



NEWS DIGEST

Focusing on the TI99/4A Home Computer

Volume 11, Number 3

April, 1992

Registered by Australia Post - Publication No. NBH5933

Five years of TND production in Wollongong



1987 to 1992

Sydney, New South Wales, Australia

\$3

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

April 1992

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) 85 5519

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

Russell Welham (043) 92 4000

Software library

Rolf Schreiber (042) 85 5519

Technical co-ordinator

Geoff Trott (042) 29 6629

TI-Faire co-ordinator

Dick Warburton (02) 918 8132

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

Geoff Trott (042) 29 6629

Liverpool

Larry Saunders (02) 644 7377

Northern Suburbs

Dennis Norman (02) 452 3920

Sutherland

Peter Young (02) 528 8775

Membership and Subscriptions

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

TIshUG Sydney Meeting

The April Meeting will start at 2.00 pm on 4th of April at Ryde Infant School, Tucker Street, Ryde. The Assembly Class will run from 10.00 am to 1.00 pm for those wishing to learn.

Printed by

The University of Wollongong
Printery Services.

Index

Title	Description	Author	Page No.
Assembly class for 1992	Software hints	Mudie, Ross	24
Beginning Forth #14	Software hints	Raguse, Earl	25
Decoding EPROM files part 4	Software hints	Takach, Ben	19
DSCAN correction	Correction	Mudie, Ross	16
Editor's comment	General interest	Relyea, Bob	1
Extended mouse support	General interest		4
Games info series 2	Zork 2	Brown, Robert	11
Hollywood Hijinx part 3	Games review	Scorpio	17
Horizon utilities	Software review		22
Keyboard reader	Software hints	Webb, Bob	2
Letter to editor	Games info	Brown, Robert	2
Newsletter update	General interest	Relyea, Bob	26
Page Pro 99 and utilities	Software review	Relyea, Bob	6
Prime number generator	Software hints	Mac, C	21
Programming music part 1	Software hints	Peterson, Jim	23
Programs to type in	Software	Peterson, Jim	13
Regional group reports	General interest		27
Secretary's notebook	Club news	Phillips, Terry	2
Sorting, part 4	Software hints	Brubaker, Ron	24
Struck by lightning	General interest	Mudie, Ross	8
Talking about modems	Hardware hints	Brown, Robert	18
Talking book	General interest	Harris, Daniel	26
Techo time	RAMdisks, back ups	Trott, Geoff	7
TI-Writer font maker	Software hints	Stringfellow, James	4
TI-Writer formatter tip	Word processing	Hoyt, Harold	10
TI-Writer replace string	Word processing	Leshner, Jim	7
Tigercub reformatter+ v1.2	Word processing	Peterson, Jim	9
Time check	Software hints	West, Loren	15
TIshUG shop report	Club news	Harrison, Percy	5
TIshUG software column	Club software	Schreiber, Rolf	3
Treasurer's report	Club news	Trott, Geoff	7
Younger set	Programs	Maker, Vincent	22

Attention Newsletter Editors

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

Check your renewal date. Do not miss out.

Editor's Comment

by Bob Relyea

I am encouraged by the nice comments that the editing team and I get for the nice-looking and informative newsletter. Just last month I got a letter from Jim Peterson who stated that our TND was, by far, the best in the TI world. Coming from far-away America, and from Mr. TI himself, that was a nice and encouraging comment and helps to keep us going in the midst of the tedious and routine nature of this job. Everybody can play a part with an article here and there, and for those of you who contribute so regularly, a big THANK YOU. ○



TI SHUG (AUSTRALIA) LIMITED

(Incorporated in N.S.W.)

P.O. Box 1089
Strawberry Hills NSW 2012

Address all correspondence to
the Secretary.

TI-FAIRE

- When:** Saturday & Sunday, 28-29 November 1992
10am to 9pm on both days
- Venue:** Ashfield Boys High School Hall (next to Western Suburbs Leagues Club), Liverpool Road, Ashfield, NSW, Australia
- Contacts:** Phone (02) 918-8132
(02) 871-1514
(042) 85-5519
- Fax (02) 638-9757
9am to 5pm weekdays (Attention: Tom Marshall)
- Transport:** Bus Routes 480 & 483 from City Terminus (Queen Victoria Building) via George Street (City), Railway Square, Parramatta Road & Liverpool Road. Alight at the Victoria Street stop. Buses run every 30 minutes.
- Train Frequent services from the City & Liverpool. Change at Strathfield if coming via Parramatta or Hornsby lines.
- Parking:** Ample car parking is available in the school grounds.
Entrance at rear of Leagues Club.
- Food:** Plenty of local restaurants and take away outlets. The Leagues Club offers snack bar facilities throughout the day.
- Faire Details:** Displays of the latest TI-99/4A hardware and software. (Available for sale at the end of the faire).
Sponsor displays, both local and overseas.
Displays of rare items of TI-99/4A hardware from the past.
Displays of owner built peripherals & expansion systems.
Computer controlled (TI-99/4A) model train layout.
RAMdisk and EPROM RAMdisk demonstrations.
Page Pro 99 demonstrations.
Greeting Cards production.
Computer Speech demonstration.
Bulletin Board System (BBS) demonstration.
Demonstration of 80 Column Card & Colour Monitor capabilities.
Latest TI-99/4A Computer Games demonstrations.
MIDI music demonstration.
Screenings of videos from past TI-Faires in the USA.
TI SHUG Shop will be selling TI-99/4A software & hardware.
TI-99/4A Computer Repair Clinic and Technical Support.
Special TI-Faire edition of the TND will available for sale.

Accommodation: The following motel nightly rates are current for March 1992:

- * Palm Court (M), Room Only (Single) \$A78 to \$A88.
17 Parramatta Road, Haberfield. Phone (02) 797-6111.
- * Metro Motor Inn (M), Room Only (Single) \$A89 to \$A95.
63 Liverpool Road, Ashfield. Phone (02) 798-0333.
- * Phillip Lodge (M), Room Only (Single) \$A78 to \$A88.
156 Parramatta Road, Ashfield. Phone (02) 798-7666.
- * Charlotte Motor Inn (M), Bed & Breakfast \$A70 (Single), \$A70 (Double).
62 Charlotte Street, Ashfield. Phone (02) 798-8918.
- * All major credit cards accepted.
- * Billeting of overseas and interstate visitors may be available if several weeks notice is given.

Mail Enquiries: TIsHUG (Australia) Limited
P.O. Box 1089
Strawberry Hills NSW 2012
Australia

Secretary's Notebook

by Terry Phillips

It is membership renewal time! Well at least it is for the majority of members. Please make sure you renew on time so that important planning work can get under way for the year - so we know how many newsdigests to print, how many steaks/sausages to buy etc. See me at the meeting and I will happily take your renewal subscriptions or make out a cheque and mail it to our post office box. Remember fees are now \$35 and there is no additional fee for the BBS, talking of which there seemed to be a lot of interest in modems at the last meeting. If you want to be able to access the BBS and you have the necessary hardware/software, then see Ross Mudie for the allocation of a user number and password.

The Faire date has been set and a particularly good, handy venue has been booked. Make sure you are in town for this event which promises to be the highlight of the year. If you plan on doing any displays make sure you talk to Ian Mullins so he can arrange banners, power and space for you.

The Faire looks like it will not only attract local and interstate members and visitors but some visitors from overseas as well. I received the following letter from Bernard Krause, a member in Mira Loma, California:

"Having only been a member for one year I have found your newsletter of the highest quality both in print and contents of any other TI group in the USA. Wonderful and easy to follow articles touching on many different topics.

Hope to hear more on the when and where of the TI-Faire you have planned. The sooner a date is set the faster I can make those plane reservations."

Two new members signed up at the March meeting and it is a big welcome to:

Bruce Boese from Culburra, and
Harry Willey from Strathfield.

Hope you can both make it to future meetings.

Next meeting will be the first of the year's buy swap and sell days. These occasions are always fun and enable you to pick up a bargain or offload unwanted items. Get in early as the best of the bargains usually go fairly quickly.

Renew Now!

Letter to the Editor

I would just like to comment on the "Jenny's Younger Set" from the Jan/Feb TND.

Firstly I would like to congratulate Vincent Maker on his efforts in programming and writing to the TND, but I would also like to add that I also write regularly for the TND. Over the last 24 months I have not written a lot of articles because I was in year 12. But I am now working in a computer related position, working 6 days a week, programming in Cobol and attending university 3 evenings a week. Since my life has settled into a regular pattern, I was able to write some more articles for the TND.

I do not write these articles for international stardom, but rather I enjoy doing it! I must, because I am up to my 20th article. (If anyone does not know by now, I write the GAMES INFORMATION articles). So if anyone does want something reviewed drop me line (details at the end of my articles).

Yours the TI way,
Robert Brown,
Games Information

Keyboard Reader

by Bob Webb, USA

This small program is one of my most used programs. I can never remember the number associated with a key press or ASCII symbol, so I threw this thing together. Let me caution you before I continue- DO NOT run this program until you have saved it, as once you start it the only way to stop it is to turn your computer off.

Function Quit and Function 4 are disabled so you cannot break in or kill it. This was done so all combinations of key presses could be viewed. If you do not want these features, delete line numbers 160 to 190. You must delete line 170 if you do not have the memory expansion hooked up or a syntax error is generated..

Once this program is running, press any key- it's associated number will be displayed. If an ASCII symbol is associated with that particular key press, it will be displayed just to the right of the number.

This program does not break any new ground, however, you might find a part of it to be of use. I have added one of my favourite little details to it. If no key is pressed for a given amount of time, it jumps to a screen saver type of subprogram.

This BLANK variable is a counter. This clock ticks away and if a key is pressed it is reset to zero and begins again. If no key is pressed it jumps down to line 400 and stays there until a key is pressed.

```
100 ! KEY LOOP
110 ! BY BOB WEBB, 6/91
120 CALL CLEAR :: DISPLAY AT(10,7):"TEST"
130 GOSUB 180
140 CALL CLEAR :: DISPLAY AT(10,7):"ENTER"
150 ! 13 IS THE ENTER KEY
160 GOSUB 180 :: IF K=13 THEN 140
170 GOTO 120
180 ! PRESS ANY KEY LOOP
190 FOR BLANK=1 TO 200
200 CALL KEY(O,K,S):: CALL HCHAR(24,16,32)
210 IF S=1 THEN 250 :: CALL HCHAR(24,16,107)
220 NEXT BLANK :: CALL CLEAR
230 CALL KEY(O,K,S):: IF S=0 THEN 230
240 GOTO 190
250 RETURN
```

This second program can be added to your own program. It has the same kind of screen saver loop in it as the first. After the GOSUB statement you can test for which key was pressed (If K=13 then X). Happy Computing, and long live our 99/4A!

```
100 ! KEY TO NUMBER PROGRAM
110 ! EXTENDED BASIC AND 32K
120 ! BY BOB WEBB, 6/91
130 ! CAUTION: YOU WILL HAVE TO
140 ! TURN OFF COMPUTER TO END
150 !
160 ! CALL LOAD DISABLES QUIT
170 CALL INIT :: CALL LOAD(-31,806,16)
180 ! ON BREAK NEXT DISABLES FCTN 4
190 ON BREAK NEXT
200 !
210 CALL CLEAR
220 BLANK=0
230 DISPLAY AT(5,5):"KEY TEST PROGRAM"
240 DISPLAY AT(7,5):"PRESS ANY KEY"
250 DISPLAY AT(9,5):"IT'S NUMBER WILL"
260 DISPLAY AT(10,5):"BE DISPLAYED"
270 !
280 ! MAIN LOOP
290 !
300 CALL KEY(O,K,S)
310 BLANK=BLANK+1
320 IF BLANK>1000 THEN 410
330 IF S=0 THEN 300
340 DISPLAY AT(12,4):K
350 DISPLAY AT(12,10):CHR$(K)
360 BLANK=0
370 GOTO 300
380 !
390 ! BLANK SCREEN
400 !
410 CALL CLEAR
420 CALL KEY(O,K,S)
430 IF S=0 THEN 420
440 GOTO 410
```

TiSHUG Software

Column by Rolf Schreiber

I have received the missing documentation files for DM1000 Version 5.0, released last month, and have passed them onto Percy Harrison. Contact the shop for the upgraded DISK A473, if you have already bought it.

Software Releases for April

This month I am releasing five disks which I hope some of you will find interesting. Most of this software falls into the category of "applications software", which are usually utility programs of one sort or another. In case you are wondering why I am not releasing exciting new games as freeware each month, the simple fact is that all the new games being written are being released as commercial items, and cost in the vicinity of \$15 each.

This month I am listing all the filenames on each disk, to help you decide whether to buy it or not. If you like this feature, then please let Percy know.

DISK A481 is called ARTCONVERT V2, and was released by Barb Berg in 1986 as a commercial product under the TRIO+ software label. The program basically converts TI-Artist fonts and instances into TI-Writer format, which can then be printed through the formatter. The program is menu driven, and has options to convert a TI-Artist instance or small font, to merge and convert two instances, to merge two converted files, or to print a converted file catalogue. There are a large number of sample instances and fonts on the disk, together with an extensive documentation file to help you through every step of the way. The disk is DSSD, and is also available as two SSSD disks.

A481A Diskname: ARTCONVERT Format: SSSD

Filename	Size	Type / Length	P
ARTCONT+	3	INT/VAR	80
ARTCONVERT	58	INT/VAR	254
BACKGNDS_S	23	DIS/VAR	80
BOOK_F	46	DIS/VAR	80
BORDERS2_S	21	DIS/VAR	80
BORDERS_S	22	DIS/VAR	80
GARF3_I	12	DIS/VAR	80
GRAPHICS_S	16	DIS/VAR	80
HELP	92	DIS/VAR	80
LBLOCK_F	33	DIS/VAR	80
LOAD	2	PROGRAM	38
OUTLINE_F	10	DIS/VAR	80
SCRIPT_F	14	DIS/VAR	80
T+ARTDATA3	4	INT/VAR	80

A481B Diskname: T+ARTDATA4 Format: SSSD

Filename	Size	Type / Length	P
BACKGNDS_C	25	PROGRAM	6144
BACKGNDS_P	25	PROGRAM	6144
BORDERS2_C	25	PROGRAM	6144
BORDERS2_P	25	PROGRAM	6144
BORDERS_C	25	PROGRAM	6144
BORDERS_P	25	PROGRAM	6144
GARF1_I	13	DIS/VAR	80
GARF2_I	23	DIS/VAR	80
GARF3_I	12	DIS/VAR	80
GARF4_I	22	DIS/VAR	80
GARF5_I	15	DIS/VAR	80
ODIE2_I	18	DIS/VAR	80
ODIE_I	23	DIS/VAR	80
T+ARTDATA4	5	INT/VAR	80
UNICORN_I	41	DIS/VAR	80

DISK A482 is an assembly based utility disk from the New Horizons User Group. The authors of the programs on this SSSD disk were also involved with the initial design of the Horizon RAMdisk. The contents of this disk are fully described in another article in this month's TND.

Diskname: HORIZON Format: SSSD

Filename	Size	Type / Length	P
DATA	2	INT/VAR	80
DEMO	4	PROGRAM	541
DEMO/PLDMP	6	PROGRAM	1259
DSRLNK	7	DIS/FIX	80
DUMP	11	DIS/FIX	80
E/ADIS	20	DIS/FIX	80
HORIZONDOC	60	DIS/VAR	80
JUSTFY	6	DIS/FIX	80
LOAD	4	PROGRAM	556
LOGO	10	DIS/FIX	80
LOGO/PRINT	3	PROGRAM	382
MMDIS	20	DIS/FIX	80
MMDIS-LOAD	2	PROGRAM	193
PLOT	9	DIS/FIX	80
RECSCR	5	DIS/FIX	80
SAVSCR	6	DIS/FIX	80
SCREENS	3	PROGRAM	348
SCROLL	9	DIS/FIX	80
SCROLLDEMO	8	PROGRAM	1588
SORT	9	DIS/FIX	80
SORTDEMO	5	PROGRAM	952
SORTDEMO	5	PROGRAM	952
STAT	23	DIS/FIX	80
TEXTJUSTFY	4	PROGRAM	638
TIME	10	PROGRAM	2083
VDP	6	DIS/FIX	80
VDPDEMO	7	PROGRAM	1425
VENUS	10	PROGRAM	2083
XBDIS	40	DIS/FIX	80
XBDIS-LOAD	2	PROGRAM	213

DISK A483 is a useful assembly based utility written by Col Christensen of the Brisbane based TI-BUG User Group. The program is called TI-Tiler and provides a means of producing electronic circuit diagrams, saving the results to disk (as well as loading designs from disk) and printing the design to a printer. The program is not limited to creating electronic circuits, but may be used for diverse applications, such as producing professionally looking keyboard overlays. One of the great features of this program is its ability to rotate the page through 90 degrees, so that text can be rotated and printed sideways.

Diskname: TI-TILER Format: SSSD

Filename	Size	Type / Length	P
DOCS-TILER	57	DIS/VAR	80
LOAD	5	PROGRAM	820
LOADO	5	PROGRAM	847
OL	33	PROGRAM	8192
OM	6	PROGRAM	1278
OVERLAY_T	23	PROGRAM	5406
T1_T	9	PROGRAM	2039
T2_T	9	PROGRAM	2039
TT	33	PROGRAM	8192
TU	6	PROGRAM	1278

DISK A484 is a printer utility written by Ed & Mike Machonis which allows you to print different labels for different applications, including address labels, disk catalog labels and keyboard overlays, to name but a few. The program is menu driven and comes with a large number of examples.

Diskname: MAC-LABELS Format: SSSD

Filename	Size	Type / Length	P
!4A/TALK	2	DIS/VAR	80
!ADV/DIAG	2	DIS/VAR	80
!BASIC	2	DIS/VAR	80
!BLANK	2	DIS/VAR	80
!CHESS	2	DIS/VAR	80
!DISKO	2	DIS/VAR	80
!EDIT/ASM	2	DIS/VAR	80
!FAST-TERM	2	DIS/VAR	80
!FORTH	2	DIS/VAR	80

TISHUG Shop with Percy Harrison

Well it is hard to believe that the first quarter of 1992 has already passed us by and before we know it the TI-Faire will be with us. I hope some of you club members will help to make this event a huge success by offering assistance and letting the committee know of your willingness to do so as soon as possible so that they can get people organised and the various tasks that need to be done sorted out and allocated.

Over the past two or three years the stocks of software and hardware held in the shop has increased substantially and has reached the stage where I can no longer bring the total stocks to our monthly meetings as it is not possible to load everything into my vehicle. From the May meeting onwards I will not be bringing Hardware stocks to the meeting so if you require any hardware items please phone me at least two days before the meeting and let me know what you want and I will bring those specific items along with me to the meeting. Also, each month I will be removing a certain number of Software disks from stock to make room for the new software being released each month so if you require any of the disks after they have been taken off the stock list please let me know and I will run them off for you and bring them to the next meeting. However, please do not ring me the night before the meeting and expect me to have them there the next day.

Most of the software listed in last months TND will be available at the April meeting along with the new releases of Freeware and Commercial disks being released this month - see "TISHUG Software Column" for details.

We have two or three new members who are wanting to expand their basic system and are looking for the hardware to do this. So, if you have, or know of anyone that has surplus equipment for sale, would you please let me know so that we can help these new members out. In particular I am looking for a PE Mini System Disk Control Card as well as TI Peripheral Expansion Boxes.

Next month I will again list all the hardware available from the shop but in the meantime we still have about six new Double Sided Half Height Drives in stock and an assortment of Games and Utility Modules at very reasonable prices.

You will have read in last months TND that we can still get 80 Column Cards (TIM/SOB) at the discounted price of \$165 so if you want one you will need to get your order and money to me as quickly as possible and when I receive the cards they will be distributed in the same order in which payment was received. To those members who have already ordered cards please note that as at the time of writing this article we had not received any cards nor have we been advised of any dispatch date. We will keep you informed of anticipated delivery dates as soon as we hear from the supplier, in the meantime I would ask you to please be patient and do not phone me as I cannot give you any more definite information. ○

continued from page 23

```
1000 FOR J=1 TO T*4 :: CALL SOUND(-999,N(A),0):: CALL SO  
UND(-999,N(A)*1.01,0):: NEXT J :: RETURN
```

You can control the tempo by changing the value of 4, but not as precisely as with the previous method, and it does not work well with bass notes. Try changing the 1.01 to 1.02 - also try erasing the *1.01 and change the following 0 to 8, for a mandolin effect.

Those are just a few of the effects you can create with just a single-note melody - experiment and see what else you can discover.

So, just imagine what you will be able to do using all three voices - coming up in part 2 of this article! ○

continued from page 8

The next door neighbour's video recorder had a blown mains fuse. A capacitor across the mains in the video recorder had obviously flashed over, blowing the sides off the capacitor and taking out the fuse. Their telephone also failed to survive. The Telecom technician was apparently very busy with repairs all over the district.

The next day, there was another lightning strike, elsewhere in the local community. This time it struck TV antenna on another house.

Lightning does not always pick the highest object in the area, as was the case with both my antenna mast and the other ordinary TV mast. Between the bowling club where they saw sparks rising off the metal hat rack and my tower is the local water tower which is higher than the top of my tower.

WHAT LESSONS WERE LEARNED FROM ALL THE DAMAGE?

A better earth system on the antenna mast is a must, to provide a more direct path to earth and hopefully cut down the effect of the surge which appears to create a massive magnetic field. (Shades of an EMP!)

Any equipment which is not in use should be left UNPLUGGED from power points, with other cables such as speakers, antenna etc disconnected from the equipment. Turning off at the power point or the switch on the equipment will NOT provide sufficient protection.

Equipment which is protected by surge protection devices, (the BBS in this case), should not have any external connection which may breach the protection. In future, when the BBS alarm system is again connected, it is planned to use a length of optical fibre cable to couple the alarm out of the protected environment.

It is also a good idea to take notice of Telecom's warning and NOT to use the telephone during a thunder storm since electrocution or hearing damage could result from a lightning discharge. When a lightning strike occurs, it raises the earth crust potential by many thousands of volts in the immediate area of the lightning strike. The telephone line is earthed at the telephone exchange. This will be far enough away for the earth crust potential not to be affected by the lightning strike. The raised earth crust potential in the area of the lightning strike may result in a flashover to the telephone line. A person using the telephone under these circumstances, may become part of the flashover path with the possibility of it being lethal.

Even ordinary TV masts should be properly earthed for lightning protection to provide a direct path to earth, rather than through your equipment or your household wiring.

DESIGN OF ADDITIONAL EARTH SYSTEM

Protective earth systems must be individually designed to meet the unique needs of each installation and site. The soil in my back yard is clay, which is one of the better earthing materials, on top of ironstone. A total of 16 domestic copper clad steel earth stakes, 1.4 metres in length have been driven below the surface and connected to the tower by 6 separate 10mm (7/1.7mm) cables. These cables radiate out from each tower leg in fairly straight lines. One of the earth lines continues to the household electricity supply MEN point on the water pipe system. The television antenna on the house is also tied down to the same point. ○

continued from page 18

right upto 2400, including 1200/75, so keep this in mind when buying your modem, as the faster it goes, the quicker you can get that new program or the interesting GAMES INFORMATION file (that is a plug!) from the BBS.

TEXPAC BBS
(02) 456-4606

24 Hours, 7 Days a Week
Sysop : Ross Mudie ○

Page Pro 99

and related utilities

by Bob Relyea

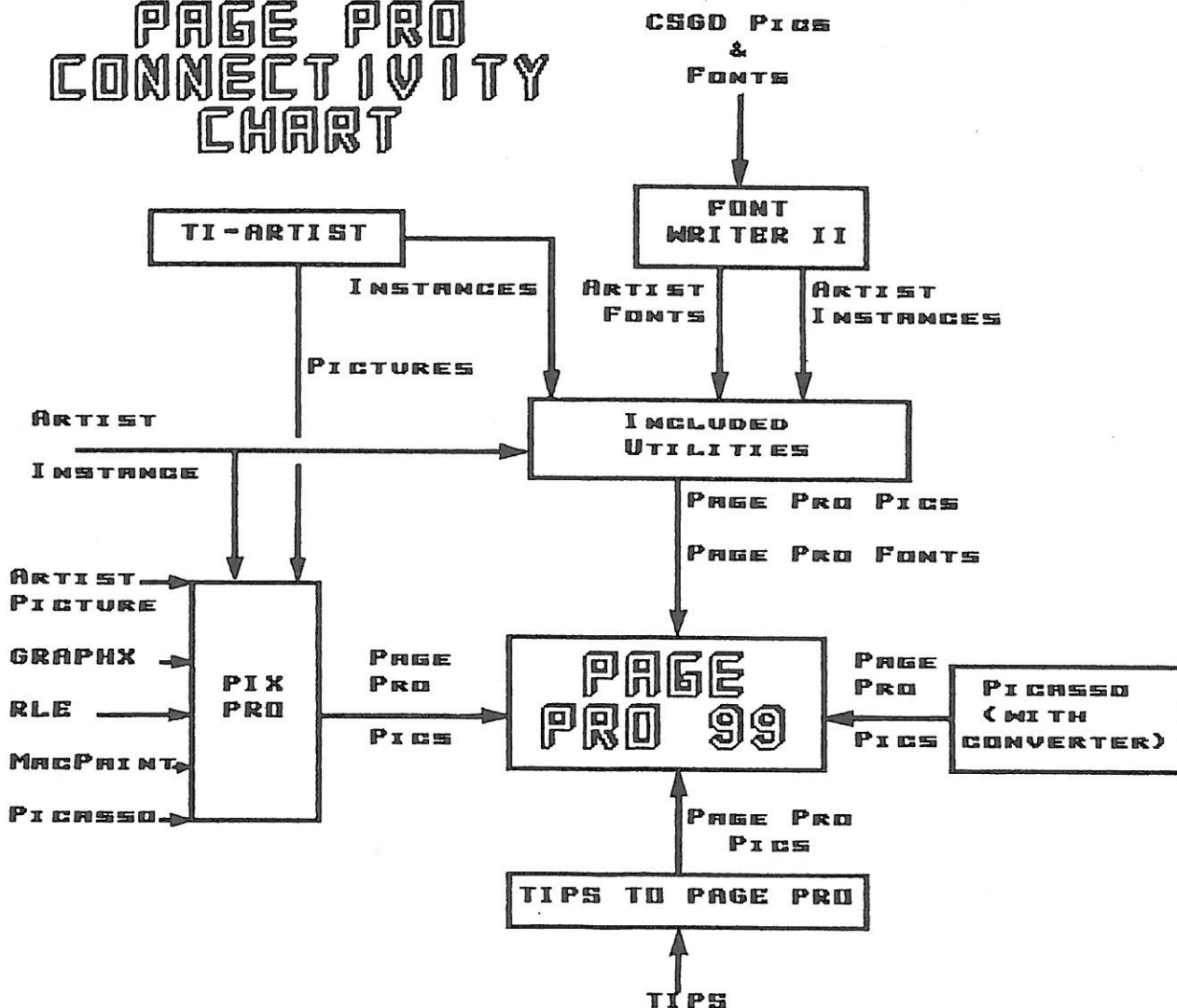
Many nice graphics programs have been written for our computer. Not everybody is aware of the fact that, even though they are not all compatible, there are many conversion programs available to enable the work of one program to be utilised by one of the others. Pix Pro is one of the best examples of that. Page Pro is one of the best graphics programs ever written for the TI-99/4A and the table below is the best that I have seen that shows how Page Pro can be used in conjunction with other graphics programs. The idea was taken from the Page Pro Times, Fall 1991, page 5 and it was redrawn with the Page Pro program. It clearly illustrates the various popular formats found on the TI-99/4A, and the utilities necessary to get them into Page Pro 99 format. I trust that it is of use to somebody.

If you do not have a copy of Page Pro, I suggest that you get one. Apart from Funnelweb & related utilities, I use this program more than any other one that I have. If you are a student or teacher or just

interested in graphics, this program is a must. I used it many times last year just helping out my kids make title pages up for school work alone. They looked great! One feature that Page Pro has that you may not be aware of is the ability to use more than one font on the same page. If you are not using a very large part of the page, then to use more than one font is relatively quick. If you have a full page then it is slow. There is virtually no limit to how many different fonts (i.e. small, large and line fonts) that can be used on the same page. Once you have used all you want of one font then use Fctn 7 to initiate the process of saving your work in 'picture' format (explained in the utilities manual). This is a way of saying that that part of the page has now been 'permanently fixed'. So, if you call up this newly-saved picture you can do more work on it with other fonts without altering what you have already done, and so the process can be repeated to your heart's content.

It is an easy utility to use and the package sold under the title Page Pro 99 has other utilities with it such as printing programs in double column format. With more people upgrading their computers to 80 columns, it is of interest that some of the Asgard utilities, such as Banner Maker, have 80 column capabilities. O

PAGE PRO CONNECTIVITY CHART



Techo Time

with Geoff Trot

RAMdisks

I have been looking at 3 RAMdisks this month for various problems. Quite often, the problems with RAMdisks that I have seen relate to bad connections due to poor soldering. There can also be problems with sockets. I have had very little time so far to look at anything so far this month but one has been fixed with one shorted connection and one wire going to the wrong place. Another is an 8K RAM version expanded to 31 RAM chips which has many memory errors. This looks like a difficult one to track down. The third RAMdisk has a problem with EPROMs being installed but it worries me because quite a few of the RAM chips have a green growth on them which would seem to indicate corrosion starting. If this is not part of the problem, it may develop into its own problem.

Hard disk back-up

I have had a major crash of my working hard disk. I realise that most of you will not be too concerned about hard disk problems, but part of what I will talk about here relates to interlace of floppy disks, which I talked about some time ago. To return to the hard disk saga, I was using a 20 Mbyte hard disk for my general use and had two RAMdisks and one 80 track floppy drive for access to the outside world as it were. It is quite painful to back up a hard disk to floppies, as I will explain shortly, so I modified a program from Col Christiansen to back up to another hard disk. Some months ago I did this when I bought another hard disk and it took an hour or so. I did this mainly to restore and tidy up the files on my working disk, so I formatted my working disk and copied all the files back on one at a time. That took much longer than the back up but gave me a hard disk which operated much faster as there was no fragmenting of files and the directory structure was all at the start of the disk.

Then my hard disk started playing up. The last thing it did before it finally became impossible was to speed up and apparently go faster than its rated speed but not recognise it. Anyway, it now seems to start OK but is no longer recognisable as a formatted disk. I am still thinking about how would be the best way to try and recover what I can from the disk. I was just about to attack its DSR to see what I could learn too! I installed the second hard disk and so was able to get at all the information that I had on the disk at the time I did the back up and for some strange reason I had copied quite a bit onto floppies recently anyway (most unusual for me) but I thought that I should save all that I had on by last hard disk before I tried to do something tricky with the bad disk and destroyed the contents of both. So this led me to do a backup of the hard disk to floppies.

Back up is a function available in the Myarc Disk Manager 5 for the hard disk. It is possible to back up the whole disk or just by individual directories. By doing individual directories I had found that if the disk had to be initialised during the back up, it took for ever to do the back up. The reason seems to be related to the interlace used on the disk. The default interlace for double density disks (used by the back up routine) is 5, which means it takes 5 revolutions to read each track (at best) or one second per track. However the back up routine requires too long to do its function than this allows so it takes 18 revolutions per track or 3.6 seconds per track. Worse than this, the back up routine goes over each disk twice, once for a verify or writing a pattern on every sector and the other writing the data. You can imagine how long it would take to do a double sided 80 track drive at 3.6 seconds per track ($2 \times 2 \times 80 \times 3.6 / 60 = 19.2$ minutes per floppy disk). So, although it is quicker to format and verify floppy disks at interlace 5, if you want to do a back up to floppies, you must work out how many floppies you will need and format them at interlace 7

before you start. Why not interlace 6? This is because the gap between tracks for interlace numbers a factor of the number of sectors per track is less than it should be (see my earlier articles) and so may take longer than interlace 7.

It took me 29 floppy disks all DSDD and 80 track to back up my hard disk (83,570 sectors). This took about 5 hours with each disk requiring $2 \times 2 \times 80 \times 7 / 5 / 60 = 7.47$ minutes at least. On top of this was the time it took to format the disks to start with, but at least this could be done in batches. Once a back up is started, it must carry on to the end as the last thing that happens is that the number of disks in the back up is written to the first disk. The advantage of a full hard disk back up is that all the directory structure is kept and will be restored. I have not tried it yet, but I think that the files will be restored unfractured, as the files are saved in one piece. I am not sure what happens to the directory structure as this is also saved sequentially. I shall let you know when I find out. O

Treasurer's Report

by Geoff Trot

This month is the time when most of our members are due to renew their subscriptions. I hope you all think that our group is still worth belonging to and that you will all renew your membership as soon as you can. We will be producing our usual top quality and informative newsletter for you and there is still a lot of excellent software being produced. If only there were enough time to enjoy it all.

Income for February	\$2210.45
Payment for February	\$368.15
Excess of income over expenses for February	\$1842.30 O

TI-Writer Replace String

by Jim Leshner, PA USA

Replace String is a very versatile time-saving function of TI-Writer. For example, if you know you are going to use a certain word several or, many times, rather than typing it out all those times, use a stand-alone letter like Q, or X, or Z, then, after completing your text, use Replace String to insert the desired word/phrase. Let's use the word "periodical". We could type Z in place of that word, and then, when your article is finished, select RS and type in the following:

/ Z / periodical /

Notice that there is a space on either side of the Z so that if you had a Zebra in your article it would not replace it. Now if you do not have a Z standing alone in your text you can press ALL at the prompt. If you do have a Z, then you can press Y or N for yes or no depending whether or not you want that letter reassigned. The TI-Writer will re-adjust everything to make the new words fit. By now many of you are way ahead of me. Yes, you can even use whole phrases. For example, you were extolling the virtues of the TI-994A EXPANSION BOX and you were going to use the phrase several times in your article. Again, any letter will do, but J is a good one as it is under your right index finger. You would type in the following to use the Replace String:

/ J / TI-994A EXPANSION BOX /

Or, suppose you had your text littered with underlined words and you decided you would rather have them double struck, the same principle applies, even for the special characters on page 146 of your TI-WRITER manual. I have also found that, in the main, it is better to use the fixed character mode while using Replace String than the word wrap mode because it does not re-format each time. You only have to be careful that you do not 'push' any characters off the end of the page. O

Struck by Lightning!

by Ross Mudie

On Tuesday, 11th February 1992, the TEXPAC BBS and some of the electronics in my home were rendered inoperative by a single, devastating direct lightning strike on the radio mast in my back yard. Analysis of the damage has provided a lot of information on things that can be done which may protect against the severity of damage in future. There are many lessons to be learned from my experience, I hope that readers will be able to relate to the advice that I offer and take steps to protect their own property from possible future damage.

Just after midday, my wife Sue saw a storm coming and went into the normal routine of unplugging the computers in the house and other "vulnerable" electronics. Eldest son Sam unplugged the power and FM aerial for his stereo system and musical system amplifiers. The BBS was normally left on since was considered to be "protected".

At 12:15, there was a deafening "explosion" overhead, Sue and Sam did not see the flash. (I guess when it is that close the body response goes into overload and not all the things happening at once are registered). Both Sue and Sam commented that their ears were ringing for a couple of hours after. The lightning struck my amateur radio tower on the top of the rotatable aerial system. The actual antenna which was struck (an experimental UHF TV antenna which youngest son Peter had built) had some of the reflector wire melted. The rotator motor below somehow survived. The resultant current to earth via the coax cable into the communications caravan and the mains wiring of the garage must have raised the earth potential in the garage and caravan by many thousands of volts. There were places where the surge flashed over from earth to active in the equipment, sending a massive surge back along the power wiring into the house. A huge current pulse must have flowed in the mains earth wiring, to the water pipe Mains Earth Neutral (MEN) point and back out into the street supply.

Just before the strike, some people playing lawn bowls at the bowling club about 200 metres away, had just gone inside the club house. They observed sparks rising into the air from the outdoor metal hat rack. (It is said that "if your hair starts standing on end when a storm is around, RUN for cover!") The people at the bowling club saw the lightning strike come down.

What was damaged?

On the tower: a TV antenna diplexer; mast head preamp; UHF preamp; communications wiring burnt off at the point of termination. It is thought that the heat from the current in the sheath of the coax cable caused the plastic tape securing the cable to melt.

In the garage adjacent to the tower: 3 disk drives, disk controller and alarm system of the BBS; a variable voltage bench power supply; building alarm power supply; one fluorescent light (both filaments, the starter and ballast); antenna rotator control system; electronic rain gauge. The colour purity of an old TV receiver in the garage was seriously affected by the magnetic pulse.

In the caravan, plugged into the garage core balance supply: A coax cable flashed over to earth of the caravan. It burnt a hole in the galvanised steel top hatch of the caravan where the flash over occurred. A battery charger and public address (which was plugged in but turned off).

In the house: a telephone; a stereo system (unplugged from mains power but connected to a 8 metre long speaker lead); the main TV front end and colour purity of its screen.

Next door: The telephone and video recorder.

The failures on the tower and items connected directly to the tower were to be expected.

The thing that surprised me was the number of items which were UNPLUGGED that had failed. ALL of these items had wires connected, capable of picking up voltage pulse if a large magnetic field had been present. The old TV in the garage was connected to a UHF preamp on the work bench, which in turn was connected to a variable power supply. Both the TV and the power supply were unplugged from the mains and antenna system. The TV suffered severe colour purity unbalance whilst the power supply regulator failed with full output voltage.

There are 7 fluorescent lights in the garage, 6 in the workshop area on two single way switches and one in the outer part of the garage which was on a 2 way switch with a fairly long length of wiring (about 8 metres). All lights were turned off at their switches. The single fluorescent light had its filaments blown open, the starter was loose in its case and believe it or not, the ballast choke was open circuit. The thing that seemed in common with some of the other failures was the length of wiring, which could pick up the magnetic pulse.

The BBS had an alarm which was running from the 12 volt power supply of the external stand alone disk drive. The wiring from the alarm went in an underground cable to the house, wide open to the magnetic pulse. The 12 volt disk drive power supply series pass transistor failed and all 3 disk drives plus the AT disk controller. The mains into the BBS and the phone line were both through a FAXGUARD, which is a surge protection device. The FAXGUARD worked because the modem and the mains side of the BBS were OK, as was the BBS phone.

The power supply of the building alarm system suffered a failed regulator due to the unclamped over voltage.

The antenna rotator control system had several flash over points from the circuit board to metal case. Earth tracks lifted off a printed circuit board in the direction indicating display, damaged diodes and capacitors in the 28 volt AC motor control and two Regulator chips. The case of the voltmeter chip which drives the Liquid Crystal Display was physically broken.

In the caravan there were three flashovers. A coaxial cable from the tower, entering the caravan via the sky-light and disconnected; flashed through the coax cable sheath to the earthed metal skylight of the caravan. This would have substantially raised the earth potential of the caravan. In a battery charger there was a flash over from earth to mains active through the side of a 3AG fuse holder. This split the fuse holder and took a small chunk out of the aluminium panel of the battery charger. In a PA amplifier which was plugged in but turned off, there was a flashover in a switch from earth to active, which also blew the active fuse. The core balance circuit breaker supplying power to the caravan was tripped. In part of the telephone communications the ring interrupter transistor was blown short circuit.

In the house a "getto blaster" stereo was disconnected from the mains, but it still failed. The antenna input was connected to a length of cable that ended in the room and one channel of the output had speaker leads longer than the other. Sure enough, the RF input and the output connected to the longer (8 metre) speaker leads failed. The magnetic field from the household wiring, which would have carried the surge under the floor of that room, is blamed for these failures.

The main TV receiver was still plugged in, but it had a mains surge filter. There was a failure in the tuner of this TV, plus colour purity problems on the screen. I guess there must have been plenty of zappies around on the external TV antenna and a big magnetic field.

continued on page 5

Tigercub Reformatter+

by Jim Peterson, Tigercub Software, USA

The following features form the software package entitled 'Reformatter':

Reformatter - CR Adder
LF Stripper - Blank Strip
Hyphenater - Dehyphenater
Justifier - Unjustifier
Marginater - Unmarginater

The TI-Writer or Funlweb Formatter can be used to reformat D/V80 text files to a greater or lesser line length, but it can garble the text while doing so, and I have seen many examples of such in newsletters.

To use the Funlweb Formatter for this purpose, the text must have carriage returns. If the ampersand, the "at" sign, the caret, the asterisk followed by two numeric digits, or the period at the beginning of a line, are present in the text, printing through the Funlweb Formatter will delete them and in some cases delete or garble the text.

When text is printed back to disk with the Formatter, it will contain large blocks of lines with nothing but a line feed, which must be manually deleted. It will also place a line feed after every line, and convert carriage returns to line feeds. These can be stripped out by printing back to disk with the C option but, contrary to the manual, they are not really stripped out - they are converted to ASCII 32 blanks, which can cause problems in some applications. Also, the carriage returns will have been stripped.

Because of all these complications, I have never been satisfied with the Formatter. Therefore I wrote this program. My program will reformat text which does not have carriage returns - such as the many files which are now being ported over from IBM - and will add the carriage returns, providing that headers are either indented or followed by blank lines and paragraphs are indented. It will strip the trailing blanks left by printing with the C option from the Funlweb Editor, and will strip line feeds caused by printing to disk from the Funlweb Formatter. It also strips the tab line.

It will of course also reformat text which does have carriage returns, to any greater or lesser length. It will automatically edit and correct hyphenation that is no longer at the end of a line due to reformatting. It will offer you the option (which Funlweb does not!) of hyphenating words. If you select this option, it will display the text and the farthest position at which a word could be hyphenated, whenever a word would be broken after two or more characters. If you elect to hyphenate, the maximum number of characters followed by a hyphen will be presented as a default.

This section features two innovations - my CALLKEY with blinking cursor (just press Y or y or N or n, no need to press Enter) and Bruce Harrison's STRACC routine. You can press Enter to accept the default, or type your own shorter string and the default string will be erased so you do not need to delete the extra characters.

An input here which is not followed by a hyphen will be rejected as a presumed error, but sometimes you will want the input to be without a hyphen, if the break is at a predetermined hyphen. In this case, just enter your input again and it will be accepted the second time. If you select the option to justify, extra blanks will be inserted between words to align the right margin, just as TI-Writer does.

The program will optionally strip blanks inserted by previous justification, and will automatically strip them before justifying. Text with a preset left margin cannot be properly reformatted, but the program will strip such margins. It will also optionally add a left margin to reformatted text.

The program checks the first 20 lines to find the left margin and the present line length. It is presumed that at least one of the first 20 lines will be a full line. You can also hyphenate and/or justify and/or add carriage returns, and/or strip blanks and line feeds, and/or add or strip margins, without otherwise reformatting, by selecting a new line length the same as the old.

The program is intended primarily for reformatting back to disk, for use with multiple-column printing programs, but it will offer the option of output to the printer, and will then let you enter printer control codes. You can even reformat to line lengths greater than 80. In this case, the printer or output file will be opened in the necessary record length.

Program listings, in any language, should never be reformatted. They will be garbled and impossible to key in correctly. This program also cannot properly reformat columnized text, text containing graphics or ornamentation, etc. I do not think any program could. This program is released to the public domain with no restrictions except that no one except myself (Tigercub Software) and non-profit user groups may charge a copying fee for it.

However, if you do find this program useful, I would be grateful if you would spend a few pennies for a postcard to tell me so. I am getting very tired of contributing programs to the TI world and never hearing a word about them again. I do not want money, just a word of thanks to encourage me to keep on writing.

And, of course, if you find any bugs please let me know.

Jim Peterson (Tigercub Software)
156 Collingwood Ave.,
Columbus OH 43213

```
100 CALL INIT :: CALL LOAD(8196,63,248):: CALL LOAD(163
76,65,32,32,32,32,255,48):: CALL LINK("A")
110 DIM I$(61),O$(250),T$(20)
120 GOTO 160
130 LL,R,H$,J$,J,M,CFLAG,M$,P$,L,Z,C$,P,X,A$,Q$,CF,IF$,
OF$,CR$,SET,K,S,U$,LM,LM$,SL$,LMS,SLM$,WO,T$( ),Y,PC,CC
,RC
140 CALL CLEAR :: CALL SCREEN :: CALL COLOR :: CALL
HCHAR
150 !@P-
160 DISPLAY AT(12,1)ERASE ALL:"Read instructions? Y/N"
:: CALL CALLKEY(12,24,"YyNn",Q$): IF Q$="N" THEN 240
170 CALL CLEAR :: CALL SCREEN(5):: FOR J=1 TO 12 :: CALL
COLOR(J,16,1):: NEXT J
180 ON ERROR 190 :: OPEN #1:"DSK1.FORM+DOCS" :: GOTO 200
190 RETURN 160
200 FOR J=1 TO 22 :: IF EOF(1)THEN 220 :: LINPUT #1:M$
:: DISPLAY AT(J,1):M$
210 NEXT J
220 DISPLAY AT(24,6):"PRESS ANY KEY" :: DISPLAY AT(24,6
):"press any key" :: CALL KEY(O,K,S):: IF S=O THEN 220
230 IF EOF(1)<>1 THEN CALL CLEAR :: GOTO 200 ELSE CLOSE
#1
240 CALL CLEAR :: CALL SCREEN(5):: FOR SET=0 TO 12 ::
CALL COLOR(SET,2,16):: NEXT SET :: CR$=CHR$(13):: ON
WARNING NEXT
250 GOSUB 890
260 DISPLAY AT(8,1):"Input filename?":"DSK" :: ACCEPT
AT(9,4)BEEP:IF$ :: ON ERROR 270 :: OPEN #1:"DSK"&IF$,
INPUT :: GOTO 280
270 RETURN 260
280 DISPLAY AT(17,1):"Output to 1":"(1) Disk":"(2) Prin
ter" :: ACCEPT AT(17,11)SIZE(-1)VALIDATE("12")BEEP:WO
290 IF WO=2 THEN DISPLAY AT(17,1):"Printer? PIO":"::"
:: ACCEPT AT(17,10)SIZE(-18):OF$ :: GOTO 310
300 DISPLAY AT(17,1):"Output filename?":"DSK":"::
ACCEPT AT(18,4)BEEP:OF$ :: F$="DSK"&OF$
310 DISPLAY AT(20,1):"Do you want to set left margin?
Y/N" :: ACCEPT AT(21,13)SIZE(1)VALIDATE("YN"):SL$ :: IF
SL$="N" THEN 330
320 DISPLAY AT(23,1):"Left margin how many spaces?" ::
ACCEPT AT(24,1)SIZE(2)VALIDATE(DIGIT):LMS :: SLM$=RPT$(
" ",LMS)
```

```

330 CALL CLEAR :: GOSUB 890
340 DISPLAY AT(10,1):"Reformat to length?" :: ACCEPT AT
(10,21)SIZE(3)VALIDATE(DIGIT)BEEP:R :: Y=MAX(80,R+LMS)
350 ON ERROR 360 :: OPEN #2:OF$,VARIABLE Y,OUTPUT :: IF
WO=1 THEN 390 ELSE 370
360 RETURN 300
370 DISPLAY AT(10,1):"How many printer codes? 0" :: ACC
EPT AT(10,25)VALIDATE(DIGIT)SIZE(-2)BEEP:PC :: DISPLAY
AT(10,1):"" :: RC=1
380 FOR J=1 TO PC :: ACCEPT AT(10,RC)VALIDATE(DIGIT)BEEP
:CC :: PRINT #2:CHR$(CC) :: RC=RC+LEN(STR$(CC))+1 ::
NEXT J
390 DISPLAY AT(12,1):"Hyphenate? Y/N" :: ACCEPT AT(12,16
)SIZE(1)VALIDATE("YN")BEEP:H$
400 DISPLAY AT(14,1):"Right justify? Y/N" :: ACCEPT AT
(14,20)SIZE(1)VALIDATE("YN")BEEP:J$ :: IF J$="Y" THEN
U$="Y" :: GOTO 420
410 DISPLAY AT(16,1):"Strip extra blanks? Y/N" :: ACCEPT
AT(16,25)SIZE(1)VALIDATE("YN")BEEP:U$
420 DISPLAY AT(18,5)ERASE ALL:"READING RECORD" ::
DISPLAY AT(20,2):"REFORMATTING RECORD" :: DISPLAY AT(22
,6):"SAVING RECORD"
430 FOR J=1 TO 20 :: LINPUT #1:T$(J):: LL=MAX(LL,LEN(T$
(J))): IF EOF(1)=1 THEN 450
440 NEXT J
450 RESTORE #1 :: FOR J=1 TO LL :: FOR K=1 TO 20 :: IF
SEG$(T$(K),J,1)<>" " THEN LM=J-1 :: J=LL :: K=20
460 NEXT K :: NEXT J
470 LINPUT #1:I$(1):: IF LM>0 THEN I$(1)=SEG$(I$(1),LM+
1,255)
480 FOR J=2 TO 61 :: IF EOF(1)THEN 530 :: LINPUT #1:I$(
J):: DISPLAY AT(18,20):J :: IF ASC(I$(J))=128 THEN 530
ELSE IF LM>0 THEN I$(J)=SEG$(I$(J),LM+1,255)
490 IF POS(I$(J-1),CR$,1)<>0 THEN 500 :: IF ASC(I$(J))=
13 OR ASC(I$(J))=32 THEN I$(J-1)=I$(J-1)&CR$
500 NEXT J :: M=J-2
510 IF (ASC(I$(61))=13 OR ASC(I$(61))=32)AND POS(I$(60
),CR$,1)=0 THEN I$(60)=I$(60)&CR$
520 IF R>LL THEN 710 ELSE 540
530 CLOSE #1 :: M=J-1 :: CFLAG=1 :: IF POS(I$(M),CR$,1)
=0 THEN I$(M)=I$(M)&CR$ :: GOTO 520 ELSE GOTO 520
540 FOR J=1 TO M :: DISPLAY AT(20,22):J :: GOSUB 900 ::
IF U$="Y" THEN CALL UNFILL(I$(J))
550 M$=P$&I$(J):: P$=""
560 CALL HSTRIP(M$)
570 L=LEN(M$)+(POS(M$,CR$,1)<>0):: IF L<R AND POS(M$,
CR$,1)<>0 THEN Z=Z+1 :: O$(Z)=M$ :: GOTO 680 ELSE IF
L<R THEN P$=M$&" " :: GOTO 680
580 IF L=R THEN Z=Z+1 :: O$(Z)=M$ :: GOTO 680
590 C$=SEG$(M$,1,R):: CALL LASTPOS(C$," ",P)
600 IF P<>0 THEN 610 ELSE Z=Z+1 :: O$(Z)=C$ :: M$=SEG$(
M$,R+1,255):: GOTO 570
610 IF R-P<3 THEN C$=SEG$(M$,1,P-1):: CALL JUSTIFY(R,C$
,J$):: M$=SEG$(M$,P+1,255):: Z=Z+1 :: O$(Z)=C$ :: GOTO
570
620 X=POS(M$," ",P+1):: IF X=0 THEN X=LEN(M$)ELSE IF X=
R+1 THEN Z=Z+1 :: O$(Z)=C$ :: M$=SEG$(M$,R+2,255)::
GOTO 570
630 IF H$="N" THEN 670
640 GOSUB 850
650 DISPLAY AT(12,1):"Hyphenate?" :: CALL CALLKEY(15,12
,"YNyn",Q$):: IF Q$="N" RQ$="n" THEN CALL
HCHAR(2,1,32,352):: GOTO 670
660 GOSUB 860 :: GOTO 570
670 GOSUB 880 :: GOTO 570
680 NEXT J
690 FOR J=1 TO Z :: DISPLAY AT(22,20):J :: PRINT #2:SLM
$&O$(J):: NEXT J :: Z=0
700 IF CFLAG=0 THEN I$(1)=I$(61):: GOTO 480 ELSE CLOSE
#2 :: STOP
710 FOR J=1 TO M :: DISPLAY AT(20,22):J :: GOSUB 900 ::
IF U$="Y" THEN CALL UNFILL(I$(J))
720 M$=P$&I$(J):: P$=""
730 CALL HSTRIP(M$)
740 IF POS(M$,CR$,1)<>0 AND LEN(M$)<=R+1 THEN Z=Z+1 ::
O$(Z)=M$ :: GOTO 840 750 IF LEN(M$)<R THEN P$=M$&" " ::
GOTO 840
760 C$=SEG$(M$,1,R):: CALL LASTPOS(C$," ",P):: IF P=0
THEN Z=Z+1 :: O$(Z)=C$ :: M$=SEG$(M$,R+1,255):: GOTO 740
770 IF P=R THEN C$=SEG$(M$,1,P-1):: CALL JUSTIFY(R,C$,
J$):: Z=Z+1 :: O$(Z)=C$ :: M$=SEG$(M$,R+1,255):: GOTO
740
780 IF R-P<3 THEN C$=SEG$(M$,1,P-1):: CALL JUSTIFY(R,C$,
J$):: Z=Z+1 :: O$(Z)=C$ :: M$=SEG$(M$,P+1,255):: GOTO
740

```

```

790 X=POS(M$," ",P+1):: IF X=0 THEN X=LEN(M$)ELSE IF X=
R+1 THEN Z=Z+1 :: O$(Z)=C$ :: M$=SEG$(M$,R+2,255):: GOTO
740
800 IF H$="N" THEN 830 :: GOSUB 850
810 DISPLAY AT(12,1):"Hyphenate?" :: CALL CALLKEY(12,12
,"YNyn",Q$):: IF Q$="N" RQ$="n" THEN CALL HCHAR(2,1,32,
352):: GOTO 830
820 GOSUB 860 :: GOTO 740
830 GOSUB 880 :: GOTO 740
840 NEXT J :: GOTO 690
850 DISPLAY AT(2,1):M$ :: DISPLAY AT(6,1):SEG$(M$,1,R):
A$=SEG$(M$,P+1,R-P-1)&"-" :: DISPLAY AT(10,1):A$&SEG$(
M$,R,X-R+1):: RETURN
860 CALL LINK("ACCSTR",14,1,LEN(A$),A$):: CF=CF+1 :: IF
POS(A$,"-",1)=0 AND CF=1 THEN 860
870 CF=0 :: C$=SEG$(C$,1,P)&A$ :: M$=SEG$(M$,P+1+LEN(A$)
-1,255):: CALL JUSTIFY(R,C$,J$):: Z=Z+1 :: O$(Z)=C$ ::
CALL HCHAR(2,1,32,416):: RETURN
880 C$=SEG$(C$,1,P-1):: CALL JUSTIFY(R,C$,J$):: Z=Z+1 ::
O$(Z)=C$ :: M$=SEG$(M$,P+1,255):: RETURN
890 DISPLAY AT(2,2):"TIGERCUB REFORMATTER+ V1.2":":":
Reformatter * Hyphenater Right Justifier * CR Adder
Unfiller * Marginater" :: RETURN
900 IF SEG$(I$(J),LEN(I$(J)),1)=" " OR SEG$(I$(J),LEN(I
$(J)),1)=CHR$(10)THEN I$(J)=SEG$(I$(J),1,LEN(I$(J))-1)
910 IF I$(J)=" " OR I$(J)=" " THEN I$(J)=CR$ :: RETURN
ELSE RETURN
920 !@P+
930 SUB HSTRIP(M$):: X=1
940 P=POS(M$,"-",X):: IF P=0 THEN SUBEXIT ELSE IF P=1
THEN 960
950 IF SEG$(M$,P-1,3)<>"-" THEN M$=SEG$(M$,1,P-1)&SEG
$(M$,P+2,255)
960 X=P+2 :: GOTO 940
970 SUBEND
980 SUB LASTPOS(A$,B$,Y):: X,Y=0
990 X=POS(A$,B$,X+1):: IF X>0 THEN Y=X :: GOTO 990
1000 SUBEND
1010 SUB JUSTIFY(R,C$,J$)
1020 IF J$="N" OR LEN(C$)=R OR C$="" THEN SUBEXIT
1030 P=1
1040 X=POS(C$," ",P):: IF X=P THEN P=P+1 :: GOTO 1040
ELSE Y,P=X :: IF POS(C$," ",P)=0 THEN SUBEXIT
1050 C$=SEG$(C$,1,X)&" "&SEG$(C$,X+1,255):: IF LEN(C$)=
R THEN SUBEXIT ELSE P=X+2
1060 X=POS(C$," ",P):: IF X=0 THEN P=Y :: GOTO 1060 ELSE
GOTO 1050
1070 SUBEND
1080 SUB CALLKEY(R,C,V$,K$)
1090 CALL HCHAR(R,C+2,30):: FOR T=1 TO 3 :: CALL KEY(O,
K,S):: IF S<>0 THEN 1120
1100 NEXT T :: CALL HCHAR(R,C+2,32):: FOR T=1 TO 3 ::
CALL KEY(O,K,S):: IF S<>0 THEN 1120
1110 NEXT T :: GOTO 1090
1120 IF POS(V$,CHR$(K),1)=0 THEN 1090 ELSE K$=CHR$(K)
1130 CALL HCHAR(R,C+2,32):: SUBEND
1140 SUB UNFILL(M$):: P=1
1150 X=POS(M$," ",P):: IF X=P THEN P=P+1 :: GOTO 1150
1160 X=POS(M$," ",P):: IF X=0 THEN SUBEXIT
1170 M$=SEG$(M$,1,X)&SEG$(M$,X+2,255):: GOTO 1160
1180 SUBEND

```

TI-Writer Formatter Tip

by Harold Hoyt, MO USA

[This article was retyped from the September, 1991 issue of the Spirit of 99.]

John Briscoe is some of that new blood that the club needed. Here is another good tip from John.

This one is a real time saver for people who use the TI-Writer Formatter in Funnelweb. You can do a disk directory while in the Editor and mark a file so that you do not have to type in the DSKn.FILENAME. This is a big help if you cannot remember the filename. If you do a disk directory while in the Formatter, apparently you cannot mark the file, so if you want to mark a file for the Input Filename, you exit the formatter, enter the Editor, do a disk directory, mark the file, exit the editor, and re-enter the Formatter. This is very clumsy and slow if you are not using a ramdisk.

continued on page 12

Games Information

by Robert Brown

Welcome to YET another one of these famous Games Information articles. Just to put you into the picture, the time is 1:15pm on Sunday 19th January (I thought you would like to know!). The reason why I tell you this is because I am waiting for a friend to ring up, so I thought I might as well write another (at least start!!) world famous GAMES INFO article.

Last month (or article), I gave you the solution to Zork I, and since then, I have had an overwhelming response to give you the solution to Zork II.

I was not going to give you the solution for some months, so you can AT LEAST have a go for YOURSELF!!, but one thing I have learnt, is never disappoint your readers, so here goes....

So, back for more, eh? Zork I was not enough... or did you get hooked? No matter, it is almost time to get started. However, first a few words of advice. The Wizard of Frobozz is a lot more troublesome than the thief ever was. This is due to his spell-casting abilities. Therefore, frequent saving of the game is recommended! Otherwise, you will be wasting a lot of time waiting for some spells to wear off.

Okay, let's do it! Get the sword and lamp, and move along South, South, South, SW to the Shallow Ford. At this point, turn on your lamp, then continue South, SE to the North End of Garden. Enter the Gazebo, and get all items on the table. Then "Exit Gazebo," and go N, NE to the Shallow Ford and fill the teapot with water.

Now, head South, SW, SW, and you will be in the Carousel Room. This is a wonderful place that spins you around, so you cannot be sure that you are going in the direction you specify. Fun, huh? The thing to do here is keep moving around until you get to the Riddle Room. That room is actually SE of the Carousel Room, but you are not likely to get there by typing SE, at least not for a while. So, every time you move, and find yourself somewhere else besides the Riddle Room, you will have to return to the Carousel Room and try again. Also, before you start moving around, drop everything but the teapot and the lamp; you will pick up the things again as you need them.

These are the directions from the various other rooms to the Carousel Room:

- Marble Hall - South
- Path Near Stream - Southwest
- Topiary - West
- Menhir Room - North
- Cobwebby Corridor - Northeast
- Cool Room - Southeast

All right, you have finally made it to the Riddle Room. Now, all you have to do is answer the riddle! Actually, it is a fairly simple riddle; just enter: Say "A Well," and you will be able to move on to the East, which is the Pearl Room. Leave the necklace for now, and continue East into the Circular Room. There is a large bucket here, just big enough for you to enter, so do that. Then pour water into the bucket, and it will rise to the top of the well.

Get out of the bucket, then go East into the Tea Room, and get all cakes except the orange one. Eat the green cake, and you will shrink down. Now, go East into the Pool Room, and throw the red cake into the pool of tears. The water will be soaked up, and you can get the package of candies. Forget about the flask; there is no use for it.

Return West to the Posts Room, and eat the blue cake. Presto! You are returned to normal size. Stretch a little, then head NW into the Low Room. Aha, there is a robot here. Tell the robot to "Go E," then go that way yourself, and you will both be in the Machine Room.

I would not be surprised if the controls to the Carousel room were here! Tell the robot to "Push Triangular." There will be a thud in the distance (something fell), and the Carousel Room is now off.

Tell the robot to go South, and do likewise. You are now in a Dingy Closet, and a red sphere sits invitingly just beyond your grasp. Try getting it, and a steel cage will come down over you (uh oh!). Not to worry, however! Once the cage is down, just tell the robot (even though you cannot see him) to "Lift Cage." Once that is done, you will be able to get the sphere.

Now, getting back to the bucket may be a problem of sorts; pushing the button has caused strange things to happen in the Low Room. So, after going North and West from the closet into the Low Room, you may have a problem when trying to go SE back to the Tea Room. Keep trying however, and you will get there sooner or later. Then, go West from the Tea Room, enter the bucket, get the water, and the bucket will descend again. Get out, drop the Teapot, head West, pick up the necklace, then West again and NW to the Carousel Room. There is a box here (the thud you heard before). Open it, and inside is a violin. You do not need to take it now, so just leave it here for the time being.

Drop the sphere, necklace and candy, then get the sword, place mat and letter opener. Move North to the Marble Hall, picking up the brick, then continue North until you come to the Ledge in Ravine. Go up to the Tiny Room, which has a locked door. Slide the mat under the door, then move the lid and insert the opener in the keyhole. Remove the opener, then pull the mat and get the key. Now, unlock and open the door and go North into the Dreary Room. Drop the key and opener, and get the blue sphere.

Now, it is South and Down to the ledge, then West and North into the Dragon Room (what would an adventure game be without a dragon?). He is not one of your nicer dragons, so be careful here. Hit him with the sword, then move South. He will follow you. Hit him a second time, and move South again. He is still following you, and you are almost where you want to be. Whack the scaly nuisance one more time, and head West into the Ice Room.

Okay, the dragon follows you in, gets silly over his reflection in the glacier, and proceeds to get himself drowned. You can drop the sword now, since you will not be needing it anymore. As his body washes away, go East and SE to the Carousel Room, then SW to the Cobwebby Room. Get the string, then go back NE and pick up the newspaper and the matches. Now move along NW, West, and West again into the Lava Room. Leave the ruby for now, and continue on Southward to the Volcano Bottom.

Aha, look what is here.... a hot air balloon! All you need is a way to get it inflated, and you just happen to have the means to do that. Get into the basket, then open the receptacle and put the newspaper inside. Light a match, then light the newspaper with the match. Make sure you keep the receptacle open! Now, just wait until the balloon rises to the Narrow Ledge, then "Land." Tie the wire to the hook and get out of the basket.

On the ledge is a gold coin. Pick that up and head South into the Library. Get the purple book, open it, get the stamp, then drop the book. You can ignore the other books; they are all worthless. Now, back North and into the basket again. Untie the wire, and wait until the balloon rises to the Wide Ledge.

Again, "Land" and tie the wire to the hook. Get out and go South. Now, it is time for a few thrills. Put the string in the brick, then put the brick in the hole in the box. Now light a match, then light the string. Now, head North IMMEDIATELY! There will be an explosion, then you can safely go back and get the crown. But do not dawdle; once you have it, get back to the basket pronto! The explosion weakened the ledge, and

it will not hold much longer. So, get in, untie the wire, close the receptacle, and wait until the balloon lands at the Volcano Bottom.

Well, you seem to be doing almost as much travelling around as in Zork I, and you are not finished yet, not by a long shot! Go North, get the ruby, then East twice and SE to (yet again!) the Carousel Room. Drop off everything but the lamp, then go NW, North, North, West, and West again, and you should now be at the entrance to the bank.

The bank is tricky, so follow directions carefully here. Go NE, then East into the Safety Depository, then South into the office and pick up the portrait, and back North again. Now, "Enter Light," and you will be in a small room. At this point "Enter South Wall," and you will be in the Depository again. "Enter Light" a second time, and you will find yourself in the vault, with a pile of bills in front of you. Get the bills, then "Enter North Wall," and once more you are in the Depository.

Now, drop the bills and portrait, and go East to the East Tellers Room. Go East again to the Depository, pick up the goodies, and "Enter Light." This time, you are in the East Viewing Room. From there, go South and you will be at the bank entrance. You have robbed the bank, and kept the alarms silent.

Okay, head along East until you reach the Dragon Room, then North to the Dragon's Lair. Leave the chest for now; instead, say "Hello Princess," and wait until she leaves. Follow her, and continue to follow her until you are both in the Gazebo, then wait. The unicorn will appear, and the princess will give you the key from around its neck, and also a rose. You can drop the rose; it has no use in the game.

Well, you are getting closer to the end of the game, but there are still a few things left to do. Exit the Gazebo, then move South, West, SW. Drop off some of the treasures, then go back NW, North, North, North to the Lair again. Open the chest and get the statuette, then return to the Carousel Room. From there, due South to the stairway. This is a good place to save the game, as you are about to enter one of the nastier mazes around.

Go down the stairs, and you will be in the Oddly-Angled Room. Things are not always as they seem here, and the exact direction to go at this point is not constant. Try East first, and if you cannot go that way, then try West. Either way, you should find a club. Get the club, and from that point, go SE, NE, NW, SW. Watch the room descriptions carefully as you do this; the little diamonds on the floor should get brighter with each move you make (you are actually "running the bases" here). If they are not doing this, restore the game and try again. When you have done it successfully, there will be a noise in the distance.

Now, you should try to go either North or South (that varies also). If one way does not work, try the other, and if neither works, try East or West. There will be a staircase going down, but you do not want to do that yet, so go Up instead, and head North until you are back in the Carousel Room.

Get the blue and red spheres and the candy. Drop a few things if you have to, but hang on to the club! Now, SW twice, and you are in the Guarded Room, with a lizard head in the doorway. Give the candy to the lizard, then unlock the door with the gold key. Open the door and go South, then West twice into the Aquarium. Throw the club at the aquarium, then get the clear sphere. Go East into the Wizard's Workroom.

Put each sphere on the stand of the same colour, then get the black sphere that appears, and go South into the Pentagram Room. Put the sphere on the circle, and the Demon will appear. You must give him ALL the treasures you have collected, as well as the gold key. This will mean at least one more trip to and from the

Carousel Room to get all the stuff. When you have given everything to the demon, tell the demon, "Give me the wand," and he will take the wand from the Wizard and give it to you.

Now, you are almost finished! Go North, East, North, North, NE, South, and you are in the Menhir Room. You need to get the Menhir out of the way for a little while, so do this: Wave the wand at the Menhir and say, "Float." The Menhir will rise up, allowing you to go SW into the kennel and get the collar. After that, go NE, then South, then Down, and Down again into the Cerberus Room.

Cerberus is no problem; just put the collar on him. Now, go East, then South, and you are in the Crypt. Turn off the lamp, and you will notice a secret door in the south wall. Open the door and go South, and you will be on the Landing. The game is over!!

Or is it? After all, There is still Zork III up ahead!!

Well, that is it for yet another month/article. If you would like me to do a review on anything, just drop me a line. (see below for more info...)

I can be contacted in the following ways...

1. By the BBS, Username: Games
Password: Forgotten!!! (Help, Ross)

For our Overseas Readers, the BBS is TEXPAC..
(That is a plug!!) Phone number : 612 456 4606
2. By Phone... (02) 743-3019 Home or 612 743 319
(02) 516-2399 Work (Until about 7pm)
612 516 2399
3. By Post... 46 Llewellyn Street
Rhodes 2138
New South Wales
A u s t r a l i a

NOW you DO NOT have any excuses for not getting in touch with me!! This article is Copyright By Robert Brown - All Rights Reserved

Just a Short Note:

(From the Author) "This article consists of over 50 sectors and 2400 words. I wrote this while I was waiting for a friend to call, or is that EX-FRIEND... because I am still waiting... the time being 3:00pm. So why do not

OTHER people get to it, and write some articles. Everyone has some spare time - well you do not have to sleep do you!!"

- Overheard one day, when the author was boasting about his writing talents (or lack of them!!). ○

continued from page 10

John's trick is to do a disk directory (Fctn 7) while still in the formatter. Arrow down to the file you want. Press the space bar, which places an invisible mark on the file. Press <Ctrl> = to return to the Formatter, then press <Fctn> D (right arrow) to place the new file name in the Formatter Input Filename. This saves the see-saw time for repeatedly loading the Formatter and Editor just to mark files.

This is super for people who are intimidated by long filenames and cannot remember if the filename was DOCS or -DOCS- or -READ-ME- or README. If you give up and use single character filenames, then a year from now when you are reviewing your disks, you will not have a clue as to what file X is for.

EDITOR'S NOTE: I tried this and discovered that you MUST have files QD and QF if you are using Funnelweb. ○

Time Check

by Loren West

This is my first attempt at programming. It filled my needs and has also added some realism to my using the computer. I have plenty of programmes, such as TI-Artist for drawing and using graphics, TI-Writer for word processing (which I am using right now), and Page Pro for doing whole pages, plus heaps of other programmes that fellow members have written and made available to other members.

This is my programme and I did it my way (I have heard that somewhere before); the idea was there but a plan of attack was needed. There were a lot of problems, but I had a lot of help from other members, who guided me in a direction that would get the job done. Thanks to all those concerned!

First, a bit about myself. I am an R.D.O. (Rostered Day Off) relief driver for a Penrith bus company; that should be Rostered Week Off, because the company likes you to save up the days until a week can be taken. The shifts are never exactly 8 hours long; some go over, some under. The week is also broken in two; on Mondays, Tuesdays and Wednesdays you do one shift, then on Thursdays and Fridays you do another. Because of my irregular shifts and pay packets I was intrigued to know if everything was as should be.

Now for the finished product... I have called it "TIMECHECK". The programme adds your hours of work for a single day only.

Most working days are split into two ie: 8.00 - 12.00, lunch, then 12.30 - 4.00. I get paid for the hours worked, although on some shifts I get a meal allowance.

Some allowances are built into the programme, eg: driver conductor allowance and before and after six allowance (ie, if we start or finish before or after 6 o'clock). These could be changed to suit your own needs.

The screen displays all that is needed, ie: total wage per day, progressive wage, hours per day and progressive hours, plus hours per shift. A hard copy can also be made, when prompted.

The first screen gives you the choice of 12 or 24 hour time. If you choose 12 hour you will be required to use am or pm. The next question is meal allowance entitlement.

The second screen asks if you want a print out, and prompts for a yes or no. If you answer yes, you must have the printer on at this stage.

The third screen is the first of two screens, to add your hours of work, and the starting and finishing times. This will be displayed on screen, you are then prompted to add another shift or not. If you choose to add another shift you will be moved to the next screen. If you do not wish to add another shift and press 'no', the programme ends. If you require a print out of only one shift, you need to go to the next screen but just use zeros.

The fourth screen is the second screen that you add your hours of work. On this screen will be displayed total pay, total hours, days pay and days hours. If you wish to add the same hours for the whole day press yes when prompted, otherwise you can go back and add another day's work onto the existing hours.

I hope that this programme is of use to somebody else as well.

```
100 ! "WORK HOURS"
110 !
120 !
130 ! by LOREN WEST.
```

```
140 !THIS PROGRAM IS TO ADD UP THE HOURS THAT YOU HAVE W
ORKED IN ONE DAY.
150 !
160 !PLUS GIVE YOU A HARD COPY OF YOUR HOURS AND PAY BEF
ORE TAX IS REMOVED.
170 !
180 !I WORK SHIFT WORK AND I COULD WORK 2 SHIFTS A DAY.
NO TWO WEEKS ARE THE SAME, WITH THIS PROGRAM I CAN CHECK
AGAINST MY PAY SLIPS.
190 !
200 !THE PROGRAM.. THERE ARE 4 SCREENS. 1st SCREEN ASKS
YOU IF YOU WANT TO USE 12 OR 24 HOUR TIME, THEN IF YOU'R
E ENTITLED TO MEAL ALLOWANCE.
210 !
220 !2nd SCREEN ASKS IF YOU WOULD LIKE A PRINT OUT.
230 !
240 !3rd SCREEN IS THE FIRST OF 2 SCREENS TO ENTER YOUR
TIME ... START AND FINISH..IF 12 HRS WERE SELECTED THEN
YOU ARE REQUIRED TO .....
250 !
260 ! SELECT"AM or PM" .
270 !
280 !4th SCREEN IS WHERE YOUENTER YOUR SECOND SHIFT OR I
F NO SECOND SHIFT ENTER 'ZEROS'.
290 !
300 !IF YOU WORK THE SAME HOURS OF THE ONE DAY DURING TH
E WEEK JUST ENTER "Y" WHEN PROMPTED BY THE SCREEN.
310 !
320 !LINE 1240 ... INSERT YOUR HOURLY WAGE HERE. IT SHOU
LD READ "T=YOUR HOURLY WAGE DIVIDED BY 60"
330 !LINE 1460 CONTAINS ALLOWENCE. CHANGE THIS TO YOUR O
WN NEEDS.
340 CALL CLEAR
350 CALL SCREEN(11)
360 DISPLAY AT(3,5):"WOULD YOU LIKE TO USE "
370 DISPLAY AT(7,5):"24 HOUR TIME
OR 12 HOUR TIME"
380 DISPLAY AT(4,5):"-----"
390 DISPLAY AT(17,5):"PLEASE ENTER 24 OR 12"
400 DISPLAY AT(20,11):">>" :: DISPLAY AT(20,16):"<<"
410 ACCEPT AT(20,14)VALIDATE("12,24")BEEP SIZE(2):TIME
420 DISPLAY AT(22,1):"ARE YOU ENTITLED TO MEAL ALLOWA
NCE Y or N" :: ACCEPT AT(23,19):MEAL$
430 IF MEAL$="Y" THEN MEAL=1.38 ELSE MEAL=0
440 CALL CLEAR
450 CALL SCREEN(7)
460 DISPLAY AT(3,2):"WOULD YOU LIKE A PRINT OUT"
470 DISPLAY AT(4,2):"-----"
480 DISPLAY AT(10,7):"PLEASE PRINT 'Y'FOR YES OR
'N' FOR NO"
490 DISPLAY AT(18,10):"_" :: DISPLAY AT(18,15):"_" ::
DISPLAY AT(20,5):"PLEASE PRESS ENTER" :: ACCEPT AT(18,13
)SIZE(1):HARDCOPY$
500 DISPLAY AT(20,5):"PLEASE PRESS ENTER NOW"
510 IF HARDCOPY$="Y" THEN 520 ELSE 530
520 OPEN #1:"PIO"
530 CALL SCREEN(13)
540 CALL COLOR(0,16,1)
550 CALL COLOR(3,16,1)
560 CALL COLOR(4,16,1)
570 CALL CLEAR
580 A=0 :: AI=0 :: GI=0 :: DE=0 :: BE=0 :: EL=0
590 GOSUB 1680
600 GOSUB 1460
610 GOSUB 1780
620 IF TIME=24 THEN 640
630 IF H=12 THEN H=0
640 IF Q$=R$ THEN 730 ELSE 650
650 IF Q$="AM" THEN 660 ELSE 670
660 O=(H+M)-1440 :: GOTO 680
670 O=(H+M)-720
680 IF R$="AM" THEN 690 ELSE 700
690 S=(FH+FM)-1440 :: GOTO 720
700 IF FH=12 THEN S=(FH+FM)-1440 :: GOTO 720
710 S=(FH+FM)-720
720 IF S<0 THEN A=O-S ELSE A=S-O :: GOTO 800
730 S=H+M
740 O=FH+FM
750 A=O-S :: GOTO 800
760 E=H
770 Z=720-(E+M)
780 X=FM
790 A=X+FM+Z
800 C=A/60
```



```

810 B=(C-INT(C))
820 D=INT(C)
830 GOSUB 1850
840 GOSUB 1410
850 DISPLAY AT(15,9):USING "##":D :: DISPLAY AT(15,19):U
SING "##":B
860 DISPLAY AT(14,9):"HOURS" :: DISPLAY AT(14,19):"MINUT
ES"
870 DISPLAY AT(23,1):"ADD ANOTHER SHIFT"
880 DISPLAY AT(24,1):"PLEASE PRESS...Y OR N"
890 CALL KEY(O,K,S)
900 IF S=0 THEN 890
910 IF K=89 THEN 1000
920 IF K=78 THEN 930
930 DISPLAY AT(23,1):"WOULD YOU LIKE TO DO ANOTHER"
940 DISPLAY AT(24,1):"PLEASE PRESS...Y OR N"
950 CALL KEY(O,K,S)
960 IF S=0 THEN 950
970 IF K=89 THEN 530
980 IF K=78 THEN STOP
990 GOTO 840
1000 CALL SCREEN(5)
1010 GOSUB 1460
1020 GOSUB 1780
1030 IF TIME=24 THEN 1050
1040 IF H=12 THEN H=0
1050 IF Q$=R$ THEN 1140 ELSE 1060
1060 IF Q$="AM" THEN 1070 ELSE 1080
1070 O=(H+M)-1440 :: GOTO 1090
1080 O=(H+M)-720
1090 IF R$="AM" THEN 1100 ELSE 1110
1100 S=(FH+FM)-1440 :: GOTO 1130
1110 IF FH=12 THEN S=(FH+FM)-1440 :: GOTO 1130
1120 S=(FH+FM)-720
1130 IF S<0 THEN AI=0-S ELSE AI=S-0 :: GOTO 1210
1140 S=H+M
1150 O=FH+FM
1160 AI=O-S :: GOTO 1210
1170 E=H
1180 Z=720-(E+M)
1190 X=H
1200 AI=X+FM+Z
1210 C=AI/60
1220 BI=(C-INT(C))
1230 DI=INT(C)
1240 GOSUB 1860
1250 GOSUB 1410
1260 GOSUB 1600
1270 DISPLAY AT(16,9):USING "##":DI :: DISPLAY AT(16,19)
:USING "##":BI
1280 DISPLAY AT(18,1):"TOTAL" :: DISPLAY AT(18,9):USING
"##":DE :: DISPLAY AT(18,19):USING "##":BE
1290 DISPLAY AT(17,9):" ---"
1300 DISPLAY AT(23,1):"ADD THE SAME HOURS" :: DISPLAY AT
(24,1):"PLEASE PRESS Y or N"
1310 CALL KEY(O,K,S):: IF S=0 THEN 1310 :: IF K=89 THEN
1320 :: IF K=78 THEN 1340
1320 GOSUB 1600
1330 GOTO 1300
1340 DISPLAY AT(23,1):"WOULD YOU LIKE TO DO ANOTHER"
1350 DISPLAY AT(24,1):"PLEASE PRESS...Y OR N"
1360 CALL KEY(O,K,S)
1370 IF S=0 THEN 1360
1380 IF K=89 THEN 530
1390 IF K=78 THEN STOP
1400 GOTO 1270
1410 T=11.36580/60
1420 G=(A+AI)*T
1430 GI=INT(G0+.5)/100
1440 DISPLAY AT(20,1):"$";GI :: DISPLAY AT(20,16):"LESS
TAX"
1450 RETURN
1460 DISPLAY AT(4,1):"START TIME  hours." :: DISPLAY AT
(4,22):"min." :: DISPLAY AT(5,1):"-----"
1470 IF TIME=24 THEN 1490
1480 DISPLAY AT(6,18):"am/pm"
1490 DISPLAY AT(8,1):"FINISH TIME  hours." :: DISPLAY AT
(9,1):"-----" :: DISPLAY AT(8,22):"min."
1500 IF TIME=24 THEN 1520
1510 DISPLAY AT(10,18):"am/pm"
1520 ACCEPT AT(4,20)SIZE(2):H :: ACCEPT AT(4,26)SIZE(2):
M
1530 IF TIME=24 THEN 1550
1540 ACCEPT AT(6,24)VALIDATE("AM,PM")BEEP SIZE(2):Q$

```

```

1550 ACCEPT AT(8,20)SIZE(2):FH :: ACCEPT AT(8,26)SIZE(2)
:FM
1560 IF TIME=24 THEN 1580
1570 ACCEPT AT(10,24)VALIDATE("AM,PM")BEEP SIZE(2):R$
1580 RETURN
1590 P=(A/60):: GOTO 1610
1600 P=((A+AI)/60)
1610 BE=(P-INT(P))
1620 DE=INT(P)
1630 TH=TH+DE :: TM=TM+BE :: ZA=ZA+GI+4.16+MEAL+EL
1640 P=TH+TM
1650 PA=P/60
1660 TM=(PA-INT(PA))
1670 TH=INT(PA)
1680 DISPLAY AT(1,1):"TOTAL $";ZA
1690 DISPLAY AT(1,20):USING "##":TH :: DISPLAY AT(1,25)
:USING "##":TM :: DISPLAY AT(2,1):"-----"
1700 IF HARDCOPY$="Y" THEN 1710 ELSE 1770
1710 A$=CHR$(27):: B$=CHR$(45):: C$=CHR$(1):: D$=CHR$(0)
1720 PRINT #1:CHR$(18);TAB(1);" B/A6.$ ";EL;TAB(1);;
" D/C.$ ";DC;TAB(1);" MEAL.$ ";MEAL
1730 PRINT #1:CHR$(27);CHR$(69);" DAY PAY $ ";A$;B$;C$;G
I;A$;B$;D$;TAB(32);"DAY HOURS ";A$;B$;C$;DE;BE;A$;B$;D$
1740 IF ZA=0 THEN DC=0.0 ELSE DC=4.16
1750 PRINT #1:A$;B$;D$;A$;CHR$(70)
1760 PRINT #1:CHR$(18):: PRINT #1:"GROSS PAY $";ZA;TAB(2
2);"TOTAL HOURS ";TH;TM
1770 RETURN
1780 IF H<6 THEN 1830
1790 IF TIME=24 THEN 1820
1800 IF R$="PM" THEN 1810 ELSE 1840
1810 IF FH>6 THEN 1830 ELSE 1840
1820 IF FH>18 THEN 1830 ELSE 1840
1830 EL=1.36
1840 RETURN
1850 IF HARDCOPY$="Y" THEN 1870 ELSE 1880
1860 IF HARDCOPY$="Y" THEN 1890 ELSE 1900
1870 PRINT #1:TAB(45);"1st shift S.";H;M;"F.";FH;FM
1880 RETURN
1890 PRINT #1:TAB(45);"2nd shift S.";H;M;"F.";FH;FM
1900 RETURN

```

continued from page 25

That is one of the remaining mysteries, if I have STRG automatically entered by putting it as an Immediate word on the screen, the way I usually do, STRG does not work right. It did at one time but not now. Manual entry somehow seems to solve it. Anyway, if you follow the prompts, you should get your entered string printed at center screen.

Now enter A\$.CENT (print CENTERed) to get it again. The word .CENT works with any string, not just A\$. Try B\$ Screen #38. To prove this is not a "put up job", enter B\$.IT, B\$ will be printed but not centered. Try STR\$(string) as follows: B\$(or any address)STR\$! "This is a String". Then try B\$.IT to verify that you in fact have that message stored in B\$. Now do B\$.CENT and, lo and behold, you will get B\$ centered on the screen.

Now do A\$.TT, select a 3 character sequence from from A\$, then use STR\$ to load the sequence into B\$. Then do A\$ B\$ 1 POS . and verify the the occurrence of B\$ is in the right place in A\$.

Now pick another character sequence from A\$, load it into B\$. Then enter A\$ B\$ 1 POS A\$ + .IT and note that you get the remaining part of A\$. Now substitute n TT for .IT, you now get a segment of only n characters. There is literally no end the flexibility. Do a little experimenting. C U next time; May the Forth be with U.

DSCAN Correction

by Ross Mudie

Bob, Credit where credit is due... In the March 1992 TND I am credited with the article on DSCAN. This is not the case, as far as I am aware the author of this article and very clever program is Jesse Slicer of Olathe KS USA. Regards...Ross Mudie.

Hollywood Hijinx part 3

by Scorpia, Copyright 1987 USA

Ok, I think you are about ready for the grand finale, as it were. Take any one of the candles from the mantelpiece, make sure you have the cloth sack and red match from the matchbox in the kitchen, then go to the Foyer for the skis.

From there go out to the Patio, where you can drop your flashlight for now (outside, there is light enough to see by from the moon), then east to the Southeast Junction, and from there north and east to the top of the beach stairs.

As you probably know, halfway down some of the steps are missing. With the skis, this is no problem at all (you always wanted to be a downhill racer, right? Here is your chance!). Wear the skis and go down the steps. Try not to be too nervous, you will make it down safely.

Once on the beach, remove and drop the skis, which have served their purpose. Looking around, you notice a green match and a small, smoldering fire. Get the match. Now, the trick here is to somehow make the matches waterproof, so you can take them through the grotto and light them afterwards.

Light the candle from the fire. Now, drip wax on the red match and the green one (and now you know why they are different colours). Once that is done, put out the candle, place it in the sack, and close the sack so the candle will stay dry. You cannot do that with the matches (or the flashlight), because you cannot open anything in the dark.

All right, head south and west to the grotto. Jump into the water (brrr!), swim down twice to the underwater passage, then west through the passage, and up again. Whew!!!

Of course, it is pitch dark down here, but you do not have to worry about any Grues (this is the REAL world, after all! grin), and you can move in the dark, even if you cannot see. So, go north twice, then up. Although you cannot tell just yet, you are in a bomb shelter.

Ok, light one of the matches (either one). You only have two moves after it is lit, so do not waste any here. Open the sack and get the candle. The first match has burned out, so light the second one, and then light the candle with it. Fortunately, the candle is a long-burning one.

Wow, what a crazy arrangement! A safe suspended in the air over a large plank, a greasy chain hanging from a closed hatch...what can you make of all this? Well careful study (and possibly a rough drawing on a piece of paper) may help to make things clearer.

First things first. Pull the chain to open the hatch. It is too far overhead to reach, but we will solve that problem right now. A close look at the plank shows that the left end (under the safe) is down. That is no good for you, so push the right end down (just like a seesaw!).

Now, what about the rope securing the safe? The knot cannot be untied, but you have the candle to burn through the rope. Do just that, burn the rope with the candle. Step on the right end of the plank and wait. Soon enough, the rope gives way, the safe falls, and you are catapulted right through the hatch and out!

Of course, that is not quite the end of things. Drop the sack, get the heavy ladder, and go back down. Put the ladder on the hooks so you can climb out later, then read the plaque on the safe.

So far, you have not seen anything that might be the combination to this safe, so it is a reasonable conclusion that the plaque holds the key to getting it open. What do we have to work with here?

The first line simply has three names: Levy, Regan, Lebling. Hmm. You know, that could be Left, Right, Left. But, what about the numbers? There are many possibilities to choose from. Still, if the names are the directions, maybe the names are the numbers, too, in disguise. Let us see...4 letters in Levy, 5 in Regan, and 7 in Lebling.

Try it: left 4, right 5, left 7. CLICK! The safe opens, and inside (as you undoubtedly suspected) is the only existing copy of Uncle Buddy's masterwork, "A Corpse Line". Take that, and go up the ladder; there is one more treasure to find, and a bit more besides!

Out of the hatch, you are on the side of a cliff. You can just manage to get up in spite of the slippery rocks, and reach the cannon emplacement. You notice that the cannon is sitting on a compartment, and the cannon itself is too heavy to move. But there is always Newton's Third Law to help out. Get a cannon ball and put it in the cannon. Light the fuse with the candle. BOOM! The cannon fires, and the recoil moves it off the compartment. Opening that, you find a catcher's mask (from that cutting horror flick "Friday the Fifteenth"), a peg, and a note.

The note congratulates you on finding all ten goodies, and invites you "down" for a final surprise. Hmm. Down where? Well, let us find out! Head back to the house, picking up the flashlight from the patio as you go (you can put out the candle now).

Head to the Foyer and upstairs to the closet. Once in the closet, put the peg in the hole. OOPS! The floor just opened below you. Ouch! You fall and hit something hard, dropping everything you had in hand. Then you go down a slide and land in the prop vault.

GASP!!!! You can hardly believe your eyes: Aunt Hildegard is here!!! And she is tied to a buzz saw!!!! And there is Herman the Horrible, standing over her!!!!

Well, do not just stand there with your teeth in your mouth! You have to save Aunt Hildy!! Grab a prop, any prop, and hit Herman with it. Since it is only a prop, it break harmlessly. The one he hits you with also breaks into hundreds of pieces. And Hildy is getting closer to that buzz saw (which is NOT a prop!) every second!!!!

So you get another prop, but that one breaks, too. Oh no! She is even closer now to an unpleasant demise!!! In desperation, you grab a third prop, and smack Herman with it. Wow! That one was for real! Herman flops to the floor, even as Aunt Hildy's graying head reaches the buzz saw!!!

With no time to lose, you rush over and untie her!! WHEW!!! You were just in time!! (it does not matter what props you use..it will always be the third one that does the job). Herman slithers out of the room and down the chute while Aunt Hildegard explains everything to you, and tells you that tomorrow, she will see her lawyers and make out a real will, leaving everything to you.

Wow, what a night this has been! You may not be a millionaire (yet), but I bet you could sell this plot to the movies...it sounds just the sort of film Uncle Buddy would have loved!!!

Hollywood Hijinx is copyrighted 1987 by Infocom Inc.

This walkthru is copyrighted 1987 by Scorpia, all rights reserved. 0

Renew Now!

Talking about Modems

by Robert Brown

(And you thought I could only talk about games!)

Since the Board of Directors of TISHUG have decided not to charge for access to the BBS, I thought I would write an article on how modems work, and explain to the non BBS members the idea of a BBS.

MODEM is an abbreviation for MODulator/DEModulator. A modem is necessary when communicating between computers using a phone line. The phone system is analog (using tones of different frequencies) and computer output is digital (0 or 1, ON or OFF). The modem's job is to convert (modulate), the computer's digital signal to analog and demodulate the analog to digital at the receiving end. Consequently, two modems are needed for data exchange over a phone line between computers.

The Hayes Smart Modem 300 was first introduced in May of 1981 and quickly became the industry standard, with a data transfer rate of 300 bits per second (bps). Larger numbered modems indicate a higher transfer speed; e.g., the Hayes Smart Modem 2400 transfers data at 2400 bps. A modem's speed is measured in bits per second, although this is commonly inaccurately referred to as "baud" rate.

Modems can be either internal (on a card inside your computer) or external (a separate box with visible lights to indicate when it is on and operating).

Data can be sent either SERIAL or PARALLEL. Serial means one bit at a time. Parallel sends one character (8 bits) at a time. Modems are serial devices.

Amplifiers are used to boost the signal when transmitting long distances on analog lines, but this results in increased noise levels. If the signal were digital, regenerators would be used instead of amplifiers, and less noise and higher accuracy would result. These benefits, plus greater speed and capacity, are what make fiber optic lines so attractive. The phone industry is headed towards digital transmission, so modems may someday be a thing of the past.

When a signal is sent, it is either synchronous or asynchronous. Asynchronous is the most common. It always has a start bit (0) and one or more stop bits (1). Synchronous sends a 128k packet of information and is used in communicating with mainframes (but not all mainframes are synchronous). Modems can be both asynchronous and synchronous devices. A typical asynchronous signal would look like the following:

start	/	data	/	parity	/	stop
0	/	#####	/	1	/	1

PARITY refers to error checking. There are three choices for error checking: ODD, EVEN, or NONE. NONE means no error checking will be done. This results in a faster transfer rate, but low reliability. EVEN means a 1 will be placed in position 9, if needed, to keep the total number of occurrences of (1s) even. ODD will use the same position to make the sum of ones an odd number. (If you are running a BBS, set your machine to NONE. This will accept a caller using ODD, EVEN, or NONE. Otherwise, they will have to use the same parity that the BBS modem is using.)

PROTOCOL is the term used for "method of communication". We as humans in Australia have a protocol of English: one person talks then the next talks, etc. The most popular, standard protocol is XMODEM, which involves the following:

SOH BLK#1 BLK#1 DATA CKSUM

The SOH character (start of heading), followed by the block numbers, signals the start of one 128 byte

block transfer. The block number and its complement assure accuracy and proper sequencing. Data can be any size up to 128 bytes (or characters). The CKSUM, known as check-sum, uses the modulo sum of the ASCII values of each character in the data field. If the checksum received equals the checksum transmitted, an acknowledgement (ACK) is sent back from the receiving computer. If a NAK is sent, (negative acknowledgement), then an error was detected and the same block of information is re-sent. After all blocks are sent, information is re-sent. After all blocks are sent,

XMODEM CRC works the same way as XMODEM, except the CRC (Cyclic redundancy check) algorithm, is a more sophisticated error checking scheme. KERMIT is fast and uses full duplex (half duplex transmits one direction at a time, full duplex transmits both directions at the same time). It sends a whole stream of data before stopping to see if there was an error. A more recent protocol, YMODEM, uses XMODEM, CRC checking, and variable packet sizes.

There are also other protocols ranging from ASCII SeaLink and even protocols with batch modes, so several files can be sent right after each other. The BBS at TISHUG does not use any of these protocols for sending files or programs to its members. A simple reason is that to download a program using XModem, you must run a program such as Fast Term or 4A Talk, and to run these you must have 32K memory installed. When the BBS was first thought about, a big percentage (me included) only had the basic TI with a cassette player, and therefore no 32K expansion.

Later on, 32K expansions were available in the console, so you could run programs such as Fast Term, but the advantage of downloading files in a certain manner (explain in a minute) would enable a larger user base and would let anyone with a console and just a modem to logon to the BBS and down and upload files and programs.

The BBS uses its own protocol, one which comes with the computer. Just as you save a file as "SAVE DSK1.FILENAME" or as "SAVE CSI". The BBS does the same, but it saves it as "SAVE RS232", which allows us the download the program over the modem, which is plugged into the RS232 port. So if the BBS is saving you a program, to receive it, you must load it. To do this you type "OLD RS232", just if you were loading a program from the RS232 port, which in fact, you are.

To upload a program to the BBS (ie send it a program) you do the exact opposite. You type in SAVE RS232 and the BBS does an OLD RS232.

To actually logon to the BBS, you will need some software. This can be TEII (on module) or Fast Term, 4A Talk or something similar. Once on the BBS, you can look at about 40 News file, which range from message to everyone, information on Games, TI Diagnostics, Amateur radio operators, lists of computer groups and BBS around Australia and other informative information.

You can also leave mail (ie messages to any other members of the BBS.) You can also download or upload some programs. Both the News files and programs to download are changed every 2 month, so you have lots of time to read them all.

The BBS is very easy to use, with a main menu and lots of sub menus, so you can not get lost, and of course there is on-line help.

So if you have a modem, why not give the BBS a call, but do not forget, you will need a Username and Password first, so ring Ross Mudie the Systems Operator (ie SYSOP) on (02) 456-2122 and arrange a username and password.

If you do not have a modem, drop me a line, and I will tell you where the best place to buy one for at the best price. The BBS will accept Baud Rates from 300

continued on page 5


```

910 FOR J=1 TO 80 STEP 5 :: D3$=SEG$(E$,J,4):: X$=SEG$(
(E$,J,1):: Y$=SEG$(E$,J+1,3):: DC=Z :: GOSUB 5000 :: D1$
=HX$ :: DC=Y :: GOSUB 5000 :: D2$=HX$
911 HEX$=D3$ :: GOSUB 6000 :: D3=DEC :: DL$=" "
915 FOR M=1 TO LB
916 IF LZ(M)=Z THEN GOSUB 8660 :: GOTO 918
917 NEXT M
918 FOR M=1 TO LB
919 IF D3=LA(M)THEN GOSUB 8640 :: GOTO 940
920 NEXT M
921 HEX$=Y$ :: GOSUB 6000 :: DEC=DEC+1
922 INPUT #3,REC DEC:YY$ :: D5$=YY$
930 IF D3=0 THEN GOSUB 8670
935 IF D3=65535 THEN GOSUB 8700 :: GOTO 970
940 GOSUB 1000
950 Z=Z+1 :: Y=Z*2
960 NEXT J
965 IF EOF(1)THEN GOSUB 8700 :: GOTO 970
966 NEXT I
970 GOSUB 7000
980 REM
990 CLOSE #1 :: CLOSE #2 :: CLOSE #3 :: END
1000 FOR K=0 TO 15
1010 IF X$=C$(K)THEN D4$=B$(K):: GOSUB 3100 :: GOTO 1030
1020 NEXT K
1030 RETURN
2990 ! PRINTOUT ROUTINE: HEADING.
3000 CALL CLEAR
3010 L=17 :: P=1 :: PRINT #2:TAB(20);"PLC Program
List." :: PRINT #2:TAB(20);"Program by BETA"
:: PRINT #2 :: PRINT #2 :: PRINT #2
3020 PRINT #2:TAB(20);P$(12)
3030 FOR I=1 TO 10 :: PRINT #2:TAB(10);P$(I);TAB(26);
R$(I):: NEXT I
3040 RETURN
3100 PRINT #2:TAB(10);D1$;TAB(16);D2$;TAB(22);D3$;
TAB(27);DL$;TAB(32);D4$;TAB(37);D5$
3110 L=L+1
3120 IF L=56 THEN PRINT #2 :: PRINT #2 :: PRINT #2 ::
PRINT #2 :: PRINT #2 :: PRINT #2 :: PRINT #2 :: PRINT #2
:: PRINT #2 :: GOTO 3130 ELSE 3160
3130 P=P+1 :: L=1 :: DISPLAY AT(10,1)BEEP:"PLEASE ATTEND
TO THE PRINTER(new page needed!)" :: DISPLAY AT
(14,1);"PUSH A KEY TO CONTINUE."
3135 DISPLAY AT(16,6);"next page is: ";P
3140 FOR BEN=1 TO 500 :: NEXT BEN ! CALL KEY(O,RV,SV)::
IF SV<>1 THEN 3140
3150 PRINT #2:TAB(15);R$(3);TAB(35);"page";P;TAB(46);R
$(1):: PRINT #2 :: PRINT #2 :: CALL CLEAR :: GOSUB 3200
3160 RETURN
3200 PRINT #2 :: PRINT #2:TAB(10);"Zeile";TAB(16);"
Byte";TAB(22);"Code";TAB(31); "OP C.";TAB(38);"OP'D
Pos./Item" :: PRINT #2
3210 RETURN
4990 REM
5000 REM DECIMAL TO HEX CONVERSION
5010 V=DC
5020 HX$=""
5030 X=INT(V/256)
5040 B=V-256*X
5050 HX$=HV$(B)&HX$
5060 IF X=0 THEN 5090
5070 V=X
5080 GOTO 5030
5090 IF SEG$(HX$,1,1)<>"0" THEN 5110
5100 HX$=SEG$(HX$,2,255)
5110 IF LEN(HX$)=1 THEN HX$="000"&HX$
5120 IF LEN(HX$)=2 THEN HX$="00"&HX$
5130 IF LEN(HX$)=3 THEN HX$="0"&HX$
5140 RETURN
5150 REM
5160 DATA 00,01,02,03,04,05,06,07
5170 DATA 08,09,0A,0B,0C,0D,0E,0F
5180 DATA 10,11,12,13,14,15,16,17
5190 DATA 18,19,1A,1B,1C,1D,1E,1F
5200 DATA 20,21,22,23,24,25,26,27
5210 DATA 28,29,2A,2B,2C,2D,2E,2F
5220 DATA 30,31,32,33,34,35,36,37
5230 DATA 38,39,3A,3B,3C,3D,3E,3F
5240 DATA 40,41,42,43,44,45,46,47
5250 DATA 48,49,4A,4B,4C,4D,4E,4F
5260 DATA 50,51,52,53,54,55,56,57
5270 DATA 58,59,5A,5B,5C,5D,5E,5F
5280 DATA 60,61,62,63,64,65,66,67

```

```

5290 DATA 68,69,6A,6B,6C,6D,6E,6F
5300 DATA 70,71,72,73,74,75,76,77
5310 DATA 78,79,7A,7B,7C,7D,7E,7F
5320 DATA 80,81,82,83,84,85,86,87
5330 DATA 88,89,8A,8B,8C,8D,8E,8F
5340 DATA 90,91,92,93,94,95,96,97
5350 DATA 98,99,9A,9B,9C,9D,9E,9F
5360 DATA A0,A1,A2,A3,A4,A5,A6,A7
5370 DATA A8,A9,AA,AB,AC,AD,AE,AF
5380 DATA B0,B1,B2,B3,B4,B5,B6,B7
5390 DATA B8,B9,BA,BB,BC,BD,BE,BF
5400 DATA C0,C1,C2,C3,C4,C5,C6,C7
5410 DATA C8,C9,CA,CB,CC,CD,CE,CF
5420 DATA D0,D1,D2,D3,D4,D5,D6,D7
5430 DATA D8,D9,DA,DB,DC,DD,DE,DF
5440 DATA E0,E1,E2,E3,E4,E5,E6,E7
5450 DATA E8,E9,EA,EB,EC,ED,EE,EF
5460 DATA F0,F1,F2,F3,F4,F5,F6,F7
5470 DATA F8,F9,FA,FB,FC,FD,FE,FF
6000 REM HEX TO DECIMAL CONVERSION
6010 DEC=0
6020 LE=LEN(HEX$)-1
6030 FOR N=1 TO LEN(HEX$)
6040 C=ASC(SEG$(HEX$,N,1))
6050 C=C-48
6060 IF C<10 THEN 6080
6070 C=C-7
6080 IF (C<0)+(C>15)THEN 6130
6090 DEC=DEC+C*(16 LE)
6100 LE=LE-1
6110 NEXT N
6120 RETURN
6130 PRINT : "ILLEGAL CHAR(";SEG$(HEX$,N,1);") IN":
"HEX VALUE ";HEX$
6140 REM
6150 RETURN
7000 REM DEFINITION TABLES PRINT-OUT ROUTINE.
7010 L=1 :: P=1 :: DE$="E = 8 BIT EINGANG" :: UN$=
RPT$("=",17):: GOSUB 8000
7020 FOR I=1 TO 96 :: INPUT#3,REC I:DF$ :: DC=I-1 ::
GOSUB 5000 :: GOSUB 8500 :: L=L+1
7030 IF L=56 THEN GOSUB 8600
7040 NEXT I
7050 GOSUB 8600
7060 DE$="A = 8 BIT AUSGANG" :: UN$=RPT$("=",17)::
GOSUB 8000
7070 FOR I=513 TO 576 :: INPUT #3,REC I:DF$ :: DC=I-1
:: GOSUB 5000 :: GOSUB 8500 :: L=L+1
7080 IF L=56 THEN PRINT #2 :: PRINT #2 :: PRINT #2 ::
PRINT #2 :: PRINT #2 :: PRINT #2 :: PRINT #2 :: PRINT
#2 :: PRINT #2 :: PRINT #2 :: GOSUB 8600
7090 NEXT I
7100 GOSUB 8600
7110 DE$="T = 8 BIT TIMER" :: UN$=RPT$("=",15):: GOSUB
8000
7120 FOR I=3073 TO 3088 :: INPUT #3,REC I:DF$ ::
DC=I-1 :: GOSUB 5000 :: GOSUB 500 :: L=L+1
7130 IF L=56 THEN GOSUB 8600
7140 NEXT I
7150 GOSUB 8600
7160 DE$="M = MERKER" :: UN$=RPT$("=",10):: GOSUB 8000
7170 FOR I=3841 TO 4095 :: INPUT #3,REC I:DF$
7180 IF DF$="" THEN 7210
7190 DC=I-1 :: GOSUB 5000 :: GOSUB 8500 :: L=L+1
7200 IF L=56 THEN GOSUB 8600
7210 NEXT I
7220 GOSUB 8600
7230 DE$="LABELS" :: UN$=RPT$("=",6):: GOSUB 8000
7240 FOR I=1 TO LB
7250 PRINT #2:TAB(1);"L ";I;TAB(14);"=" ;LA$(I)
7260 NEXT I
7270 REM GOSUB 8600
7280 RETURN
8000 PRINT #2:TAB(10);R$(3);TAB(32);"DEFINITIONEN";TAB
(68);"SEITE";P :: PRINT #2 :: PRINT #2
8010 PRINT #2:TAB(10);DE$ :: PRINT #2:TAB(10);UN$ ::
PRINT #2
8020 PRINT #2:TAB(17);"ADR";TAB(23);"ORT";TAB(27);
"BEZEICHNUNG" :: PRINT #2 :: =L+7 :: RETURN
8230 GOSUB 8000 :: RETURN
8500 OP$=SEG$(DF$,1,4):: AD$=SEG$(HX$,2,3):: OB$=SEG$(
DF$,9,40)
8510 PRINT #2:TAB(10);OP$;" = ";AD$;TAB(23);OB$ ::
RETURN

```

```

8600 P=P+1 :: L=1 :: DISPLAY AT(10,1)BEEP:"PLEASE ATTEND
TO THE PRINTER(new page needed!)" :: DISPLAY AT
(14,1):"PUSH A KEY TO CONTINUE."
8610 DISPLAY AT(16,6):"next page is: ";P
8620 CALL KEY(0,RV,SV):: IF SV<>1 THEN 8620
8630 RETURN
8640 REM LABEL MANIPULATING SUBROUTINE
8650 D5$="L "&STR$(M):: RETURN
8660 DL$="L "&STR$(M):: RETURN
8670 D5$="O 0" :: RETURN
8700 DISPLAY AT(10,1)BEEP:"End of the program listing;
Put new page in the printer,push a key when ready."
:: GOSUB 8620 :: RETURN
8800 REM

```

PGM. EPROM DECODING PROGRAM

PROGRAM FLOW CHART

main	sub	
!!save reminder		
10 DIM		
30-60 PGM.NAME		
entry of file NAMES		
100-110		
OPEN FILES		
120-180		
DISPLAY and ENTRY		
of Header info.		
190-220		
PLC op code		
mnemonic data		
assignment.		
850	3000-3040	Printout routine for header information
open Printer	3200-3210	Printout routine for column headings
860-866		
input of label information and conversion of HEX number to decimal	6000-6120	Hex to decimal conversion
870		
Read data of Dec to Hex conversion table		
900-966		
Input Eprom file Rec. assign to E\$		
decode each Hex word using nested For-Next loop to convert to Hex resp.to decimal	5000-5140	Decimal number to Hex conversion
911	5000-5140	(see above)
915-920	6000-6120	(see above)
test for occurrence of any label	8660	Label manipulating sub-routine
921-935	8640	Label manipulating sub-routine
match the definition record to the Hex code	6000-6120	(see above)
940	8670	"NOP"-code definition routine
Mnemonic decoding	8700	code <FFFF=dec65535=End of program
950	1000-1020	Mnemonic assignment routine
line number and eprom address count incremented	3100-3160	Program line printout routine with line and page counter, automatic or manual paper feed option, printing of page headers
965 End of file error trap	8700	End of program routine
970		
Definition tables	7000-7280	Definition Table printout sub-routine using other sub-routines
Printout routine	8000-8020	page headers
	8500-8510	Line printout routine
	8600-8630	New page routine
990 close files, end		

Date 22.7.91 Name: INP-TEST No. BU/913.5 P.11
PGM. EPROM DECODING PROGRAM

LOG OF VARIABLES

Var.	Description	Locations in program
As(110)	EPROM CODE EXPRESSED IN ASCII	10 900
F\$(13)	Header information questions	10 130 140 160 170 3020 3030
R\$(12)	Header info. Answers	10 180 850 3030 3150
B\$(15)	Single bit processor op.INSTRUCT	10 190 200 1010
C\$(15)	HEX string variables 0-F	10 210 220 1010
HV\$(255)	HEX value data table	10 870 5050
D\$	DISK FILE NAME:EPROM DATA	60 100
Z	Line number counter	900 910 950
Y	BYTE number counter	900 910 950
I	Loop counter/control variable	900 960 3030 860
J	Loop counter/control variable	910 960
K	Loop counter/control variable	1000 1020
M	Loop counter/control variable	915
N	Loop counter/control variable	6030 6110

LOG OF VARIABLES

Var.	Description	Locations in program
BEN	Loop counter - PAUSE	3140
D1\$	HEX line number string var.	910 3100
D2\$	HEX Byte number string var.	910 3100
D3\$	HEX EPROM CODE string var.	910 3100 916
X\$	OP CODE IDENTIFIER	910 1010
Y\$	OPERAND/POS./ITEM IDENTIFIER	910 921
DC	Line resp. Byte number transfer Var.	910 7190
HX\$	Hex string of line resp. Byte No.	910 5020 5050 5090 5100 5110-5130
D4\$	OP CODE MNEMONICS string var.	1010 3100
L	Printout line counter	3010 3110 3120 3130 7010 7190
P	Printout Page counter	3010 3130 335 3150
D5\$	Operand/POS./item.string var.	3100 916 922 7010
RV	Key- Return var.	3140
SV	Key- Status var.	3140
V	Hex conversion transfer var.	5010 5030 5040 5070
X	Hex conversion manipulator	5030 5040 5060 5070
B	Hex conversion manipulator	5040
LB	Label numbers var.	860 915 7240
LA\$(10)	Label code string variable	865 866 7250
DEC	DECIMAL NUMBER converted from Hex	6010 6090
LE	DEC conversion manipulator	6020 6100
C	DEC conversion manipulator	6040 6050 6060 6070 6080 6090
DL\$	PRINTOUT string for label	911 3100
HEX\$	HEX to Dec conversion: HEX number	6020 6030
LA(1)	Label code string converted to Decimal	866 919
LZ(1)	LABEL CODE IDENTIFIER conv.to DEC	866 916
E\$	Currently decoded EPROM CODE	900 910
D3	EPROM CODE converted to Decimal	900 910 911 919 930 935
YY\$	Current Definition Record (string)	922
DFI\$	DRIVE NO & DEFINITION FILE NAME	60
DES	DEFIN NAME STRING variable	7010 7060 7110 7160 7230
DF\$	DEFIN RECORDS	7020 7070 7120 7170 7180
UN\$	Underline string	7010 7060 7160 7030 8010

File :#1 EPROM File D/F80
File :#2 PIO (printer)
File :#3 Definition

Prime Number Generator

by C. Mac, Hunter Valley

Here is a program for finding prime numbers that works. It runs in Extended Basic, is a bit slow finding the larger numbers but is OK for finding numbers equal to or less than, say 50. The program works by taking the range of numbers that you give it via the prompts and then looks through them systematically finding and throwing out any divisible by 2. After it has done this it repeats this for 3, 4, 5 etc until it reaches the end of the list, the list being the last number that you provided. The numbers that are left are the primes. For the non-mathematician, a prime number is one that cannot be divided by any number other than 1 or itself. Eratosthenes you may have guessed is a Greek Mathematician of antiquity (ca. 230 B.C.) who, in addition to this 'sieve' that bears his name, did many formidable feats such as calculating the Earth's diameter and his value was very close to the correct measurement.

EDITOR'S NOTE - if you are checking for primes of all numbers up to N then you only need to check the divisors up to N/2 if it is an even number or (N-1)/2 if it is an odd number. So, you could probably speed the program up a bit by specifying to the computer that it only check numbers up to N/2 or (N-1)/2.

When the program runs it asks you for the number "starting at" which is where you want it to begin and then it asks you for the number "going to" which is the last number you are interested in. These two numbers supply the range of numbers for the computer to check.

```

100 ! SIEVE OF ERATOSTHENES
110 ! C. MAC
120 ! JUNE 1985
130 ! FOR H'V 99ERS NEWS
140 CALL CLEAR
150 PRINT "SIEVE OF ERATOSTHENES-"
160 PRINT "FOR FINDING PRIME NUMBERS"
170 INPUT "STARTING AT ? ":Q
180 INPUT "GOING TO ? ":A
190 DIM B(1000)
200 FOR S=2 TO A
210 B(S)=S
220 NEXT S
230 FOR K=2 TO A
240 FOR T=K TO A
250 IF B(T)/K=1 THEN 270
260 IF B(T)/K=INT(B(T)/K)THEN B(T)=0

```

continued on page 27

Horizon Utilities

from the New Horizons User Group, USA

This disk contains several utility routines written in TMS 9900 Assembly Language. The minimum system requirements are a Texas Instruments 99/4A Home Computer, a monitor or television set, 32K memory expansion, a disk based system with at least one disk drive and the Extended BASIC Command Module.

The utility routines included on this diskette are as follows:

QUICK SORT, SCREEN SAVE, SCREEN RECALL, VDP PEEK, VDP POKE, LINE SCROLLING and TEXT JUSTIFY.

QUICK SORT: by David R. Romer and John Clulow

This utility does a full ASCII sort on any string array. You should make sure to use OPTION BASE 1. To load this utility, use: CALL LOAD("DSK1.SORT"). In order to link to this utility, with your data stored in an array, use: CALL LINK("SORT",A\$(),A), where A\$() is the name of your array and A is the number of items to be sorted or the highest array element used).

SAVE SCREEN: by Ken Hopkins and John Clulow

This utility allows the entire graphics portion of VDP RAM to be saved on disk in a couple of seconds. The saving of screens on disk, means that your programs can generate graphics much faster than ordinary BASIC instructions while at the same time drastically reducing associated program memory requirements. In order to load this utility, use: CALL LOAD("DSK1.DSRLNK","DSK1.SAVSCR","DSK1.RECSCR"). In order to link to this utility, use: CALL LINK("SAVSCR","DSK1.NAME",SC), where DSK1.NAME is the output file name and SC is the screen color to be used with the graphics screen.

RECALL SCREEN: by Ken Hopkins and John Clulow

This utility allows the entire graphics portion of VDP RAM to be recalled from disk in a couple of seconds. The recalling of screens from disk, means that your programs can rapidly recreate the original graphics display complete with color, character patterns, sprite motions, sprite shapes and positions. Once any screen is loaded from disk, it may be treated as it had been generated by the program. In order to load this utility, use: CALL LOAD("DSK1.DSRLNK","DSK1.SAVSCR","DSK1.RECSCR"). In order to link to this utility, use: CALL LINK("RECSCR","DSK1.NAME") where DSK1.NAME is the name of the file you used when you saved a screen using the SCREEN SAVE utility. A screen takes about three seconds to load from disk.

VDP PEEK: by John Clulow

This utility allows you to access VDP RAM within an Extended BASIC program. Now you can directly modify the character pattern table, the screen image table, the color pattern table, the sprite motion table, the sprite pattern table and other tables. This utility differs from those of the Editor/Assembler or Mini Memory in that data is transferred via a string of bytes. The byte value 65 would be represented as CHR\$(65) and byte value 235 would be represented as CHR\$(235). The maximum length of a string is 255 bytes, which allows for rapid transfer of data. In order to load this utility, use: CALL LOAD("DSK1.VDP"). To link to this utility use: CALL LINK("VPEEK",A,B,A\$), where A is the starting VDP RAM location to be used in reading data, B is the number of bytes to be read and A\$ is the return string for the VPEEK utility.

VDP POKE: by John Clulow

This utility allows you to access VDP RAM within an Extended BASIC program. Now you can directly modify the character pattern table, the screen image table, the

color pattern table, the sprite motion table, the sprite pattern table and other tables. This utility differs from those of the Editor/Assembler or Mini Memory in that data is transferred via a string of bytes. The byte value 65 would be represented as CHR\$(65) and byte value 235 would be represented as CHR\$(235). The maximum length of a string is 255 bytes, which allows for rapid transfer of data. In order to load this utility, use: CALL LOAD("DSK1.VDP"). To link to this utility, use: CALL LINK("VPOKE",A,A\$), where A is the starting VDP RAM location to be used in writing data and A\$ is the data string for the VPOKE utility.

LINE SCROLLING: by John Clulow

This utility allows you to scroll any string horizontally across any screen line providing the string length is not greater than 255 characters. To load this utility, use: CALL LOAD("DSK1.SCROLL"). To link to this utility, use: CALL LINK("SCROLL",A\$,A), where A\$ may be any valid string expression, not an array, and A may be any numeric expression which when evaluated is a whole number between 1 and 24.

TEXT JUSTIFY: by John Clulow

This utility allows you to either right or left justify a string expression for 28 columns. The string will be left justified only if the first character in the string is a "+". The string will be right justified only if the last character is a "+". However, if neither the first or the last character is a "+", then the string will be both right and left justified. The justified, 28 character string is returned in the same string variable used to pass the string to the utility. To load this utility, use: CALL LOAD("DSK1.JUSTFY"). To link to this utility, use: CALL LINK("JUSTFY",A\$), where A\$ may be any valid string expression, except an array.

Note: These utilities were made available to us by the New Horizons User Group. O

Jenny's Younger Set

I am glad to have received that letter in the mail from our faithful contributor, Vincent Maker. Vincent had to make quite an effort this month as he wrote from the hospital and since he did not have his computer, he wrote his article on paper and sent it to me in care of the Editor. Now, if somebody in the hospital can get a program off to me, then why not some of the rest of you? We hope here from TISHUG that you get well quickly Vincent, and by the time that this magazine reaches you that you are fully recovered. Here is Vincent's letter and program:

Dear Jenny,

Here is a tutorial on how to write a guessing game. First, think of a number such as 2 (see line 100), then use a letter such as B (see line 110) to stand for your guess. When you have guessed the correct answer the computer will tell you so and you may go back and play the game again. Just type in 'RUN' to have another 'go'.

```
100 A=2
115 CALL CLEAR
110 INPUT "WHAT IS THE MYSTERY NUMBER?":B
120 IF A=B THEN PRINT "RIGHT" :: GOTO 140
130 IF A<B OR A>B THEN PRINT "WRONG" :: GOTO 115
140 END
```

To make the game a total mystery change line 100 to read:

```
100 A=INT(10*RND)
```

Maybe some of you young people can write a similar program and make it a bit longer. Vincent Maker O

Programming Music part 1

by Jim Peterson, Tigercub Software, USA

A while ago, I wrote an article about music programming in which I said that it was easy but that you almost had to know how to read music. Well, it is still easy to program, but no longer necessary to know how to read it.

Personally, I am about like the country fiddler who admitted that he could read music a little, but not enough to hurt his playing. I know just a little about reading music but that has been all I needed to know to program more than 50 songs. And, if you have ever heard my Tigercub Country or Tigercub Gospel disks, you will know that I have programmed those songs in a wide variety of styles.

Now, I have put together a few little routines to enable anyone to program music on the TI-99/4A very easily, and in many ways. You DO NOT need to know how to program and you DO NOT need to know how to read music!

First, key in this one-liner and save it as DSK1.SCALE,MERGE

```
100 DIM N(36):: F=110 :: FOR J=1 TO 36 :: N(J)=INT(F*1.0
59463094 (J-1)+.5):: NEXT J :: N(0)=40000
```

Next, NEW to clear memory and then key in this music program, which we will use as an example to experiment with.

```
110 T=2 :: A=13 :: GOSUB 1000 :: T=1 :: A=18 :: GOSUB 10
00 :: GOSUB 1000 :: T=3 :: GOSUB 1000
120 T=1 :: A=20 :: GOSUB 1000 :: A=22 :: GOSUB 1000 :: A
=23 :: GOSUB 1000 :: T=2 :: A=27 :: GOSUB 1000 ::
T=4 :: A=25 :: GOSUB 1000
130 T=1 :: A=30 :: GOSUB 1000 :: A=29 :: GOSUB 1000 :: T
=5 :: A=27 :: GOSUB 1000 140 T=1 :: A=25 :: GOSUB
1000 :: A=27 :: GOSUB 1000 :: A =25 :: GOSUB 1000
:: A=22 :: GOSUB 1000 :: T=5 :: A=25 :: GOSUB 1000
:: T=2 :: GOSUB 1000
150 T=1 :: A=27 :: GOSUB 1000 :: GOSUB 1000 :: T=3 :: GO
SUB 1000 :: T=1 :: A=22 :: GOSUB 1000
160 A=25 :: GOSUB 1000 :: A=22 :: GOSUB 1000 :: T=2 :: A
=20 :: GOSUB 1000 :: T=4 :: A=18 :: GOSUB 1000
170 T=1 :: GOSUB 1000 :: A=20 :: GOSUB 1000 :: T=5 :: A=
22 :: GOSUB 1000 :: T=1 :: A=18 :: GOSUB 1000
180 A=22 :: GOSUB 1000 :: A=27 :: GOSUB 1000 :: T=6 :: A
=25 :: GOSUB 1000 :: T=1 :: A=18 :: GOSUB
1000 :: A=20 :: GOSUB 1000
190 T=6 :: A=22 :: GOSUB 1000 :: T=2 :: A=18 :: GOSUB 10
00 :: A=20 :: GOSUB 1000 :: T=4 :: A=18 :: GOSUB
1000 :: STOP
```

Save that by SAVE DSK1.SHEN just so you do not lose it, but keep it in memory, and enter MERGE DSK1.SCALE to get that one-liner back in.

The music you just keyed in is in one voice without harmony. Let's see what you can do with just one voice. Put in a line 105 D=200 and another line -

```
1000 CALL SOUND(T*D,N(A),0) :: RETURN
```

Enter RUN, wait a second, and listen. If you did not make any mistakes in keying in the music, you should hear a fairly pleasant single-note rendition of a beautiful old folk song.

Maybe you would prefer a higher key? Here is the neat part about starting with that formula in line 100 - besides the fact that it lets you key in frequencies in shorthand. To change key, just change that 110 in line 100 to a higher frequency number. They are listed in the "blue book" that came with your computer, but if you lost it they go upward 110, 117, 123, 131, 139, 147, 156, 165, 175, 185, 196, 208, 220. You can also lower the key, providing you do not cause the lowest note in your music to go below frequency 110. In the piece you

keyed in, the lowest note number used was 13 so you could go down 12 steps. The frequencies are not in the book, but they go 110, 104, 98, 92, 87, 82, 78, 73, 69, 65.

Want the music faster or slower? Just change the 200 in line 105.

Now let's see what else we can do with single-note music. Try this -

```
1000 CALL SOUND(T*D,N(A),0,N(A)*1.01,0):: RETURN
```

Has a richer sound, does it not? How about this?

```
1000 CALL SOUND(T*D,N(A),0,N(A)/2,0):: RETURN
```

Or combine the two -

```
1000 CALL SOUND(T*D,N(A),0,N(A)*1.01,0,N(A)/2,0)
:: RETURN
```

Multiplying a note by 1.01 in another voice will always give a more resonant sound, and dividing a note by two (providing its note number is not less than 13) will always be in harmony - so will multiplying by two, or by four.

How about some real deep down bass music? The TI's tone generators can only go down to frequency 110, but the noise generator can be tuned far below that. The timber of the sound is different and does not blend too well with the tones, so use it with caution - but it is great for a tuba solo. Try this -

```
1000 CALL SOUND(T*D,N(0),30,N(0),30,N(A)*3.75,30,-4,0)::
RETURN
```

Want to go deeper? Try changing the 3.75 to 1.875 - too deep to even be musical, is it not? Maybe you could improve it by raising the frequency in line 100.

Try changing the 3.75 to 7.5 - not bad, is it? So try doubling it again to 15 - oops! When you go that high you get some very sour notes! So, go back to 7.5 and change one of those N(0) to N(A) and change the 30 following it to 0. Pretty good, so try also changing the other N(0) to N(A)*1.01 and the 30 after it to 0.

If any of those effects sound like something you might want to try in a piece of music someday, clear the memory with NEW, key it in and save it with SAVE DSK....,MERGE using a different filename for each one. Then, after you have keyed in some music, you can very quickly merge in different routines and try them. You will find that different ones go better with different songs.

The routines we have been trying all play music with a very strong beat. For a smoother effect, try this -

```
1000 FOR J=1 TO T :: CALL SOUND(-2999,N(A),0):: GOSUB
1100 :: NEXT J :: RETURN
1100 FOR D=1 TO 99 :: NEXT D :: RETURN
```

You will notice one thing right away; with this method, a series of the same note gets run together into one long note. Later we will look at ways to get around that.

To change the tempo of the music, just change the value of 99 in line 1100. Try this method in combination with the effects we tried previously.

Here is another one that gives a very nice effect -

```
1000 FOR J=1 TO T :: CALL SOUND(-999,N(A),0):: GOSUB 110
0 :: CALL SOUND(-999,N(A)*1.01,0):: GOSUB 1100 :: NEXT J
:: RETURN
1100 FOR D=1 TO 8 :: NEXT D :: RETURN
```

Or for a more mournful sound -

continued on page 5

Sorting part 4

by Ron Brubaker, USA

The previous work has introduced the basic concepts of sorting through tracing the evolution of the bubble sort and its variations for sorting numeric or string data with or without the use of pointers. The bubble sort was chosen for its simplicity. It will be recalled that, while the larger numbers fell to the bottom, the smaller numbers "bubbled" to the top (although the direction may be reversed with ease). Although the former behaviour led to the most efficient version of this sorting routine, little was done to increase the rate at which the smaller numbers moved to the top. This behaviour is a logical consequence of the manner in which the elements to be compared were chosen. Since adjacent elements were compared, movement was limited to exchanges between adjacent positions. Numerous algorithms have been developed to improve upon this approach in the quest for greater speed.

THE SHELL SORT

The shell sort is one of the most successful variations of the bubble sort. In many respects it is similar to the bubble sort. The primary difference lies in the use of a larger interval between the numbers to be compared. In the following program lines 150 and 160 determine the size of the interval to be utilised. The formula on line 150 will give a value that is one less than the largest multiple of two that is less than the length of the list to be sorted. For example, if $N = 15$, then since 2 cubed is 8, $L = 7$. Line 160 then sets the initial value of L to the integer of $L/2$ (i.e. 3 in the case above). It is not important to understand exactly how this works but only that this method of determining the interval has been found to produce the fastest results for this type of sort.

```

10 REM **** GENERATION OF A LIST OF RANDOM NUMBERS ****
20 REM
30 DIM A(100)
40 PRINT "HOW MANY NUMBERS DO YOU WANT?"
50 INPUT N
60 PRINT
70 FOR I=1 TO N
80 A(I)=INT(RND*100)+1
90 PRINT A(I)
100 NEXT I
110 PRINT
120 REM
130 REM ***** SHELL SORT ROUTINE *****
140 REM
150 L=(2*INT(LOG(N)/LOG(2)))-1
160 L=INT(L/2)
170 IF L<1 THEN 320
180 FOR J=1 TO L
190 FOR I=J+L TO N STEP L
200 K=I
210 T=A(I)
220 IF A(I-L)<=T THEN 260
230 A(I)=A(I-L)
240 I=I-L
250 IF I>L THEN 220
260 A(I)=T
270 I=K
280 NEXT I
290 NEXT J
300 GOTO 160
310 REM
320 REM *** ROUTINE TO PRINT SORTED LIST OF NUMBERS ***
330 REM
340 PRINT
350 FOR I=1 TO N
360 PRINT A(I);
370 NEXT I
380 END

```

On the first pass through an array of 15 values, J has a value of 1 and I varies from 4 to N , in steps of 3. Examine the following output which shows some of the immediate results produced.

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

a) 64 48 2 79 36 5 66 71 100 24 14 67 57 13 34
b) 64 48 2 79 36 5 79 71 100 24 14 67 57 13 34
c) 64 48 2 66 36 5 79 71 100 24 14 67 57 13 34
d) 64 48 2 66 36 5 79 71 100 79 14 67 57 13 34
e) 64 48 2 66 36 5 66 71 100 79 14 67 57 13 34
f) 24 48 2 64 36 5 66 71 100 79 14 67 57 13 34
g) 24 48 2 64 36 5 66 71 100 79 14 67 79 13 34
h) 24 48 2 64 36 5 66 71 100 66 14 67 79 13 34
i) 24 48 2 64 36 5 64 71 100 66 14 67 79 13 34
j) 24 48 2 57 36 5 64 71 100 66 14 67 79 13 34
k) 24 13 2 57 14 5 64 36 100 66 48 67 79 71 34
l) 24 13 2 57 14 5 64 36 34 66 48 67 79 71 100
m) 2 5 13 14 24 34 36 48 57 64 66 67 71 79 100

The first line shows the original unsorted numbers. In line (a) the fourth value, 79, is compared to the first value, 64. Since these are in order, no change is made. In line (b), the seventh value, 66, is compared to the fourth value, 79. In this case, the numbers are out of order so the fourth value, 79, is written in the seventh position and in line (c) the number, 66, is written to the fourth position, completing the swap. When the tenth value is compared to the seventh position in line (d) the larger value, 79, is placed in the tenth position. On successive lines the value, 24 is compared to the fourth value, 66, which is moved in line (e) to the seventh position, and to the value, 64, which is moved in line (f) to the fourth position. Finally, the value, 24, is written to the first position in line (f). Thus, each time a smaller element is found it is carried toward the beginning of the list until it fits into place. This behaviour can be traced in lines (g) through (j) as the number 57 is moved from the thirteenth position to the fourth.

This entire process is next repeated for every third value starting with the second value (i.e. $J = 2$). In this case only the final version is shown on line (k). Then the process is repeated for every third value starting with the third value ($J = 3$). The final result is shown in line (l). Note that the array is still not sorted. At this point the program goes back to line 160 and reduces the interval to the integer of one-half the previous value or in this case, one. The final result is shown in line (m).

Although all of this sounds complicated compared to the simple bubble sort, the bottom line is that this sorting routine is nearly ten (that is TEN!) times as fast as the bubble sort. The secret lies in the fact that less comparisons are required and less swaps are made. Why? Essentially because swaps are made over greater distances.

Next month we will see how this routine is amendable to variations shown in the previous issues which will include the use of a pointer. ○

Renew Now!

Assembly Class for 1992

by Ross Mudie

The assembly class topic for April 1992 will be USING DSRLNK. All TISHUG members are very welcome, the class will start at 10am, 4th April at Ryde Infants school. A new room is being used upstairs at the northern end of the building. All members planning to attend this class should read through pages 262 and 291 to 304 of the Editor Assembler manual before attending the class. It will be essential for class members to bring their own Editor Assembler books and note pad to this class. I will not be available to conduct the assembler class in May and it is likely that further work on using DSRLNK will continue into the June class. ○

Beginning Forth - part 14

by Earl Raguse, UGOC, CA USA

STRING HANDLING WORDS

Forth has very few special string handling words. However, as you saw last time, we do not really need them. However, for the XB aficionados we will make some. They can be convenient, as you will see. Also, Forth does not have special data types. A string is really no different from a number, both are a series of digits. The Forth interpreter assumes all input is a string, if after a search, it cannot find a text word in any vocabulary, it tries to convert it into a number using the current base. Failing that, it prints the text on the screen followed by a "?".

Two words which are useful for strings are ." (dot quote) and !" (store quote). The first (".) causes the following characters, numbers or text, to be printed, up to the first (") that is encountered. The word (!") plays by essentially the same rules, except that the string is stored at the designated address.

The only other word in Forth which appears to have been designed for strings is WORD, but it in fact does not care if the intercepted string is text or numbers. WORD does compute the length of a "string" of characters and inserts it as the first byte in the string. This is referred to as a "counted string". The end of a string is determined by encountering a designated character (like enter). I recommend reading the TIFM page 65 for an explanation of WORD.

Two simple things we often wish to do is find the length of an uncounted string, (not all strings in Forth are counted by WORD), and to center a string on the screen or page. XBASIC does have a length command, LEN(A\$), but no centering command; however, most of us know how to do it anyway. Forth has neither, but almost everybody knows how to do it, by the techniques used above if no other way. Just in case you do not, I have included Screen #38 and #39 which have both GLEN (Get LENgth) and CENTER, also POS and SEG\$, along with a dumb little demo to allow you to exercise them.

The first time I wrote GLEN (GetLENgth), it had a bug in it. It did not work right if the string length was zero. I do not know why you would want to get the length of a null string, but to cover all bases, I have revised it as shown on Screen #38, along with the revised CENTER which used it. Also on Screen #38 is SEG\$, and POS, just like XB. Screen #39 has a few words to exercise these words ranging from TT thru STRG. Screen #38 initializes two VARIABLES A\$ and B\$ of max 66 characters each. B\$ is further loaded with the string EARL.

I am also introducing a couple of useful new words +UNDER and NIP, rather popular in the Forth community, but unheard of when TI was writing Forth. It turned out that I did not need NIP, but I left it there anyway. The word +UNDER is often used when you have a couple of values on the stack which will become the limits of a DO loop. +UNDER lets you easily add 1 to the second value from the top, (the limit) so the loop will execute the desired number times.

When these words are all finished, they look simple enough, except for POS maybe, but I had one heck of a time getting them so that they all worked the way they are supposed to. I spent 2 days trouble shooting code that had worked at one time. I finally suspected the equipment, and tried it on my other system, and presto, it worked! Two whole days wasted. I am still not sure what the problem is (or was), it is now gone. Oh! the joys of computerizing!

GLEN works only for strings which have been INITIALized with 64 (min) BLANKS. If there is junk in the ALLOTted space, GLEN will find the string longer than you think it is. It works by starting 64 spaces past the beginning address of the of the string (in this

case the variable A\$), then checking each character to see if it is a non-blank, when found, the loop is left with LEAVE, and the index is subtracted from the initialized string length of 64.

Note the sequence DO DUP I - C@ 32 - IF I SWAP LEAVE THEN LOOP DROP. I had previously put SWAP after LOOP, as SWAP DROP, to swap and drop the remaining address on the stack which was put there by IF, (note that this is NIP). The trouble with that is if the string had no length there would be no I to SWAP with.

CENTER is a rather simple word, which uses GLEN. Once the LENgth is known, we merely subtract it from the screen width, and divide by two, then SPACES (TAB in XB) that amount to the right.

SEG\$(str1 pos len -- str2 len) works just like Extended Basic. It expects a string address, a position in that string and the length of the desired segment. A new address is computed, using the offset from the beginning of the string, for the new string. The length is that specified. +UNDER does this nicely.

POS(str1 str2 len -- al len) also works very much like XB. It expects the addresses of two strings on the stack, and the number of characters (len) of the second string to search for in the first string. POS outputs the position where the match is found, and the specified length, so that the result may be used with TYPE, (or TT), or CMOVE.

POS uses CSTR\$ (Compare STRING\$). Its rather complex, so I will not go into detail. If you want detail talk to me about it. On Screen #39, we have an abbreviation TT for the "hate to type crowd" and the word STR\$. STR\$ initializes any address, and permits one to enter a string with the words !" string". The word MOVIT uses GLEN and is handy when one does not know the string length for CMOVE.

The word .IT (printIT), uses GLEN and TT to type out a string from any address. I will discuss .SEG (print SEGment and .CENT (print CENTERed) subsequently. After Screens #38 and #39 are loaded, you will get the prompt to enter STRG.

```
SCR #38
0 \ STRING OPERATORS EGR 2/28/89
1 FG IT : IT ; : NIP SWAP DROP ;
2 O VARIABLE A$ 64 ALLOT O VARIABLE B$ 64 ALLOT
3 O VARIABLE LA O VARIABLE LB O VARIABLE $POS
4 : INIT 66 BLANKS ; A$ INIT B$ INIT B$ !" EARL"
5 : 2DUP OVER OVER ; : +UNDER ROT + SWAP ;
6 : GLEN ( a -- u) 64 + 65 O DO DUP I - C@ BL -
7 IF I SWAP LEAVE THEN LOOP DROP 65 SWAP - ;
8 : CENTER ( a --) GLEN 40 SWAP - 2/ SPACES ;
9 : SEG$ ( al pos len -- a2 len) SWAP 1- +UNDER ;
10 : POS ( al a2 u1 -- u2) \ Str$1 Str$2 begin -- pos
11 >R 2DUP GLEN LB ! GLEN LA ! O $POS ! ( al a2)
12 SWAP DUP R> + SWAP LA @ LB - + 1+ ( a2 a3 a4)
13 SWAP DO ( a2) DUP I SWAP LB @ CSTR$ 1 $POS +!
14 IF LEAVE THEN LOOP DROP $POS @ ;
15 -->
```

```
SCR #39
0 \ STRINGS STUFF EGR 2/1/89
1 : TT -TRAILING TYPE ; : STR$ DUP INIT ;
2 : GSTR ( n-- ) CLS 5 12 AT ." INPUT "
3 ." A STRING UP TO " DUP ." CHAR"
4 1 16 AT 1+ GSTR$ ; : LOC CLS 10 4 AT ;
5 : .SWFE LOC ." SHORT WAIT FOR EFFECT"
6 1 WAIT 13 6 AT CLS ;
7 : .STMT .SWFE ." AND HERE IT IS !" CR ;
8 : MOVIT ( al a2 --) OVER GLEN CMOVE ;
9 : .IT ( a --) DUP GLEN TT ;
10 : .SEG ( al begin len --) SEG$ TT ;
11 : .CENT ( al --) DUP DUP CR CENTER GLEN TT ;
12 : STRG CLS A$ INIT 40 GSTR PAD A$
13 MOVIT .STMT A$ .CENT QUIT ;
14 12 12 AT CLS ." ENTER 'STRG'"
15
```

continued on page 16

Newsletter update

by Bob Relyea

TIBUG (Brisbane), November, 1991: Editorial; 'Bits and Pieces' by Col Christensen, a good hardware article; Computer Basics, From Digits to Digital by Garry Christensen; What's New (an article on recent software); Image and Print-Using by Col Christensen; Tips from the Tigercub #35; Trading Post; Module Library; Extended Basic, 7-Games-IN-1 to type in; Games by Steve Burns.

ATICC (Adelaide), November, 1991: Coordinator's Message; Editorial; Dip's Tips by David Perkovic; Hints 'n' Helps; various articles from overseas and from our magazine.

Tit BITS (Western Australia), August, 1991: Hear Ye! (an exhortation to keep going with the TI); Club Services; A Printer Utility; various reprints from overseas; Kids' Stuff.

October, 1991: Editorial; Page Pro 99, a brief review; Tips - TI Artist - Page Pro Graphics by Geoff Warner; various reprints from overseas; Committee News; 3.5 inch disk drives; The Brother M - 1209 Printer; Notes on Alpinar.

LA TOPICS, October, 1991: Thoughts from the President ; Crackerbarrel by Chick De Marti; Funnelweb 4.0 Menu Flow Chart; XBasic Miscellany by Earl Raguse; Simple Programs by Jim Peterson; TIM = TI-Image Maker by Christopher Pratt; User Review; Recommended Monitor Guide and List; Club Meetings.

TI FOCUS, November, 1991: News and Views; Editor's Notes; The President's Two Cents Worth; SWAP OS with Horizontal Ramdisk; Fast Extended Basic by Lucie Dorais; You don't Have to have it all! by Jim Peterson; This and That, Q Base; The Joy of Ramdisks by Ted Peterson; TI Tricks & DV80Append; What is Fuzzy Logic?.

THE PUG PERIPHERAL, October, 1991: Club News by Garry Taylor; Tips #58; Modems, Part 2 by Al Kinney; Programs that Write Programs by Jim Peterson; From the Librarian by Sue Harper.

November, 1991: Club News by Garry Taylor; Repairlog (thirty four items mentioned! by Jan Janowski; From the Librarian by Sue Harper; Letter to the Editor (from Tony McGovern); Items for Sale; The Kiddie Corner by Sue Harper.

UGOC ROM, November, 1991: The Member Ship by Stan Corbin; Getting Acquainted by Gene Bohot; Keyboard ASCII Codes by Earl Raguse; TipLabel By Earl Raguse; XB Miscellany #7 by Earl Raguse; Making a Display Fixed File by Earl Raguse; President's Message by Siles Bazerman; Dips Chips and Sips by Siles Bazerman.

THE BOSTON COMPUTER SOCIETY: Listen by Justin Dowling; A huge article on Funnelweb V 4.4 described by Charles Good; Create Your Own Cursor by Bob Turner.

Tidbits, November, 1991: President's Bit by Gary Cox; In the News; Chicago Fair Report; A Visit With The Tigercub; From the Teachers Desk; Editorial Licence; Editor's Bit.

LEHIGH 99'ER COMPUTER GROUP, August, 1991 (very small issue): New Software Available; Progress Report on the Accelerator Chip; The Cryptographer's Aide; The Animator by Brad Snyder.

October, 1991: Presidents's Message; My Quick and Dirty Calendar by John Geisinger; TI-Writer Tips; How to Remove the Cover from the PE Box; TI Express. Configuring Funnelweb the Quick and Dirty Way; Extended Basic Tips.

November, 1991: TI Oddities - The Ultimate; Hardware Hacker at Heart; The Home Computer by Jim Peterson.

VAST NEWS (Compiled and edited by Jim Ely from various sources including MICROpendium Magazine), October, 1991: New Software from Asgard & Harrison; Secretary's Slate; Member Profile; From the Editor's Desk.

Talking Book

by Daniel Harris

It is a while since I sent an article upline, in fact probably not since I got most of this new equipment centre into place. I have first of all a simple program to type across-

```

100 REM TITLE ***TALKING BOOK
200 REM THURSDAY 27/02/92
300 REM DANIEL N. HARRIS
400 REM THIS PROGRAM RUNS IN
500 REM *****
600 REM *****
700 REM **CONSOLE BASIC**
800 REM **USING TERMINAL **
900 REM **EMULATOR II**
1000 REM ***DISK OR CASSETTE**
1010 REM ***CAN BE USED FOR **
1020 REM **THE FILES TO BE **
1030 REM **READ*****
1040 REM *****
2000 REM ***CONCATENATION LOOP
2010 CALL KEY(O,K,S)
2020 IF S=0 THEN 2010
2030 A$=A$&STR$(K)
2040 IF LEN(A$)=32 THEN 5000
2050 IF LEN(A$)>31 THEN 5000
2060 REM**ONE OR THE OTHER!
2070 GO TO 2010
2080 REM **THERE ARE NICER LOOPS THAN
2090 REM**THIS FOR 30-CHARACTER STRINGS
3000 REM**WHICH I CAN RECALL WRITING MYSELF.
5000 PRINT "TAPE OR DISK"
6000 INPUT "T/D":R$
7000 IF R$="T" THEN 8000 ELSE 10000
8000 OPEN #1:"CSI",FIXED,INTERNAL
9000 GO TO 20000
10000 OPEN #1:"DSK1",SEQUENTIAL,UPDATE,VARIABLE 30
20000 PRINT #1:A$
30000 CLOSE #1

```

NOTES: Since the start of the program has scrolled up the screen, I can only say that from here the control passes to the concatenation loop. There are nice ways to build such loops that use HCHAR to display the string as it builds on the screen and enable back-arrow editing. I think I have shown such loops in two earlier programs published, such as the all-purpose program with drawing and typing. For the next part of the program the tape or disk file just constructed can be read either by a separate or by a module program thusly-

```

100 REM THURSDAY MORNING SPECIAL IN
200 REM 27TH FEBRUARY 1992
300 REM D.N.HARRIS
400 OPEN #1:DEV$(AS YOU USED TO DO THE FILE)
500 INPUT #1:A$
600 OPEN #2:"SPEECH",OUTPUT
700 PRINT #2:A$
800 CLOSE #2
900 GO TO 500

```

NOTE: This is very rough programming but I trust everybody knows how to write in console Basic. The idea of the program is to enable you to set up a lot of notes on tape or disk and then be able to listen to them through the speech synthesizer. The computer can only hold about 12K, but using external files to run say a whole 90 minute microcassette side of files to speech could be the basis of computer-aided learning. I am probably over my time allocation for this log-on so I should sign off.

Those of you who built an analog joystick will have by now scored 150,000 or better with Munchman. What a difference to get the wrist impact out of the game! I am considering giving disaster forecasts also getting somebody to type a program that enables you to make your own forecasts, this would need a little coverage of the Unified Field Theory and its practical exponents, Biefeld and Brown, and mention Professor Varotsos at Athens University. One has to remind users that BASIC

continued on page 14

6 7 10,12 18,5 23,5

Regional Group Reports

Meeting Summary For APRIL

Banana Coast	12/04/92	Sawtell
Central Coast	11/04/92	Saratoga
Glebe	09/04/92	Glebe
Hunter Valley	11/04/92?	Boolaroo
Illawarra	13/04/92	Keiraville
Liverpool	10/04/92	
Northern Suburbs	23/04/92	
Sutherland	17/04/92	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 meetings are held regularly at the Boolaroo Ambulance Station. All welcome. Please contact Geoff Phillips on (049) 428 176 for details.

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, at the home of Geoff & Heather Trott, 20 Robsons Road, Keiraville. A variety of activities accompany our meetings, including Word Processing, Spreadsheets and hardware repairs. Contact Lou Amadio on (042) 28 4906 for more information.

LIVERPOOL Regional Group

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

NORTHERN SUBURBS Regional Group

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

SUTHERLAND REGIONAL REPORT

The February meeting saw a change of venue, with Derek Wilkinson providing the hospitality at his Gynea home. I took a much needed holiday down the south coast and visited Narooma, Tathra and even got as far as Mallacoota in Victoria. No computing for at least two weeks. Apparently most of the meeting was devoted to social chatter but eventually some time was spent in reviewing the various graphics programmes available for the T.I. Good to see that everyone enjoyed themselves. 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.

This is the first BUY, SWAP and SELL day of 1992. So bring along any gear that you have to sell or swap and perhaps look around for a bargain for yourself. There should be games set up for anybody interested. Come along for a relaxing time of fellowship to take in any new ideas in the TI world.

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

May	12 April
June	10 May

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. Longer articles should be to hand well before the above dates to ensure there is time to edit them.

TISHUG Meetings for Sydney

April

The first buy, swap and sell day. Bring along all your surplus equipment and swap it for some-one else's, or bring you wallet and snap up all the bargains.

May

The first all day tutorial session. Your chance to learn about using software, writing programs or understanding hardware. We can provide anything that you want but you must tell us what you would like and at what level you would like it.

June

New software and hardware to be demonstrated. Watch this space for more details.

July

The second buy, swap and sell day. This one is on the first Saturday of the school holidays but plan to take the day off and see what is about.

August

New software and hardware to be demonstrated. Watch this space for more details.

September

The second all day tutorial session. Your chance to learn about using software, writing programs or understanding hardware. We can provide anything that you want but you must tell us what you would like and at what level you would like it.

October

The third buy, swap and sell day. This one is in the middle of the school holidays but plan to take the day off and see what is about.

November

The TI-Faire will be a few weeks after this meeting so it may be taken up with the organizational requirements of this big day. New software and hardware to be demonstrated. Watch this space for more details. Time to think about nominating for positions on the board. I am sure there will be some vacancies this year!

December

The Annual General Meeting followed by some festive eats and drinks. There will probably be a bit of celebration after the TI-Faire, if we are all still friendly after the event. Make sure that you attend and give your support to all the workers in the club. O

continued from page 21

```
270 NEXT T
280 NEXT K
290 FOR U=Q TO A
300 IF B(U)<>0 THEN PRINT B(U)
310 NEXT U
```

O