TISHUG NEWS DIGEST

Focusing on the TI99/4A Home Computer

Volume 10, Number 11 December, 1991

Registered by Australia Post - Publication No. NBH5933

Season's Greetings

Sydney, New South Wales, Australia $3

We have changed our postal address. From now on please use PO Box 1089, Strawberry Hills NSW 2012.

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

TIsHUG News Digest

December 1991

All correspondence to:

P.O. Box 1089
Strawberry Hills, NSW 2012
Australia

The Board

Co-ordinator	
Dick Warburton	(02) 918 8132
Secretary	
Terry Phillips	(02) 797 6313
Treasurer	
Geoff Trott	(042) 29 6629
Directors	
Rolf Schreiber	(042) 84 2980
Russell Welham	(043) 92 4000

Sub-committees

News Digest Editor	
Bob Relyea	(046) 57 1253
BBS Sysop	
Ross Mudie	(02) 456 2122
BBS telephone number	(02) 456 4606
Merchandising	
Percy Harrison	(02) 808 3181
Publications Library	
Warren Welham	(043) 92 4000
Software library	
Rolf Schreiber	(042) 84 2980
Technical co-ordinator	
Lou Amadio	(042) 28 4906

Regional Group Contacts

Central Coast	
Russell Welham	(043) 92 4000
Coffs Harbour	
Kevin Cox	(066) 53 2649
Glebe	
Mike Slattery	(02) 692 8162
Hunter Valley	
Geoff Phillips	(049) 42 8176
Illawarra	
Lou Amadio	(042) 28 4906
Liverpool	
Larry Saunders	(02) 644 7377
Northern Suburbs	
Dennis Norman	(02) 452 3920
Sutherland	
Peter Young	(02) 528 8775

Membership and Subscriptions

Annual Family Dues	$35.00
Associate membership	$10.00
Overseas Airmail Dues	A$65.00
Overseas Surface Mail Dues	A$50.00

TIsHUG Sydney Meeting

The Annual General Meeting will start at 2.00 pm on 7th of December at Ryde Infant School, Tucker Street, Ryde. After the meeting there will be a Christmas party for all members and their families.

Printed by
The University of Wollongong
Printery Services.

TIsHUG News Digest ISSN 0819-1984

Index

Lost at the Tutorial Day 2nd November

Peter Mudie lost his digital multimeter at the tutorial day. The multimeter was being used where Peter was fixing RAMdisks. If anyone has picked the meter up by mistake or knows where it is, please call Peter on (02)456 2122.

Editor's Comment

by Bob Relyea

Helping other users has always been a strong point with our club. I was encouraged to see a number of users present at the last meeting encouraging and offering help to several other members present who were having difficulties. This was especially true of a user down from New Castle way, Keith, I think his name was who is not able to attend meetings very often since moving up there a while back. He came down with a load of questions and went back with most of them answered and a happy man indeed. That is what we like to see. It is not the type of help that would be readily available with other computers and we offer it at no cost! In the process of doing my bit with Multiplan and Word Processing, another user came by as I was going through a few things with TI-Artist and he helped me out heaps with a couple of quick demonstrations. I would not mind seeing him do a tutorial at the next Tutorial Day.

Thank you to all who contributed articles this year. We have a good magazine worth keeping and admired by users around the world. Let us face it, it is what keeps us together! As part of the editorial staff, I am looking forward to next year, the year of the Fair, working with you and anticipating some nice articles and a continuation of the TI spirit. o

TI Fair, a Few Thoughts

by Garry Christensen, TI-BUG Brisbane

When I last contacted Rolf Schreiber I was able to pass on to him my support for a TI Fair that TIsHUG has proposed. It is my belief that the time has come for another Fair in Australia and I am sure that this one will be a success.

Rolf did ask me to put some thoughts about organizing a Fair down on paper and asked me to offer any tips that I had learned from staging the Fair in Brisbane in 1988. I felt that this might be an opportunity to put my views to you all, perhaps they be of some use.

Firstly, what are the aims of a Fair? In my opinion, the primary function of a Fair is to provide a focus for a meeting of TI99/4A users from throughout the country. All to often there are names that you see in newsletters or voices that you hear on the phone but these people remain essentially unknown. The Fair brings TIers together.

There is an exchange of knowledge and experience at Fairs. Those attending talk about their projects and ideas, how their respective groups operate, new software and a stack more topics that I could not even begin to list.

This exchange is important. It can revitalize a group or give members new ideas or perhaps a new way of doing something. Everyone seems to pick up some added enthusiasm for the TI99/4A.

The second aim of a Fair is to display new software and hardware. This might be the latest that is available commercially or perhaps it is freeware. It might be from the US or it could be locally produced.

Next, everyone must have the opportunity to take something home with them. This means software and hardware for sale, be it through sale of commercial goods or copying of programs from the library. A second hand table or a swap session is always popular.

Along the same lines is the production of a special newsletter. This is available to all who attend. It might contain articles that are not usually in a newsletter and will certainly be larger that normal. Here is something that everyone can hold and say that it was part of the TI Fair.

A Fair also provides an opportunity for publicity, locally, nationally and internationally. It gets the name of the group known and perhaps introduces new members to the group.

Finally, there should be a major drawcard. The attendance by a popular or well known TIer is great, or perhaps some fancy or rare piece of hardware (a TI99/8 springs to mind) or maybe some spectacular raffle prize. These are the things that will motivate members and other groups to attend.

OK, so there are the aims. How do you go about achieving them? I might cover this section chronologically (I love that word). First start advertising the Fair in the US. This could be done by writing to MICROpendium for an article in the NEWSBYTES column, advising of the date of the Fair and inviting all readers to spend some time in Australia. I am sure that they will also give a listing in the Fairs column.

Try to have a message placed on each of the major BBSs in the US and ask that those who read the message spread it to other bulletin boards. Also ask all groups with whom you exchange newsletters to place a small notice in their newsletter (do the notice up for them and enclose it with the next newsletter). Encourage other groups that receive their newsletters to reproduce the notice as well. This gets the word around. If any TI99/4A users from overseas decide to visit, they will need plenty of time to prepare.

Do not forget the members of groups in Australia. Contact the other groups and provide a similar notice for their newsletters. Most importantly, keep them up to date as the arrangements are made. The president, secretary or editor will probable pass the information on to the members. That keeps reminding them that it is still on. Even people travelling interstate will need time to prepare, whether it be to arrange time off work or just make arrangements for the weekend.

Next write to every distributor of TI99/4A software and hardware inviting them to send their products for demonstration and sale. The software or hardware for sale can be on consignment basis or if the funds are available, the group could buy it. Invite representatives of the company to attend the Fair (pay particular attention to people with whom you are already in regular contact). Perhaps a couple of travel brochures (glossy, colour ones) about Australia will help.

Do not expect a big response from these letters. I had replies to less that 20% of the letters sent. If no reply is received within a couple of months, write again. Sometimes they need a little prompting. For those who do reply, finalize the arrangements as early as possible and remember to allow plenty of time for any goods to be sent to Australia. We had a bundle of software arrive on the Monday after the Fair. Make sure that the people sending the software or hardware are aware of possible delays.

Most importantly, keep them up to date. One or two letters during preparations will do a lot for their confidence in your ability to represent them effectively.

Also approach computer companies that support you in Sydney to set up displays. Companies that sell printers (say Epson or Cannon) may send along a representative with a couple of samples. Do not forget Texas Instruments. They may not have much to offer any more but there will certainly be some interest. You might like to be bold enough to ask them for a donation to help with advertising or even a sponsorship.

Selection of venue. Try to pick a venue that is reasonable central for your own members. Simple economics will tell you that the majority of attendees will live in the Sydney area. There should also be a variety of styles of accommodation reasonably close by. I do not think that a convention centre is necessary, simply because it is too expensive and most visitors will not be staying there anyway. Anywhere that is big enough, well presented and easily found will probable suit. Consider also that people have to eat. Food outlets should not be far away and a booth selling coffee, tea, cold drinks, biscuits and sandwiches will always do well. I am sure that your group will be little different to Brisbane where membership is mostly men and the wives have no interest in computers. My wife dedicated herself to the refreshments booth and had a great time meeting everyone.

As the date draws a little closer, it comes time to start preparing a special edition newsletter. It should have some articles of general interest in the TI99/4A field, something from the regular writers in News Digest, reviews of software and hardware that will be demonstrated or on display at the Fair (more about that later) and perhaps advertisements for the suppliers of the hardware and software. Approach representatives of other groups to submit something about their group or something topical and perhaps the regular authors of articles in other groups newsletters as well. The aim is to make it a truely Australian newsletter (because this is the Australian Fair).

In the weeks before the Fair, the software and hardware from the suppliers should be arriving. This is an opportunity for many people to become involved in the Fair. Volunteers will be needed to demonstrate these products and in turn they will have the opportunity to try out something new. The organizers should distribute the incoming hardware and software to the other members as quickly as possible to allow them to become familiar with it. Depending on the persuasive powers of the newsletter editor, reviews could also be written.

It is also time to approach the media. Many radios and some television stations have community billboard type segments. Let them know the details of the Fair. Write to every newspaper, radio station and TV station and advise them of the upcoming Fair. These will probable receive best response if accompanied by or in the form of a press release. In Brisbane, we received no response from the media but then I knew nothing about getting the attention of the media then (I do not know much now either).

On both the weekend before and the weekend of the Fair, try advertising in a major newspaper. Where to advertise is something that I cannot even offer a suggestion because I do not know the Sydney papers. Nothing fancy because it will not have a lot of effect anyway since the people who read those sections have little interest in the TI99/4A. It may attract a couple of people though so it is worth it.

If you have visitors arriving from overseas, arrange to have someone to meet them at the airport and provide help if it is needed. This also applies to interstate members who are arriving by air, rail or bus. If deregulation of the airlines still has the effect that it has now, there will be quite a number of people flying in.

The program for the weekend of the Fair is really up to the organizers and depends on what is available. It may begin with an informal get together on the Friday evening, mostly for visitors I suppose. On the day there will probably be presentations and demonstrations, speeches and the usual. There should be time for visitors to circulate in between programmed events. In 1988 we ran out of time, even with a fairly brisk program. Try to get started by 9.00 am and do not expect to finish in mid-afternoon. If at all possible, a venue that allows equipment or at least tables to be set up the night before would be ideal.

There will be volunteers needed to man the disk library table or the second hand table. What will probably be needed is computers. Many guests will not be able to provide their own so it will be up to the members of the group to make their's available for the day. A Fair without enough computers would be a real pain.

A night out after the Fair always seems to be a great success. It is a time that is enjoyed by all and is certainly recommended.

And that is the Fair over. If most visitors are accommodated in the same area you may like to consider a farewell meeting or perhaps a barbeque style brunch on Sunday morning although most will have to get away to be ready for work on Monday morning.

Well those are my thoughts. I am sure there will be some points that you will not agree on and there will be a equal number of suggestions that you have to add to preparations. I urge you to speak up. I urge you to help out. Organizing a Fair is not difficult. The largest part is the number of letters that have to be written and with a computer and a mail list, that is not really a problem. I arranged the Brisbane Fair largely without help, mostly because I did not know what I was going to do next. A small committee would have no problems at all.

What is needed is your ideas, your support in presenting the software and hardware on the day and your attendance. Do that and you will have a Fair that must be a resounding success for you personally, for the Sydney group and for TI99/4A users throughout Australia.

Secretary's Notebook

by Terry Phillips

There seemed to be an interested bunch of members at the November meeting, at least up until the time when I had to leave due to another commitment.

Bob Relyea was giving his knowledge of Multiplan to a group of members trying to come to grips with the complexities of this useful utility, while Larry Saunders and Alf Ruggeri had a fairly large group for their talks and demonstrations of Page Pro, TIPS and other graphic programs. Larry was completely set up with even a video camera, so I hope he got some interesting shots. Daniel Harris was trying his hardest to attract some members to his demonstrations of Console Writer, but he was plagued with printer problems. Dick Warburton, Peter and Ross Mudie were busy throughout the day with RAMcard enquiries and fixes, while, as usual, the shop was under the competent stewardship of Percy Harrison. Ross's model train system was also set-up and attracted much interest particularly from the younger, and young at heart, members.

As all members would (or should) be aware, the December meeting is AGM time. The agenda for this important event, together with the minutes of the 1990 AGM and Financial Statements were included in the November TND. If you are planning to attend the AGM, and this is one meeting that should be high on your agenda, then please bring them along for reference on the day.

After the AGM it will be time for the partying. Yes the balance of the day after the formalities will be devoted to a social function with plenty of food and drink. Russell Welham has been given the onerous task of catering for this day and he tells me he has this well in hand. To help defray costs, there will be a small charge of $4 per family, but this will be good value.

As I foreshadowed in the October TND, we have had no choice but to make a change to the club's mailing address. As from 28 October, 1991, our post office box address has changed to:

PO BOX 1089
STRAWBERRY HILLS NSW 2012

Please make sure that all future correspondence you have with the club is addressed to the new box number.

There are two new members to welcome this month and they are:

Robert Erskine from Epping, and,
Jim Stubbs from Ryde.

A big welcome to you both.

A thanks to Ian Mullins. Ian has volunteered to take publicity on the group to various shopping centres for display on notice boards. With a bit of luck we may attract some new members through this form of advertising.

Treasurer's Report

by Geoff Trott

I would like to wish all our members a very merry Christmas and a Happy New Year. I am looking forward to a few months without deadlines to look at a few items that have arrived while I have been too busy. There is a very interesting letter from Pierre Garouche in France which I will examine closely and reply to and I owe letters to Tony McGovern and Barry Traver amongst others.

Income for October	$2436.80
Payments in October	$5806.50
Excess of expenses over income for October	$3369.70

TIsHUG Shop with Percy Harrison

We have managed to get hold of a very limited number of Peripheral Expansion Boxes, TI Disk Control Cards, RS232 Cards and 32K cards so those members who only have a very basic system can now add Disk Drive, Expanded Memory and Printer facilities at a reasonably low price compared with what these items cost before Texas Instruments withdrew the TI from the marketplace.

Should you want to expand your system please phone me and reserve the items you need as when the current stock has been sold it may be some considerable time before any more become available. Technical support and servicing of this equipment is available through the club at unbelievable low cost.

Orders and prepayments for the TIM 80 Column card were overwhelming and our initial order of 15 units has sold out. Any further purchases at the $165 price will depend on getting at least 10 firm orders otherwise the price will be around $205.

PRICE LIST.

5.25 in. DSDD Disks (Boxes of ten)	$6.00
5.25 in. HD Disks (Boxes of ten)	$10.00
3.5 in. DSDD Disks (Boxes of ten)	$10.00
5.25 in. DSDD Half Height Drive (New)	$65.00
12 Volt AC Transformer	$3.50
13 Volt Arlec Transformer	$12.00
8.5, 17 Volt Transformer	$25.00
60 VA Transformer	$20.00
MFC Printed Circuit Board	$30.00
MFC Kit (Disk Controller	$102.50
Music Kit with PCB	$65.00
32K Memory PC Board	$7.00
Eprom Ram PC Board	$45.00
Eprom Ramdisk Basic Kit	$35.00
Funnelweb Eprom Set (3 Eproms)	$36.00
TI Artist Eprom Set (2 Eproms)	$24.00
32K Static Ram IC (62256)	$10.00
8K Static Ram IC (6264LP)	$5.00
74LS08 IC (quad Schottky)	$0.50
1K Resistor	$0.05
Exchange Console	$30.00
ROS Version 8.14	$12.00
Peripheral Expansion Box	$150.00
Multifunction Card	$220.00
RS232/PIO Card	$100.00
Modem PE Card (300 Bd)	$60.00
PE Ramdisk (256K)	$200.00
Printer (Serial)	$120.00
Modem (300/1200 Bd)	$100.00

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

COMMERCIAL SOFTWARE.

Artoons SSSD	$12.00
Character Set Graphic Design Cataloguer	$6.00
Character Set Graphic Design I	$12.00
Character Set Graphic Design II	$10.00
Character Set Graphic Design III	$14.00
Disk Utilities (Memorial Edition)-DSSD	$11.00
Disk Utilities (Memorial Edition)-SSSD	$12.00
Display Master	$15.00
Genial Traveler (SSSD)	$6.00
McPaint (5 Disk Set)-DSDD	$10.00
McPaint (10 Disk Set)-DSSD	$20.00
Microdex 1 (SSSD)	$16.00
Microdex II (SSSD)	$11.00
Nuts and Bolts #1 (DSSD)	$6.00
Nuts and Bolts #1 (SSSD)	$7.00
Page Pro 99 version 1.6	$28.00
Page Pro Utilities	$17.00
Page Pro Applications #1	$2.00
Page Pro Line Fonts	$9.00
Page Pro Medical Clipart-DSDD	$10.00
Page Pro Medical Clipart-DSSD	$13.00
Page Pro Templates Vol1-SSSD	$8.00
Page Pro Templates Vol3-SSSD	$8.00
Picasso Publisher Version 2.0	$14.00
Picasso Publisher Support Disks	$6.00
Picasso Applications Disk	$2.00
Rockrunner (SSSD)	$15.00
Spell It! (DSDD version)	$24.00
Spell It! (SSSD version)	$27.00
The Missing Link (TML)	$28.00
The Missing Link Companion Disk	$2.00
TI Artist Plus	$25.00
TI Sort SSSD	$15.00
Tris Module	$25.00
Typewriter Module	$25.00

Packaging and postage charges:

Due to a change in the way Australia Post charge for Packages I am not able to list the postage rates at this time. Instead of their usual method of charging by weight they have now introduced a system whereby they charge by volume which makes it difficult to assess the actual rates. I am looking into the new system and hope to be able to give some indication of the postal charges in the next issue of TND. One thing for sure is that the rates will be higher than they have been in the past.

I would like to take this opportunity to thank all our members for their tremendous support throughout the year and to wish you all a very merry Christmas and a Happy and Prosperous New Year.

TIsHUG Software Column by Rolf Schreiber

December Software Releases

Software to be released will include some of the disks brought back by Ben Takach on his recent trip to the USA. As well there will be new software from Bruce Harrison of Harrison Software, several disks from Jim Peterson's Tigercub Software, new TIPS disks, and a suitable disk for the Christmas season.

For those of you who have been having trouble using TIPS picture files with the TI Print Shop (TIPS) program, here is a tip passed on from Larry Saunders: make sure that all the filenames of the TIPS pictures that you want to use with TI Print Shop (ie the TIPS program itself) start with the letters "GR" and end either with the letters "TXT" or "XXX", otherwise the files will not load. More on TIPS elsewhere in this issue.

Commercial Software

The following items have arrived from Asgard Software and will be available at the December meeting.

Page Pro Banner Maker
Page Pro Poster Maker
Pix Pro
Legends
Disk of Dinosaurs
Rock Runner
Screen Preview
Spell It! (Hard Disk version)

Multiple copies have been ordered of the more popular items. If you cannot make it to the meeting then contact Percy Harrison and arrange to buy what you want by mail order.

The following software is on back order, and should be available soon:

TI-Base
TI-Artist Plus!
Fonts & Borders I
Fonts & Borders II
Fonts & Borders III
TI Casino
Son of the Disk of Dinosaurs
Filmlib for TI-Base

continued on page 34

Visiting TI99/4A Users in USA

by Ben Takach

Jim Peterson is the first on my phone list whenever I visit Columbus Ohio. Besides his programming talents, he is a dedicated member of the CONNI user group. This fast developing mid-Western city does not offer many attractions to visiting tourists, however it is a city where one can make many friends. People have time to talk, and everybody is very polite.

Jim alerted me to the user group meeting on the following Saturday. The usual meeting place of the group is in the canteen of a chemical company adjacent to the football stadium on the Olentangy River road. There was a non-televised game scheduled on that Saturday and the expectation of heavy traffic scared away many members. The handful of members present were just as enthusiastic as any in our group. The numerous warm comments praising our newsletter made me proud to be a member of TIsHUG.

The problems of the group are the same as ours: Decreasing membership, diminishing new software, members changing to PC's and its clones. One has to face it, we are flogging a dead horse. The group has 2 operating BBS. One is run by the current president Chuck Grimes, the other by Irvin Hott.

Chuck remarked that the BBS is used more and more by out of town members. He related that the Chicago group is one of the strongest in the States. Their annual fair attracts the most vendors and it is a hectic affair. The disk of the month was distributed by the president, and I was given a complimentary copy. It consists of 2 notched disks, all four sides filled with archived programs. Guess what is the highlight of the collection! It is the Funnelwriter Version 4.00, complete with docs and all. Tony and Will have promoted Australia more than Crocodile Dundee and the Quantas Koala put together.

The other titles on the disks are neatly indexed and documented on the READ--THIS file copied to the first of the 2 disks, and reprinted below.

****

Disk of the Month (DOM) from CONNI
by Chuck Grimes - President
for September 1991 #54

Side A

```
Diskname DOM-09/91A Total Sectors 358  Free Sectors 00   Date 09-16-91 Files 4
-------------------------------------------------------------------------------
                               Sector
Filename    File Type Size FDR  Start End Pro Comment
-------------------------------------------------------------------------------
FASTICE^    I/F  128    12 0004 0005 000F No  E/A-5 music with X/B loader
                                              the loader has instructions in it
                                              to use on most E/A-5 files
FW44A^      I/F  128   278 0002 0022 0136 No  Funnelweb version 4.4 part A files
NUMACC^     I/F  128    50 0003 0137 0167 No  assembler ACCEPT AT for numeric
                                              variables, included is source and
                                              demo file from Bruce Harrison
READ--THIS D/V   80     18    5 0168 0000 No  the doc file for this DOM
```

Side B

```
Diskname DOM-09/91B Total Sectors 358  Free Sectors 1    Date 09-16-91 Files 2
-------------------------------------------------------------------------------
                               Sector
Filename    File Type Size FDR  Start End Pro Comment
-------------------------------------------------------------------------------
D-G/GR^     I/F  128    66 0003 0144 0167 Yes CSGD graphics from the collection
                                0004 0020     of Joy Warners
FW44B^      I/F  128   291 0002 0022 0143 Yes Funnelweb version 4.4 part B files
                                              and the 80 column files.
```

Side C

```
Diskname DOM-09/91C Total Sectors 358  Free Sectors 1    Date 09-16-91 Files 2
-------------------------------------------------------------------------------
                               Sector
Filename    File Type Size FDR  Start End Pro Comment
-------------------------------------------------------------------------------
FWDOC^      I/F  128   298 0002 0022 014A Yes Funnelweb version 4.4 docs main
H-M/GR^     I/F  128    59 0003 014B 0167 Yes CGSD graphics from the collection
                                0004 0020     of Joy Warners
```

Side D

```
Diskname DOM-09/91D Total Sectors 358  Free Sectors 1    Date 09-16-91 Files 3
-------------------------------------------------------------------------------
                               Sector
Filename    File Type Size FDR  Start End Pro Comment
-------------------------------------------------------------------------------
CONV^       I/F  128    10 0017 0018 0020 No  hex to dec converter, see the docs
FDOC80^     I/F  128   170 0002 0022 00CA No  docs for 80 column Funnelweb
                                              version 4.4 files
MICRO^      I/F  128   177 0003 00CB 0167 No  programs from MICROpendum w/docs
                                0004 0016     7 little programs
```

There you have another big DOM from CONNI. This month the main program is the latest Funnelweb version 4.4 with all the 80 column files and documentation. This is a do everything program from Extended BASIC.

Also please note that we have adopted a new symbol to indicate that the file is archived. The "^" indicates that the file is archived with ARCHIVER V3.03g. This we hope will become the standard as it only takes one space of the limited file name.

Look next month for more good things and more CSGD graphics from the collection of Joy Warner. Until next time, enjoy. Chuck Grimes

****

The unarchived files produced the complete FW-4.00 version as well as 21 files of the various MICROpendium programs and DOCs and 225 CGSD graphics files. Each of the graphics is a 2 sector I/V 254 file. It is indeed a generous set of disks of the month for $4.00!

The members interest is diverse, just like TIsHUG's. There is hardly any call for programs on tape. Most of the members use one or more hard disks. The problem of hard disk backup with Myarc's disk manager prompted many members to install a second hard drive, and backup by directory copy from one to the other. This, like all other remarks were familiar tales.

Reflecting on that Saturday morning, I believe I have learned the reason for our success: it is universal involvement. Everyone contributes in their own way. The burden is not carried by one person alone. We in Sydney are on the right track. It was a pleasant and friendly few hours amongst instant friends.

I did manage to visit Irvin Hott. His street address must be one of the hardest to find locations in Columbus. My taxi driver was utterly lost from the time we left the downtown area until we arrived. The fact that he hailed from Ethiopia with limited knowledge of the English language made matters even more difficult. Irvin lives with 2 cats (each of which is large enough to serve three generous chinese meals), and 3 dogs. Two rowdy pups and his trusty seeing eye dog. Irvin is totally blind. He has run the official CONNI BBS since January 1986. Well, most of us have a reasonable understanding of the problems of our dedicated SYSOP, Ross Mudie blessed with the benefit of sight, just imagine Irvin's problem. His contribution of speech programs ought to be promoted to help other sightless people in our community. TI's speech synthesiser gave him the chance to enjoy the fun of computing, to pave the way and open the door to other electronic products to overcome his handicap. I was amazed by the number of simple and affordable "peripherals" he uses to sense the functions of his TI99/4A.

He was eager to provide me with copies of his speech programs in the hope that we would pass it on to Australian enthusiasts who are similarly affected. Irvin is a great guy, attending meetings, travelling to TI99/4A fairs accompanied by his dog. Besides the speech synthesiser, he has a small transistor radio resting on top of his console. This is tuned to a quiet spot of the frequency band. The wireless picks up the RF radiation from the console. He will listen to the clicks and noises emitted by the speaker and he could tell me with incredible accuracy what was going on in the system. The monitor naturally is not switched on, but he turned it on for my benefit. At times it was hard to concentrate, because Tonka the guide dog was jealous of the two mischievous puppies and all three were competing for my attention. Irvin would announce what was going on almost as fast as the screen displayed it. I was emotional about his undeserved handicap when I arrived, but this quickly vanished when I saw him in action. Irvin does not need any sympathy. He is as competent as anyone blessed with 20-20 vision.

Of course after the initial success with the TI99/4A, many more sophisticated faster and easier to program electronic devices have been released. He uses a multi-function device called the Braille and Speak. This is almost a pocket computer. It has a 256k memory, of which some 184k is available for user programs or files. It is -naturally- battery backed and fitted with RS232 interface. He uses it as a notebook, an address book and as an interface to the TI99/4A as well as his PC.

He also uses a braille printer to get hard copies of his files he wishes to edit. This is an ingenious printer, combined with an integral speech synthesiser, which will announce the selected printer setting and produces a crisp well defined braille printout. Again, nothing is perfect in this fast changing world, this printer has its own problems. Braille is not a letter by letter translation of a written text. It has its own abbreviations or syntax to speed up reading. When it is printing a text file from disk or computer memory, it naturally will print faithfully every character. Thus experienced Braille readers will get confused.

I was amazed by Irvin's incredible memory, locating specific disks instantly amongst hundreds stored in his trays. He was formatting, copying and labelling disks faster than I could do it, unerringly picking up blank disks, jackets and the Dymo label marker (his Dymo prints labels in Braille). I was oblivious to my surroundings on the way back to the hotel. My thoughts were with Irvin and his four-legged friends. What a waste of effort if one leaves things at this point. Just a column in the news digest, which is soon forgotten. And yet all his efforts could help many talented people groping in eternal darkness. I have wondered ever since, does Irvin have dreams? And if so, what shape to they take? My dreams right now are to pass on Irvins achievements to those who need them most. Would it not be an ever-lasting monument to TISHUG?

Also in the Columbus area, I met the USA counterpart of our John Paine. The two do not look alike, but this small matter is the only difference. His name is Lee Bendick. He lives in a small city about 40 minutes drive from Columbus Ohio. Lee's collection of TI stuff is incredible (this is really an understatement, I used it for want of a better expression). You name it -he has it! I was playing with a TI99/4A in his Aladdin's cave basement which has 2 PE boxes, both filled with cards. This unit has drives from DSK1 to DSK9, and in addition drives from DSKA to DSKK (I kid you not, it is a TI99/4A!). Next to it is a Geneve and a 386 mini tower sharing the same monitor. He uses two TI-CC40 hand held computers just for fun. He also has a TI99/2, alas it has no power supply yet, and a TI99/8.

Now this is a mind-boggling piece of equipment, considering it was made in 1983. The title screen comes up showing the characteristic TI99/4A format, but this is where the similarity ends. It has its own PEbox interface card, 64K RAM on board, EPROM based XB and Pascal on board and it can access 15MB of RAM directly. Peeling off the cover reveals that the motherboard is a prototype with several PC boards plugged in to IC sockets on the motherboard. Each of these in turn are populated with up to 7 or 8 IC's. The high cost of high capacity RAM chips and state of the art LSI chips in those days were the obvious reason for this compromise. The PEBox connection socket is a multi-contact telephone connector similar to the centronic plugs and sockets. Obviously the need for a more reliable interface was recognised by the TI99/4A designers shortly after the release of the TI99/4A! Its Extended BASIC is a combination of XB and DOS. Fully compatible with the TI99/4A XB as far as programs go.

We played with the many weird and wonderful "top secret" TI99/4A programs oblivious to time. I only noticed after Lee dropped me off at my hotel that it was three in the morning -just like when you drop in on John Paine for a brief chat.

OCTOBER IN LA

Los Angeles is a good place to unwind after 3 weeks of hectic business trip across the industrial cities of the Mid West. It is the last watering hole before the long haul to Sydney. Its steamy-humid climate is much like the one we are used to.

Many years ago, by the whim of my travel agent, the Airport Hilton became my temporary LA home. It was almost new then, impressive, functional and the easiest spot to reach the attractions of the sprawling greater Los Angeles area. The International airport is a few minutes ride by the courtesy shuttle, and from there one can reach any spot without the need to ask many questions. I hired a car during brief visits of the past, but this is no longer necessary. I have learned and understand this this metropolis now: its international airport, LAX, is the hub. Bus or shuttle operators will take you anywhere from there for a fraction of a taxi fare.

The LAX Hilton, like all of us, has aged over the years. One notices the odd cracks appearing in the hand basin, loose or cracked wall and floor tiles in the bathroom. Spots of cigarette burns and the signs of the hopeless struggle of the house-keeping staff to remove chewing gum from the carpets do tell the tales of passing years. The hotel has a lived-in feeling. Although it is a modern structure, it will not age gracefully like the few remaining grand old hotels built in the early years of this fast disappearing century. Many new Hotels have changed the skyline of the Airport district since Los Angeles hosted the Olympic Games, but I remain faithful to LAX Hilton.

Los Angeles is one of the most expensive parts of the USA, and lets face it, is not the most attractive tourist spot by a long shot. After a single one week stay you can truly say you have seen it all. Year by year the once glamorous Hollywood loses some of its gold plating and the internationally known Sunset Boulevard is just one of the many LA Boulevards in bad need of a good scrub and a face-lift. LA, unlike a European city, does not get more attractive with age. It is brash loud and busy, not Medieval or Victorian. It is up to the developers to turn the once famous ticky-tacky and plaster into sterile glass and concrete when the price is right. If you try to capture some of the swashbuckling Errol Flynn atmosphere or the chilling Alfred Hitchcock's thrills, then do yourself a favour, keep your dreams of yester years and do not visit Hollywood.

As it happened, I spent the first week of October in Los Angeles this year. Not much has changed since my last visit, perhaps the friendly barmaid fills her itsy-bitsy teeny-weeny skirt a lot more than she did the last time she served me, but it only proves that everything is dynamic around us and we live in the age of constant, rapid growth and change.

continued on page 8

TI99/4A World News

edited by Jim Peterson, Tigercub Software, USA

This article appeared under the heading TI World News in the October 1991 issue of Spirit of 99 - author Jim Peterson

Mark Schafer of the Bluegrass 99ers has written a defragmenter program to defragmentise a disk on the disk itself, rather than copying files from one disk to another. It is available from Mark for $7. His address is 539 Whitaker Street, Morehead, KY 40351.

Bill Gaskill's newsletter for TI-Base users has ceased publication with Volume 2 No. 4. Bill says that refunds have been sent to all subscribers and the newsletter is no longer available, neither are any products offered by PRK DataBasics and Junction Softworks.

Texaments has released Fonts, Frames, Fun. A 3 disk package of 11 fonts, 39 frame borders and 22 instances for use with TI-Artist and TI-Artist Plus - cost $12.95. Also released The Missing Link Font Pak with 29 screen fonts for use with The Missing Link - cost $7.95. The price of the TI-Artist Companion 2 disk sets numbered 2 through 13 has been reduced from $9.95 to $7.95. The address is 53 Center Street, Patchogue, NY 11772.

According to an article in the West Penn Newsletter, Tony Lewis and Al Beard have been collaborating for over a year on a math coprocessor card for the PE-Box. The card is now functional although not 100% complete. It uses the Motorola 68881 math coprocessor chip and is expected to be accessible from the TI99/4A, the Geneve or the 99105 accelerator card.

Bruce Harrison (Harrison Software) has finished an assembly game called ScudBusters. The game idea is to shoot down Scuds with Patriot missiles. The game will be released at the Chicago Faire. He has also written and released to the public domain a disk full of random number generating routines in assembly.

Tigercub Software is now offering a free bonus disk with each 10 TI-PD disks purchased.

Barry Boone has written a program to convert digitised sound from PC format to TI/Geneve format and another program to play the sound on the TI/Geneve. The sounds are reported to be incredibly realistic, but they also apparently use up an incredible amount of memory. The programs will become available through Texaments.

The following article appeared in the Spirit of 99 newsletter of October 1991 under the heading Random Access to my Memory Banks by Jim Peterson.

The 99105 Accelerator, to be priced at $250, will speed up the TI99/4A by a factor of 5. In combination with the PEB RAM interface card, to be priced at $90, the increase will be 10 fold. These are remarkable devices, and I am sure the hardware hackers will love them, but do the computer users really need them?

Personally, every program I am apt to use is on my RAMdisk, for almost instant access. Prescanning has eliminated those initialisation delays. The programs generally run as fast as I want them to.

I know that Stephen Shaw over in England is writing graphics plotting routines that take hours and hours to run, but that is the kind of thing you boot up and start just before you go to bed, nothing is going to speed it up so much that you would want to sit and wait on it.

Yes, sorting routines are a bit slow, even in assembly, but the TI99/4A just does not have the memory to sort any sizeable amount of data anyway. TI-SORT gets around that limitation, to some extent, by writing and reading a scratch file on disk, but disk access is necessarily slow. Other chores you might like to speed up, such as archiving, are slow because of disk access. I know, I know! The hard drive overcomes that limitation. Sorry my time is valuable, but not that valuable!

And maybe you would like to speed up Multiplan, or TI-Base. Here again the TI99/4A is handicapped by lack of memory. It was designed as a Home computer, not a Business computer. I have only two projects I would like to put in a database, and both are too large to be practical on a TI99/4A. If I ever get around to working on them, I will put my money into a second hand PC (pardon the word) rather than add more hardware to my little home computer.

But if you absolutely have to speed things up, maybe there are cheaper ways. For years there have been references to changing a crystal, costing about $4.50, in the console. Finally, in the Hoosier newsletter of August 1991, I ran across an article downloaded from Delphi, in which Jesse Slicer explains exactly how to do it. He says you will gain an increase of 19.3% in microprocessor speed, but you will experience problems with terminal programs and graphics-intensive programs. So he tells how to install the new crystal, in addition to the old one, with a switch to alternate between them.

In the same newsletter are two other Delphi downloads. One is a step by step description of Mike Ballman's method of adding 32K of RAM memory to the 16 bit bus for a speed increase of about 50% It requires two chips costing about $13 and a couple of other doohinkies, price not mentioned.

The other file describes a modification to this method by the late John Guion. Now all of this is complete Greek to a technoklutz like me, but if some hardware hacker was to start buying up $25 consoles, making these modifications and reselling them at a reasonable price, I might be interested.

Perhaps a better way to speed things up is through the software rather than the hardware. A few years ago Ryte Data was marketing a so called BASIC Compiler. It was not really that, but it did speed up Extended BASIC programs quite drastically. Unfortunately it had so many limitations and created so many bugs that it was worthless. Last year, someone on the east coast announced that they were writing a true BASIC compiler, but I have heard no more about it. I believe that J. Peter Hoddie and Paul Charlton examined the idea previously and decided it was impossible within the memory limitations of the TI99/4A.

Ideally, programs should be written in assembly for maximum speed and efficiency - as Mike Ballman would point out. Trouble is there are very few programmers with his skill, and very few programs being written in assembly.

But actually, Extended BASIC is a perfectly satisfactory programming language, easy to write and easy to modify, and can be fast enough for any purpose if it is supplemented by CALL LINK's to assembly to do whatever Extended BASIC cannot do fast enough or cannot do at all. Some of the best programmers have gone that route, but most of us do not know how to write those assembly routines to link to. We need someone to do it for us. Some extremely useful routines have been written for us by such skilled programmers as Karl Romstedt, Barry Traver, Bruce Harrison, Bud Wright, Ross Mudie and others. We need more of them.

Just think. If, back in 1984 when I put out my first 100 Nuts & Bolts Extended BASIC sub-programs, some assembly genius had started putting out disks full of assembly sub-programs. What a wonderful programming language we would have by now, and what wonderful programs might have been written!

o

Extended BASIC Tips

by Bob Relyea

Making A Menu
Part 2

In last month's article I outlined three steps in making a menu for programming purposes and I explained step one. The intent of this article is to finish off the article by getting into steps two and three.

Steps Two & Three

Once you have made a choice you must have a mechanism of having the computer go off to a pre-arranged place in the program to do what it is asked to do. A common method of doing this is to use the CALL KEY subroutine. If you look this up in your Extended Basic manual (p. 109) you find the instructions of setting this up to be:

```
CALL KEY(key-unit,return-variable,status-variable)
```

A whole article could be given to this statement alone but I will give you just what you need to know to get the 'show on the road'. A common input for 'key-unit' is 0 (zero) as this means that you can potentially use any key on the entire keyboard. 'Return-variable' is where the input goes that directs the program off to the line-number to execute the subprogram. This is what we will have to come back to and explain. Status-variable is just an indicator for the computer as to whether or not a key has been pressed. It also keeps the program from continuing until you press a key. All you need to do here to accomplish this is to put S=0 in the appropriate place. More on this when we come to it.

Right! Now just for completeness, I will repeat the three program lines that I started with last month:

```
100 DISPLAY AT(2,1):"1.  CALCULATE AREA."
110 DISPLAY AT(5,1):"2.  CALCULATE VOLUME."
120 DISPLAY AT(7,1):"3.  CALCULATE SURFACE AREA
```

Now, for the program to be 'menu-driven', you must provide for the program to go off and do each choice above and then return to the menu for further instructions. This WILL NOT be done by simply pushing number 1, 2 or 3 on the keyboard if you are using the CALL KEY statement. The reason is that number one on the keyboard is ASCII number 49 (see p. 196, Extended Basic manual). Likewise, number 2 is ASCII number 50 and number 3 is ASCII 51. You, therefore, have to 'trick' the computer so when you press key 1, for example, it thinks you have asked for ASCII 49. This is where the 'return-variable' comes into it. As the title suggests you put a variable is this part of the statement and then give the limits of the variable in the lines that follow. We only want numbers 1, 2 and 3, which correspond to ASCII 49, 50 and 51. The easiest thing to do here is to give the variable the name of K for return-variable, and then set up a line so that when you press 1 the computer 'sees' 49. The following program line does this:

```
140 ON K-48 GOTO 200,300,400
```

If you do your sums you will see that since K is equal to 49, 50 and 51, that K-48 is equal to 1, 2 and 3 which is just what we want!

The full CALL KEY statement would be:

```
130 CALL KEY(0,K,S) :: IF S=0 THEN 130
```

The second part of the line is just telling the computer that if no key has been pressed to stay on that line until one has been.

There should be another line before 140 so in case you press another key apart from 1, 2 or 3 that it will wait around until you do without giving an error message. This can be done by keying in:

```
135 ON K<49 THEN 130 :: IF K>51 THEN 130
```

Now, what about the lines 200, 300 and 400 mentioned in program line 140? These are the lines that your subroutines are written after pressing keys 1, 2 or 3 respectively. There you write a few lines to calculate the area, volume or surface area and at the end of the subroutine you write a line that directs the flow of the program back to the menu from which you can make another choice.

The first and next-to-last line of the subroutines should be CALL CLEAR. You should also have a choice on the menu to abort when you are though making choices and want to end. This is usually accomplished by-

```
125 DISPLAY AT(11,1):"4.  END."
```

Line 140 above would then be altered to read:

```
140 On K-48 GOTO 200,300,400,500
```

Line 500 simply reads "END" which terminates the program. Admittedly, what I have written is just an outline, but the rest is left up to you to add a bit of substance and make up a simple program to run a few routines from a menu.

Let me know how you got on.

continued from page 6

In the good old time cultural milestones of communities across Europe attracted many visitors from near and far. The fasching in the Rhineland, the exodus of the Swiss herds up the Alps in the spring, the Munich October fest were popular tourist attractions. One could pasticipate without the most basic knowledge of 1he German language and enjoy every minute of the long celebration. Words were not essential after 2 or 3 Steins!

The jet age made Munich too small to entertain the thirsty world beating a path to the Lowenbrau. Now the October Fest goes on a world tour. Such unlikely spots as the Fairfield showground, where only the locals may visit for 11 months of the year will be turned into a mini-Munich in October complete with large quantities of the genuine amber liquid imported especially for the occasion.

This year's Oktober Fest caught up with me in the equally unlikely place, the LAX Hilton. It was complete down to the 'k' in Oktober and the Bavarian beer. Appropriately the menu was also Waschecht Bavarian. The lentil soup and a selection of bratwursts with sauerkraut and spatzle was a tempting change after weeks of mass-produced cholesterol-free, low fat diet. The Mexican waitress had trouble reciting the page full of special menu -minus the umlauts- but otherwise printed in the German language.

This was one occasion where I broke my own golden rule never to order a national dish outside the country of its origin. The German printing fooled me! The (no doubt) Mexican chef had his own idea about the lentil soup, bratwurst and sauerkraut. Any patriotic German cook would have instantly fried the Mexican perpetrator of this unashamed rape of a Bavarian kitchen in boiling olive oil after the first mouthful. Well, errare humanum est, we all make mistakes. Next time around, I will celebrate Oktober Fest in LA with hotdog, hamburger and Budwiser.

Wanted

A PIO printer suitable for the TI99/4A that will print Epson or Star compatible graphics. Contact Clyde on (02)797 6242.

Coming to Grips with TIPS

by Alf Ruggeri

I first became aware of Ron Wolcott's TIPS program through the advertisement in the February 1990 issue of MICROpendium. I was fascinated by its stated features and the vast library of graphics. When the program became available through our club shop in August 1990 as freeware (version 1.6), I wasted no time in acquiring it to put it through its paces. Unfortunately the all essential ....TXT and ....XXX image files were not available at the time so I could not advance beyond the first three screen prompts. Reading through the disk document files I was able to establish that its features were very similar to PRINTSHOP for PCs, a program I had seen proudly demonstrated by my brother-in-law on his PC some time earlier.

Other document files on the disk went on to describe various graphics conversion and viewing programs but once again unusable without Ron's image files. One of these conversion programs was Ed Johnson's Tips to Page Pro 99 picture conversion utility, and being a very keen devotee of Page Pro, how could I not but firmly grasp Ed's words "to help quickly expand the picture library of Page Pro 99 owners". With hindsight I do not know whether Ed's program has been all that beneficial to my well being, bleary eyes at all hours of the night and early morn not to mention the many near collisions on the domestic front. Especially after my wife had come to accept that what had originated as a hobby had become a condition of fully blown TI99/4A-mania, only to be further provoked by this latest onslaught. Thanks Ed!.

Ron's image files along with version 1.7 of TIPS was received by our user group in December of 1990. I made several valiant sorties into the TIPS program but I found I was a trifle overawed and somewhat disorientated by the many highways and byways that traversed the functions and sub-functions of the program. In the end I concluded that I could make best use of TIPS by using the converted graphics in Page Pro.

In spite of this conclusion I have always had the niggling doubt, it could be called guilt, that my initial sorties were perhaps not valiant enough, and sometime in the future, the proverbial round tuit, I should really come to grips with TIPS.

Well the round tuit came earlier than I expected, when at our last user group meeting, a fellow member suggested that a tutorial on TIPS be conducted. To maintain some degree of credibility as a TIPS user I agreed to the request and what follows is the end result. Mind you, I have had to placate my dear tolerant wife on several occasions, to offset the many hours spent in the study, "playing computers" (not my words) to arrive at this point. Enough of the homily and my domestic situation, let me now take you for a stroll through TIPS.

What can TIPS do for you?

It can produce the following items:

A/ Greeting cards.
B/ Banners with horizontal or vertical characters with or without pictures.
C/ Labels, envelopes (tractor feed) or letter heads.
D/ Decorative calendars.
E/ Posters or flyers, these can be produced in normal right reading or mirror reverse to be used with a special printer ribbon to iron onto t-shirts etc.

TIPS supports a two colour, multi colour printout, for those who have a colour printer.

Do I hear the sceptics groan that we already have these type of programs? Well that may be so, but all the above features are incorporated on the ONE program, and with the truly vast library of images, it is a very usable program. Ron's estimate of over 2000 images is I feel very conservative. My bleary calibrated eyeballs seem to recall as having inspected over 3000 unique images.

A Comparison with PRINTSHOP for PCs

In preparing my material for this tutorial, I have had another look at the PRINTSHOP program, and to my surprise, all of that program's printout features are practically the same as for TIPS, but with the added double bonus for us, of labels and mirror reverse printed posters.

Of course a most obvious difference between PRINTSHOP and TIPS is that the former allows the user to screen edit the entire composition of characters and graphics, before sending the information to the printer, whereas TIPS's input requests of characters and graphics are immediately printed. This shortcoming in TIPS is easy to accept as:

(1) PRINTSHOP resides in 640K of RAM whilst TIPS's RAM environment is only 7.5% of the PC.
(2) The TI99/4A's operating speed is only 20% of the slowest 286 machines.
(3) PRINTSHOP is entirely written in assembly, whilst TIPS although bristling with assembly links is strictly an Extended BASIC program.

On the whole, considering the limitations of the TI99/4A, the performance of TIPS is just as elegant as PRINTSHOP.

Operational Requirements

To use TIPS the minimum requirements are:

A/ Extended BASIC module.
B/ 32k extended memory.
C/ PIO/RS232 facilities.
D/ An Epson compatible printer.
E/ One single sided single density disk drive.

A double sided drive or drives would be preferable, but if you can tolerate a little disk swapping, a SSSD drive will suffice, as all the main program files occupy only 206 sectors.

If you are using a SSSD drive, it is necessary to have the two image files, i.e. ....TXT and ....XXX on a separate disk. Users of DSSD drives should locate all the main program files and the two image files on the one disk. In the case of DSSD drives, it is recommended to leave the disk in the drive, as in the normal flow of the program, it will be necessary to choose alternate TIPSFONTS and images.

My Modifications to TIPS

In order to make the program easier to drive, from my point of view, I have found it necessary to make some cosmetic alterations to TIPS. I have been extremely careful not to compromise in any way, the original functional integrity of Ron's TIPS version 1.7.

The need for the alterations arose because of the similarity of function and sub-function names on the input prompt line, and their dissimilar response in the different parts of the program.

What I have done is identify by an on screen message, the name of the currently active part of the program. I have given the different parts menu names. The multi input command lines that form these menus, have been rearranged so that the available options are displayed on the left hand side of the screen, as a single column under the header of the particular menu name. These menu names, when used in conjunction with the functional flowchart I have prepared, serve as sign posts to guide the user through the different parts of the program. Unless the user is very familiar with the program it is advisable to have the functional flowchart close at hand. Mind you, the unmodified program will of course produce exactly the same printout results, but with a lot more user perseverence.

Another modification was carried out to the LOAD and TIPSX files, but this will be discussed in the Card Making procedure of this tutorial.

Using TIPS

For the readers of this article, I have chosen not to describe or rather paraphrase from Ron's document file the significance and operation of every function and sub-function. With one exception in the Banner Making procedure, Ron's documentation, after a little hands on experience, is very descriptive. For the owners of TIPS who have not yet printed out the document file, and prospective owners, the contents of the document file was published in the December 1990 issue of the TIsHUG News Digest.

Let us go! If you look at my flowchart, you will see that after the LOAD file is booted up, the program flow divides into two paths. On choosing 2 for TIPSSHOW, turn your printer on and respond to a few simple prompts. You will be rewarded with a hardcopy catalogue of high definition pictures with their respective names. The program will then terminate. Save the catalogue for future reference.

Reboot the LOAD file and choose 1 for TIPS. You will be greeted with an instruction screen that I added, the screen will convey function entry and semi-hot key function and environment information.

Proceed to the Primary Menu. The only reason why I named this menu "PRIMARY" is because it is the first one encountered. If need be the program can be terminated from this menu.

Select P and type in a valid image name from the hardcopy catalogue to proceed to the Process Menu. You will be able to see from the flowchart, that the Process Menu is at the hub of all printout features. It is wise to leave the printer on and on line as most functions will access the the printer immediately. If the PIO/RS232 cannot locate the printer, the program will crash and the LOAD file will have to be rebooted.

From this point on I will only describe the procedures for generating the printout products. The other functions of the Process Menu have been very well documented by Ron Wolcott.

Banner Making

There are six available resident fonts, 1 to 6, that can be used for banners. There are also three FN## and one FT## image fonts in the ten disks set that constitute the entire TIPS package. I will describe the image fonts at the end of this procedure.

If you wish to use a different font other than that chosen after the instruction screen:

(1) Press <CTRL> and O (semi-hot key) and then <ENTER> to call up the Other Menu.
(2) Select F and make your choice.
(3) Select R to return to the Process Menu.

To activate the banner feature:

1/ Select M and type in your banner message. The input line is capable of accepting up to 126 characters and I can only presume that all 126 characters will result as a banner printout. In any case if a longer banner is required, after the initial banner message has been printed out, a new banner message can be typed in.
2/ Select B for banner or T for totempole from the next prompt, the two options are self descriptive.

The printer will now merrily sing to itself for a while.

If desired, a large graphic of the chosen image, chosen immediately before accessing the Process Menu, or reselected with semi-hot key I for image, can be placed at either one or both ends of the banner message by selecting P. The same banner or totempole option is available.

When I was forming the signpost menus, I was determined not to change in any way, Ron's original function names so as not to clash with his document file. However in an impromptu beta test by Larry Saunders and Percy Harrison, it was suggested by Larry that besides correcting one typographical error, probably committed between the hours of 2 and 3 a.m., that the banner functions in the Process Menu should be more readily identified. The result of that suggestion is that the MSG option is now displayed as MSG_B and the PIC option as PIC_B. It goes without saying that the function of these two options is still as per Ron's MSG and PIC function names.

The exception in documentation clarity, mentioned at the beginning of Using TIPS, concerns Ron's description of the MSG function and its use with the FN## and FT## image fonts. I must have read and reread the instructions at least a dozen times. I thought I understood every semantic implication, but after two hours of trying to coax the typed in message to be printed out in the corresponding character image graphics, I gave up and went to bed. Yes, I had nightmares about the darn thing. I would very much appreciate enlightenment on this problem, from any source. I have inspected the FN## and FT## image fonts files with Terrence Murphy's c99 TIPS View utility and they are quite charming. Whoever composed those images is obviously a perfectionist.

Card Making

1/ Select C from Process Menu.
2/ Select F from Card Main Menu
3/ Select M from Card Front Menu. Type in the message to appear at the bottom of the card front.
4/ Select I or 3 for a single or a 3x4 image montage to appear on the card front. For this demonstration select 3.
5/ Reply Y to the PAUSE PRINT prompt for the purpose of producing four different rows of pictures.
6/ Select I from Pause Menu and enter the next image name.
7/ Select R from Pause Menu for the second row of printed graphics.
8/ Select I from Pause Menu and enter the next image name.
9/ Select R from Pause Menu for the third row of printed graphics.
10/ Select I from Pause Menu and enter the next image name.
11/ Select R from Pause Menu for the fourth row of printed graphics.
12/ Select R from Pause Menu to return to the Card Front Menu.
13/ Select M and type in the message to appear at the top of the card front.
14/ Select B to return to the Card Main Menu.
15/ Press <CTRL> and I together (semi-hot key) and then <ENTER> to select the image for the inside of the card.
16/ Press <CTRL> and V together (semi-hot key) and then <ENTER> to check that the desired image has been chosen. Press any key to return to the Card Main Menu.
17/ Select I from Card Main Menu.
18/ Select V from Card Inside Menu.
19/ At this stage the verse message can either be entered, or a previous verse file loaded.
20/ Select L and follow the prompts.
21/ Type in the verse filename.
22/ Select B to return to the Card Inside Menu.
23/ Select M for the message above the card inside picture. Type in the message.

24/ Select I to print out the verse and large graphic.
25/ Select M for the message to appear at the bottom of the picture. Type in the message.
26/ Select M for the message to appear below that of step 25.
27/ Select F to form feed the paper.
28/ Select B to return to the Card Main Menu and then E to return to the Process Menu.

Comments on Card Making Method

The above method represents only one possible approach, it was chosen only for the purpose of demonstration. Individual users will probably develop better methods. For example a date can be merged with the verse in the Card Inside Verse Menu. The verse can be updated in the same menu although my own preference is to edit and manually centralize each of the lines in TI-Writer. Better use of the semi-hot keys can be made.

Printer Tab Modification

At this point I would like to thank Russell Welham for supplying me with TIPS background information at such a short notice. Amongst this information was an article from the January 1991 issue of MICROpendium. The article submitted by Edward Machonis, is an addendum suggested by Ron Wolcott for versions 1.6 and 1.7 of TIPS. The addendum identifies a tabbing problem that may occur with Epson printers when printing out the inside portion of the card. The problem, simply stated is a skewed printing of the first row of 3x4 images. From my own experience, with my Epson LX-850 Printer, the printed messages above and below the internal picture or pictures are not centralized about the right half of the card inside and this appears to be part of the same problem.

Ron's suggestion to Edward was to modify a hexadecimal string in the LOAD file and line 2520 in the TIPSX program. Both of these were carried out. Unfortunately the problem still exists after several checks of the modifications.

If your printer behaves in an erratic way, I suggest you reverse the modifications in the LOAD and TIPSX files. Ron's modifications were to replace with the Sector Editor in Birdwell's UTILITIES, the hexadecimal string 1B4428009 with 1B66002800. When I first inspected this modification, I noticed that the replacement string was one character longer than the string to be replaced. I suspected that this might have been a typographical error, it was confirmed when the Sector Editor could not locate hexadecimal string 1B4428009 in any part of the LOAD file. On further inspection of the LOAD file, I did locate the hexadecimal string 1B44280009 in the 8th and 21st sectors of the file as further mentioned by Ron. The replacement was carried out and I proceeded to modify line 2520 of the TIPSX program, by booting the program and BREAK with <FCTN-4> at the first prompt.

The original line 2520 was as follows:

```
2520 PRINT #1:CHR$(27);CHR$(
68);CHR$(40);CHR$(0);CHR$(9)
;CHR$(27);CHR$(76);CHR$(170)
;CHR$(1)
```

The replacement line 2520, currently in my modified TIPSX program is:

```
2520 PRINT #1:CHR$(27);CHR$(
102);CHR$(48);CHR$(40);CHR$(
27);CHR$(76);CHR$(170);CHR$(
1)
```

Before you consider reversing Ron's suggested modifications, as I have translated them, I would strongly urge you to inspect the MICROpendium article for yourself. It has been pointed out to me that I am not unknown for making typographical errors of my own.

In the previously mentioned impromptu beta test, Larry's Star NX1000, behaved perfectly while Percy's early model Star Gemini's operation was similar to mine. In the addendum article, Ron was reported as promising a universal tab in version 1.8 of TIPS. I do not know if version 1.8 of TIPS has been released. In any case, if you follow my suggested demonstration procedure of Card Making, the offending effect will be reduced to a minimum.

Calendars

1/ Select O from the Sign Main Menu if a font, different to that already chosen is required.
2/ Select F to specify the font chosen for the Month and Year line.
3/ Select R to return to the Sign Main Menu.
4/ Select P.
5/ Select C and enter the year and month (numeral).
6/ Reply Y to the pause print prompt.
7/ Select F to specify the font for the Names of the Day line.
8/ Select R.
9/ Select F to specify the font for the Number of the Days lines.
10/ Select R.
11/ After the entire calendar is printed out the Sign Poster Menu will be presented.

Posters

In the Sign Poster Menu, the following features are available:

I will immediately print a large graphic of the chosen image.
M will immediately print a normal size line of characters (maximum 30) of chosen font.
H will immediately print a large size line of characters (maximum 15) of chosen font.
L will immediately print a line of native printer font at double width (maximum 40).

Both posters and calendars can be produced in the Sign Heat Menu. The result will be a left to right mirror reversal of the Sign Poster Menu products. From Ron's file, I understand a special printer ribbon is required to make these graphics heat transferable (ironing) to fabrics.

Labels, Letter Heads, Envelopes

1/ Select the image for the label in the transition prompt between the Primary Menu and the Process Menu. Alternatively select another image in the Process Menu.
2/ Select L from the Process Menu.
3/ Select H for header + 3.
4/ Type in the header line (maximum characters is 36).
5/ Type in line 1 (maximum characters is 71).
6/ Type in line 2 (maximum characters is 71).
7/ Type in line 3 (maximum characters is 71).
8/ Enter the number of number of labels, if labels are to be printed, or 1 for a letter head.
9/ Enter the required spacing for labels or envelopes, or 1 for letter heads .
10/ Enter the overstrike option, to overstrike, enter N. The image will be printed on the left hand side of the four entered text lines.
11/ Select 0 (zero) to return to the Process Menu.

I think that envelope printout would require tractor feed mounted envelopes, as per labels.

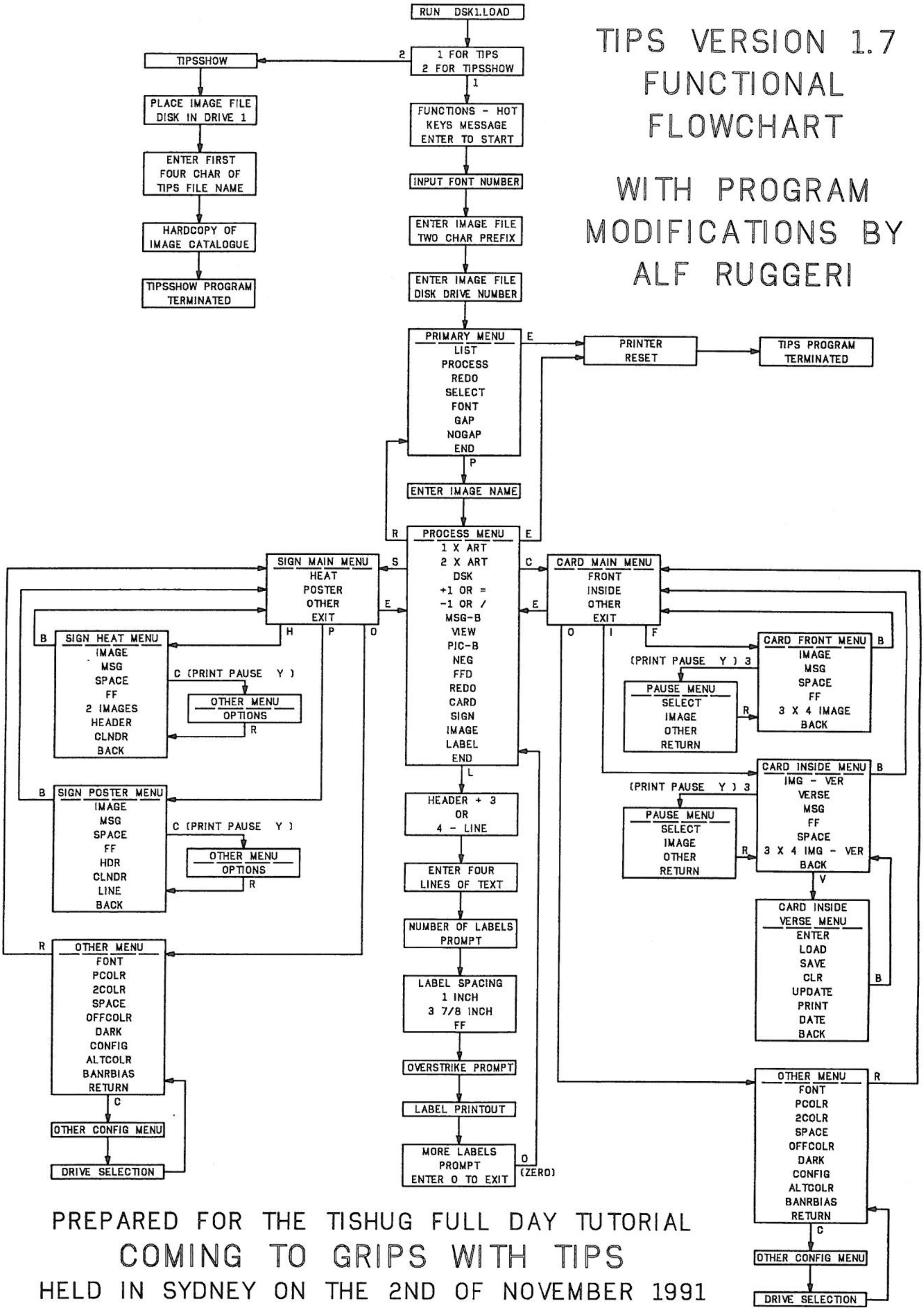
"Other Menu" Options

There are four multicolour options available, upon which, I wish I could make some meaningful comment, but not having access to a colour printer, it must remain a venture for others to pursue.

continued on page 24

TIPS VERSION 1.7 FUNCTIONAL FLOWCHART

WITH PROGRAM MODIFICATIONS BY ALF RUGGERI



PREPARED FOR THE TISHUG FULL DAY TUTORIAL

COMING TO GRIPS WITH TIPS

HELD IN SYDNEY ON THE 2ND OF NOVEMBER 1991

Printing Graphics with a Tandy DMP-105

by Geoff Trott

The second part of this article discusses solving the problem by using assembler routines called from Extended BASIC. Ross Mudie has had a series of excellent articles on this general topic which I recommend you read for more information. I will just present my solution to this particular problem, which speeds up the processing required from more than 1 hour to about 5 minutes.

As a reminder, the printed output from Page Pro uses the printer in graphics mode and expects the printer to respond to Epson type commands. This means that 8 pixels are printed vertically at the same time as the print head moves across the page. The DMP-105 only prints 7 pixels vertically as it moves across the page so that 7 sweeps across the page for the Epson type printers must be done in 8 sweeps for the DMP-105. To make things a bit more difficult, the pixel positions in the data sent to the printer must be reversed. For example, to print two dots in the first and third rows of a sweep for an Epson type printer, the binary code 10100000 would be sent. To do the same for the DMP-105 the binary code would be 10000101, where the left most 1 is always present for graphics data. The DMP-105 also needs a different sequence of control codes to enter and leave graphics mode and to set up the density. These different commands are rather minor compared to the problems of manipulating the data.

The Extended BASIC main program is similar in structure to that presented last month. The differences are in the processing of the data, which is done by a call to the assembler subroutine. This routine has 4 arguments, all of them strings or character arrays. The first of these is the data read from the input file (the printed output of Page Pro), the second is the output to be printed, the third string is the one which carries the bits left over from one sweep to the next and so is input and output and the fourth is a single character which tells the routine how many shifts of the data are required. When one shift is required, the third string is initialised to be empty and when seven shifts are required the third string is processed to provide the data for every eighth sweep. The fourth argument is a character to make the passing of a small number a bit more simple than passing a floating point number. The listing of the program follows with comments interspersed to help your understanding.

```
100 ! SAVE DSK1.GPRINT
110 ! A program to convert the output from Page Pro to a
    form suitable for a DMP-105 printer
120 ! Written by Geoff Trott for David Peters, both of
    TIsHUG, Australia, October 1991
128 ! dimension arrays of strings
129 ! each row of data takes 10 strings
130 DIM A$(11),B$(11),C$(11)
139 ! set up default input file and output device names
140 FI$="DSK2.PPPIC" :: FO$="PIO.CR.LF"
150 CALL CLEAR :: DISPLAY AT(1,4):"Graphics data
    converter" :: DISPLAY AT(2,2):" For Page Pro output
    files" :: DISPLAY AT(3,4):"and DMP-105 printers"
160 DISPLAY AT(4,3):"Written by Geoff Trott"
169 ! load in assembler routine using named disk
170 CALL INIT :: CALL LOAD("DSK.DMP-105.GPRINT;O")
180 DISPLAY AT(6,2):"Input file name" ::
    DISPLAY AT(7,1):FI$
189 ! get input file name
190 ACCEPT AT(7,1)SIZE(-28)
    VALIDATE(UALPHA,DIGIT,".@$*_/;"):IN$
199 ! get output device name
200 DISPLAY AT(9,2):"Output device name" ::
    DISPLAY AT(10,1):FO$ :: ACCEPT AT(10,1)SIZE(-28)
    VALIDATE(UALPHA,DIGIT,".@$*_/;"):OU$
209 ! open input and output
210 OPEN #1:IN$,DISPLAY ,INPUT ,VARIABLE 254 ::
    OPEN #2:OU$,DISPLAY ,OUTPUT,VARIABLE 96
219 ! read in first few strings from file and send out
220 LINPUT #1:D$ :: PRINT #2:CHR$(27);CHR$(21);
228 ! commands to set up DMP-105 for graphics mode
229 ! depending on density determined by number of data
230 LINPUT #1:D$ :: A=ASC(SEG$(D$,4,1))::
    B=ASC(SEG$(D$,3,1)):: LL=256*A+B :: NS=10 ::
    NC=LL/NS ::
    PRINT #2:CHR$(27);CHR$(18+NC/48);CHR$(18)
239 ! 99 lines done in groups of 7; start of main loop
240 FOR I=1 TO 99/7 :: FOR J=1 TO NS :: LINPUT #1:A$(J)
249 ! do first processing from A$ to B$ remainder in C$
250 CALL LINK("SHUFFL",A$(J),B$(J),C$(J),"1")
259 ! print first row
260 PRINT #2:B$(J)
270 NEXT J
279 ! D$ is carriage return line feed for next row
280 LINPUT #1:D$ :: PRINT #2:D$ :: LINPUT #1:D$
290 FOR J=1 TO NS :: LINPUT #1:A$(J)
299 ! input from C$ and A$ to output B$, remainder C$
300 CALL LINK("SHUFFL",A$(J),B$(J),C$(J),"2")
309 ! print second row
310 PRINT #2:B$(J):: NEXT J
320 LINPUT #1:D$ :: PRINT #2:D$ :: LINPUT #1:D$
330 FOR J=1 TO NS :: LINPUT #1:A$(J)
340 CALL LINK("SHUFFL",A$(J),B$(J),C$(J),"3")
349 ! print third row
350 PRINT #2:B$(J):: NEXT J
360 LINPUT #1:D$ :: PRINT #2:D$ :: LINPUT #1:D$
370 FOR J=1 TO NS :: LINPUT #1:A$(J)
380 CALL LINK("SHUFFL",A$(J),B$(J),C$(J),"4")
389 ! print fourth row
390 PRINT #2:B$(J):: NEXT J
400 LINPUT #1:D$ :: PRINT #2:D$ :: LINPUT #1:D$
410 FOR J=1 TO NS :: LINPUT #1:A$(J)
420 CALL LINK("SHUFFL",A$(J),B$(J),C$(J),"5")
429 ! print fifth row
430 PRINT #2:B$(J):: NEXT J
440 LINPUT #1:D$ :: PRINT #2:D$ :: LINPUT #1:D$
450 FOR J=1 TO NS :: LINPUT #1:A$(J)
460 CALL LINK("SHUFFL",A$(J),B$(J),C$(J),"6")
469 ! print sixth row
470 PRINT #2:B$(J):: NEXT J
480 LINPUT #1:D$ :: PRINT #2:D$ :: LINPUT #1:D$
490 FOR J=1 TO NS :: LINPUT #1:A$(J)
500 CALL LINK("SHUFFL",A$(J),B$(J),C$(J),"7")
509 ! print seventh row followed by eighth row
510 PRINT #2:B$(J):: NEXT J
520 LINPUT #1:D$ :: PRINT #2:D$ :: FOR J=1 TO NS ::
    PRINT #2:C$(J):: NEXT J :: PRINT #2:D$ ::
    LINPUT #1:A$(1)
530 NEXT I
539 ! finished main loop, just do last row
540 FOR J=1 TO NS :: LINPUT #1:A$(J)
550 CALL LINK("SHUFFL",A$(J),B$(J),C$(J),"1")
559 ! print seven bits of last row
560 PRINT #2:B$(J):: NEXT J
570 PRINT #2:D$ :: S$=CHR$(0):: T$=CHR$(128)::
    U$=CHR$(129)
580 FOR J=1 TO NS :: FOR K=1 TO 48
589 ! print last bit either on (U$) or off (T$)
590 IF SEG$(C$(J),K,1)=S$ THEN PRINT #2:T$;ELSE
    PRINT #2:U$;
600 NEXT K :: PRINT #2 :: NEXT J
610 PRINT #2:D$;CHR$(30):: CLOSE #1 :: CLOSE #2
620 STOP
```

The processing required is bit shifting. That is the pattern of bits (or binary digits) need to be moved to the right a certain number of positions. This can be done in BASIC by dividing by a power of 2. Dividing by 2 moves the bit pattern one position to the right, while dividing by 8 moves the bit pattern three positions to the right. Think of decimal numbers and dividing by powers of 10 if that helps. Dividing numbers is quite a lengthy process in BASIC while in assembler shifting is one of the intructions available for use. So it makes a lot of sense to do the processing of the 47520 bytes of data in assembler. In assembler the byte of data is loaded into the left half of the 16 bit register and then shifted the desired number of positions which will move part of it into the lower byte of the register. No bits get lost in this process and the two items of data are nicely divided into the two bytes in the register. Reversing the order of the bits is a bit more difficult and shows up one of the few inadequacies of the TI9900 assembler language. What needs to happen is that the data is moved to the right out of one register and shifted left into another register. In most processors this is done using the carry bit. This bit holds the bit shifted out of the register and usually it can be the bit shifted into another register. The TI9900 processors use the carry to catch the bit falling out of the register but there is no simply way to feed it into the other register. What I have done is to shift a zero into the output register least significant bit, to shift the bit out of the input register into the carry, test the carry and if it is one, increment the output register (set the least significant bit to 1). This is a bit messy but would be worse if we wanted to shift into the most significant bit. The code for the routine follows now with a few comments to help your understanding.

```
       DEF SHUFFL
*
*  CALL LINK("SHUFFL",A$(J),B$(J),C$(J),"N")
*
*      A$(J) input string
*      B$(J) output string for printer
*      C$(J) input carried over, output bits to be
*            carried over
*      "N" number of shifts required
*
*      Written by Geoff Trott, TIsHUG, for David Peters
*      For use with Page Pro output files and DMP-105
*      printers    October 1991
*      REF  STRREF,STRASG     for BASIC
*
FAC    EQU  >834A
STRREF EQU  >2014             for Extended BASIC
STRASG EQU  >2010             for Extended BASIC
*
BUFFA  BYTE 96
       BSS  99
BUFFB  BYTE 96
       BSS  99
BUFFC  BYTE 96
       BSS  99
BUFFN  BYTE 3
       BSS  3
*
* move the data into the buffers from the BASIC program
*
SHUFFL
       CLR  R0
       LI   R7,>6000           initialise size to 96
       LI   R1,1
       LI   R2,BUFFA
       MOVB R7,*R2
       BLWP @STRREF           move A$ into buffer
       CLR  R0
       LI   R1,3
       LI   R2,BUFFC
       MOVB R7,*R2
       BLWP @STRREF           move C$ into buffer
       CLR  R0
       LI   R7,>300
       LI   R1,4
       LI   R2,BUFFN
       MOVB R7,*R2
       BLWP @STRREF           move "N" into buffer
       MOVB @BUFFN+1,R0
       SWPB R0
       ANDI R0,15             convert to number
       MOV  R0,@FAC           save number of shifts
       CLR  R1                zero R1 for later use
       DEC  R0
       JNE  MAIN              jump if n>1
*
* if N=1 set up C$ to be empty and the same size as A$
*
       LI   R2,BUFFC
       MOVB @BUFFA,R0         R0 must have been zero
       MOVB R0,*R2+           buffer C same size as A
       SWPB R0
LOOP
       MOVB R1,*R2+           R1 is zero
       DEC  R0 C
       JNE  LOOP
MAIN
       LI   R2,BUFFA
       MOVB *R2+,R1           R1 was zero
       LI   R3,BUFFB
       MOVB R1,*R3+           set up string length
       LI   R4,BUFFC+1
       SWPB R1                get number of bytes
*
* shift data in A$ the number of places specified and
* add to C$.  Put the bits shifted out of A$ into C$.
* reverse the 7 bits to be printed into B$.
*
BUFLP
       MOV  @FAC,R0           get number of shifts
       CLR  R5
       MOVB *R2+,R5           data from A$
       CLR  R6
       MOVB *R4,R6            data from C$
       SRL  R5,0              shift no of times in R0
       AB   R5,R6             add to C$
       SWPB R6                start bit reversal
       CLR  R7                into R7
       INC  R7                set ms bit first
       LI   R0,7              bit counter
BACK
       SLA  R7,1              shift in
       SRL  R6,1              shift out
       JNC  SKIP              skip if not 1
       INC  R7                set to 1
SKIP
       DEC  R0                do 7 times
       JNE  BACK
       SWPB R7                ready to save
       MOVB R7,*R3+           data to B$
       SWPB R5                remainder
       SRL  R5,1              position bits
       MOV  @FAC,R0
       CI   R0,7
       JNE  STORE             jump unless last one
*
* after doing a 7 bit shift, the contents of C$ are the
* last row to be printed so reverse the bits.
*
       SWPB R5
       CLR  R7
       INC  R7                set ms bit
REVER
       SLA  R7,1              do bit reversal
       SRL  R5,1
       JNC  SKIP1
       INC  R7
SKIP1
       DEC  R0
       JNE  REVER
       SWPB R7
       MOV  R7,R5             put in correct register
STORE
       MOVB R5,*R4+           data to C$
       DEC  R1
       JNE  BUFLP
*
* All finished so send the data back to B$ and C$
*
       LI   R1,2
       LI   R2,BUFFB
       CLR  R0
       BLWP @STRASG           send B$ back
       LI   R1,3
       LI   R2,BUFFC
       CLR  R0
       BLWP @STRASG           send C$ back
       RT
       END
```

The first version of the program did not quite work but showed promise. This is the next attempt which should do better but needs a good checkout for any remaining problems. David Peters is doing this and I will report any further developments as they occur. O

Hollywood Hijinx part 1

by Scorpia, Copyright 1987 USA

Welcome to the wild world of "B" movies and "B" adventures! (grin) Your Uncle Buddy, king of the the "B" flicks ("Vampire Penguins of the North", "Atomic Chihuahuas from Hell", etc) may not be around anymore, and dear Aunt Hildegarde (world famous rose-grower and party-giver) may have just kicked the bucket, but that is all to the good for you. All you have to do is spend one night in their beachfront bungalow, find ten "treasures", and you will be a millionaire. What could be easier?

Probably plenty of things. Still, that kind of money is worth a little effort (maybe even a lot of effort!). Besides, if you do not get the inheritance, your vile cousin, Herman (the Horrible) will probably get it instead, and that is something you REALLY do not want!!

So, the funeral's over, Aunty Hildy has been laid to rest, and you are standing outside the beach house, next to a statue of Buck Palace, the fighting mailman. There are many ways to go from here; most of the puzzles in the game can be solved independently of each other (except the last two), and you can do the adventure in almost any order. This walkthru shows you only one way of finishing the game; there are others. Feel free to experiment along the way (make sure you save first, tho!).

Let us see if we can get into the house. The door, of course, is locked, and naturally you do not have the key. Or do you? If you read the back of Uncle Buddy's photo (which contains poetry to rival a Vogon's in its awfulness), you will note something about the movie titles; each has a direction in it. Specifically, west, east, and north. Turn the Buck Palace statue to those directions in that order. Aha! A click from the north! (By the way, you can drop both the photo and the letter from Aunt Hildegarde; you will not be needing either of them after this).

Going in that direction yourself, you end up on the front porch, with a mailbox nearby. Inside the mailbox (open it first!) is a yellow paper, a card, and a copy of The Status Line. The card is a business card with Roy G. Biv on it, and the yellow paper has odd lines with an X in the middle. The Status Line is just a prop, and you can leave it in the box (or read it and drop it). You can also drop the card after looking at it, but hold on to the paper; it is one half of the map of the hedge maze.

Open the door, turn on your flashlight, and go north into the Foyer. There is a closet here, and a very interesting one at that. In fact, it is more than just a closet; it is also an elevator, which is about to come in handy. Open the closet door and step inside. Leave the skis and bucket for now; pull the third peg on the wall, which will bring you up to the second floor.

Open the door and step out north. You are in the middle of the hallway. Overhead is a panel in the ceiling; you do not need to worry about that for the moment. Turn the newel by the stairway, to fix the steps in place so you can now go up and down the stairs. Then go west and south to the overdone Hollywood bathroom.

Overdone may be an understatement here (there is no accounting for some people's tastes in decoration). However, you need not stay long. Just look under the mat and pick up the red punchcard you find there. You will be collecting more of them as the game proceeds.

Now go north and all the way east to the end of the hallway. Hmmm. Looks like there is something stuck in the window. Be careful here! Grab hold of the bag first, THEN open the window. Aha! Opening the bag reveals a Maltese Finch, chief prop from the movie of the same name. You have found your first treasure!

Now return to the closet, and pull the second peg to bring you back down to the Foyer. Get the bucket and skis, then leave the closet. Drop the skis in the Foyer (saving time for later on), then north again to the Game Room. Unlock and open the patio door, then go west to the Dining Room.

Here you find a thin sheet of paper, also with strange lines and an X on it. If you put the thin paper over the yellow one, you have a complete map of the hedge maze! This is a good place to turn on the script, and run off a copy of the map on your printer; it will be helpful in getting through the maze, which you will be doing shortly.

Return to the Game Room and go out the Patio Door. Pick up the orange punchcard that is lying there, and continue north into the beautiful garden. However, you will have to admire the flowers another time, there is much to do yet and the game has just begun.

Go NW to the West Garden and take the shovel, then NE to the North Garden. Note that there is a small pond here; that will be important later. Continue North, and you are at the entrance to the Hedge Maze. Since you have the map, you do not need specific directions for this. Just be advised that it is a long trip both ways, so go carefully. When you reach the X spot, dig with the shovel. You uncover a rubber stamp, from that terrific Buck Palace movie, "Address Unknown". Another treasure!

Drop the shovel, and make your tedious way back through the maze to the entrance (you will note from the map there is only one way in or out; too bad). Go south to the North Garden, and fill the bucket with water. Being rusty, it will leak slowly, but if you go straight back to the house, you will have enough time.

Go right from the patio to the Game Room and from there to the closet in the Foyer. Put the bucket on the third peg, then leave and immediately go upstairs to the middle hallway, open the door, and step into the shaft and wait.

Since you are on top of the closet, which is now rising towards the second floor, you are actually on your way to the third floor. When you get there, open the door, and step into the dusty attic. Now you are over the spot where you saw the panel from the second floor earlier. And sure enough, there is the panel, with handy ladder attached. You also note an old trunk. Unfortunately, it does not seem to want to open.

However, that is no problem. When you open the panel, the trunk lock also opens. Now you can get the treasure inside, the fire hydrant from that eerie thriller, "Atomic Chihuahuas From Hell". With hydrant in hand, go down the steps to the second floor hallway, then down the stairs, and make your way to the Living Room, where you can drop off the treasures you have found so far. However, you will be needing that cloth sack later, so take the finch out of the sack, and just drop off the finch.

Still in the Living Room, enter the fireplace. You note immediately that one of the bricks seems loose. Take it and drop it; an indigo punchcard falls to the floor. Boy, they sure seem to be all over the place! Get the card, then start climbing up, up, up, until you reach the roof (there is nothing in the other holes along the way).

Now hop over to the other chimney and go down. You cannot go too far, but that does not matter. You have found another treasure, a life-size prop from that chilling motion picture, "Vampire Penguins of The North". Take the penguin, then go back the way you came to the Living Room, and drop off your latest find.

Hollywood Hijinx is copyrighted 1987 by Infocom Inc.

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

Reviews

by Stephen Shaw, England

REVIEW: THE RING COMPANION.

Consists of 2 disks. NOTUNG SOFTWARE. US$8+2.50 POST. This two disk set is a companion to the music disks by Ken Gilliland, "RICHARD WAGNERS GREATEST HITS" and "RICHARD WAGNERS DER RING DES NIBELUNGEN". We managed no less than three version of The Ring on TV over Christmas/New Year, so it may be a little more familiar to you than last year! This disk set contains:

A huge picture of The Ring, with a dwarf and Rheinmaiden, composed of eight TI artist pics-formatted for printing with TI Artist Plus! but any other _P printer will do. Plus a pic of Richie Wagner. Plus 12 pictures and instances of Ring cartoons by Arthur Rackham. Plus some text about The Ring (roots, characters, Wagner). Plus a program to demonstrate 30 "leitmotifs". Quote a package for lovers of Wagner.

REVIEW: SON OF THE DISK OF DINOSAURS.

This also consists of 2 disks. NOTUNG SOFTWARE, $12+2.50 POST. A follow up to the Disk of Dinosaurs published by Asgard, this disk set contains a further landscape to place dinosaurs into, and 25 TI artist instances of individual dinosaurs, which Notung Software used to very good effect in a very unusual Christmas card! There is a further 62 frame cartoon adventure with Thug, but this time using Comic Show 4.0, so you cannot slow down the rather rapid movement! Plus text on the dinosaurs, and a quiz which allows you to build a dinosaur by answering correctly ten questions. THE source for a TI Artist instance of a Pachycephalosaurus!

REVIEW: BOOK: THE TURING OMNIBUS: 61 Excursions in Computer Science, by A K Dewdney. Hardback, 415 pages. W H FREEMAN AND CO. ISBN 0 7167 8154 9

We have previously reviewed the earlier book, The Armchair Universe, and this one follows on from that, but this time the articles are a little heavier and we do not have the simple algorithms to give us fast graphics output. Rather a set of articles loosely connected to computing, the reading and understanding of which just may help you to resolve computing problems faster or easier. Something of a recreational math book. Some of the chapter headings include:

Error correcting codes; karnaugh maps; the fastest sorts of sorts; the chaitin-kolmogoroff theory.... This book is intended as a "semi popular compendium of computer science" rather like a touring bus (pun intended) visiting selected landmarks of computer science. You have been warned!

REVIEW: BOOK: THE MAGIC MACHINE: A HANDBOOK OF COMPUTER SORCERY, by A. K. Dewdney, paperback, 357 pages. W.H. Freeman. ISBN 0 7167 2144 9

This 1990 book is a little easier to take, it has a few easy to translate algorithms to turn into programs, and in theory 16 out of 28 chapters could give rise to a program of some sort, although some could be hard to sort out! Once more lots of general reading to improve your approach to problems! Including even a mention of transactional analysis; other headings include:

Balls in boxes; Trains of thought (some classic recreational math puzzles); Palmiters protozo; Catching biomorphs.

This book is a collection of recreational material that was previously published in Scientific American under the heading "computer recreations".

REVIEW- DISK-GAME- BACKSTEINE by Quinton Tormanen. US$10 plus $4 post from COMPRODINE, 1949 Evergreen Avenue, Fullerton, Ca, USA, 92635.

Well that it the worst thing about this offering out of the way anyway, with a high score of zero out of ten for the awful name (unless you are German, or speak the language, when you will know the title is BRICKS). This program was written in an English speaking country (sort of) for sale in same. No excuse.

It is a Breakthrough type program- where you move a bat at screen bottom and a bouncing ball batters away at a wall of bricks, and if the ball hits the floor it is lost.

Not original. However the IMPLEMENTATION is absolutely superb, well up to modern arcade standards, with the ball and some bricks throwing shadows on a patterned background. Bricks take one two or three hits to knock out, and may drop one of six different bonus prizes, which make your bat bigger or affect the ball in some way. There are also some bricks which cannot be destroyed. The game comes with an assortment of fifty screens (made up of different patterns of bricks) and there is an editor to make your own up.

Game play is with joystick one for one player. Editing can be with joystick one, or with more control (eg slower) with joystick 2 or even ESDX keys.

The graphics and implementation of this game are exceptional. You may not be playing it none stop for the next year, but it is an excellent game to while away an hour or so every now and then, and contemplate just what our ancient machine is capable of... and was capable of in 1980. Why did TI not release modules of this quality! Highly recommended.

======

REVIEW- EXTENDED BUSINESS GRAPHS from Comprodine $7 plus $4 post.

This program originated with Great Lakes Software. It is quite old, and rather shows its age. Basically you key in your data, and have a few graphing options. Graphs can be printed out in a single format, and CANNOT BE SAVED TO DISK! in any format.

A little restricted perhaps, but there are not many graphing programs available, and this new price is not bad. I will supply illustrations of ALL the types of graph it is capable of! which illustrate what it can do better than mere words.

REVIEW- PICTURE IT- disk from COMPRODINE for US$10.

A graphics utility program with several options, none of them very fast, but quite useful for all that.

Dating back to 1987, this program is by Rodger Merritt, who runs Comprodine.

The program disk has several utilities...

BANNERS... to print instances up to 12 lines high and/or text in a specially supplied 8 inch high font, sideways, much enlarged. You get to choose which ASCII character represent a "pixel" so there is plenty of room to vary contrast. Banners seen at Romiley and Chester AGMs were prepared with this. Instances can be text prepared with any TI Artist font of course!

INSTANCES... can transfer a TI Artist instance into an XB program, either as redefined characters or as defined sprites. The output file is a merge format DV163 file.

TI WRITER... and you can save instances to disk in a format which prints out via the TI Writer FORMATTER (uses a lot of >TL's). If you need the program, you need it!

continued on page 32

Jenny's Younger Set

There has not been too much for the Younger Set this year apart from what Vincent Maker has put in. He has come up with some nice ones for the last issue for the year and appropriate for Christmas and New Year. Why not some of you young people out there making a New Years resolution to send in an article or two for the magazine in 1992? It would be most welcome and you can put it on a cassette and give to the Editor or one of the directors at one of the meetings. I wish all of you a Merry Christmas and an enjoyable New Year. Here is the letter and the programs sent in by Vincent.

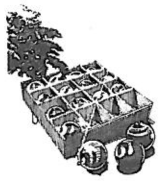
Dear Jenny,

First of all, MERRY CHRISTMAS to you and all the younger set. I have two programs for you this time, one being a musical program and one being the NEW YEARS PROGRAM which, if used on NEW YEARS EVE, can be pressed a second before and will come up.

Wishing a MERRY CHRISTMAS to all,

VINCENT MAKER

```
100 CALL CLEAR
110 PRINT "1.WE WISH YOU A MERRY XMAS."
120 PRINT "2.AS GENTLE AS SILENCE."
130 PRINT "3.ALL"
140 PRINT "4.END"
150 CALL KEY(0,K,L)
160 IF L=0 THEN 150
170 IF K=49 THEN 210
180 IF K=50 THEN 770
190 IF K=51 THEN 210
200 END
210 CALL SOUND(250,147,0)
220 CALL SOUND(250,196,0)
230 CALL SOUND(175,196,0)
240 CALL SOUND(175,220,0)
250 CALL SOUND(175,196,0)
260 CALL SOUND(175,175,0)
270 CALL SOUND(250,165,0)
280 CALL SOUND(250,165,0)
290 CALL SOUND(250,165,0)
300 CALL SOUND(250,220,0)
310 CALL SOUND(175,220,0)
320 CALL SOUND(175,247,0)
330 CALL SOUND(175,220,0)
340 CALL SOUND(175,196,0)
350 CALL SOUND(250,175,0)
360 CALL SOUND(250,147,0)
370 CALL SOUND(250,175,0)
380 CALL SOUND(250,247,0)
390 CALL SOUND(175,247,0)
400 CALL SOUND(175,262,0)
410 CALL SOUND(175,247,0)
420 CALL SOUND(175,220,0)
430 CALL SOUND(250,196,0)
440 CALL SOUND(250,196,0)
450 CALL SOUND(250,165,0)
460 CALL SOUND(175,147,0)
470 CALL SOUND(175,147,0)
480 CALL SOUND(250,165,0)
490 CALL SOUND(250,220,0)
500 CALL SOUND(250,175,0)
510 CALL SOUND(497,196,0)
520 CALL SOUND(250,147,0)
530 CALL SOUND(250,196,0)
540 CALL SOUND(250,196,0)
550 CALL SOUND(250,196,0)
560 CALL SOUND(497,175,0)
570 CALL SOUND(250,175,0)
580 CALL SOUND(250,196,0)
590 CALL SOUND(250,175,0)
600 CALL SOUND(250,165,0)
610 CALL SOUND(497,147,0)
620 CALL SOUND(250,220,0)
630 CALL SOUND(250,247,0)
640 CALL SOUND(175,220,0)
650 CALL SOUND(175,220,0)
660 CALL SOUND(175,196,0)
670 CALL SOUND(175,196,0)
680 CALL SOUND(250,294,0)
690 CALL SOUND(250,147,0)
700 CALL SOUND(175,147,0)
710 CALL SOUND(175,147,0)
720 CALL SOUND(250,165,0)
730 CALL SOUND(250,220,0)
740 CALL SOUND(250,175,0)
750 CALL SOUND(497,196,0)
760 IF K=51 THEN 770 ELSE 100
770 CALL SOUND(250,131,0)
780 CALL SOUND(250,165,0)
790 CALL SOUND(497,175,0)
800 CALL SOUND(250,220,0)
810 CALL SOUND(250,175,0)
820 CALL SOUND(497,147,0)
830 CALL SOUND(375,175,0)
840 CALL SOUND(175,165,0)
850 CALL SOUND(250,175,0)
860 CALL SOUND(747,175,0)
870 CALL SOUND(250,131,0)
880 CALL SOUND(250,165,0)
890 CALL SOUND(497,175,0)
900 CALL SOUND(250,220,0)
910 CALL SOUND(250,175,0)
920 CALL SOUND(497,147,0)
930 CALL SOUND(375,175,0)
940 CALL SOUND(175,165,0)
950 CALL SOUND(1250,175,0)
960 FOR T=0 TO 250
970 NEXT T
980 CALL SOUND(250,175,0)
990 CALL SOUND(250,220,0)
1000 CALL SOUND(497,262,0)
1010 CALL SOUND(250,294,0)
1020 CALL SOUND(250,247,0)
1030 CALL SOUND(497,196,0)
1040 CALL SOUND(250,165,0)
1050 CALL SOUND(497,196,0)
1060 CALL SOUND(250,165,0)
1070 CALL SOUND(250,196,0)
1080 CALL SOUND(497,247,0)
1090 CALL SOUND(747,262,0)
1100 CALL SOUND(175,175,0)
1110 CALL SOUND(497,165,0)
1120 CALL SOUND(250,131,0)
1130 CALL SOUND(250,165,0)
1140 CALL SOUND(497,175,0)
1150 CALL SOUND(250,220,0)
1160 CALL SOUND(250,175,0)
1170 CALL SOUND(497,147,0)
1180 CALL SOUND(425,175,0)
1190 CALL SOUND(175,165,0)
1200 CALL SOUND(250,175,0)
1210 CALL SOUND(425,175,0)
1220 GOTO 100
```



This one is the New Year's Program. It has an interesting use of colours and just right for welcoming in the New Year. Thank you Vincent for putting the effort into it.

```
1 CALL MAGNIFY(2)
100 CALL CLEAR
110 INPUT "WHEN YOU ARE READY TO RUN THE NEW YEARS
PROGRAM PRESS ENTER ":JKL$
120 CALL CLEAR
125 CALL SCREEN(1)
130 CALL SPRITE(#1,49,4,50,50,0,0)
140 CALL SPRITE(#2,57,3,50,70,0,0)
145 CALL SPRITE(#4,57,10,50,90,0,0)
150 CALL SPRITE(#3,50,8,50,110,0,0)
160 FOR T=0 TO 13
170 CALL COLOR(T,12,2)
180 NEXT T
190 DISPLAY AT(5,7):"** HAPPY NEW YEAR **"
200 GOTO 200
```

○

continued from page 32

I shall try to reduce a certificate for illustration purposes, but we need rather a large reduction so it may not work out too well... If you need to produce certificates, this is the most directly applicable TI program, and pretty inexpensive too! ○

```
1 REM COPYRIGHT 1984, EMERAL
D VALLEY PUBLISHING CO.
100 REM ****************
110 REM * MISSILE MATH *
120 REM ****************
130 REM BY J.C. ADELMANN
140 REM AND THE HCM STAFF
150 REM HOME COMPUTER MAGAZI
NE
160 REM VERSION 4.3.1
170 REM TI-BASIC
180 REM TI-EXTENDED BASIC
190 CALL CLEAR
200 DIM MS1(22),MS2(12)
210 RESTORE 2000
220 FOR I=1 TO 22
230 READ MS1(I)
240 NEXT I
250 FOR I=1 TO 12
260 READ MS2(I)
270 NEXT I
280 PRINT TAB(9);"MISSILE MA
TH"
290 PRINT TAB(3);"DO YOU NEE
D INSTRUCTIONS?   (Y/N)"
300 INPUT R$
310 IF R$="Y" THEN 1800
320 CALL CLEAR
330 GOSUB 1060
340 CALL SCREEN(11)
350 INPUT "WHAT NUMBER DO YO
U WANT TO  PRACTICE WITH? (1
-9)":S
360 IF (S<1)+(S>9)THEN 350
370 Y$=STR$(S)
380 INPUT "DO YOU WANT A TIM
ES TABLE?  (Y/N) ":R$
390 IF (R$="N")+(R$="n")THEN
 490
400 CALL CLEAR
410 FOR K=0 TO 9
420 PRINT "       ";S;" * ";K
;" = ";S*K
430 PRINT
440 NEXT K
450 PRINT "   PRESS ENTER TO
 CONTINUE"
460 CALL KEY(0,RET,STA)
470 IF STA=0 THEN 460
480 IF RET<>13 THEN 460

490 REM DOMATH2
500 CALL CLEAR
510 Q=0
520 E=0
530 CALL CLEAR
540 CALL SCREEN(6)
550 CALL HCHAR(18,1,106,224)
560 FOR X=1 TO 10
570 RANDOMIZE
580 R(X)=INT(10*RND)
590 T=0
600 ANS=INT(3*RND)+1
610 FOR J=1 TO X-1
620 IF R(X)=R(J)THEN 640
630 GOTO 650
640 T=1
650 NEXT J
660 IF T=1 THEN 580
670 W$=STR$(R(X))
680 B=R(X)*S
690 C$=STR$(B)
700 L=LEN(STR$(B))
710 U=ASC(W$)
720 V=ASC(Y$)
730 GOSUB 1270
740 CALL HCHAR(21,19,106,2)
750 CALL HCHAR(19,24,106)
760 FOR I=6 TO 28
770 CALL HCHAR(22,I,106)
780 NEXT I
790 CALL HCHAR(21,15,U)
800 CALL HCHAR(21,16,42)
810 CALL HCHAR(21,17,V)
820 CALL HCHAR(21,18,61)
830 E=0
840 FOR I=11 TO 22
850 CALL HCHAR(19,I,MS2(I-10
))
860 NEXT I
870 CALL KEY(3,RET,STA)
880 IF (RET<65)+(RET>67)THEN
 870
890 CALL HCHAR(19,24,RET)
900 IF (RET-64)<>ANS THEN 92
0
910 E=1
920 GOSUB 1920
930 Q=Q+E
940 NEXT X
950 FOR DELAY=1 TO 500
960 NEXT DELAY
970 CALL SCREEN(11)
980 CALL CLEAR
990 PRINT "NUMBER RIGHT IS";
Q
1000 PRINT "NUMBER WRONG IS"
;10-Q
1010 PRINT "PERCENT RIGHT IS
";(Q/10)*100
1020 INPUT "DO YOU WISH TO P
LAY AGAIN?  (Y/N) ":D$
1030 IF (D$="Y")+(D$="y")THE
N 330
1040 CALL CLEAR
1050 END
1060 CALL COLOR(3,2,11)
1070 CALL COLOR(4,2,11)
1080 CALL COLOR(7,2,11)
1090 CALL COLOR(2,2,11)
1100 CALL COLOR(5,2,11)
1110 CALL COLOR(6,2,11)
1120 CALL COLOR(8,2,11)
1130 CALL CLEAR
1140 CALL SCREEN(2)
1150 RESTORE 1780
1160 FOR I=97 TO 102
1170 READ A$
1180 CALL CHAR(I,A$)
1190 NEXT I
1200 FOR I=104 TO 106
1210 READ A$
1220 CALL CHAR(I,A$)
1230 NEXT I
1240 CALL COLOR(9,10,1)
1250 CALL COLOR(10,11,1)
1260 RETURN
1270 FOR G=0 TO 2
1280 IF G<>ANS-1 THEN 1360
1290 IF L<>1 THEN 1330
1300 N=ASC(C$)
1310 M=106
1320 GOTO 1420
1330 M=ASC(SEG$(C$,1,1))
1340 N=ASC(SEG$(C$,2,1))
1350 GOTO 1420
1360 RANDOMIZE
1370 M=INT(10*RND)+48
1380 IF M<>48 THEN 1400
1390 M=106
1400 N=INT(10*RND)+48
1410 IF M*10+N=B THEN 1370
1420 IF E=1 THEN 1450
1430 CALL SOUND(200,110,0)
1440 GOTO 1580
1450 CALL SOUND(150,262,15,3
30,15,392,15)
1460 CALL SCREEN(5)
1470 CALL SOUND(150,262,15,3
30,15,392,15)
1480 CALL SCREEN(6)
1490 CALL SOUND(150,262,15,3
30,15,392,15)
1500 CALL SCREEN(7)
1510 CALL SOUND(150,294,15,3
49,15,440,15)
1520 CALL SCREEN(8)
1530 CALL SOUND(150,262,15,3
30,15,392,15)
1540 CALL SCREEN(9)
1550 CALL SOUND(150,294,15,3
49,15,440,15)
1560 CALL SCREEN(6)
1570 CALL SOUND(200,330,15,4
15,15,494,15)
1580 CALL HCHAR(G*4+3,20-Q,9
7)
1590 CALL HCHAR(G*4+3,20-Q+1
,32)
1600 CALL HCHAR(G*4+4,17-Q,1
04)
1610 CALL HCHAR(G*4+4,18-Q,9
9)
1620 CALL HCHAR(G*4+4,19-Q,1
06)
1630 CALL HCHAR(G*4+4,20-Q,G
+65)
1640 CALL HCHAR(G*4+4,21-Q,1
01)
1650 CALL HCHAR(G*4+4,21-Q+1
,32)
1660 CALL HCHAR(G*4+5,17-Q,1
05)
1670 CALL HCHAR(G*4+5,18-Q,1
00)
1680 CALL HCHAR(G*4+5,19-Q,M
)
1690 CALL HCHAR(G*4+5,20-Q,N
)
1700 CALL HCHAR(G*4+5,21-Q,1
02)
1710 CALL HCHAR(G*4+5,21-Q+1
,32)
1720 CALL HCHAR(G*4+6,20-Q,9
8)
1730 CALL HCHAR(G*4+6,20-Q+1
,32)
1740 FOR DELAY=1 TO 100
1750 NEXT DELAY
1760 NEXT G
1770 RETURN
1780 DATA "0103070F1F3F7FFF"
,"FF7F3F1F0F070301","03070F1
F3F7FFFFF","FFFF7F3F1F0F0703
","000000002060E0E0"
1790 DATA "E0E06020","000000
38381010FF","FF10103838","FF
FFFFFFFFFFFFFF"
1800 REM

1810 REM *INSTRUCTIONS*
1820 REM
1830 CALL CLEAR
1840 PRINT "MULTIPLICATION G
AME"
1850 PRINT
1860 PRINT "EACH MISSILE HAS
 AN ANSWER   CHOOSE A B OR C
"
1870 PRINT
1880 PRINT "PRESS ANY KEY TO
 CONTINUE"
1890 CALL KEY(0,KY,S)
1900 IF S=0 THEN 1890
1910 GOTO 320
1920 IF (LEN(C$)=1)THEN 1940
 ELSE 1930
1930 CALL HCHAR(21,20,ASC(SE
G$(C$,2,1)))
1940 CALL HCHAR(21,19,ASC(SE
G$(C$,1,1)))
1950 FOR I=6 TO 27
1960 CALL HCHAR(22,I,MS1(I-5
))
1970 NEXT I
1980 CALL HCHAR(22,28,ANS+64
)
1990 RETURN
2000 DATA 84,72,69,106,67,79
,82,82,69,67,84,106,65,78,83
,87,69,82,106,73,83,106
2010 DATA 89,79,85,82,106,65
,78,83,87,69,82,63
```

Rambles

by Stephen Shaw, England

EXTENDED BASIC BUG:

I am grateful to Bruce Harrison of Harrison Software for pointing this one out in the December issue of Micropendium- if you have a disk system, and have a disk of XB programs which has a program called LOAD, your system will automatically load the LOAD program from DSK1 when you select Extended Basic. It also kicks RANDOMIZE into touch, not only for the LOAD program but also for any program that the LOAD program loads and runs. If the computer manages to find a LOAD program, then your RND will always provide the same sequence every time you boot up, even if your program has a RANDOMIZE.

Try it! Put in a little program on your disk like this:

```
100 RANDOMIZE
110 FOR T=1 TO 6
120 PRINT INT(100*RND)
130 NEXT T
140 RUN "DSK1.LOAD"
```

and watch the six numbers repeat again and again, in defiance of the RANDOMIZE command.

Thus if you use a LOAD menu program to put in XB programs such as games or graphics or what have you, you may as well forget about RANDOMIZE... except...
Bruce has provided an answer- insert:

```
70 CALL INIT
80 CALL PEEK(-31880,[,])
90 CALL LOAD(-31808,[,])
```

and now try it... see, different numbers....
(Yes Virginia you can use [and] as variable names!)

NOTE that you must omit the CALL INIT if it has already been used for example to load an assembly utility such as The Missing Link else you will destroy the utility! Call Init is also not required if you use the Triton version of XB titled Super Extended Basic on the module label and (c)1987 TYPC.

Bruce tells us this fix can be attributed to Harry Wilhelm and dates back a couple of years.

From New Scientist Christmas issue...

Santa delivers to a very long road. He notices that at one house, when he removes the first digit and then squares the remaining digit he is back to the original number. Here, let us do this one for you:

125.......25 x 25 = 125

What is really quite amazing is that when he gets to the 25th house number which satisfies this rule, he finds it is his own house! What number is his house?

I have taken out a subscription to a pretty expensive magazine called RECREATIONAL AND EDUCATIONAL COMPUTING which at 30p per page is not too cheap- issues are bimonthly or so and you can receive some sample copies by sending US$15 to:

REC, 909 Violet Terrace, Clarks Summit, PA, USA, 18411.

Here is a puzzle from one issue:

In the land of make believe there is an eccentric jailer (gaoller?) who takes the following action on New Years Day:

At minute 0 all 100 doors are locked shot. At minute 1 he goes to every cell and opens the door. At minute 2 he goes to door 2 and if it is open, closes it. If it is shut he opens it. The he goes to door 4 and does the same thing and so on to every SECOND cell. At minute 3 he begins at cell 3, and reverses the state of its door, carrying on with every third cell (Reversing 3,6,9,12 etc). and so on and so on and so on. At minute 100 he just reverses cell 100.

Now then... how many prisoners can walk free- eg which cell doors are open?

With reference to the puzzle set on p53 of issue #30, regarding the three hands of a clock...

For ease of reference, refer to the position of ALL three hands in terms of the clock face reading in hours- that is if the second hand is pointing to the hour number 3, we refer to the position of that hand as 3. The easiest way to approach this problem is to consider firstly just two hands- the hour hand and the minute hand. Instinctively we can feel that these two hands for an angle of 120 degrees (or 4 hours!) about 24 times in 12 hours and then repeat.

When the hour hand moves from 12 to 1, a passage of one hour, the minute hand moves from 12 to 12, moving through 12 hours (hour numbers!). Thus the difference in movement between the two hands is 12-1=11n. Our problem requires that the difference between the hour and minute hands is 4 hours (on the clock face) so we must solve for 11n=4. Because a clock face is circular (you noticed!) there is more than one value of n we can use.

We can now write a program for values of n from one upwards until 12 hours have elapsed and the numbers repeat. It is necessary to continually reduce the number to keep it within the 12 hour period.

Then, having our 22 values when the hour and minute hands are exactly 4 hours apart we can examine the position of the second hand to see where it lies at these very limited number of times. If it is precisely four hours from each of the hour and minute hand, our problem is solved!

```
100 FOR N=4 TO 96 STEP 4
110 IF N/12=INT(N/12) THEN 180 ! hands coincide
120 HOURS=N/11
130 MINS=12*hours
140 IF MINS>12 THEN MINS=MIN-12 :: GOTO 140
150 PRINT HOURS ! note 0.09' is 1/11th
160 PRINT MINS
170 PRINT "====="
180 NEXT N
```

---- and then ----

```
1 REM THREE HANDED CLOCK
2 REM DATA=TIME ELAPSED FOR HOUR AND MINUTE HAND TO FORM
  120 DEG ANGLE EXPRESSED IN HOURS ELAPSED ON HOUR HAND
3 REM S SHAW JAN 1991
4 REM TI99/4A EXTENDED BASIC
5 REM
100 DATA 0.36363636
110 DATA 0.72727272
120 DATA 1.45454545
130 DATA 1.81818181
140 DATA 2.54545454
150 DATA 2.90909090
160 DATA 3.63636363
170 DATA 4.0
180 DATA 4.72727272
190 DATA 5.09090909
200 DATA 5.81818181
210 DATA 6.18181818
220 DATA 6.90909090
230 DATA 7.27272727
240 DATA 8.0
250 DATA 8.36363636
260 DATA 9.09090909
270 DATA 9.45454545
280 DATA 10.18181818
290 DATA 10.54545454
```

```
300 DATA 11.27272727
310 DATA 11.63636363
320 DATA 99,99
330 READ HOURS :: PRINT "======"
340 IF HOURS=99 THEN STOP
350 MINS=12*hours
360 IF MINS>12 THEN MINS=MINS-12 :: GOTO 360
370 PRINT SEG$(STR$(HOURS),1,6)&"---"&SEG$(STR$(MINS),
    1,6)
380 SECS=60*hours
390 IF SECS>1200 THEN SECS=SECS-1200 :: GOTO 390
400 IF SECS>60 THEN SECS=SECS-60 :: GOTO 400
410 PRINT SECS/5 ! CLOCK FACE READOUT IN HRS
420 COUNT=COUNT+1
430 IF INT(COUNT/8)=COUNT/8 THEN 440 ELSE 450
440 FOR PAUSE=1 TO 4000 :: NEXT PAUSE
450 GOTO 330
460 END
```

This is a pretty fast way of sorting this problem out. The ticking second hand? Well expressed in terms of the hour numbers, a second is 0.2 hr, so if one of the 22 solutions comes within 0.2 of satisfying our needs, maybe ticking does have some significance....?

Much harder query: If the perfect solution is not possible, can we find the time that the hands are closest to our requirement? For this purpose, differences are added regardless of their sign-

We require: 120 120 120
We get say: 119 118 123
Error 1 + 2 + 3 = error 6.

I do not have an answer to this one and look forward to hearing from you.

My thanks to Walter Allan for correspondence received on this one!

==============

CUFFLEY...

The meeting at CUFFLEY went well in January, although not as well attended by SOUTHERN users as could have been hoped. It was mentioned that Cuffley being actually North of London was to be considered a Northern show... hmmm. Those who did attend had a good chat amongst themselves and there were some real bargains to be picked up for those who wanted them.

Cuffley was attended by about 38 TI users - (The Sydney group in Australia, now starting its tenth year, also had 38 attending its AGM in December 90...!).

There was good representation from the DORTIG Dorset group, and Martin Blythe from Database was there.

I have also been taken to task for not pointing out in very blunt language that you really can still obtain a very very large number of items of software on module or disk, and almost every module ever made is still available from somewhere. Before you are tempted to look at another brand of computer, consider what software you need. There is a very good chance you can obtain an adequate working system for your TI, probably at a much lower cost!

MICROpendium has been mentioned in almost every issue. They review new items and tell you where to get them. There is also regular advertising of older material especially TI modules at low low prices.

==============

Some time ago I printed a membership application form for the Sydney group, which was utterly ignored by everyone. They now have a second UK member to join me! They publish a 24 page (size A4 but still small print!) magazine 11 times a year and overseas subs are A$45 by slow seamail or A$60 by airmail. Their articles include machine code, hardware, and Forth as well as some you are already familiar with.

BETTER BASIC PROGRAMS...

A simple puzzle:

I somehow manage to find a pound (100 pence) and wish to purchase items which are priced at 14p, 17p, 22p, and 39p. In what combinations may I purchase these items so that I exactly spend my pound?

We can quickly get our calculators out, work out the maximum number of each I can buy for a pound, and come up with code like this...

```
100 FOR A=0 TO 7
110 FOR B=0 TO 5
120 FOR C=0 TO 4
130 FOR D=0 TO 2
140 IF 14*A+17*B+22*C+39*D=100 THEN PRINT A;B;C;D
150 NEXT D
160 NEXT C
170 NEXT B
180 NEXT A
```

Can you follow the reasoning behind this code? This little program takes about 25.4 seconds to find the four possible answers. We know our console is not the fastest in the world, so can we speed up this work? The firs t improvement is interesting- curious too. This simple amendment:

```
100 FOR D=0 TO 2
110 FOR C=0 TO 4
120 FOR B=0 TO 5
130 FOR A=0 TO 7
140 IF 14*A+17*B+22*C+39*D=100 THEN PRINT A;B;C;D
150 NEXT A
160 NEXT B
170 NEXT C
180 NEXT D
```

produces nearly a 10% speed-up, and runs in about 23.2 seconds. Most odd.

We can however speed things up a great deal more by applying a little thought to the problem. If we buy a 39p item we have only 61p left which at most allows us to buy 3 items at 22p- in this case the C loop does not need to test for a value of 4. If we buy two 39p items we have only 22p left and can only buy one 22p item. The following code reduces the number of steps in the loop by taking account of this sort of limitation. The first line is slightly wasteful but will make the flow a little clearer:

```
100 FOR D=0 TO 100/39 ! think about it
110 FOR C=0 TO (100-39*D)/22
120 FOR B=0 TO (100-39*D-22*C)/17
130 A=INT((100-39*D-22*C-17*B)/14)
140 IF 14*A+17*B+22*C+39*D=100 THEN PRINT A;B;C;D
150 NEXT B
160 NEXT C
170 NEXT D
```

This code would you believe gives us the same four answers in just 2.82 seconds or thereabouts. However we can still speed things up a little. Look at lines 130 and 140. Provided A does have an integer value our problem has a solution, and we can simplify this part like this:

```
130 A=(100-39*D-22*C-17*B)/14
140 IF A=INT(A) THEN PRINT A;B;C;D
```

This improves the speed to around 2.24 seconds- about the time we saved with our first improvement. Just a little thought before coding a problem can save much un-needed work by the computer and speed things up to a degree which is greater than the difference between a 4Mhz computer and a 32MHz computer -all other things being equal, which they never are!

Thanks to Recreational & Educational Computing Newsletter Vol 1 #4. Sample copies $15, from 909 Violet Terrace, Clarks Summit, PA, USA, 18411

TENTH ANNIVERSARY...

Easter 1981 I keyed in my first program on a NTSC TI99/4 borrowed from Texas Instruments (those were the days. Try cadging an Amiga from ukw!) and subsequently received an invoice from TI dated June 1981 for a fully expanded TI99/4 side-car system -the UK PAL version was then available, but with sound from a little loudspeaker in the console as TI had not then worked out how to rejig the modulator, made for European PAL, to work properly on UK PAL. They never did get it ENTIRELY right! despite many changes of modulator design.

I became a founder member of the UK User Group, and by January 1982 had a program published in Computer and Video Games Magazine (still going but no listings these days). I was also a founder subscriber to 99er Magazine, and was soon making my first purchase of modules from the USA, they were so much cheaper there than TI was selling them for here!

As the year progressed I wished to share the software I was buying, and did the proper thing by establishing licensing arrangements and sold programs as Stainless Software, and so we went on... upgrading to a PEB and 99/4A in due course, continuing to write for anyone who wanted my material.

In due course, the collected writings were assembled into a book- I was asked to write it by a commercial publisher, who made the first payment before a word was written. WRITTEN ENTIRELY WITH TI WRITER, it brought in a little money, all of which was spent on new programs and books!

I love games, especially strategic games. I enjoy programming. And the TI has so many superb programs, so many languages to explore (still discovering new things about Basic!). I have a STACK of computer books full of programming ideas, and hundreds of programs to write, if I can ever find the time (anyone need programming ideas, mainly strategy or puzzle games?).

My TI has at this moment clocked up 11,044 hours, and has outlived the TV I bought to use with it! I am quite happy to go on using it for another ten years, even longer maybe, only an irrepairable breakdown is going to separate us!

Now, a maths program which can take two numbers as input and will output the greatest common divisor and the least common multiple.

For example, if we factor two numbers (list all numbers which multiplied together result in that number):

24 = 1 x 2 x 2 x 2 x 3
78 = 1 x 2 x 3 x 13

Of the divisors listed the largest number common to both is 3 and the least common multiple (the lowest number which divisible by both numbers) is 312 (that is 24 x 13 and also 78 x 4).

```
100 DISPLAY AT(1,1)ERASE ALL:"PROGRAM TO FIND":"":"THE
    GREATEST COMMON DIVISOR":"":"AND THE":"LEAST COMMON
    MULTIPLE"
110 DISPLAY AT(8,1):"Spencer Earnshaw/M W Ecker/ S Shaw
    1991"
120 DISPLAY AT(14,1):"FIRST NUMBER:":"":"SECOND NUMBER:"
130 ACCEPT AT(14,15)VALIDATE(DIGIT):A
140 ACCEPT AT(16,15)VALIDATE(DIGIT):B
150 AA=A :: GCD=B
160 R=AA-INT(AA/(GCD+ABS(GCD=0)))*GCD
170 IF R<2 THEN GCD=R+GCD*(1-R):: GCD=GCD*ABS(GCD>0)+ABS
    (GCD=0):: GOTO 190
180 AA=GCD :: GCD=R :: GOTO 160
190 DISPLAY AT(1,1)ERASE ALL:"THE TWO NUMBERS:":"  ";A;"
    ";B
200 DISPLAY AT(4,1):"GREATEST COMMON DIVISOR:    ":TAB(
    13);GCD
210 DISPLAY AT(6,1):"LEAST COMMON MULTIPLE IS:  ":TAB(1
    3);A*B/GCD
220 DISPLAY AT(12,1):"ANY KEY FOR ANOTHER"
230 DISPLAY AT(16,12):"< key > " :: CALL KEY(5,K,S)::
    DISPLAY AT(16,1):" " :: IF S<>1 THEN 230
240 RUN
250 END
301 ! from recreational and educational computing
302 ! $13 sample copies $36 subscription (Europe)
303 ! 909 Violet Terrace Clarks Summit PA USA 18411
```

CHRISTMAS PUZZLE and I do have a program that solves this one and will (probably) print it next issue if nobody send me a better solution. The only hint I will give is that the computer will point you in the right direction. (Anyone do cryptic crosswords?).

Puzzle- Santa Claus (or if you prefer, Father Christmas) was delivering presents on Christmas Eve, along a loooong street. When he got to house number 125 he noticed that dropping the first digit left 25, a number which is of course divisible by 25. When he got to the 25th house which satisfied this condition, he discovered it was his own. What was the number? All 25 numbers satisfy the one condition: When you remove the first digit, all the remaining digits form a number divisible by 25. Leading zeroes as in 00025 are of course ignored and 00025 is treated as 25 which is divisible by 25.

TEXT READ BY COMPUTER!

The disk library has recently received the text of the New Testament on disk. I thought it would be nice to have the computer read it to me... hence the following program. It has been put together specifically to deal with the Bible text, and only takes account of the letters A to Z, in upper OR lower case.

It is written to be used with the Terminal Emulator 2 (TE2) module and of course requires the speech synthesiser peripheral.

Using TE2 we program in Basic, and cannot use the XB form of LINPUT A$ to read the text from disk. As some punctuation commonly found in text represents "end of record" when using Display format files, we have added a comma to the INPUT line, which ensures that we miss no text. In XB LINPUT ignores the commas and so on (=LINE INPUT).

Apart from the need for the comma after A$ there is nothing too strange about this little program. As TE2 only "speaks" upper case we have to go through the string to alter it all to upper case, and for the purpose of the intended usage, certain characters are excluded from the string to be spoken. The chapter and verse numbers used in the Bible are very out of place if you insert them as you read!

Hope you find this of use!

```
100 REM READ DV80 FILE
110 REM  WITH EMBEDDED COMMAS
120 REM AND LOWER CASE
130 REM AND USE TE2 TO READ
140 REM IT OUT LOUD.
150 REM THIS PROGRAM ONLY
160 REM SPEAKS A-Z and a-z
170 REM NUMBERS ARE REMOVED
180 REM RECOMMEND THAT TEXT IS PLACED ON A RAM DISK IF
        AT ALL POSSIBLE FOR SMALLER PAUSES
190 REM
200 REM S SHAW FEB 91
210 REM
220 OPEN #1:"DSK3.LUKE1",INPUT
230 OPEN #2:"SPEECH",OUTPUT
240 INPUT #1:A$,
250 IF LEN(A$)<1 THEN 240
260 PRINT A$
270 A$="__"&A$
280 FOR T=1 TO LEN(A$)
290 A=ASC(SEG$(A$,T,1))
300 IF A>57 THEN 320
310 IF A<>32 THEN 360
320 IF A<95 THEN 350
```

continued on page 26

Sorting, part 2

by Ron Brubaker, USA

The following is a modified version of the program segment given in last month's article. Two variables C and S have been added to permit measurement of the efficiency of the routine. These are initialised in lines 155 and 156. The counter in line 175 counts the number of comparisons that are made and the counter in line 185 counts the number of swaps that have been performed. (EDITOR'S NOTE - all of the programs in this series have been Listed to disk from working programs so there should be no errors.)

```
10 REM **** GENERATION OF A LIST OF RANDOM NUMBERS ****
20 REM
30 DIM A(100)
40 RANDOMIZE
50 PRINT "HOW MANY NUMBERS DO YOU WANT";
60 INPUT N
70 PRINT
80 FOR I=1 TO N
90 A(I)=INT(RND*100)+1
100 PRINT A(I);
110 NEXT I
120 PRINT
130 REM
140 REM *** SIMPLE BRUTE FORCE SORT - WITH COUNTERS ***
150 REM
155 C=0
156 S=0
160 FOR J=1 TO N-1
170 FOR I=1 TO N-1
175 C=C+1
180 IF A(I)<=A(I+1)THEN 220
185 S=S+1
190 T=A(I)
200 A(I)=A(I+1)
210 A(I+1)=T
220 NEXT I
230 NEXT J
240 REM
250 REM **** ROUTINE TO PRINT SORTED LIST OF NUMBERS ****
260 REM
270 PRINT
280 FOR I=1 TO N
290 PRINT A(I);
300 NEXT I
310 PRINT
315 PRINT
320 PRINT C;"COMPARISON'S WERE MADE"
330 PRINT
335 PRINT S;"SWAPS WERE MADE"
```

The resulting output is the following:

```
HOW MANY NUMBERS DO YOU WANT? 15

64 48 2 79 36 5 66 71 100 24 14 67 57 1 3 34
```

And finally,

```
2 5 13 14 24 34 36 48 57 64 66 67 71 79  100

196 COMPARISONS WERE MADE

56 SWAPS WERE MADE
```

An additional routine has been added that will print the sorted list and the results accumulated by the counters.

Note that the number of comparisons is what would be predicted from the loop structure of the program (i.e. 196 is N-1 squared).

From the loops shown above it would seem unnecessary to extend the inner loop to N-1 on every pass since the largest number falls immediately to the end of the first pass and on each succeeding pass an additional number falls into position near the end of the list. Thus, the inner loop can terminate after one less step on each pass. This can be accomplished very easily by changing line 170 to terminate the inner loop at N-J instead of N-1. This change and the resulting output is shown below.

```
10 REM **** GENERATION OF A LIST OF RANDOM NUMBERS ****
20 REM
30 DIM A(100)
40 RANDOMIZE
50 PRINT "HOW MANY NUMBERS DO YOU WANT";
60 INPUT N
70 PRINT
80 FOR I=1 TO N
90 A(I)=INT(RND*100)+1
100 PRINT A(I);
110 NEXT I
120 PRINT
130 REM
140 REM ***** SIMPLE BUBBLE SORT *****
150 REM
155 C=0
156 S=0
160 FOR J=1 TO N-1
170 FOR I=1 TO N-J
175 C=C+1
180 IF A(I)<=A(I+1)THEN 220
185 S=S+1
190 T=A(I)
200 A(I)=A(I+1)
210 A(I+1)=T
220 NEXT I
230 NEXT J
240 REM
250 REM ***** ROUTINE TO PRINT LIST OF NUMBERS *****
260 REM
270 PRINT
280 FOR I=1 TO N
290 PRINT A(I);
300 NEXT I
310 PRINT
315 PRINT
320 PRINT C;"COMPARISONS WERE MADE"
330 PRINT
335 PRINT S;"SWAPS WERE MADE"
```

In this case the output is:

```
HOW MANY NUMBERS DO YOU WANT? 15

64 48 2 79 36 5 66 71 100 24 14 67 57 1 3 34

2 13 14 24 34 36 48 57 64 66 67 71 79 1 00

105 COMPARISONS WERE MADE

56 SWAPS WERE MADE
```

Note that the number of comparisons has decreased to 105 but that the number of swaps is still 56 and the output is still sorted. It is reasonable to assume that the revised program runs nearly twice as fast as the original.

It was also noted above that the sort was completed after less than N-1 passes. However, it cannot be assumed that all combinations of fifteen random numbers will be sorted in the same number of passes that the test data shown in these examples required. The safest way to determine if the sort has been completed is to check to see if any swaps have been performed on the last pass. The following version of the sort routine has been modified by replacing the outer FOR/NEXT loop with a looping structure that loops until no further swaps have been made on the last pass through the inner loop.

```
10 REM **** GENERATION OF A LIST OF RANDOM NUMBERS ****
20 REM
30 DIM A(100)
40 RANDOMIZE
50 PRINT "HOW MANY NUMBERS DO YOU WANT";
60 INPUT N
70 PRINT
80 FOR I=1 TO N
```

continued on page 35

TI-Bits Number 12

by Jim Swedlow, CA USA

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

99/FEST WEST/87

I went to LA 99'ers FEST WEST on Saturday May 16th at the Shrine Exposition Hall. For those of you who could not go, here are some impressions.

The Hall has a large area on the main floor and two mezzanines. One mezzanine was for Amiga users and the other for the TI-99/4A. The first floor was crammed with sellers of clones and computer supplies.

The booths at the 99/FEST has user groups (LA 99ers, Pomona Valley, BUG, San Fernando Valley) and vendors. Craig Miller was there with the TRITON XT TURBO working from a 99 keyboard. MYARC was there with the GENEVE. RAVE was there with their new keyboard. TAPE, Bytemaster, DIJIT and many others were there. It was quite an event.

There is no way to tell you everything so I will share some verbal snapshots.

1. The most popular exhibit was the GENEVE. There was always a crowd there. They ran TI cartridges that had been dumped to disk. Heard it said that they will soon have a chip that will allow it to work with the TI disk controller.

2. The TRITON XT CLONE really does work off the TI keyboard. Craig had Flight Simulator on a composite colour monitor and it looked pretty good. Text on a colour composite monitor, however, is poor at best.

3. Richard Mitchell said that the next (December, 1986) issue of The Smart Programmer is in the mail and January should be out in six weeks or so. Hope so, its a good publication.

4. STAR RUNNER from DataBiotics is really our old friend TI RUNNER with some graphics changes. That game, in my humble opinion, is one of the best games ever written for the TI.

5. Ken Hamai went down to the main show and found two WICO track balls for the TI and a $5 full height TI drive. Only Ken could find that stuff. He also had a double widget -- this you have to see to believe.

6. The folks from RAVE said that they had sold more of their keyboards to California owners than to any other state.

7. Did a little survey among some of the big names in TI land and the general consensus is that our orphan has at least two more good years left.

8. The conference was on line through GENIE with the Ontario TI FEST. It had been hoped that England would join on line but that did not work out. Was interesting to watch all the folks talking at once.

9. DIGIT had their RGB monitor interface working. They were running the LINES program that comes with MINI MEMORY. Wow, are those colours sharp.

10. There were lots of good deals downstairs. I found a disk box that hold 100 disks for $5. I have seen that same box sell for $9 to $19 in stores. You have to shop around, however. That disk box sold for $6.50, two for $12, $6, $5.50 and $5.

11. Up and down the TI area you could see folks working on their computers and programs. It was really nice to see all those keyboards and systems up and working in one place.

12. LA 99ers said that they will be selling an interface card to use an IBM type keyboard with the 4A for about $80. They had one there in a 4A but it was not on a live system. Look for an announcement in a month or two.

13. RAVE and MG talked about making the RAVE keyboard also work with the TRITON TURBO XT. That would be nice.

All in all, it was great.

EXPANDING YOUR SYSTEM

I have had a couple of conversations recently about system expansion. There are so many products available today — the TRITON TURBO XT, the Rave keyboard, RAM disks, the GENEVE — that it is difficult to make decisions.

It is not my intent to tell you what to do or buy, rather I hope to give you some food for thought.

The first step in any expansion plan is to do a needs assessment. Does your current system meet your needs? Where does it fall short? TI Writer and Multiplan are quite powerful for some applications and inadequate for others. The 48 key keyboard that TI gave us has limitations but is workable. Perhaps you just want to be able to walk into a store again and buy some NEW software.

The next step is much harder. How much can you invest? As my boss once said, this is where the rubber meets the road. Your system may fall very short of your needs and wants, but if the dollars are not there, they are not there.

Just for discussion purposes, lets assume that you have a P box with RS232, 32K and one or more drives and that you can afford to spend $700 to $1000 on upgrading your system. How should you spend the money? Here are two different routes.

GO BIG BLUE: One way is to buy one of the many TURBO XT clones that are available these days. If you shop around, you can find a complete monochrome graphics system with DOS for $700 to $800 and a colour system for under $1000. This will not leave you much for software but there is lots and lots of good fairware out there.

A TI SUPER SYSTEM: You could invest you money in upgrading your current TI system:

Item	Low	High
Print Spooler	$100	$100
RAM Disk	150	200
RAVE 99 Keyboard	185	200
Upgraded Disk Drives	100	300
Composite Colour Monitor	150	200
Total	$685	$1000

PRINT SPOOLER: There are many options, including the CORCOMP Triple Tech Card. You will find a print spooler invaluable. Luckily, TI opted for a standard Centronics parallel interface. You plug your printer cable into the spooler and the spooler's cable into your printer. Then all those long delays during printing are gone and you have real multi-tasking! When you print something with the Formatter, for example, about a quarter of the way through, your TI is done and ready for other things while your printer just clacks away. You will love it.

RAM DISK: If you have not seen one of these demonstrated, it is an electronic disk. To your system it looks just like a disk drive but many times faster. For example, you press the key for the Formatter and in one or two SECONDS you have the Formatter. You may have passed the point where computers seem like magic but a RAM disk will.

KEYBOARD: My two major complaints about my TI are the lack of an 80 column screen and the keyboard. Although we have yet to see a fully compatible 80 column card, RAVE 99 does sell a full size keyboard for about $200 and a device for attaching an IBM style keyboard for about $150 (comes out to around $200).

Look for other devices in the near future.

DISK DRIVES: If you have single sided drives, you can increase your storage capacity. One way is to upgrade to double sided drives. All three Disk Controller Cards (TI, MYRAC and CORCOMP) support double sided drives. Twice the space is a joy. If you have trouble with installing them, you may need help from an experienced user group member.

Another approach is to replace your TI Disk Controller card with one from MYRAC or CORCOMP. This will turn your drives into double density drives (ALL of the drives TI sold were single sided double density drives - the TI disk controller card did not support double density). You may have limited compatibility problems as a few programs that access disk drives do not work with non-TI cards.

Jumping from SSSD to DSDD quadruples your disk storage space!

COLOUR MONITOR: If you are still using a TV, you will be amazed at the improvement that you will get from a monitor. Crisper colours. Clarity from edge to edge and minimal distortion.

TWO WORDS ON PRICE: Shop around. As I was preparing for this article, I found quite a range in prices for the same item and items for the task at hand. Prices quoted here are middle of the range.

So, folks, there you have it - two ways to spend your $700 - $1000. Now, if only I could get my hands on the bucks!

MILLER GRAPHICS' NIGHT MISSION

I should have bought this long ago, but the money and the opportunity never came together.

As a game, it is a good one. The speed is amazing when you consider that it is 100% Extended Basic. This alone justifies the $20 price tag.

The real value, in my opinion, is the documentation. Sure, it has the game instructions, but it also includes three other sections.

First, it explains the program, line by line. You WILL learn new Extended BASIC programming techniques by reading this.

Second, it has a list of CALL LOAD's and CALL PEEK's, some of which are new.

Third, it has a very interesting section on 'The Power of AND'. This is a completely new programming device.

As a game, as a tutorial or as both, this is a wise purchase.

XB PROGRAMMING TIP

I was working on a program and I ran into something interesting.

I had a variable (A) that could only be either 9 or 7. I needed to separate the beginning of a string. If A was 9, I needed the first 5 characters in the string, otherwise I needed the first 4 characters.

My first approach was:

```
IF A=9 THEN N$=SEG$(A$,1,5
ELSE N$=SEG$(A$,1,4)
```

That seemed bulky so I tried this:

```
N$=SEG$(A$,1,5+(A=7))
```

[This works because if A is 7, the expression (A=7) returns -1. Otherwise it returns 0. If this is not clear, try this loop:

```
FOR A=6 TO 8 ::
PRINT A;A=7;5+(A=7) ::
NEXT A
```

Note that the parenthesis are only needed for 5+(A=7). Without them, your TI would compare 5+A to 7.]

Back to my tip. That was better, but I did a bit of fiddling and realized that 0.6 times 9 is 5.4 and 0.6 times 7 is 4.2. I tried this:

```
N$=SEG$(A$,1,INT(A*.6))
```

Then I wondered if the INT was needed. It turns out that SEG$ (like many TI functions) automatically rounds off.

How can you tell if a function rounds? Try using it in a loop:

```
FOR I=1 TO 2 STEP .1
PRINT I;SEG$("ABC",1,I)
NEXT I
```

My final expressions was:

```
N$=SEG$(A$,1,A*.6)
```

Enjoy.

continued from page 11

BANRBIAS has also interesting implications but it deals with the FN## and FT## image fonts and as mentioned earlier, I hope someone can shed some light on that problem.

The Dark option is very useful when using partially spent printer ribbons. Up to four levels of overstrike options are selectable.

Overall Comments on TIPS

As mentioned at the very outset of this tutorial, I am a keen devotee of Page Pro 99. With the many native fonts inherent to that format, coupled with the portability of high resolution digitized images, capable of being located anywhere on an A4 size printout, TIPS must be seen to have some limitations.

However these limitations of large graphics and preset areas of placement on greeting cards, are the very same limitations that exist with PRINTSHOP for PCs.

It must have been a truly monumental task for Ron Wolcott, to literally have filled the proverbial thimble of our vintage computer, with all the contents of the bucket of the modern PC, without spilling too many drops. Not to mention the countless hours he must have spent collecting all his images from bulletin boards, converting them to TI99/4A format and then categorizing them. I can only imagine that he too enjoys a very tolerant domestic situation.

Such an effort as TIPS for whatever purpose it is used, should be rewarded not only by the purchase price of the freeware but by a real donation to the author. I have discussed these sentiments with our co-ordinator Dick Warburton and he has assured me that he will recommend to our board that Ron Wolcott be honoured by making him our Author of the Month in the very near future.

I hope that through this tutorial more members of our TI99/4A community will be motivated to buy and use TIPS.

continued on page 28

A Few Tips on TIPS

by Ed Machonis, USA

If you have never heard of TIPS, or heard very little, pull up a chair. TIPS stands for TI Print Shop. It is a program written by a gentleman named Ron Wolcott that will print Banners, Signs, Greeting Cards and Address Labels. This in itself is quite a feat, but there is more. It enables the easy incorporation of graphics into each of the above. But there is still more! Accompanying the program are a series of files chock full of pictures you can use in your Banners, Signs, etc.

Now hold on to your seat. The set I obtained contained 3,223 pictures or graphics. One thing you soon learn with TIPS and that is everything is done on a grand scale. The pictures have been converted from public domain graphics available for IBM computers. The amount of work representing this assembly and conversion staggers the imagination. Still holding onto that seat? Ron has placed his program and all his files in the public domain. Yes Virginia, there is a Santa Claus and for us TI99/4Aers his name is Ron Wolcott.

The set has been placed on all the National networks such as Compuserve, Delphi and Genie and many local BBS's. This is evidently the method of distribution, I do not even have Ron's address to say thanks!

While at TICOFF I was pleased to find that the Central Westchester TI User Group was making TIPS available for just about the cost of the disks and packaging. The files were archived onto 7 Double Sided Disks (Cost $7) or 8 flippies ($8). Since Art Byers, the president of the group, is Sysop of the TI99/4A section on Delphi, the programs were, so to speak, straight from the horse's mouth. I lost no time in plunking down your group's money for a double sided set. (A noted TI99/4A personality at TICOFF told me it cost them $89 in network charges to download a set of files).

In response to my query, Art told me that the CW Group will make the double sided set available to anyone writing in for the Show price ($7.00) plus $2.00 shipping. The address is:

The Central Westchester 99ers Club,
PO Box 812,
SHRUB OAK, NY 10599

While your at it, do yourself a favour. Order a catalogue of the images in the TIPS library so that you will know what picture is on what file. The 110 page catalogue, punched for a 3 hole binder, is available for $9.00 plus $2.00 shipping.

Why spend 11 bucks when Ron has included a program that will print out the catalogue for you? According to the literature I picked up at TICOFF it will take you 25 hours of printing time. Add to that the cost of the paper and a new ribbon, as well as the wear and tear on your printer and disk drive, and you have found yourself another bargain!

I believe the 25 hour figure is based on an older version of TIPS. The documentation for versions 1.3 and 1.4 state that an assembly language file named TIPSOBJ2 has been added to speed up the printing of graphics. My own measurements indicate that the catalogue can now be printed in about 11 hours, about 50% of which is Disk Drive operation and the rest printer. Whether 25 hours or 11 hours, the time can be better spent using and enjoying the program rather than baby-sitting a printer. (You must enter a new file name every 25 minutes or so). The QB99ers have purchased a catalogue which the members may borrow for copying at their own expense. If you cannot have it copied for less than 10 cents a page, send your order to Central Westchester.

If you do decide to print your own catalogue, here is another tip, offset your paper to the left by about 3 or 4 characters to leave a wider left margin for 3 hole punching. If you do not care to devote either the time or money for an image catalogue, or just cannot wait, I have written a TINY GRAM (what else?) that will print out the descriptive names of all images in less than half an hour. While TIPS will do the same job, it prints the names in a single column, using two pages for a single graphic file. The TINY GRAM will print the names in 5 columns and uses only half a page for a file of names. You will find the listing at the end of this article. A copy of the program will be placed on the TIPS program disk in our library.

As noted above, the files on the set of seven DSSD disks are archived. (Really, 6 DSSD disks and 1 SSSD). They de-archive to 17 double sided disks or 33 single sided disks. (I told you everything about this program is on a grand scale!) The CW group has thoughtfully provided a copy of Archiver version 3.03 on one of this disks to help you extract the files. No documentation, but the program is completely menu driven and you do not need any documentation. This program is Fairware and if you find it useful send Barry Boone some bucks. (If the files were not archived, the CW group would have to charge $17 for the DSSD set plus $4.50 shipping. A SSSD set would run $33 plus $8.50 shipping. Disks are $1 a piece and shipping charges are 50 cents for the first disk and 25 cents for each additional disk).

The 39 archived files unpack to 98 files of some 10,615 sectors (like I said, a GRAND scale). There are only 12,206 sectors available on the 17 DSSD disks. Clearly, some thought must be given to allocating these files to their respective disks. I searched the documentation for any hint to no avail. After some false starts, I came up with an arrangement which makes it easy to locate the individual files and uses a minimum number of disks. (I was able to put all the files on 16 DSSD disks but some of the files were not in a logical order).

A brief description of the files will help you understand the file allocation. The main body of the files consist of graphic images in Internal Fixed format. There are generally 126 images in a file and its name ends with the letter TXT. It is 348 sectors long. The names of the images are contained in a separate file whose name ends with XXX and it is 9 sectors in length. The two files take up 357 sectors and just fit on a single sided disk. Two such collections will fit on a DSSD disk.

The main body of files begin with the letters GR, followed by two letters indicating the alphabetical range of the initial letters of the image names and ending either with TXT or XXX. File GRABTXT will contain images whose initial letters begin with either A or B. The names of the images in this file are contained in file GRABXXX. Since there is no GRAA file, all images whose names begin with A will be in the GRAB file. The image named BIRD could be either in GRAB or GRBC. COOKIE MONSTER might be in GRBC, GRCC or GRCD. Since all names are in alphabetical order, it would probably be in one of the later files and actually is found in GRCD along with COOKIE MONSTER1, COOKIE MONSTER 2 and COOKIES.

The TIPS program searches for file names beginning with the letters GR in the drive you designate where images can be found. The range for these files runs from GRAB to GRWZ. Since, except for GRWZ, each file fills a SSSD, there is no room for adding additional images. The files I received included GAAZ (Animals A to Z); GMAZ and GWAZ (Men and Women A to Z, respectively. No mixing of the sexes here!); GPAZ (People, male and female by activity); GTAZ (Transportation); and PGAZ, the Peanuts Gang from A to Z. These files vary in length depending on the number of images included and as with the GR files, the file names end with either the TXT or XXX suffix.

Also included were files H1AZ and H2AZ, containing Holiday images, mostly Christmas. The package also contained files C1AZ through C6AZ which were collections of Frames, Signs, Faces and Symbols. All of these files contained 100 images and were 276 sectors (TXT suffix) and 8 sectors (XXX suffix) in length.

Three files with Fonts, FN11, FN22 and FN33 are included bearing the usual TXT or XXX suffix. These fonts are apparently only used for Banners. The TIPS program also uses four other resident font styles for Signs and Cards. These other font files must be on the same disk as the TIPS program.

The four resident fonts do not have any lower case letters and only a limited number of special characters.

When you place a disk in the drive designated for images, TIPS will give you the range of image names for each GR file on that disk. If TIPS does not find a GR prefix filename, you will be given the chance to enter a two letter filename prefix. Inputting the two letter prefix will generate a range of image names on that file. The program will (l)ist out to your printer the image names in any file upon request.

A nice user friendly feature of TIPS is that all features are selected by just entering the initial letter of the function. The letter can be entered in either upper or lower case.

Before you can extract the archived files you must format your disks. Archiver does not format disks and this is the only improvement I could think of for this fine program. While you are formatting the disks, name them in accordance with the disk catalogues shown at the end of this article. You will need either 17 or 33 disks depending whether your drive is single or double sided. You can de-archive using just one drive and swapping disks as prompted, but it is a lot easier with two drives. If you have SSSD drives, you might give some though to placing the files on 17 flippies. The saving in disk storage space is not to be sneezed at.

Keep track of the name for each disk as you will place the files on their respective disks in accordance with the catalogues appended to this article.

File names of most of the archived files are descriptive of their contents. See the Catalogue of Archived Tips Disks at the end of this article. Note that Disk TIPS/ARC01 contains files TIPDOC/1*4, TIPS/1*4 and TIPS/UTILS.

The first file to de-archive is TIPS/UTILS. Cataloguing this ARC file would produce the following listing:

```
ArcFile:  TIPS/UTILS

CTIPSVUA        33   Program        8192   P
CTIPSVUB        21   Program        5054   P
CTIPSVUDOC       7   Dis/Var          80
CTIPS_IA        33   Program        8192   P
CTIPS_IB        31   Program        7590   P
CTIPS_IDOC      13   Dis/Var          80
GDAZTXT          4   Int/Fix          53
GDAZXXX          2   Int/Fix          16
LOADX           30   Program        7284
TIPSSHOWX        7   Program        1478
TIPSX           95   Int/Var         254

Total Files     11   Total Size: 266
```

All files should be copied onto the disk named TIPS-PROGM except CTIPSVUDOC and CTIPS-IDOC. In response to the query "Extract all files?" answer N and accept all files except the above two. The program LOADX is a special loader written by Irwin Hott and loads the files TIPSX (the main program) or TIPSSHOWX (prints the image catalogue). When you are through, change the file name LOADX on your TIPS-PROGM disk to just LOAD for auto-load from extended basic. Do not remove the suffix X from the other two files.

The next file to unpack is TIPS/1*4. Cataloguing it produces this listing:

```
ArcFile:  TIPS/1*4

TIPS           84   Int/Var        254
TIPS2RD         3   Program        371
TIPSDV80        3   Dis/Var         80
TIPSFONT0       2   Int/Var         80
TIPSFONT1       8   Int/Var         80
TIPSFONT2       8   Int/Var         80
TIPSFONT3       8   Int/Var         80
TIPSFONT4       8   Int/Var         80
TIPSOBJ        21   Dis/Fix         80
TIPSOBJ2       27   Dis/Fix         80
TIPSSHOW        7   Program       1465
TIPSTAB1        7   Int/Fix        129
TIPSTAB2        3   Int/Fix        129

Total files:   13   Total size:    189
```

If you are using DS disks, copy all of these files onto your TIPS-PROGM disk in the interests of keeping all files together. However, with SS disks, there will not be enough room for TIPS or TIPSSHOW which duplicate TIPSX and TIPSSHOWX. Again respond No to "Extract all files?" and then accept all but TIPS and TIPSSHOW. You are now through with the TIPS-PROGM disk.

Next extract all files from ARC file TIPDOC1*4 onto your TIPS-DOCS disk. Go back to ARC file TIPS/UTILS and extract the two DOC files onto your TIPS-DOCS disk. You now are through with TIPS-DOCS which is really a SSSD disk.

Except for Archiver and its LOADer, extract the remaining files on disk TIPS/ARC01 onto their respective disks per the appended catalogues. All the FN files will fit on one DS disk but each will require a separate SS disk.

The rest is a piece of cake. When you are all done, print out disk labels using the latest version of CATALABEL (which prints the disk name in Expanded Compressed Bold type). It is the fastest way to label a large quantity of disks.

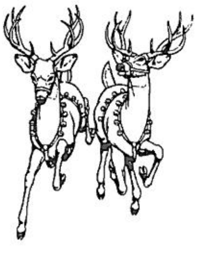
The documentation for the program are laconic at best. The best way to learn is to keep trying every function and see what happens. All you are going to ruin is paper. One more tip: when the program asks you to name a drive for the image disk, be sure to have a disk in that drive before naming it. The program immediately looks for a disk in that drive and crashes if it does not find one. One of my first tasks will be to install an error trap at this point although Ron does warn that the program is full with little room for changes.

The Label function can only use resident font No. 1 and this is what you will get no matter what you ask for. I am amazed at the range of sizes in which each graphic can be printed. About 3/4" high on labels, 3" high on Cards and about 6" high on Signs and Banners. When converted to TI Artist instances, size 1ART will fill about 25% of the screen and 2ART will fill about 75% of the screen, leaving little room for adding anything else.

Note: this article also applies to version 1.7, which is currently available at the shop. o

continued from page 21

```
330 A=A-32
340 IF A=61 THEN 360
350 B$=B$&CHR$(A)
360 NEXT T
370 A$=B$
380 B$=""
390 PRINT #2:A$
400 IF EOF(1)THEN 420
410 GOTO 240
420 CLOSE #1
430 CLOSE #2
440 END
```



o

XB tips Number 13

by Jim Swedlow, CA USA

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

ON PROGRAM FILES

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 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 XB programs, Assembly language files saved in program format and data files produced by some modules (Personal Record Keeping, Adventure, etc).

Your computer cannot distinguish between the various types of program files. They all look the same. 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.UTIL1.

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 would 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? Best I can figure is that 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.

Upon return from the subroutine, your 4A looks for the next instruction. It does not consider the ELSE. Hence, PRINT "TEST" is ignored while the second GOSUB 40 is executed as it follows a line separator.

Either of these line 20's work:

```
20 IF I=2 THEN GOSUB 40 ELSE GOSUB 40

20 IF I=2 THEN GOSUB 40 :: PRINT "TEST" ELSE PRINT
"TEST" :: GOSUB 40
```

The first one works because there is no line separator after the ELSE so control passes to the next line, as it should.

The second one works because there is a line separator after the GOSUB so the ELSE is functional.

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. Note this coding in line 170 of this month's program. It makes a better listing when using NEATLIST (Source: a TIGERCLUB program).

DISK*LABEL PATCH

My program in the September ROM works with a GEMINI printer but has some problems with an EPSON.

To fix this, change the beginning of line 240 to this:

```
PRINT #2:CHR$(15);F$(0) . . .
```

MAIL*LABEL

This month's program prints any of three types of labels for mailing disks: warnings for the Postal Service, your name and address and any five lines you want.

If your printer is not PIO, use the correct name in line 150. Include your name and address in lines 290-310. Be sure and include the leading comma in line 290 and the trailing comma in line 310.

```
100 ! MAIL*LABEL
110 ! VERSION XB.1.0
120 ! 04 Oct 85
130 ! By Jim Swedlow
140 !
150 DIM E$(5):: OPEN #1:"PIO " :: PRINT #1:CHR$(18);CHR$
(15);
160 DISPLAY AT(8,10)ERASE ALL BEEP:"MAIL*LABEL": :"Press
For": :" 1 Address Label ":" 2 Custom Label":" 3
Warning Label":"  4   End"
170 CALL KEY(3,K,S):: IF S<1 OR K<49 OR K>52 THEN 170 ::
IF K=50 THEN 190 :: IF K=52 THEN 240 ELSE IF K=49 THEN
RESTORE 290 ELSE RESTORE
180 FOR I=1 TO 5 :: READ E$(I):: NEXT I :: GOTO 200
190 DISPLAY AT(10,1):"Input Label lines": : : : : : : ::
FOR I=1 TO 5 :: ACCEPT AT(I+11,1)BEEP:E$(I):: NEXT I
200 DISPLAY AT(18,1):"How many labels? 9" :: ACCEPT AT(
18,19)SIZE(-2)VALIDATE(DIGIT)BEEP:N$ :: IF N$="" THEN 20
0 :: S=VAL(N$)
210 DISPLAY AT(10,1): : : :" Printing": : : : : :
220 FOR I=1 TO S :: FOR K=1TO 5 :: PRINT #1:CHR$(14);E$
(K):: NEXT K :: PRINT #1: : :: NEXT I
230 FOR I=1 TO 5 :: E$(I)="" :: NEXT I :: GOTO 160
240 PRINT #1:CHR$(18);:: CLOSE #1 :: CALL CLEAR :: STOP
250 DATA "        FIRST CLASS"
260 DATA "        MAGNETIC MEDIA"
270 DATA DO NOT BEND * DO NOT X-RAY
280 DATA DO NOT EXPOSE TO MAGNETISM
290 DATA ,Your Name
300 DATA Street Address
310 DATA "City, CA 90ZIP",
```

Contract Bridge version 3.00

by John H. Bull, USA

EDITOR'S COMMENT: I have a copy of this game and use it often, restricting my efforts almost exclusively to Rubber Bridge. I am not an expert Bridge player but I do enjoy a good game and I have received a lot of enjoyment from this disk. Let me know what you think of it.

INTRODUCTION

This option (the file is "TUTOR") will show you how the game is played and how the mechanics of this program work. The bidding and play of all hands are entirely by you but must be within the rules - the program does not allow you to make an illegal bid or play an illegal card.

The hands are not in memory but are randomly dealt - each set of hands is different. The size of the 28 column screen limits the display of a set of hands to 8-card suits. Suits longer than 8 cards will be dealt, bid, and played - but only the first eight cards in the suit will be seen. It happens very seldom and it should not be a problem.

Once or twice through this option should be enough if you already know how to play bridge. It is not likely to improve your game. Several deals will help if you have not played before.

RUBBER BRIDGE

This option (the file named "RUBBER") is as much as possible like an ordinary social game of Bridge but instead of four players there are just you and the computer. You play the South hand and also the Dummy if you win the bidding and are Declarer. If you are the Dummy you just collect the tricks. You are competing against the cards and the computer.

The level of play is intelligent but not expert - very few of the conventions of expert bidding and play are used. The computer pretty much uses the tips on bidding and play that are given in the Introduction.

DUPLICATE

With this option ("DUPLICATE" is the file name) you play sets of hands that have been previously dealt and played. You compete, not against the computer and the cards, but against the score made by another player who has played the same cards. The fifty boards on the flip side of the disk were played by me and, as you will see, I do not qualify as an expert. The option to "[S]ave your play" will let you substitute your bidding, play, and score for that board. Also, you can play the NORTH hand, your partner's, when it is Declarer, if you choose to.

Scoring for Duplicate is different - there are no rubbers but bonuses of 300 and 500 points are given for nonvulnerable and vulnerable games. Game scores do not carry over from one board to the next. Penalties for being set four and more tricks at a doubled contract are higher - 400 points each for the fourth and higher tricks.

The deal is rotated to the left, as in Rubber Bridge, and so is the vulnerability: When South is dealer no one is vulnerable; when West is dealer W-E is vulnerable; when North is dealer N-S is vulnerable; and when East is dealer both pairs are vulnerable.

Additional boards can be dealt and saved to disk with the "Deal Boards" option of Duplicate. Each board is a separate file named "BOARDn", where "n" is the board #. The program will accept 99 boards numbered from 1 to 99. The original disk will hold 49 more boards numbered 51 to 99 and additional sets of up to 99 boards can be put on separate disks. Each disk must be named "BRIDGE".

The "Random Deal" option is quick and easy but remember that you will be replacing (and losing) any Board on the disk that has the same number. To "Display a Board" is also simple. In Duplicate play it is also, technically, cheating.

The "Arrange a Board" option is not as friendly as the others. The cards must be entered in proper order since no sorting is done. In entering the cards, do not press the keys too quickly. Allow a fraction of a second between key presses or else you will lose a card or two in a series. This option lets you stack the deck or replay hands from the newspaper columns.

PROGRAMMING

To program really expert bidding and play would be approximately the same size task as programming the TI99 to play a good game of Chess and the program would be pretty slow unless you did it in A/L. If you would like to try your hand at it feel free and good luck! I have a "work copy" of "RUBBER" with fairly full REMs and the source files for "DEAL/O3" and "CHARBR/O" that I will be happy to share with you.

The Screen Dump that is included in the utility for dealing duplicate boards was published by Jerry Stern in the June 1989 MICROPENDIUM. It is a clever piece of programming and works fine with my NX1000 printer. You may have to consult his article and change the default device name to suit your printer. It is VERY SLOW on the first board but be patient. Succeeding boards print much faster. Thank you Jerry! especially for line 29360. This Screen Dump is (C)1989 Jerry L. Stern; All Rights Reserved; Used with Permission.

I have worked on this program for about 5 years, off and on, learning as I went. The first versions were kept on cassette tape and I used Minimemory with the Line-by-Line Assembler to write the first versions of the A/L "DEAL" program. It has been fun. The K'Town 99'ers have given encouragement, advice, and sometimes crucial wisdom and information. Art Gibson, Mike Dodd, Mike Epperson, Joe Simmons, and Bill Sheridan have been especially helpful.

There are no KNOWN bugs but I will be grateful if you will let me know what you find. Meanwhile, I hope you have fun with it. JHB

continued from page 24

Addendum 7th November 1991

Early this week I received some feedback from Russell Welham on the "Coming to Grips with TIPS" presentation program. Russell commented on the length of time that the initial screen was displayed.

The initial screen, to identify the original author, Ron Wolcott, and the fact that the presentation program was not the original TIPS V1.7, but my modification, was added in the wee hours of Saturday morning 2nd November 1991, prior to the tutorial. The duration of the title screen display, was gauged on the reading time of that screen, for the first time, by one of the members of my family. For those members of our user group who were present at my tutorial, and later purchased a copy of the presentation program from our club shop; should you share Russell's comment, I would recommend the following modification to the TIPSX program.

The program line to be modified is line 4480, the modified line is as follows:

```
4480 FOR GRIPS=1 TO 2800 ::
NEXT GRIPS :: RETURN
```

The initial screen display time will now be approximately 15 seconds.

Best regards to all for the coming festive season.

Hints, Tips and Answers
BASIC and Extended BASIC
by Bill Sponchia, USA

1. The following program will load, run and then remove itself from memory and will put you in XB without going to to disk drive #1 looking for the program "LOAD". (You could place it on disk as a dummy LOAD file).

```
100 CALL INIT::CALL LOAD(-31952,255,0,255,0)
110 END
```

2. The following program should be saved in the MERGE format as a handy utility:

```
1  CALL CLEAR
2  OPEN #1:"DSKn.",INPUT,RELATIVE,INTERNAL  !!n = disk
   drive #
3  INPUT #1:A$,J,J,K
4  DISPLAY "SIZE=";K;" USED= ";J-K
5  FOR LOOP= 1 TO 127
6  INPUT #1:A$,A,J,K
7  IF LEN(A$)=0 THEN 10
8  DISPLAY A$;TAB(12);J
9  NEXT LOOP
10 CLOSE #1
11 INPUT "NAME OF PGM TO DELETE ELSE TYPE GO ":DEL$
12 IF DEL$="GO" THEN 15
13 DELETE "DSKn."&DEL$  !!n = disk drive #
14 GOTO 11
15 STOP
```

When programming this program can be used to catalog a diskette without getting out and loading in a disk manager. It is also written to allow for the deleting of files from the diskette.

To use just merge in with the program you are presently working on (of course it is assumed you started it a line 100 and did not use lines 1 to 15).

3. To turn off the QUIT key (FCTN =): CALL LOAD(-31806,16) Note: Now you must use "BYE" to quit BASIC and get back to the title screen.

4. In an IF-THEN-ELSE statement if you refer to a variable without any other type of relationship then this means "does not equal zero"

eg - IF X THEN 140 ELSE 100 - means If X does not equal 0 then go to 140 but if it equals 0 then go to 100.

5. When using XB prescan it will enable you to run programs that the computer will normally reject. Two things that I have been able to do are - (a) mixing FOR-TO statements and IF-THEN-ELSE statements under one line number; and (b) input two NEXT's but have only one FOR. Here is the program showing these:

```
100 J=0
110 !@P-
120 FOR J=1 TO 20 :: IF J/2=INT(J/2) THEN PRINT J ELSE
    NEXT J
130 IF J>19 THEN 150
140 NEXT J :: !@P+
150 END
```

6. To automatically return to the Master Title Screen (or Menu for Vermenu users) instead of "END" insert the following:

```
CALL INIT::CALL PEEK(2,A,B)::CALL LOAD(-31804,A,B)
or CALL INIT::CALL LOAD(31804,0,36)
```

7. To restrict a CALL KEY statement to taking only one input at a time, no matter how long a particular key is held down -

```
100 CALL KEY(0,K,S)::IF S<1 THEN 100
```

By restricting the status to +1 this will overcome the problem of the sometimes repeating key.

8. To get TRUE random numbers install this line into your program:

```
CALL PEEK(-31880,A,B) :: CALL INIT :: CALL LOAD
(-31808,A,B)
```

9. To erase the program from memory but not erase the screen (and not disturb any assembly routines in lower memory:

```
CALL INIT::CALL LOAD(-31952,255,231,255,231)
```

10. The manual tells you that there are 16 different character sets that you can redefine and change colours on. Actually there are 17 - Set #0 is never mentioned.

11. When LISTing a program and you see a line reference to "32767" this means (unless you actually used that line #) that you resequenced the program while you had a GOTO (or GOSUB, etc) to a non-existing line.

12. To LIST a portion of a program to the printer then enter the following command: LIST "filename":line number range

eg: LIST "PIO":130-240 - will LIST to PIO lines 130 to 240 (inclusive)

13. Did you know that you could identify your GOSUB routines within the program without using the "!" or REM statement. You are allowed to put one word (string) after the GOSUB line number.

Here is an example program:

```
10 CALL CLEAR::PRINT "HERE I GO.."
20 GOSUB 50 DELAY_ROUTINE ::PRINT " I am BACK!"
30 END
50 FOR T=1 TO 400::NEXT T::RETURN
```

14. Here are some interesting redefinitions for characters. To use them the proper format is "CALL CHAR(##,string) where "##" stands for the number of Character to be redefined and "string" is one of the following (or any other that you may have).

```
000804027F020408 = right arrow
00102040FE402010 = left arrow
081C2A4908080800 = up arrow
0008080B492AC108 = down arrow
00FF             = solid line
0000FE2828282828 = pi symbol
00083C4848483C08 = cent sign
0002020404482810 = check mark
```

15. The IMAGE statement (eg - 100 IMAGE ###.##) can be used with the DISPLAY AT statement using the following format -

```
DISPLAY AT(5,12):USING 100:A
```

16. Instead of using the IMAGE statement you can define a variable in the image you would like the output to look and then say "USING variable name".

```
eg - 100 F$="###.##"
     110 DISPLAY AT(12,1):USING F$:A
```

Of course, unlike the IMAGE statement which can be anywhere in the program, the variable would have to be defined BEFORE using it in a DISPLAY AT or PRINT statement.

17. When using the DISPLAY AT statement you can use TAB to properly locate where further information is to be displayed.

eg: To set up the following display

```
      MAIN MENU:

      1 - Edit
      2 - Add
      3 - Exit
```

you can set up each line with an individual DISPLAY AT statement or you can do the following:

```
DISPLAY AT(5,5):"MAIN MENU";TAB(7);"1 - Edit"
;TAB(7);"2 - Add";TAB(7);"3 - Exit"
```

This will put the information on 4 separate lines because when the computer tries to perform the TAB(7) it finds that that location has been already bypassed on the present row and therefore it automatically goes to the next row.

18. Did you know that you could delete a file when you close it. The statement is: CLOSE #1:DELETE

19. When programming in XB it pays in two ways to squeeze as many statements as you can into each program line. The first reason is that it saves memory by eliminating line numbers; the second is that it speeds up execution by eliminating the need for the program to process extra lines of code.

20. Another method to save memory by reducing the size of a program is to replace a constants used with a variable. This is assuming that that constant is used a number of different times in the program.

21. When you are editing a program and accidentally erase a line by pressing FCTN 3 to get the line back simply type in a single quote mark and then press ENTER. This gives a syntax error and the erased line is back because the change was not syntactically correct and thus not acceptable. The putting in of the quote mark must be done before moving from the line that was erased.

22. If you need to know if CALL INIT has already been executed in your program put in the following lines:

```
10 CALL PEEK(8198,A,B)::IF A=170 AND B=85 THEN xxx
ELSE CALL INIT !!xxx = line number to go to if CALL INIT
already executed.
```

23. Here is a use of the MIN and MAX statements:

MIN - If a variable is restricted to being no higher than 6 you would normally say IF A>6 THEN A=6 however you can say A=MIN(A,6)

MAX - If a variable is restricted to being no lower than 6 you would normally say IF A<6 THEN A=6 however you can say A=MAX(A,6)

24. You can LIST a program to disk by stating LIST "DSKn.program". This gives a D/V 80 file which is then readable by TI-Writer. This can be helpful for putting program listings in documents and allows you to use RS to amend or FS to locate something in a long program.

25. Here is a short program to write DATA lines which can then be merged into another program.

```
100 ON WARNING NEXT
110 DISPLAY AT(10,1)ERASE ALL:"ENTER FIRST LINE NUMBER:
    ":: ACCEPT AT(10,25)BEEP VALIDATE(DIGIT)SIZE(4):LN
120 DISPLAY AT(12,1):"ENTER INCREMENT"::
    ACCEPT AT(12,17)BEEP SIZE(3)VALIDATE(DIGIT):I
130 DISPLAY AT(14,1):"ENTER FILENAME:" :: ACCEPT AT(14,
    16)BEEP VALIDATE(UALPHA,DIGIT)SIZE(10):FN$
140 OPEN #1:"DSK1."&FN$,VARIABLE 163
150 DISPLAY AT(2,6)ERASE ALL:"PRESS ENTER TO END"::
    DISPLAY AT(22,1):"ENTER A LINE OF DATA:"::
    LINPUT D$
160 IF D$="" THEN 190
170 PRINT #1:CHR$(INT(LN/256)&CHR$(LN-256*INT(LN/256))
    &CHR$(147)&D$&CHR$(0)
180 LN=LN+I::
    GOTO 150
190 PRINT #1:CHR$(255)&CHR$(255)
200 CLOSE #1::
    END
```

26. This will save your DATA lines in a Merge format almost ready to be merged into you program. Before this can be done you must do the following:

i) Type NEW and press ENTER to clear memory
ii) MERGE in the saved DATA lines. (ie - MERGE DSK1.filename
iii) EDIT each DATA line by retyping (typing over) the word DATA
iv) SAVE the edited DATA lines in the MERGE format (ie - SAVE DSK1.filename, MERGE)

It is now ready to be put into your program.

THIS ARTICLE HAS BEEN PUT TOGETHER FROM MANY SOURCES BY BILL SPONCHIA.

Colour Monitor Fault

on the Wang PM005, by Dry Cell

Symptom: Press the Power switch and we get the telltale hiss while the switch is pressed but the monitor does not come on. Actually this was the ultimate state. Over a period of time the Monitor did switch on after the switch had been activated once or twice.

TO REMOVE THE FRONT PLASTIC CASE
FOR ACCESS TO THE MAINS SWITCH.

* Pull plugs 13 (red) and 16 (white) from the side RGB Board.
* Pull the Red plug from the H/V/Size potentiometer Plug in Board.
* Pull the Large Red Plug (with a wire loop) from under the tube on the bottom.
* Pull the Tube socket board from the Picture Tube.
* Undo the two Phillips Head screws in between the rails.
* Undo the two Phillips head screws that hold swivel clips. Remove the clips.
* Pull the Ultor plug out of the Tube Ultor Socket.
* Lift the tube slightly to unsolder the red lead from the tube wire holder.

REMOVE THE PICTURE TUBE AND ATTACHED BOARDS FROM
THE PLASTIC FRONT.

The Mains switch is a Double pole Double Throw type and fits in a plastic shell that clips into the Mains input board. There is a fuse on this board which is a particular type and obviously should not blow too often as it would require to remove the front plastic face to do it. The Mains Switch clips into its Plastic Shell.

The fault was found to be the switch itself because the neutral side was not remaining thrown but would break again when the switch was released. Since a replacement was not available the temporary cure was to solder the active and neutral leads to their respective lead mates and thoroughly insulate them from each other and the rest of the monitor electronics. This switch had a shortcoming in that there was little visual indication that the monitor had been left on. When a replacement is obtained an attempt will be made to fit a rectangular LED in with a dropping resistor to the Positive 12 volt regulator which is situated fairly close to the Mains Board. In the meantime it works well as is but the hole in the front of the monitor is unsightly.

Since the components on the Main Board are not easy to access some observations were made and voltages taken from the RGB board on the side which is easy to get at. The readings were taken as indicated below.

Observations of the three large IC's V700 on the RGB Board is a TDA 3301 B V520 is a TDA 1870 and V453 is a TDA 7812A.

As the readings are easily measured on the RGB Board on the side, readings are recorded using a Beckman DMM as shown below.

continued on page 34

A Look at Assembler #7

Counting

by R.A. Green, Ontario Canada

This is a variation of the "string searching" problem we looked at last time. This time, as well as searching for the first blank in a string, we also want to know the number of non-blank characters that we have searched. The TI99/4A has some neat instructions for counting: INC and DEC for counting by ones, INCT and DECT for counting by twos. So, using what we learned last time, we immediately drop into Assembler Language (and get rid of all the casual readers who thought an article on counting would be easy) and use the neat INC instruction to count the non-blanks as we search past them.

```
       LI    R1,RECORD            R1->the string
       LDB   *R1+,R2              Load length byte
       A     R1,R2                R2->just past end
       MOVB  @BLANK,*R2           Put blank at end
       CLR   R3                   Zero non-blank counter
LOOK   IFB   *R1+,EQ,@BLANK,DONE  Find a blank
       INC   R3                   Count the non-blank
       JMP   LOOK                 and keep looking
DONE   EVEN
       ....
RECORD BSS   82             Length, string, 1 blank
```

So there we are, a nice little 4 instruction loop that counts the number of characters up to the first blank and which does not have to test for the end of the string because we put a blank there.

Our 4 instruction loop does have two jumps in it though. Seems excessive to need two jumps for just one loop, maybe we can do better.

```
       LI    R1,RECORD            R1->the string
       LDB   *R1+,R2              Load length byte
       A     R1,R2                R2->just past end
       MOVB  @BLANK,*R2           Put blank at end
       CLR   R3                   Zero non-blank counter
COUNT  INC   R3                   Count characters
       IFB   *R1+,NE,@BLANK,COUNT Find a blank
       DEC   R3                   Correct count
       ....
RECORD BSS   82             Length, string, 1 blank
```

That is better! We now have a 3 instruction loop because we counted all the characters looked at, then corrected the count at the end because we counted the blank. Hey this Assembler Language Game is fun, eh?

Now there is only one little problem. This code makes a mistake. Admittedly it corrects its mistake, but it does count one too many and then has to back up. That is because the code was written by a person who learned as a kid to count begining at nothing (zero). Computers could not care less. If we factor out the counting loop, what do we have?

```
       CLR   R3             Zero non-blank counter
       DEC   R3             Correct count
```

This is exactly equivalent to:

```
       SETO  R3             Reset count
```

So, now counting from -1 instead of zero.

```
       LI    R1,RECORD            R1->the string
       LDB   *R1+,R2              Load length byte
       A     R1,R2                R2->just past end
       MOVB  @BLANK,*R2           Put blank at end
       SETO  R3                   Non-blank counter -1
COUNT  INC   R3                   Count characters
       IFB   *R1+,NE,@BLANK,COUNT Find a blank
       ....
RECORD BSS   82             Length, string, 1 blank
```

We now have quite a nice tight piece of code now. When you think about it though you realize computers cannot count, they can add 1 to a number, but it is the programmer who is counting. People count because it is easier than arithmetic but computers do not care.

Now we get to the point of this article on counting. When you have to count, do it well, but do not count unless you have to; make the computer compute not count. So here is the final version of the code for this problem.

```
       LI    R1,RECORD            R1->the string
       LDB   *R1+,R2              Load length byte
       A     R1,R2                R2->just past end
       MOVB  @BLANK,*R2           Put blank at end
       MOV   R1,R3                Save start of string
LOOK   IFB   *R1+,EQ,@BLANK,LOOK  Find a blank
       S     R3,R1          R1 = count of non-blanks
       ....
RECORD BSS   82             Length, string, 1 blank
```

Notice the loop is now only 2 instructions and has changed from counting to just looking.

Next time we will look at data compression and maybe have a chance to use our string searching and counting loops.

"I haven't lost my mind! I've got it archived on a disk here somewhere."

"Either that, or I've accidently uploaded it to CompuServe."

A Look at Using GPLLNK

by R.A. Green, Ontario Canada

In a previous article in this newsletter I gave a general purpose GPL link routine. This time I will describe how to use some of the GPL subroutines.

Some of the descriptions of the GPL routines in the Editor Assembler and MiniMem manuals are not complete and there are no actual examples of coding. In the following I will describe some GPL routines in detail and then give a coding example using them. Remember, if you are using the Editor Assembler or MiniMem GPLLNK, you must set the GPL STATUS byte at >837C to zero just before calling GPLLNK.

```
*-------------------------------*
* GPL Routine >0016             *
* Load Standard Character Set   *
*-------------------------------*
```

This routine is slightly misnamed. The character set loaded is the 8 pixel high upper case characters used on the title screen. The patterns for characters 32 to 95 are loaded at the VDP address specified in FAC (>834A). Character 32, the first one loaded, is a blank. Character 95, the last one loaded is the underscore.

When using the Extended BASIC setup for the VDP, the pattern for the blank is loaded at VDP address >0400. When using the Editor Assembler setup for the VDP, the pattern for the blank is loaded at VDP address >0900.

```
*-------------------------------*
* GPL Routine >0018             *
* Load Small Character Set      *
*-------------------------------*
```

Again, this routine is slightly misnamed. The character set loaded is the "standard" one used by BASIC, Extended BASIC, Editor Assembler, etc. The characters are the 7 pixel high upper case characters. The patterns are loaded for the same characters as for Routine >0016.

```
*------------------------------*
* GPL Routine >004A            *
* Load Lower Case Character Set *
*------------------------------*
```

At last, this routine is named correctly. It loads the patterns for characters 96 to 126 into the VDP at the address specified in FAC. Character 96 is the single quote and character 127 is the tilde. When using the Extended BASIC setup for the VDP, the pattern for the quote is loaded at VDP address >0600. When using the Editor Assembler setup for the VDP, the pattern for the quote is loaded at VDP address >0B00.

```
*------------------------------*
* GPL Routine >0034            *
* Accept Tone                  *
*------------------------------*
```

This routine issues a BEEP. It makes use of the sound list player in the system ROM. The sound list player can process a sound list in either VDP RAM or in GROM. The sound list for BEEP is in GROM so no VDP memory is used. As with playing any sound list, interrupts must be enabled at least once every 1/50th of a second (in Australia). The sound list will not even begin to play until the next VDP interrupt occurs.

If required, your program can wait till the BEEP is finished by looking at the byte at >83CE (with interrupts enabled) until it becomes zero.

```
*------------------------------*
* GPL Routine >0036            *
* Bad Response Tone            *
*------------------------------*
```

This routine issues a HONK. It works in the same way as Routine >0034.

```
*
*TITLE:   GPL Routine Examples
*AUTHOR:  R.  A.  Green
*
*
*PAD definitions
*
FAC     EQU  >834A          GPL Floating ACcumulator
SCTR    EQU  >83CE          Sound list counter byte
KBNO    EQU  >8374          Keyboard number for KSCAN
KEYVAL  EQU  >8375          Key value from KSCAN
*
        REF  RAGLNK         GPL Link
        REF  VMBW,KSCAN
*
*EXAMPLE 1: Load character definitions for the
*           BASIC or Extended BASIC environment.
*
        LI   R0,>0400       VDP address for blank
        MOV  R0,@FAC        To FAC
        BLWP @RAGLNK        Call my GPLLNK routine
        DATA >0018          To load >20 to >5F characters
        LI   R0,>0600       VDP address for quote
        MOV  R0,@FAC        To FAC
        BLWP @RAGLNK        Call my GPLLNK routine
        DATA >004A          To load >60 to >7E characters
*
*EXAMPLE 2: Load character definitions for the
*           Editor Assembler, MiniMemory or
*           TI-Writer environment.
*
        LI   R0,>0900       VDP address for blank
        MOV  R0,@FAC        To FAC
        BLWP @RAGLNK        Call my GPLLNK routine
        DATA >0018          To load >20 to >5F characters
        LI   R0,>0B00       VDP address for quote
        MOV  R0,@FAC        To FAC
        BLWP @RAGLNK        Call my GPLLNK routine
        DATA >004A          To load >60 to >7E characters
*
*EXAMPLE 3: Load large capitals definitions for game
*           title screen in the Extended BASIC
*           environment.
*
        LI   R0,>0400       VDP address for blank
        MOV  R0,@FAC        To FAC
        BLWP @RAGLNK        Call my GPLLNK routine
        DATA >0016          To load >20 to >5F characters
*
*EXAMPLE 4: Prompt user with a BEEP then wait for
*           key press.
*
        LI   R0,>02C2       Bottom line of screen
        LI   R1,MSG1        "PRESS ENTER TO CONTINUE"
        LI   R2,23          Lenght of message
        BLWP @VMBW          Write message to screen
        BLWP @RAGLNK        Call my GPLLNK routine
        DATA >0034          To start a BEEP
WAIT1   LIMI 2              Allow interrupts for BEEP
        LIMI 0              Stop them
        MOVB @KB,@KBNO      Set Keyboard number
        BLWP @KSCAN         Check for key press
        CB   @KEYVAL,@ENT   Was key ENTER?
        JNE  WAIT1          Jump no, wait
*
*EXAMPLE 5: Issue error message with HONK
*           and wait until it finishes
*
        LI   R0,>02C2       Bottom line of screen
        LI   R1,MSG2        "INVALID INPUT"
        LI   R2,13          Lenght of message
        BLWP @VMBW          Write message to screen
        BLWP @RAGLNK        Call my GPLLNK routine
        DATA >0036          To start a HONK
        LIMI 2              Allow interrupts for HONK
WAIT2   MOVB @SCTR,R0       HONK finished yet?
        JNE  WAIT2          Jump no, wait
        LIMI 0              Stop them
*
*DATA
*
MSG1    TEXT 'PRESS ENTER TO CONTINUE'
MSG2    TEXT 'INVALID INPUT'
KB      BYTE 0              Keyboard number
ENT     BYTE 13             ENTER key value
*
        END
```

continued from page 16

REVIEW- JIFFY FLYER- DISK - US$10 from Comprodine + $4 post.

Rather an old program but I did not seem to have reviewed it yet. It is a remarkably simple program! It enables you to quickly prepare a single sheet poster, using a border (several are part of the program); a large type heading, a small type announcement over several lines (several fonts are part of the program), and either an internal border or four corners made up of the same graphic, which you load from disk- the small pictures are to be in CSGD format, several are supplied, and the Disk Library has many more.

And that really is that!

The format is pretty fixed, which makes for a fairly straight forward and easy to use program- the more powerful a program the less easy it is to use (The Printers Apprentice is about ten times harder to use than this program!). A very similar utility can be found in ARTIST PRINTSHOP from the very same publisher, with extra facilities, but priced at $25. Artist Printshop is the more versatile but if your budget is tight, this little utility, at less than half the price, may well suit.

REVIEW- CERTIFICATE 99- disk from Comprodine, $7 plus $4 post.

This is again a rather old program, which allows you to fill an A4 sheet of paper with a computer produced certificate, with a choice of borders, a choice of graphics, a choice of fonts, and a choice of signatures (perhaps Mrs Thatcher is no longer appropriate?). There are support disks available from both Comprodine AND Notung with extra fonts etc etc.

continued on page 17

Decoding EPROM files

File Handling Techniques part 2

by Ben Takach

This is the second part of a series of articles that Ben has put together with his usual meticulous care and presentation. There are more to come so we hope you enjoy them and benefit from them. What follows below carries on from last month's article so refer back to it to get the proper flow.

```
10 ! SAVE DSK1.FILE-I/O
100 CALL CLEAR :: DISPLAY AT(10,1):"Insert 'DEFIN-FILE'
    disk in drive 1, lift lever and push any key."
105 CALL KEY(0,R,S):: IF S<>1 THEN 105
110 OPEN #2:"DSK1.DEFINITION",RELATIVE 4100,INTERNAL,
    FIXED 40
112 INPUT "LAST REC. No. ON THE FILE:? ":TOTAL
115 GOTO 220
120 PRINT "This routine enables you to type in new
    records one after the other sequentially": :
130 REM ENTRY SECTION
140 ! INPUT "LAST REC. No. ON THE FILE:? ":TOTAL
145 PRINT : : : :: PRINT "ENTER 'XXX' TO LEAVE THIS
    PROGRAM SEGMENT.": :
150 INPUT "RECORD "&STR$(TOTAL+1)&" ":BUFFER$
155 IF LEN(BUFFER$)>40 THEN PRINT "Record length exceeds
    40 characters, re enter it!": : : :: GOTO 150
160 IF BUFFER$="XXX" THEN 220
170 PRINT #2,REC TOTAL+1:BUFFER$
180 TOTAL=TOTAL+1
190 IF TOTAL<4100 THEN 150
200 REM PRINT FULL MESSAGE
210 PRINT "FILE FULL"
220 REM CONTROL SECTION
230 PRINT : :"1=PRINT FILE"
240 PRINT "2=PRINT A RECORD"
250 PRINT "3=CHANGE A RECORD"
260 PRINT "4=ADD A RECORD"
270 PRINT "5=LEAVE THE PROGRAM"
271 PRINT "6=NUL STR.REC.ENTRIES"
272 PRINT "7=PRINTOUT OF RECORDS"
273 PRINT "8=ENTER RECORDS SEQUENTIALLY"
274 PRINT "9=CHANGE GROUPS OF RECORDS": : : : : :
280 INPUT "ENTER YOUR CHOICE? ":X
290 PRINT
300 IF X<1 THEN 220
310 IF X>9 THEN 220
320 ON X GOTO 1000,2000,3000,4000,5000,6000,7000,120,
    8000
1000 REM PRINT A FILE. Unedited TI routine. It would be
     nice to read it on the screen!
1010 RESTORE #2,REC 1
1020 FOR COUNT=1 TO TOTAL
1030 INPUT #2:BUFFER$
1040 PRINT COUNT;BUFFER$
1050 NEXT COUNT
1060 GOTO 220
2000 REM PRINT A RECORD. Edited TI routine. It will let
     you see as many records as you wish! Compare this
     with the original TI version.
2010 GOSUB 10000
2020 INPUT #2,REC RECNUM:BUFFER$
2030 PRINT RECNUM;BUFFER$
2031 PRINT
2032 PRINT "PRINT AN OTHER RECORD? (Y/N)"
2033 PRINT
2034 CALL KEY(0,RV,SV):: IF SV<>1 THEN 2034
2035 IF RV=89 OR RV=121 THEN 2000
2040 CALL CLEAR
2050 GOTO 220
3000 REM CHANGE A RECORD. Original TI routine, it is not
     very convenient to use!
3010 GOSUB 10000
3020 INPUT "ENTER NEW DATA? ":BUFFER$
3030 PRINT #2,REC RECNUM:BUFFER$
3040 GOTO 220
4000 REM ADD RECORD. Unedited TI routine. It is not very
     convenient to use!
4010 IF TOTAL=4100 THEN 210
4020 TOTAL=TOTAL+1
4030 INPUT "ENTER DATA?":BUFFER$
4040 PRINT #2,REC TOTAL:BUFFER$
4050 GOTO 220
5000 REM LEAVE THE PROGRAM
5010 CLOSE #2
5020 END
6000 REM NULL STRING RECORD ENTRIES.
6010 INPUT "FIRST REC.No. TO THE FILE:? ":ST
6020 INPUT "LAST REC.No. TO THE FILE:?  ":LST
6030 BUFFER$="                                        "
6040 FOR I=ST TO LST
6050 PRINT #2,REC I:BUFFER$
6060 REM
6070 NEXT I
6080 PRINT "LAST RECORD No. ON FILE IS: ";I
6090 GOTO 220
7000 REM PRINTER ROUTINE
7010 CALL CLEAR :: DISPLAY AT(4,1):"PRINTOUT DEVICE ?
     PIO" :: ACCEPT AT(4,19)SIZE(-10):P$
7020 OPEN #3:P$,OUTPUT
7030 INPUT "FIRST REC.No.TO THE PRINTER? ":FST
7040 INPUT "LASTT REC.No.TO THE PRINTER? ":LST
7050 FOR I=FST TO LST
7060 INPUT #2,REC I:BUFFER$
7070 PRINT #3:TAB(10);I;TAB(18);BUFFER$
7075 IF EOF(2)THEN 7090
7080 NEXT I
7090 GOTO 220
8000 REM CHANGE GROUPS OF RECORDS
8010 ! This is my preferred style of programming .
     The screen remains uncluttered. You will always
     see how far did you progress!
8015 ! It gets around the single line entry limitation
     of the ACCEPT AT command.
8020 CALL CLEAR
8030 DISPLAY AT(2,1):"First REC.No.to be changed?:"
8040 DISPLAY AT(5,1):"Last  REC.No.to be changed?:"
8050 !
8060 ACCEPT AT(3,3)SIZE(-4)VALIDATE(DIGIT):FST
8070 ACCEPT AT(6,3)SIZE(-4)VALIDATE(DIGIT):LST
8080 IF LST<FST THEN GOSUB 8300 :: GOTO 8060
8090 IF FST<0 AND FST>4100 THEN GOSUB 8300 :: GOTO 8060
8100 IF LST>4100 THEN GOSUB 8300 :: GOTO 8070
8110 GOSUB 8500 :: R=22 :: C=1 :: GOSUB 8400
8120 IF Y=1 THEN Y=0 :: GOTO 8060
8130 FOR I=FST TO LST
8140 INPUT #2,REC I:BUFFER$
8144 BUF$=SEG$(BUFFER$,1,28):: FER$=SEG$(BUFFER$,29,12)
8150 DISPLAY AT(10,1):"Current RECORD ";I;" is:"
8160 DISPLAY AT(11,1):BUF$
8165 DISPLAY AT(12,1):FER$
8170 DISPLAY AT(14,1):"Enter new RECORD ";I;" :"
8180 REM
8181 LINPUT BU$
8182 IF LEN(BU$)>40 THEN DISPLAY AT(19,1):"Record length
     exceeds 40 characters, reenter it!" :: GOTO 8150
8185 BUFFER$=BU$ :: DISPLAY AT(19,1): :: DISPLAY AT(20,
     1): :: DISPLAY AT(22,1): :: DISPLAY AT(23,1):
8186 DISPLAY AT(16,1):BUFFER$
8187 REM
8190 GOSUB 8500 :: R=22 :: C=1 :: GOSUB 8400
8200 IF Y=1 THEN Y=0 :: GOTO 8181
8210 PRINT #2,REC I:BUFFER$
8220 R=16 :: C=1 :: GOSUB 8400 :: R=17 :: GOSUB 8400
8230 NEXT I
8290 GOTO 220
8300 DISPLAY AT(22,1)BEEP:">>Invalid entry, REENTER!<<"
     :: RETURN
8400 DISPLAY AT(R,C): :: RETURN
8500 DISPLAY AT(22,1)BEEP:">>Any correction? (Y/N)  <<"
8510 CALL KEY(0,RV,SV):: IF SV<>1 THEN 8510
8520 IF RV=ASC("Y")OR RV=ASC("y")THEN Y=1
8530 RETURN
10000 INPUT "WHICH RECORD? ":RECNUM
10005 IF RECNUM=-1 THEN 220
10010 IF RECNUM<1 THEN 10030
10020 IF RECNUM<=TOTAL THEN 10050
10030 PRINT "INVALID RECORD NUMBER (ENTER -1 TO GET OUT
      WITH GRACE)" :: PRINT : : :
10040 GOTO 10000
10050 RETURN
```

Now I will return to the original objective of creating an eprom decoding program. In the definition file created by the above program, one has to assign a specific 3 digit hex code (3 nybbles) to each item on the list. For example a certain limit switch (which is an input) is connected to input card 1 terminal 5. This switch may be labelled SW65. The address of input card 1 terminal 5 is >004. The entry in the definition file will have to be: REC4: SW65 1/5 safety guard is closed. This entry will identify the switch, its terminal on the PLC and the function of the switch. Naturally the record numbers will be consecutive decimal numbers. Eventually these will have to be converted to hex numbers, respectively hex numbers have to be converted to decimal.

In order to keep the decoding program short, and to be able to use it with different PLC programs one has to write to every one of the 4100 records even if some of these have no operand. The finished file is therefore likely to have several hundred null string records. The entry section of the above program will auto increment, however pressing the enter key 2000 times in succession is not practical. In any case it would not necessarily exclude the already mentioned garbage. A better approach is to use option 6 first and fill the entire file with nothing (spaces).

This brings us to the next point. The computer will loop through the operation 4100 times. The disk control card will save 6 records to its buffer, then it will access the drive and print 1 sector. This will be repeated some 684 times (684x6 records = 4104 records, thus the last sector will only have 2 records). This operation will take quite some time! Things can be speeded up by opening the file on the ram disk and when completed (in a matter of minutes), then it should be copied to disk. Now one can go over it again and enter each group of records in the desired sequence, by choosing option 8.

This example highlights the difference between a "fair weather" program and an "all weather" program; the latter is not only faster and easier to use, but it will not crash under the stormiest conditions. It is a bad practice to limit a program to the bare essentials, likewise indiscriminate over-use of protection will make the program cumbersome and unwieldy. It also shows that it is quicker to start with an existing program and change it to suit, rather than write one from scratch, although the finished article is generally neater and cleaner if one designs it from scratch.

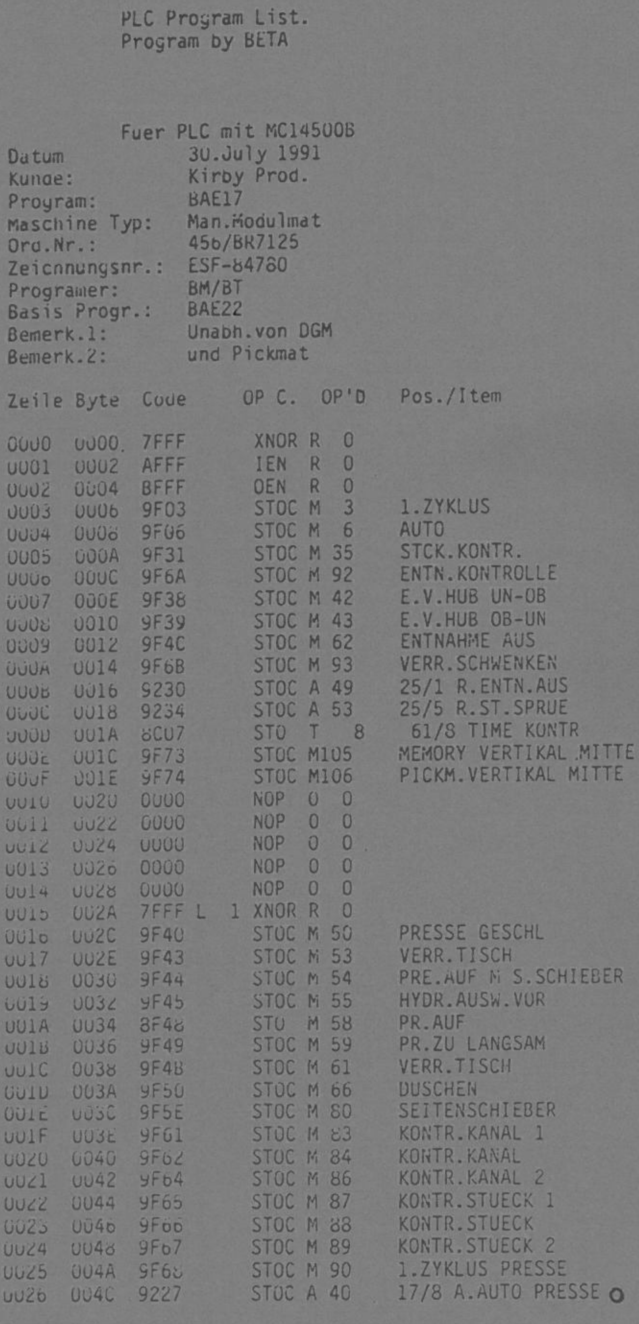
Once the problem of converting the eprom code and the definition file in XB compatible format has been solved, we can design the conversion program.

Essentially we will have a Hex word input, which has to be decoded and printed out in the following format:

```
                                               I/O CARD &
                                               TERMINAL POS.

LINE    EPROM   EPROM  INSTRUCTION  OPERAND    INP.DEFINITIONS
NUMBER  ADDRESS CODE   MNEMONIC                OUTP.DEFINTNS

(4 HEX  (4 HEX  (4 HEX  (UP TO 4     (UP TO 4    (MAX 23
DIGITS) DIGITS) DIGITS) CHARACTERS) CHARACTERS) CHARACTERS)
```



```
          PLC Program List.
          Program by BETA


          Fuer PLC mit MC14500B
Datum                30.July 1991
Kunde:               Kirby Prod.
Program:             BAE17
Maschine Typ:        Man.Modulmat
Ord.Nr.:             456/BR7125
Zeicnnungsnr.:       ESF-84780
Programer:           BM/BT
Basis Progr.:        BAE22
Bemerk.1:            Unabh.von DGM
Bemerk.2:            und Pickmat

Zeile Byte  Code       OP C.  OP'D   Pos./Item

0000  0000. 7FFF        XNOR R  0
0001  0002  AFFF        IEN  R  0
0002  0004  BFFF        OEN  R  0
0003  0006  9F03        STOC M  3    1.ZYKLUS
0004  0008  9F06        STOC M  6    AUTO
0005  000A  9F31        STOC M 35    STCK.KONTR.
0006  000C  9F6A        STOC M 92    ENTN.KONTROLLE
0007  000E  9F38        STOC M 42    E.V.HUB UN-OB
0008  0010  9F39        STOC M 43    E.V.HUB OB-UN
0009  0012  9F4C        STOC M 62    ENTNAHME AUS
000A  0014  9F6B        STOC M 93    VERR.SCHWENKEN
000B  0016  9230        STOC A 49    25/1 R.ENTN.AUS
000C  0018  9234        STOC A 53    25/5 R.ST.SPRUE
000D  001A  8C07        STO  T   8    61/8 TIME KONTR
000E  001C  9F73        STOC M105    MEMORY VERTIKAL .MITTE
000F  001E  9F74        STOC M106    PICKM.VERTIKAL MITTE
0010  0020  0000        NOP  0  0
0011  0022  0000        NOP  0  0
0012  0024  0000        NOP  0  0 .
0013  0026  0000        NOP  0  0
0014  0028  0000        NOP  0  0
0015  002A  7FFF L  1 XNOR R  0
0016  002C  9F40        STOC M 50    PRESSE GESCHL
0017  002E  9F43        STOC M 53    VERR.TISCH
0018  0030  9F44        STOC M 54    PRE.AUF M S.SCHIEBER
0019  0032  9F45        STOC M 55    HYDR.AUSW.VOR
001A  0034  8F48        STO  M 58    PR.AUF
001B  0036  9F49        STOC M 59    PR.ZU LANGSAM
001C  0038  9F4B        STOC M 61    VERR.TISCH
001D  003A  9F50        STOC M 66    DUSCHEN
001E  003C  9F5E        STOC M 80    SEITENSCHIEBER
001F  003E  9F61        STOC M 83    KONTR.KANAL 1
0020  0040  9F62        STOC M 84    KONTR.KANAL
0021  0042  9F64        STOC M 86    KONTR.KANAL 2
0022  0044  9F65        STOC M 87    KONTR.STUECK 1
0023  0046  9F66        STOC M 88    KONTR.STUECK
0024  0048  9F67        STOC M 89    KONTR.STUECK 2
0025  004A  9F68        STOC M 90    1.ZYKLUS PRESSE
0026  004C  9227        STOC A 40    17/8 A.AUTO PRESSE ○
```

Figure 1: Example of EPROM program Listing

continued from page 30

Pins of V700 on RGB Board

Test Point	Voltages	Pin	Volts	Pin	Volts
608	0.15	1	2.2	19	0.6
609	0.26	3	10.5	20	3.4
611	0.68	4	10.0	21	5.7
612	1.71	5&6	0.8	23	3.8
613	0.89	7	3.0	24	1.5
614	3.08	8	3.0	26	1.4
615-R	3.5	9	0.6	27	1.1
616-G	4.82	10	6.5	28 & 732	0
617-B	5.2	14	5.2	29	9.6
635	105.3	15	5.6	30	2.1
636	77.9	16	0.8	31	3.4
637	78.3	17	4.8	32	3.0
732 & Pin 28	0	18	5.0	33 & 327	7.6
327 & Pin 33	7.6			39 & 40	11.9 ○

continued from page 4

We have been able to obtain a better deal on bulk purchases of software from Notung Software, so there will be a price reduction of about 5-10% for their software only.

TI-Base has been on back order for quite a while now and will be available as soon as the new version is released. A brand new manual is being produced for the new version (V4.0), so it should be a significant upgrade over V3.02.

For those of you using the TI-Base software, keep you eyes open for any reference manuals on dBase II, which is an older, superceded version of the well known relational data base for the IBM PC market. Apparently Dennis Faherty, the author of TI-Base, used a lot of the features found in dBase II when he developed TI-Base, so many of the commands in TI-Base, as well as the way it functions, is very similar to dBase II. ○

Regional Group Reports

Meeting Summary For December

Banana Coast	08/12/91	Sawtell
Central Coast	14/12/91	Saratoga
Glebe	12/12/91	Glebe
Hunter Valley	14/12/91	Boolaroo
Illawarra	16/12/91	Keiraville
Liverpool	13/12/91	
Northern Suburbs	26/12/91	
Sutherland	20/12/91	Jannali

BANANA COAST Regional Group
(Coffs Harbour Environs)

We never miss meeting at Kerry Harrison's residence 15 Scarba St. Coffs Harbour, 2 pm second Sunday of the month. Visitors are most welcome. Contact Kerry 52 3736, Kevin 53 2649, Rex 51 2485 or John 54 1451.

CENTRAL COAST Regional Group

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

GLEBE Regional Group

Regular meetings are normally on the Thursday evening following the first Saturday of the month, at 8pm at 107 Arundel St. Glebe. Contact Mike Slattery, (02) 692 8162.

HUNTER VALLEY Regional Group

The first regional meeting of the group was held at the Boolaroo Ambulance Station to discuss future plans. All welcome. Please contact Geoff Phillips on (049) 428 176 for details and the exact date for future meetings.

ILLAWARRA Regional Group

Regular meetings are normally held on the second Monday of each month after the TIsHUG Sydney meeting, except January, at 7.30pm, Keiraville Public School, Gipps Rd, Keiraville, opposite the Keiraville shopping centre. A variety of activities accompany our meetings, including Word Processing, Spreadsheets and hardware repairs. At the last meeting we looked at Bob's malfunctioning ramdisk and discovered that a track had been accidentally cut (easily fixed!). We also looked at aspects of TI-Base. Contact Lou Amadio on (042) 28 4906 for more information.

LIVERPOOL Regional Group

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

NORTHERN SUBURBS Regional Group

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

Come and join in our fun. Dick Warburton.

SUTHERLAND REGIONAL REPORT

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

Peter young
Regional Co-ordinator

TIsHUG in Sydney

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

DECEMBER MEETING - 7TH DECEMBER

The main aim of the December meeting will be to have the Annual General Meeting and then a relaxing time of fellowship around some nice food. Make sure you bring the November issue containing all the minutes of the last AGM.

****

The cut-off dates for submitting articles to the Editor for the TND via the BBS or otherwise are:

February 12 January

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

** o

continued from page 22

```
90 A(I)=INT(RND*100)+1
100 PRINT A(I);
110 NEXT I
120 PRINT
130 REM
140 REM ***** SIMPLE BUBBLE SORT *****
150 REM
155 C=0
156 S=0
160 J=1
165 Q=0
170 FOR I=1 TO N-J
175 C=C+1
180 IF A(I)<=A(I+1)THEN 220
185 S=S+1
186 Q=1
190 T=A(I)
200 A(I)=A(I+1)
210 A(I+1)=T
220 NEXT I
230 IF Q=0 THEN 250
235 J=J+1
240 REM
250 REM ***** ROUTINE TO PRINT LIST OF NUMBERS *****
260 REM
270 PRINT
280 FOR I=1 TO N
290 PRINT A(I);
300 NEXT I
310 PRINT
315 PRINT
320 PRINT C;"COMPARISONS WERE MADE"
330 PRINT
335 PRINT S;"SWAPS WERE MADE"
```

The variable Q is simply a flag that is set to zero each time the inner loop is initiated and is set to 1 if a swap is made. Thus, if the inner loop is exited with Q still having a value of zero the sort is complete.

The following results were obtained with the above version of the sorting routine.

```
HOW MANY NUMBERS DO YOU WANT? 15

64 48 2 79 36 5 66 71 100 24 14 67 57 1 3 34

2 5 13 14 24 34 36 48 57 64 66 67 71 79  100

102 COMPARISONS WERE MADE

56 SWAPS WERE MADE
```

The number of comparisons decreased slightly indicating that some savings were realised by this modification. Although this program is not likely to set any speed records it is now quite efficient and can be quite useful.

Next month we will look at the modifications necessary to use this program to sort string variables as well as look at the concept of using a pointer. o