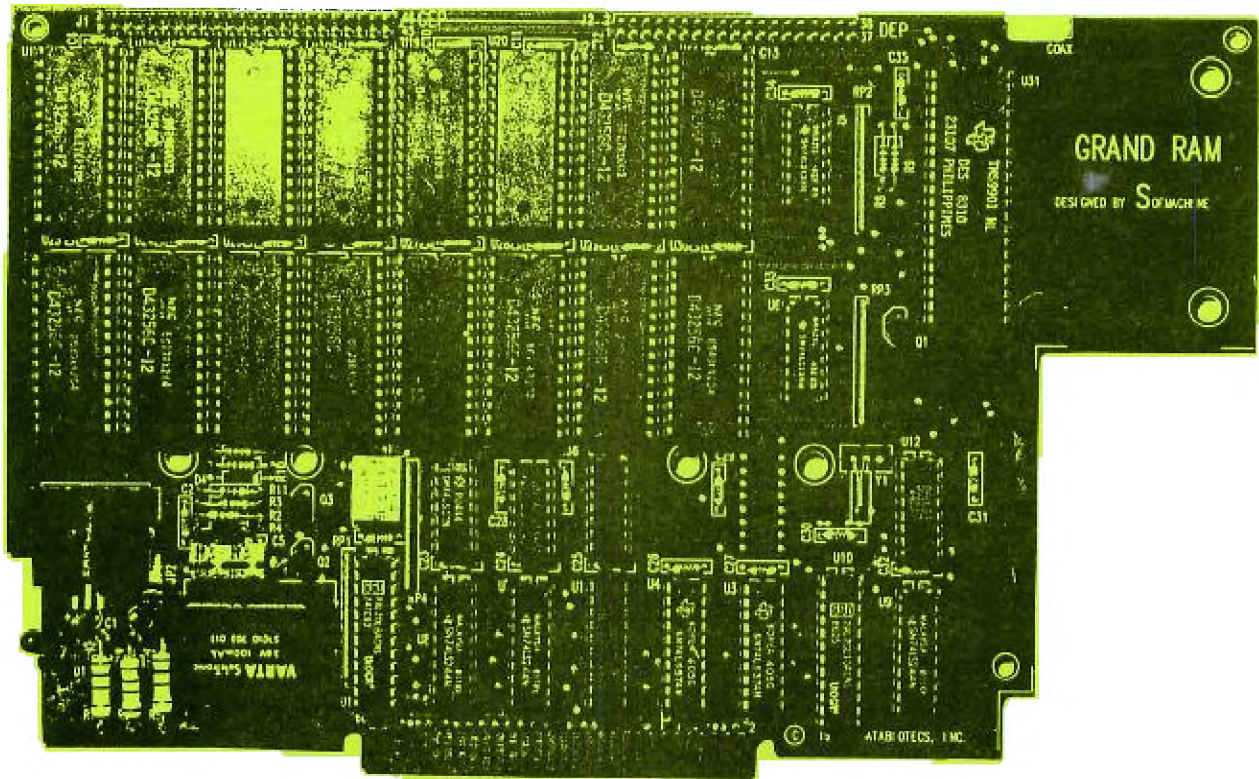
Focusing on the TI99/4A Home Computer

Volume 8, Number 11

November, 1989

Registered by Australia Post - Publication No. NBH5933

Grand RAM Card



by DataBiotics

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

\$2

TiSHUG News Digest

Index

November 1989

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
Rolf Schreiber (042) 84 2980
Directors
Robert Peverill (02) 602 4168
Russell Welham (043) 92 4000

Sub-committees

News Digest Editor
Geoff Trott (042) 29 6629
BBS Sysop
Ross Mudie (02) 456 2122
BBS telephone number (02) 319 1009
Merchandising
Steven Carr (02) 608 3564
Publications Library
Warren Welham (043) 92 4000
Software library
Terry Phillips (02) 797 6313
Technical co-ordinator
Lou Amadio (042) 28 4906

Regional Group Contacts

Carlingford
Chris Buttner (02) 871 7753
Central Coast
Russell Welham (043) 92 4000
Coffs Harbour
Kevin Cox (066) 53 2649
Glebe
Mike Slattery (02) 692 0559
Illawarra
Geoff Trott (042) 29 6629
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 \$25.00
Overseas Airmail Dues AUS\$50.00

TiSHUG Sydney Meeting

The next meeting will start at 9.30 am on 4th of November at Shirley House, Ethel Street, Burwood, just at the side of Woodstock. Entry fee of \$2 per family.

Printed by
The University of Wollongong
Printery

| Title | Description | Author | Page No. |
|---------------------------------|------------------------|-----------------|----------|
| Co-ordinators report | General news | Warburton,Dick | 2 |
| Communicators | BBS information | Mudie,Ross | 5 |
| Communicators extra | BBS information | Mudie,Ross | 5 |
| Direct I/O interface | Hardware project | Amadio,Lou | 7 |
| Disk utilities, XB tips | Software review, hints | Swedlow,Jim | 23 |
| Extended BASIC Tutorial | Software hints | McGovern,Tony | 13 |
| Fairware author of month | Club news | Trott,Geoff | 3 |
| Forth tidbits 7, 8 and 9 | Software hints | Winkler,lutz | 27 |
| Forth to you too! | Software hints | | 27 |
| From the bulletin board | Mail to all | | 5 |
| Full day workshop | Club news | Sheehan,Craig | 4 |
| Glossarial on acronyms | General interest | Murrell,William | 6 |
| I/O interface PCB hints | Hardware project | Amadio,Lou | 8 |
| Newsletter update | General interest | Amadio,Lou | 34 |
| Printer problems | Software hints | Peterson,Jim | 31 |
| Program to type in | Statistical parameters | Livingstone,E | 17 |
| Rambles from UK | General interest | Shaw,Stephen | 12 |
| Regional group reports | General interest | | 35 |
| Secretary's notebook | Club news | Phillips,Terry | 3 |
| Shahzada endurance test | General interest | Mudie,Ross | 9 |
| Techo time | Hardware project | Amadio,Lou | 7 |
| Ten commandments for technician | General interest | | 8 |
| They're off | General interest | Trott,Geoff | 1 |
| TI-Base tutorial | Data base | Smoley,Martin | 21 |
| Tigercub public domain catalog | Software review | Peterson,Jim | 25 |
| Tigercub software catalog | Software review | Peterson,Jim | 15 |
| Tips from the tigercub #36 | Software hints | Peterson,Jim | 19 |
| TiSHUG software column | Club software | Phillips,Terry | 3 |
| XXB documentation | Software review | Traver,Barry | 30 |
| Younger set | Program, adventure | Maker,Vincent | 33 |

TiSHUG Fairware Author of the Month
The Fairware Author for this month is Stuart Olsen for his Program Mass Transfer. All Donations collected at the meeting and sent in will be mailed to him this month

They're off

by Geoff Trott

It has been very hectic for the last week and a half. If you are reading this before the November meeting in Sydney, it is because Rolf has done a fast (and excellent as usual) job of pasting up and the printer has delivered on time. I was only away for one week but it seemed to put everything back about two weeks. Of course we have a short deadline this month. While I was in Adelaide I had a chat with Colin Cartright and Fred Cugley. It turns out that Colin lives a few hundred metres from my brother as the crow flies. I brought back Fred's hard disk controller card to try out with my hard disk as they do not have any hard disks there yet and they have not been able to get the card to work with floppies alone. I was not able to make it work with floppies alone at work and I have not had time to try it with a hard disk at home. There is only one more TND to be produced this year after this one and then I may get a bit of free time for a week or two. Apart from that we had an interesting talk about computers and the TI99/4A in particular. Fred has been at the helm of the Adelaide group since it started and feels that if he stops then no one will take over and the group will disappear. This seems to have happened to the Canberra group. It would be a pity if this were to happen and I offered the resources of the editorial staff of TiSHUG to help them as much as we could if they need it. Colin is negotiating to do the repairs of all TI99/4A parts for TI. He is self employed and would be doing it as part of his business. He was busy with a monitor when I talked to him.

continued on page 34

Co-ordinator's Report

by Dick Warburton

I have had some interesting experiences at work recently, with computers. Our Regional Office has a number of the latest and brightest IBM systems or clones. A number of them are being used, at times. However it becomes clear that very few staff have any idea of how to use a computer effectively, outside a very limited area. Without exception, the women seem limited to a single program they use in very limited ways, for example Word Perfect or a simple data base. Over the past week, the need to get some simple work done outside of the normal routine, created confusion and mild panic as the deadline approached. Staff are changed regularly, and often do not get adequate training on the major software packages provided. The result is that expensive equipment lies idle or is used in very routine and simple ways. I finished up taking my trusty TI99/4A into work, set it up and finished the job without a hiccup. It did the job simply and effectively. This set me thinking. I am sure that some of the expensive software bought by inexperienced managers from slick salesmen might be useful. My guess is, that most of it is a waste of money, and the computers are under used, because the majority of staff simply do not know how to use it effectively. The systems using MS DOS are too complex and difficult for the average user. I suspect that many other people are buying computers which are far more powerful than they need, or will ever use. My TI99/4A seems to grow as the need arises. In fact having tried a variety of systems at work, I find it a relief to go back to such a simple, easy to use machine. At this stage, I have yet to see another machine with a battery backed RAMdisk. The RAMdisk makes the TI99/4A a pleasure to use. It seems to me that the TI99/4A has become such a useful and effective computer primarily because of the activities of clubs like TISHUG around the world. I wonder how far we can go with its development. I was amused by the reaction of our office computer guru to the TI99/4A. He was very surprised that we are able to run Multiplan effectively on the TI99/4A, because "it could not possibly have enough memory".

A recent Asgard News article, referred to the new chip development by TI, namely the 34020 graphics chip. This chip was developed from the earlier 9900 CPU, and is now apparently the best graphics chip available. It is a 32 bit chip, and while it is magnificent for graphics, eg in VGA cards, it is also an excellent CPU. Asgard make the point that it should be relatively easy to design a high functioning computer around it, which can emulate the TI99/4A, and become the machine of the future for TI99/4A users. It would mean fantastic graphics, a massive increase of addressable memory, and tenfold increase in speed where it is needed. I wonder if we have anyone in Australia who could take up the challenge of the 34020 and its successors. I find the continuing expansion of the TI99/4A fascinating, especially when the results are achieved by the efforts of club members, and not commercial enterprise.

Our own small efforts continue locally. Our thanks to Lou Amadio, for his tremendous effort this year in getting a simple expansion system off the ground. It would be great if we could interest outsiders in this project. I want to build one myself for work use. It is clear that the RAMdisk is not yet finished. If the demand is sufficient we will get more boards and parts to build them. There is apparently considerable interest in the new project to build a RAMdisk with 64K EPROMs fitted on it. Craig Sheehan has done all the preliminary work, tested its logic in my PE box, and believes that the operating system is basically simple to build. Craig assures me that it is feasible to put up to fifteen 64K EPROMs on a single 32K chip RAMdisk. EPROMs cost approximately 12 dollars each, while 32K chips cost 2 or 3 times that amount. Imagine all your basic software in EPROMs, instantly available. We would be able to use RAMdisk memory in other ways, eg to run data bases, etc. I am quite excited about this prospect.

I am proposing that we set up a project group, to meet probably fortnightly, to develop and work on our individual, and perhaps group projects. We would need to keep in close contact with Newcastle and our friends in Wollongong, so that we can work together, and make better progress. Who knows what would develop if we pooled our individual skills and talents in this way. We need a place to meet which is relatively central, where we can spread out, and do our own thing. Does anyone have any ideas about where we could meet. While talking about hardware, I want to remind you about our console repair group which meets at Cyril Bohlsen's home each month on the Friday evening before the main meeting. I am hopeful that we can train more members to diagnose and repair consoles, because I intend to advertise our willingness to do so for TI99/4A owners who wish to join the club. Keep your eye on the Trading Post. I say again, if I can learn to fix consoles, anybody can. Just a word of thanks to John Paine for his invaluable help to someone like me. Imagine if we could clone John Paines and get them working on a project together.

We need some volunteers for some small tasks. I want a member or members to accept the responsibility of contacting visitors and new members, both at the meetings and at other times. We need to make anyone new welcome, and we need to meet their needs. Please help if you can. If you see someone you do not know at a meeting, go and talk to them. Offer to help them if they need your help.

Secondly I would like someone to accept the responsibility for placing ads in the Trading Post each fortnight, and to ring up any adds for TI99/4A equipment offered for sale so that we can advise people about the club.

Thirdly I want a member to be willing to collect the names and addresses of TI99/4A owners who are not members at this time. We will contact them and bring them up to date about the TI99/4A.

Next meeting promises to be a great day. Craig's tutorials sound really interesting. I hope the console repair guys get an opportunity to go to some other activity. They were so busy last time. The tutorial day will be held at Shirley House, behind Woodstock. Other meetings coming will include our Christmas barbeque, and a committee is busy working on it. I hope that we can run another hardware sale day fairly soon, as the last one was so popular. Most of you know that I am unhappy about the actual location for our meetings. We cannot rely on getting sufficient rooms available at any time. Craig has been frustrated by the room changes, and the lack of space. It is generally accepted that young people come to computers through games interest. I feel it is important to have a games room for our younger visitors. I have found a cheaper more suitable site, and hope that we can make the change in 1990.

Do not forget to try our printer re-inking service. See Rob Peverill. He does a good job if the ribbons I got back are any guide.

Do not forget to use our services this month. Call up the bulletin board, or if not a member of the BBS, then ring Ross Mudie and join that exclusive band of insomniacs. It is fun.

Do not forget to go to your next local or interest group meeting.

Do not forget our library next meeting, and do not miss our disk library copying service.

See you all at the next meeting, Dick Warburton. o

For Sale

Cicada 300, 1200/75 baud modem complete with phone in perfect condition. \$140 or nearest offer. Contact Percy on (02)808 3181.

Secretary's Notebook

by Terry Phillips

Members at the October meeting will recall some discussion on possible change of meeting venue. As mentioned Dick Warburton is going to source out his contacts to see if any likely school venues may be available in the Burwood area. I think we would all agree that Woodstock itself is a fairly handy place to hold meetings and it is certainly in a very nice setting. The big problem encountered is the allocation of rooms. While we are guaranteed one room, and on most occasions end up having two, we cannot be guaranteed the same room at each meeting. In other words as Woodstock is a community centre we have to fit in with their needs. As an example, our next meeting is a full day event and we would need at least five rooms all day. This we cannot have as Woodstock has a long standing reservation for a wedding reception which will occupy all the downstairs area throughout the afternoon. Woodstock management have offered us the use of Shirley House for the day. For those who do not know, Shirley House is managed by the NSW Blind Society and is just over the back fence, through the tennis court, from Woodstock. While it may not be ideal, I am sure we can adapt and make the most of what we have available.

When, and if, we do move our meeting venue it will need to be to a place that is reasonably accessible to the majority of members. With this in mind I have done a break-up of the membership data base and this shows the following dispersion:

| Members: | Area: |
|----------|--------------------------------------|
| 1 | Sydney City |
| 6 | Eastern Suburbs |
| 12 | Inner Western Suburbs |
| 23 | North Shore and Northern Beaches |
| 13 | Ryde, Pennant Hills, Carlingford |
| 29 | Paramatta, Hills, Blacktown areas |
| 9 | Penrith, Blue Mountains |
| 8 | Liverpool and surrounds |
| 5 | Bankstown and surrounds |
| 7 | Campbelltown and surrounds |
| 23 | Hurstville, Sutherland and surrounds |
| 6 | Newcastle area |
| 8 | Central Coast |
| 22 | Wollongong and South Coast |
| 15 | Other NSW country |
| 15 | Interstate |
| 4 | Overseas |

From this it will be seen that membership is fairly widely scattered throughout all areas so it will not be an easy task to please everyone if we do move. If you have any thoughts on this matter please discuss them with one of the Directors.

The group continues to attract new members and it is a big welcome to the following:

Robert Keast of Dapto
Mark Stuart-Street of Figtree
Sylvia Kwok of Chatswood

Hope you can all get along to some of the meetings.

It was good to see Rolf back on deck at the last meeting after his recent spate of bad health. I also hear that our former Treasurer, Bert Thomas is on the sick list and has been unable to get down to the last few meetings. Hope you are soon on the mend Bert.

An energetic committee has been formed and this group will be the organizers behind this years final meeting to be held on the 2nd of December. This meeting will be a social affair with food and drink being provided by the club. There will also be some prizes for the younger ones in attendance. Come along with the family to this one as it promises to be a great day.

That is it for this month. See you at the meetings.

TIshUG Software

Column by Terry Phillips

Again this has been a quiet month in so far as new software arrivals. In fact no new disks have been received. If any member has anything they would like to donate to the club, by all means feel free to do so.

The first lot of TI-Sort sold out at the October meeting, so a further order has been dispatched for another 10 copies. In addition as there appears to be still a lot of interested buyers for TI-Base and TI-Artist a further 5 copies of each are being obtained. These should all be available at the November meeting.

As the November meeting is a full day event, the whole of the disk library will be there for you to choose from, so bring along a few blank disks to get what you need.

See you at the meeting.

Fairware Author of the Month

We, the users of the TI99/4A, rely on many people for our enjoyment of our computer, none more so than those who have written software which we use and rely on every time we use our computer. Some of this will be commercial software which we should have paid for and received value for our money in the form of a working program with good documentation, but the majority of software will be Fairware, which may not have cost anything and yet still provides a working program and good documentation. Software authors who produce good useful programs and release them for us all to enjoy under the fairware concept are the ones who are keeping us all going. If you look at the price of commercial programs for other computers which do the jobs that we are able to do with our fairware programs, you will find that \$100 will not buy more than 1 program and you may well need \$1000 to get a state of the art program. Fairware software costs the price of a disk initially but if we use the program the onus is on us to send a contribution to the authors to repay them for their efforts and encourage them to continue development and perhaps write a new program as well. We can be sure that these authors are not relying on our contributions to live, as they do not ask enough and we do not send enough, if anything at all.

TIshUG now offers us alternatives to sending the money direct to the author. Of course sending the money direct to the author is the best way to get on an author's mailing list and to ask some pertinent questions about the software or about improvements which might be made in the next release. TIshUG is offering to collect money from us for fairware authors and to send it on in the correct currency. This allows us to contribute each month by mail or in person to the monthly fairware collection or to send in a contribution to be spread amongst several products and their authors. If you use this last method, be sure to send in a list of software and the amounts for each.

The Fairware software product for this month (same as last month because of administrative problems) is Mass Transfer from Stuart Olson in Illinois. This is a terminal emulation program which I find easy to use with our BBS and with very few problems. It has Xmodem protocol available, which is not supported on the BBS but is an excellent way of sending files to another user directly. It also has a multiple file transfer Xmodem in which only one transfer is initiated and all the files are sent and appear on the receiver's disk as the separate files with their correct names. If you use this program then please send in a contribution. Do not forget the other authors of fairware software that you use and take up TIshUG's offer of forwarding your contribution on to them.

Full Day Tutorial

TIIsHUG's final full day workshop for 1989 will be held at Shirley House, Ethel Street, Burwood on Saturday, November 4. Shirley House is directly behind Woodstock Community Centre, and can be reached by walking through the gate on the tennis court at Woodstock.

The doors will open at 9:30am and formal activities begin at 10am. A lunch break will be taken between 1pm and 2pm, after which the afternoon activities will continue until 5pm. To cover the cost of hiring the venue, there will be an entry fee of \$2.00 per family. You will need to bring your own lunch, but refreshments will be provided. To aid in planning your day, a summary of the events are given below.

1. Electronics for Beginners. Presented by Ross Mudie. 10am to 1pm.

This is a beginners tutorial that will explain the basic concepts behind electronics. No knowledge will be assumed.

2. Hands on Beginners Extended BASIC: Presented by Bob Relyea. 2pm to 5pm.

This hands on tutorial will teach programming in Extended BASIC with no assumed knowledge. In order to gain maximum benefit from this class, it is hands-on, so bring your console, monitor, a table, power board and extension leads as well as an Extended BASIC Module and manual. An expansion system is not required.

3. Console repairs. Lou Amadio. All day.

TIIsHUG's technical coordinator, Lou Amadio, and some members of the console repair SIG will repair faulty consoles subject to availability of parts and time.

4. Program clinic - Funlwriter. With Dick Warburton. 10am to 11 am and repeated at 3pm to 4pm.

Dick will assist any member who is having trouble using TI-Writer's or Funlwriter's editor and formatter on an individual basis.

5. Program clinic - TI-Artist. Presented by Darren Telford. 11am to 12noon and repeated 2pm to 3pm.

For the budding artists of the group, Darren will demonstrate how pictures are built up using the computer step by step.

6. Program clinic - TI-Base. Presented by Terry Phillips. 12noon to 1pm and repeated 3pm to 4pm.

If you have been having difficulty in setting up TI-Base or creating command files, Terry will answer your queries on a one-to-one basis.

7. Program Debugging. With Craig Sheehan. 11am to 12noon and repeated 4pm to 5pm.

If you are having problems debugging an Extended BASIC, Assembly or 'C' program, bring it on tape or disk to the workshop and Craig will help you debug it.

8. Hardware Advice - Hard Disks. Presented by Ben Takach. 10am to 11am and repeated 2pm to 3pm.

Considering expanding into a hard disk some time in the future? If so, Ben will demonstrate one for you and give advice on obtaining one.

9. Hardware Advice - RAMdisks. With Dick (RAMdisk) Warburton. 12noon to 1pm and repeated 4pm to 5pm.

Dick will demonstrate how to configure RAMdisks and will give advice on how to gain the best advantage from your RAMdisk.

10. Copython of software library. Terry Phillips. All day.

Take advantage of this opportunity to access the club's extensive software library. The only limitation will be a maximum of six disks at time (if there are other members waiting) and software available through the club shop will not be available for copying. For disk users, bring blank disks, whilst cassette based users are asked to bring blank cassettes and their cassette recorder.

11. Club Shop and Publications library. Steven Carr and Warren Welham. All day.

If you want to purchase any items from the club shop or borrow and publications from the library, these services will be available all day.

Amongst all of these activities, there is bound to be something that will interest you, so bring your \$2 entry fee, lunch, a note pad and pen as well as any other items needed for the activities you wish to attend and, simply enjoy yourself. ○

Craig Sheehan, Meeting Coordinator

continued from page 12

My reader has written in asking if I can stimulate some interest by setting tests! Well, I will try anything once. So, is there anyone living out there? Here is a little test to find out.

The object of this exercise is to stimulate a little active programming, and as it may have been some time since you last did any of this, we shall start off with something nice and easy.

No prizes. No closing date. As many entries as you like. Any language you like, as many languages as you like! In each issue the best response will be printed and I am not entering, so if no one enters, you will not see any results, so there! Entries on disk or paper.

Test #1:

Here is an example first:

Take a four digit number (3025)

Separate it into two halves, of two 2-digit numbers (30 and 25)

Add these two numbers (30 + 25 = 55)

Square the result (55 x 55 = 3025)

And we have our first number!!

Required: A program (in any language) which will print out (to screen) every four digit number for which this is true.

This is a programming test in which the problem is spelt out. Now you must put together the best problem solving algorithm you can and program it! If you find BASIC easy, send in your BASIC entry then try to do it in another language. You have several to try: Extended BASIC, Forth, Pilot, Lisp, Logo, Assembly, c99, TP99, what have you?

From Los Angeles, comes a 13 page tutorial on TI-Base by Bill Gaskill. This is for version 1.02 but is equally applicable to Version 2. A trifle long to reprint here.

+++++

There are some lovely programs in some of the newsletters I see. Some of them are widely reprinted, yet I cannot offer them to you. Why? Because they contain significant errors which stop them functioning. Some of the programs we have reprinted contain errors (the ones we do not have the time to actually key in!) but we have always printed corrections (maybe an issue or two later!). In so many cases though the other groups do not print corrections; maybe nobody keys the programs in? Where at all possible (and time is sometimes a little short!) the listings you find here are from working programs, so if you have any problems, please drop us a line! Then we can print a correction. ○

=====

The Communicators

Special Interest Group for Users of the TEXPAC BBS
by Ross Mudie

Members may have missed my regular column over the last few months, I have been extremely busy with participation in a week long Endurance Horse Ride (no I was not riding I was doing radio, PA and computer with the T199/4A) and a very heavy work load. The work load problems stem from the number of people having problems with Fax machines which experience difficulty in communicating through the telephone network. There are an estimated 300,000 Fax machines in Australia at the present so I guess a few problems can be expected. My job these days is one of national technical support to Telecom people all over Australia to help to resolve Fax problems. The Endurance Horse Ride was also very interesting and is the subject of another article.

THE LIGHTNING SEASON.

Now to the topical items for this month's column. The summer season is fast approaching and with it the violent summer electrical storms. Your computer and modem are particularly susceptible to serious damage from lightning induced surges which enter via either the power mains or the telephone line. Even when the telephone line is underground it can still convey lightning surges into your modem. The best advice that I can offer to protect against lightning surge damage is to UNPLUG both the telephone line and power cord from your modem and computer when a storm is approaching. This does little to protect against the unexpected surge however. Telecom can fit a lightning suppressor, (for a fee), but it is not guaranteed that it will definitely protect against a very near lightning strike. The Telecom suppressor is a gas arrester which connects to each leg of the line and to a Telecom earth. Mains surge suppressor filters can be purchased to protect against surges on the power lines.

Regardless of surge suppressors the best way to protect against lightning damage is to UNPLUG from the mains and the telephone line.

PROBLEMS WITH TOUCHPHONE 200.

The recommended way to connect your modem to the phone line is in Telecom's mode 3. Mode 3 connects the incoming telephone line through a 611 permitted attachment socket before going to any other telephones as shown in Figure 1.

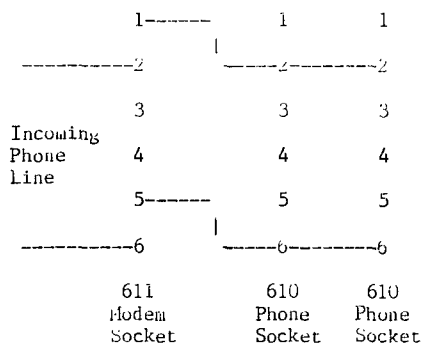


Figure 1.

When the modem connects to line it disconnects the telephones, no problems. If the modem is unplugged the contacts in the 611 socket make a connection from 2 to 1 and 6 to 5 so the telephones work normally.

Many modem users however just plug the modem into a 610 socket in parallel with other telephones or use a parallel type double adaptor. The problems with this is that if someone picks up a parallel phone then data corruption will occur. If a Telecom Touchphone 200

(TF200) is in parallel with the modem then after about a quarter of an hour the phone memory capacitor will discharge and the TF200 will become low impedance due to the effect of the fast charge circuit of the TF200. This results in severe data corruption.

The best method of overcoming the problem is have Telecom install a 611 socket in mode 3 for your modem, else ensure that any parallel TF200s are unplugged before the modem is used. Mode 3 'double adaptors' are also available, but this will only disconnect the phone plugged into the 'double adaptor', not other phones on the installation. Take special care to ensure that you have a mode 3 double adaptor as from the outside it will look very similar to a parallel type double adaptor. The 611 socket and the permitted attachment socket on a mode 3 double adaptor will have two raised half circles around the screw on the lid of the socket to differentiate it from the normal 610 socket. o

The Communicators extra

by Ross Mudie, 12th October 1989

PROTECTION UNIT FOR LIGHTNING SURGES.

I have just become aware of a new product which will provide protection from surges, both on the power mains and the telephone line. The unit, which is named "FAXGUARD", plugs into both the 240V power mains and the telephone socket. The computer equipment or Fax machine plugs into surge protected modem and power outlets on the FAXGUARD.

The 240V power circuit is surge clamped between active and neutral and from both lines to earth. There is also a series limiting element.

The telephone line also has series and shunt protection elements. The FAXGUARD provides clamping to mains earth under transient conditions thus no specialized telecommunications earth is required.

FAXGUARD carries Telecom Australia approval and is suitable for user installation. FAXGUARD is manufactured by Critec Pty Ltd of Hobart and may be purchased from Component Resources (NSW) Pty Ltd of Girraween NSW, phone (02)688 4528. The price quoted on 12/10/89 is \$195 + sales tax for one off purchases. o

From the Bulletin Board

MAIL TO : ALL
MAIL FROM : CO-ORD

The console repair group meets at Cyril Bohlsen's place at 4 Madeline Street, Northmead on the Friday night before the main meeting. A variety of things get done. Some consoles are diagnosed and repaired. Disk drives get looked at, monitors are discussed, power supplies get fixed, and generally we all have a good time. We put in a dollar each and Cyril's wife puts on a good supper. There is equipment available. The club has bought a console tester, chips, we have burned some of our necessary EPROMS, and we have some old boards to cannibalize. If you have an old console you do not want, give it to a good cause, the console repair group. We hope to be fixing consoles for new members soon. We need more members to learn how to become brain surgeons for consoles. Be in it. It is a good night. See you there next meeting.

Dick Warburton

MAIL TO : ALL
MAIL FROM : REX

Wanted. Starting Forth by Leo Brodie. Please leave message or ring (066)51 2485

continued on page 11

Glossarial on Acronyms

by William Murrell, USA

As a newcomer to Computing having had my TI99/4A for a couple of years, and not at all knowledgeable about computer language, I found that in reading computer magazines, the use of acronyms tended to make me think that a new language had been invented which somehow had never been taught at school. Reality set in when the cost of putting everything down on paper was calculated. Printing and space demands plus the cost of that space, made the use of acronyms an essential part of publishing.

We have had acronyms with us for many years, there everywhere around us and part of our everyday life. We have a tendency to overlook them, unless they are a part of the things we do at work or as part of the professional language we use. There are many of you who may read this article who will already understand what most of the acronyms mean. This is really for those of us who require a little help to further our knowledge. There may even be some of you that have no idea what acronym means. According to my understanding of the word (which unfortunately is not listed in my Dictionary), the use of the beginning letter of a word to mean the word. For example, PC for Personal Computer.

Some of the books dealing with computers have in the back of them, a listing of words, called a Glossary. This word, according to the dictionary, means "A vocabulary explaining antiquated or difficult words or phrases". Now how about that! The Computer is antiquated! Well, seeing as how many new ones are brought out every year, we can suppose, that those that get superseded, are antiquated, and with the cost of advertising space, it is the use of acronyms which makes for less space so less cost. This will be a Glossarial, which means "Containing explanation". Some acronyms are self explanatory, others need a little help, in which case the Glossarial will be Glossarial.

| ACRONYM | EXPLANATION |
|---------|---|
| A/D | Analog to Digital converter |
| AL | Assembler Language |
| ASM | Assembler |
| AT | Advanced Technology |
| BASIC | Beginners All-purpose Symbolic Instruction Code |
| BAUD | Rate Unit for character transmission |
| BIOS | Basic Input/Output System |
| BIT | Binary Digit, either 1 or 0 |
| BOOT | To load an operating system |
| BPS | Bits Per Second, see baud |
| BUG | Error(s) in program |
| BUS | Group of wires leading to specific destination |
| CAD | Computer Aided Design |
| CBASIC | CP/M BASIC |
| CCP | Console Command Processor |
| CGA | Colour Graphics Adaptor |
| CP/M | Control Program for Microprocessor |
| CPU | Central Processing Unit |
| CRT | Cathode Ray Tube (TV and Monitor) |
| CSGD | Character Signal Graphics Design |
| DB | Data Byte (one byte of data) |
| DDT | Dynamic Debug Tool (disk based electronic unit) |
| DIP | Dual Inline Package |
| DLE | Data Link Escape |
| DMA | Direct Memory Access |
| DMM | Direct Memory Mapping |
| DOS | Disk Operating System |
| DSDD | Double Sided Double Density |
| DSQD | Double Sided Quad Density |
| DSSD | Double Sided Single Density |
| DTA | Data Transfer Area |
| DTAB | Data Transfer Area (Buffer) |
| DTP | Desk Top Publishing |
| EDI | Electronic Document Interchange |
| EGA | Enhanced Graphics Adaptor |

| | |
|---------|--|
| EPROM | Erasable Programmable Read Only Memory (ultra-violet erase and reprogrammable) |
| FIFO | First In First Out buffer |
| FLIPPY | Single Sided disk mangled to become a poor man's double sided disk |
| FLOPPY | Disk made from flexible magnetic material |
| GLITCH | Bug or error caused by electrical surge or noise |
| I/O | Input/Output |
| IDC | Insulation Displacement Cable Connectors |
| ISO | International Standards Organization |
| K | Kilobyte=1024 bytes (KBYTE) |
| KB | Kilobaud=1000 baud (KBAUD) |
| LCD | Liquid Crystal Display |
| LSB | Least significant Byte or Bit |
| MDA | Monochrome Display Adaptor |
| MSB | Most Significant Byte or Bit |
| MSBASIC | Microsoft BASIC (IBM and Compatibles) |
| NAD | Name and address (File or Program) |
| NLQ | Near Letter Quality (Printers) |
| PCB | Printed Circuit Board |
| PEL | Picture Element also called Pixel (Dot of Video Graphics) |
| PGA | Professional Graphics Adaptor |
| PIP | Peripheral Interchange Program |
| RAM | Random Access Memory (can be both written to and read) |
| RGB | Red Green Blue (Video signal, colour components) |
| ROM | Read Only Memory (Permanent and cannot be altered) |
| RrGgBb | Enhanced Graphics Adaptor |
| TPA | Transient Program Area |
| TSR | Terminate and Stay Resident Receiver/Transmitter |
| TTL | Transistor to Transistor Logic |
| TTY | Teletype Video Signal |
| UART | Universal Asynchronous Receiver/Transmitter Chip |
| XT | Extended technology |

There are still hundreds of acronyms that have not been mentioned (not by me anyway). Quite a few are listed in the printer manuals for the operation of the printer. These of course deal with ASCII and most of those can be found by perusing the ASCII tables.

Now just a few notes about monitors and cards. These have been a particular bug with me.

| | |
|---------------------------|--|
| MDA | Monochrome Display Adaptor, green or amber, displays text 80 columns by 25 lines, with character cell 9x14 dots, TTL, digital |
| CGA | Colour Graphics Adaptor, RGB, TTL, digital, RES (Resolution) 600x200 pixels horizontally by vertically on screen area, 2 colours from palette of 16 on screen. |
| EGA | Enhanced Graphics Adaptor, RrGgBb, TTL, digital, RES 640x350, 16 colours from palette of 64 on screen. |
| VGA | Video Graphics Adaptor, RGB, analog, Multiscan, High RES 640x480, 16 colours from palette of 256,000 on screen. Low RES 400x320, 256 colours from a palette of 256,000 on screen |
| PGA | Professional Graphics Adaptor, RGB, analog, Multiscan, RES 640x480, colours 256 from a palette of 4096 |
| Multiscan | Monitors that can display several different scan frequencies depending on mode setting. |
| Scan Frequencies | Horizontal and Vertical scan rates, measured in lines per second to be displayed |
| Graphics Design (display) | Displays images as groups of pixels. Position of each pixel is determined by software (program). |
| Text Display | Text characters are pre-determined, by hardware (ROM) or software (disks, programs, RAMdisk, etc.) |
| GBFN | Good Bye For Now (WM) |

For Sale

TI99/4A console, complete for \$60. Phone (042)84 2960.

Techo Time

with Lou Amadio

Disk Drives

Rolf informed me that the double sided (full height) disk drives, sourced by new member Bob Keast, sold out in a very short time. These were going for \$35 each. If anyone else is interested in purchasing a drive, please let one of the committee members know. I understand that there are some single sided drives for sale at about \$20 each and two 35 track drives in excellent condition for \$10 each.

I/O Interface PCBs

PCBs for the Direct I/O Interface project described in recent issues of the TND should be available by the time that you read this newsletter. Price is as yet unknown.

RGB Interface

Geoff Trott is currently working on an interface suitable for connecting a Wang colour monitor to the TI99/4A console.

P-GRAM Card

Since I have been on holidays, I have not had a chance to build the P-GRAM card. I should, however, be able to finish it this month.

Next Month

The hardware project next month will be a DIY 32k memory expansion for the console.

More on Direct I/O Interface

by Lou Amadio

The article published last month was not to be last word on this subject. Indeed, I may never hear the end of it! Interest in this project seems to be quite high as it is a low cost way to expand your system. Furthermore, I expect that the club shop will soon have available kits of the Peter Schubert Multifunction Card (see above).

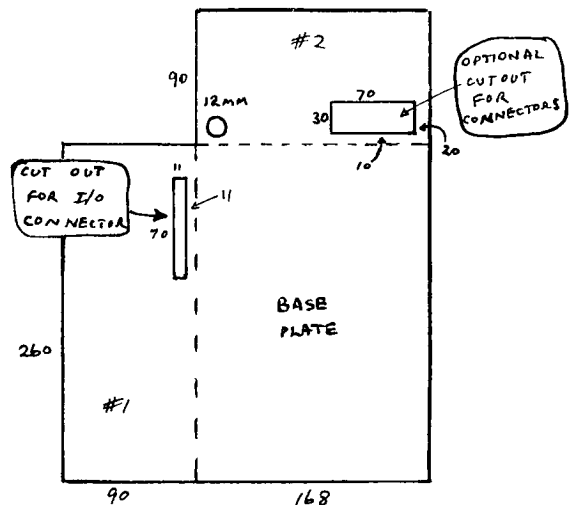
Although the Interface, as designed, can be used in a free standing system, I elected to mount mine inside a metal box. I designed the box to hold not only two PEB cards, but also a slimline disk drive as well. The lower slot in the I/O Interface was used for an optional card (in my case a RAMdisk), while the upper slot was used for the disk controller card in a semi-fixed configuration. By this I mean that the DCC was bolted in the upper position and supported by the aluminium chassis. I then mounted the disk drive on top of the two PEB cards. The finished box was about the size of a full height drive and the depth of the console.

Building the Box

Mark and cut the three pieces of 1 mm thick aluminium sheet, as indicated on the attached diagrams, one for the base plate, one for the drive cradle and one for the cover. Parts of the diagram are numbered for easy reference

Base Plate

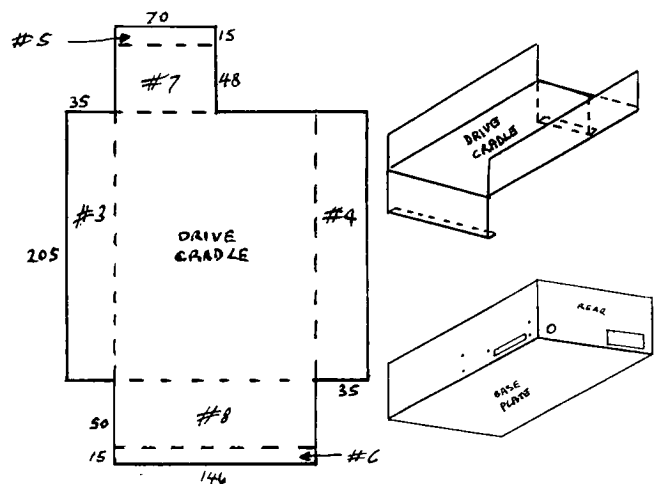
Cut out the slot in the base plate, as indicated, for the 44 way console connector. Drill a 12 mm hole in the rear (#2) for the interface power cables. You may also wish to cut a suitable hole in the rear of the base plate (#2) in order to facilitate connection to the outside world (eg RS232/PIO). Bend sides #1 and #2 upwards to form two sides of an open box. Join #1 and #2 at the top edge with a piece of scrap aluminium for added strength.



The completed I/O PCB (see separate article, TND Oct 1989) is then bolted to face #1 of the base plate. Plug in a PEB card into the lower connector, align the I/O PCB against side #1 of the base plate and accurately mark and drill the five 3mm mounting holes for the supports. Solder the power cables to the I/O PCB and mount the PCB using 20 mm bolts with 4 mm standoffs. Attach 8mm rubber feet beneath the base plate at the four corners.

Disk Drive Cradle

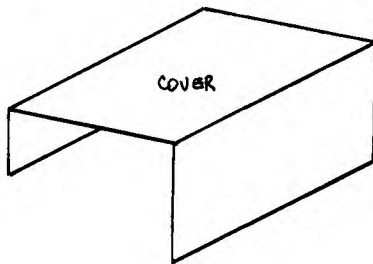
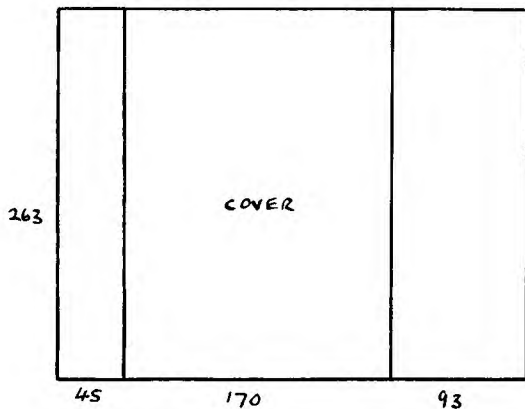
As the drive cradle has a number of complex bends, these must be done in the right order. Firstly, bend the two sides marked #3 and #4 up. These form the sides which support the disk drive. Next bend tab #5 up and tab #6 down (see diagram). Then bend supports #7 and #8 down as indicated. Drill two 2mm holes in tab #5 and three 2mm holes in tab #6. Mount the drive cradle on the base plate with the RHS of both the cradle and the base plate aligned. Mark and drill the 5 holes (for the tabs) in the base plate to accept small self tapping screws. Note that the front edge of the drive cradle (tab #6) goes BENEATH the front edge of the base plate. Also ensure the any screws do not interfere with the PEB cards. Drill the four holes on the sides #3 and #4 to secure the disk drive.



Cover

Sides #9 and #10 of the top piece should only be bent after final assembly and careful measurement of the remainder of the box. Drill retaining holes in side #10 to align with the screw holes on the disk drive side. Side #9 may be attached to the main chassis with self tapping screws and pieces of scrap aluminium. The 20mm gap at the front LHS of the box may be filled with a

piece of scrap aluminium bolted to the base plate. I finished off the front of my completed expansion box with a piece of Marvplate (steel sheet covered with black plastic). ◦



I/O Interface PCB Construction Hints

by Lou Amadio

If you are using the PCB design for the Direct I/O Interface, please note the following:

- 1) Prior to mounting any components, trim the PCB to within 3 mm of the top and bottom tracks to reduce overall height and to facilitate mounting in a box if required. You may have to relocate the PCB mounting holes. Do not trim the short sides
- 2) To assist current capability, tin the PCB track for the +9 volt supply line. For the same reason, the earth returns for the +9 volt (pins 1, 3 and 5 on the 60 way connector) should be connected together and then to the central earth rail.
- 3) Mount the 60 way connectors flush with the PCB.
- 4) Mount the 44 way connector at least 10 mm proud of the PCB.
- 5) Mount all components and wiring flush with the PCB so as not to foul the PCB cards.
- 6) Mount 0.1 uF caps across the 4 tantalum capacitors (solder side).
- 7) When making the through-link connections, bend the wire top and bottom so that there is no possibility of it falling out during soldering. This is particularly important where the links are beneath the 60 way connectors as there is no way of resoldering them later.
- 8) Take care not to insert the resistors in the wrong holes (ie those meant for through links). If necessary, refer to the connection tables published last month. ◦

Ten Commandments for Technicians

Author unknown, courtesy John Paine

1. Beware the lightning that lurketh in the undischarged capacitor, lest it cause thee to bounce upon thy buttocks in a most untechnician-like manner.
2. Cause thou the switch that supplieth large quantities of juice to be opened and thusly tagged, that thy days in this earthly vale of tears be long.
3. Prove to thyself that all circuits that radiateth and upon which thou worketh are grounded and thusly tagged lest they lift thee to radio frequency potential and causeth thee to make like a radiator, too.
4. Tarry thou not amongst those fools that engage in intentional shocks for they are surely non-believers and are not long for this world.
5. Take care that thou usest the proper method when thou takest the measure of a high voltage circuit lest thou incinerate both thyself and thy meter, for verily, though thee hast no account number and can be easily surveyed; the test meter doth have one and, as a consequence, bringeth much woe unto the supply room.
6. Take care thou tampereth not with interlocks and safety devices, for this incurreth the wrath of the supervisor and bringeth the fury of the safety inspector upon thy head and shoulders.
7. Work thou not on energized equipment, for if thou so dost, will thy fellow workers surely buy beers for thy widow and console her in other ways.
8. Service thou not equipment alone, for electrical cooking is a slothful process and thy might sizzle in thy own fat for hours upon a hot circuit before thy maker sees fit to end thy misery.
9. Trifle thou not with radioactive tubes and substances lest thou commence to glow in the dark like a lightning bug and thy wife have no further use for thee except thy wages.
10. Causeth thou to be tagged all modifications made by thee upon equipment, lest thy successor tear his hair and go slowly mad in his attempt to decide what manner of creature hath made a nest in the wiring of such equipment. ◦



At the Assembler Tutorial Weekend (April 1989)

Shahzada Endurance Horse Ride

by Ross Mudie

INTRODUCTION.

I have recently been involved in a 5 day endurance horse ride where my TI99/4A computer was used as the event bred computer. I also provided Public Address and operated the radio base. In this article I will introduce the event and the part played by the computer in the overall event. I will discuss some of the problems that were encountered and how various aspects of the computer interacted with the people at the event. I have also included an insight to some of the more unusual things that happened, although I guess that the subject of the whole article will be unusual to quite a number of readers.

The TI99/4A computer used a 32K memory expansion, Extended BASIC, 2 disk drives, RS232 card, RAMdisk, Triple Tech (clock and printer buffer), BMC BX80 printer and small colour TV. The computer was set up in my communications caravan which also housed the CB Radio base, Public Address system and my living and sleeping quarters.

THE EVENT AND THE SETTING.

The Shahzada Memorial Endurance Test is an annual 400km endurance horse ride which was named after the pure bred Arabian stallion, Shahzada. In 1920, 1921 and 1922 Shahzada won long distance endurance events and ensured his place in the history books. This event was the ninth Shahzada Memorial Endurance Test and was held in the St Albans (NSW) area, which is 20km up the MacDonald River from Wisemans Ferry. St Albans is set in a picturesque valley with steep hills all around. The valley, which is fed by the sleepy MacDonald river, contains fruit trees, grazing land and farm houses. The "township" of St Albans consists of the Settlers Arms Inn, an 1836 two storey sandstone building with a small shop and some accommodation buildings. Each day horse and rider must travel a colour marked course along roads, fire trails, across paddocks and through narrow bush tracks. The first leg each day was 50km and the second leg 30km. There was a compulsory one hour minimum break between the legs of the event which always occurred in town. The start time on the first day was 5am for all entrants whilst on subsequent days the start time was at the time that each rider wanted to start after 4am. The maximum riding time each day was set at 12 hours and all riders were required to be in by 5pm.

There is always very strict veterinary supervision and care at these events. The horses must recover to a heart pulse rate not exceeding 55 beats per minute (bpm) after the first leg on the first day and not exceeding 60 bpm after all other legs of the event. The pulse rate is checked 30 minutes after the return to town. The body weight of the horse was also weighed using a walk on scale platform which provided an electronic display. The Vet checks for lameness and back and leg problems. Any horse found lame or injured is scratched from the event.

INFORMATION STORED IN THE COMPUTER.

The following details were entered into the computer in random order. The files were organized in numeric, event entrant order.

NAMES - (A global file).

- Rider name, Class in the event, Previous event participation, Horse name, state, sex, age, breed, Sire and Dam. (Parent details of the horse).

VETDATA - (A global file).

- Information collated prior to the event included: Weight carried (rider and saddle), pulse, respiration, temperature and body weight.

DAYDATA - One file for each day, hence DAYDATA1, DAYDATA2 etc.

- The information contained was:

Previous cumulative time, start and finish time for each leg, pulse after each event leg and horse body weight. The computer calculated the riding time, average speed in km/h, day total time and total cumulative time. Details of Vet outs or withdrawals were stored in the DAYDATA files in addition to temperature and weather conditions 4 times per day. As the cumulative time was calculated it was also placed in the next DAYDATA file in the previous time column.

PRINT OUTS AVAILABLE.

PRE-RIDE INFORMATION - contents of the NAMES and VETDATA files (except the Sire and Dam information).

DAILY EVENT DATA - Rider number order, showing riders' names, class and the contents of the appropriate DAYDATA file.

DAILY EVENT DATA - In order of total riding time.
- Available slowest or fastest first.
- Gives rider numbers, names and contents of the DAYDATA file except scratched entries are not printed.

NAMES LISTS - In order of total riding time.
- Available slowest or fastest first.
- Gives rider numbers, class, horse name and total cumulative time.

All print outs included a standard event header, a date and time stamp (obtained from the Triple Tech clock) and a page number. The date and time are invaluable in sorting out which is the latest print out.

The weather conditions were a requirement of the chief vet who wanted to be able to equate the failures and performance to the prevailing weather conditions. The high humidity from the misty mornings coupled with low 20s spring time temperature was responsible for a number of the vet outs in the event.

When an entrant was out of the event that entrant was removed from the next day and all subsequent days numeric list print out. Whenever there was a break in the sequential number order one blank line was provided in the numeric order print out to draw attention to the discontinuity of the number range. The "empty" numeric list was of great advantage to the check points who knew what horse numbers to expect. Some of the riders were very surprised when the people on the check points knew their names (each rider wears a rider number).

A routine was provided in the program to produce a list in order of slowest to fastest. The ordered list of entrant numbers was then saved on disk for later reuse. The ordered list was then used to specify the order of reading of names and data out of the relevant RAMdisk files to produce a time ordered print out of 'data for a day' or 'names' either fastest to slowest or slowest to fastest. The ranking of fastest to slowest was popular amongst the riders and strappers whilst the list from slowest to fastest was used when arranging the order of presentation at the end of the event.

The print outs were of a great deal of interest to many of the people at the event. Print outs were pinned up on notice boards under the canvas annex outside the caravan. By the end of the week the grass in the notice board area was worn out to almost dust bowl conditions.

At the end of the event the computer produced a summary print out entitled "Veterinary Report" which extracted the details of each successful entrant from the Daydata files and printed a certificate for presentation. The production of this report, when prepared manually on previous years, required the Vet to have very little, if any sleep on the last night of the

event. The computer produced all of the reports in about 40 minutes, leaving the Vet the task of writing his comments on the appropriate part of the report only. The available time scale precluded typing up the comments.

THE COMPUTER AND THE PROGRAM.

I only had 5 weeks from when I received firm details of the event until the event was on, so the program had to be written, tested and debugged very quickly. In all a total of about 140 hours in 4 weeks of "spare time" was used. (The last week before the event was used to prepare other things.) I had a choice of trying a data base program, such as TI-Base of which I knew nothing, or to stick with TI Extended BASIC and linked assembly which I know reasonably well. There was no room for uncertainty since the task had a finite, very short deadline; TI Extended BASIC with assembly links was chosen. I used the 40 column utilities which I wrote three and a half years ago. I was very glad that I had fully documented the source files, it made the task of minor changes very easy and quick.

I needed an ACCEPT AT routine which allowed a size of 107 characters. This was where my 40 column accept routine was used in combination with a 40 column display routine. When entering or editing any day data the existing day data for that entrant was called up from the RAMdisk onto the screen. The string was then searched using POS to find the first occurrence of 5 consecutive spaces from which a TAB template was searched until the next TAB point was found. The TAB point was then passed into the 40 column ACCEPT routine which placed the cursor in the logical next input position in the entrant data on the screen. The data could then be entered or if some other change was needed then the cursor could be moved anywhere in the data field on the screen with the normal FCTN left and right arrow keys to allow the required change to be made.

The format of all files was RELATIVE which permits random access to any record in the file for reading or saving. This overcame the memory limitations of the computer. As soon as a record was modified and enter was pressed the string was checked to see if any calculations of time or average speed were required or if the entrant was out of the event. The calculations were a bit slow because Extended BASIC is quite slow when string manipulation is required. This is an ideal job for a linked assembly routine and I hope to do this before the event is held again next year. The wait of a few seconds whilst the computer was doing the sums for me and putting the answer in the right place was not wasted however; it gave just enough time to sort the vet card from which the information had just been obtained into numeric order with the other cards. As soon as a record was complete it was saved back to the RAMdisk as a precaution against power loss. A copy of the record was placed on the bottom of the screen to cover against the inevitable interruptions and the "now where was I?" on getting back to the computer. If a cumulative time had been calculated it was placed in the Previous Time column of the next DAYDATA file. All the time calculations were performed in minutes and then converted back to hours and minutes. The previous year's computer program (which I was not involved in) presented time in hours and decimal parts of hours (metric time??). That is 3.75 meant 3 hours 45 minutes; computers are supposed to work for us humans so I made the computer work harder and present all times in hours and real minutes.

A "flag" array was held in memory and on the RAMdisk to allow decisions of an entrant's status to be determined very quickly. A status value of 0 was for an unused entrant number. The names routine changed the status flag to 6 in the array and RAMdisk file when a name was entered. When any entrant withdrew from the event, vetted out or was disqualified the key strings were found by using POS after the entry of Daydata. This resulted in the flag for the affected entrant being changed to a value of between 1 and 5 to reflect the day of scratching from the event.

The program was configured for 219 entrants and the number ranges used were 1 to 34, 36 to 89, 100 to 104 and 201 to 209. The different number ranges indicated different classes of entrant. The flag array was used to inhibit printing of unused entrant numbers.

When the program first initialised each day it read the day character from the Triple Tech clock and set itself to the appropriate day with Sunday being treated as day 1 and Saturday as day 5 so that the program always had a valid day for use in the DAYDATA files. Provision was made to change the day by keyboard command so that print outs of different days could be made or so that data could be examined or changed if necessary for different days.

The computer print outs were keenly sought after with copies being taken back to the various camps and then scrutinized and discussed around the camp fires. The amount of data across the page was a problem. I chose to use compressed print to achieve 142 column maximum on my 80 column BX80 dot matrix printer. Some of the older people with sight problems had trouble reading the compressed print, so next year I hope to be able to provide a print out for half the day using standard size 80 column print or the whole day using compressed print size. The demand for print outs and the need for constant supervision of the printer was very time consuming. Two part self carboning printer paper was used to halve the number of physical print outs required but the extra thickness of the two part paper caused a number of paper jams. Next time a photocopier will be arranged to speed up the process of supplying copies. At this event I totally wore out 2 printer ribbons and could have used 2 more. Once on site at an event in a reasonably remote location like this it just is not practical to duck down to the local computer shop to pick up another ribbon. Thus any such involvement requires very careful planning and considerable foresight. After the event I made the computer print outs up into an A4 sized booklet. The book was produced on a photocopier on A3 size paper, then folded in two and stapled making a neat A4 size booklet of 28 pages.

The importance of the computer was further shown in the routine at the end of the day. Starting as soon after 6pm as the computer print outs were available, a pre-ride talk was held for the next day. It was quite a mad rush to finalize data entry for the day, check it against the inevitable typing or interpretation of other people's writing errors and then produce a list of entrants still in the event in order of cumulative riding time. (By the end of the week I had the routine flowing very smoothly.) Armed with details of who was leading, who was running last and who was out of the event the pre-ride talk was held under a light in the main street of St Albans in between two 44 gallon fire drums. The chief steward, Halifax Hayes, would stand on a ladder or a section of tree trunk (fire wood) and talk about the day's successes and failures and the course of the next day's event. The chief Vet also addressed the group each day from the rather makeshift podium and would discuss the main reasons for vet outs on that day and warn of the hazards of the next day and how to best manage the horses considering the expected weather conditions of the next day.

SOME OF THE OTHER THINGS THAT HAPPENED.

There was a problem with the 27MHz CB radio getting into the PA system. These problems are not easily fixed once the event is under way. The best "quick fix" that I could come up with was a relay with its contacts in series with the PA amplifiers output lines. The relay was operated by the push to talk button of the PA microphone. This worked fine until someone else tried to make an announcement whilst the CB radio was in use. The use of UHF FM CB radio equipment is planned for next time which should not cause the same RF detection problems as the AM or SSB.

Use of the 27MHz CB was an experience in how bad communications can be. The computer provided "birdies" on most channels of the CB with some channels unusable.

The interference from Taiwanese fishing boats caused a lot of problems on 27MHz, due to the constant chatter and whistling. The signals on CB from outside the area were much stronger than those from just a few kilometres away.

On the return path of each leg of the event a check point was established between 3 and 10km from town. (There were other check points for safety or control purposes.) As each entrant passed through the final check point the rider number was recorded and radioed back to town. Once the rider number was confirmed it was announced on the PA to enable the strappers to be up at the finish line ready to look after the horse as soon as it arrived. (The riders were second class citizens compared to the care given to the horses.)

Just to make the job a bit more interesting, I performed the three tasks of radio operator, PA announcer and Computer operator. Obviously priority had to be given to the radio communications and PA tasks. There always seemed to be a reasonable level of interruptions running when I was trying to get data into the computer thus the details of the previous entry that had been placed in the computer was a real blessing.

The response that I received from the people at the event was very encouraging. This made for an event that I enjoyed and that passed all too quickly. A point that I would make here also is that in this type of situation often one must continue to get the work done (computer entry etc.) rather than be able to join in the fun. In all I spent 9 days and 7 nights away from home. The 2 days prior to the event were occupied by setting up PA, radio, computer and other communications equipment. The 2 days after the event saw endless computer print outs being produced, all too slowly, together with the dismantling and packing away of all the equipment.

During the week the only problem with the TI99/4A was the need to clean the contacts of the Extended BASIC cartridge. The problem of the 2 part paper jamming in the printer was however a continual problem requiring constant supervision.

On the Wednesday of the event a notice was received from the electricity supply county council advising that the power would be turned off for the whole of the valley area from 8am to 3pm on Friday, the last day of the event, for power line maintenance. The locals were absolutely up in arms! They risked not having the event computer (first priority) and both the storage and provision of food for the people would also be in jeopardy. After the first phone call the planned power outage "had to" proceed as planned. About an hour later, after a barrage of irate phone calls the planned outage had been deferred to the alternate date.

In the town there was a fellow known as "doctor Sorebones" who offered a 45 minute "soothing massage" for ailing riders. He was not getting a lot of business at first until one of the strappers suggested a massage for a horse. Dr Sorebones had never given a horse a massage before but he was willing to give it a go! When he started working on the horse, it looked around with an expression of "What the dickens are you up to?" After a few minutes the mood of the horse was visibly changed and I am reliably advised that if the horse could have spoken it would have been saying "oohh, aahh, give me more!"

St Albans (NSW) is quite an isolated little rural community which was further born out by the fact that the area is served by a weekly 'Fruit and Vegetables' truck. In the city and urban life that most of us lead these types of services are long gone and either forgotten or just never known of by our younger generation. The fruit and vegetable man and his wife told me that they greatly enjoyed providing their service to the community and of course what did not sell they ate!!

One memorable announcement on the PA came from Halifax Hayes (who always was wearing shorts and thongs even in the frosty mornings). "The weather forecast for

overnight is freezing and in the morning minus 4 degrees. The firewood collection crew is about to leave and needs more helpers." Most days at least one truck would go out with chain saws to collect fallen trees for firewood. The open camp fires were a welcome relief in the cool of the evening and a number of fires were lit in 44 gallon fire drums. It is a good fire when the sides of the fire drum are glowing cherry red!

If you are ever up in the Wisemans Ferry area, with some time to spare, take either ferry from the Sydney side and 20km of steady drive up the Forgotten Valley. There is 10km unsealed on both roads, cattle grids in Settlers Road and just when you start to think that the road leads nowhere the Settlers Arms comes into view. Both roads on the other side of St Albans are unsealed as well. The Branch Road (over the big bridge from the Settlers Arms) is a no through road which winds through numerous farms. About 8km from St Albans on the Branch Road there is a low level wooden bridge, definitely worth seeing. The bridge foundations were apparently seriously affected in a flood resulting in the bridge being rather deformed. The bridge is bent like a banana and the decking is on an angle of at least 15 degrees. Understandably the bridge has a 2 tonne load limit, (my old Landcruiser weighs 1.9 tonnes, we took this one very slowly!) The other road out of the town will ultimately bring you out north of Kulnura after meandering through the St Albans Common and passing the quarry at MoGo. This road is all dirt, very narrow in parts and only recommended for the adventurous.

I was asked well before the finish of the event if I would be willing to do it again next year. Even after the onerous chore of packing up I guess I will be back there again next year, a lot better organized and prepared for a whole set of new problems. Is not it funny the sort of things that a computer can get one involved in?

OK that is my story of why the BBS did not get much attention for a couple of months around August. The Endurance Ride was a lot of hard work that used every spare moment and demanded total dedication but also provided material for what I hope has been an informative and interesting article for you to read and may have provided some readers with programming ideas. Now, how about a lot of other people, is it not time you sat down at your own TI99/4A and wrote an article for your club magazine? Just what unusual things do you do with your TI99/4A computer?

continued from page 5

MAIL TO : ALL
MAIL FROM : LARRY

Just some of the new programs that are coming soon from the USA, Tris, Karate Challenge, Mission Destruct, Legends II, Press (about 1st Oct), Zoom Flume, Witch's Brew, Wizard's End Artist Borders IV and V, Artist Fonts #6, #7 and #8, Artist Instances #10, #11 and #12, Disk of Pyrates, Page Pro Fonts #1 and #2, Page Pro Pics #1 to #7, Music Pro, The Animator, Pix Pro and TI-Writer V4.10 from Canada. Will be doing write ups about them as soon as they arrive.

Bye For now Larry

MAIL TO : ALL
MAIL FROM : COMPUTEX

I am wanting to purchase a VHF modulator for my computer. I need VHF specifically because I have only VHF on my TV. If anyone has a spare unit could they please contact me on 80-2854 after 6 pm.

Thanks

MAIL TO : ALL
MAIL FROM : BANANA

I have for sale a Bell 103 modem which runs on a small 9 volt battery. It is in A1 condition and has a cable for the TI99/4A all ready to go for \$30 or offers. Leave a message if interested on the BBS for BANANA. Thank you, Kevin Cox.

continued on page 20

Rambles

by Stephen Shaw, England

Hello and welcome. As ever, my address, the same as the Group's disk library, is:

10 Alstone Road STOCKPORT, Cheshire, SK4 5AH
and I welcome your letters, especially your queries and requests, but if you want a direct reply a stamped addressed envelope is requested (if you are overseas, make that two International Reply Coupons, purchasable at your local post office).

A note for international dissemination. The Chicago group have discovered that they have had real problems in the past year: their postal distributor was ill and mail was not forwarded (resulting in members being struck off the data base); they have changed their membership database (resulting in more members being dropped off), they have changed their BBS phone number; their "hotline" telephone number has been inactive; their Chairman has been very seriously ill; and they are onto their third membership secretary in little over a year, as one became bogged down in his paid work, and the other suffered severe health problems. Their brand new President Don Jones is very sorry and please would anyone offended please drop him a line now as matters are now in hand.

Anyone think we have problems now? Life for any user group is made very much easier the more members actually contribute!

The Chicago group, possibly one of the largest "User" groups for any computer, even have a member in Tianjin, so now we can add China to the list of countries with a working orphan!

The Chicago group have also listed the TI99/4A peripherals which did not make it (not in any quantity anyway!). How many of you have heard of:

1. Modem CARD (Novation Cat 300 baud modem on a card), no Terminal Emulator 2 needed!
2. IEEE488 controller card.
3. VCR Controller.
4. Debugger card (code for which is in DEBUG for Editor Assembler).
5. GROM library peripheral.
6. Hex bus controller (to connect battery operated RS232/PIO, colour printer and wafertape drive).
7. Special PEB interface card for the 99/8.
8. TI's own hard disk controller card.

and of course there were (and still are) a number of cards produced by third parties for all sorts of things. Now perhaps you can understand why the Peripheral Expansion Box has room for 8 cards?

Depending on how you look at the records, the Chicago group has from 600 to 800 members!

Following on from the product announcement in the last issue of TI*MES about the new word processor, Press, the Lima (Ohio) group reporting on a get together organised by the Chicago group, saw a demonstration of an unfinished product. Additional information is that the spelling checker is on three disks. files are not saved in the usual Text style of DV80 format, but utilities are provided to convert between Press and TI-Writer formats.

From the Hunter Valley in Australia, Tony McGovern advises of a "fundamental flaw" in the ROS for the HRD (=RAM operating system and Horizon RAMdisk respectively!). No problem for you if you have all TI peripherals and stick to them. Problems if you have the Myarc disk controller, the Dijit AVPC card or an Australian made RAMdisk. Tony's comments refer to Version 7.3 ROS and earlier, and he has advised both interested parties of the difficulty.

Liz Lawrence, age 11, has three favorite games: TI Runner, Jumpy, and Flooraway. One of these was written in the UK. Two of these are in machine code, the UK program is in pure and simple TI BASIC. Do not give up on BASIC too fast! Liz has scored 34500 on Jumpy.

From earlier days:-

```
100 CALL INIT
110 CALL LOAD(-31879,0)
--
200 CALL PEEK(-31879,A)
210 PRINT "TIME=";A/50;"SECONDS"
220 GOTO 220
```

Requires Mini Memory or Editor Assembler plus 32K RAM or Extended BASIC plus 32K RAM. You can put code between lines 110 and 200, and time how long that code takes to process. Maximum time is about 5 seconds then it goes back to zero.

Got a DF80 machine code program that does not autostart, and need to know the name to make the !*#\$\$ program start? Well, the easy thing to do is load Funnelweb and load the file with that. Funnelweb then gives you a list of DEF names, and it is usually the first you choose. The harder way is:

Using Editor Assembler or Mini Memory, using BASIC load the DF80 file by:

```
CALL INIT
CALL LOAD("DSK1.FILENAME")
(remember you may need more than one file to make the program work!).
Now, type:
```

```
CALL PEEK(16176,A,B,C,D,E,F)
PRINT CHR$(A);CHR$(B);CHR$(C);CHR$(D);CHR$(E);CHR$(F)
```

and there is the start name to use.

[From an anonymous bulletin board, dated 8th June 1987]

Extra Terrestrial Mail. Just before Christmas I received a newsletter from the Chicago group. Not too odd you may think. Well, it was the April 30th issue.

What do you mean you do not subscribe to MICROpendium department: and look what you are missing.

In the October 88 issue (I keep saying, I get mine by seamail, OK!):

Articles on: BASIC (Using Trace); c99(arrays); Configuring Funnelweb; The Geneve; Review of Bunyard Hardware Manual (recommended); and several mini reviews and lots of advertisements .

In the November 88 issue:
Machine code, amending a CHARA1 file easily; c99(using filenames); Geneve, Review of TI Base Version 2.0, micro reviews and lots of advertisements; news that Navarone lost all interest in their TI99/4A products as of December 1987 and will not consider any support for them (even if you bought them November 30th 1987!).

New Users:

Cassette troubles:-

- a. The remote control (the small 2.5mm plug) works on about half the worlds recorders. The other half use a different polarity, which matters as the switching is electronic, not mechanical. If you can switch the wires in the socket around it should work.
- b. Because electronic switching is used, there may be a voltage drop across the electronic device used, and if you are using batteries and the voltage is low, the drop may be enough to prevent proper operation. Mains power is highly recommended but it may still (rarely!) be necessary to leave the remote plug out.

+++++ continued on page 4

Extended BASIC Tutorial

by Tony McGovern, Funnelweb Farm

Our next example will be a good start on a non-trivial utility program for printing out TI BASIC or Extended BASIC listings on a 80 column printer in two side by side columns which preserve the normal screen listing format. If you just LIST "RS232.BA=...." then the computer sends it out in DIS/VAR 80 format and it is up to you to tell the printer how to handle it. Something approaching screen image format is only obtained (with extra paper consumption) with the printer margins set way in. 80 column print out beats none at all by miles but let us try to be fancier. If you do not have disk or printer then this lesson will not be of immediate use, but will still be a good example to work through as a programming exercise. We might as well do something useful.

First we figure out what needs to be done, and work out a set of procedures that can be CALLED as needed. The program will do only the minimum necessary to do the job properly. Bells and whistles can be added later. In one or two places we shall make provision for adding extras (bells and whistles have nothing on speech) by dummy sub-programs which can be filled in later. For a good discussion of the use of such "stubs" see the excellent book by R. Mateosian, "Inside BASIC Games". The detailed coding examples in this book are in Apple or Trash-80 BASICs (does anyone remember any more what a TRS-80 was?), but Mateosian develops ideas in a form much more in tune with a TI Extended BASIC sub-program realization than with these less capable BASICs.

So let us start designing our program by deciding what we want it to do. We want the output nicely formatted on the page with top and bottom margins, in 2 columns each in screen image (28 characters per line) format. More columns (assuming the output device will handle them) are no problem: once you can count to 2 then 3 is easy. Lines of BASIC are not to be split from one column to the next or from one page to the next. Some things commonly encountered in printed listings, such as indenting of FOR-NEXT loops do not fit at all well with the multi-statement lines of Extended BASIC (but might with TI BASIC listings) so will not even be thought about here. On the other hand insertion of spaces before REM or SUB statements greatly improves the readability of Extended BASIC listings, without doing violence to the idea of being screen list compatible. Page numbering is no big deal to add (a console only Extended BASIC program can fill 6 pages).

At the other end of the business the LISTING to be printed is assumed taken from a disk file such as DSK1.LIST where it has been written by LIST "DSK1.LIST". A trivial difficulty easily taken care of is the blank first record written by LIST. The real problem is that LIST does not care about preserving Extended BASIC lines as distinct entities. Each Extended BASIC line starts out as a separate print record and if it is less than 80 characters long stays in one piece. Extended BASIC lines can easily extend into 2 print records and more (BASIC lines much less frequently), but LIST places no markers to show which print records contain the start of Extended BASIC lines. So if we are going to meet our specification that Extended BASIC lines be treated exactly as in a screen list then something more subtle than a simple LINPUT is needed. There is one of our most important building blocks identified: SUB BASICLINE(...).

Any utility program needs title and advice screens so there is SUB TITLES to keep all the details from cluttering the main program. The program will also need SUB OPTIONS(...) to handle file and device name entry and print options which might be offered.

Now the real core of the program is the way in which it must assemble a whole page before printing

anything because line feed moves ever on (at least it does on our TI99 printer). So we need SUB PAGEBUFFER(...) to take the output of BASICLINES, chop it into screen format hunks and decide where these are to be located on the page. Then we need SUB PRINTPAGE(...) to massage the completed pages and ship them off to the printer. That about sums up the sub-programs that are called directly from the main program, and all that is necessary is to figure out the initialization: DIMs, default filenames etc etc, and to write the logic for program flow.

Before we start writing any code we should decide what utility sub-programs are to be used by those already defined. As the list is written into columns SUB WRITECOL(...) is a good candidate for repeated use, and SUB WRITEPAR(...) to take a line of BASIC and return it chopped up into 28 character lines to WRITECOL. Since BASICLINE fetches the input records it is the appropriate place to detect End Of File. We might as well use PRINTPAGE to wipe the slate clean before writing a new page.

Let us dress up the input of filenames and Yes/No responses a little as SUB FILENAME(...) and SUB YN(...), with SUB MORE(...) to end it all. Other useful utility sub-programs which will be included are SUB TXTCOL(..) to change display colours in one CALL, SUB KEYCON to carry the burden of "press any key to continue", and SUB DELAY(..) is always handy.

That about finishes the roster of procedures necessary to make up the listing program, and now the detailed coding can start after some thought on the necessary chains of parameter passing. The principle that you should plan your programs from the top down and code them from the bottom up is just as valid in Extended BASIC as it is in TI-LOGO or TI-Forth where the form of the language makes it difficult to do otherwise. Sub-programs make it possible to go the same way in Extended BASIC with ease. Less capable dialects of BASIC make it a lot harder to keep your thoughts organized and your code on the rails.

The actual program will now be listed piece by piece and commented on in detail. The listing has been transferred into this TI-Writer file from a working copy of the program using a more elaborate version. The present program is actually a simplified version of the one originally written, but is powerful enough to do a useful job. (I have changed the format to make it easier to read I hope. GWT)

```
100 REM ** SIMPLIST **
110 REM * PRINTER LIST *
120 REM ** FROM DISK **
130 REM -FUNNELWEB FARM-
140 OPTION BASE 1 :: DIM PRLN$(66,2)

150 REM * DEFAULT VALUES *
160 CALL TITLES :: SFIL$="DSK1.LIST" ::
    PDEV$="RS232.BA=4800"
170 CALL KEYCON
```

The first part of the main program shown here sets default values and DIMensions the string array PRLN\$ for two columns of 66 lines each. The top and bottom few lines will be left blank so that page format is obtained without sending printer control codes. A 66 line per page, 80 column printer is assumed.

```
180 REM * New File Entry *
190 CALL OPTIONS(SFIL$,PDEV$):: ENDFILE=0 ::
    LINPUT #1:NEW$

200 REM * New Page Entry *
210 CALL PAGEBUFFER(PRLN$(,),ENDFILE)
220 CALL PRINTPAGE(PRLN$(,),PDEV$):: IF ENDFILE=0
    THEN 210

230 REM * End OR Next *
240 CLOSE #1 :: CLOSE #2 :: CALL MORE(NM):: IF NM THEN
    CALL SPEAK("GOODBYE"):: GOTO 250 ELSE 190
250 STOP
```

OPTIONS returns file and device names as entered there, and the remainder of line 190 resets the End of File flag, and throws away the first line of the list-file. At new page entry the page buffer is filled and then printed out repeatedly until it runs out of listing, and then it asks if you are finished. that is all there is to the main program folks. And now to the sub-programs that do all the work.

```
260 SUB TITLES
270 CALL CLEAR :: CALL SCREEN(11)::
  DISPLAY AT(12,6)BEEP:"PRINTER LISTING"
280 SUBEND

290 SUB OPTIONS(S$,P$):: DISPLAY ERASE ALL ::
  CALL TXTCOL(16,5)
300 CALL FILENAME(1,2,"Edit as needed and ENTER","N?")
310 CALL FILENAME(4,4,"Source file for listing",S$)
320 CALL FILENAME(8,4,"Printer devicename",P$)
330 CALL YN(" Change mind ?","N",22,5,I):: IF NOT(I)
  THEN CALL HCHAR(22,1,32,64):: GOTO 300
340 DISPLAY ERASE ALL :: IF S$="" OR P$="" THEN
  DISPLAY AT(1,2)BEEP:"NO INPUT/OUTPUT POSSIBLE" ::
  CALL DELAY(500):: GOTO 300
350 OPEN #1:S$,DISPLAY ,INPUT ,VARIABLE 80 ::
  OPEN #2:P$,DISPLAY ,OUTPUT,VARIABLE 80
360 SUBEND
```

TITLES here is little more than the barest stub, but you can fill that out to your own fancy. OPTIONS takes down the file names, does some checking, and opens the files.

```
370 SUB PAGEBUFFER(PRLN$(,),EFL)

380 REM * New Col Entry *
390 PLN=6 :: COL=COL+1 :: IF COL>2 THEN COL=0 :: SUBEXIT
  ELSE PRINT "":"** Reading column #";COL:"": ""

400 REM * New Para Input *
410 IF EFL THEN PRINT "":" "***** END of FILE *****":
  "":"" :: SUBEXIT ELSE CALL BASICLINE(NEW$,EFL)::
  PRINT NEW$: ""
420 CALL WRITECOL(PLN,COL,PRLN$(,),NEW$)
430 IF NEW$="END of COL" THEN 390 ELSE 410
440 SUBEND
```

The new column entry in PAGEBUFFER resets the line counter PLN to top of page with a margin, increments the column count, and exits back to the main program if the page is full. If not it tells BASICLINE to fetch a new program line and WRITECOL to enter it in the page buffer. If BASICLINE says it has read the last line it exits and lets the main program worry about that, otherwise it gets another BASIC line or starts a new column. A stub here, CALL SKIPLINE(NEW\$,SK), could have uses.

```
450 SUB BASICLINE(N$,E)
460 N$="" :: IF NX$="" THEN LINPUT #1:NX$
470 N$=N$&NX$ :: IF LEN(NX$)<80 OR EOF(1)THEN NX$="" ::
  E=EOF(1):: SUBEXIT ELSE LINPUT #1:NX$
480 PX=POS(NX$," ",1):: IF PX<2 OR PX>6 THEN 470
490 P=POS(N$," ",1):: IF PX<P THEN 470
500 NR=-1 :: FOR I=1 TO PX-1 :: C=ASC(SEG$(NX$,I,1)::
  NR=NR AND C>47 AND C<58 :: NEXT I :: IF NOT(NR)THEN
  470
510 IF SEG$(N$,LEN(N$),1)=" " THEN 470
520 IF VAL(SEG$(NX$,1,PX-1))<VAL(SEG$(N$,1,P-1))
  THEN 470

530 REM ** Check Quotes
540 N Q,I=0
550 I=POS(N$,CHR$(34),I+1):: IF I THEN NQ=NQ+1 ::
  GOTO 550 ELSE IF NQ<>2*INT(NQ/2)THEN 470
560 SUBEND
```

The procedure BASICLINE which retrieves complete lines of BASIC code from the LIST file is the only part of the program with decision flow complex enough to warrant planning out separately on paper beforehand. I am not going to reproduce this here, but you can work out your own and see if it leads to similar code. The problem comes when the procedure has read in a line exactly 80 characters long. Does the next LIST record

then represent a continuation of the same line of BASIC or is it the start of a new BASIC line? This difficulty cannot be ignored if screen list format is to be preserved since 28 into 80 does not go exactly. The procedure provides a cascade of tests each of which checks whether the record being scrutinized should be appended as a continuation of the previous BASIC line. A few more rare cases could be tested for along the same lines. There is at least one (that I know of) unlikely case which BASICLINE cannot resolve even in principle. Can you spot it? It does seem to work well already though. The intricate input code is needed since a VARIABLE file can only be read sequentially, and if the battery of tests says that the last record LINPUTted does start a new BASIC line, then this must be saved until BASICLINE is called the next time.

Just be thankful for static variables in Extended BASIC sub-programs! You also have to take care not to set off the End of File alarm prematurely.

```
570 SUB WRITECOL(P,C,P$(,),N$):: IF NC THEN P=6 :: NC=0
580 IF P>=57 THEN N$="END of COL" :: NC=-1 :: SUBEXIT
590 CALL WRITEPAR(P,C,P$(,),N$)
600 SUBEND
```

Now that WRITECOL has the line of BASIC it sends it off to be formed into a paragraph. This simplified program handles coming to the end of a column in a slightly wasteful way that is very simple to program. A normal Extended BASIC program line lists at most on 5 screen lines, and no matter how tricky you are in entering longer lines the program has already limited it to a string variable (maximum length 255 or 10 screen lines) or has crashed with an error. The simple minded solution is to exit with End of Col message if the proposed starting line for the new paragraph is past a fixed place somewhat short of the end of the column. The value entered, line #57, is a compromise between making the program totally bullet proof or wasting space. A better approach is to print as far as possible, testing each new paragraph to see if it fits, and if not, holding it over for the next column. If you wondered why the string was called NEW\$, then spare a thought for OLD\$ which vanished without trace during program simplification for tutorial purposes.

```
610 SUB WRITEPAR(P,C,P$(,),N$)
620 P=P+1 :: IF LEN(N$)>28 THEN P$(P,C)=SEG$(N$,1,28)::
  N$=SEG$(N$,29,LEN(N$)-28):: GOTO 620 ELSE P$(P,C)=N$
  :: N$=""
630 SUBEND
```

Sub-program WRITEPAR almost was called SALAMI as it slices up NEW\$ and assigns the slices to successive print lines. Once entered line 620 loops on itself recursively until the remaining piece fits on a screen line. It assumes range checking has been done before entry. In retrospect, 5 years later, that still looks a fine line of code.

```
640 SUB PRINTPAGE(P$(,),D$):: PRINT "":"** Page print
  started"
650 PRINT "":"** Assembling print lines":" and printing
  to" :: PRINT "":" " ;D$
660 FOR I=1 TO 66 ::
  PRINT #2:TAB(9);P$(I,1);TAB(45);P$(I,2)::
  P$(I,1),P$(I,2)= "" :: NEXT I
670 SUBEND
```

Not much needs be said about PRINTPAGE beyond noting that line 660 formats a single print record from the two column entries and erases the page buffer as it goes.

```
680 SUB YN(A$,B$,R,C,X)
690 DISPLAY AT(R,C)BEEP:A$&" (Y/N) "&B$ ::
  ACCEPT AT(R,C+LEN(A$)+7)VALIDATE("YN")SIZE(-1)BEEP:
  A$ :: X=A$=B$ :: R=R+2 :: SUBEND
```

```
700 SUB KEYCON :: DISPLAY AT(24,6)BEEP:"ANY KEY TO
  PROCEED"
710 CALL KEY(3,I,ST):: IF ST=0 THEN 710 ELSE DISPLAY
  ERASE ALL
720 SUBEND
```

continued on page 35

Tigercub Software Catalog #4

by Jim Peterson, Tigercub Software, USA

NUTS & BOLTS DISK No. 2 \$15.00 postpaid

Another 108 utility subprograms in MERGE format. Contents include:

- 20 character fonts and related routines including giant, enlarged, double-height, double-width, script, sideways and underlined characters, etc.
 - 21 screen display routines including horizontal and vertical scrolling, centering, titling, etc.
 - 3 joystick routines for one or two joysticks.
 - 13 math routines including every conversion between binary, hexadecimal, decimal, and more.
 - 6 very unusual graph routines, one for printer.
 - 3 self-changing routines to permit use of a variable in a GOSUB, GOTO or RESTORE.
 - 1 speech routine and 2 sound effect routines.
 - 4 word processing subprograms including screen formatting, plural endings, replacing strings.
 - 5 utilities including the INIT check, instant colour changes, resets, reading memory size.
 - 10 programming utilities to edit and save screens, print screens, call disk catalog, etc.
- Also 4 file handling, 2 menu routines, 6 sorting routines for 2-dimensional arrays, etc., etc.
- With 10 pages of documentation, containing an example of the use of each subprogram.

NUTS & BOLTS DISK No. 3 \$15.00 postpaid

Another 140 utility subprograms, in 120 files, in MERGE format. Contents include:

- 19 screen character fonts and related routines including HALFCHAR at 56 to 64 characters per line, stylized FANCYFONT, lopsided FUNNYFONT, and SPOOKY, HEBREW, JAPANESE, DOUBLE HIGH, OLD ENGLISH, OLDSTYLE, and the greatly enlarged BIGLOW, SUPERCHAR, SKINNY and THINLINE.
- 17 screen display routines including BACKDROP, BACKFORTH, BURST and SLURP, a 32-column DISPLAY AT, DIAGBAN for a diagonal banner and SLIDE to slide it off the screen, WALKING text, UPSCROLL and even a WINDOW routine.
- 6 screen formatting routines to display text without wraparound, fill and adjust, right justify, reduce proportionately, delete columns, etc.
- 8 plotting routines to create circles of sprites, high resolution circles, lines and boxes, etc.
- 6 joystick and keyboard routines, a joystick mouse, etc.
- 32 math routines including decimal to fraction, markdown and markup, MOD, MEDIAN and VARIANCE, acceptance within tolerance, 2s complement, etc., etc.
- 4 time and date routines for Julian to calendar conversion, a sprite timer and a timing bar.
- 9 string handling routines to convert adjectives to adverbs, verbs to past tense or add "ing", simulate the LEFT\$, RIGHT\$ and MID\$ of other Basics, find the last occurrence of a substring, simulate MAX and MIN with strings, etc.
- 15 file handling routines including fast binary searches, a versatile keysearch, routines to handle tabbed fields, etc., etc.
- 10 input and accept routines, and 9 miscellaneous routines including the extremely useful BXB to run BASIC programs in Extended BASIC or use 17 character sets in Extended BASIC.

With 11 pages of documentation giving an example of the use of each subprogram.

DISK COLLECTIONS. These are full disks containing 5 or 6 of my programs, all of one category (no substitutions, please!). Now for only \$5 postpaid! And the remaining disk space has been filled with public domain programs of the same category, selected from my Public Domain

library of over 3000 programs. I am not selling public domain programs, I am just including them as a free bonus to fill up the disk. Therefore I take no responsibility for them, and their titles are subject to change. Some of them may also be on TIPS disks.

C1. TIGERCUB'S BEST

TCX-1001 ALLEY CRAPS, TCX-1039 WHITEWATER RUN, TCX-1058 SCRUM, TCX-1120 HAUNTED GRAVEYARD and TCX-1129 MECHANICAL APTITUDE TEST, plus Public Domain programs BOMBER, FOUR IN A ROW, GAME OF CAT AND MOUSE, IMLOST, MR. KROAKERS, LEAPER, LEFT/RIGHT, MAZZO and MOTORCYCLE JUMP.

C2. PROGRAMMING TUTOR

TC-83 HANDY DANDIES, TC-107 HANDY DANDIES #2, TC-113 HANDY DANDIES #3, TCX-1115 COLOUR PROGRAMMING TUTOR, TCX-1134 MUSIC PROGRAMMING TUTOR and only enough room for SPRITE EDITOR.

C3. PROGRAMMER'S UTILITIES

TC-84 RANDOM CHARACTER GENERATOR, TCX-1118 4x4-BIT GRAPHIC UTILITY, TCX-1122 TIGERCUB GRAPHICS DESIGNER, TCX-1126 4x4-BIT DOODLER, TCX-1137 SOUNDMAKER, TCM-2001 SORT TESTER and Public Domain CATWRITER, COMPACTOR, COMPARE, CROSSREF, DISKLISTER, DISKMEM LISTER, DOWNLOAD CONVERTER, EXTRACTOR, GO-SEARCH, LINE EDIT, SINGLE PIXEL DRAWING, QUICKLOAD, SHRINK, EDITOR.

C4. BRAIN GAMES

TCX-1013 PICK UP STICKS, TCX-1086 VEGA, TC-136 THREE COIN WEIGHING PUZZLES, TCX-1004 REVERSO, TCX-1005 BASSACKWARDS, TCX-1102 MATCH A PATCH and Public Domain programs ACME HOTEL, CHECKERS, DELIVER THE CAKE, L-GAME, MASTERMIND, OTHELLO, RAINBOW SQUARES.

C5. BRAIN TEASERS

TC-46 MISSIONARIES CANNIBALS, TC-75 ELECTION, TC-89 FOUR BY FOUR PUZZLE, TC-97 TOWERS OF HANOI, TCX-1088 THREE BUCKETS PUZZLE, TCX-1127 OLD-TIMER PUZZLE and Public Domain puzzlers 15 PUZZLE, HEXAPAWN, INDIANS, PREACHER LAWYER SALESMAN, THE LAST ROBOT, MOUSE IN A MAZE, QUEEN BOARD GAME, ROTATE and SHOOT THE STARS.

C6. BRAIN BUSTERS!

TC-79 CAN OF WORMS, TCX-1038 RITHMATIK, TCX-1041 LONG DIVISION CRYPTOGRAMS, TCX-1060 NIMBO, TCX-1081 GLUNK, plus Public Domain puzzlers 100%, ADDITION MAGICIAN, ARITHMAGRAPHS, BAGELS, DIGITRON, FOUR IN A ROW, GOING HOME, GOMOKU, IQ MATH, TIGERCUB MATH PUZZLE, MAWARI, MULTIPLICATION MADNESS, ONE CHECK, ONE TO FIVE, OTHELLO and SPHINX.

C7. MANEUVERING GAMES

TC-117 CAROM, TCX-1022 PLANETARY DEFENSE, TCX-1104 GETAWAY, TCX-1105 SHEEP DOG, TCX-1133 WHITE KNIGHT, TCX-1135 BARS & BALLS and Public Domain games AHAMUR, ARTILLERY BATTLE, BOWLING, ROMEO & JULIET, SKY RESCUE, COVER-UP, TI-TROGMAN, VERSAILLES and ZAN QUEST.

C8. ACTION GAMES

TC-37 PILLBOX TANKS, TC-65 MICE IN A MAZE, TC-94 FIREFLIES, TC-101 ROOKIE OUTFIELDER QUICK DRAW, TCX-1034 RUNAWAY PIG, TCX-1093 GLEEP SHOOT ZOOK SHOT and Public Domain games ANTARES, AUSSIE FIGHTER, BALLOONS, BEETLE WALK, BRIDGE GUARD, CHICKEN, INDY 500, INVASION, LILY PADDER and PATSCRAM MISSION.

C9. REFLEX AND CONCENTRATION

TC-25 NERVOUS BREAKDOWN, TCX-1036 SIMON SAYS, TC-112 BINGLE, TCX-1017 JELLY BEANS, TCX-1026 MEMORY SHARPENER, TCX-1027 QUICK 'N THINK, plus Public Domain AIR DEFENSE, AIR TRAFFIC CONTROLLER, AIR RESCUE, ARROW ZAP, ATTACKER, BAT ATTACK, BONKERS, ADVENTURE WITH DRACULA and ECG WARS.

C10. TWO-PLAYER GAMES

TC-51 SNERK, TCX-1054 PLAIN OF JEWELS, TC-55 GOLDEN SQUARES, TCX-1053 DRY GULCH, TCX-1059 MIDNIGHT TRAIL, TCX-1108 RANCH WAR, plus Public Domain GAME OF ONE OR TWO, FOUR IN A ROW, BLOCK 'EM, CANYON BOMBER, GOLF, HOME RUNS, ISOLATION and JCL SQUARES.

C11. KID'S GAMES

TC-32 BUTTERFLY FLOWERS, TCX-1033 POT OF GOLD, TC-45 TIRKLE, TC-91 SLINKY, TCX-1012 KID STUFF and TCX-1040 WAWALAND, plus AUTOMATIC MOUSE MAZE, KWIK DRAW, MARKSMAN, MISS MUFFET, PI-RING SQUAD, RIBBIT, SCHOOL DAZE, SHAPE ART and TURTLE HOP.

C12. MORE GAMES

TC-18 LOST PLANE, TC-23 ANDROMEDAN INVASION, TC-57 BUNNY BEAGLE, TC-77 COLUMBUS PINBALL, TCX-1014 DRUNKEN SAILOR, TCX-1035 SUBMARINE HUNT, plus COMPU-GOLF, CRAZYCLIFF, DIAMOND DROP, SUPER TI DOGFIGHT, HOME BOUND, INSANITY, KNOCK OUT, TIGERCUB SPRITE CHASE, WALLBREAKER and WORMS SPIDERS.

C13. WORD GAMES

TC-20 SCRAMBLE, TC-125 HIEROGLYPHY, TCX-1044 SCRAMBULATION, TCX-1050 SQUINCH, TCX-1069 BAZOO and TCX-1076 CHANGEROO, plus ANAGRAMMER, WORLD CURRENCIES, NATURAL ELEMENTS, NATIONS OF EUROPE AFRICA, FALCO'S HANGMAN, MILLER'S HANGMAN, WORLD MYTHOLOGY, WORD GAME and WORD SCRAMBLE.

C14. ELEMENTARY MATH

TC-31 KINDERMATH, TCX-1063 KINDERDIVVY, TC-103 SIXBONES, TCX-1062 KINDERTIMES and TCX-1110 KINDERMINUS, plus ADD & CARRY, BASIC ADDITION & SUBTRACTION, ADD-ED, BE A CLOWN, BE FRUITFUL AND MULTIPLY, ELEMENTARY MATH, MATH PRACTICE, COLOUR MATH, MULTIPLICATION PRACTICE, TAKE AWAY.

C15. MIDDLE/HIGH SCHOOL MATH

TC-82 FRACTION MATH, TC-130 MATH HOMEWORK HELPER, TCX-1015 MENTAL MATH PRACTICE, TCX-1016 FAST ADDITION PRACTICE, TCX-1098 ROMAN NUMBERS, TCX-1119 CASTING OUT NINES and Public Domain programs MATH BASEBALL, CATAPULT CAPERS, DECIMAL TO FRACTION, SCIENCE FRICTION, COORDINATE GEOMETRY, LONG DIVISION, MAGIC NINES, MATH QUIZ, RUSSIAN MULTIPLICATION.

C16. VOCABULARY AND READING

TC-19 HANGMAN, TCX-1073 SYNONYMY, TCX-1074 SPEEDER READER, TCX-1123 HOMONYMY, TCX-1124 ANTONYMY, TCX-1131 JUNIOR SPEEDER READER and Public Domain programs ADJECTIVE TO ADVERB, LEARNING TO ING IT, ANIMAL MULTITUDES, NATIONS OF AMERICA, NOUN TO ADJECTIVE, PLURAL ENDINGS, TENSE TIME.

C17. MUSICAL EDUCATION

TC-49 TUNEPICKER, TCX-1010 TONE TEST, TCX-1011 MUSICAL NOTE RECOGNITION, TCX-1056 NAME THAT TUNE, TCX-1072 MUSICAL SCALES, TCX-1109 TIGERCUB KEYBOARD ORGAN and excellent Public Domain programs INTERVALS, RHYTHM, MUSIC MAKER, STAFF and INVENTION IN F.

C18. KALEIDOSCOPIES AND DISPLAYS

TC-67 MILLION MIRAGES, TC-92 KALEIDOVISION, TC-95 JEWELS ON VELVET, TC-99 MULTIVISION, TCX-1128 TEN THOUSAND SIGHTS, plus ANDREW'S SPRITES, AURORA, BOX ART, COLOUR BURST, COLORSQUARES, COLORVISIONS, COLUMBIA THE GEM OF THE OCEAN, ESCHER ART, ETERNITY, FASCINATION, HYPNOSIS PATTERN, KALEIDOSCOPE, KALEIDOVISION, PATCHES, SNOWFALL ON GANYMEDE, QUICK KALEIDOSCOPE and TIGERCUB SPRITE DEMO.

Tigercub Care disks #1, #2, #3 and #4 \$5.00 each, postpaid

These disks were prepared for distribution to user group newsletter editors, in exchange for their newsletters, but are available to anyone else for \$5 per disk, postpaid.

No. 1 contains Tips from the Tigercub numbers 42 through 45 in text format (not ready-to-run programs), plus printer utility programs PRINTALL, 28-COLUMN CONVERTER and HYPHENATOR.

No. 2 and No. 3 are filled with text articles and lessons, mostly on Extended BASIC programming, including the Programs That Write Programs series and the Putting It All Together series. No. 4 contains Tips from the Tigercub numbers 46 through 52 in text format.

TIGERCUB COUNTRY, a full disk of 22 musical selections, many of them with lyrics for a sing-along. This disk is free with any order, if requested, but please include \$1 for my cost of the disk, packing and postage.

continued from page 29

```
: CELL ( row# col# --- addr )
  DUP >R ( duplicate address and save it to the
           return stack )
  @ ( fetch number of columns )
  ROT ( bring row# to top of stack )
  * + ( multiply #col by row# and add to col# )
  R> ( retrieve address from return stack )
  4 + ( address of first member )
  + ; ( add offset )
```

Note that this is NOT a defining word and it is set up to access bytes not 16-bit words. Now we can use SCORES and CELL to enter numbers in the array, just keep in mind that in FORTH we always start the count at zero:

```
75 0 2 SCORES CELL C! ( puts 75 in row 1, column 3 )
90 1 1 SCORES CELL C! ( puts 90 in row 2, column 2 )
```

To check if the values were placed in the correct cells, we enter:

```
0 2 SCORES CELL C@ . 75 ok
1 1 SCORES CELL C@ . 90 ok
```

With the defining word 2D-ARRAY you can set up arrays to you heart's content. But since my purpose here is to explain <BUILDS...DOES> I will not go into any further details about arrays. Instead I will give you a defining word with practical value. Its author is Michal (sic) Jaegerman of Edmonton, Alberta, Canada.

```
: PRINTS ( n1, n2, .. number of n's ---- )
  <BUILDS HERE SWAP DUP C, ALLOT
  HERE 1-
  DO I C! -1 +LOOP
  HERE =CELLS DP !
DOES> SWCH COUNT TYPE UNSWCH ;
```

This should convince you that anything which can be done with a colon definition can also be achieved with a defining word. If you dissect the compile-time portion of PRINTS you find that it stores (compiles) the number of parameters needed into the first byte, then uses that number to store the parameters with a DO-LOOP. The run-time action consists of sending the parameters to the printer. Since all printer commands are essentially alike (one or a sequence of ASCII values are transmitted to the printer) a defining word is a perfect way to create them. For example, to set my printer up for emphasized mode, I need to send ESC E. Thus

```
27 69 2 PRINTS EMPHASIZED
```

creates the word EMPHASIZED which, when invoked, sends the command to the printer. In a like manner, you can create words for all printer commands you use frequently. All that is required are the ASCII values for the command itself, the total number of values, PRINTS and a word which you select (hopefully with some mnemonic value).

I hope the foregoing has provided some help in understanding <BUILDS...DOES>. Keep in mind that there is no need to define a defining word unless you want to create a number of words with similar functions. If that is not the case, stick with colon definitions.



```

4 REM E. LIVINGSTONE SEATTLE
WA 98107
5 CALL CLEAR
10 PRINT "**** STATISTICA
L ****"
15 PRINT
20 PRINT "**** PARAMETERS
****"
25 PRINT
30 PRINT
35 PRINT
40 PRINT
45 PRINT
50 PRINT
55 PRINT
56 PRINT
57 PRINT
58 PRINT "TYPE 1 FOR MENU"
60 INPUT T
65 IF T=1 THEN 70
70 CALL CLEAR
80 PRINT
85 PRINT "**** MENU
****"
90 PRINT "*****"
*****"
100 PRINT
110 PRINT "(1) MEAN, STANDARD
DEVIATION, AND STANDARD ERR
OR FOR GROUPED DATA."
"
111 PRINT
120 PRINT "(2) MOMENTS, SKEWN
ESS, AND KURTOSIS."
"
121 PRINT
130 PRINT "(3) CHI-SQUARE EV
ALUATION."
131 PRINT
140 PRINT "(4) PERMUTATIONS
AND COMBINATIONS."
"
141 PRINT
150 PRINT "(5) NO-REPETITION
PROBABILITY."
151 PRINT
160 PRINT "(6) LINEAR INTERP
OLATION."
161 PRINT
170 INPUT U
180 IF U=1 THEN 1021
190 IF U=2 THEN 2021
200 IF U=3 THEN 3031
210 IF U=4 THEN 4031
220 IF U=5 THEN 5021
230 IF U=6 THEN 6005
240 GOTO 5
250 END
1000 REM **THIS PROGRAM**
**COMPUTES THE MEAN**
**STANDARD DEVIATION**
**AND STANDARD ERROR**
1010 REM **FOR GROUPED DATA*
*
1021 CALL CLEAR
1022 PRINT
1023 PRINT "MEAN, STANDARD DE
VIATION, AND STANDARD ERR
OR FOR GROUPED DATA."
1024 PRINT
1025 PRINT
1026 PRINT
1027 PRINT
1028 PRINT
1029 PRINT "ENTRIES OF VALUE
S AND FRE- QUENCIES ARE MAD
E WITH APPROPRIATE NUMB
ERS"
1030 PRINT "SEPERATED BY A C
OMA (IE., 5,3)"
1031 PRINT

```

```

1032 PRINT "INPUT 0,0 FOR LA
ST VALUE AND FREQUENCY RE
SPECIALLY."
1033 PRINT
1034 PRINT "TYPE 1 TO GO ON.
"
1035 INPUT V
1036 IF V=1 THEN 1039
1039 CALL CLEAR
1040 LET A=0
1045 LET B=0
1047 LET C=0
1050 D=0
1060 CALL CLEAR
1070 PRINT "** ENTER SAMPL
E VALUE **: **: AND FR
EQUENCY **"
1080 INPUT X,F
1090 IF F<=0 THEN 1170
1100 LET A=A+1
1110 LET B=B+F
1120 LET G=X*F
1130 LET C=C+G
1140 LET E=(X^2)*F
1150 LET D=D+E
1160 GOTO 1070
1170 LET H=C/B
1180 LET I=SQR(D-(B*(H^2)))
1190 LET J=I/SQR(B)
1195 CALL CLEAR
1200 PRINT
1210 PRINT
1220 PRINT "** NUMBER OF S
AMPLES **: **: ENTERED
=:A
1230 PRINT
1240 PRINT "** MEAN =":H
1250 PRINT
1260 PRINT "** STANDARD
**":** DEVIATIO
N =":I
1270 PRINT
1280 PRINT "** STANDARD
**":** ERROR ="
;J
1285 PRINT
1286 PRINT
1287 PRINT
1288 PRINT
1289 PRINT
1290 PRINT
1300 PRINT "TYPE 1 TO GO ON,
O TO STOP"
1310 INPUT L
1320 IF L=1 THEN 1340
1330 GOTO 70
1340 PRINT
1350 GOTO 1021
1360 END
2000 REM **THIS PROGRAM COM-
PUTES THE FIRST 4 MOMENTS,
SKEWNESS, AND KURTOSIS**
2021 CALL CLEAR
2022 PRINT "COMPUTES THE FIR
ST FOUR MOMENTS, WHERE T
HE FIRST MOMENT IS THE ME
AN OF THE"
2023 PRINT "DISTRIBUTION, AN
D THE SEC- OND MOMENT IS TH
E VARIANCE."
2024 PRINT
2025 PRINT "SKEWNESS IS THE
DEPARTURE OFA FREQUENCY DIST
RIBUTION FROM SYMMETRY."
2026 PRINT
2027 PRINT "KURTOSIS EXPRESS
ES RELATIVE PEAKEDNESS OF A
FREQUENCY DISTRIBUTION."
2028 PRINT
2029 PRINT
2030 PRINT "TYPE 1 TO GO ON.
"
2031 INPUT V

```

```

2032 IF V=1 THEN 2035
2033 STOP
2035 CALL CLEAR
2036 LET A=0
2040 LET B=0
2050 LET C=0
2060 LET D=0
2070 PRINT "NUMBER OF SAMPLE
S ="
2080 INPUT N
2085 PRINT
2086 PRINT
2090 FOR I=1 TO N
2100 PRINT "SAMPLE ="
2110 INPUT X
2114 PRINT
2120 LET A=A+X
2130 LET B=B+(X^2)
2140 LET C=C+(X^3)
2150 LET D=D+(X^4)
2160 NEXT I
2170 LET J=A/N
2180 LET E=(B/N)-(J^2)
2190 LET F=(C/N)-((3*J*B)/N)
+(2*(J^3))
2200 LET G=(D/N)-((4*J*C)/N)
-(3*(J^4))
2220 LET K=F/(E^(3/2))
2230 LET L=H/(E^2)
2231 PRINT
2232 PRINT
2233 PRINT
2234 PRINT
2240 PRINT "1ST MOMENT =":J
2241 PRINT
2250 PRINT "2ND MOMENT =":E
2251 PRINT
2260 PRINT "3RD MOMENT =":F
2261 PRINT
2270 PRINT "4TH MOMENT =":H
2271 PRINT
2280 PRINT "MOMENT COEFFICIE
NTS:"
2281 PRINT
2285 PRINT "SKEWNESS =":K
2286 PRINT
2290 PRINT "KURTOSIS =":L
2295 PRINT
2300 PRINT "*****"
*****"
2305 PRINT
2310 PRINT "TYPE 1 TO CONTIN
UE, O TO STOP"
2320 INPUT Z
2330 IF Z=1 THEN 2350
2331 PRINT
2332 PRINT
2333 PRINT
2340 GOTO 80
2350 PRINT
2360 GOTO 2021
2370 END
3010 REM **THIS PROGRAM COMP
UTES CHI-SQUARE EVALUATION
ON THE OBSERVED TO EXPECTED
FREQUENCIES**
3031 CALL CLEAR
3032 PRINT "COMPUTES CHI-SQU
ARE EVALUA- TION FROM INPUT
D OBSERVED AND EXPECTED FRE
QUENCIES."
3033 PRINT
3034 PRINT
3035 PRINT "TO TERMINATE THE
EVALUATION, INPUT 0 FOR THE
LAST EXPECT-ED FREQUENCY."
3036 PRINT
3037 PRINT
3038 PRINT
3039 PRINT
3040 PRINT
3041 PRINT

```



```

3042 PRINT
3043 PRINT
3044 PRINT "TYPE 1 TO GO ON"
3045 PRINT
3046 INPUT V
3047 IF V=1 THEN 3049
3048 GOTO 3031
3049 CALL CLEAR
3050 LET J=0
3055 PRINT "OBSERVED FREQUEN
CIES ="
3060 INPUT D
3061 PRINT
3062 PRINT
3070 PRINT "EXPECTED FREQUEN
CIES ="
3080 INPUT E
3081 PRINT
3082 PRINT
3083 PRINT
3090 IF E=0 THEN 3130
3100 LET K=((D-E)^2)/E
3110 LET J=J+K
3120 GOTO 3050
3130 PRINT "CHI-SQUARE =" ;J
3131 PRINT
3140 PRINT "*****"
*****"
3150 PRINT "TYPE 1 TO CONTIN
UE,      0 TO STOP"
3160 INPUT L
3170 IF L=1 THEN 3190
3171 PRINT
3172 PRINT
3173 PRINT
3180 GOTO 80
3190 PRINT
3200 GOTO 3031
3210 END
4010 REM **THIS PROGRAM COMP
UTES PERMUTATIONS AND COMBIN
ATIONS**
4020 REM **WHERE N=NUMBER OF
ITEMS, AND R=SIZE OF GROUP
SELECTED**
4031 CALL CLEAR
4032 PRINT "THIS PROGRAM COM
PUTES COMBI-NATIONS AND PERM
UTATIONS,"
4033 PRINT "WHERE N=NUMBER O
F ITEMS, ANDR=SIZE OF GROUPS
SELECTED."
4034 PRINT
4035 PRINT
4036 PRINT
4037 PRINT
4038 PRINT
4039 PRINT
4040 PRINT
4041 PRINT
4042 PRINT
4049 PRINT "ENTER N,R"
4050 INPUT N,R
4060 IF N<R THEN 4270
4070 IF R<0 THEN 4290
4071 PRINT
4072 PRINT
4080 LET T=N
4090 GOSUB 4310
4100 LET A=T
4110 LET T=N-R
4120 GOSUB 4310
4130 LET B=T
4140 LET P=A/B
4150 LET T=R
4160 GOSUB 4310
4170 LET C=A/(T*B)
4180 PRINT "PERMUTATIONS =" ;
P
4181 PRINT
4182 PRINT
4183 PRINT

```

```

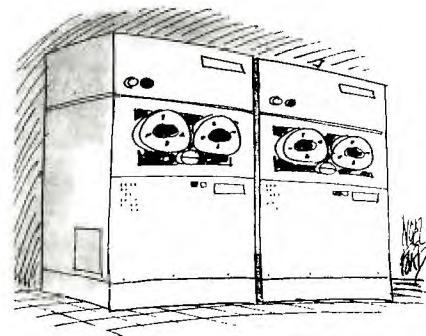
4190 PRINT "COMBINATIONS =" ;
C
4191 PRINT
4192 PRINT
4193 PRINT
4194 PRINT
4195 PRINT
4196 PRINT
4197 PRINT
4200 PRINT
4210 PRINT "TYPE 1 TO CONTIN
UE,      0 TO STOP."
4220 INPUT L
4230 IF L=1 THEN 4250
4231 PRINT
4232 PRINT
4233 PRINT
4240 GOTO 80
4250 PRINT
4260 GOTO 4031
4270 PRINT "N<R, INVALID INP
UT"
4280 GOTO 4032
4290 PRINT "R<0, INVALID INP
UT"
4300 GOTO 4031
4310 IF T=0 THEN 4390
4320 IF T=1 THEN 4390
4330 LET J=1
4340 FOR I=2 TO T
4350 LET J=J*I
4360 NEXT I
4370 LET T=J
4380 GOTO 4400
4390 LET T=1
4400 RETURN
4410 END
5010 REM **THIS PROGRAM COMP
UTES NO-REPITIONS PROBABIL
ITIES WITH GIVEN POPULATION
AND GIVEN SAMPLE SIZE**
5021 CALL CLEAR
5022 PRINT "THIS PROGRAM COM
PUTES      PROBABILITIES WI
TH NO      REPETITIONS, YOU
PROVIDE"
5023 PRINT "POPULATION AND S
AMPLE SIZES."
5024 PRINT
5025 PRINT
5026 PRINT
5027 PRINT "      *****
*****      "
5030 PRINT "      *****
*****      "
5031 PRINT
5032 PRINT
5035 PRINT "TOTAL POPULATION
="
5040 INPUT M
5041 PRINT
5042 PRINT
5050 PRINT "SIZE OF SAMPLE =
"
5060 INPUT N
5070 LET J=1
5080 FOR I=1 TO N-1
5090 LET K=1-(I/M)
5100 LET J=J*K
5110 NEXT I
5115 PRINT
5116 PRINT
5117 PRINT
5120 PRINT "PROBABILITY =" ;J
5130 PRINT
5135 PRINT
5136 PRINT
5137 PRINT
5140 PRINT "TYPE 1 TO CONTIN
UE,      0 TO STOP."
5150 INPUT L
5160 IF L=1 THEN 5030
5161 PRINT

```

```

5162 PRINT
5163 PRINT
5170 GOTO 80
5180 END
6000 REM **THIS PROGRAM COMP
UTES LINEAR INTERPOLATION**
6005 CALL CLEAR
6006 PRINT "LINEAR INTERPOLA
TION:      IF Y IS A FUNCTI
ON OF X,      AND Y1 - Y2 ARE
THE VALUES"
6007 PRINT "OF THE FUNCTION
AT X1 - X2, RESPECTIVELY, Y
MAY BE COMP-UTED FOR ANY VAL
UE OF X."
6008 PRINT
6009 PRINT "SEPERATE VALUES
WITH COMAS."
6010 PRINT
6013 PRINT
6014 PRINT "      *****
*****      "
6015 PRINT
6016 PRINT
6020 PRINT "KNOWN VALUES OF
X,      (X1,X2) ="
6030 INPUT X1,X2
6035 PRINT
6036 PRINT
6040 PRINT "KNOWN VALUES OF
Y,      (F(X1),F(X2)) ="
6050 INPUT Y1,Y2
6055 PRINT
6056 PRINT
6060 PRINT "INTERPOLATE F(X)
AT X ="
6070 INPUT X
6080 LET G=((X2-X)*Y1)+((X-X
1)*Y2)
6090 LET F=G/(X2-X1)
6098 PRINT
6099 PRINT
6100 PRINT "INTERPOLATED F(X
) =" ;F
6110 PRINT
6111 PRINT
6112 PRINT
6120 PRINT "TYPE 1 TO CONTIN
UE,      0 TO STOP."
6130 INPUT L
6140 IF L=1 THEN 6160
6141 PRINT
6142 PRINT
6143 PRINT
6150 GOTO 80
6160 PRINT
6170 GOTO 6005
6180 END

```



'You mark my words — microcomputers will take us all over one day.'


```

160 T=1 :: DISPLAY AT(20,1):"Choose - ":" (1) Manual":
(2) Automatic": (3) Retrace": (4) Load"
170 ACCEPT AT(20,1)VALIDATE("1234")SIZE(1)BEEP:Q :: ON
Q GOTO 180,230,290,400
180 GOSUB 430
190 CALL KEY(3,K,ST):: IF ST=0 THEN 190 ELSE
D=POS("EDXSQP",CHR$(K),1)+1 :: ON D GOTO
190,210,210,210,210,200,360
200 CALL CLEAR :: GOTO 160
210 D=D-1 :: IF ABS(D-D2)=2 OR R+(D=1)=0 OR R-(D=3)=25
OR C+(D=4)=2 OR C-(D=2)=31 THEN 190 :: GOSUB 490 ::
IF D<>D2 THEN GOSUB 440
220 GOSUB 480 :: GOSUB 500 :: GOTO 190
230 GOSUB 430 :: RANDOMIZE
240 D=D+1+(D=4)*4 :: CALL KEY(0,K,ST):: IF ST=0 THEN 260
250 IF K=80 THEN 360 ELSE IF K=81 THEN CALL CLEAR ::
GOTO 160
260 T=INT(4*RND+2)*2-INT(2*RND):: FOR J=1 TO T :: IF
D<>D2 THEN GOSUB 440
270 GOSUB 480 :: CH=128-(D=1)-(D=3):: CALL
GCHAR(R,C,G):: IF G<>32 THEN IF INT(2*RND+1)<>1 THEN
CH=G
280 GOSUB 510 :: NEXT J :: GOTO 240
290 IF LEN(K$(1))=0 THEN DISPLAY AT(24,1):"CAN'T DO
THAT" :: GOTO 170
300 CALL CLEAR :: GOSUB 430 :: FOR J=1 TO T :: FOR JJ=1
TO LEN(K$(T)):: D=POS("EDXS",SEG$(K$(T),JJ,1),1)
310 IF D=0 THEN 350 :: IF D<>D2 THEN GOSUB 440
320 GOSUB 480 :: CH=128-(D=1)-(D=3):: CALL
GCHAR(R,C,G):: IF G=32 THEN GOSUB 510 :: GOTO 350
330 K=ASC(SEG$(K$(T),JJ+1,1)):: IF K<>67 AND K<>79 AND
K<>85 THEN JJ=JJ+1 :: GOTO 330
340 GOSUB 470 :: GOSUB 510
350 NEXT JJ :: NEXT J :: GOTO 170
360 IF LEN(K$(1))>0 THEN 370 :: DISPLAY AT(12,1)ERASE
ALL:"CAN'T DO THAT!" :: GOTO 160
370 DISPLAY AT(12,1)ERASE ALL:"Save to - ":"
(1)Cassette": (2)Disk" :: ACCEPT
AT(12,1)VALIDATE("12")SIZE(1):S :: IF S=1 THEN OPEN
#1:"CS1",INTERNAL,OUTPUT,FIXED 192 :: GOTO 390
380 DISPLAY AT(16,1):"Filename DSK" :: ACCEPT
AT(16,13):F$ :: OPEN #1:"DSK"&F$,INTERNAL,FIXED
192,OUTPUT
390 PRINT #1:T :: FOR J=1 TO T :: PRINT #1:K$(J)::
K$(J)=" " :: NEXT J :: CLOSE #1 :: GOTO 160
400 DISPLAY AT(12,1)ERASE ALL:"Load from - ":"
(1)Cassette": (2)Disk" :: ACCEPT
AT(12,13)VALIDATE("12")SIZE(1)BEEP:L :: IF L=1 THEN
OPEN #1:"CS1",INTERNAL,FIXED 192,INPUT :: GOTO 420
410 DISPLAY AT(16,1):"Filename? DSK" :: ACCEPT
AT(16,14)BEEP:F$ :: OPEN #1:"DSK"&F$,INTERNAL,FIXED
192,INPUT
420 INPUT #1:T :: FOR J=1 TO T :: INPUT #1:K$(J):: NEXT
J :: CLOSE #1 :: GOTO 300
430 CALL CLEAR :: CALL COLOR(13,5,11):: R,R2=12 ::
C,C2=14 :: D2=3 :: CH=129 :: CALL HCHAR(R2,C2,CH)::
RETURN
440 CH2=128+((D2=1)*(D=2)*3)+((D2=1)*(D=4)*5)+
((D2=3)*(D=2)*2)+((D2=3)*(D=4)*4)+((D2=2)*(D=1)*4)+
((D2=2)*(D=3)*5)
450 CH2=CH2+((D2=4)*(D=1)*2)+((D2=4)*(D=3)*3):: CALL
HCHAR(R2,C2,CH2):: RETURN
460 CALL KEY(3,K,ST):: IF ST=0 THEN 460 ELSE IF
POS("COU",CHR$(K),1)=0 THEN 460
470 GOSUB 490 :: IF K=67 THEN CH=134 :: RETURN ELSE IF
K=85 THEN CH=G :: RETURN ELSE RETURN
480 R=R+(D=1)-(D=3):: C=C+(D=4)-(D=2):: RETURN
490 IF Q<>1 THEN RETURN ELSE K$(T)=K$(T)&CHR$(K):: IF
LEN(K$(T))<193 THEN RETURN ELSE T=T+1 :: RETURN
500 CH=128-(D=1)-(D=3):: CALL GCHAR(R,C,G):: IF G>32
THEN GOSUB 460
510 CALL HCHAR(R,C,CH):: R2=R :: C2=C :: D2=D :: RETURN

```

I think that educational programs should teach, not just test. This one makes up the kind of problems we all hated in school, but if you get the answer wrong it will show you how to work it.

```

100 CALL CLEAR :: RANDOMIZE
110 DATA LUMBERJACK,CUT,CORDS OF WOOD,BOY,PICK,QUARTS OF
BERRIES,ELEPHANT,EAT,BALES OF HAY,COW,GIVE,GALLONS
OF MILK
120 FOR J=1 TO 4 :: FOR L=1 TO 3 :: READ M$(J,L):: NEXT
L :: NEXT J

```

```

130 A=INT(5*RND+2):: IF A=A2 THEN 130 ELSE A2=A
140 B=INT(9*RND+2):: IF B=B2 THEN 140 ELSE B2=B
150 C=INT(9*RND+2):: IF C=C2 THEN 150 ELSE C2=C
155 X=B/C/A :: IF LEN(STR$(X))>4 THEN 130
160 D=INT(4*RND+1):: IF D=D2 THEN 160 ELSE D2=D
170 DISPLAY AT(3,1)ERASE ALL:"IF";A;M$(D,1);"S CAN
";M$(D,2);B;M$(D,3);" IN":C;"DAYS,"
180 DISPLAY AT(6,1):"HOW MANY ";M$(D,3);" CAN 1
";M$(D,1);" ";M$(D,2);" IN 1 DAY?"
190 ACCEPT AT(9,1)VALIDATE(NUMERIC)BEEP:Q
200 IF Q<>X THEN 300 :: DISPLAY AT(11,1):"CORRECT!"
210 DISPLAY AT(23,1):"PRESS ANY KEY" :: CALL
KEY(0,K,ST):: IF ST=0 THEN 210 ELSE 130
300 DISPLAY AT(11,1):"NO -":"IF";A;M$(D,1);"S CAN
";M$(D,2);B;M$(D,3);" IN":C;"DAYS,"
310 DISPLAY AT(15,1):"THEN";A;M$(D,1);"S CAN
";M$(D,2);B;"/";C;M$(D,3);" IN 1
DAY":B;"/";C;"=";B/C
320 DISPLAY AT(19,1):"SO 1 ";M$(D,1);" CAN
";M$(D,2);B/C;"/";A;M$(D,3);" IN 1
DAY":B/C;"/";A;"=";X :: GOTO 210

```

Here is a new way to put a title on the screen:

```

100 !SCATTERPRINT by Jim Peterson
110 CALL CLEAR :: M$="TIGERCUB SOFTWARE" :: L=LEN(M$)::
IF L>28 THEN 110 ::
C$=SEG$( "ABCDEFGHIJKLMNPOQRSTUVWXYZ[\",1,L)
120 FOR J=1 TO L :: RANDOMIZE :: X=INT(LEN(C$)*RND+1)::
Y=ASC(SEG$(C$,X,1))-64
130 DISPLAY AT(2,13-L/2+Y):SEG$(M$,Y,1)::
C$=SEG$(C$,1,X-1)&SEG$(C$,X+1,25):: NEXT J
140 GOTO 140

```

This one is very basic, but if you have Terminal Emulator II, Speech Synthesizer, and a pre-school child, it is a fine way to learn the alphabet, the keyboard, to spell his name, or just to have fun with: try a string of Ks for a train chugging uphill.

```

100 OPEN #1:"SPEECH",OUTPUT
110 CALL KEY(3,K,S)
120 INPUT M$
130 PRINT #1:M$
140 GOTO 120

```

Memory full, Jim Peterson

continued from page 11

```

MAIL TO : ALL
MAIL FROM : LARRY

```

I have just put two programs into BBS users section. EZKUTILS that is one of the missing parts and TITRIS*V2 that is a improved version of TITRIS that is in the DOWNLOAD area of the BBS.

Bye for now LARRY.

```

MAIL TO : ALL
MAIL FROM : BANANA

```

My MiniMemory battery has gone flat. Where is the best place to obtain one? I do believe at one stage the shop was selling them but I have not heard any more about that.

Signed Kevin Cox

I may be contacted on the BBS under BANANA

```

MAIL TO : ALL
MAIL FROM : LARRY

```

I have a very new version of TELCO that has a lot of improvements to it.

```

MAIL TO : ALL
MAIL FROM : SCI-FIBBS

```

Hi all!

Can anyone who has the documentation or information on EZKEYS please send me mail about it? I have figured out quite a bit but would like to know if there is more. Ta.

Regards, Greg.

TI-Base Tutorial #4

by Martin Smoley, NorthCoast 99'ers

Copyright 1988, by Martin A. Smoley

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.

```
* Command File MOVED1    10/27/88
* Save as MOVED1/C
* Move Data from TNames to NEWNames
*
CLEAR
CLOSE ALL
* SET TALK OFF
SET RECNUM OFF
SET HEADING OFF
SELECT 2
USE NEWNames
TOP
SELECT 1
USE TNames
TOP
LOCAL NUMT N 4 0
REPLACE NUMT WITH 1
  DO DSK2.MOVED2
SET RECNUM ON
SET HEADING ON
SET TALK ON
RETURN
```

```
* Command File MOVED2    10/27/88
* Save as MOVED2/C
*
WHILE .NOT. (EOF)
  SELECT 2
  APPEND BLANK
  REPLACE 2.NM WITH NUMT
  REPLACE 2.LN WITH 1.LN
  REPLACE 2.FN WITH 1.FN
  REPLACE 2.MI WITH 1.MI
  REPLACE 2.SA WITH 1.SA
  REPLACE 2.CT WITH 1.CT
  REPLACE 2.ST WITH 1.ST
  REPLACE 2.ZP WITH 1.ZP
  REPLACE 2.PH WITH 1.PH
  REPLACE 2.XP WITH 1.XP
  REPLACE 2.GP WITH 1.GP
  REPLACE 2.ID WITH 1.ID
  REPLACE NUMT WITH NUMT + 1
  SELECT 1
  MOVE
ENDWHILE
CLOSE ALL
RETURN
```

Here we are again TI-Base users. In this months tutorial I intend to back track, back paddle, and change my mind on a lot of things. Last month I said I almost had the manual wrapped up and I could stop writing large tutorials. I will attempt to cut the tutorial size, but I am discovering things that people either do not understand or do not even realize that TI-Base can do. I will attempt to remember that every item we cover is probably brand new to you, the reader. I started to fall into that trap where the writer (me), thinks that what he is saying should be completely clear to everyone. So, I will slow down and try to explain things more clearly.

The Command File on this page answers a question I received and covers some new points in the manual. The question was, "I completely set up a data base and typed in 100 names and addresses. I then realized I needed

one more field and one field that was Ccharacter should have been Nnumeric." I will attempt to cover this problem and a multitude of other things at the same time, because they all work together. So please try to bear with me for a while.

Let us work with TNames because we already have it typed in. Next, I am going to switch the wrong field problem to, "a Nnumeric field that should have been Ccharacter" (further explanation later). So, let us say you just typed in TNames and you have entered 100 names. We only have 5 names in TNames but the Command Files (MOVED1 and MOVED2) will not care if there are 5, 100 or 999 names in the data base. Now, when you CREATED TNames you typed in the information listed below. As you can see the first 10 items are Ccharacter type fields and the last item is a Nnumeric type field. Look it over.

```
arrows to move, enter to advance
FIELD DESCRIPTOR TYPE WIDTH DEC
1 LN C 15
2 FN C 15
3 MI C 2
4 SA C 25
5 CT C 20
6 ST C 2
7 ZP C 5
8 PH C 12
9 XP C 5
10 GP C 5
11 ID N 7 0
```

[TNames STRUCTURE]

Now, after entering all those names I decided that I should have put in another field. The field I want should be before the first field in this data base (before LN). I want it to be a number or Nnumeric field, and hold a number up to 999 with no decimal places. Since this is a number for each name and address record, I would like the first record to start with 1 instead of TI-Base's setup which is zero (0), and I would like the computer to put the numbers in for me. Next I decided I wanted the last field to be a Ccharacter field instead of a Nnumeric field, as it is now. If you use MODIFY STRUCTURE as described in the manual, adding a field will destroy the data and the names will be lost. So we will do it another way. Remember, this is a tutorial and this demonstrates programming techniques. The idea of adding a field or changing another does not have to be logical.

Here is one quick reminder. I keep several copies of my data bases on different disks. If one of these new Command Files I am testing wipes out the data base, I want to have a backup. So, with all this in mind, I decided to CREATE a new data base, leave it empty and transfer what I wanted from the old data base (TNames), into the new one. I typed CREATE NEWNames. When TI-Base gave me the structure entry screen, I typed in everything you see below.

```
arrows to move, enter to advance
FIELD DESCRIPTOR TYPE WIDTH DEC
1 NM N 4 0
2 LN C 15
3 FN C 15
4 MI C 2
5 SA C 25
6 CT C 20
7 ST C 2
8 ZP C 5
9 PH C 12
10 XP C 5
11 GP C 5
12 ID C 7
```

[NEWNames STRUCTURE]

Compare NEWNAMES structure to TNAMES structure. You will see that there are now 12 fields instead of 11. Notice that the first field is now NM, a N)umeric field, the size is 4 and 0 decimal places. The rest of the fields match TNAMES except ID which I have changed to a C)h)aracter field with a length of 7. When I entered all the information above and I was on the last 7 in ID, I pressed FCTN[8] to save and end the creation screen. When TI-Base asked if I wanted to enter data now, I answered N)o. At that point I had created NEWNAMES and it was completely empty. NEWNAMES is the data base I really need, so all I have to do is move all the data from TNAMES over to it and I will be happy.

I whipped up MOVED which I later turned into MOVED1 and MOVED2 (further explanation later). I still prefer FunnelWeb to produce my Command Files, but the TI-Base way is to type MODIFY COMMAND MOVED1 <E>. TI-Base will then start the procedure of producing the Command File named MOVED1 and place you in the EDITOR screen. Type in all the lines you see down to and including the first RETURN. At that point press FCTN[8] to SAVE/END and TI-Base will finish making MOVED1 for you. You may have to press FCTN[9] to break out of the editor at this point.

When you get back to the dot prompt start over, and with MODIFY COMMAND MOVED2 <E>, type the second Command File as you see it. I will start with MOVED1 and go through it. Remember that * in the first column means comment line and TI-Base will not execute that line when it reads the *. So you are probably looking at * SET TALK OFF and scratching your head. Normally I set talk off to keep the screen clear. In this case it made me nervous because I new that TI-Base was doing some real thrashing on two data bases and it concerned me. Seeing all the lines go by on the screen does not give you much more control, but at least you do not feel so left out, and you can see certain problems if you watch which lines execute. You can remove the * and turn off the screen junk any time you wish. SELECT 2 is where the good stuff starts.

I am going through this again. I hope I do not bore the people who already understand this aspect of SELECT. TI-Base has 5 areas. A different data base can be opened in each area. These data bases will remain open and you can work on all of them, but not all at once. let us say that each area is a cardboard box. You have 5 boxes. You must do any major work on only one box at a time, however, you can do minor work on the other 4 by reaching over into those boxes and picking items out. If you use the command SELECT, you can change the box that you wish to do major work on. If you are unsure of yourself, you should not only SELECT the important work area, you should also tell TI-Base exactly where things can be found by using the (x.) directive, where x is one of the area numbers (1 to 5) (further explanation later). Therefore, SELECT 2 means take box number 2 in your hands. USE NEWNAMES means place all the stuff called NEWNAMES into whatever box you are holding. In this case it is box 2. TOP means make sure that when we look into this box later the first thing we see is the first record in NEWNAMES. With NEWNAMES this is not important because it is empty, but with TNAMES it could be important.

Note: TI-Base does not associate the name NEWNAMES with box 2. You must remember what you have placed in which box. SELECT 1 means put box 2 down and pick up box number 1.

USE TNAMES and TOP is the same as above but using TNAMES this time. LOCAL NUMT 4 0, is the variable I will use to feed numbers into the field named NM. REPLACE NUMT WITH 1, puts the number 1 into NUMT. I previously said I would like to start numbering with 1. The next line is DO DSK2.MOVED2. This line runs the Command File named MOVED2 which is similar to a sub-program and is located on drive 2. I mention drive 2 because you could change the 2 to any drive you wish. If you have this Command File on drive 3, make it DSK3. etc. When this line is executed TI-Base runs MOVED2 and begins to do all the real work.

When TI-Base hits the statement WHILE .NOT. (EOF) we are working with box number 1. That is because it was selected last and therefore is still the current selection. It is also the box where we are keeping TNAMES. If we selected 2, where NEWNAMES is, the file is empty, so we would get an EOF signal and the WHILE would not execute. In our case it does execute, so we go through all the statements between WHILE and ENDWHILE. I immediately SELECT 2. This is necessary because I want to APPEND BLANK. To TI-Base this means, append one complete record (which in this case is all 12 fields) onto the end of the data base which is in box 2 (which in this case is NEWNAMES).

Since there is now some real space in NEWNAMES that we can fill with data, we will do so. REPLACE 2.NM WITH NUMT, takes the 1 which we previously placed in the variable NUMT and copies it into the field named NM in box 2. The 2. is to tell TI-Base box 2. It is not really necessary because we are in box 2 from the previous SELECTION. However, it helps me understand what is going on when I read over old Command Files a few weeks after I am done with them and I cannot remember what they were for in the first place. REPLACE 2.LN WITH 1.LN tells TI-Base to copy the last name from box 1 into the last name field in box 2. TI-Base works only with the box numbers, but for our understanding, we are saying take the last name we have in TNAMES and copy it into the blank last name field we have created in the new data base NEWNAMES. Notice I use the term COPY and not MOVE.

The next 10 REPLACE commands are the same as the one I just described. There is however, a difference in what is happening to the field in REPLACE 2.ID WITH 1.ID. Take a look at ID on the structure screens on the previous pages. You should notice that ID in TNAMES is a N)umeric field and it is a C)h)aracter field in NEWNAMES. This is one way we can change a field type. This change is not important at this time, but I received a question on the matter so I thought others might be having this problem. You can also convert C)h)aracter fields to N)umeric fields in the same manner. If that is the case, you must remove all characters from every field first. In other words the field may contain numbers only, at the time of conversion. You can also leave a particular field blank and use EDIT to type in data at a later time. OK, the last REPLACE. REPLACE NUMT WITH NUMT + 1 is an accumulator (remember from last month?). We are telling TI-Base to take whatever number is in the variable NUMT, add 1 to it, and place the new total back into NUMT. So the next time we REPLACE 2.NM WITH NUMT the result will be 2, 3, 4 and so on.

The next line (SELECT 1) is important. We must reSELECT number 1 (TNAMES) before the MOVE directive which is in the following line. In NEWNAMES we are at the EOF and have no place to move to. Also, when we hit the ENDWHILE and loop back to the WHILE .NOT. (EOF) statement we must already be in the full data base for the same reason. We will continue to jump back and forth, add new space to NEWNAMES and move data from TNAMES to NEWNAMES until we hit the End Of File (EOF) in box number 1 (which is TNAMES). Sounds easy right? Well it is for the Command File. At the EOF TI-Base jumps out of the loop CLOSES ALL the open data bases and RETURNS to the Command File named MOVED1. In MOVED1 it simply turns ON all the stuff we turned OFF previously and RETURNS you to the dot prompt.

For Your Information: First, I am covering the business of SELECTing a slot over again because if you wish to really use TI-Base you must fully understand the basics. If TI-Base can handle 17 fields in a data base and it can open 5 data bases at the same time (slots 1 to 5), then you have the potential of actively working with 85 fields at the same time. We have just worked with 23 fields at the same time. Think of how complicated a situation it could be with 85. This is why you must take the time to fully understand the basics of this language.

Next, I still prefer to use FunnelWeb to write and edit my Command Files. I realize that this will not be convenient for you under certain conditions, for

example, if you have TI-Base running and you would like to change the name of the data base you will use in slot number 1. If the Command File is too big to load with MODIFY COMMAND, you must leave TI-Base, load FunnelWeb, make the correction and then reload TI-Base. So, from now on I will try to keep the Command File segments small enough to be modified without leaving TI-Base. I still recommend that you use some means to produce D/V 80 type Command Files.

Last for now, I have changed my mind about creating large menu-type systems at this time. You may recall my mentioning this idea last month. I will stay with smaller and I hope more easily understood utility type Command Files for a while. Several people have told me that they are already lost. So I will slow down a little and try to accommodate everyone, if possible.

OK, back to work. While working on this tutorial and writing NOVED1 and NOVED2 I definitely did not get things right the first time. This created a related problem that covers some new items in that old manual. When I ran NOVED1 it filled NEWNAMES with junk, I literally mean junk. Since APPEND means stick more stuff on the end, the second time there was twice as much junk in NEWNAMES. What I needed was another Command File that would clean out NEWNAMES. So I whipped up the Command File named CLEAR.D.

```
*          10/25/88
* Command File CLEAR.D
* Save as CLEAR.D/C
* Clear Data from NEWNAMES
```

```
CLEAR
CLOSE ALL
* SET TALK OFF          |-----|
SET RECNUM OFF         | A SPECIAL NOTE FROM MARTY |
SET HEADING OFF       | I am presently testing    |
USE NEWNAMES          | TI-Base Version 2.0.     |
SORT OFF              |-----|
TOP                   | The improvements look     |
WHILE .NOT. (EOF)     | very very impressive.   |
DELETE RECORD        |-----|
MOVE
```

```
ENDWHILE
PACK
* Second time through.
```

```
TOP
WHILE .NOT. (EOF)
DELETE RECORD
MOVE
ENDWHILE
PACK
```

```
* That should do it.
CLOSE ALL
SET RECNUM ON
SET HEADING ON
SET TALK ON
RETURN
```

This Command File opens or USEs NEWNAMES, unSORTs the data base and starts at the TOP or first record. The WHILE loop loops until it hits the EOF, and while it is looping, TI-Base is DELETing records. In this application DELETE RECORD actually means mark the record TI-Base is presently looking at for later removal. This is for 1 record only. Therefore, we MOVE to each record, one at a time, and mark them all. When we hit the EOF, the WHILE kicks out and we execute PACK. PACK is a program segment of TI-Base and it resides on the TI-Base program disk. It permanently removes the records which have been marked for deletion. For some reason, unknown to me, I kept winding up with a record still left in NEWNAMES. Not always the same record but a record. The easiest way to handle this was to rerun the loop. This brings NEWNAMES up empty every time. You can check this by typing USE NEWNAMES at the dot prompt and then DISPLAY ALL. You will get a data base empty message. Well, I am going to break off here. I still have enough material for hundreds of tutorials, unfortunately it is all in my head. Remember that I have a tutorial disk available and you can join the NorthCoast Users Group, see tutorial 3 for details. And send those questions to me, I need to know what you need to know. Good luck. Marty.

Extended BASIC Tips

#12 and #13, by Jim Swedlow, CA USA

(This article originally appeared in the User Group of Orange County, California ROM, excerpted by S Shaw)

More on Print Separators

If, in a PRINT USING, you follow the print list with a print separator, your TI99/4A will honour it. Consider this:

```
10 FOR I=1 TO 30
20 PRINT USING "## ## ":I,I*I;
30 NEXT I
```

Instead of one column of numbers and their squares (which you would have without the semicolon after the I*I) you will get four right justified columns on your screen. Also works with printers and with DISPLAY USING.

Terminal Emulator 2 Ability

With a program in memory and your Speech Synthesizer attached, enter LIST "SPEECH" and see what happens. Strange! (Source: Regina in COMPUTE)

Line Number Table

What, you ask, is a line number table? Glad you asked! When your TI99/4A stores a program in memory or on disk, it is broken into two parts: the line number table and the line contents. In this program:

```
100 ! T1
110 FOR I=1 TO 10
120 PRINT I
130 NEXT I :: CALL CLEAR
```

The line number table looks like this:

| HEX | DECIMAL |
|-----------|-----------|
| 0082 37B5 | 130 14261 |
| 0078 37C2 | 120 14274 |
| 006E 37C6 | 110 14278 |
| 0064 37D3 | 100 14291 |

Note that the line numbers are in descending order. The second number is the memory location of the line contents (this is only an example). For example, the contents of line 100 start at >37D3. The > symbol means that the number is in hex or hexadecimal or base 16.

When you RUN a program your TI99/4A, during pre-scan, reserves memory for variables and other things. That is why you get different responses to SIZE before and after RUNNING a program.

While your computer always keeps the line numbers in descending order, it shifts the memory locations of the line contents around without regard to order as you edit/modify your program.

If you save a program to disk in MERGE format, enter NEW and then MERGE your program back into memory, your line number table should be returned to order. This may or may not have any real advantage, however.

What use is this? It depends. If you want to go looking in memory and learn how TI99/4A stores and executes a program, this is the first step.

If not? Well, now you know that each line number takes four bytes of memory (two for the number and two for the memory address). A line separator (::) takes only one byte. Conclusion: multi-line statements save memory!

Many files are stored in DISPLAY VARIABLE 80 format: TI99/4A writer files, Editor Assembler source files, text files, documentation files, etc. This month's program reads your disk for DV80 files and displays a menu on your screen.

You can then print up to 99 copies of any file with a single keystroke. Further, if your printer is GEMINI/EPSON compatible, you can activate double strike and/or emphasized print.

If you get strange error messages when you first run this program, change the ON ERROR statements in lines 170 and 270 (twice) to ON ERROR STOP and change the !@P- in line 150 to !. After you have finished debugging, restore these commands.

Enjoy!

```

100 ! FILE|PRINT
110 ! VERSION Extended BASIC 1.0
120 ! 26 MAY 85
130 ! By Jim Swedlow
140 !
150 N,@=1 :: DIM N$(24),S$(1),E$(1),D$(1),T$(1):: GOTO
160 :: CALL CLEAR :: CALL KEY :: A,B,D,E,H,T,A$,P$
:: !@P-
160 S$(0)="OFF" :: S$(@)="ON" :: P$=CHR$(27)::
E$(0)=P$&"F" :: E$(@)=P$&"E" :: D$(0)=P$&"H" ::
D$(@)=P$&"G" :: T$(@)=CHR$(12):: P$="PIO"
170 DISPLAY AT(@,10)ERASE ALL BEEP:"FILE|PRINT": :
:"Reading Disk Directory" : : ON ERROR 330 :: H=0 ::
OPEN #@:"DSK1.",INPUT ,RELATIVE ,INTERNAL :: INPUT
#@:A$,A,B,B
180 INPUT #@:A$,A,B,B :: IF A$<>"" THEN IF ABS(A)<>2 OR
B<>80 THEN 180 ELSE H=H+@ :: IF H<25 THEN N$(H)=A$
:: GOTO 180
190 CLOSE #@ :: ON ERROR STOP :: IF H>24 THEN
DISPLAY AT(3,10):"WARNING": "Only the first 24
files were": "read, there are more." :: GOSUB 310
200 DISPLAY AT(5,@):: GOSUB 300 :: B=14 :: FOR A=@ TO H
:: B=15-B :: DISPLAY
AT(INT((A+@)/2)+6,B):CHR$(64+A);" ";N$(A):: NEXT A
210 DISPLAY AT(20,@):"Press < > to change": "<1>
Emphasized <2> # Copies<3> TOF at End <4> Db1
strk<5> Printer <6> Stop"
220 GOSUB 300
230 CALL KEY(3,B,A):: IF A<@ THEN 230 ELSE IF B>64 AND
B<65+H THEN 270 ELSE IF B=54 THEN CALL CLEAR :: STOP
240 IF B=49 THEN E=@-E :: GOTO 220 ELSE IF B=52 THEN
D=@-D :: GOTO 220 ELSE IF B=51 THEN T=@-T :: GOTO
220
250 IF B=50 THEN ACCEPT
AT(3,26)SIZE(-2)VALIDATE(DIGIT)BEEP:N ::
N=MAX(N,@):: T=T-(N>@):: T=MIN(T,@):: GOTO 220
260 IF B=53 THEN ACCEPT AT(5,9)SIZE(-20):P$ :: IF P$=""
THEN 260 ELSE 210 ELSE 230
270 DISPLAY AT(20,@):"Printing ";N$(B-64): : : : :
ON
ERROR 340 :: OPEN #2:P$ :: PRINT #2:E$(E);D$(D):: ON
ERROR 330 :: OPEN #@:"DSK1."&N$(B-64):: FOR A=@ TO N
:: RESTORE #@
280 IF EOF(@)=0 THEN LINPUT #@:A$ :: PRINT #2:A$ :: GOTO
280
290 PRINT #2:T$(T):: NEXT A :: PRINT #2:E$(0);D$(0)::
CLOSE #2 :: CLOSE #@ :: GOTO 210
300 DISPLAY AT(3,@):"Emphasized ";S$(E);TAB(17);"#
Copies";N:"TOF at end ";S$(T);TAB(17);"Db1 strk
";S$(D):"Printer ";P$ :: RETURN
310 DISPLAY AT(20,@)BEEP:" Press any key to continue"
320 CALL KEY(3,A,B):: IF B<@ THEN 320 ELSE RETURN
330 DISPLAY AT(3,@)BEEP: : " Disk and/or file were not
": " found in drive 1 ": "Insert disk in drive 1"
:: ON ERROR 350 :: CLOSE #@ :: ON ERROR 350 :: CLOSE
#2 :: GOSUB 310 :: GOTO 170
340 ON ERROR 350 :: CLOSE #2 :: DISPLAY
AT(20,@)BEEP:"Printer name is invalid" :: B=53 ::
GOTO 260
350 RETURN NEXT
360 END

```

When you catalog a disk, there are five types of files: Program, Display Fixed, Display Variable, Internal Fixed and Internal Variable.

A program file varies from the others as they are a collection of individual records (data) while a Program file is a "memory image".

Memory image means essentially that, when writing to disk, your TI99/4A takes what is in RAM and dumps it to disk. In other words, it is a copy or image of what is in memory.

There are at least three types of program files: BASIC and Extended BASIC programs, Assembly language files saved in program format and data files produced by some modules (Personal Record Keeping, Adventure , etc).

It is only when you attempt to load and execute that your computer notes the difference (and it often locks up if you load a program file that does not match the application).

Program files (be they programs or data) load significantly faster than other files. Why? Well, think of a bag of marbles. When in memory image, loading the file into RAM is a matter of pouring the bag from disk to CPU. In any other format, however, you take the marbles one by one from disk to memory. Dumping the bag is simply faster.

Finally, Assembly language programs in mirror image format can be loaded from the Editor Assembler module or from TI-Writer's option 3. In both cases, the module will auto boot a program entitled DSK1.UTILL.

IF THEN ELSE with GOSUB

I bumped into this while debugging a program. What output would you expect from this:

```

10 FOR I=1 TO 3
20 IF I=2 THEN GOSUB 40
   ELSE PRINT "TEST"; ::
   GOSUB 40
30 NEXT I :: STOP
40 PRINT I :: RETURN

```

You might expect this:

```

TEST 1
2
TEST 3

```

If you enter and run this program, you will actually get this:

```

TEST 1
2
2
TEST 3

```

Why two 2's? When I=2, the program flow is thus:

Line 20 tests I and finds that it is 2. We then GOSUB 40, PRINT I and RETURN. Upon return to line 20, we look for the next instruction.

The ELSE part is ended with the separator :: , the code AFTER the separator is treated just as though it was on another line, that is as a separate instruction.

The message here is that when using a GOSUB inside an IF THEN ELSE, do not execute a GOSUB immediately before an ELSE that is followed by a line separator.

An Alternate for ELSE

The following two lines execute in the same manner:

```

20 IF A=3 THEN 200 ELSE A=5
20 IF A=3 THEN 200 :: A=5

```

This only works when the sole code between THEN and ELSE is a line number.

Tigercub Public Domain Catalog #3

by Jim Peterson, Tigercub Software, USA

900. HOME UTILITIES (351) Floor Plan; Bartender (Burchfield); Buying a Home? (J. Simpson); Calendar (M. Tsukroff); Calendar (A. Blundell); Calendars (A. Swales); Cost of Credit (J. Peterson); Coupon Catalog (B. McFadden); Decorating Helper; Freezer Inventory (K. Romstedt); Home Record Keeper (B. Rutherford); Grocery Shopper, Grocery List (J. Peterson); Measurement Formulas (J.L. Webb); Home Secretary (M. Subbaiah)

901. HOME UTILITIES #2 (342) Kitchen Aids (D. Connelly); Ladies' Panel Skirt (Regena); Monthly Expenses (C. Iliff Jr); Mortgage Comparison; Needlepoint Calculator; Paint Calculator (J. Peterson); Phone Number Wording (D. Smith); Phone Toll Storage (M. Nomina); Phono Album and Tape File, Book//Article Keyword File (C. Good); Savings and Loan Analysis; Floor Covering Program; Telephone Cost Monitor; General Will.

902. HOME UTILITIES #3 (350) Address Manager (K. Romstedt); Computer Aided Design (S. Johnson); Electronic Card File (M. Greeley); Calendar Program (G. Lambert); Videotape Data Base, Appointment Calendar.

907. SCREEN DRAWING, DOODLING (155) Colour Crayon (Cooper/Falkin); Draw Program (D. Mason); Dynamic Doodler (T. Falco); Kwikdraw (L. Long); Low Resolution Drawing Painting (S. Anderson); Make Your Mark (R. Taylor); Quick and Dirty Doodling (J. Peterson); Draw Program; Welcome.

909. HIGH RESOLUTION DRAWING (287) DRAW (anonymous), high resolution drawing with assembly speed, and dump to printer. Requires Editor/Assembler.

910. CHARTS & GRAPHS (178) Bar Graph (C. Leonard); Bargraphs (K.B. Koch); Bar Graph Plot/Message (D. Schell); Bars and Graphs (J. Gunter); Bar Chart (S. Johnson); General Bar Grapher (E. Benson); Graphprint, Graphpage Maker, Segmented Bargraph (J. Peterson)

912. CALCULATORS & CONVERTERS (345) Adder Upper, Binary to Decimal, One Finger Figurer, Tigercub Multimemory Calculator (J. Peterson); TI-59 Calculator (L. Thelander); Binary Numbers, Hexadecimal Numbers (G. Schechter); Computer Calculator (J. Schell); Conversions #1 and #2 (R. Hiser); Conversions (D. Irwin); Days Dates (P. Powers); Decimal to any Base (J. Aaron); Day of Week (J. Hamilton); Exposure Program (P. Yorke); TI Calculator (C. Bobbitt); Hexadecimal to Decimal Converter (Carpenter/Shaw); Large Pocket Calculator (T. Arnold); Metric to Imperial Conversions (J. Luck); Base Change Calculator, Swedish Base Conversion, Base to Decimal Conversion, PRK Calculator, Calculator, Converter and #2, Decimal/Hex/Binary, Decimal to Fraction, Hexadecimal to Decimal.

913. CALCULATORS & CONVERTERS #2 (147) Cyber Abacus (S. Williams); Easter (Nahigian/Hodges); Handy Dandy Adder (J. Shattuck); Printing Calculator (D. Hicks); Romstedt's Converter (K. Romstedt); Radix Number Converter (G. Steffen); Unit Converter (W. Richardson); Universal Calculator.

915. FINANCIAL MATH (339) Expense Account (R. Paret); Trend Line Analysis (C. Garcia); Breakeven Analysis (S. Morabito); Job Costs (L. Derusha); Personal Banking (D. Migicovsky); Rule of 78 (H. Templeton); Budgetron, Budget Worksheet, Breakeven Analysis #2, Interest Projection, Mortgage Comparison, Mortgage Computation, Mortgage Machine, Loans, Stock Market, Stock Record Program.

916. FINANCIAL PROGRAMS (356) Amortization Schedule (M. Holgers); and #2 (J. Roche); Compound Interest (C. Good) Estate Tax Securities (R. Shumaker); Debt Calculator (K. Romstedt); Financial Math

(C. Ehninger); Financial Statement Ratio Analysis (C. Colton); Investment Analysis (A. Robertson); Investments (G. Lockhart); The Cost of Credit (J. Peterson); Saving (J. Moskowitz); Loan Calculator (H. Button); Loan Interest (D. Thomas); Loan Calculator (J. Peterson); Rule/78 (D. Kreutner); Present Value, Amount, Amort Schedule, Annuity, Conventional Loan Analysis, Finance, Investment Analysis, Loan Calculator.

918. CHECKBOOK PROGRAMS (203) Checkbook Writer V4.41 (M. Nomina, archived); Checkbook Program (J. Taylor); Checkbook/Statement (G. Venechuk); Checkbook Register (R. Eister)

920. BUSINESS PROGRAMS (146) Cash Flow (J. Moskowitz); Self Help Tax Cut (K. Kuehne); Mailing List Address Program (A. Cox); Bidding.

950. GENEALOGY Genealogical Data Base (Hunter Valley UG) in archived format; Family Tree (Bill Longmuir) with instruction program.

970. ASTROLOGY, NUMEROLOGY, c (171) Astrological Horoscope; Astrology (G. Horowitz); Numerology (D. Cook); Talking TI Tarot (Migicovsky) with TELL speech; Zodiac Sign(G. Schechter).

980. RADIO UTILITIES (220) Amateur Radio Directory (R. Eister); Antenna Position for Geosynchronous Satellites (E. Stork), and another (anonymous); Directional Antenna Computer (R. Fetters); Log Periodic Dipoles (Taylor/Welham); Polar Plot Modeling (Robertson); Name Call Sign File, Ant #1, Calculate Antenna Position, DX Contest Dupe Finder, Parabolic Reflector, Polar Orbiting Satellite, Polar Plot Modeling, QNI, Geostationary Satellite.

990. SPORTS (329) Ping Pong Scoreboard (P. Mulvaney); Baseball statistics, Bowling statistics, Golf Handicapping (T. Castle); Handycapper (Nickel); League Scheduler (Burwells/ Peterson); Magic Number (Statt/de Marti); Fumble (D. Baker); Tournament Fishing (Ed Will); Individual Bowling statistics, Power Rater, Trifecta.

1100. CHARACTER & SPRITE EDITORS (346) Auto Character Definer (Texas Instruments); Character Definer (Regena/Powers); Character Definer (I. Pegg); Char Definer Program (R. Mitchell); Char Definer Worksheet (C. Bobbitt); Character Designer/Editor (S. Meadows); Sprite Definer IV (K. Romstedt); Sprite Designer (G. Krawzyk); Sprite Editor (N. Shah); Sprite Editor #2; Sprite Master (T. Kaplan); Graphics Generator (B. Traver); Tinychar (M. Stanfill); Character Programmer (K. Kargl); Sprite V3.0 (Bill Hudson; archived; requires E/A module)

1101. PROGRAMMERS' UTILITIES (346) For/Next Checker, Go-Search, Linewriter, Supertrace, Decompactor, Progwriter, Checker, REM Remover, by Jim Peterson; GOSUB/N, Char Coder, Merge Editor, Program Checker, by B. Traver; Compare, Trace, by Mike Dodd; Extended BASIC Program Compressor, Nesting Error Analysis, by Karl Romstedt; Program Compactor (John Dow); Extractor (George Steffen); Program Rewriter (Paul Yorke); 2 Dimensional Sort (I. Swales); Memoff Memon (C. Miller); Shrink (M. Rosenfeld); SuperXB V1 (Art Byers), archived; Unmerge (Ed Neu); BXB (Jim Peterson)

1102. SORTS, SCRAMBLES, SEARCHES (228) Tigercub Sortwatch, 2 Dim. Swap Sort, Wazzit Sort, High Scramble, Simulation of Card Shuffle, Disksort (J. Peterson); 2 Dim. Sort (I. Swales); Jeb's Sort (J. Hamilton); Relative File Sort (J. Prescott); Peek Scramble (J. Mitchell); TI-Writer Text Sort (J. Keisler); Bubble Sort, Easy Sort, Heap Sort, Integer Flag Sort, Long Shell Sort, Quick Sort, Resort Sort, Selection Sort, Shaker Sort, Short Shell Sort, Swap Sort, Shuttle Sort, Insertion Sort, Sort Demonstration, Sort Demonstration #2, Sorts, Quick Scramble, Tablesort; Binary Search Demonstration (C. Zarger)

1105. AUTO-LOADERS (217) Auto Program Booter (C. Bobbitt); Auto Load (B. Bartlett); Disk Indexer (de

Frank), Disk Indexer #2 (de Frank/Traver), #3 (de Frank/Hughes); Disk Loader (C. Scally); Free Loader (Karl Romstedt); General Purpose Program Loader (Ehninger/Hammans) and #2 (anonymous); Tigercub Menu Loader (Jim Peterson); Two Column Menu; Loadmaker (K. Romstedt) Hawkins' Loader (F. Hawkins)

1106. DISK CATALOGERS (268) Cat/4/Disk (T. Knight); Catalog (M. Chartier); Twocat, Disk Matcher, Doublecat (J. Peterson); Disk Lister (F. Hawkins); Disk Lister #2 (S. Davis); Double Lister (D. Wakely); Diskrunner (M. DeFrank); Flippy Disk Cataloger (B. Floyd); Library Program (M. Boyle); TI 99/4A Catalog Program (J. Rodrigue); Triple Lister (J. Boylan); Disk Catalog; Supa Sorta (E/A Option 3)

1107. CHARACTER SETS, ETC. (353) Big Char, Two Die Utility, Charcopy, Heavychar, Lower to Upper Case Convert, Random Character Generator (J. Peterson); Big Letters (P. Yorke); Character Enlarger (M. Jaegermann); Slanted Characters (E. Janensch); Character Set (J. Hedstrom); Charload (J. Stringfellow); Greek Alphabet (P. McPhie); Inverse Lower Case (R. Kellogg); Russian Keyboard Configuration (M. Dery); Double Height (P. Dunderdale); Large Upper Case (D. Carrera); Large Letters Graphics Aid (B. Knecht); Mirror Typing (D. Huggett); Hex Codes; Lower Case; Pretty Face; Hebrew; Japanese; Lower Case #2; French; Math Chars; Russian; Large Character Set; Flip (R. Tamashiro)

1110. ASSEMBLY UTILITIES (357) Assembly Opt 5 Menu Loader (Wright/ Romstedt); Assembly to Call Load, Call Load to Assembly, Convert to Call Load, Hidden, by Tom Freeman; Convert (Zerba Zorab); CRU Look V2.0 (T. Kaplan); E/A Opt 5 Loader (B. Boone); Call Peek (C. Good); Memory Manipulator (Michael/ Albright); Sector Surgeon (The TI Elf); Read (Bud Wright); Peeker (C. Miller); Memory Peeker (Ed York); Memory Image File Copier, Hex Dump of Memory, Disassembler, and twenty CHARAL sets.

1111. ASSEMBLY UTILITIES ROUTINES (328) Box Demonstration (R. Terry); ACE (P. Bagnaresi); Windowing (J. Stringfellow); Screen Pager Utility (M. St. Vincent); Hexadecimal to Decimal, Decimal to Hexadecimal (R. Lathouse); Large Characters (J. Bohos); Sector (R. Jones); Sort; Disassembler; Hex Dump of Speech; Help Screen.

1112. NEW HORIZON ASSEMBLY UTILITIES (269) by John Clulow, David Romer, Ken Hopkins. Fast assembly screen saver, sort, horizontal scroll, etc., etc. With archived source code.

1119. HARDWARE UTILITIES (169) Disk Drive Speed Check (Boone/Renfro); TV Test Patterns (G. Dombroski); TV Adjustment (P. Yorke); TV Colour Test (C. Bobbitt); Burglar Alarm (R. Lumsden); Memtest2 (D. White); VCR Title Generator (J. Theis); Screen Test (P. Saturn); VCR Counter Table (H. Hoyt); Sound Show (Bill Gronos); Dator Testbild.

1120. SOUND EFFECTS (197) Church Bells (R. Davy); Sound Effects Catalog #1 and #2 (R. Wilson); Shaping TI Sounds (S. Kaye); Sounds of Space (J. Peterson); Call Tone; Drums; Funny Noises; Quickie; Sound Developer; Sound Effects #1, #2; Sounds #1 and #2.

1130. DISK LABEL/JACKET PRINTERS (296) Disk Name Printer (B. Rutherford); Print an ID, Disk Labeler for Gemini (Ed Machonis); Cassette Liner Maker (D. Swallow); Cassette Sleeve Maker (C. Provance); Disk Labeler (F. Hawkins); Disk Labeler V2,0 (B. Neal); Disk Labels (J. Shregardes); DiskLabel V3.0 (Neal/Justice); Gunned Label Maker (K. Romstedt); Label 1, Label P (D. Kreutner); Tigercub Labeler (J. Peterson); Disk Jacket for Gemini 10X, Epson, Gemini SG-10 (W. Brown); Disk Sleeve Printer (Baxter/Crowder); Graphic Jacket (J. Bartle); Printjack (Proctor/Floyd); Diskette Labeler; Print a Tag (Ed Machonis)

1131 GEMINI PRINTER UTILITIES (224) (Not for NX-10) Gemini Printer Setup (A. Rice); Tigercub Labeler, Tigercub Printall, Long Page Printall, Slash Zero,

Tigercub Downloader, Chinese Printing, Cryptprint (Jim Peterson); Style a Line (Ed Machonis); Gothic Lettering (R. Mainzer); Old English; Printer Setter for Gemini (G. Lambert); Gemini 10X Printer Setup, Gemini Printer Setup, Gemini Graphics Demonstration, Gemini Type Styles.

1132. WORD PROCESSING UTILITIES (182) LF Stripper, CR Adder (J. Swedlow); Stripper (G. Steffen); Strip, 28 Column Converter, Word Count, Hyphenator, Tigercub Keysearch, Tigercub Unfiller (J. Peterson); Word Counter (R. Mudie); Printer Initialization (K. Romstedt); Multiprint (Steffen/Machonis) Doubleprint (T. Freeman); Breakfile (B. D/V80 Word Count utility (requires E/A module); Tablesort.

1133. BANNERS, GRAPHS, ETC. (203) Banner (J. Perry); Banner #2 (J. Evans); Basic Banner, Mazemaker (S. Karasek); Maze (Schaal/Weaver); Graphsheet Maker (J. Behnke); Graphprint, Tigercub Segmented Bargraph, Pagemaker, Wordsearch Maker, Tigercub Magic Square Maker, Streamer Printer (J. Peterson)

1135. SPEECH UTILITIES DEMONSTRATIONS (355) Alphon, Alphon Printer (Dallas Phillips); Alphon Demonstration (R. Albright); Alphon Demonstration (Barb Berg); Allophone Codes, Annie (Bonnie Snyder); Howie's Demonstration (H. Rosenberg); Feeling Groovy (T. Decicco); Verbose (D. Braden); Find More Speech (D. Gilchrist); Ernie Bert; Speech Demonstration (Regena); Speech Editing Program (M. Bauer); Vocabulary of Speech Synthesizer (Robbins); Singing Voice with Scales; Speechload.

1140. MUSIC COMPOSERS (288) Music Writer (R. Albright); Aid to Programming Music (E. Dodd); Composer (D. Cook); Music Creator (B. Charbonnet); Music Machine (S. Shaw); Music Magic (C. Burris); Musicoder (B. Traver); Player Piano (D. Leiner)

1141. ASSEMBLY MUSIC COMPILER (265) To create and compile music in assembly. Anonymous.

1145. TELECOMMUNICATIONS AIDS (342) Encoding Decoding Program (Wm. Meredith); Mail Prep (J. Clulow); Off Line Source Mail (Clulow/Westner); Phonemake (S. Olsen); PRBase Modem Utilities (R. Baker); TELI Log On File Generator (G. Smith); Tex-Thello for Modem; Battle of the Sea Lords; Magic File Manipulator (B. Hatheway); 40 Column BBS Mail Prep (Tamashiro/Knecht)

1150. PROGRAMMING TUTORIALS (348) Game Programmer (B. Kafer); Music Programing, Sortwatcher, About Sub programs (Jim Peterson); Animation (W. Balthrop); Call Sound Effects (Tom Moran); Angry Bull (Regena/Powers); Basic Primer II (Sam Moore Jr); The Quote Puzzle (J. Peterson); Screen Graphics Mini Tutorial (John Taylor); Tiny Tips #1 (Ray Kazmer); Sound Tutorial (Chick de Marti); Time Sorts (Art Byers)

1160. ASSEMBLY TUTORIAL #1 (231) Assembly Made EZ, Help Files (Iain Johnson)

1161. ASSEMBLY TUTORIAL #2 (289) Assembly Made EZ, Subroutines (Iain Johnson)

1162. ASSEMBLY TUTORIAL #3 (357) Linking to Assembly, Disk 1 (Ross Mudie)

1163. ASSEMBLY TUTORIAL #4 (358) Linking to Assembly, Disk 2 (Ross Mudie)

1164. ASSEMBLY TUTORIAL #5 (340) Linking to Assembly, Disk 3 (Ross Mudie)



Forth to you too! Session 3

Author unknown

As I mentioned earlier, there is an elegant way to boot whatever you want your system disk to be ready for, but before we can proceed with the installation of an autoboot we need to consider the following.

Since Forth is a disk-based system (none of it comes in a command module or in the console itself), it occupies memory which otherwise would be available for programming. That, in my opinion, is the reason TI provided many of the utilities as LOAD options. Look at the menu and also Appendix F. Some of the options, the editor for example, are essential, others are rarely needed. For instance, if you are not programming in Code there is no need to clutter up the memory with -CODE and -ASSEMBLER. Similarly, if you are not going to operate with graphics then there is no need for -VDPMODES etc. It is not very likely that you will run out of memory while still in the learning process but why boot unnecessary stuff? I consider only -DUMP, -COPY and -PRINT along with the editor as essential for now. To see how fast memory is occupied even with your extra 32K, do this (assuming you are in Forth):

```
: FREE SP@ HERE - . ;  
(colon FREE SP@ HERE minus dot semicolon)
```

Now enter FREE. You should get an answer of about 14140 (9790 if you opted for the 64-column editor). If you want to see how fast memory shrinks with each LOAD option boot a few more, but enter FREE in between them. (If you are convinced enter COLD.)

Here is my suggestion for a beginner's boot:

1. -PRINT, to be able to list or dump screens
2. -COPY, to copy and move screens
3. -DUMP, to look at the parameter stack
4. -BSAVE, this makes the binary image save (and load) possible

If you want to use a printer there is one more item to check. Look at screen 72 in the manual or, for practice, call it up from your disk. Look at where it says "RS232.BA=9600". This routine is written for a serial printer operating at 9600 baud. If you have a PIO printer you must modify screen 72 as shown below and FLUSH it to your system disk.

```
SCR# 72  
0 ( ALTERNATE I/O SUPPORT FOR PIO PRNTR O4/27/84 LW )  
1 0 CLOAD INDEX BASE->R DECIMAL 68 R->BASE CLOAD STAT  
2 0 0 0 FILE >PIO BASE->R HEX  
3 : SWCH >PIO PABS @ 10 + DUP PAB-ADDR ! 1- PAB-VBUF !  
4 SET-PAB OUTPUT F-D" PIO" OPN 3  
5 PAB-ADDR @ VSBW 1 PAB-ADDR @ 5 + VSBW PAB-ADDR @ ALTOUT ! ;  
6 : UNSWCH 0 ALTOUT ! CLSE ;  
7 : ?ASCII ( BLOCK# --- 1 : )  
8     BLOCK 0 SWAP     400 + SWAP  
9     DO I C@ 20 > + I C@ DUP 20 < SWAP 7F > OR  
10     IF DROP 0 LEAVE ENDIF LOOP ;  
11 : TRIAD 0 SWAP SWCH 3 / 3 * DUP 3 + SWAP  
12 DO I ?ASCII IF 1+ I LIST CR ENDIF LOOP  
13 -DUP IF 3 SWAP - 14 * 0 DO CR LOOP  
14 OF MESSAGE OC EMIT ENDIF UNSWCH ;  
15 R->BASE --->
```

To make sure that everything is OK with your new version of screen 72, enter -PRINT, let the system boot it, turn on your printer and when you see the 'ok' type:

```
SWCH ." THIS IS A PRINT TEST" CR UNSWCH
```

Make sure there is a space between ." (DOT-QUOTE) and THIS. If your printer responds with THIS IS A TEST, pat yourself on the back and play with SWCH ." xxxxxxx" UNSWCH some more. If not, you will have to start over again, and this time pay close attention, particularly to spaces!

Before we proceed with the actual set-up for your autoboot take a quick look at any screen between 8 and 19, no not in the manual, on your display (remember nn EDIT). Not much there that is legible, but believe it or not on those few screens resides every Forth word that is identified in the Glossary as a resident word, only they are saved in a binary form. We will do the same with the LOAD options you decide upon by the use of BSAVE. So let us go. First, start off with COLD, then boot your options by entering the appropriate words (-PRINT, etc) and as the final one -BSAVE. Find the apostrophe key (FUNCT[0], that is 0 not zero!). This is also a Forth word pronounced tick (page 3, Glossary). Now enter:

```
' TASK 22 BSAVE .
```

Here is what is happening. We are saving in binary form all that has been added to the dictionary (by booting the LOAD options) starting at screen 22. We can afford to wipe out 22 and some of the following screens because they contain the 64 column editor which you have either booted already (so it is in the autoboot dictionary) or you are not going to use it. The final dot will print on your display the first screen available after the BSAVE. All other LOAD options are still intact and can be booted when needed. Now for the finishing touches enter

```
EMPTY-BUFFERS 3 EDIT
```

and carefully erase all but lines 0, 1, 2, 13 and 15. On line 2 change 20 LOAD to read 22 BLOAD. You might want to replace the word BOOTING on line 0 with some other phrase which would let you know that you are using your new system disk. On line 5 put: 0 DISK_L0 ! and depending on how many drives you have and whether they are single or double sided enter one of the following: (if you have only one single sided drive you can skip this)
90 DISK_SIZE ! 180 DISK_HI ! (for 2 single sided drives)
180 DISK_SIZE ! 180 DISK_HI ! (for 1 double sided drive)
180 DISK_SIZE ! 360 DISK_HI ! (for 2 double sided drives)

Note that these words use the underline, not the hyphen. (If you have double density drives it is not quite that simple, you have to make several modifications to screens 33 and 40 as well as the disk header before you can utilize DSDD to its fullest extent.) After you have FLUSHed your edit to the disk, go COLD and, with any luck, you will have a working, auto-booting system disk.

As I mentioned before, you should be learning how Forth works and get Brodie's Starting Forth (or Programming Forth by Chirlian.) These tutorials will continue, but are not intended as a substitute for a textbook. It is time to start learning and to do that read Chapters 1 and 2 of Starting Forth. I went through the book first and annotated each page with the corresponding remarks from the manual's Appendix C (Notes on Starting Forth). This helped to lessen my initial confusion (at least somewhat).

Forth Tidbits

by Lutz Winkler, USA

F-Tidbit #7

Forth Style (or My Pet Peeves About Forth Program(mer)s)

It has been said that Forth programmers are rarely able to read their own programs a couple of months after writing them. Judging by some of the material I have seen, this certainly could be true. The most frequent sin is omission of the stack picture. By convention, this is the (---) placed after a word's name. You immediately know something about the word. It could be that it needs a parameter (n--) from, or leaves something on (--n or --f) the stack. In either case, one can tell right away. If the word is neutral (no stack effect) it is quite all right to omit it. The key is consistency.

Selecting a good name seems to be a problem for many. There is nothing quite as bad as an abbreviation which has no meaning to the person trying to dissect the program. I am one of those who does not like to just type them in, I like to understand and learn from them, too. Choose words so they convey an idea of what they do, for the sake of the reader/user. For example, words that begin or end with a period always perform an output function, regardless of whether it is to the screen or a printer. In similar fashion, a question mark at the beginning indicates that a test is being performed (a la ?KEY). A trailing ? means an input prompt.

Then there is the matter of the program as it is written on the screens. Sometimes it is difficult to pick out the colons to see where a definition begins. Screens are crammed as if the programmer's life depended on saving one lousy screen. ("By gum, I am going to save me one lousy screen, come h--- or high water", seems to be their motto.) I have probably passed over some fairly useful stuff simply because the style was so bad and I did not feel like fighting my way through the morass.

These are my pet peeves after looking at lots of Forth screens. How should it be done and what should it look like? Here is an example. This little program does not do much except print labels. But I think it illustrates good Forth style very well:

```

0 ( ADDRESS LABELS - 1 label Size: 1 x 3 inches )
1 BASE->R DECIMAL 33 C MIZE 72 CLOAD TRIADS
2 : LAB ; ( boundary for running )
3 : CLEARPAD PAD 150 BLANKS FILL ; ( 150 spaces into PAD )
4 : DASHES 30 0 DO ."-" LOOP ; ( display 30 dashes )
5 : FORMAT. CLS 14 3 AT ." LABELS"
6           2 5 AT DASHES 2 7 AT DASHES
7           2 9 AT DASHES 2 11 AT DASHES
8           2 13 AT DASHES ;
9 : CONTENT? ( put 5 lines of up to 30 characters into PAD )
10          2 5 AT PAD 30 EXPECT 2 7 AT PAD 30 + 30 EXPECT
11          2 7 AT PAD 60 + 30 EXPECT 2 11 AT PAD 90 + 30 EXPECT
12          2 13 AT PAD 120 + 30 EXPECT
13 : LINES ( retrieve contents of PAD )
14          150 0 DO PAD I + 30 -TRAILING TYPE CR 30 +LOOP ;
15 -->

```

```

0 ( ADDRESS LABELS - 2 )
1 : HOW_MANY? ( --n, number of labels to be printed )
2           2 16 AT ." How many ? : " QUERY INTERPRET ;
3 : .LABELS ( n-- , output n labels to printer )
4           SWCH 0 DO LINES CR LOOP UNSWCH ;
5 : ?MORE ( --f, continue program if YES )
6           2 16 AT ." More ? (Y/N) "
7           KEY DUP 89 = SWAP 121 = OR 0= ;
8 : REMIND CLS CR ." Enter FORGET LAB" ;
9 : TEST PAD 150 65 FILL 1 .LABELS CLEARPAD ;
10 : LABELS BEGIN CLEARPAD FORMAT. CONTENT? HOW_MANY?
11           .LABELS ?MORE
12           UNTIL REMIND ;
13 CLS 2 10 AT ." Enter TEST for printer line-up" CR
14          2 12 AT ." LABELS to run program" CR
15 R->BASE

```

As you can see, the top level word by itself gives an almost complete description of how the program goes:

```

BEGIN      Start an indefinite loop
CLEARPAD   Fill the PAD area with blanks
FORMAT.    Put line guide on the screen
CONTENT?   What goes on the label
HOW_MANY?  Input how many copies of this label to print
?MORE      Print more labels? (Leaves a false flag for UNTIL)
REMIND     Issue reminder to use LAB for FORGETting the program

```

In addition to the above, there are only three more:

```

DASHES     Put 30 dashes on the screen (line guide)
LINES      Retrieve the contents of PAD
TEST       Print 5 lines of capital As for printer alignment

```

Note that all colons are at the beginning of the lines and very easy to spot. True, it is very short - but the longer a program is, the more helpful and important good style becomes.

F-Tidbit #8 Conversion of Forth Screens to D/V 80 (and Back to Screens)

Some time ago, there was a program floating about to convert Forth screens to DIS/VAR 80 files. If memory serves me correctly, it was 13 (!) screens long and required a lot of typing by the user even including that ugly F-D" word. Now, in a recent issue of the LA 'TopIcs' newsletter (with credit to the TINS newsletter) I spotted a three screener to do the same thing. Though 16 line screens were shown, the article stated that the code is Wycove and I/O specific words would have to be replaced to make it work with TI-Forth. If you do not know Wycove's I/O words (or what they do) this can be a chore. For those who need to include Forth screens in text files but were stuck with that 13 screen conversion program, here are 2 screens that will accomplish the same thing and will even convert a DV80 file into Forth screens.

```

0 ( SCREENS>FILE ) BASE->R DECIMAL 68 CLOAD STAT 0 CLOAD F>S
1 HEX 0 VAL 1 E BUF 4E ALLOT
2 PABS @ 2+ 1400 FILE XF XF
3 : FNAME ( --cnt ) CR ." Filename: " PAD 40 20 FILL
4           PAD 1+ 30 OVER OVER EXPECT -TRAILING 2- -DUP ;
5 : SETFILE ( n-- ) IF XF SET-PAB VRBL 50 REC-LEN DUP
6           ROT 1- C1 + PAD PAB-ADDR @ 9 + ROT VMBW
7           ELSE DROP APPND THEN ;
8 : S>F FNAME SETFILE OPN
9           CR ." Start from screen: " . :
10          CR ." End with screen: " . ! . 1+ SWAP
11          DO I BLOCK 400 OVER + SWAP
12          DO I BUF 20 MOVE BUF 40 -TRAILING 1 MAX WRT DROP
13          40 +LOOP LOOP CLSE
14          CR CR ." S>F, F>S or FORGET BUF" CR ;
15 R->BASE -->

```

```

0 ( FILE>SCREENS )
1 BASE->R HEX
2 : EOF STAT 1 AND 0= ;
3 : F>S FNAME SETFILE OPN
4           CR ." Destination screen: " ( INTERPRET
5           CR ." How many screens?: " ( INTERPRET
6           OVER + SWAP
7           DO EOF IF I B - UPDATE 400 OVER OVER 20 FILL OVER + SWAP
8           DO EOF IF RD I ROT CMOVE ELSE LEAVE THEN
9           40 +LOOP
10          THEN
11          LOOP CLSE FLUSH
12          CR CR ." F>S, S>F or FORGET BUF" CR ;
13 R->BASE
14 CLS CR ." Enter S>F for screens to DV80 file,"
15          CR ." or F>S for DV80 file to screens." CR CR ;S

```

The original version of this routine came from Michal Jaegermann of Edmonton, Alberta, Canada, and was only one screen long. It was written to work with the enhancements he has incorporated in his -FILE load option. The version shown above works with the standard load option and grew into 2 screens because some prompts have been added. This may be a step backwards (from Michal's point of view at least) but still an improvement considering the alternatives available.

One feature needs to be mentioned: once a file has been opened and one screen (or more) have been written to it, more screens can be appended to the same file. This is done by using <enter> instead of entering the filename again. For example, if you have already written screens 50 to 59 to a file named DSK2.SCREENS and now want to include screens 72 to 75 in the same file, do NOT enter the same file name again, just press the <enter> key. If you were to enter the same DSK2.SCREENS file name, it would result in a NEW file by that name being opened and your old file (containing 50 to 59) would be over written. This, however, is assuming that you have not done anything else meanwhile and the PAB for DSK2.SCREENS still exists.

Note also that the contents of each screen is written to the file consecutively, that is without any breaks. As you add your text to the DV80 file, you probably want to insert a couple of blank lines between the screens and (as I have done here) add line numbers.

F-Tidbit #9

The <BUILDS...DOES> Construct in TI-Forth

This tutorial is being written upon request by a TI-FORTH enthusiast who says that he has a problem figuring out <BUILDS...DOES>. It has been a long time since I concocted the last "tutorial", but I am not quite out of practice (having sporadically engaged in issuing "tidbits") and what follows is my humble attempt to oblige.

Somebody (I wish I could remember who) once wrote that "the most powerful programming tool you have in Forth is the ability to define new defining words." The problem lies in "separating the compile-time action of the defining word from the run-time action of the defined word." How true. Let us see if we can shed some light on the matter.

If you are familiar with the DEF statement in Extended BASIC we can use it as a starting point because <BUILDS...DOES> is somewhat akin to it. When you define a function in Extended BASIC nothing visible happens when the program is loaded. Yet the function has been placed in memory ready to perform if called upon. In other words, it has been compiled. However, as soon as the program runs and encounters the name of the function it will be executed as defined; the run-time action. This may be a bit of oversimplification but should give you an idea of what is meant by compile-time and run-time action.

<BUILDS...DOES> allows you to define defining words which is indeed a powerful tool. What it means is that the defining word, when invoked, creates a new word which does exactly what the defining word specifies. I am sure you have used defining words already without being aware of it: VARIABLE and CONSTANT are defining words. Since they are coded in assembly and part of the kernel you may not have realized that both use <BUILDS...DOES>. If they were written in high-level and listed somewhere on one of the screens, they would look like this:

```
: VARIABLE <BUILDS , DOES> ; ( n --- )
: CONSTANT <BUILDS , DOES> @ ; ( n --- )
```

As you can see, there is very little difference in their definitions at least in the compile-time action which is the part between <BUILDS and DOES>. Both store the parameter n in the first available memory cell after <BUILDS has created a new dictionary entry. Thus, no matter whether you use 5 VARIABLE FIVE or 6 CONSTANT SIX, each contains or represents a certain value. The significant difference lies in their run-time action. In VARIABLE there is no run-time action (nothing follows DOES), while CONSTANT performs a fetch during run-time. The result is that when you invoke FIVE only its address is left on the stack and you must use a @ or ? to retrieve or retrieve and display its current value. On the other hand, SIX also leaves its address on the stack but the @ (after DOES) uses it to fetch its value and puts it on the stack. This is precisely what you have known all along: You must fetch the value of a variable when you need it while a constant's value is put on the stack by invoking its name. Now let us see how we can separate the compile-time and run-time actions of a defining word:

```
def-word <BUILDS (compile-time action) DOES> (run-time action)
```

or (in terms of VARIABLE and CONSTANT) this is what happens:

```
5 VARIABLE FIVE
<BUILDS (create a dictionary entry named FIVE, store 5)
DOES> (no action, i.e., FIVE's address is left on the
stack and @ must be used)
```

6 CONSTANT SIX

```
<BUILDS (create a dictionary entry named SIX, store 6)
DOES> @ (fetch, i.e., 6 is left on the stack)
```

Now look at the definitions of the words 2VARIABLE and 2CONSTANT in Appendix C, page 5, of the TI-Forth manual (Notes on Starting Forth). These words are intended for double-precision (32 bit) numbers but you will find that they follow the same pattern with two exceptions: when you initialize a 2VARIABLE it automatically defaults to zero (the 0. following <BUILDS does that) and CONSTANT uses 2@ instead of @. In order to utilize double numbers you must, of course, use double number operators.

Are you beginning to see the light? Take a look at the welcome screen (#3). Here we find colon definitions galore to boot the load options like -EDITOR, -GRAPH, etc. They all perform the same task, i.e., boot something beginning with a specified screen number. Let us define a defining word and do away with those definitions. We know that the words being defined must contain a screen number and at run-time will have to do a LOAD starting from that screen. Let us call our defining word BOOTS:

```
: BOOTS <BUILDS , DOES> @ LOAD ; ( n --- )
```

The compile-time action consist of storing the beginning screen number n which is done by the comma after <BUILDS. (If you do not know about comma, look up its definition in the manual's glossary.) At run-time the screen number is fetched and LOAD is performed. Now BOOTS can take care of the load options if we use statements like those below instead of the colon definitions.

```
33 BOOTS -SYNONYMS (same as : -SYNONYMS 33 LOAD ; )
34 BOOTS -EDITOR (same as : -EDITOR 34 LOAD ; )
etc. etc.
```

By now you may have gotten the misleading impression that <BUILDS...DOES> can only store parameters. Do not be fooled, it can do anything a colon definition is capable of as the following examples will show. Here, for instance, is a defining word to initialize two-dimensional arrays:

```
: 2D-ARRAY ( #rows #columns --- )
<BUILDS 2DUP ( duplicate the parameters )
, , ( store them )
* ALLOT ( multiply them and allot the space )
DOES> ;
```

Note the 2DUP word which is not part of TI-Forth. You can substitute OVER OVER or add 2DUP (: 2DUP OVER OVER ;) to your dictionary before defining 2D-ARRAY. As you can see, the compile time action contains a little more than just a comma or two to store parameters: it computes the number of bytes to allot to accommodate the array. (If you expect to have numbers greater than 255 you would have to multiply by 2 again in order to allot space for 16-bit words.) While it is not imperative to store the dimensions of the array in the first memory cells as this word does, we will see shortly why this is a good idea.

Assume we want to use this array to store the scores (tests or golf or whatever) of four people over a period of 26 weeks, meaning that the array needs to be 4x26, and we will call it SCORES.

```
4 26 2D-ARRAY SCORES
```

creates the array. To make sure it contains the correct dimensions, we can check the first two memory locations with

```
SCORES ? 26 ok and SCORES 2+ ? 4 ok
```

To digress a bit from <BUILDS...DOES> for the moment, here is the reason for placing the array's dimension in its first two cells: they can be used to access any location within the array with the following word:

continued on page 16

XXB Documentation

by Barry Traver, PA USA

What is XXB? Well, just as XB is Extended BASIC, so XXB is Extended Extended BASIC. And just as Extended BASIC gives you 40 commands not found in TI BASIC, so XXB (Version 1.0) provides you with 25 commands not normally available in Extended BASIC.

Those commands, in alphabetical order, are as follows: ACCEPT, BYE, CHRSET, CLRTXT, COLOR, DSPLY, INIT, KEYR, MODE28, MODE40, NEW, NEWCLR, NEWSSET, NOQUIT, OKQUIT, PEEKC, PEEKV, POKEC, POKEV, READ, RELOAD, RETURN, RUN, TXTCOL, and WRITE.

Some of these new commands are very powerful (eg, ACCEPT, DSPLY, and INIT, generously contributed by J. Peter Hoddie) while others are of a more limited utility, but I think you will agree that the total offers up some tremendous facilities, especially for the creative Extended BASIC programmer.

Since XXB is disk-based, the good news is that XXB itself can be Extended. (The LOAD program on this disk is just one example of such a possible extension, since it includes Peter's useful ARRAY program along with the other 25 commands.) In fact, you can make your own XXB, using the approach set forth in my article on "Extended BASIC Assembly Language 1-2-3" in the January 1986 issue of Super 99 Monthly.

We will attempt to make our later versions of XXB, however, compatible with earlier ones, so that you will not have problems running the programs in GENIAL TRAVELER. Thus XXB (Version 1.0) includes a form of "RAW" (READ and WRITE) that is compatible with an earlier form of "RAW" that I made available. There is lots of room left, so we should be able to add some new commands to future versions of XXB without affecting present commands.

The 25 new commands can be placed in the following categories: (1) Disk Access, (2) Character Sets, (3) Text Mode, (4) Graphics Mode, (5) Peeks and Pokes, and (6) Miscellaneous.

Let us take them up, category by category.

(1) Disk Access

and WRITE ("RAW" stands for Read And Write) allow you to read or write a single sector on a disk. The format is as follows:

CALL LINK("READ",DRIVE;SECTOR,A\$,B\$), where DRIVE is the drive number, SECTOR is the sector number (remember: the first sector is sector zero!), and A\$ and B\$ are two 128-byte strings that contain the information from the sector.

The reason we need two strings is that one string just is not quite big enough: a sector contains 256 bytes of information, and the maximum length of a string is 255 bytes!

CALL LINK("WRITE",DRIVE,SECTOR,A\$,B\$) is similar, where DRIVE is drive number, SECTOR is sector number, and A\$ and B\$ are the two 128-byte strings of information to be written on the disk.

Many have put my "RAW" routines to good use, including John Johnson (with his PRBUTL, a very nice PRBASE utility) and Rich Mitchell (who knows as much about Microsoft Multiplan (tm) as anyone I know!). My Archiver program on this disk uses READ and WRITE routines.

(2) Character Sets

CALL LINK("SET") gives you a new character set, with large capital letters and true lower case.

CALL LINK("CHRSET"), when used along with CALL CHARSET, restores the old character set. (CALL CHARSET in Extended BASIC does not restore the lower case letters nor clear character sets 13 and 14, so CALL LINK("CHRSET") makes that simple to do. This can be important if you do any program chaining, which does not automatically restore character definitions!)

(3) Text mode

Text (40-column) mode is not normally available in Extended BASIC, unless you have, for example, Oak Tree Systems' "DEP" (Display Enhancement Package), a useful utility which is a sort of XXB itself. Our XXB gives

you some features, however, which are not available in the DEP, especially thanks to J. Peter Hoddie, who gave us some really nice routines: ACCEPT, DSPLY, and INIT.

CALL LINK("MODE40") puts you into text mode. Warning: text mode is an area in which the Extended BASIC programmer must walk with care. Two things to remember: (1) You must use CALL LINK("MODE28") in order to return safely to graphics mode, and (2) any kind of error in text mode can throw you into never-never land!

Peter's CALL LINK("INIT",LEFTMARG, RIGHTMARG,MODE) lets you set your left and right margins as well as the mode. For text mode, MODE must be 40, but some creative use of LEFTMARG and RIGHTMARG can allow some interesting possibilities for ACCEPT and DSPLY.

CALL LINK("TXTCOL",F,B) sets the foreground and background colors for text mode.

Peter's CALL LINK("DSPLY",R,C,M\$) is very similar in operation to Extended BASIC's DISPLAY AT, but DISPLAY AT works in a meaningful way only in normal graphics mode, while Peter's CALL LINK("DSPLY",R,C,M\$) can be used in either graphics mode or text mode, provided that CALL LINK("INIT",L,R,M) has been properly set.

His CALL LINK("ACCEPT",R,C,L,V\$,M\$) is an amazing facility, because it offers something not even available in Myarc's Extended BASIC IV (or whatever they call it): a multi-line ACCEPT AT! And you can set the Length and the Validation string (features we are used to from Extended BASIC), as well as suggest a default Message.

See the file called ACCEPT/S on this disk for more comments on ACCEPT, DSPLY, and INIT, which work not only in text mode, but also in graphics mode.

Many CALLs accessed while in text mode create a screen "glitch," so CALL LINK("KEYR",R\$) is provided to avoid that problem. (One irritation that comes with working with text mode from Extended BASIC is that apparently all CALLs to user-defined subprograms will cause such glitches.)

CALL LINK("CLRTXT") will clear the screen in text mode. It operates in a similar way to CALL CLEAR in graphics mode.

Finally, CALL LINK("MODE28") takes you safely back to graphics mode in Extended BASIC.

(4) Graphics mode

Peter Hoddie's CALL LINK("INIT",LEFTMARG,RIGHTMARG,MODE) also has its uses in graphics mode, where the mode should normally be set to 32 (although 31 and 33 are also useful, see the STRINGDEMO program on this disk).

His CALL LINK("DSPLY",R,C,M\$) and CALL LINK("ACCEPT",R,C,L,V\$,M\$) grant us some special uses also in graphics mode (see the BOX/DEMO program for one example).

My CALL LINK("KEYR",R\$) is a good replacement for the following common code:

```
100 CALL KEY(O,K,S) :: IF S=0 THEN 100 ELSE R$=CHR$(K)
```

Thus it has some use in graphics mode as well as in text mode, for which it was originally written.

(5) Peeks and Pokes

My PEEKs and POKEs are different from usual in that they deal entirely with strings rather than values, but this is much more memory-efficient for Extended BASIC.

CALL LINK("POKEC",L,S\$) and CALL LINK("POKEV",L,S\$) poke strings into CPU RAM and VDP RAM respectively.

Likewise, CALL LINK("PEEKV",L,S,S\$) and CALL LINK("PEEKV",L,S,S\$) are able to peek into a certain location for a String of designated Size, again for either CPU RAM or VDP RAM.

So far as VDP RAM is concerned, there is no compensation for the >60 (96 decimal) offset for the screen, but that should not be a handicap, since we have other utilities to deal with the screen (e.g., ACCEPT, DSPLY, etc.).

(6) Miscellaneous

CALL LINK("NOQUIT") disables the "Quit" key, while CALL LINK("OKQUIT"), a useful feature while working in Extended BASIC.

continued on page 33

Problems with Printers

by Jim Peterson, Tigercub Software, USA

When a program is written on one TI99/4A console, it is a pretty sure bet that it will run on any other TI99/4A console, unless the programmer has used some of the special features of the CorComp Disk Controller, Super Extended BASIC, or whatever. But when a programmer writes a program to output to his own printer, it is by no means certain that it will work with your printer. As far as printer compatibility is concerned, it is a jungle out there. Anarchy, chaos and total confusion!

To begin with, if the printer has a parallel port it must be opened with "PIO", otherwise with "RS232" followed by the baud rate, or something else again for AXIOM. And you may have to add .LF to suppress line feeds or .CR to suppress carriage returns. Next, its output and its response to control codes is partly controlled by those idiotic, microscopic, inaccessible, fragile-looking inventions of the devil called dip switches. And finally, the output is mostly controlled by the printer control codes in the program itself.

Somewhere among the thousands of publications on computers, someone must have written a comprehensive guide to writing and modifying software for printer compatibility. If anyone knows of such, please tell me! I have read literally thousands of user group newsletters over the past several years, and have seen many mentions of "fixes" to various problems, but never a detailed article. I have called printer manufacturers, and they have been most helpful in suggesting that I buy one of their expensive manuals for each of their models. I have talked to programmers with much more experience in writing printer programs than I have, and they tell me it is very difficult, even with the manual at hand, to modify a program for a particular printer without having access to that printer for testing.

I have no experience in programming for any printer other than my trusty old Gemini 10X, and my few attempts to modify programs for other printers have mostly ended in failure. However, I have borrowed several manuals and attempted to chart the differences. I had hoped to compile and publish a complete conversion chart, until I realized the complexity of the problem. Anyway, perhaps I can pass on a few tips to programmers, to help them make their programs as widely compatible as possible, and possibly I can give users a little bit of guidance to help in modifying programs to suit their printer.

In the following, in order to be brief, I have mentioned control code sequences by their ASCII numbers, such as 27 66 1. This would be programmed as CHR\$(27);CHR\$(66);CHR\$(1) or, since ASCII 66 is within the printable range, it might be CHR\$(27)&"B"&CHR\$(1). 27 77 n means that for n you substitute an ASCII code, within an allowable range, according to what you want to accomplish.

There seem to have been four systems of printer control codes used with the printers commonly found in the TI99/4A world; Epson, Micronics, Axiom, and Okidata. The Micronics people tell me that they "used the Micronics emulation until the introduction of the current NX series, when they switched to the IBM emulation". The IBM emulation appears to be the same as the Epson mode except that it has a different set of special character symbols. In fact, many current Epson compatibles have an optional IBM mode.

The Micronics mode and the Epson mode are quite similar, although with aggravating differences. Okidata and Axiom are way out in left field. Since Micronics, Epson and Panasonic (which is basically Epson compatible) seem to be by far the most popular in the TI99/4A community, and most software is written for them, it might be wise to avoid the Okidata. I have also seen mention of problems with Diablo and Centronics, but I have no information on those.

Any of the ASCII codes from 0 to 127 can be used as a printer control code. If the ASCII code is above 31, it must be preceded by ASCII 27, known as the escape code, which is universally used to alert the printer that the following ASCII codes are to be interpreted as controls rather than printed as characters. If the printer recognizes an ASCII code below 31, or one or more ASCII code immediately following ASCII 27, as a valid control code, it acts upon them but does not print them. This is why, if you insert "control U" codes in a line of text, the text will be shifted left. However, if the codes are not recognized as valid, the ASCII codes below 32 or above 126 are printed as a blank space, the others are printed as the character they represent. This is why that puzzling E, G, S or whatever shows up on the first line of a printout, if a program is not compatible with your printer.

Some printer commands require a sequence of three or more ASCII codes, of which the first is 27, the second could be anything above 31 and the remainder could be anything at all. If your printer does not recognize the second ASCII code as valid, but then comes to an ASCII code below 31 which it does recognize, it acts on that ASCII code as if it was a single command, which is why your printer sometimes "goes crazy". The ASCII codes below 27 are quite standardized, and many of them have names, such as BEL for 7 (activates buzzer) which are also commonly used in telecommunications. ASCII 10 (line feed), 12 (advance to next top of form) and 13 (carriage return) seem to be universally recognized. For some reason, Panasonic owners seem to have trouble with line feeds when running programs written for other printers.

ASCII codes below 27 are not preceded by the escape code 27. Some printers will optionally recognize 14 (double width for one line) and 15 (cancel 14) preceded by 27, but programmers should avoid this since other printers will treat the 27 as a blank space. The NX-10 recognizes 27 10 as a command to reverse the paper one line and 27 12 to reverse to top of page.

The escape code 27 can be input from the TI99/4A keyboard by depressing the CTRL key and the period key together. The actual ASCII code is 155 but printers, other than the Axiom, will accept it as 27. This is handy when opening the printer in immediate mode or writing a routine for your own use, but should be avoided in programs being distributed because the character prints out as a blank space which will probably confuse anyone trying to modify the program.

I have studied the manuals, and attempted to chart some of the codes, for the Gemini 10X, SG-10 and Star NX-10; Epson FX-80, FX-85/185 and its IBM mode; Panasonic KX-P1080; MX Graftrax Plus; Brother M-1009; Seikosha 550A or 550TI by Axiom; and Okidata (model unknown). Due to differences in terminology, it is not easy to relate them to each other.

The IBM mode of the Epson FX-85 seems to be entirely compatible with its Epson mode except that it lacks some features. The FX-80 seems to be entirely compatible with the FX-85 except lacking NLQ and a few specialized codes (and I did not get into comparing graphics capabilities of any of these printers). The Brother M-1009 is also apparently highly compatible. The MX Graftrax Plus, another Epson model, is entirely compatible but lacks some features (no graphics capability?). The Panasonic KX-P1080 is very compatible but also has several unique codes of its own for setting tabs and spacing, etc. The Star NX-10 also seems to be in complete agreement with the FX-85, but with some additional codes for unique features such as reversed paper feed. I would guess that in actual practice these may not be as compatible as they seem. And of course, any maker's newer or more expensive models have additional codes for features not found on older or cheaper models.

The Seikosha 550A or 550TI, made by Axiom, recognizes a few of the common ASCII codes between 7 and 14, but from there on it has entirely its own system of codes. Many of these same codes are used by

Epson/Micronics for entirely different purposes, so running a program written for either on the opposite printer can be guaranteed to produce pure garbage. Okidata likewise recognizes a few of the low codes and then goes into its own system, frequently in direct conflict with the Epson standard. Due to the terminology used in its manual, I am not sure what some of them do.

The Gemini 10X, made by Star Micronics, has long been superseded by newer models, but stocks are still being sold by discounters. It has been a sturdy workhorse, long popular with TI99/4A owners, and a great many programs have been written using its printer control codes. These are 90% compatible with Epson, but that other 10% causes a great deal of trouble. The differences are described below. The Star SG-10/15 was a transitional hybrid, switchable by dip switch 2-2 between the Micronics mode and the IBM mode. The Micronics mode is completely compatible with the Gemini 10X (except for download characters) and with some additional features, NLQ and proportional printing, and a slashed zero option. The IBM mode seems to be very compatible with the Epson standard. This printer was superseded by the Star NX-10, which is again Epson compatible.

A major incompatibility between programs written for the Gemini 10X or SG-10, and Epson-compatible printers, is that Micronics recognizes 27 66 1 to select pica, 27 66 2 to select elite and 27 66 3 for compressed, and on the SG-10 also 27 66 4 to select NLQ and 27 66 5 to cancel it. On Epson/IBM printers, 27 66 is the beginning of a series of codes used to set vertical tabs. Actually, since pica is the default, there is no need to program for it except to cancel condensed print, for which purpose 18 is recognized by both Micronics and Epson. (Avoid using 27 80 to return to pica because Micronics does not understand it and might misinterpret it to change default tabs.) Similarly, 15 will select condensed print on both the Micronics and Epson. Unfortunately, there is no compatible code for elite; Epsons use 27 77 to select elite, but 27 77 n is used by the 10X, SG-10 to set the left margin n spaces, so that misinterpreting these codes can be catastrophic! The Epsons use 27 108 n to set the left margin, but this is not recognized by Micronics.

The other major difference is 27 51 n which sets the line feed to n/144" on the 10X and on the SG-10 in Star mode, but to n/216" on Epson compatibles and on the SG-10 in IBM mode. The "fix" here is to multiply the value of n by 1.5 when running a Micronics program on an Epson printer. The same applies to 27 74 n which sets a one-time line feed of n/144" or n/216". Micronics uses 27 82 n to set the margin at the top of the page, but Epson recognizes this as a command to switch to one of the international character sets, which can produce some interesting results. The Epsons use 27 114 n to set the top margin, but Micronics does not know this one. Micronics uses 27 55 n to select an international character set, but the Epson will read 27 55 as a command to cancel 27 54 which selected a special character set. I am not sure what that means, but the results will surely be undesirable.

There may also be a conflict between the Micronics 27 98 n, which performs a one-time tab of n columns, and the Epson 27 998 n n 0, which "sets vertical tabs in channel", whatever that means.

Several codes, common to both Epson and Micronics, use 1 as the 3rd ASCII code to turn on a feature and 0 to cancel it. For instance, 27 87 1 turns on double width (expanded) print and 27 87 0 cancels it. Also, subscript is selected by 27 83 1 and superscript by 27 83 0. Some of the Epson/IBM compatibles will accept either an ASCII code or numeric 0 or 1 (i.e., "1" or CHR\$(1)) for that third code, for which reason you will often see program coding such as CHR\$(27);"W1". These should be avoided when programming for general distribution, because the older Micronics recognize only the ASCII code. If I understand my notes from the

manual correctly, the Panasonic KX-P1080 also accept ASCII 129 or 177 in lieu of ASCII 49 or "1" and ASCII 128 or 176 in lieu of 48 for "0"!

According to the manuals, ASCII 141 can be substituted for ASCII 13 on the Brother M-1009, and ASCII codes 128 to 255 can be substituted for 0 to 127, respectively, on the Epson, but there seems to be no good reason to confuse the other printers by using those!

Different printers also have different sets of symbols in ASCII codes 160 to 254. The Gemini 10X and SG-10 in Star mode has one set, the SG-10 in IBM mode has an entirely set which I presume is also on the Epson in IBM mode, and I think that the Epson has still a different set. This causes problems when running some banner or graphing programs which access these characters. Different printers also vary in the number of international character sets available and the sequence of their access codes.

I have never become involved in graphics printing, and I failed to chart all the graphics codes when I had borrowed manuals available, so I cannot comment on compatibility here. I have not heard of any problems except that some Axiom models are apparently incapable of graphics, and there is also sometimes a problem with thin white horizontal lines through the picture; possibly because of the n/144" and n/216" difference in line spacing between Micronics and Epson?

With downloadable characters, we find another jungle which I am not too anxious to explore. The Gemini 10X has a quite simple and efficient method, and I once published in MICROpendium a DOWNCHAR program to design these characters on screen, dump them to the printer for editing, and save them to disk. I have also written a routine which will convert a sequence of any length of standard or reidentified screen characters into a D/V 80 file of download character printer codes.

But, here the SG-10 Star mode departs from compatibility with the 10X. Its system offers much greater capabilities but is also quite complex and entirely different. I tried, and failed, to convert my routine for use on the Epson FX-85; its system is somewhat similar to that of the SG-10 but again different. I am told that the Epson RX-80 does not support downloadable characters, the LX-80 only allows 6 and some Panasonics only allow 40 of them. I have seen an article describing a method of creating downloadable NLQ characters but unfortunately the name of the printer being used was not mentioned.

This article is obviously incomplete and probably inaccurate. Perhaps it will inspire someone to write something better. In the meantime, programmers could help out a great deal by putting REMS in their programs giving the name of the printer they are writing for, and REMS after every printer control command indicating its purpose. I regret that I have not been in the habit of doing that!



At the Assembler Tutorial Weekend (April 1989)

Jenny's Younger Set

Dear Jenny,

This is Crocodile Jones. I have received another letter.

Dear Crocodile Jones

I am stuck with the pirate on Treasure Island in adventure number 2. How do I get the pirate on the ship?

Christopher Maker

Dear Christopher,

Take your shovel with you when you go on the ship with the pirate and dig on the beach. Then wake the pirate in the grave yard.

Crocodile Jones

Dear Jenny,

Here is a program quiz on the Peloponnesian war. I hope you like it and others enjoy using it.

Vincent Maker

```

100 RIGHT=0
110 WRONG=0
120 REM PELOP WAR QUIZ IN XB
130 REM BY VINCENT MAKER.
140 CALL CLEAR
150 DISPLAY AT(5,1):" PELOP WAR QUIZ

                                BY
V. MAKER."
160 FOR E=0 TO 250
170 NEXT E
180 CALL CLEAR
190 DISPLAY AT(5,1):"WHO LEAD THE FIRST INVASION OF
ATTICA?"
200 DISPLAY AT(7,1):"A,KING ARCHIDAMUS
                                B)PERICLES                C)KING
LEONIDAS                D)IT IS NOT KNOWN"
210 DISPLAY AT(24,1):"PRESS THE CORRECT LETTER."
220 CALL KEY(O,J,K)
230 IF K=0 THEN 220
240 IF J=05 THEN RIGHT=1 ELSE WRONG=1
250 CALL CLEAR
260 DISPLAY AT(3,1):"WHICH IMPORTANT ATHENIAN        DIED
DURING                THE GREAT PLAGUE?"
270 DISPLAY AT(7,1):"A,KING ARCHIDAMUS
                                B)PAUSANIUS                C)PERICLES
D)CLEON"
280 DISPLAY AT(24,1):"PRESS THE APPROPRIATE LETTER"
290 CALL KEY(O,H,T)
300 IF T=0 THEN 290
310 IF H=07 THEN RIGHT=RIGHT+1 ELSE WRONG=WRONG+1
320 CALL CLEAR
330 DISPLAY AT(5,1):"WHO ORGANISED A FIFTY YEAR PEACE
LATE IN THE WAR?"
340 DISPLAY AT(5,1):"A,CLEON
                                B)BRASIDAS
C)ALIBIDIES
D)NICIAS"
350 DISPLAY AT(24,2):"PRESS THE RIGHT LETTER."
360 CALL KEY(O,F,G)
370 IF G=0 THEN 360
380 IF F=06 THEN RIGHT=1+RIGHT ELSE WRONG=WRONG+1
390 CALL CLEAR
400 DISPLAY AT(3,1):"WHICH OF ATHEN'S ALLIES
REVOLTED AFTER THE CAPTURE OF PLATAEA?"
410 DISPLAY AT(5,1):"A)HEGARA
                                B)MYTILENE
C)SPARTA                D)NORE"
420 DISPLAY AT(24,1):"WHICH ONE IS IT?"
430 CALL KEY(O,E,O)
440 IF O=0 THEN 430
450 IF E=00 THEN RIGHT=RIGHT+1 ELSE WRONG=WRONG+1
460 IF RIGHT=0 THEN A$="DO NOT TAKE THIS UP FOR A
LIVING. NONE OUT OF FOUR."
470 IF RIGHT=1 THEN A$="NOT ALL THAT GOOD. ONE OUT OF
FOUR."
480 IF RIGHT=2 THEN A$="FIFTY-FIFTY. TWO OUT OF FOUR"
490 IF RIGHT=3 THEN A$="NOT BAD. THREE OUT OF FOUR."
500 IF RIGHT=4 THEN A$="PRIMTY GOOD. FOUR OUT FOUR."

```

```

510 PERCENT=100*RIGHT/4
520 CALL CLEAR
530 PRINT "YOU SCORED ";RIGHT;" OUT OF FOUR. THAT IS
";PERCENT;"%."
540 DISPLAY AT(5,1):A$
550 END

```

continued from page 30

CALL LINK("NEW") prepares the way for a New program, without clearing the screen, while CALL LINK("NEWCLR") prepares the way for a New program and clears the screen.

The next three CALL LINKs came as a result of information provided by Craig Miller in the excellent manual that accompanies Night Mission.

A CALL LINK("RUN") re-runs the program, but without the pre-scan that normally takes place when that is done. CALL LINK("RELOAD") really is equivalent to RUN "DSK1.LOAD". And CALL LINK("BYE") closes all files and returns on to the title screen.

CALL LINK("RETURN") can be used as a test to see whether XXB has been loaded into memory.

XXBSUPPORT

One disadvantage of accessing assembly language routines from Extended BASIC is the necessity of repeatedly typing in CALL LINK. Well, if one is prepared to sacrifice some of the speed of XXB, that can be avoided by MERGEing in XXBSUPPORT with your Extended BASIC program.

The result is that no longer will you have to do a CALL LINK("BYE"): a CALL BYE will be sufficient. Likewise a CALL LINK("DSPLY",R,C,M\$) can now be done with a simple CALL DSPLY(R,C,M\$). And so on.

Be sure to read the REMs in the XXBSUPPORT file. One fact that is noted is that this is not a useful approach to use in text (40-column) mode (because of the screen "glitches" caused, as noted elsewhere), but it is passable elsewhere (again, with some sacrifice in speed).

XXBSUBPROG

The XXBSUBPROG file provides some additional support of a different kind. The new commands in XXB can be combined with user-defined subprograms in Extended BASIC, to allow yet more new facilities.

One example is the ability to display a string either vertically or diagonally with ease and speed, though this is ordinarily somewhat laborious in Extended BASIC. (See STRINGDEMO on the disk.) Another is the ability to put up a rectangular graphic display quickly at any location. (BOX/DEMO is an example of this.)

Do not miss two subprograms that make good use of the PEEKC and POKEC routines. SUB SAVE_SCREEN(A\$()) and SUB PUT_SCREEN(A\$()) allow you to save an entire screen into four 192-byte strings. Those strings can be put on the screen again with ease. No demonstration is provided, so it is up to you to try out these XXBSUBPROG on your own.

Once you have saved the screen into four strings, you can do anything you want with them, eg, manipulate them or store them on disk. The strings, however, contain no information on character redefinitions (or sprites, for that matter).

XXBSUBPROG is just the start of a library of useful subprograms which put to work the 25 new commands in the present version of XXB. With your help, we can expand this library (as well as add new assembler language subroutines to XXB)!

I hope you enjoy XXB as much as I enjoyed putting it together for you!

For Sale

Peripheral expansion Box with full height single sided single density disk drive (90K) in working condition (with some marks on the PEB) for \$200. Phone (042)84 2960.

continued from page 1

Is there anyone who would like access to a system with hard disk and colour monitor with 80 column card? All you have to do in return for the use of such a system is to do a bit of editing each month and produce the TND. Do not be shy or backward! There must be someone itching to try out such a system and eager to read all the information that comes to the Editor first. There is an enormous amount of information about waiting to be edited. We have increased the number of pages we print each month and still the amount of material to be printed increases. One could also argue that some of the starting articles for beginners should be reprinted for all those of us who are always beginning anew. Seriously, if someone would like to try the Editor's job I would be very happy to pass it on and spend more time on hardware and even writing learned articles on hardware and software.

Lou had a phone call from someone from North Queensland having problems with his CorComp mini disk system not allowing speech from Terminal Emulator II. We were about to try out Rolf's similar system to see if it also had problems and it died in front of our regional group meeting. We did see it work once. What I actually wanted to say was that we should send to those isolated members the names and addresses of their nearest member as they may want to have someone to talk to at less cost than a call to Wollongong. Perhaps if members would like to be in something like that they could drop a line to our hard working secretary and he can arrange it.

Newsletter Update

with Lou Amadio

Local Newsletters

ATIOC August 1989: Fred Cugley laments on the total lack of support from Myarc on his Geneve and HFDC problems, Jim Derk on printer interfacing, very handy TI99/4A Error Code Reference Chart, a Handy Dandy Reference Chart and Transliteration And TI-Writer.

Bug Bytes, September 1989: Apart from the wrong date (ie 1898), Bug Bytes this month informs us that Press is close to being completed, hard disk software (Hardmaster, Back-up and Emulate) has been released by the Christensens, other software includes Lotto-gold and Tapemaster (this program is used to copy memory image files from disk to cassette), advice on how to patch errors in the Pools program, technicalities of cassette data storage, Graphics by J.Peterson, a hard disk "DSK1" file emulation selector, hard and floppy drive configurations, explanation of a "Nibble", BASIC program "Easy Sorter" and finally a user group questionnaire.

Hunter Valley 99ers, August 1989: New committee elected with Peter Smith at the helm, Tony McGovern on assembly programme contest to code a routine to replace the Squish code in TI-Writer, Struggling Forth with Richard Terry on how to expand on a basic idea and start programme, Joe Wright reviews a Forth programme, an excellent Extended BASIC tutorial by Tony McGovern, Alan Franks has a source of Mitsubishi drives for \$60 each,

Hunter Valley 99ers, September 1989: Editorial by Peter Smith on how to attract new members, new technical manual to be released by Tony Lewis (USA), Parsec level 3 hint, FCTN[J] plus SPACE emulates FCTN[4], Struggling Forth by Richard Terry, Bob Carmony talks of Amerisoft Extended BASIC extensions and a number of CALL LOADS which can be used in the immediate or program modes and a BASIC "Weird Sounds", Extended BASIC tutorial by Tony McGovern and finally user information on the Myarc HFDC card.

Overseas Newsletters

CIM 99, Sept 89: Zork II adventure game schematic.

Lehigh 99ers, Spring 1989: Review of Form Shop editor produces text and graphics using the keyboard,

review of Crossword Composer and TI-Artist for beginners Vol 2. Summer 1989: 40_UTIL allows 40 column mode with Extended BASIC, Page Pro Quick Reference Sheet, Assembly Language by Adrian Robson on Boot Drive tracking.

Northern NJ UG, September 1989: availability of John Birdwell's disk Utilities V4.0a, how to disable the "Out of Paper" function on printer so as to address envelopes (ESC 8), GEME (GEM) look alike operating environment, for the Geneve and a "Remind Me" memory data base program.

Ottawa UG, June 1989: UG to set up a TI99/4A system at a local mall in order to attract new members, release of *Music-Pro* by Ottawa UG, good article on "Teaching PRBase" by Bill Sponchia, "A Look at Assembly Language Subroutines", GRAMULATOR report, more on "Fast Extended BASIC" and TI BASIC control codes by Steven Shaw.

ROM Orange County CA, August 1989: Jim Swedlow on Talking to Your Printer with TI-Writer, an insight into Calendar Programs, Multiplan Demonstration provides a quick step through this very versatile spreadsheet programme.

Sacramento 99er UG, May to August 1989: setting up a Geneve, how to cool a Geneve with an auxiliary fan and a FORTRAN tutorial.

Spirit of 99, September 1989: New products include F-DOS, <G> graphics, Tips for greeting cards, Pix for colour two tone pictures, Genprog for the Geneve, Auditor Accounting and Music-Pro, The Problem With Printers by Jim Peterson, a BASIC fortune teller programme, a quick way into level 3 of Parsec, review of The Maze of Grog, TI-Sort features windows, Page Pro 99 for hard disks, Columntext and Margintext for DVSO files, MDOS 0.95H, The Debugger outputs to RS232 to avoid corrupting the programme being debugged, programme to allow TI-Artist to load without looking to all the drives for each file and TIBase tutorials 7.1 and 7.2 by Martin Smoley.

The Tacoma Informer, June/July 1989: Overview of Genial Traveler Diskazines (disk magazines), transliterate and TI-Writer.

August 1989: Working With Numbers in BASIC by Joe Nollan, TI International Expo 89 held in September, Orphans by Steve Mickelson is a short history of the TI99/4A including what we can expect in the near future such as MIDI interface, Z80 card, IBM co-processor, kermit software, GIF software and a stand alone Geneve.

TIdbits (Mid South) UG, September 1989: Club offer to pay half the price of software for a newsletter review, "Giant Artist Posters" program, Zenoboard (multifunction PCB for the console) are available for \$US18.50, Legends II now available, Barry Boone offering EXEC for the Geneve to load assembly from MDOS without GPL interpreter, TIBase V2.02 and TI-Sort from Inscebot, AV-Indexer for cassettes and VCR labels, Harrison Word Processor is menu driven, how to add a 4th drive amongst other things to your TI DCC system, history of console power supplies and a remote console power supply saves on heat build up.

For Sale

Some Peter Schubert designed multi-function printed circuit boards are now available from the shop for \$30.

A multi-tap transformer suitable for the Lou Amadio direct I/O interface project is available for \$20.

A full kit of components to make up the multi-function controller card will soon be available for approximately \$100.

Enquire at the shop for exact prices and availability.

Regional Group Reports

Meeting summary.

| | | |
|------------------|----------|-------------|
| Banana Coast | 12/11/89 | Sawtell |
| Carlingford | 15/11/89 | Carlingford |
| Central Coast | 11/11/89 | Toukley |
| Glebe | 9/11/89 | Glebe |
| Illawarra | 20/11/89 | Keiraville |
| Liverpool | 10/11/89 | |
| Northern Suburbs | 23/11/89 | |
| Sutherland | 17/11/89 | 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.

CARLINGFORD Regional Group.

Regular meetings are normally on the third Wednesday of each month at 7.30pm. Contact Chris Buttner, 79 Jenkins Rd, Carlingford, (02)871 7753, for more information.

CENTRAL COAST Regional Group.

Regular meetings are normally held on the second Saturday of each month, 6.30pm at the Toukley Tennis Club hall, Header St, Toukley. Contact Russell Welham (043)92 4000

GLEBE Regional Group.

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

ILLAWARRA Regional Group.

Regular meetings are normally on the third Monday of each month, except January, at 7.30pm, Keiraville Public School, Gipps Rd, Keiraville, opposite the Keiraville shopping centre. 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)644 7377 (home) or (02)708 5916 (work) for more information.

NORTHERN SUBURBS Regional Group.

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

Come and join in our fun. Dick Warburton.

SUTHERLAND Regional Group.

Regular meetings are held on the third Friday of each month at the home of Peter Young, 51 Jannali Avenue, Jannali at 7.30pm. Group co-ordinator is Peter Young, (02) 528 8775. BBS Contact is Gary Wilson, user name VK2YGW on this BBS.

The September meeting saw the return of Ron Stewart from his recent visit to the U.S.A. Not a whole lot to report in the way of new software, as the TISHUG membership is kept well informed of goings-on overseas through the Newsletter Update and direct contact with the Club's Executive. Ron did "test drive" his new HFDC card, without the benefit of a hard drive at this stage. Seems like there is always something new, just around the corner.

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 Woodstock Community Centre, Church street, Burwood.

Regular items include news from the directors, the publications library, the shop, and demonstrations of monthly software.

November 4 - Full day workshop. Starts at 10am. A collection of tutorials, program workshops and demonstrations that will be of interest to any member, whether a programmer, a program user or a hardware hacker. Mark this date on your calendar.

December 2 - Christmas Party

Craig Sheehan (Meeting coordinator).

For Sale

TI99/4A computer with all documentation, power supply and modulator and a brand new American (NTSC) TI99/4A which needs conversion.

Peter Schubert's 32K memory expansion with RS232.

Speech synthesizer, original joysticks, joystick adapter and widget board.

19 game cartridges with documentation, 140 tapes and 3 program books.

\$400 for the lot. Phone (043)59 1532.

continued from page 14

```
730 SUB FILENAME(R,C,M$,D$)
```

```
740 DISPLAY AT(R+1,C):RPT$("-" ,LEN(M$)):
```

```
    DISPLAY AT(R,C):M$ :: IF D$<>"N?" THEN DISPLAY
```

```
AT(R+2,C):D$ ELSE SUBEXIT
```

```
750 ACCEPT AT(R+2,C)SIZE(-15)BEEP:D$ :: SUBEND
```

```
760 SUB MORE(NM):: DISPLAY ERASE ALL ::
```

```
    CALL TXTCOL(3,12):: CALL YN("More
```

```
listings","N",16,2,NM):: SUBEND
```

```
770 SUB DELAY(A):: FOR A=1 TO A :: NEXT A :: SUBEND
```

```
780 SUB TXTCOL(A,B):: CALL SCREEN(B):: FOR I=0 TO 12 ::
```

```
    CALL COLOR(I,A,B):: NEXT I :: SUBEND
```

The FILENAME routine writes an underlined heading, DISPLAYs the default response, and ACCEPTs the reply. If it is asked no question, "N?", it expects no answer. The other SUBs just do their job when called. YN acts like input routines familiar in other TI modules.

```
790 SUB SPEAK(A$):: CALL PEEK(-28672,SP):: IF SP=96 THEN
```

```
    CALL SAY(A$) ELSE CALL DELAY(5*LEN(A$))
```

```
800 SUBEND
```

This is a last little goodie tagged on so that you may add speech prompts to your program where desired. A bald CALL SAY has the annoying behaviour that it seems to take forever in giving up the attempt if no speech synthesizer is attached. Line 800 checks that speech is connected and line 820 substitutes a controlled delay if not. CALL SPEAK("...") can then be inserted anywhere it is wanted in the program.

So there we have it, a worked out example of a non-trivial and useful program that makes essential use of the sub-program facility of Extended BASIC. It shows that the Extended BASIC programmer can, with a style that finds natural expression in the language without undue contortions, follow the general principles of "structured programming" without getting hung up in the Swiss straight jacket so beloved by some proponents. The program as presented is a cut down version of the all singing, all dancing model, COLIST, which has now grown to >22K and uses 48 sub-programs. In all the versions, sub-programs have been an essential tool for program development. Now it is time to take retrospective look at what at what we have done and chase a few more subtleties.