

(042) 8861
8802



NEWS DIGEST

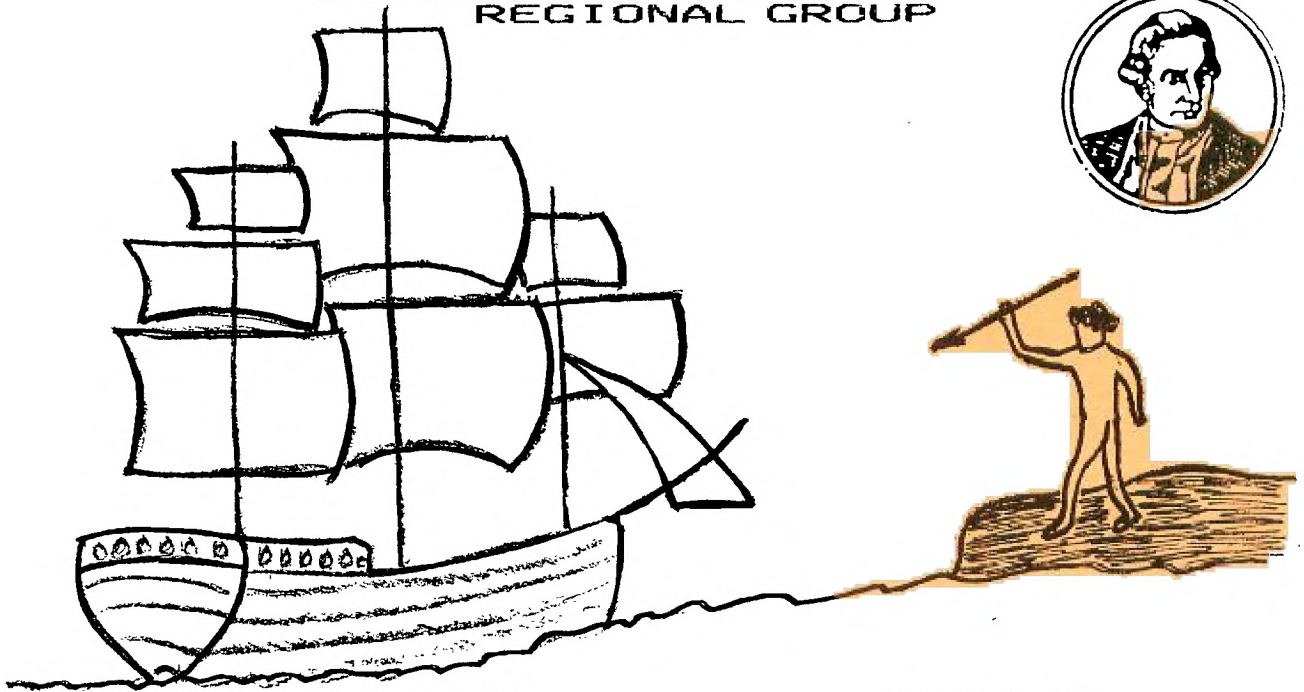
Focusing on the TI-99/4A Home Computer

Volume 7, Number 1

January/February 1988

Registered by Australia Post - Publication No. NBH5933

SUTHERLAND SHIRE
REGIONAL GROUP



BOTANY BAY

KURNELL

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

\$ 2



TISHUG NEWS DIGEST

TiSHUG News Digest ISSN 0819-1984

TiSHUG(Australia) Ltd
TiSHUG NEWS DIGEST

JANUARY/FEBRUARY 1988

Correspondence to:

PO Box 214
REDFERN NSW 2016

DIRECTORS:

Co-ordinator:

Chris Buttner..Tel.(02)8717753

Secretary/Librarian:

Terry Phillips.Tel.(02)7976313

Treasurer:

Bert Thomas....Tel.(047)541535

Technical:

John Paine....Tel.(02)6256318

Peter Schubert.Tel.(02)3585602

SUB-COMMITTEES:

Publications:

Bob Montgomery.Tel.(042)286463

Sysop:

Ross Mudie.....Tel.(02)4562122

Merchandising:

Cyril Bohlsen..Tel.(02)6395847

REGIONAL COMMITTEE MEMBERS:

Glebe:

Mike Slattery..Tel.(02)6920559

Central Coast:

Russell Welham.Tel.(043)924000

Liverpool:

Arto Heino.....Tel.(02)6038956

Illawarra:

Rolf Schreiber.Tel.(042)842980

Carlingford:

Chris Buttner..Tel.(02)8717753

Sutherland:

Gary Wilson....Tel.(02)5431314

Northern Suburbs:

Dennis Norman..Tel.(02)4523920

Coffs Harbour:

Keir Wells.....Tel.(066)551487

MEMBERSHIP AND SUBSCRIPTIONS:

Joining Fee.....\$ 8.00

Annual Family Dues.....\$25.00

Dues O'seas Airmail...US\$30.00

Publications Library....\$ 5.00

Texpac BBS.....\$ 5.00

BBS Membership:

Other TI User Group

Members.....\$10.00

Public Access.....\$25.00

Title	Description	Author	Page
A first lesson in XB programming	Software hints	Jim Peterson	15
Adding sound to monitor	Hardware hints	Alf Culloden	25
Agenda for 1988 AGM	Club news		8
All 256 characters in TI-Writer	Word processing		22
Author index volume 6 1987	General interest		30
Co-ordinators report	General news	Chris Buttner	2
Editorial	General interest	Bob Montgomery	1
FORTRAN	Software hints	D.N. Harris	14
From the bulletin board	Mail to all		4
Letter	Publicity	Les Tomlinson	4
Letter	Software author	Mark Beck	4
Link-it #15	Lower to upper case	Ross Mudie	8
Minutes of 1987 AGM	Club affairs	Terry Phillips	3
Program to type in	Addition exerciser	Geoff Trott	19
Program to type in	Banner	A. Saunders	16
Program to type in	Computer Notebook	Steve Whitehead	18
Program to type in	Learning music note	Colin Lees	16
Program to type in	Lower case characters	Phil Maier	19
Program to type in	Mouse squish	W.M. Johnson	17
Program to type in	Warrior	Mike Towers	18
Programming music with TI	Software hints	Jim Peterson	12
Programs that write programs #5	Software hints	Jim Peterson	14
RAMcard tip	Software hints	Hal Tonkin	6
Regional group reports	General interest		35
Secretary's notebook	Club news	Terry Phillips	3
Software piracy	General interest	Chris Bobbitt	9
Stop press - AGM	Club news		35
TI-Writer tips	Word processing	Geoff Trott	22
TiSHUG shop	For sale	Cyril Bohlsen	10
TiSHUG software column	Club software	Terry Phillips	11
Techo time	Fault of the month	John Paine	6
Techo time	Peripherals	John Paine	5
The Forth column	DV80 files to Forth	Mike Carroll	23
The Forth column	Forth and DSDD disks	Rich Stanford	23
The communicators	BBS information	Ross Mudie	7
The teaching computer	Software hints	Jim Peterson	21
Tidbits seven	Software hints	Wade Bowmer	13
Title index volume 6 1987	General interest		26
Younger set	Program, adven. hints	Vincent Maker	20

What will 1988 bring our beloved orphan computer?

Last year saw 32K expansions, RAMdisk cards, Printer buffers and Mini expansion systems being offered. Much of this was through the efforts of the Technical Coordinator, John Paine and his helpers.

From an Editor's viewpoint it was great to see the number of members contributing to the magazine with quality articles. Thanks go to those committee members who always submitted articles regularly. More importantly, how did you feel about the efforts of the 1987 committee?

There were a number of issues that rose during the year. Two of those were; communicating with other groups, internationally; and the lack of new quality software. Both were addressed during the year with some mixed results with other groups and with the announcement of a software competition sponsored by the club.

With the February meeting being the Annual General Meeting it is important that you have your say by being there.

This will be the last issue of the magazine that I will be editing. Whoever becomes Editor in 1988, I wish him well.

The editorial committee of Geoff Trott and Rolf Schreiber have put this issue together.

Bob Montgomery

Co-ordinator's Report

by Chris Buttner

Between you, me and the gatepost there are no gateposts, but if I have your attention, everything is as it should be. Now for the moment of truth. If you don't care how the club is managed, skip this section and go directly to EXPO 88. On the other hand, if you are interested in the future of your club read on.

We will meet in February to elect five Directors. As I mentioned in a previous article, these five persons determine club activity for the next twelve months. They are also responsible for whatever happens in the club. In essence, not a great deal has changed in this regard from our prior, unincorporated status. Every organisation needs a person or group of people to provide some form of leadership, to be answerable to the group and to accept responsibility for decisions made on behalf of the group.

An important difference is the way we now go about it. Our conduct is now regulated by the Memorandum and Articles of Association and also the NSW Companies Code. Whereas before there were certain things we chose to do, now there are certain things we MUST do. For example, we must elect five directors.

In the past, every position was filled by election at the AGM. Members were elected to fill "positions", and they then formed our club executive. These elected members were responsible for the particular aspect of club activity to which they were elected. The election was usually carried out by a show of hands. This has all changed in a way which provides greater flexibility for the long-term, continued existence of the club.

To begin with, if there are more than five nominations for directors, there will be a ballot. If only five nominations are received the ballot is dispensed with and those members are automatically the directors. Provided we have the necessary five directors, the club will continue to function. The Memorandum and Articles of Association, which you can think of as our "constitution", sets out what those directors can do, and how they go about it.

Once the directors are elected, they must elect a Chairman, and set about managing the club affairs. This is where the maximum flexibility has been provided. For instance, the directors could, if they so wished, take on the total workload of the Secretary, Treasurer, Magazine, Shop, Library, BBS and so forth. It would be a daunting task for just five! (My personal opinion is that the positions of Secretary and Treasurer should be filled by directors, because the directors must have a secretary to record minutes etc., and should have ready access to the financial standing of the club.) On the other hand, the directors have the power to appoint sub-committees for any purpose and to delegate powers to them. (This was done last year with the creation of the Editorial Sub-committee, Library Sub-committee and Shop Sub-committee.) It is conceivable the club could outwardly function in much the same way it has for the past two years. Whether it does or not is primarily up to you.

Do not sit back and take for granted things will automatically go on as before. Talk to those people who nominate for the position of director and find out what they think about the club and the way it should be managed. When you know this, you will be able to make an informed vote at the AGM.

Above all, be prepared to do something for the club. Cyril Bohlsen, our shop manager will not be able to take on the running of the shop in 1988. If no one comes forward, shop activity may have to be drastically scaled down or perhaps even abandoned. This is just one example of the need for members to participate, not only in enjoying club services and activities, but also in assisting to provide them for others.

As your outgoing Co-ordinator, I want to pay tribute to those members who have contributed substantially to the enjoyment each of us has obtained from the club over the past twelve months. Firstly, those with a high profile. Cyril Bohlsen and his band of helpers in the shop who have worked tirelessly to

ensure the financial viability of this club. John Paine and Peter Schubert, who have ensured our machines firstly keep running and secondly that we enjoy the latest technological advances at moderate cost. Now for those "behind the scenes" workers: Ross Mudie who has managed our BBS so ably, mothering and nursing it through modifications to the stage where it must be one of the most reliable BBS systems available anywhere. Russell Welham and his son Warren who have managed the publications library; and last but definitely not least, our Editorial committee headed by Bob Montgomery and Rolf Schreiber. It is very easy to be complacent about things, particularly if they happen regularly, and I am sure there are some members who have forgotten that the magazine is now published in Wollongong. Who would have thought this possible two years ago! Bob and his group have produced what I believe is the best TI club magazine anywhere, in terms of content and quality of production, at the same time as reducing production costs. The magazine production is a good example of the old adage "where there's a will there's a way". To all of you who have contributed in some way, great or small, to our past year's achievements, I extend sincere thanks.

EXPO 88 and TI-Faire, Brisbane.

Are you going? Then here is the information you have been waiting for. The Faire will be held-

on: 21st May 1988

at: Brisbane College of Advanced Education, Carseldine.

The venue is approximately 20 minutes drive north of the city centre and close to public transport. There are many motels and caravan parks in the area.

Who else will be there-

Myarc; Textaments and Rave 99 (represented with demonstration software keyboard and enhanced memory card etc.); and other user group representatives.

The Faire is recognised by the EXPO 88 Authority and is registered as an official Australian Bi-Centennial event.

Members of the Brisbane Users Group will endeavour to provide billets if you need cheap accommodation. To arrange this, or for further information about the Faire, you should write to Garry Christensen, the BUG Co-ordinator at 36 Hensell Street, Kippa-Ring. 4020. The BUG is conducting a raffle to help defray some of the cost of staging the Faire, and tickets in this raffle will be available at the AGM - Cost \$1.00 each. First prize is a 300/300, 1200/75 baud modem and second prize a box of diskettes.

RENEWALS

With 1987 behind us, and the Christmas extravaganza almost paid for, it is time to think about membership renewals. This will be a matter for the new Directors to consider but be warned! It is not far away now, so if need be, start saving. ○

continued from page 3

Moved by Rolf Schreiber, seconded Peter Varga that an amount owing to TISHUG by Shane Andersen (\$233.76) be written off as a tangible gesture of appreciation for Shane's dedication to the group. Motion carried on a show of hands.

Software commendation awards were presented to Arto Heino, Terry Johnsen, Shane Ferrett and John Paine. Other recipients, Dennis Hodgson, Ross Mudie and Craig Sheehan were not present to accept their awards.

Item 14 - Close of meeting

There being no further business the Chairman closed the meeting at 4.50pm advising that it was too late to set up the shop in the auditorium but goods would be available in the car park.

T Phillips
Hon. Secretary ○

Secretary's Notebook

by Terry Phillips

Firstly, a big welcome to the following new members who have joined us during the past 2 months:

Chris McCarthy - Helensburg
 Michael Keane - Oyster Bay
 Graham O'Connor - Lakemba
 Graeme Kemp - Ruse
 Trevor Warham - Seven Hills
 David Rogers - Woollahra

Hope you all enjoy your stay with the group and that you can all get along to the Saturday monthly or your local Regional group meetings.

The next meeting, as you all are no doubt aware, is the Annual General Meeting. The date is Saturday 6 February, 1988, and the venue is Burwood RSL Club Ltd, 96 Shaftesbury Road, Burwood. Shaftesbury Road runs parallel to Burwood Road - one street over on the city side, coming from the west. In fact it is very close to our regular meeting spot, Woostock. Walk to the end of Church Street and you can not miss it. Like most clubs, this one has strict dress rules, which you will need to comply with or the RSL Club management may refuse you admittance. Typically, shorts are OK, provided long socks and shoes are worn. Shirts should have a collar. Wear sensible casual dress and there should not be any problems. The club has been booked from 1pm until 4pm, so given 30 minutes to get in and settled the AGM should commence at 1.30pm. PLEASE BE ON TIME. The RSL club's other commitments mean we cannot extend our 4pm deadline.

SPECIAL MAILINGS - Shortly after the closure of nominations, I will mail to each financial member a list of nominees, together with the names of their nominators. Already mailed is a copy of the Memorandum and Articles of Association. This is your personal copy to retain for reference purposes. If your copy has by some chance not arrived, let me know so that I can send you a replacement.

So, with 1987 over, how did we perform. Membership records show that during the year, we attracted 38 new members and finished the year with 305 financial members. Approximately one-third of the membership as at the time of the last AGM did not renew their subscriptions. Will this happen again, or are we down to a nucleus of dedicated TI users who will remain with this computer through-out 1988? My early prognosis is that we will lose a sizeable chunk of our current membership. I base this on the fact that there were 10 memberships which expired on either 30 November or 31 December. Reminders sent to these people brought only 2 renewal responses.

With the majority of members due to renew on 30 April, 1988, this will be the telling time for the group throughout 1988.

PENFRIEND WANTED - Jack Zawediuk, Editor of the Lehigh User Group newsdigest, writes to say that his daughter is interested in a penpal, male or female, in the 14 to 17 years age group. Some of her interests are biking, talking on the phone and going out with friends. If interested, write to Sheila Zawediuk, 719 N 12th Street, Allentown, PA 18102 USA.

JIM PETERSON writes on the Picasso Publisher program of Arto's, which apparently has been uploaded and downloaded on just about every bulletin board across the USA. Jim writes -

"Was very sorry to hear about the unfortunate affair of Arto Heino's Picasso Publisher. In case you don't have the details, it seems that some user in Australia sent a copy to Jerry Steinberg in California, and allegedly stated that it was fairware. Jerry wrote a rave review for the Los Angeles UG Newsletter, and stated that it had been placed in the UG library. Terrie Masters, the President and Editor of the newsletter, added a postscript that she couldn't see any fairware notice on it

and that therefore it would not be distributed from the library until clarification was received from Australia. However somebody, uploaded it to Genie, and it was promptly downloaded and uploaded back up to most of the BBS's in the country, I imagine. I met Terrie at the Chicago Faire and asked her if she had heard from Australia. She hadn't, so I told her about the paragraph in the Sydney Newsdigest, which I had just received. She cornered Scott Darling, the Sysop of Genie, and had me tell him about the Sydney article. His only comment was 'Well, it's too late now.' Terrie asked me for a copy of the Sydney article, which I sent to her, and I expect that the next LA newsletter will carry an editorial with some very harsh words about uploading copyrighted software to BBS's. I also wrote an article which will be published in the next Spirit of 99 newsletter, and hopefully republished in many others, telling what happened and recommending that all user groups take up a collection for Mr Heino, from whoever has downloaded the program and intends to keep it. I also publicised the matter on our three local BBS's and brought it up at our user group meeting. One of our members told me that he had already sent payment to Mr Heino, and another one gave me \$20 to send along with my payment to him. This is an excellent example of what happens when people decide to ignore copyright notices."

Well said Jim! I have received the newsletter referred to by Jim which carries his article and have sent a copy to Arto for his information. See the Spirit of 99, December issue.

SOFTWARE COMPETITION UPDATE: As at early January, 4 entries have been received. Remember they must be lodged by no later than the February meeting date.

Annual General Meeting - 7/2/87
 Minutes - Agenda items 12 to 14
 Recorded by Terry Phillips

Item 12 - Life Member Nominations

Moved Fred Morris, seconded Ray Brown that John Robinson and Terry Phillips be nominated as life members for their contributions to TISHUG. Motion carried by show of hands.

Item 13 - General Business

(a) Illawarra RG Proposals

Motion by Illawarra RG - That the AGM agree in principle with TND article (JAN/FEB issue) written by Illawarra RG, namely that TISHUG operate with no surplus and re-organise contact with RG's in smaller groups other than central meeting place.

Moved Geoff Trott, seconded Rolf Schreiber. Speakers against the motion were Mel Copeland, Ben von Takach and Brian Graham.

Motion defeated on show of hands. Fred Morris asked the new committee to investigate more effective ways of contact including software distribution.

(b) Other Matters.

Chris Buttner advised the meeting of the need to appoint an auditor, preferably with suitable accounting qualifications. Peter Varga put forward the name Peter Rhona.

Russel Welham queried whether all meetings could be held at the Burwood RSL. The committee advised they would look into this matter.

Gerry Levy asked if pensioners were eligible for a reduction in membership fees. Chris Buttner advised no under TISHUG's present constitution.

Percy Harrison asked for a vote of appreciation for the outgoing committee. This was done by acclamation.

continued on page 2



Letters to the Editor

From the Bulletin Board

Les Tomlinson,
93 BARNETTS ROAD,
BEROWRA HEIGHTS
N.S.W. 2082
AUSTRALIA

12th January 1988

Dear Editor,

Shane Anderson makes some valid points in his letter regarding promotion of the User Group in December 1987 TND. I think we do need to indulge in a little aggressive promotion, firstly, to try and maintain and possibly increase numbers, and secondly, to let the hundreds, possibly thousands of TI owners out there know that their purchase of our little orphan some four years ago was NOT a mistake.

We need to inform this lost legion that help is at hand, in the form of fault diagnosis, repair facilities, technical assistance, a vast supply of software, a first class BBS, and of course, the excellent mini-PE System developed by Peter Schubert and John Paine. The problem is how to achieve all this for the maximum results from the minimum outlay. TI owners do not fit into one easily identifiable group, and targeting them could be difficult. As a first step I would suggest ads be placed in the Computer columns in the Saturday Herald and Telegraph and the Weekly Trading Post. Also, the For Sale columns in some of the more widely distributed weekly local papers (such as Maxi-Ads) may be worth a try.

One other avenue that may be worth trying are the Bulletin Boards provided free of charge by Supermarkets, etc. in many Shopping Centres. A brief ad could be printed on an appropriate sized card and sent out to members via the monthly TND for insertion on one or more of these boards in the member's local Shopping Centre. We would need to follow up any responses to an advertising campaign with an information sheet detailing the functions of the Group, the new equipment available and possibly a sample copy of the TND.

These are just a few thoughts on some of the things that could be done. Perhaps someone else in the Group with more experience in advertising and promotion may be able to come up with some better ideas, but whatever happens, I think we do need to become a little more aggressive to ensure the Group and our favourite orphan survives and grows.

Regards,
Les Tomlinson.
(BBS User "UNCLE")

Open Letter from Mark Beck

My name is Mark Beck and I am the author of the CREATIVE FILING SYSTEM. I am in the process of completely rebuilding the CFS by including the best items from other data bases into the CFS. I need your help in several ways.

First of all, I need the addresses of all CFS users in your group. This will allow me to add their names to my update notification list. It will also allow me to get their ideas and input. Currently I have about 175 registered users on my list. With the millions of TI99/4A computers still in use I know there are more than 175 actual and potential CFS users out there. I need to reach these people and your user group can help by getting the word out in your newsletter.

Secondly, I need your ideas on what you consider to be the requirements of the ultimate database. Please keep in mind the restrictions of the TI99/4A when formulating your ideas.

I pledge my support by improving the CFS to meet the needs of the users. The CFS is the only program I have created and I plan to keep on improving the CFS.

Co-ordinators note:

If ever there was a sincere plea from a software author this has to be it! If you use CFS, for goodness sake tell him and let him know your likes and dislikes. Also, do not forget to show your appreciation of his program in a tangible way.

Here are some examples of mail from the BBS. If you can help and do not have access to the BBS, send a letter to the secretary with all the details, and we will publish the result and send the answer to the originator of the mail.

MAIL FROM : LARRY

SENT ON Sunday 03/01/88 at 22:14:54

All the EXTENDED BASIC programs, for example, TIRUNNER1XB can be loaded on any RAMcard with a small change to the load. For example, in TIRUNNER1XB and 2XB change RUN "DSK1.TIRUNNER1XB" to the drive number of the RAMcard you are putting it on. You will then find it will run without changing your ramcard to drive 1.

(TI-runner calls up screens from a disk file, so another change may be needed for this program. ED)

MAIL FROM : SYSOP

SENT ON Sunday 3/1/88 at 2200Hrs

re Program BIGFOOT.

I found that this program would intermittently refuse to speak via the speech synthesizer. The problem was one of timing in the 8 bit bus on a speech read. I have modified BIGFOOT2XB which is in the Users' Upload/Download area for anyone affected by this problem. BIGFOOT1XB is OK, no change needed. If anyone missed out on getting BIGFOOT when it was on the BBS in December then let me know via BBS mail I will put it on again as well.

MAIL FROM : DOBELL

SENT ON Thursday 31/12/87 at 17:17:45

I have a Peter Schubert modem card in my PE box. Peter tells me that the card, with a minor modification, can have auto dial facilities. The hard part is to find someone to write the software. I would have thought this was a very desirable option, so I wonder why it has not been done already. We are currently running a software competition, and software to do this would have to be in the race, just in time saving alone. Is there anyone out there willing to have a go?

ps Mass Transfer has an auto dial menu. Is all it needs software?

MAIL FROM : REDDRAGON

SENT ON Wednesday 30/12/87 at 22:26:41

Help wanted!!! Does anyone know the password needed to get the vector plotter in Hitchhiker's Guide to the Galaxy. REDDRAGON...

MAIL FROM : GOWFAR

SENT ON Tuesday 29/12/87 at 16:23:12

Help needed!!

Can anyone with ASSEMBLY knowledge please help with this problem. I have none but that is not the problem - or is it? The problem is that I have a Compact and Protected assembly file (no source code) that has a minor problem in it. What I want to do is to be able to extract, from the computer's memory, what would look like an ASSEMBLY listing. Alternatively, if you are adept enough, can you tell me how to get my assembly program back into a source code, if possible, as I would not know where to begin. I have E/A and MINI-MEM if that can help. ANY HELP APPRECIATED. P.S., it is not a major problem, it is just that the program will not support CD on the modem. That is all I want it to do. If you have any handy CALL LOADs that you think may work, please pass them on too. Ta!!

MAIL FROM : LARRY

SENT ON Saturday 12/12/87 at 15:23:25

I have had a lot of problems getting Memory Plus and Horizon RAMcard working without errors. I have now got all the problems sorted out. I found the review in Micropendium was wrong. The Memory Plus must be set at >1000 and the Horizon RAMcard at >1400 or the computer memory will jam every time. The Horizon ROS version 7.1 will not work with Memory Plus but version 6.4 does very well.

Techo Time with John Paine

A new year is now with us, and perhaps we should reflect on some of the achievements of the year gone past. The striking feature of the year gone by I believe, was that a lot of members suddenly realised that they could contribute their own labors and energies to keeping and improving their systems, by building their own peripherals and gadgets.

The RAMDISK project was certainly the catalyst needed to lift the level of confidence of many a member, in their own ability to help themselves. After the successful completion of the RAMDISK, people were asking about building 256K print buffers. These same people at some time or other said "I can't do that!" or "I am hopeless at soldering and"

Last year saw the introduction of pre-assembled and tested 32k memory upgrades for non-expanded systems, rather than just a kit of parts. Late last year technology caught up with this peripheral, and the single chip version of the 32k was released. This small card will now be the only 32k memory expansion available through the club.

1987 also saw the introduction of the mini-PE system, which allowed an economical approach to expansion for those who could not acquire a standard PE box. The fully optioned configuration consists of a double sided double density disk controller, up to two RS232 ports, a parallel printer port and 32k memory expansion. The prototype of the PE Box multifunction card was also demonstrated, and will be available by the time this article goes to print. Other projects and developments included the manufacture of modules to run on VIATEL and to provide simple word processing and diagnostics.

All in all not a bad sort of year for a dead computer.

1988 will see the introduction of a RAMdisk for the mini-PE system and a real time clock project which can be installed in the console, on a RAMdisk board or even in the speech synthesiser. Your imagination will be the only limitation. Software to set and read the clock has already been developed and the unit has now been debugged and is running fine.

What else, I hear you say..... Well that is up to you !!!!!!!!!!!!!!!!!!!!!!!!!!!!!

Peripherals for the TI99/4A

As the title of this article suggests I will try to indicate, in general terms, some of the peripherals available for our machine, and where possible give details of the minimum requirements needed to support these peripherals.

I will make the assumption that the most interest will be shown by those members who have just the barest of systems, because they are the people that have asked for something like this continually. It is not as if their pleas have fallen on deaf ears, but rather it is lack of time to get around to producing something. Any way, here goes.

The basic system that most of us purchased consisted of a console, power transformer and a little black box that plugged into the TV, called a modulator. The console had 16k of video display memory and with a cassette interface built in to save all those programs that took hours to type in.

At the time that most of us bought the computer we also had the option to purchase a large grey box called the Peripheral Expansion Box or PEB as it is now known. The PEB comes with a heavy flat cable which plugs into the side of the computer and the PEB consists of a power supply and a card cage which has 8 slots to accommodate 7 extra cards to expand the system. The first slot accommodates an interface card which provides buffering and conversion from the 44 way expansion plug on the console to the 60 way expansion bus inside the PEB.

The cards available for the PEB originally comprised of 32k memory, RS232/PIO, Disk Controller and PCode card. A number of third party producers now have other cards available. Also a number of third party

producers offer alternate expansion units, that usually incorporate some of the features of the original cards supplied by TI, and in most cases these features are greatly enhanced. Generally these expansion units are classified as micro-expansion systems or mini-expansion systems. More on these later.

The 32k memory card allows the user to run machine language (assembly) programs as well as extending the memory for large Extended BASIC programs. BASIC programs cannot utilize this extra memory. The memory card was one of the first peripherals that came under the scrutiny of third party developers and is now available in many forms. The most common form for those without the expansion box is as an add on board mounted in the console and powered by the console. Due to changes in chip technology since 1983 these memory boards have had the memory chip count reduced from over 16 chips to just 1 chip. The single chip 32k memory expansion is available from the club shop.

The RS232/PIO card allows the computer to communicate with other devices, for example printers, modems, plotters, and is mandatory for word processing and telephone communications. The RS232 section allows for serial transfer of data and the PIO (Parallel Input Output) allows parallel transfer of data (mainly to parallel or centronics type printers).

There are many third party cards available from the US from manufacturers such as CorComp and Myarc and as of today we have our own locally produced multifunction card which has RS232 and PIO as options.

Disk Controllers would be perhaps the most common peripheral after the 32k memory expansion. The original TI Controller (card version) is capable of formatting, reading and writing to disks in the double sided single density mode, although this capability could not be utilized with the original disk drives supplied by TI. These drives were only single sided. By retro-fitting double sided drives, the effective capacity of the disk system is doubled.

Third party disk controllers (Myarc, CorComp, Mechatronic and the locally produced AT card) all support double sided double density formats which double the capacity again over the original TI Controller. In fact Myarc have a version which will run quad density with the appropriate type of disk drive.

To run a disk system the minimum requirement is some type of disk controller, disk drive(s), a power supply for controller and the disk drives, and some form of disk management software. All third party suppliers of controller cards usually supply this software as part of the package. They do not normally supply the disk drives and power supplies or the necessary interface cables.

Up until now I have only mentioned the cards that are available for the PEB. Because PEBs are no longer available, third party manufacturers also produce mini-PE, systems which usually combine all three of the functions mentioned previously. CorComp produce a neat little black box that plugs into the console expansion port and combines the functions of disk controller, RS232 and PIO as well as 32k memory. Myarc have produced something similar, but the locally produced AT mini-system will allow a lot more versatility because each function listed above can be added at any time. The AT system does not look as pretty as the mass produced American systems but offers a lot of enhancements the others do not. (See Ross Mudie's review of last year).

For those who do not wish to use a mini type system, the Hunter Valley group described a 3 slot home built expansion and in 1986 there were details in this magazine of the Poorman's System, which could support 2 cards.

Other third party products available include a host of RAMdisks, Realtime clocks, and print buffers.

A RAMdisk is a mass memory board which, with the appropriate software, can emulate the functions of a disk drive. The advantages are speed of data transfer and flexibility of the system. The major disadvantage of the RAMdisk is the capital cost compared to the conventional disk. To operate a RAMdisk, you still need a minimum of one physical disk drive to load up the RAMdisk.

RAMcard Tip

RAMDISK RECOVERY.

by Hal Tonkin, (Ottawa Users' Group December 1987)

There are two types of memory devices used in RAMdisks. They can be divided into the dynamic memory type, which cannot be easily backed up (data retained) but which have the advantage of lower cost and board density and the static memory type, which can very easily be backed up with batteries (hold data after power is turned off), but are usually more expensive per byte of data of storage and usually take up a lot more board space.

Examples of the first type of RAMdisk would include the Myarc 512K and CorComp 512K. The Horizon RAMDisk and TISHUG RAMdisk are examples of the second type.

RAMdisks are available as card peripherals but CorComp manufacture their 512k RAMdisk as a stand alone product for those users without the PEB and a standalone version of the TISHUG RAMdisk is under development.

Realtime clocks for the TI have been available for some time, with cards available from CorComp (Triple Tech Card) and from MPB. A locally developed clock will be announced in the next issue of the magazine, which can be built into almost any existing peripheral or console. CorComp also released a version of their clock as a stand alone unit.

To many members this article may seem to be a little old hat and very general. It has been aimed primarily at those users who still only have the basic console and may wish to upgrade their systems. O

Fault of the Month

This month's fault concerns a problem which, although easily fixed (eventually), presented a problem in trying to explain.

The console involved seemed to work reasonably well for long periods of time, and then would suddenly play up something fearful. Turn the machine off and then on and all seemed OK.

Before carrying on, a brief history of this console is in order. The owner was until about 15 months ago, purely a cassette operator. He decided to upgrade to disk and built his own Poorman's system for the disk system. He already had a 32K memory board fitted internally and ran a Peter Shubert stand alone RS232 and combination modem box.

As can be deduced, with the overall system quite a load was presented to the data bus buffer which had to drive all the above peripherals and the internal GROMs, sound chip and what ever was plugged into cartridge port. Also, the speech synthesiser was also connected up internally in the console.

When Extended BASIC was plugged in the problems seemed to occur more frequently, and the Extended BASIC dirty module port lockup was suspected. The owner instigated a regular cleanup routine for the module port but the problem continued to get worse.

The immediate problem appeared to be related to the load placed on the buffer by all these peripherals so I immediately replaced the 74LS245 buffer with a new one from stock. Instead of curing the problem, the console would accept no load at all on the bus. Plug in Extended BASIC or a super cartridge and crash.....all this with the 32K memory, speech and modem disconnected.

The problem was purely that the nice new buffer chip simply would not buffer. Since replacing the buffer chip again, all was OK.

MORAL..... assume new chips may not work.. After testing the original chip and using it in a situation of less load and finding it to be OK, I am forced to say to those with systems with everything running internal to the consolebeware.

With tongue in cheek, I hope to give some details of a real time clock in next month's magazine, which can run in the console or be built into an external peripheral device. The speech box sounds like a good place.....

Finally

I will not be standing for election as a director of TISHUG Australia in 1988. The reasons are many and varied, BUT I am not deserting this machine and will try to assist anyone with technical problems. I will also endeavour to write the odd article, but cannot be expected to contribute to every edition. O

If you back up your Ramdisk before turning off the power every time you use it, then read no further. But if you are like the rest of us mortals, who take a backup copy only once a month or so, then this article may help.

Every now and then when I turn on the power to the system, I find that the Ramdisk Operating System (ROS) has been corrupted. If the system will still come up and I can get extended BASIC to run, then it is a simple matter just to reload the ROS. Usually the files are still OK, and at this point I breathe a sigh of relief, and then take a backup. However, last week when I turned the power on, there was nothing - no TI banner, no menu, no response on the keyboard, just a little red light on the ramdisk indicating that it was looping in its power-up routine. PANIC!!! I had just finished entering a C program, and did not want to have to type it all in again. The ramdisk instructions say that when this happens, the only way out is to power down, remove the card, remove the batteries, wait, re-install the batteries, re-insert the card, and re-load the ROS. However, this would also destroy the contents of the rest of the memory - that is, all my precious files. There had to be another way. This time there was, and if this happens to you, perhaps the following procedure will also save your day.

1. Turn off the expansion box and the console.
2. Put a mini-memory cartridge in the console.
3. Turn on the console ONLY. At this point the TI banner should come up, and after pressing a key, the mini-memory menu should appear.
4. Select option 2, EASY-BUG.
5. Now turn on the expansion box. Since the console was already on, it did not try to execute any of the power up routines of the cards in the expansion box.
6. Select CRU single bit I/O by typing C then type C1000<ENTER> 1<ENTER> to enable your ramdisk. CRU address >1000 is where my ramdisk address is set. If yours is at a different address, then enter the appropriate address instead of >1000. When you turn on the bit, the red light on the ramdisk card should come on indicating that it has been selected. If it does not, then this recovery procedure will not work.
7. Exit CRU single bit I/O by typing a period.
8. Examine memory address >4000 by typing M4000<ENTER>. This is the address of the first byte of the ramdisk memory and should contain the value >AA indicating that there is a device present.
9. Alter the byte at memory address >4000 to be >00 to indicate that there is no device present by typing 00<ENTER>.
10. Exit CPU memory addressing by typing a period.
11. Turn off the CRU bit >1000 by typing C1000<ENTER> 0<ENTER>. The red light on the ramdisk should go out.
12. Turn off the expansion box, then the console.
13. Remove the mini-memory cartridge and insert extended BASIC.
14. Turn on the console, and then the expansion box.
15. Type FCTN= to reset the system. All the cards in your expansion box except the ramdisk should reset themselves, and the system should not hang up since the ramdisk did not try to go through its power-up routine.
16. Put your ROS disk in drive 1, and select extended BASIC.
17. Reload the ROS, and with luck your files will still be there.
18. Take a backup NOW, before anything else goes wrong.

This procedure may not work all the time, but it did for me, and may be worth trying before removing those batteries.

Retyped by Ross Mudie, 23rd December 1987. O

The Communicators

Special Interest Group for Users of the TEXPAC BBS.
by Ross Mudie, 5th January 1988.

The year 1987 saw considerable change in the TEXPAC BBS. The facilities of the BBS were enhanced during the year as detailed in the December 1987 TND magazine. Since that article was printed another new area has been added to the BBS as detailed later in this column. The user log which is maintained on disk at the BBS has been summarised as follows. During the year from 1/1/87 to 31/12/1987 the BBS received 5053 valid calls from 112 users and was on line for 1583 hours & 17 minutes. The whole summary has been available on the BBS for users to compare their usage with that of others.

New material for the BBS is a constant problem. During 1987 I relied very heavily on a willing few who have overseas or local contacts or are prepared to write programs or tutorial material. My sincere thanks to ALL contributors to the BBS in 1987. I hope that more of our members will be prepared to provide topical and usefull material to the benefit of the rest of the membership in 1988. In a club like TISHUG the benefits to the members are directly proportional to the willingness of the members to work together and contribute their talents to the benefit of all.

On the 1st of December a new major facility came on line in the BBS. This is a user program Upload/Download area. The new facility maintains the principle of supporting users with the minimum of system expansion to utilise the facility. Part way through December an enhancement was provided which allows Program Information files to be provided by the uploader.

There are a number of BBS users who do not have printers to allow a hard copy to be taken from the instructions for the Upload/Download area so they are reproduced here. The User program area has been very popular, filling to capacity within a few days. Most programs will be left in the area for approximately one month. A planned future enhancement will allow the uploader of a program to delete the program.

1. HOW TO OPERATE USERS' PROGRAM AREA.

This facility which came on line on 1st December 1987 allows users to upload memory image BASIC & X/B programs. It allows users with only the most BASIC of TI99/4A configurations to upload a program or to download programs placed on the system by others.

Version 3Z of the BBS provides an information file for each program.

** LIMITATIONS.

a) Maximum program size of 50 disk sectors, ie 12536 bytes of program plus the header sector. (This is 11952 bytes of program space free when SIZE is taken on a TI99/4A with 32K memory expansion attached). Programs greater than 46 sectors will require the user to have invoked CALL FILES(1) if a disk controller is attached to the computer. The BBS will warn disk users to invoke CALL FILES(1) at download time.

b) The system will store a maximum of 12 programs at present.

c) Deletion is password protected.

d) The BBS reserves 150 sectors for text type mail since this facility shares the BBS mail disk.

** OPERATION.

The Users' Programs area is menu item 1 in the Extended Facilities area. If there are programs already in memory then a menu is given followed by the prompt:

1 Download from BBS, 2 Del, 3 Prog Info,
4 Upload a program to BBS, 5 Exit >

[1] DOWNLOAD from BBS.

By selecting 1 from the facility prompt you will be prompted:

Enter NUMBER for program >

Either press ENTER only to escape or the number for the selected program followed by ENTER. The normal download sequence is then followed, on screen instructions are provided.

[2] DELETE. (Password Required).

Used by SYSOP or aide to remove old programs. There is a delay of up to 30 seconds when DELETE has been used to rebuild the menu.

[3] PROGRAM INFORMATION.

By selecting 3 from the facility prompt you will be prompted:

Enter NUMBER for program >

You will then read the information provided by the user who uploaded the program.

[4] UPLOAD a program to BBS.

a) Enter the required name for the program in the menu, observing the following limitations.

- i) Maximum 10 characters.
- ii) No period character in names.
- iii) No spaces in names.
- iv) The name END is invalid.
- v) The first character must be less than ascii 91.

If the BBS doesn't like a name you use it will re-prompt at a higher prompt. You can escape this prompt by pressing ENTER.

b) Upload instructions are then given as follows:

- i) Exit to BASIC or X/B. (BASIC can be used for X/B if you don't try to list or run).
- ii) Load the program from your tape or disk.
- iii) Upload the program once in memory by using SAVE RS232 or SAVE RS232/2 as appropriate. (The number 256 byte blocks will commence to count down after a 15 second delay).
- iv) When your cursor returns, then return to terminal mode and press ENTER, ONCE ONLY.
- v) You may then enter information about the program which has just been uploaded.

If you make it back to terminal mode before the BBS has rebuilt the menu then there may be a delay before the menu is given again. If nothing has happened after 30 seconds then try ENTER or the digit 1 followed by enter.

On return to terminal mode the BBS allows entry of information pertaining to the program just uploaded. The BBS provides a header of showing when a program is uploaded, the program name and who it was uploaded by. The user can enter into an open ended file any operating instructions for the program. The MAIL routine is used to put this information in the file, so both lower and upper case in addition to normal punctuation may be used. This file can be loaded by the SENDMAIL program. At the end of the file press enter twice after the > prompt without entering anything else to exit. Ctrl H can be used to back space corrections in the current line only.

[5] EXIT.

This returns you to the Extended facilities prompt.

** IMPORTANT POINTS TO NOTE.

Please remember that the responsibility of only putting worthwhile programs on this area rests with the person doing the upload. The SYSOP only has control over deletion from the area. Programs in this area can be downloaded by any user, they are NOT private like the mail system. The printer log at the BBS will show program names and when uploaded or downloaded. If an error occurs in transmission during upload (such as noise on the phone line) then an I/O error will be given and the program will not be saved on the BBS.

Do not upload a program which is in the Extended BASIC protected mode as it will not download in BASIC, wasting the time and STD call costs of some users. Programs uploaded in protected mode will be deleted by SYSOP.

2. BBS ACCESS.

Access to the BBS is permitted for registered users only due to problems caused in the past by unregistered people. Requests for access should be sent to the Secretary TISHUG, PO Box 214 Redfern, 2016. Subscription rates are shown on page 1 of the TND magazine. ○

Linking Extended BASIC to Assembly

15 LINK IT 15

CONVERTING LOWER CASE TO UPPER CASE.
by Ross Mudie, 21st December 1987.

This routine was written in response to a request from TishUG member Robert Brown. Extended Basic can be used to convert the characters in a string from lower case to upper case, but it is a bit slow, especially if the string is long. This assembly routine copies the string into the assembly program from the variable \$S using the utility STRREF, then tests to see if the variable was empty. The length of the string is then placed in a register which is used as a counter and the values equal to lower case "a" and "z" are placed in registers for testing. Each byte of the string in the buffer is then tested to see if it is in the value range of lower case alpha. Each character which is in range has Hex 20, or ascii 32 subtracted from it, converting it to upper case. When the whole string has been tested and converted as necessary the string is passed back to extended BASIC in the same string variable.

One very important purpose of this article is to help our members who are trying to come to grips with assembly. After typing in the source file using an editor (Editor/Assembler, Funnelweb, etc) save the file to disk in DISplay VARIable 80 format. Then load the Assembler. The Source file name is the name you saved the source file under. For the source file I used DSK1.US and for the object file DSK1.UO, (I always use short simple file names). If you want a list file you can specify PIO. to go to a parallel printer, (the "." is needed). For the options specify R because all the Register usage is prefixed with "R", L to get the list file and S to get a symbol table in the list. The "C" option must not be used with extended BASIC because the X/B loader will not recognise compressed object code. If a list file is not required then press enter at the List file option and omit the L and S in the options.

One final comment before getting into the program.... There is only one way to get into assembly and that is get your hands on the keyboard, your head in the book and put in a lot of hard work!

```
* This program converts lower case characters in a
* string into upper case. It will handle strings up to
* 255 bytes in length. Type in the extended BASIC
* program to test the utility from the keyboard.
* Ross Mudie 21/12/87.
* EXTENDED BASIC TEST PROGRAM.
* 100 ! SAVE DSK1.LOAD
* 110 CALL INIT ! Load assembly utilities
* 120 CALL LOAD("DSK1.UO") ! Load assembly Object file
* 130 ACCEPT $S ! Input for up to 255 bytes
* 140 CALL LINK("CONVUP",$S) ! Do the conversion on $S
* 150 DISPLAY $S ! Show the results on screen
* 160 GOTO 130 ! Go back for another try
* SOURCE FILE LISTING.
* S=US O=UO My Source and Object file names
```

```
DEF CONVUP CALL LINK("CONVUP",$S)
```

```
STRREF EQU >2014 Utility which gets a string from X/B
STRASG EQU >2010 Utility which sends a string to X/B
GPLWS EQU >83E0 Address of GPL Work Space (for X/B)
STATUS EQU >837C GPL STATUS byte address
FF BYTE >FF Used for max string length in BUFFER+0
BH20 BYTE >20 Used to subtract from lower case to
EVEN make UPPER case

CONVUP MOV R11,@SAVRTN Save the return address
LWPI WS Load our own register Work Space
CLR RO Element zero in variable $S
LI R1,1 Argument 1 in link list
MOVB @FF,@BUFFER Max string length in BUFFER+0
LI R2,BUFFER Where to put the string
```

```
BLWP @STRREF Get the string from extd BASIC
MOVB @BUFFER,R3 Get actual string length byte
SRL R3,8 Make byte sized value into word
MOV R3,R3 Is it a null string?
JEQ END If a null go back to ext'd BASIC
LI R4,>6100 Hex 61 is 'a' lower case
LI R5,>7A00 Hex 7A is 'z' lower case
LI R6,BUFFER+1 The value in R6 is now the
* address of the first byte
* of the string in the BUFFER
LOOP1 CB *R6,R4 Test the byte in BUFFER pointed
* to by R6 for value less than lower case 'a'
JLT NEXTBY Jump if Less than Hex 61 (a)
CB *R6,R5 Test against value of lower case z
JGT NEXTBY Jump if Greater Than Hex 7A (z)
SB @BH20,*R6 Subtract >20 from byte in BUFFER
* position pointed to by R6 to convert
* lower to upper case for that byte

NEXTBY INC R6 Point to next byte in BUFFER
DEC R3 Count down in string byte counter
JNE LOOP1 Finished yet?
BLWP @STRASG Send string back to X/B, R0,R1,R2
* are unchanged & length byte still in BUFFER+0
END CLR RO To prevent false indication of
MOVB RO,@STATUS errors on return
LWPI GPLWS Reload GPL Work Space for X/B
MOV @SAVRTN,R11 Reload return address in R11
B *R11 Branch to address in R11 to
* return to extended BASIC
SAVRTN BSS 2 These are down here so that they
WS BSS 32 are easy to delete if the program
BUFFER BSS 256 is converted to CALL LOADs.
* Refer to "HOW TO CONDENSE ASSEMBLY IN
* X/B USING CALL LOADS" in TND September
* 1987, page 20.
END
```

SINCLAIR
RESEARCH



'Sooner or later, Clive, someone is going to find out'

TishUG (Australia) Limited
Annual General Meeting - 6 February 1988
Burwood RSL Club Limited - 96 Shaftesbury Road Burwood

To commence 1.30pm. Please be at the meeting venue by 1pm so that business can get under way promptly at the commencement time.

Agenda

1. Meeting opening.
2. Apologies.
3. Reading and confirmation of minutes.
4. Correspondence and dealing with same.
5. Reading and dealing with recommendations from the Board - Life Members and other matters.
6. Directors Reports, presentation of Accounts and Auditor's Report.
7. Unfinished business from last AGM (if any).
8. Election of Returning Officer and two (2) Scrutineers.
9. Election of Directors.
10. Election of Auditor.
11. New business (if any).
12. Meeting closure.

Software Piracy

by Chris Bobbitt, Asgard Software

The home computer industry is over ten years old. Ever since the mid-70s, people have been able to own, and on occasion, even get useful work out of personal computers.

For almost the same amount of time, people have been pirating (or obtaining illicit copies to put it nicely) other people's programs. For almost as long, software companies have been worrying about it. The first programmer to copy protect his software may have thought he had found the answer, but all he really succeeded in doing was putting thousands of people into a vicious cycle of creating and breaking copy protection. Nowadays it might not be unlikely that the best cryptographer in the world lives just across the street.

A whole industry has been created to serve the needs of both the software manufacturers and the consumers. A vast amount of time has been spent writing sophisticated programs to protect and unprotect software, a task which is really as meaningful as sand in the desert. Thousands of people have devoted valuable time and energy that could have been diverted to more useful tasks just to prove some basic theories of the market system that are literally unchanged since Adam Smith himself.

Capitalists (and, unknowingly, the entire software industry is dominated by these people) believe that every single conscious action by a human being has some economic motivation. This motivation may be intangible and not necessarily related to the more familiar market economy. They believe that there is a give and take in every human thought and action that follows the same principle that underlies the market system. Justice, as a concept, is merely the personification of "value given for value received". Piracy has the same economic motivations as any other action, most of which are based on this concept of justice.

There are very few real reasons why people openly give and receive illicit copies of programs, but there are literally hundreds of rationalizations and, surprisingly, many are valid. However, almost all are merely derivations of each other. As a rule, people usually do not take what is not theirs unless they regard not taking it as the more grave injustice to themselves. People pirate software because:

- (a) it is over-priced;
- (b) it is unavailable and/or the company is unknown or out of business;
- (c) as a warning or a preview for friends;
- (d) because it is not useful to the user and thus worthless and not worth keeping; and
- (e) people just plain do not know that it is copyrighted.

The arguments for these rationalizations are excellent. Most software is highly overpriced - net profits in excess of 200% to 300% are not uncommon, while profits of up to 1000% are standard in some sub-markets (the ones that TI wants to get into). It is not that these companies are greedy (it might be though!), it is simply the fact that any company will always charge the most that the market will bear, and it just so happens that even though the majority of people have never wanted to pay that much for bad software, until recently there was always a sizable minority that would. When almost all software is useful to only a particular audience (or none at all), and prices are high, people can readily justify pirating software to protect themselves from being taken!

A lot of software companies do not go to any length to make their software visible, or for that matter their whole company visible. Granted, advertising is very expensive, but many small companies do not even bother to put their company address in the program listing or documentation! Most people are honest enough to buy a copy of a program if they received it pirated, find the program is useful, know where the company is, and if the price is right. There is also the opposite extreme where a software company is very visible, but just does not know how to market

their programs adequately, or make purchasing a program complicated. In this case people feel they are doing a service by making copies for friends who could not otherwise obtain them.

Sometimes people will pay a lot of money for a program that does not function adequately or as advertised. It irritates them, and makes them feel cheated. It is pretty easy to justify giving out copies of this software either as a warning to other potential buyers, or as a means of revenge against the company. This description can readily serve as an explanation for both reasons 'c' and 'd'.

Finally, sometimes a program is just so amateurish that people can readily mistake the software for public domain, and are completely unaware of its commercial nature. Much of the stuff from magazines and books falls into this category. This is merely stupidity on the part of the software manufacturer.

There may be a few other reasons for stealing software, but I believe these are the major economic rationalizations. Minor variations on these themes can be found in any defense of piracy.

I am not saying that piracy is acceptable. I am just saying that there ARE valid, justifiable reasons for piracy. This article was written to explain how to make these reasons invalid. If all these reasons are unjustified, then anyone left who still pirates programs is truly a thief! With no morality to hide behind, the naked truth prevails.

What most people, capitalists included, do not realize about the market system is that it is self-correcting. If there is an inequity in the system, where value received is less than value given, the system makes sure it does not remain long (if it is left to itself that is). This is as true anywhere else as in the software industry.

For years people have been pirating software because of this inequity in the system. Pirating has put many software companies out of business, and even destroyed whole markets. However, pirating is merely a symptom of what is wrong with the industry - an easy fix for an economic inequity, the market's reaction to these manufacturers' nearsighted attitudes. Until recently, however, it was not the force that threatens the destruction of the software industry as we know it - and the industry was not in as critical a situation as it now is. That is why it is now a "controversy".

Actually, this coming "shakeout", if you will, in software manufacturers is a good thing! Manufacturers of overpriced but good software are forced to lower their prices, but will survive, while manufacturers of junk go out of business as competition increases and quality goes up. Many of the software companies on the losing end of the equation whine and moan about the pirates - people who are just functioning as an oar to right the boat.

These manufacturers copy protect software, and put up other obstacles to seek to preserve their market, to delay the inevitable, but the winnowing out process can never be stopped. It is a feature of the market system as a market matures. A monopoly, or government intervention, would only serve to solidify this inequity permanently, and destroy the innovation that is the hallmark of the industry. Copy protection only serves to delay what will happen no matter what, and is self defeating in the long run.

Basically, what I am saying is that pirates are a natural response to the current situation. Innovation in this industry, combined with more useful (not necessarily more advanced) software and the ravages of pirates, will destroy those companies with poor software, while the companies with a more enlightened attitude towards the matter will survive and prosper.

A software company can almost assure success if it sees this process, and acts accordingly. If a company takes steps now, they can prosper enormously while the rest of the industry suffers through the market's version of natural selection. The best prepared companies will survive and prosper. The dinosaurs will die. Several companies already practice what I preach. I am merely an observer of an obvious fact that seems to elude people actually involved. Companies like Borland International have pointed the way for the rest of us to follow.

continued on page 10

TISHUG Shop with Cyril

I have been asked by some of our members why the club does not sell disk drives to go with the Mini-PE systems we sell. I think the main reason must be that we could not carry the warranty on these items in the same way as firms like M.D.I. of Kensington, who will repair faulty items sold by them. Also there are a variety of configurations required by the members, which would require us to carry a large stock of these items.

NOTE:- M.D.I. are selling Full Height DS/DD reconditioned drives for \$85.00 and Half Height DS/DD reconditioned drives for \$110.00. They also sell power supplies for these units.

RAM DISK CARD PARTS :-

Mostly sold out with the exception of the Battery set at .. \$14.00

THIRD RUN of RAM CARD P.C.Bs.

If the response is large enough we will have a third run of RAM cards, this will require people putting up the \$35.00 with their order. We will need at least 50 orders for boards to retain the same pricing as for the previous runs.

If we only get orders for 25 cards the price will be about \$45.00 each, but if we can increase the order to at least 50 the price should be about \$35.00 each.

At the time of writing this article we only have about six positive orders.

PRINTER BUFFER PARTS :-

- (a) P-BUFF PCB, EPROM, CRYSTAL, 8255, Z80\$56.00
- (b) 41256 memory chips (8 req'd.)\$40.00
- (c) Computer shaver board and components ..\$18.00
- (d) Printer shaver board and components ..\$18.00
- (e) Plastic box (D/S 2508) small\$ 9.00
- (f) " " (D/S 2505) large\$ 9.50
- (g) 9 volt transformer\$ 6.50

NOW THE STANDARD ITEMS:-

- (a) New DS/DD 5 1/4" Disks (box)\$12.00
- (b) T.I. Joystick handles\$ 0.50
- (c) Peter Schubert's mini-expansion units
DS/DD Disk controller card\$190.00
Mini-PE mother board (with one of either):-
32K mem : PIO : RS232 port)\$85.00
Extra options on mother board
32K memory\$50.00
RS232 port\$50.00
Finished painted box for Mini PE\$35.00

SECOND HAND ITEMS :-

- (a) Grom Ports\$12.00
- (b) Ivory Console Cases\$ 2.00

BOOKS :-

- (a) Back issues of SND\$ 1.00
- (b) Technical manual\$15.00
- (c) TI-writer manual\$15.00
- (d) Editor Assembler manual\$28.00
- (e) TI LOGO Curriculum guide\$10.00
- (f) Micropendiums\$ 3.00
1986-June to Dec./1987-Jan. to Oct.

SOFTWARE:-

- (a) Club Software Tapes\$ 3.00
- (b) Club Software Disks\$ 5.00
- (c) Picasso Publisher V2.0 (Arto Heino's) \$20.00
NOTE :- Bring or send your original copy of Picasso to the shop for an updated copy.

MODULES:-

- (a) VIATEL-1 (for serial port 1)\$25.00
- (b) VIATEL-2 (for serial port 2)\$25.00
- (c) CART WRITER\$25.00
(similar to Console-Writer)
- (d) DIAGNOSTICS\$25.00
(will test your TI gear)

NOTE:- None of these modules will operate on the 1983 Version 2.2 console, without a

NOTE:-

The price of multiple disk software has been reduced :- (that is for single density software that will not fit on one disk).

- 2 disk set....\$ 8.00
- 3 disk set....\$10.00

When ordering disk based software could you please state what type of disk drives you have (that is SS/SD, DS/SD or DS/DD).

POSTAGE:-

Please note that with all mail orders YOU have to pay postage and packaging.

If you are phoning the SHOP please note that I am NOT normally available before 7pm week days. (02)639 5847

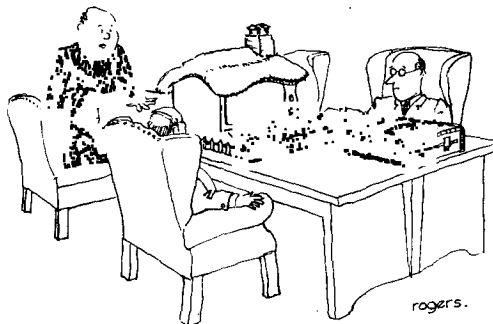
continued from page 9

What I suggest is that if you own a company, you should offer a product that you yourself could or would use, for a price that has some relation to the size of the market and to the pocketbooks of the customers. Make sure your advertising specifies who the program is for. Sell the software unprotected, and encourage buyers to make backups for their own use (cuts down on the customer relations work at least) as a sign of trust. And finally, make your software visibly professional, and readily available to anyone interested in finding it. Several software companies in the TI99/4A industry do this now. Miller's Graphics offers innovative software at an excellent price, and is very well-known. CSI Design Group offers a lot of original software at excellent prices too, though they could stand a little more publicity. As a matter of fact, a quick look through MICROpendium will give you four or five names of companies that are doing it right now.

On a more personal level; the company I work for, Asgard Software has been experiencing growth in excess of 100% per month, at a time when the market is shrinking, by simply following my own advice. Whereas four months ago we could barely afford to buy a classified ad, we are now doing well. We do not sell a single item for more than \$20.00, our software is designed to be functional for a specific audience, and our advertising reflects this fact. Our manufacturing methods are down so well that we can even make a reasonable profit selling a package like GRAPHX COMPANION for \$7.00 (and that includes postage), while other manufacturers can not even duplicate a disk for \$7.00! Any software that is manufactured in-house is not protected, while software by outside authors is protected at the author's discretion, even though we do not encourage it.

I am not saying Asgard will succeed. All I am saying is that the market is going in a definite direction, and it is in the best interest of every software manufacturer to recognize this fact, and change their operation to get the most out of it. Software is becoming big business, but now is the time when small businesses CAN become big businesses. At least until the government regulates the industry.

In conclusion, the software industry can only hold itself responsible for the growth of piracy. Survival in the dawning industry will be determined by whether you accept this fact or not.



'As you see, gentlemen, our marketing survey produced some unexpected results.'

TISHUG Software Column by Terry Phillips

Welcome back to all members. I trust everyone had an enjoyable Christmas and that so far, 1988 has been kind to you all.

Software to be available through the shop at the February meeting - assuming there is time during the afternoon to open the shop - will be the following disks and tapes:

DISK:

- 40 COLUMN UTILITIES - a very nicely programmed utility recently received from the USA. Assembly programmers, both experienced and beginner alike, should have loads of fun playing around with this one. The disk, unfortunately for those with single sided/single density systems, runs to 369 sectors, just a little bit too much to be accommodated on one disk. Therefore to suit all tastes, the disk will be released in either double sided or double density versions or on 2 disks for those who need it this way, with all the documentation files on the second disk.
- GRAPHIC LABELLER - I am told that this one has been around for some time, but until recently I had not seen it. The program allows you to print to self adhesive labels which can be used for mailing purposes or disk identification or, I suppose, any other use you can think of. There are dozens of picture files on the disk that you can add to your label. Files ending in /GR on Arto Heino's Picasso pictures disk are also compatible with this program.
- and 4. RLE PICTURES - a host of new pictures for MAX/RLE have been received, some good and some just so-so. What I will do is release these a few at a time commencing this month with 2 disks full. On one of the disks will be an updated loader for the pictures which runs out of Extended BASIC.

TAPE:

TAPE 1988/1 contains some games, educational programs and utilities. Titles are:

CALCULATOR - good screen calculating utility in Extended BASIC.

CASSETTE SLEEVE - an excellent program to produce cassette sleeves. In Extended BASIC with printer required.

CITY ATTACK - Space shooting type of game. Quite good fun to play in Extended BASIC.

FITZ - a problem solving game where you have to put the various shaped pieces into their proper places. Difficult to play. In Extended BASIC.

HARRIED HOUSEWIFE - a match up type game that has been around a long time. This version adapted to run from Extended BASIC.

ADJECTIVE/ADVERB - good educational game in Extended BASIC.

NOUN/ADJECTIVE - in a similar vein to ADJECTIVE/ADVERB program. Also in Extended BASIC.

Plus on the same tape 5 programs on an education theme written for use with a light pen. Now probably not many of you have a light pen, so maybe this will stimulate some interest in this type of software. The joystick does substitute for a light pen in some of the programs.

Now in case you missed any of the software issued through the shop during 1987, here is a list of what was available during the various months. This software is all still available through mail order if copies are not held at the shop on meeting days.

JANUARY/FEBRUARY:

Checkbook/Budget Manager - Disk
 Out on a Limb/Alphanum Delight - Disk and Tape
 Calendars - Disk
 More Surprises - Disk
 Pilot Language - Disk
 Tunnels of Doom Data Bases - Disk and Tape
 More Surprises - Tape (32K Expansion)

MARCH:

Titronic Data - Disk
 March 1987 - Disk
 Sprite Editor - Disk
 Musicprexn - Disk
 1987/3 - Tape

APRIL:

Startrek Themes - Disk
 South Pacific Volume 1 - Disk
 TI Sings - Disk
 1987/4 - Tape

MAY:

DM1000 (V3.5) - Disk
 Diskhacker - Disk
 Funnelweb (V3.4) - Disk
 1987/5 - Tape
 1987/5A - Tape

JUNE:

Keir Wells - Disk
 Marty Kroll's Cataloguer - Disk
 Utilities 1 - Disk
 Utilities 2 - Disk
 1987/6 - Tape

JULY:

Creative Filing System (V7.0) - Disk
 Assembly Utilities - Disk
 1987/7 - Tape

AUGUST:

Modules 1 (Utilities) - Disk
 Modules 2 (Education) - Disk
 Saxsaan Adventure - Disk
 German Vocabulary plus more - Disk
 1987/8 - Tape
 1987/8A - Tape

SEPTEMBER:

September - Disk
 Funplus - Disk
 Wade's Programs - Disk and Tape
 1987/9 - Tape (Wade's programs plus some others)

OCTOBER:

1987/10 - Disk and Tape (32K)

NOVEMBER:

1987/11 Financial - Disk and Tape
 1987/11A Music - Disk and Tape
 1987/11B Video Labeller - Disk
 1987/11C TI 123 - Disk

DECEMBER:

Software Catalog - Disk
 Disk Aid - Disk
 Funnelweb (V4.0) - Disk
 Channel 99/1 - Disk
 Christmas Music - Disk and Tape
 1987/12A - Tape

Generally, there was a full description in each issue of the TND as to what was on each tape or disk. If you are a newer member and do not have that particular copy, then write to me for full details.

See you at the AGM. ○



"You'll receive your software manual very soon. We're nearly finished with the final binding procedure."

Programming Music on the TI

by Jim Peterson, Tigercub Software

You do not need to know very much about programming in order to program music on the TI99/4A, but it is almost essential that you know how to read sheet music. You do not really have to be able to play any instrument, or to know anything more about music, but it does help, the sheet music will probably have several notes for each chord, and a knowledge of music will help you to decide which three of those to use.

Music is programmed using the CALL SOUND subprogram. Each CALL must have a duration and at least one frequency with a volume.

The duration is measured in milliseconds (thousandths of a second) and can be from 1 to 4250. The User's Reference Guide admits that the actual duration obtained may be off by as much as 1/60 of a second. Actually, any duration between 1 and 50 will be of the same length, and thereafter the actual duration increases in steps of about 8. This really causes no problems, because it is best to assign the desired duration of the shortest note to a variable, and then multiply that variable for longer durations. This also makes it very easy to change the tempo of the music.

When a CALL SOUND is executed, the frequencies will continue to play for the specified duration while the computer continues to execute the next statement. Thus, animated graphics can be combined with music, providing that the graphics commands can be executed before the end of the duration.

However, if the duration is given a negative value, between -1 and -4250, the CALL SOUND will be stopped and replaced by the next CALL SOUND. Thus, a CALL SOUND with a negative duration, placed within a loop, will sound continuously until the loop ends.

The frequency is given in Hertz, which is one cycle per second, and can be from 110, which is A below middle C, to a high of 44733 which is far above the range of hearing of most humans. However, the Reference Guide admits that the actual frequency may vary by as much as ten percent. In fact, the deviation can be much greater than that, because any frequency from 31953 to 43733 ends up as exactly 37287! However, the frequencies within the audible range of music are close enough.

Each frequency must be given a volume, which can be from 0 (loudest) to 30 (inaudible). Actually, there are only 15 volume levels, because the odd numbers have the same volume as the preceding even number.

The CALL SOUND may optionally contain one to three frequencies, each with its own volume, and may also optionally contain a "noise", also with a volume. The "noise" may be a negative number from -1 to -8; of these, -1 to -4 are called "periodic noise" and -5 to -8 are called "white noise". The User's Reference Guide mentions that the -4 and -8 noises vary with the frequency of the third tone specified, that is the only clue we were given that bass notes are possible.

To create a bass note, the CALL SOUND must contain three frequencies and a -4 noise. The first two frequencies may be musical notes, or frequencies above audible range if only the bass is wanted, and the volume may be audible or inaudible. The third frequency will be sounded in the -4 noise, and will be two octaves and a 7th below its actual frequency, therefore the 3rd frequency is given an inaudible volume of 30 unless the actual frequency can be used in a 7th chord.

The easiest way to program bass notes is to multiply the frequency by 3.75 in the 3rd voice. This will lower it by two octaves.

The formula to create a musical scale of 4 octaves, starting with frequency F, is DIM N(48):: F=110 :: FOR J=1 TO 48 :: N(J)=INT(F*1.059463094*(J-1)):: NEXT J

Once this has been executed, music can be programmed using the array subscripts, such as CALL SOUND(100,N(1),0,N(5),0)

However, most music programmers prefer to assign the frequency values to mnemonic variables, such as A1=110 for A of 1st octave, C2S=277 for C sharp of 2nd octave, etc. Music is then programmed by CALL SOUND(100,A2S,0) etc.

An even better method, perhaps, is to set up the array, as above, and then assign its values to mnemonic variables, such as A1=N(1). This makes it possible to change the key of a composition by simply changing the value of F.

Most musical compositions consist at least partly of repetitive phrases. The first thing to do when programming from sheet music is to go through the score and mark each series, of more than a few notes, which is repeated at least once within the score. Then, program each of these phrases as a GOSUB.

The simplest method of music programming consists of nothing but a long list of CALL SOUNDS. This permits very fast music to be played, but it is tedious to program and takes up much memory.

Another method sometimes used is to place all the notes in DATA statements and then use a loop to READ them and play them in a single CALL SOUND, such as FOR J=1 TO 800 :: READ A,B,C,D :: CALL SOUND(A,B,0,C,0,D,0):: NEXT J

This method has limitations of speed, because DATA is read rather slowly, and is difficult to debug.

Some programmers get around the speed limitation by reading all the notes into an array, either from DATA statements or from a separate disk file, and then play them in a loop: FOR J=1 TO 1000 STEP 4 :: CALL SOUND(A(J),A(J+1),0,A(J+2),0,A(J+3),0):: NEXT J

This method can take as much as three minutes to initialize before the music begins, and is very difficult to debug.

The best method of all, first used by Sam Moore Jr., and refined by Bill Knecht and others, uses the mnemonic variables, and GOSUBs to a single CALL SOUND (or to one of 4 CALL SOUNDS, to play 1 to 4 notes), with the variables in the CALL SOUND being redefined from the mnemonic variables before each GOSUB. Since the GOSUB is executed very rapidly, there is no limitation on speed, and the variables need be specified only when they change. For instance, the GOSUB might be

```
1000 CALL SOUND(D*T,N1,V1,N2,V2,N3,V3) :: RETURN,
with D having been predefined as the basic shortest duration. Then, a program line might read 100 T=1 :: V1,V2,V3=0 :: N1=C3 :: N2=G2 :: N3=E1 :: GOSUB 1000 :: N2=E2 :: N3=G1 :: GOSUB 1000 :: N2=C2 :: N3=E1 :: GOSUB 1000 ..etc. to play a continuous melody note with a pattern of accompaniment notes.
```

As a variation of the above, the GOSUB may be to a loop which plays the chord with a negative duration and a decreasing volume, as FOR V=0 TO 30 STEP 5 :: CALL SOUND(-99,N1,V,N2,V,N3,V):: NEXT V :: RETURN

This gives the "decay", or fading volume, which is typical of piano music or other music where a string is struck or plucked. The slowness of loop execution places some limitations on the use of this technique in Extended Basic programming, but it has great potential in assembly, FORTH, or compiled Basic.

Many other effects are possible. Alternating two CALL SOUNDS in a loop, with the frequency in the second being a multiple of 1.01 to 1.03 times the first, gives a tremolo effect: FOR J =1 TO T :: CALL SOUND(-99,N1,V1):: CALL SOUND(-99,1.03*N1,V1):: NEXT J

Alternating the volume rather than the frequency, or both, can give other tremolo effects.

Two notes in one CALL SOUND, with the second being multiplied by about 1.03 over the first, gives a richer sound: CALL SOUND(100,N1,V1,N1*1.03,V1)

Extending the -4 note into the melody range gives a reedy effect: CALL SOUND(100,N1,V1,N2,V2,3* 3.75*N3,30,-4,V3)

TI music tends to have too strong a pulse, because the notes start and stop so abruptly. Therefore, it is better to play the preceding note through a rest, perhaps at a decreased volume, and the last note of a piece should always be faded out through a loop of decreasing volume.

Tidbits Seven

by Wade Bowmer, TISHUG

Welcome! to Tidbits Seven! If I may be so bold, I'd like to point out that the festive season just passed is *Christmas*, not *Xmas*. I am a Christian, and shortening "Christ" to "X" is something I find degrading. *XBASIC*, but not *Xmas*.

That Wonderful ACCEPT!

First up is **ACCEPT**. This versatile statement is so powerful, that it has a few bugs. (But of *course* !)

Name: Accept

Type: I/O Statement

Format: **ACCEPT** [*option list*] *variable*

Purpose: To allow user input anywhere on-screen, independent of scrolling.

Parameters: One variable can be supplied only, either numeric or string. However, an array element with a calculated subscript will cause erratic behaviour.

Discussion: **ACCEPT** provides an alternative to **INPUT** or **LINPUT**. It cannot be used to obtain information from files, however. Without any options, **ACCEPT A\$** corresponds to **LINPUT A\$**; only **ACCEPT** does not beep.

Options:

AT (*row, column*) sets the position for **ACCEPT**ing data. *row* and *column* are both numeric expressions. However, *column* has a limited range of 1 to 28 (corresponding to **TAB** columns and to columns 3 to 30 in **HCHAR**, **VCHAR** and **GCHAR** subprograms), with anything else producing an error (* **BAD VALUE**). *row* has limits of 1 to 24, but anything less than 1 is the same as 1, while anything greater than 24 is reduced by 24 repeatedly until the resulting number is within range. If this option is omitted, the characters in columns 31 and 32 on row 24, on which **ACCEPT**ance subsequently takes place, are changed to edge characters, code 31.

BEEP causes a tone identical to that of **INPUT** and **LINPUT**. It is the same as **CALL SOUND(-200,1397, 2)**.

ERASE ALL will clear the screen (place the space character in all positions) prior to input.

SIZE (*numeric-expression*) sets up a window in which input is **ACCEPT**ed. If the *numeric-expression* + *column* in the **AT** option exceeds 29, the rest of the line is used for input, as when this option is not used. The window used is actually cleared unless the *numeric-expression* is negative, which allows a default value to be present.

VALIDATE (*parameter,...*) sets up keystrokes that are allowed. A disagreeable tone sounds when a disallowed key is pressed.

Parameters:

UALPHA permits all uppercase letters, **A>Z**.

DIGIT permits only numbers, **0>9**.

NUMERIC permits **0>9**, a decimal ".", + - and "E".

string-expression permits those characters found in the evaluated *string-expression*.

Now to the fun part: Quirks.

1. Array variables with a calculated subscript will cause erratic behaviour as follows: **ACCEPT** will not work at all should the **VALIDATE** option be missing. If **VALIDATE** is present however, then the **SIZE** window (should the **SIZE** option be present) is disabled, and input spills over to several lines, ignoring Extended BASIC's "margins".

This is *definitely a bug!* so avoid it *totally*. The best solution is to use a temporary variable. A string variable is preferable, with numbers being converted with the **VAL** function.

2. A predefined window (using a negative expression in the **SIZE** option) will not be checked for permitted characters specified in the **VALIDATE** clause. This however, is particularly easy to circumvent by simply setting up the window beforehand with the permitted characters.
3. The predefined window can be cleared using **FCTN-3/SHIFT-T** (**ERASE**) or **FCTN-1/SHIFT-F** (**DEL**), which will cause problems should the blank be entered into a numeric variable. The solution: use a temporary string variable instead and check for a null string before using **VAL**.
4. The most recent keycode in the **KEY** subprogram affects the key unit used to scan the keyboard for **ACCEPT**. This ultimately permits the keyboard mode used to be selected by the programmer.
5. The key used to terminate **ACCEPT**, either **ENTER**, **FCTN-X** or **FCTN-E**, can be read immediately afterwards with **CALL KEY**. This has application in Extended BASIC Text Editors, or when input needs to be more than one line long.

A Problem with SUB.

In my very first article, I mentioned the trick of redefining the standard subprograms. If you define a subprogram with the same name as one of the standard ones, then it will not produce an error. But you then cannot use the original, pre-defined subprogram:

How about an example?

```
10000 SUB CLEAR(CLR) :: IF ABS(CLR) > 16 THEN
SUBEXIT ELSE IF CLR=0 THEN
10010 CALL SCREEN(ABS(CLR)) :: IF CLR<0 THEN SUBEXIT
10020 DISPLAY ERASE ALL :: SUBEND
```

I remember including this example way back then. But it still works. I have another idea.

```
10000 SUB CLEAR :: SUBEND
10010 SUB SCREEN :: SUBEND
10020 SUB KEY :: SUBEND
```

And so on. You can see that by redefining them all like this means that you can't use them! How's that for a programming joke!

To de-define them, you have to **SAVE** the program, **OLD** or **RUN** another one, edit the program or **RUN** the program *without* the redefinitions.

Fascinating, isn't it?

Praiseworthiness.

Peter Schubert will remember that in September I borrowed a PIO interface. The reason was quite simple: I absolutely had to hang a printer off my TI, and didn't have an interface.

Why did I suddenly and temporarily need a printer? Well, I am in Year 11 at Inaburra High School, and some organization in Queensland was running a computer competition for High School Students. Two of my friends and I entered, and I toted my TI-99/4A to school one day, to use it to enter the competition. (There were two other groups; one used a Crummydore 64, and the other used one of the school's IBM PCs.)

As for the display, I borrowed one of the school's videomonitors, and did likewise for the printer. And the interface worked like a beauty!

We completed four of the five questions, complete with a runtime printout, and began the fifth, but ran out of time. Out of all the participants in NSW and Queensland, 40 certificates were issued, and *my group received one!*

So, what do you do with your **TI-99/4A**?

continued on page 14

FORTRAN

by D. N. Harris

The BASIC the Texas Instruments 99/4a uses is in fact almost PURE ANSI BASIC which is VERY CLOSE TO FORTRAN.

The differences between the two languages are in the number of built-in functions, which in FORTRAN have to be defined, but have default values in BASIC, and in the formatting of lines, which in FORTRAN must be tightly, rigidly, exactly, inflexibly, COLUMN-COUNTED at the LEFT MARGIN. So as not to give the impression that FORTRAN is impossible, I am going in this article to cover REM STATEMENTS which in FORTRAN are known as C STATEMENTS where C stands for COMMENT.

In any carefully written BASIC programme, you of course put your name and the date, and ask for a million dollars freeware from anybody who might appreciate the programme. Fields of asterisks are used as underliners so that other people can skip the stuff about how the programme is dedicated and try to find the bug you left in, causing it to crash when a loop variable over-incremented and you left no bug-trapping to stop the thing, and the programme tried to set the screen to say colour 29.

In FORTRAN the thing would be set up;

```
C *****
C *****D.N.HARRIS*****
C *****9th NOVEMBER 1987*****
C *****EXAMPLE FORTRAN PROGRAMME*****
C *****SUGGEST $1,000,000 FREWARE*****
C *****
__9999_END
```

When you want to something other than make a comment, FORTRAN is waiting to snow you with error messages. It is bloody-minded about commas and full-stops, and insist that there be the correct number of spaces, so the C must be the FIRST character on the line, unless you want to use a 5-digit line number. It is fairly safe to leave blank this space. The next columns can be left blank also, if you want to start at low numbers of line numbers,

```
__1 END <---THEN A SPACE MUST BE LEFT AFTER THE
LINE NUMBER BEFORE THE COMMAND.
```

Another way to write REMS is to put an Exclamation mark and some Asterisks after the command, as in Extended BASIC.

```
__9 END !*****END OF PROGRAMME.
```

In this first lesson, REMS IN FORTRAN, I have shown that there is already one thing you can do, WRITE C STATEMENTS. Later I will show the way FORTRAN handles strings. That will enable you to put comments into FORTRAN and get it to say "Hello!" or "Don't forget to put a decimal point at the end of the integral part of the number"

THE SEVENTH COLUMN LEFT BLANK IN THE ABOVE MIGHT HAVE A NUMBER IN IT FOR FIELDS GREATER THAN 80 COLUMNS ON A FORTRAN CARD.

THE SIXTH COLUMN CAN HAVE A NUMBER FOR CONTINUATION.

```
__91<---MEANS THIS IS CONTINUED ON CARD 1.
```

FORTTRAN is THE LANGUAGE FOR ROBOTS AND TELEMETRY although it can be learned and made to serve like BASIC, and if a well-written programme in BASIC or in FORTRAN is running, the only thing you might notice is

the FORTRAN is quicker. If I write an emulator using EXTENDED BASIC a subset FORTRAN having 28-6-22 characters will be possible, and the continuation option will have to be used an awful lot, and perchance there will be some feedback by the members about the demand for language emulators. These can be loaded from Cassette and will give student practice in writing FORTRAN, and perchance COBOL,PASCAL,RPGII,C, and various other languages. Perchance the demand for more will result. One ought to be able to plug a cartridge in and run FORTRAN, and the lack of such a cartridge may be the consequence of lack of user demand. It is certainly available for the Microbee computer, and the Atari users have a FORTRAN.

Programs that Write Programs

part 5

by Jim Peterson

In addition to writing programs in MERGE format, the same techniques can be used to analyze or modify programs which have been SAVED in MERGE format. The D/V 163 file editor in Part 2 of this series was an example.

Here is a simple program to remove REM statements -

```
100 DISPLAY AT(3,5)ERASE ALL
:"REM REMOVER": : : "Program
must be SAVED in": "MERGE for
mat by": "SAVE DSK(filename),
MERGE"
110 DISPLAY AT(12,1): "FILENA
ME? DSK" : : ACCEPT AT(12,14)
:F$ : : DISPLAY AT(14,1): "NEW
FILENAME? DSK" : : ACCEPT AT
(14,18): NF$
120 OPEN #1: "DSK"&F$, VARIABL
E 163, INPUT : : OPEN #2: "DSK"
&NF$, VARIABLE 163, OUTPUT
130 LINPUT #1: M$ : : A=POS(M$
, CHR$(131), 1) : : B=POS(M$, CHR
$(154), 1) : : A=MAX(A, B) : : IF
A=3 THEN 150 : : IF A=0 THEN
PRINT #2: M$ : : GOTO 150
140 PRINT #2: SEG$(M$, 1, A-1) &
CHR$(0)
150 IF EOF(1) <> 1 THEN 130 : :
CLOSE #1 : : PRINT #2: CHR$(2
55) & CHR$(255) : : CLOSE #2
```

The REM statement will begin with either a !, which is CHR\$(131), or REM which is CHR\$(154). So, line 130 reads in the lines one at a time. A finds the position in the line of ! and B finds the position of REM; one or the other, or both, will not be present and will equal 0. Then MAX finds the larger of A and B, which will be whichever one is present, or 0 if neither.

If ! or REM is in the 3rd position, immediately after the 2-byte line number, we want to delete the line entirely, so we do not reprint it. If A=0 then neither ! nor REM is present, so we reprint the entire line in the new file.

Otherwise, the REM statement is obviously a tail remark, so we reprint to the new file the segment of it starting with the first character and consisting of the number of characters one less than the position of the ! or REM. And, since we have lopped off the end of the line, we do not forget to replace the end-of-line marker CHR\$(0).

If we have not reached the end of the file, we go back for the next line. Otherwise, we close the old file, but we remember to add the end-of-file marker to the new file before we close that too.

continued from page 13



A first lesson in X/B programming

by Jim Peterson, Tigercub Software

Extended BASIC is nothing more than BASIC with a lot more words added. If you have learned anything about BASIC programming, it will also apply to Extended BASIC.

A PROGRAM is just a numbered series of instructions to the computer, written in more or less plain English, telling the computer to perform a certain task. The computer will follow these instructions in the order they are numbered, except when you tell it to GOTO or GOSUB to another part of the program.

The instructions are ordered by LINE NUMBERS. You can type these in, but it is easier to just start out by typing NUM and Enter. The computer will then automatically give you line numbers starting with 100 and advancing by 10 to 110, 120, etc. This is so that you can later squeeze more instructions in between, using 105, etc. If you need to get out of automatic numbering, in order to correct a line or insert a line, just press Enter twice. To start automatic numbering again, just type NUM, space, and the next line number you want, such as NUM 130.

In Extended BASIC, you can put several instructions under one line number, by putting a double colon (:) between them. But, while you are still learning, please DO NOT! Why not? Well, when you tell the computer to do something it can not do, or can not understand, it will give you an ERROR message, either when you Enter the line or when you run the program, and it will tell you the line number that is causing the problem - but if you have several instructions under that line number, you will not know which one is wrong!

The first instruction we will learn is PRINT. This tells the computer to print something on the bottom line of the screen, and then scroll up one line. Try entering NUM, then:

```
100 PRINT 1
110 PRINT 2
120 PRINT 3
```

Then enter:

```
100 PRINT A
110 PRINT B
120 PRINT C - and RUN it.
```

It printed a 0 three times, did it not? Why? When you tell the printer to print anything other than a numeric digit (or a math symbol or decimal combined with a number) it thinks that you are telling it to print the VALUE of a VARIABLE. And if you have not previously told it otherwise, that value is zero. Try this:

```
100 A=10
110 PRINT A
```

So what is a VARIABLE? If you suffered through high school algebra, you may recall equations such as; $S \times T = D$, where S equals speed and T equals time and D equals distance. You could give S and T any values you wanted to, in order to calculate how far something would go at a certain speed in a certain time. T and S and D are VARIABLES. We use them a great deal in programming and you will soon see why.

Now, suppose you really wanted to print the letter A. That's easy, just put it in quotation marks and the computer will know what to do.

```
100 PRINT "A"
```

In either BASIC or Extended BASIC, the instruction DISPLAY works just like PRINT.

```
100 DISPLAY 999
110 DISPLAY "HELLO"
```

Text scrolling up from the bottom looks rather "cheap", compared to those computers which display text from the top of the screen downward. In Extended BASIC we can put the display wherever we want by using DISPLAY AT followed by a row and column number in parentheses. There are 24 rows on the screen and 24 columns when you are using PRINT or DISPLAY.

```
100 CALL CLEAR
110 DISPLAY AT(1,1):
120 DISPLAY AT(24,1):24
130 GOTO 130
```

We slipped in a couple of new instructions there. CALL CLEAR just erases everything on the screen (actually it fills the screen with the blank space you get by hitting the space bar). GOTO tells the computer to go to another line number. In this case, it goes back to itself over and over and keeps the program running so it will not print READY and scroll that first line off the screen. Use FCTN(4) to get out of it. Try experimenting with DISPLAY AT to put different numbers, words or phrases wherever you want them on the screen. You will find that if you specify a row number greater than 24, the computer will just subtract 24 until it gets down to a number within range.

```
100 DISPLAY AT(25,35):"WHERE?"
```

In some programs you may see PRINT followed by # and a number or variable. This is an instruction to print to a printer, to a disk, a speech synthesizer, or whatever. Actually you can print to the screen by:

```
100 PRINT #0:"SEE?"
```

but there is usually no reason to do so. Now, a few words about print separators. Try this:

```
100 PRINT 1:2:3
110 PRINT 1,2,3
120 PRINT 1;2;3
130 PRINT "A";"B";"C"
```

See what happens? The colon (:) causes the computer to skip to the beginning of the next line before printing again. The comma (,) causes it to jump half the width of the screen before printing again. And the semicolon causes it to print one item right after another, EXCEPT that numbers are always printed with a blank space before and after them (a negative number has a minus sign (-) instead of a blank before it). Now try:

```
100 PRINT 1:2:3:
110 PRINT 1,2,3,
120 PRINT 4
130 PRINT 1;2;3;
140 PRINT 4
```

The colon after the 3 in line 100 was useless because the computer would advance to the next line anyway. The comma after the 3 in line 110 caused line 120 to print half a screen width after the 3. And the semi-colon after the 3 in line 130 caused line 140 to print immediately after the 3. Do a lot of experimenting with this, until you know just what the print separators will do. Try:

```
100 PRINT "A","BCDEFGHIJKLMNOPQ"
```

See what happens when the item after the comma is more than half a screen in length? Now try this:

```
100 PRINT 1: : : 2
```

And try this:

```
100 CALL HCHAR(1,1,42,768)
110 DISPLAY AT(5,1):"TEST"
120 DISPLAY AT(7,1)SIZE(4):"TEST"
130 DISPLAY AT(9,1):"TEST";
```

Line 100 just fills up the screen with something so I could show you that DISPLAY AT erases the remainder of the line unless you specify the length of what you will print with SIZE or, much easier, put a semicolon after the text.





TISHUG NEWS DIGEST

```

100 !@P-
110 ! *****
120 ! * *
130 ! * BANNER *
140 ! * *
150 ! * BY *
160 ! * *
170 ! * A SAUNDERS *
180 ! * *
190 ! * TIBUG *
200 ! * *
210 ! *****
220 !
230 !@P+
240 GOTO 250 :: CALL CHARSET
:: MESSAGE$,ANS$,C$,D$,F$,G
$,H$,ONE$,TWO$,THREE$,FOUR$,
FIVE$
250 CALL CLEAR :: CALL CHAR(
48,"00000000E0BFBF00E000
0000000000000068FEFE0000
0000000000")
260 CALL MAGNIFY(4):: CALL S
CREEN(14):: CALL SPRITE(#1,4
8,5,25,256,0,-10):: CALL SOU
ND(4250,-3,15)
270 CALL CHARPAT(65,A$):: CA
LL CHARPAT(66,B$):: CALL CHA
RPAT(78,N$):: CALL CHARPAT(6
9,E$):: CALL CHARPAT(82,R$)
280 S$=RPT$("O",16):: CALL C
HAR(36,B$&S$&A$&S$):: CALL C
HAR(40,N$&S$&N$&S$):: CALL C
HAR(44,E$&S$&R$&S$)
290 X=36 :: FOR I=2 TO 4 ::
CALL SPRITE(#I,X,5,25,256,0,
-10):: FOR D=1 TO 253 :: NEX
T D :: X=X+4 :: NEXT I :: CA
LL SOUND(2500,-3,15)
300 CALL POSITION(#2,A,B)::
IF B<94 THEN CALL MOTION(#2,
0,0):: CALL MOTION(#3,0,0)::
CALL MOTION(#4,0,0)ELSE 300
:: CALL DELAY(500)
310 DISPLAY AT(12,15):"BY" :
: CALL DELAY(300):: DISPLAY
AT(18,11):"A SAUNDERS" :: CA
LL DELSPRITE(#1):: CALL DELA
Y(300)
320 !@P-
330 DISPLAY AT(23,13):"TIBUG
" :: CALL DELAY(600):: CALL
DELSPRITE(ALL):: CALL CLEAR
:: CALL CHARSET :: OPEN #1:"
PIO" :: PRINT #1:CHR$(27);"O
"
340 DISPLAY AT(2,1)ERASE ALL
:"TYPE IN YOUR MESSAGE SUCH
AS 'MERRY CHRISTMAS EVERYBODY
' AND WATCH ME PRINT THE
MESSAGE IN"
350 DISPLAY AT(5,12):"LARGE
LETTERS ON" :: DISPLAY AT(6,
1):"YOUR PRINTER." :: DISPLA
Y AT(9,1):"USE UPPERCASE LET
TERS ONLY PLEASE."
360 DISPLAY AT(13,1):"138 CH
ARACTERS MAXIMUM." :: INPUT
MESSAGE$ :: FOR I=1 TO LEN(M
ESSAGE$):: A=ASC(SEG$(MESSAG
E$,I,1))
370 IF A=32 THEN ONE$,TWO$,T
HREE$,FOUR$,FIVE$=RPT$(" ",1
2):: GOTO 410
380 A$=RPT$(CHR$(A),42):: B$
=RPT$(CHR$(A),30):: C$=RPT$(
CHR$(A),6):: D$=RPT$(CHR$(A)
,12)
390 E$=RPT$(" ",12):: F$=RPT
$(" ",16):: G$=RPT$(" ",18)::
H$=RPT$(" ",6)

```

```

400 ON A-64 GOSUB 430,440,45
0,460,470,480,490,500,510,52
0,530,540,550,560,570,580,59
0,600,610,620,630,640,650,66
0,670,680
410 CALL PRINTER(ONE$):: CAL
L PRINTER(TWO$):: CALL PRINT
ER(THREE$):: CALL PRINTER(FO
UR$):: CALL PRINTER(FIVE$)::
PRINT #1:CHR$(10):: NEXT I
420 DISPLAY AT(12,4)ERASE AL
L:"ANOTHER BANNER? N" :: ACC
EPT AT(12,20)SIZE(-1):ANS$ :
: IF ANS$="Y" THEN 340 ELSE
CALL CLEAR :: CLOSE #1 :: EN
D
430 ONE$=F$&B$ :: TWO$=F$&E$
&C$&E$&C$ :: THREE$=F$&E$&C$
&G$&C$ :: FOUR$=TWO$ :: FIVE
$=ONE$ :: RETURN :: !A
440 ONE$=F$&A$ :: TWO$=F$&C$
&E$&C$&E$&C$ :: THREE$=TWO$
:: FIVE$=F$&H$&D$&H$&D$ :: F
OUR$=TWO$ :: RETURN :: !B
450 ONE$=F$&H$&B$ :: TWO$=F$
&C$&E$&G$&C$ :: THREE$=TWO$
:: FOUR$=TWO$ :: FIVE$=F$&H$
&B$ :: RETURN :: !D
470 ONE$=F$&A$ :: TWO$=F$&C$
&E$&C$&E$&C$ :: THREE$=TWO$
:: FOUR$=TWO$ :: FIVE$=TWO$
:: RETURN :: !E
480 ONE$=F$&A$ :: TWO$=F$&G$
&C$&E$&C$ :: THREE$=TWO$ ::
FOUR$=TWO$ :: FIVE$=TWO$ ::
RETURN :: !F
490 ONE$=F$&H$&B$ :: TWO$=F$
&C$&E$&G$&C$ :: THREE$=F$&C$
&H$&C$&G$&C$ :: FOUR$=THREE$
:: FIVE$=F$&H$&C$&G$&C$ ::
RETURN :: !G
500 ONE$=F$&A$ :: TWO$=F$&G$
&C$ :: THREE$=TWO$ :: FOUR$=
TWO$ :: FIVE$=ONE$ :: RETURN
:: !H
510 ONE$=F$&G$&G$&H$ :: TWO$
=F$&C$&E$&G$&C$ :: THREE$=F$
&A$ :: FOUR$=TWO$ :: FIVE$=O
NE$ :: RETURN :: !I
520 ONE$=F$&H$&C$ :: TWO$=F$
&C$ :: THREE$=F$&C$&E$&G$&C$
:: FOUR$=F$&A$ :: FIVE$=F$&
G$&G$&C$ :: RETURN :: !J
530 ONE$=F$&A$ :: TWO$=F$&G$
&C$ :: THREE$=F$&E$&C$&D$ ::
FOUR$=F$&H$&C$&G$&C$ :: FIV
E$=F$&C$&E$&G$&C$ :: RETURN
:: !K
540 ONE$=F$&A$ :: TWO$=F$&C$
:: THREE$=TWO$ :: FOUR$=TWO
$ :: FIVE$=TWO$ :: RETURN ::
!L
550 ONE$=F$&A$ :: TWO$=F$&E$
&G$&C$ :: THREE$=F$&G$&D$ ::
FOUR$=TWO$ :: FIVE$=ONE$ ::
RETURN :: !M
560 ONE$=F$&A$ :: TWO$=F$&G$
&H$&D$ :: THREE$=F$&G$&C$ ::
FOUR$=F$&H$&D$ :: FIVE$=ONE
$ :: RETURN :: !N
570 ONE$=F$&H$&B$ :: TWO$=F$
&C$&E$&G$&C$ :: THREE$=TWO$
:: FOUR$=TWO$ :: FIVE$=ONE$
:: RETURN :: !O
580 ONE$=F$&A$ :: TWO$=F$&G$
&C$&E$&C$ :: THREE$=TWO$ ::
FOUR$=TWO$ :: FIVE$=F$&G$&H$
&D$ :: RETURN :: !P

```

```

590 ONE$=F$&H$&B$ :: TWO$=F$
&C$&E$&G$&C$ :: THREE$=TWO$
:: FOUR$=F$&D$&H$&G$&C$ :: F
IVE$=ONE$ :: RETURN :: !Q
600 ONE$=F$&A$ :: TWO$=F$&G$
&C$&E$&C$ :: THREE$=F$&E$&D$
&E$&C$ :: FOUR$=F$&H$&C$&H$&
C$&E$&C$ :: FIVE$=F$&C$&G$&D
$ :: RETURN :: !R
610 ONE$=F$&H$&C$&E$&D$ :: T
WO$=F$&C$&E$&C$&E$&C$ :: THR
EE$=TWO$ :: FOUR$=TWO$ :: FI
VE$=F$&H$&D$&E$&C$ :: RETURN
:: !S
620 ONE$=F$&G$&G$&C$ :: TWO$
=ONE$ :: THREE$=F$&A$ :: FOU
R$=ONE$ :: FIVE$=ONE$ :: RET
URN :: !T
630 ONE$=F$&H$&B$&C$ :: TWO$
=F$&C$ :: THREE$=TWO$ :: FOU
R$=TWO$ :: FIVE$=ONE$ :: RET
URN :: !U
640 ONE$=F$&E$&G$&D$ :: TWO$
=F$&E$&D$&C$ :: THREE$=F$&D$
:: FOUR$=TWO$ :: FIVE$=ONE$
:: RETURN :: !V
650 ONE$=F$&A$ :: TWO$=F$&H$
&C$ :: THREE$=F$&E$&D$ :: FOU
R$=TWO$ :: FIVE$=ONE$ :: RE
TURN :: !W
660 ONE$=F$&C$&G$&E$&C$ :: T
WO$=F$&H$&D$&H$&D$ :: THREE$
=F$&G$&C$ :: FOUR$=TWO$ :: F
IVE$=ONE$ :: RETURN :: !X
670 ONE$=F$&E$&C$&D$ :: TWO$
=F$&G$&H$&C$ :: THREE$=F$&D$
&D$ :: FOUR$=TWO$ :: FIVE$=O
NE$ :: RETURN :: !Y
680 ONE$=F$&D$&G$&H$&C$ :: T
WO$=F$&C$&H$&C$&G$&C$ :: THR
EE$=F$&C$&E$&C$&E$&C$ :: FOU
R$=F$&C$&G$&C$&H$&C$ :: FIVE
$=F$&C$&H$&G$&D$ :: RETURN :
: !Z
690 !@P+
700 SUB PRINTER(A$):: FOR I=
1 TO 3 :: PRINT #1:A$ :: NEX
T I :: SUBEND
710 SUB DELAY(A$):: FOR D=1 T
O A :: NEXT D :: SUBEND

```

```

10 ! LEARNING MUSIC NOTE
15 ! COLIN LEE/S'PORE(0512)/
14,1,84
20 ! WRITTEN SPECIALLY FOR A
LL TIUP GROUP MEMBERS
22 ! XBASIC ; 11800 BYTES FR
EE
25 RANDOMIZE
30 CALL CLEAR
35 CALL MAGNIFY(4)
40 CALL SCREEN(8)
45 CALL CHAR(96,"FFFF")
50 CALL CHAR(104,"0000000000
00000003070F1F1F1F0F06008080
8080808080808080808000000")
55 CALL CHAR(112,"5050505050
505050")
60 GOSUB 310
65 CALL CLEAR
70 FOR I=8 TO 15 :: DISPLAY
AT(I,5):"LEARNING MUSIC NOTE
S" :: CALL SOUND(100,880,2):
: DISPLAY AT(I-1,5):"" :: NE
XT I

```

```

75 FOR R=6 TO 14 STEP 2
80 CALL HCHAR(R,1,96,32)
85 NEXT R
90 FOR I=30 TO 0 STEP -.8 ::
  CALL SOUND(100,110,I,-6,I):
  : NEXT I
95 FOR I=25 TO 1 STEP -.8 ::
  CALL SOUND(-100,I*30+110,0)
  : NEXT I
100 ! SPRITE POSITIONING
105 RRD=INT(RND*12)+1
110 RQ=RRD*8
115 CALL SPRITE(#1,104,2,RQ,
10,0,5)
120 IF RRD>7 THEN 170
125 NOTE=72-RRD
130 IF RRD=1 THEN FR=784
135 IF RRD=2 THEN FR=698
140 IF RRD=3 THEN FR=659
145 IF RRD=4 THEN FR=587
150 IF RRD=5 THEN FR=523
155 IF RRD=6 THEN FR=494
160 IF RRD=7 THEN FR=440
165 GOTO 210
170 ! BELOW C
175 NOTE=79-RRD
180 IF RRD=8 THEN FR=392
185 IF RRD=9 THEN FR=349
190 IF RRD=10 THEN FR=330
195 IF RRD=11 THEN FR=294
200 IF RRD=12 THEN FR=262
205 !MAIN LOOP
210 DISPLAY AT(3,5):"WHAT NO
TE IS THIS?"
215 CALL SOUND(500,FR,2)
220 CALL KEY(0,KK,S)
225 IF S=0 THEN 220
230 IF KK=NOTE THEN 255
235 DISPLAY AT(3,4):"THAT WA
S THE NOTE" :: CALL HCHAR(3,
25,NOTE)
240 CALL SOUND(1000,FR,0)
245 CALL DELSPRITE(#1):: CAL
L LD :: CALL ER
250 GOTO 105
255 ! CORRECT NOTE
260 DISPLAY AT(3,5):""
265 DISPLAY AT(20,7):"*****
IGHT*****"
270 DISPLAY AT(24,7):"*****
*****"
275 DISPLAY AT(22,7):"***WEL
L DONE***"
280 FOR I=1 TO RRD :: CALL C
OLOR(#1,16):: CALL SOUND(100
,770,5)
285 CALL COLOR(#1,2):: CALL
SOUND(100,990,5):: NEXT I
290 CALL DELSPRITE(#1):: DIS
PLAY AT(20,1):""
295 DISPLAY AT(22,1):"" :: D
ISPLAY AT(24,1):""
300 GOTO 105
305 ! INTRO
310 DISPLAY AT(8,4):"LEARNIN
G MUSIC NOTES"
315 DISPLAY AT(14,4):"PRESS
THE RIGHT KEY FOR THE" :: DI
SPLAY AT(17,2):"NOTE SHOWN O
N THE SCREEN"
320 DISPLAY AT(24,3):"<PRESS
ANY KEY TO BEGIN>"
325 CALL KEY(0,K,S):: IF S=0
THEN 325 ELSE RETURN
330 SUB ER
335 DISPLAY AT(3,1)SIZE(30):
""
340 SUBEND
345 SUB LD
350 FOR I=1 TO 500 :: NEXT I
355 SUBEND

```

```

100 REM CHANNEL 99 HAMILTON
USER'S GROUP
110 REM MOUSE SQUISH
120 REM A6-80047-JGU
130 REM *****
140 REM * W.M.JOHNSON *
150 REM *****
160 DIM WORD$(26)
170 DATA "ARE THE FINEST","A
RE THE BEST BUT ONE","ARE PR
ETTY GOOD","ARE FAIRLY GOOD
","ARE SO SO"
180 DATA "I WOULD RATHER NOT
SAY","ARE SORT OF POOR","AR
E VERY POOR","ARE LOUSY","ST
INK"
190 DATA "STAND ROTTING IN Y
OUR SOCKS","SHOULD BE SHOT",
"SHOULD BE HANGED","ARE DEVO
ID OF HOPE"
200 DATA "SHOULD GARGLE FIRS
T","COULD NOT CATCH FLYS","A
RE A MALLET LOVER","SHOULD L
AY OFF DRINK"
210 DATA "SHOULD TRY KICKING
IT","ARE A MOUSE LOVER","AR
E KIND TO ANIMALS","SHOULD N
OT TRY AGAIN","SHOULD GIVE U
P"
220 DATA "SHOULD LET SOMEONE
ELS PLAY","SHOULD TRY HEAD
STANDS","SHOULD TAKE UP GOLF
PLEASE"
230 FOR WD=0 TO 25 :: READ W
ORD$(WD):: NEXT WD
240 CALL CLEAR :: CALL SCREE
N(2):: OFF=50 :: CALL MAGNIF
Y(3)
250 FOR S=9 TO 14 :: CALL CO
LOR(S,1,1):: NEXT S :: FOR C
=3 TO 8 :: CALL COLOR(C,16,1
):: NEXT C
260 RANDOMIZE
270 PRINT TAB(15);"xy":TAB(1
5);"z| hi":TAB(17);" |pq":TAB
(18);"rs":TAB(18);"tu":TAB(1
8);"j"
280 FOR I=1 TO 10 :: PRINT :
: NEXT I
290 CALL CHAR(96,"FEFEFE00EF
EFEF"):: CALL CHAR(120,"0307
0F1F3F7FFFF0080C0E0F0F8F0E0
7F3F1F0E0400000C0C0E070381C
0E07")
300 CALL CHAR(124,"COA090482
4120905"):: CALL CHAR(128,"0
0000000303030300000000FEFEF
EFE0303030303030303030303030
EFEFEFE")
310 CALL CHAR(136,"000000040
E1F3F7F0103070E1C38F0E0FFFFF
F7F3F1F0F07E0F0F8F0E0C08000"
)
320 CALL CHAR(140,"030509112
24488F"):: CALL CHAR(132,"FF
FFFF")
330 CALL CHAR(104,"00F0F1B1
F3F0F070080C0E0E0E0C0807C7C3
E1E0F0F3FFF")
340 CALL CHAR(112,"03070F0E1
D1B374E80C0E060B0B070F0FDF3C
F6F683F1F1F1F8F8F0F0E000E0C0"
)
350 CALL CHAR(116,"1F3F7F7FF
FFFFEFC0C0808080000000")::
CALL CHAR(40,"2040808EBF7F7F
7F3F1F0F040C18180000081C3698
FCF6FFF0C8080404020200")
360 CALL HCHAR(14,1,96,320):
: DISPLAY AT(24,6):"SQUISHED
";TAB(24);"HITS"

```

```

370 CALL HCHAR(10,15,128)::
CALL HCHAR(10,16,129):: CALL
HCHAR(11,15,130):: CALL HCH
AR(11,16,131):: CALL HCHAR(1
1,17,132)
380 CALL CHAR(44,"0018180C04
06193F3F2F7F9F8F804020000306
0C0C0C0D8F8FFF0FC983C1C0800")
390 CALL HCHAR(12,17,136)::
CALL HCHAR(12,18,137):: CALL
HCHAR(13,17,138):: CALL HCH
AR(13,18,139):: CALL HCHAR(1
1,19,140)
400 FLAG=0 :: CALL COLOR(10,
10,1,11,13,1,9,7,1,12,13,1)
410 CALL SPRITE(#1,40,15,90,
10)
420 MOUSE=MOUSE+1 :: IF MOUS
E=OFF THEN GOSUB 490
430 CALL KEY(1,K,S):: IF S=0
THEN 420
440 CALL COLOR(13,13,1,12,1,
1,14,13,1,13,1,1):: CALL SOU
ND(50,-6,0)
450 CALL POSITION(#1,ROW,COL
):: IF COL>120 AND COL<130 T
HEN 500
460 STRIKE=STRIKE+1 :: DISPL
AY AT(24,19)SIZE(4):STRIKE :
: IF STRIKE>24 OR HIT>9 THEN
530
470 FOR I=1 TO 50 :: NEXT I
:: CALL COLOR(13,13,1,14,1,1
):: FOR D=1 TO 10 :: NEXT D
:: CALL COLOR(12,13,1,13,1,1
)
480 IF FLAG=1 THEN 400 ELSE
430
490 MOUSE=0 :: OFF=INT(RND*2
50)+25 :: CALL MOTION(#1,0,I
NT(RND*50)+1):: RETURN
500 CALL PATTERN(#1,44):: CA
LL MOTION(#1,20,0):: CALL SO
UND(250,5000,0):: CALL SOUND
(150,110,0):: FLAG=1
510 FOR NOISE=3000 TO 500 ST
EP -100 :: CALL SOUND(-50,NO
ISE,0):: NEXT NOISE :: CALL
SOUND(75,110,6,250,6)
520 HIT=HIT+1 :: DISPLAY AT(
24,1)SIZE(4):HIT :: GOSUB 49
0 :: GOTO 460
530 CALL CLEAR :: DISPLAY AT
(8,1):"AS A MOUSE CATCHER YO
U" :: X=STRIKE-HIT
540 DISPLAY AT(15,1):WORD$(X
):: DISPLAY AT(20,5):"PLAY A
GAIN Y OR N"
550 CALL KEY(0,K,S):: IF S=0
OR S=-1 THEN 550
560 STRIKE,HIT=0 :: IF K=ASC
("Y")THEN 240 ELSE END

```

continued from page 19

```

380 CALL HCHAR(10,10+I,CS1+A
I(I)):: CALL HCHAR(11,10+I,C
S1+BI(I)):: CALL HCHAR(13,10
+I,CS1+ANS2):: CALL HCHAR(9,
10+I,CS1+10):: NEXT I
390 FOR I=1 TO 500 :: NEXT I
:: GOTO 210
400 END

```



TISHUG NEWS DIGEST

```

100 REM CHANNEL 99 HAMILTON
USER'S GROUP
110 REM WARRIOR
120 REM A1-BO010-JGU
130 REM *****
140 REM * BY MIKE TOWERS *
150 REM *****
160 GOSUB 1870
170 CALL CLEAR
180 DIM MON$(6)
190 DIM WEAP$(6)
200 GOLD=0
210 HITP=INT(RND*2)+17
220 EXPE=0
230 RANDOMIZE
240 DATA OGRE,CLUB,ORC,SPEAR
,GOBLIN,DAGGER,HOBGOBLIN,MAC
E,KOBOLD,AXE,LIZARDMAN,SWORD
250 FOR LOOP=1 TO 6
260 READ MON$(LOOP),WEAP$(LO
OP)
270 NEXT LOOP
280 CALL CLEAR
290 F=INT(RND*100)+1
300 IF F>60 THEN 320
310 GOSUB 900
320 PRINT "I AM IN A OPEN FI
ELD WHAT SHALL I DO?"
330 PRINT "1. GO WEST"
340 PRINT "2. GO EAST"
350 PRINT "3. GO SOUTH"
360 PRINT "4. GO NORTH"
370 CALL KEY(3,K,S)
380 IF K<>49 THEN 400
390 GOSUB 460
400 IF K<>50 THEN 420
410 GOSUB 570
420 IF K<>51 THEN 440
430 GOSUB 680
440 IF K<>52 THEN 370
450 GOSUB 790
460 F=INT(RND*10)+1
470 IF F>6 THEN 490
480 GOSUB 900
490 CALL CLEAR
500 PRINT "I AM IN A COURTYA
RD.":"WHAT SHALL I DO?":"1.
GO EAST":"2. GO WEST"
510 CALL KEY(3,K,S)
520 IF S=0 THEN 510
530 IF K<>49 THEN 550
540 GOTO 280
550 IF K<>50 THEN 510
560 GOTO 1570
570 F=INT(RND*10)+1
580 IF F>6 THEN 600
590 GOSUB 900
600 CALL CLEAR
610 PRINT "I AM IN AN ORCHAR
D.":"WHAT SHALL I DO?":"1. G
O WEST":"2. GO EAST"
620 CALL KEY(3,K,S)
630 IF S=0 THEN 620
640 IF K<>49 THEN 660
650 GOTO 280
660 IF K<>50 THEN 620
670 GOTO 1570
680 F=INT(RND*10)+1
690 IF F>6 THEN 710
700 GOSUB 900
710 CALL CLEAR
720 PRINT "I AM IN A GARDEN.
":"WHAT SHALL I DO?":"1. GO
NORTH":"2. GO SOUTH"
730 CALL KEY(3,K,S)
740 IF K<>49 THEN 770
750 IF S=0 THEN 730
760 GOTO 280
770 IF K<>50 THEN 730
780 GOTO 1570
790 F=INT(RND*10)+1
800 IF F>6 THEN 820

```

```

810 GOSUB 900
820 CALL CLEAR
830 PRINT "I AM IN A ROCK GA
RDEN.":"WHAT SHALL I DO?":"1
. GO SOUTH":"2. GO NORTH"
840 CALL KEY(3,K,S)
850 IF S=0 THEN 840
860 IF K<>49 THEN 880
870 GOTO 280
880 IF K<>50 THEN 840
890 GOTO 1570
900 W=INT(RND*6)+1
910 M=INT(RND*6)+1
920 PRINT "I AM BEING ATTACK
ED BY A ";MON$(M);".";"HE
IS USING A ";WEAP$(W);"." :
"WHAT SHALL I DO?"
930 PRINT "1. FIGHT"
940 PRINT "2. RUN"
950 PRINT "3. SURRENDER"
960 CALL KEY(3,K1,S1)
970 IF S1=0 THEN 960
980 IF K1<>49 THEN 1400
990 CALL CLEAR
1000 PRINT "WHAT SHALL I FIG
HT WITH?":"1. SWORD":"2. DAG
GER":"3. BOW"
1010 CALL KEY(3,KF,SF)
1020 IF KF<>49 THEN 1300
1030 CTK=INT(RND*10000)+1
1040 IF CTK<(7500+EXPE)THEN
1220
1050 PRINT "YOU KILLED HIM!!
":"SHALL I SEARCH THE BODY
?":"1. YES":"2. NO"
1060 EXPE=EXPE+INT(RND*15+1)
1070 CALL KEY(3,KS,SS)
1080 IF SS=0 THEN 1070
1090 IF KS<>49 THEN 1210
1100 GOL=INT(RND*10)+1
1110 CTD=INT(RND*1000)+1
1120 IF CTD>800 THEN 1190
1130 EXPE=EXPE+GOL
1140 PRINT "I FOUND ";GOL;"
GOLD PIECES!"
1150 FOR DELAY=1 TO 500
1160 NEXT DELAY
1170 GOLD=GOLD+GOL
1180 RETURN
1190 PRINT "I AM DEAD, I CAU
GHT A DISEASE.":"I HAD
";GOLD;" PIECES OF GOLD AND
";EXPE;" EXPERIENCE POINTS"
1200 END
1210 RETURN
1220 HITP=HITP-(INT(RND*6)+1
)
1230 IF HITP<1 THEN 1280
1240 PRINT "I WAS HIT BUT I'
M OK"
1250 FOR DELAY=1 TO 300
1260 NEXT DELAY
1270 GOTO 920
1280 PRINT "UHOH!! I HAVE BE
EN SLASHED TO PIECES!!":"I
AM DEAD!!":"I HAD ";GOLD;" G
OLD PIECES AND ";EXPE;" EXP
ERIENCE POINTS"
1290 STOP
1300 IF KF<>50 THEN 1350
1310 CTK=INT(RND*10000)+1
1320 IF CTK<(5000+EXPE)THEN
1340
1330 GOTO 1050
1340 GOTO 1220
1350 IF KF<>51 THEN 1010
1360 CTK=INT(RND*10000)+1
1370 IF CTK<(6500+EXPE)THEN
1390
1380 GOTO 1050
1390 GOTO 1220
1400 IF K1<>50 THEN 1480

```

```

1410 EXPE=EXPE-10
1420 CTS=INT(RND*1000)+1
1430 IF CTS<(500+EXPE)THEN 1
220
1440 PRINT "WHEW! I MADE IT.
"
1450 FOR I=1 TO 300
1460 NEXT I
1470 RETURN
1480 IF K1<>51 THEN 960
1490 CALL CLEAR
1500 EXPE=EXPE-20
1510 CTS=INT(RND*100)+1
1520 IF CTS>15 THEN 1220
1530 PRINT "WHEW! HE DECIDED
I WAS PEACEFUL AND LEF
T."
1540 FOR DELAY=1 TO 300
1550 NEXT DELAY
1560 RETURN
1570 F=INT(RND*10)+1
1580 IF F>6 THEN 1600
1590 GOSUB 900
1600 CALL CLEAR
1610 PRINT "I AM OUTSIDE A C
ASTLE.":"WHAT SHALL I DO?":"
1. GO CASTLE":"2. GO WEST":"
3. GO EAST":"4. GO SOUTH"
1620 PRINT "5. GO NORTH"
1630 CALL KEY(3,K,S)
1640 IF S=0 THEN 1630
1650 IF K<>49 THEN 1670
1660 GOTO 1750
1670 IF K<>50 THEN 1690
1680 GOTO 570
1690 IF K<>51 THEN 1710
1700 GOTO 460
1710 IF K<>52 THEN 1730
1720 GOTO 680
1730 IF K<>53 THEN 1630
1740 GOTO 790
1750 F=INT(RND*10)+1
1760 IF F>7 THEN 1780
1770 GOSUB 900
1780 CALL CLEAR
1790 PRINT "I HAVE BEEN TRAP
PED!":"WHAT SHALL I DO?":"1.
GET OUT":"2. STAY AND FIGHT
"
1800 CALL KEY(3,K,S)
1810 IF S=0 THEN 1800
1820 IF K<>49 THEN 1840
1830 GOTO 1570
1840 IF K<>50 THEN 1800
1850 GOSUB 900
1860 GOTO 1750
1870 CALL CLEAR
1880 PRINT TAB(11);"MIKE TOW
ERS":TAB(12);"PRESENTS": :
TAB(13);"WARRIOR": : : : :
: : : : :
1890 CALL KEY(3,K,S)
1900 IF S=0 THEN 1890
1910 RETURN

```

```

100 REM CHANNEL 99 HAMILTON
USER'S GROUP
110 REM COMPUTER NOTEBOOK
120 REM VERSION 1 1984
130 REM B2-BO056-JAU
140 REM *****
**
150 REM * BY STEVE WHITEHEAD
*
160 REM *****
**
170 CALL CLEAR :: PRINT "DO
YOU WANT TO :":"1 CREATE A N
EW FILE?":"2 RECALL AN OLD F
ILE?"

```

```

180 INPUT NF :: IF NF<1 OR N
F>2 THEN 180 :: IF NF=1 THEN
GOTO 190 ELSE 270
190 GOSUB 210
200 GOTO 290
210 CALL CLEAR :: DISPLAY AT
(12,1):"NEW FILENAME:
"
220 ACCEPT AT(12,14)SIZE(-10
):FN$
230 OPEN #1:"DSK1."&FN$,OUTP
UT,VARIABLE 126,SEQUENTIAL,I
NTERNAL
240 PRINT #1:"NOTEBOOK"
250 CLOSE #1 :: RETURN
260 GOTO 290
270 CALL CLEAR :: DISPLAY AT
(12,1):"OLD FILENAME:
"
280 ACCEPT AT(12,14)SIZE(-10
):FN$
290 DIM A$(200)
300 CALL CLEAR :: I=1 :: CH=
0 :: PRINT "LOADING FILE"
310 OPEN #1:"DSK1."&FN$,INPU
T,VARIABLE 126,SEQUENTIAL,I
NTERNAL
320 INPUT #1:A$(I)
330 IF EOF(1)THEN 340 ELSE I
=I+1 :: GOTO 320
340 CLOSE #1 :: CALL CLEAR :
: PRINT I;"RECORDS IN FILE"
350 PRINT : "1=SEARCH BY KEYW
ORD": "2=ADD DATA": "3=RECALL
ANOTHER FILE": "4=CREATE ANOT
HER FILE": "5=LIST DATA TO PR
INTER": "6=END"
360 INPUT C :: IF C<1 OR C>6
THEN 360 :: ON C GOTO 370,4
30,620,640,660,490
370 INPUT "KEYWORD=":K$
380 FOR J=1 TO I :: IF POS(A
$(J),K$,1)=0 THEN 420 ELSE C
ALL CLEAR
390 PRINT :A$(J): : "PRESS"
: " S TO SEARCH MORE": " C TO
CHANGE THIS RECORD": " K FOR
ANOTHER KEYWORD": " Q TO QUIT
SEARCH"
400 CALL KEY(O,K,S):: IF S=O
THEN 400 ELSE IF K=83 THEN
420 ELSE IF K=75 THEN 370 EL
SE IF K=81 THEN 350 ELSE IF
K=67 THEN 410 ELSE 400
410 PRINT "RE-ENTER DATA:" :
: LINPUT A$(J):: CH=CH+1 ::
GOTO 390
420 NEXT J :: GOTO 350
430 CALL CLEAR
440 PRINT "ENTER # TO END.":
: "ENTER DATA:"
450 LINPUT AD$ :: IF AD$="##"
THEN 480
460 IF LEN(AD$)>125 THEN DIS
PLAY : "LINE TOO LONG--TRUNCA
TED!" :: CALL SOUND(100,110,
0):: AD$=SEG$(AD$,1,125):: D
ISPLAY :AD$
470 I=I+1 :: A$(I)=AD$ :: CH
=CH+1 :: GOTO 450
480 GOTO 350
490 GOSUB 510
500 END
510 IF CH=O THEN 550
520 PRINT "UPDATING FILE"
530 OPEN #1:"DSK1."&FN$,OUTP
UT,VARIABLE 126,SEQUENTIAL,I
NTERNAL
540 FOR J=1 TO I :: PRINT #1
:A$(J):: NEXT J :: CLOSE #1
:: RETURN
550 STOP

```

```

560 CALL CLEAR :: DISPLAY AT
(10,1):"ENTER PRINTER NAME:
"
570 ACCEPT AT(10,21):PN$
580 OPEN #1:PN$
590 PRINT #1: : :: FOR J=1 T
O I :: PRINT #1:A$(J):: NEXT
J :: CLOSE #1
600 GOTO 350
610 END
620 GOSUB 520
630 GOTO 270
640 GOSUB 520
650 GOTO 190
660 GOSUB 520
670 GOTO 560
30000 !*****
30010 !Subroutine to re- *
30020 !define the lowercase*
30030 !character set to *
30040 !true lower-case. *
30050 !It also shifts the *
30060 !other charcters up *
30070 !one row to allow for*
30080 !the descenders. *
30090 !*****
30100 ! *
30110 !Author: Phil Maier; *
30120 !TICHUG in Canberra. *
30130 ! *
30140 !*****
30150 ! *
30160 ! Shift all chars
30170 ! *
30180 FOR I=32 TO 96
30190 CALL CHARPAT(I,C$)
30200 CALL CHAR(I,SEG$(C$,3,
14))
30210 NEXT I
30220 ! *
30230 ! Slash the zero
30240 ! *
30250 CALL CHAR(48,"38444C54
64443800")
30260 ! *
30270 ! Lower-case alphas
30280 ! *
30290 RESTORE 30360
30300 FOR I=97 TO 122
30310 READ C$
30320 CALL CHAR(I,C$)
30330 NEXT I
30340 RESTORE
30350 RETURN
30360 DATA 0000748C848C7400
30370 DATA 8080B8C484C4B800
30380 DATA 0000788480847800
30390 DATA 0404748C848C7400
30400 DATA 00007884F8807800
30410 DATA 1824207020202000
30420 DATA 0000748C8C740478
30430 DATA 4040784444444400
30440 DATA 1000101010101800
30450 DATA 0010001010107000
30460 DATA 80808090E0908800
30470 DATA 2020202020203000
30480 DATA 00006C9292929200
30490 DATA 0000586444444400
30500 DATA 0000788484847800
30510 DATA 0000B8C4C4B88080
30520 DATA 0000748C8C740404
30530 DATA 0000586440404000
30540 DATA 0000788078047800
30550 DATA 1010381010101800
30560 DATA 00004444444C3400
30570 DATA 0000444428281000
30580 DATA 0000828292926C00
30590 DATA 00006C2810286C00
30600 DATA 00004444443C0438
30610 DATA 00007C0810307C00

```

```

100 REM ADDITION EXERCISER
110 REM by Geoff Trott
120 REM Illawarra Regional G
roup, TISHUG
130 REM 2 NOVEMBER 1983
140 DIM NUMS(11),A(22),B(22)
,AI(22),BI(22):: FOR I=0 TO
10 :: READ NUMS(I):: NEXT I
150 DATA "003844444444438",
"0010301010101038", "00384404
0C38607C", "0038440418044438"
,"00182828487C0808"
160 DATA "007C407804044438",
"0038444078444438", "007C040C
18306040", "0038444438444438"
,"003844443C044438", "000000"
170 CS1=112 :: UL=CS1+15 ::
PL=ASC("+"):: CS2=CS1+16 ::
FOR I=0 TO 10 :: CALL CHAR(C
S1+I,NUMS(I)):: CALL CHAR(CS
2+I,NUMS(I)):: NEXT I
180 CALL CHAR(UL,"000000FFFF
000000"):: CALL CLEAR :: PRN
T "MAXIMUM VALUE FOR DIGITS
": : INPUT "FIRST NUMBER (2
-9): " :D1 : INPUT "SECOND N
UMBER (2-9): " :D2
190 PRINT : D1=D1+1 :: D2=D
2+1 :: IF D1>10 THEN D1=10 E
LSE IF D1<2 THEN D1=2
200 RANDOMIZE :: IF D2>10 TH
EN D2=10 ELSE IF D2<2 THEN D
2=2
210 ND=INT(22*RND)+1 :: CALL
CLEAR :: CALL SCREEN(5):: C
ALL COLOR((CS1-24)/8,2,1)::
CALL COLOR((CS1-16)/8,2,1)::
CALL COLOR((CS2-24)/8,2,16)
220 CALL COLOR((CS2-16)/8,2,
16):: FOR I=1 TO ND :: A(I)=
INT(RND*D1):: B(I)=INT(RND*D
2):: NEXT I
230 IF (A(1)+B(1))>8 THEN IF
B(1)>1 THEN B(1)=B(1)-1 ::
GOTO 230 ELSE A(1)=A(1)-1 ::
GOTO 230
240 FOR I=1 TO ND :: IF A(I)
>0 THEN GOTO 260 ELSE BI(I)=
10
250 NEXT I
260 FOR J=I TO ND :: AI(J)=A
(J):: NEXT J
270 FOR I=1 TO ND :: IF B(I)
>0 THEN GOTO 290 ELSE BI(I)=
10
280 NEXT I
290 FOR J=I TO ND :: BI(J)=B
(J):: NEXT J
300 FOR I=1 TO ND :: CALL HC
HAR(10,10+I,CS1+AI(I)):: CAL
L HCHAR(11,10+I,CS1+BI(I))::
NEXT I
310 CALL HCHAR(12,11,UL,ND)::
CALL HCHAR(11,10,PL):: ANS
1=10
320 FOR I=ND TO 1 STEP -1 ::
CALL HCHAR(9,10+I,CS1+ANS1)
:: CALL HCHAR(10,10+I,CS2+AI
(I)):: CALL HCHAR(11,10+I,CS
2+B(I))
330 ANS=A(I)+B(I):: IF ANS1=
10 THEN GOTO 340 ELSE ANS=AN
S+ANS1
340 ANS1=INT(ANS/10):: ANS2=
ANS-ANS1*10 :: IF ANS1<>0 TH
EN GOTO 350 ELSE ANS1=10
350 CALL HCHAR(13,10+I,CS2+1
0)
360 CALL KEY(5,CD,STAT):: IF
STAT=0 THEN GOTO 360 ELSE I
F (ANS2-CD+48=0)THEN 380
370 FOR J=1 TO 50 :: CALL HC
HAR(13,10+I,CS1+10):: NEXT J
:: GOTO 350

```

continued on page 17

Jenny's Younger Set

Happy New Year to all the younger gang!

I received quite a lot of pieces of paper from Vincent Maker. Thanks Vincent, for the Merry Christmas; bad luck it did not quite make it for the December issue. Better late than never I always say! How about some of you others sending in something about your holidays, even your best scores in the games you have played. Don't tell me you haven't played any! Here are Vincent's contributions.

Dear Jenny,

I hope you like the program I sent you. It might come in useful for someone who would like to run a new year's program on new year's eve. An idea would be to run it and let the new year's message come up at midnight. Taht is let the PRESS ANY KEY or whatever come up and press the key at midnight. However, I wish everyone reading this, a very Merry Christmas.

Wishing you a Merry Christmas,
Vincent Maker.

```

100 REM *****
110 REM *NEW YEAR'S PROGRAM*
120 REM *
130 REM * BY VINCENT MAKER *
140 REM *
150 REM * FOR GABRIELLE *
160 REM * IN XB*
170 REM *****
180 CALL CLEAR
190 DISPLAY AT(3,3):"THE NEW YEAR'S PROGRAM"
200 DISPLAY AT(5,3):"BY VINCENT MAKER"
210 INPUT "OLD YEAR?":YEAR1
220 PRINT
230 INPUT "NEW YEAR?":YEAR2
240 PRINT
250 PRINT "PRESS ANY KEY WHENEVER YOU ARE READY"
260 CALL KEY(0,H,J)
270 IF H=0 THEN 260
280 CALL CLEAR
290 CALL SCREEN(2)
300 DISPLAY AT(4,5):"*HAPPY NEW YEAR*"
310 DISPLAY AT(11,3):"GOODBYE TO ";YEAR1;"!!"
320 DISPLAY AT(15,3):"WELCOME TO ";YEAR2;"!!"
330 FOR T=0 TO 9 :: CALL COLOR(T,12,2) :: NEXT T
340 CALL SCREEN(2)
350 PRINT "ALL THE BEST FOR THIS YEAR!!"
360 GOTO 360

100 REM FOOTY TAB PROGRAM IN XB
110 REM BY VINCENT MAKER
120 CALL CLEAR
130 DISPLAY AT(5,7):"PICK THE WINNER BY VINCENT
MAKER"
140 PRINT "TEAM 1:"
150 INPUT AS
160 PRINT "TEAM 2:"
170 INPUT BS
180 A=0
190 B=0
200 PRINT "HAS ";AS;" HAD A WIN LAST WEEK (Y/N)?"
210 INPUT ABS
220 IF ABS="Y" THEN 240
230 A=1
240 PRINT "HAS ";BS;" HAD A WIN LAST WEEK (Y/N)?"
250 INPUT BCS
260 IF BCS="N" THEN 280
270 B=1
280 PRINT "WHO HAS MORE POINTS ON THE TABLE AT THE
MOMENT (IF IT IS THE SAME TYPE 'S', IF TEAM1 DOES
TYPE 1 OTHERWISE TYPE 2)?"
290 INPUT DS
300 CALL CLEAR
310 IF A>B THEN WINNERS$=AS
320 IF B>A THEN WINNERS$=BS
330 IF A=B THEN WINNERS$="DRAW, TAKE THE TEAM WITH THE
START"
340 INPUT "TRY AGAIN (Y/N)?:ES
350 IF ES="Y" THEN 120
360 END
    
```

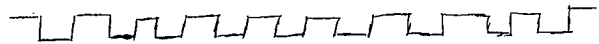
```

100 I=0
110 REM FOR GABRIELLE
120 REM NEW IMPROVED GUESSING GAME BY VINCENT MAKER
130 RANDOMIZE
140 B=INT(RND)
150 CALL CLEAR
160 INPUT "A NUMBER BETWEEN 0-10, PLEASE ":A
170 I=I+1
180 IF A<>B THEN 230
190 PRINT "** * WELL DONE * * YOU GOT IT
IN ";I;" GOES"
200 INPUT "GUESS AGAIN (Y/N)?:AS
210 IF AS="Y" THEN 100
220 END
230 IF A>B THEN 260
240 PRINT : "IT'S MORE THAN ";A;" !"
250 GOTO 160
260 PRINT : "IT'S LESS THAN ";A;" !"
270 GOTO 160
    
```

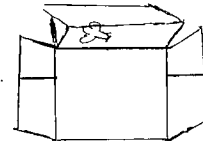
Dear Sirs,

Here is an Adventure project you might like to use in your magazine.

yours faithfully,
Vincent Maker



Voodoo Castle #4, by Vincent Maker

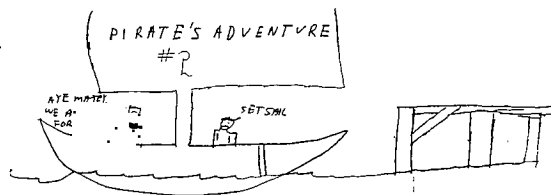


PROBLEM WITH CRACK? The medium said a moving bag.. Could this be the Voodoo bag in the laboratory? (It could have another name, however). Think about how to move it.

PROBLEM WITH IDOL? Dusting is a good way of cleaning.

PROBLEM WITH CHARMS? Move the kettle for the other charm.

The jail cell door will not open again after it closes. You need the 4 leaf leaf in the room with the window.



THE BEST OF THE ADVENTURES



The Teaching Computer by Jim Peterson

I still consider the TI-99/4A to be a HOME computer, although many users have expanded it far beyond that.

And what are the uses of a home computer? Primarily, entertainment and education - and, if you can justify the expense of an RS232, printer and modem, word processing and telecommunications.

The importance of the computer in education has been overemphasized in TV advertising, to the extent that people were offended by suggestions that, if they failed to buy a home computer, they were condemning their children to a life of failure. One man even formed an organization to oppose the TI-99/4A ads!

However, the educational potential of the TI-99/4A was never realized. To teach any subject, a planned series of lessons is required. And, since each lesson is soon learned and no longer needed, the individual lessons must be inexpensive.

Texas Instruments did put out several educational modules, but they were not in any planned series and, in those days, they were not cheap. The Plato series is carefully planned, and excellently designed from an educational standpoint - but it does not take full advantage of computer capabilities, and is far too expensive.

Third party manufacturers did put out some fine educational modules but again, no planned series, and I do not know of any of them still supporting the TI. In the early days, Micro-Ed and a few others put out some good educational software on cassette. As far as I know, Kidware and Tigercub are the only ones still offering educational programs on cassette.

There is also a vast amount of public domain software written by amateur programmers. It ranges in quality from mediocre to excellent, consists mostly of pre-school teaching or basic math drills, and would take much effort to organize into any course of education.

The market for educational software has been so poor that it is doubtful that any more will be produced. So, if you need it, you'll have to write it for yourself!

Ideally, educational programs should be written by teachers, because they know how to teach. Unfortunately, it seems that few teachers have learned how to write programs, although some of their students have. A cooperative effort could have resulted in a large pool of good educational public domain software available to schools. However, the educational establishment as a whole has been so brainwashed by the Apple peddlers and the disciples of Logo that no support is given to anything else.

What makes a good educational program? If possible, it should be interactive - it should, 1. teach a lesson; 2. test to see if you have learned the lesson; 3. if you have not, go back and teach it again; 4. if you have learned, go on to the next lesson.

This is not always practical in a single program, but can be done by having one program run the next program. I used a variation of this technique in "Casting Out Nines", which teaches a method of checking long multiplication and division problems. The student must correctly solve several problems at each step before the program will continue to the next step.

The program should give some kind of spoken, printed, graphics or musical reward for correct answers, and a corresponding admonition for wrong answers. The nature of these depends on the age level of the student. In "Kindermath" I use a groan and a frowning face which can be changed to a smile and music by a correct answer. In "Kinderminus", the problem is

displayed in the center of a multicolored kaleidoscope which changes patterns for each correct answer or turns black for a wrong answer. For an older student, a short game to be played could be offered after successfully answering a series of questions.

Computer graphics and sound should be fully utilized, but not allowed to become boring. The same musical salute after each correct answer soon gets tiresome. In some of my programs, a note is added to a tune for each correct answer, and after several such answers the entire tune is played. "Kindermath" uses several nursery tunes in succession.

The "stupid computer syndrome" should be avoided. This occurs when random selection causes the same question to be asked twice in succession. It is easily prevented by the simple statement IF Q=Q2 THEN (go back for another selection) ELSE Q2=Q.

The computer is especially well adapted to teaching math, because it can generate an infinite number of random problems to be solved. When the problem requires keying in an answer to a multi-digit problem displayed on the screen, input should be accepted from right to left in the same way as it was being worked on paper. My "Math Homework Helper" does this with addition, subtraction and multiplication problems of any size, and also helps the student by refusing to accept an incorrect digit.

When possible, if a wrong answer is given the student should be shown how to work the problem. In one of our Extended Basic classes we analyzed a math quiz program which generated random problems in the form of "IF 3 BOYS CAN CATCH 12 FROGS IN 4 DAYS, HOW MANY FROGS CAN 9 BOYS CATCH IN 8 DAYS?". If the answer was wrong, a screen display explained, "NO, THAT'S WRONG. IF 3 BOYS CAN CATCH 12 FROGS IN 4 DAYS, THEN 3 BOYS CAN CATCH 3 FROGS IN ONE DAY"....etc., through the problem. Similar routines could be written for a wide variety of time/speed/distance problems, etc.

The basics of music education can also be easily taught by computer. The TI-99/4A can generate any musical tone required, and the piano keyboard, guitar fingering, musical notation, or whatever can be graphically displayed. John and Norma Clulow, Regena, and Bob Pomictter have written some excellent programs of this type.

Educational programs requiring much text are more difficult because of the limited memory capacity of the computer. Many of the public domain programs of this kind seem to have been typed in directly from a textbook, and there is really little reason for computerizing them. Some good module software is also of this type - nicely programmed, but very soon learned and discarded.

Most speed reading programs on the market are loaded with a data base of sentences, which are flashed on the screen briefly and the student is then asked to repeat them. He soon begins to recognize them from memory, even though he may think he is reading faster. Also, the purpose of speed reading is to grasp the meaning of a sentence, not its exact wording. My "Speeder Reader" and "Junior Speeder Reader" programs bypass these faults by being loaded with a wide selection of individual nouns, verbs, adjectives and modifiers which are randomly selected and combined into an infinite number of sentences. Then, the student is asked any one of several different questions about the sentence, requiring a one-word reply.

Public domain spelling programs are popular but not very practical. In order to tell the student to spell a word, it must be either printed on the screen or spoken by the speech synthesizer. In the former case, he is already being shown how to spell it. The vocabulary of the speech synthesizer is rather small. The TE II permits an unlimited vocabulary, but the pronunciation is not very clear - a sentence may be understood, but a single word out of context is more difficult.

continued on page 22

TI-Writer Tips

by Geoff Trott, Illawarra Regional Group

While doing some work on editing a file for the TND, I came across a problem which I had not found before, so I thought I should share it with you all. The problem relates to the use of commands in the source file which are used by the Formatter to cause the printout to be as required. If you do not use the Formatter, then this problem will never occur to you, so you may stop reading now, or if you like to read how an idiot survives the TI-Writer jungle, then continue reading.

I prefer to run the editor so that the screen does not window. To do this, and still use the whole screen for data, I set the right margin to column 39 using the Tab command. Then I turn off the line numbers with FCTN(0). When you go to command mode, the line numbers appear whether you want them or not. Then, in order to get the output to any other width and to do all sorts of other fancy things like centering, indenting, justifying and the like, Formatter commands are put in at the appropriate places in the text. These are identified by a "." as the first non space character in a paragraph and must be terminated by the end of paragraph character (CR). More than one command can be put on one line by separating them with ";", except for the TL command, which must be on its own.

So far so good! We all know that much don't we? The problem I found was with the indent command IN. This has a number after it which says how far and in which direction to indent. What was present in the file was several indent commands which typically looked like this: ".IN +5 <CR>"; where <CR> represents the end of paragraph character. Everything looked fine, except it did nothing! In desperation I retyped the first one, which was on the first line along with the margin set up commands, and it then worked but the next one did not. How frustrating, what is going on here?

I was desperate enough to look at the data on the disk and there was no obvious difference between the one which worked and those that did not, except... the light dawned! If you look carefully, there is a space character between the 5 and the <CR> character. The one I had retyped did not have such a space, as the space appeared during a reformat operation. This obviously upsets the calculation of the indent, and so the Formatter does nothing. Remove the spaces and hey presto, all is working again. Serves me right for using the Formatter do I hear you shout? Everyone is entitled to their opinion, and I shall continue to use the Formatter, no matter how loud you shout at me, although I respect your right to have a different opinion, and to express it.

Another interesting effect came when printing the Forth screens in this issue, when whole lines did not get printed. All these lines started with the Forth word "<.>", although not in the first column. These had obviously been interpreted as Formatter commands, even though there were none there, and the whole line ignored. This was easily remedied with a Transliterate to another character.

continued from page 21

Ron Binkowski published a "Speller" program in the 99'er which overcame this problem by allowing a separate phonetic spelling of the word and an "as in" phrase which could be phonetically spelled until it sounded right.

In my "Miss Spell", I programmed each word in a correct and incorrect spelling, randomly showed either one to the student and asked him if it was correct and, if not, to spell it correctly. In "I & E Spelling" I programmed every word containing those difficult "ie" and "ei" combinations, randomly showed them on the screen with those two letters replaced by blanks, to be filled in. If the answer is wrong, the student is shown a screen displaying the "I BEFORE E EXCEPT AFTER C" rule with all its exceptions.

USING ALL 256 ASCII CHARACTERS WITH TI WRITER

by David A. Aragon, San Antonio Area 99ers

I decided to write a short article on accessing the characters above ASCII code 127 because I had a need to utilize some of the special characters on my printer (Gemini 10X) in a report I was writing. I looked through every article I could find, asked everyone I knew, and even looked in the TI Writer manual, but to no avail. It took me a little while to figure it out, which is embarrassing because it's so simple, and while I'm not sure how many of you have a need to access these codes, there might be a few of you who do, and maybe this will help you. Please remember that the procedure I give is for the Gemini printer. Some modification may be necessary for other printers.

If you look at the chart on page 240 of your Gemini User's manual, several things should be noted. First, by using this chart you can quickly determine the ASCII value for any particular character you need. You will notice too that, since all the characters available from the keyboard are used, it was impossible to use the them. Now, if you look even closer you will notice that there are several codes that the printer ignores, specifically ASCII codes 1 to 6, 16, 21 to 26, 28, 29, and 31. Lastly, you should notice that the codes from 128 to 159 do the same thing as the codes from 0 to 31. This might lead you to wonder if the codes from 160 to 255 are the same as the codes from 32 to 127. Well, they are...almost, the difference being that above ASCII code 127 the eighth bit is 1 instead 0. You can see this by comparing the binary code for character 63 (00111111) and the binary code for character 191 (10111111) for example.

By using .TL and a few of the unused lower codes it became simply a matter of redefining that character to tell the printer what I wanted it to do. For instance, ".TL 26:32,27,62" tells the printer to print a space and then do an ESC >. This tells the printer to process all subsequent data with the eighth bit regarded as 1. On the other hand, ".TL 28:27,61,32" tells the printer to print a space after doing an ESC = which is the opposite of ESC > in that all subsequent data is processed with the eighth bit regarded as a 0 (see pg. 139 Gemini User's guide). Now, to call them into use, press CTRL U SHIFT Z to go into the special character or graphics mode and CTRL U FCTN Z to get back out of it (see pg 146 TI Writer manual). What goes in between them would be the letter(s) below ASCII code 127 that you want to include in your document. The space character is not necessary for this to work, but I recommend it to make things a little easier. The character generated takes up a space on the screen but is not printed. By putting the space character (32) in the to line things up to look right. It now becomes much simpler to print formulas or even include simple graphics in your document.

Finally, the best educational programs of them all are those that teach a person something while he thinks he is just having fun. Word games are of this type, whether the popular "Hangman" or "Scrabble" or whatever.

My "Tirkle" is just a very simple little game for children, based on the early computer game called "Hurkle". However, teachers have told me that they find it very useful for teaching young children logical thinking and compass directions.

The possibilities are endless - and so little is being done!

The Forth Column

Forth and Double Sided, Double Density Disks

The following is verbatim from CorComp Cursor, a newsletter to TI-99/4A user groups. When I called to see if they minded if I put this up here I was told the intent of the newsletter was that it be given the widest dissemination possible. So here is what they say about TI-Forth and DSDD (and other) set-ups. Hope you find it useful.

Rich Stanford
75236,43

TI FORTH

This article is intended for all TI FORTH users who have (or plan on having) double density and/or double sided disk capabilities. While the techniques described should work with any disk controller capable of double density, the author's CorComp 9900 Disk Controller card is the only one that has been tested. The purpose of this article is to illustrate both how to access the additional screen capacity and how to modify the FORTH words and disk to be compatible with the new format and Disk Manager.

Throughout this article, lowercase letters used in a FORTH definition will indicate a variable value to be entered. The following terms will be used to refer to the various formats a FORTH disk may have.

90 SCRIN or SSSD - the original 90 screen single sided single density format.

180 SCRIN - either a SSDD or DSDD disk when comment applies to both.

360 SCRIN or DSDD - a double sided double density disk.

SSDD - a single sided double density disk.

DSDD - a double sided single density disk.

The first step is to use Disk Manager to format (initialize) a 180 or 360 SCRIN disk. Next, you must copy FORTH from the 90 SCRIN disk to the new 180 or 360 SCRIN disk. The disk copy feature of CorComp's Disk Manager will do this properly for you. If you have two drives, the FORTH-COPY word in the -COPY screens will also do it properly (do 0 DISK_LO ! first). However, if you are using TI's Disk Manager II, after copying the three files you must use FORTH to copy screens 1 to 9 because Disk Manager II puts them in the wrong place! To do this, enter the following for each of the nine screens.

n BLOCK UPDATE (where n is the screen number to be read from old disk)

FLUSH (after inserting the new disk - note: up to five screens may be entered at a time)

Now edit screen 3 of your new disk and add the following commands:

x DISK_SIZE ! (where x=180 or 360 as appropriate)
y DISK_HI ! (where y=x times 1, 2, 3 or 4 depending on the number of drives you have)

Unfortunately, TI FORTH does not provide a method for configuring each drive individually. Therefore, the user must be cognizant of which screens are available on each drive when there are differences between them.

At this point, FORTH can be booted and it will recognize the full capacity of your 180 or 360 SCRIN disk. You can create, edit, list, and load from screens greater than 89. However, neither Disk Manager nor FORTH-COPY will recognize this disk as having more than 90 screens. To fix this problem you must modify the -COPY screens (39 and 40), the disk header (sector 0) and, the SYS-SCRNS file header (sector 4).

First edit screen 39. Change the value 90, which appears once in DTEST and twice in FORTH-COPY to 180 or 360 as appropriate. Next, edit screen 40 as follows:

Line 3 - change 168 to 2D0 for 180 SCRIN or 5A0 for 360 SCRIN.

Line 4 - change 944 to 1244 for SSDD or DSDD (no change for DSSD).

Line 5 - replace entire line with:

```
DUP 1D + 2028 SWAP ! DUP 12 + a SWAP ! DUP 14 + 24 0 FILL
```

where a = 0201 for DSSD, 0102 for SSDD, or 0202 for DSDD.

Line 10 - change 165 to 2CD for 180 SCRIN, or 59D for 360 SCRIN.

Line 13 - change 4016 to C02C for 180 SCRIN, or C059 for 360 SCRIN.

Next edit screen 33 to modify the FORMAT-DISK word to:

```
: FORMAT-DISK 1 + a 33616 ! 18 SYSTEM ;
```

where a = 258 for DSSD, 513 for SSDD, 514 for DSDD.

Finally, you need to create a word that will modify the header sectors on your new disk. This word only needs to be executed once, since copies of this disk, once it is modified, will not require modification. Here is the way to do it:

```
HEX 0 DISK_LO ! ( removes disk fence )
: DD-FORTH 0 BLOCK UPDATE ( read screen 0 and mark as updated )
  DUP A + a SWAP ! ( a = 2D0 for 180 SCRIN, 5A0 for 360 SCRIN )
  DUP C + b SWAP ! ( b = 944 for DSSD, 1244 for SSDD or DSDD )
  DUP 10 + c SWAP ! ( c = 2D28 for all versions )
  DUP 12 + d SWAP ! ( d = 201 on DSSD, 102 on SSDD, 202 on DSDD )
  38 + C8 FF FILL ( flag all sectors as in use )

  1 BLOCK UPDATE ( read screen 1 and mark as updated )
  DUP E + f SWAP ! ( f = 2A0 for 180 SCRIN, 570 for 360 SCRIN )
  DUP 1C + g SWAP ! ( g = 4D20 for 180 or 360 SCRIN versions )
  DUP 1E + h SWAP ! ( h = 2805 for 180 SCRIN, 5205 for 360 SCRIN )
  20 + i SWAP ! ( i = F029 for 180 SCRIN, F059 for 360 SCRIN )
  FLUSH ; ( write modified screens to disk )
DECIMAL DD-FORTH
```

Now your new high capacity copy of FORTH is fully compatible with Disk Manager, the FORTH format, copy, test, and header words and your double density and/or double sided drives and controller. Enjoy!

Special thanks to Jim Vincent who is the FORTH columnist for the 99'ers Users Group Association.

Files in DVSO format to Forth.

by Mike Carroll
SYGOP,Tampa TIBBS

The following listing is a Forth program that will allow you to take Forth programs in DVBO format and convert them into runnable Forth screens. It is called FILTRN. Type this program into your Forth system using one of the editors available on your system disk. Then you can LOAD the program starting with the starting screen. Read the instructions carefully to get an understanding of how this program will work. You will have to change the lines that are on page 1 or screen one so that they load the proper help screens. You will need to look at your starting screen and figure out what number to substitute for the loading of the help screens in lines 6,7,8 or so of screen one where it says (POINT TO HELP SCREENS).

Take out everything but the actual program when trying to run it, which means get rid of these instructions and also where it asks for MORE?.etc. Study the TI FORTH manual if you have any questions. GO FORTH!!!!!!

(SET UP FILE WORDS AND POINT TO HELP SCREENS) (Page 1)

```
DECIMAL
68 LOAD FILE ( CALL UP -FILE SCREENS )

: HELPTRAM 118 LOAD ; ( POINT TO HELP SCREENS )
: HELPSCR 120 LOAD ; ( Change the # before )
: HELPPSK 122 LOAD ; ( each load if needed. )

: BEEP 52 GPLLNK ; ( GPL POKE FOR TONE ) -->
```


(-FTRAN2 - FORTH TO DIS/VAR 80 TRANSFER ROUTINES MCA 19JUL84)

HEX

(Page 2)

(SET OP PERIPHERAL ACCESS BLOCK)

```
0 VARIABLE FILBUF 50 ALLOT ( CREATE AN 80 [>50] CHR BUFFER )
  PABS @ A + ( DATA IS 10 [>A]BYTES INTO PAB )
  FILBUF 1900 ( PLACE THE BUFFER AT VDP >190D )
  FILE FILTRAN ( REFER TO THIS PAB AS FILTRAN )
```

(SET DISK FILE PARAMETERS)

```
: SETFILE FILTRAN SET-PAB ( CALL UP PAB AREA )
  SQNTL DSPLT ( SEQUENTIAL, DISPLAY, )
  VRBL 50 REC-LEN ; ( VARIABLE 80 [>50] ) -->
```

(-FTRAN2 - FORTH TO DIS/VAR 80 TRANSFER ROUTINES 16AUG84)
(DEFINE FILE OUTPUT ROUTINE) (Page 3)
DECIMAL

```
: PUTFILE CR ." INSERT DATA FILE DISK." CR KEY DROP
  APPND FILTRAN OPN ( OPEN THE PAB FOR OUTPUT )
  1+ SWAP DO I DUP CR ." COPIING SCREEN #" . CR
```

```
BLOCK 16 0 DO DUP FILBUF 80 BLANKS ( LOOP FOR 16 RECORDS )
  FILBUF 64 CHOVE ( COPT 64 CHAR FROM BUFR )
  64 WRT ( WRITE REC# I [64 CHRS] )
  64 + ( INC BUFFER ADDR [1 LW] )
  LOOP DROP LOOP ( BACK FOR NEXT RECORD )
  CLOSE ( CLOSE THE FILE )
```

BEEP ." DISK FILE COMPLETED." CR CR CR QUIT ; -->

(-FTRAN2 - FORTH TO DIS/VAR 80 TRANSFER ROUTINES 16AUG84)

(DEFINE FILE INPUT ROUTINE) (Page 4)

```
: GETFILE CR ." INSERT DATA FILE DISK." CR KEY DROP
  IMPT FILTRAN OPN ( OPEN THE PAB FOR INPUT )
  1+ SWAP DO I DUP CR ." LOADING SCREEN " . CR
  DUP BLOCK ( GET SCREEN ADDRESS )
  16 0 DO FILBUF 80 BLANKS ( LODP FOR 16 RECORDS )
  RD BROP ( READ REC FROM DISK )
  DUP FILBUF SWAP 64 CHOVE ( MOVE LINE INTO BUFFER )
  64 + ( INC BUFFER CHAR COUNT )
  LDOP BROP ( BACK FOR NEXT RECRD )
  BLOCK BROP UPDATE LOOP ( UPDATE THE DICTIONARY )
  CLOSE ( CLOSE THE FILE )
  BEEP ." ALL SCREENS LOADED." CR CR CR QUIT ; -->
```

(-FTRAN2 - FORTH TO DIS/VAR 80 TRANSFER ROUTINES 16AUG84)

(FORTS SCREEN TO DIS/VAR 80 FILE ROUTINE) (Page 5)

: SCR-BSK (scr#[e] scr#[e] ---)

```
CLS 9 3 GOTOXY ." SCREEN TO DISK TRANSFER"
  9 4 GOTOXY ." ====="
  CR CR CR CR CR
  SETFILE
  BEEP
  ." ENTER FILE DESCRIPTOR WORD"
  CR CR
  ." THEN TYPE 'PUTFILE'."
  CR
  QUIT ; -->
```

(-FTRAN2 - FORTH TO DIS/VAR 80 TRANSFER ROUTINES 16AUG84)

(DIS/VAR 80 FILE TO FORTH SCREEN ROUTINE) (Page 6)

: DSK-SCR (scr#[e] scr#[e] ---)

```
CLS 9 3 GOTOXY ." SCREEN TO DISK TRANSFER"
  9 4 GOTOXY ." ====="
  CR CR CR CR CR
  SETFILE
  BEEP
  ." ENTER FILE DESCRIPTOR WORD"
  CR CR
  ." THEN TYPE 'GETFILE'"
  CR
  QUIT ;
```

(-FTRAN2 - FORTH TO DIS/VAR 80 TRANSFER ROUTINES 16AUG84)

CLS 6 1 GOTOXY ." FORTH FILE TRANSFER SCREENS " (Page 7)

```
6 2 GOTOXY ." =====" CR CR CR CR CR
  ." THE -FTRAN SCREENS ADD SEVERAL NEW" CR
  ." WORDS TO YOUR DICTIONARY. THEY ARE:" CR CR
  ." SCR-DSK ( scr#[e] scr#[e] --- )" CR
  ." DSK-SCR ( scr#[e] scr#[e] --- )" CR
  ." HELPTRAN ( this screen )" CR
  ." HELPSCR ( aid for scr-dsk )" CR
  ." HELPSDK ( aid for dsk-scr )" CR
  ." GETFILE ( reads a disk file )" CR
  ." PUTFILE ( writes a disk file )" CR CR
  ." THESE WORDS ALLOW YOUR FORTH DISK" CR
  ." TO READ SCREENS WRITTEN BY TI-WRITER" CR
  ." OR DOWNLOADED WITH TE II VIA CTRL-2." CR CR
  BEEP ." HIT ANY KEY " KEY DROP -->
```

(-FTRAN2 - FORTH TO DIS/VAR 80 TRANSFER ROUTINES MCA 19JUL84)

CLS 25 0 GOTOXY ." HELPTRAN, cont." CR CR CR (Page B)

```
  ." IN ORDER TO COMPLETE A TRANSFER" CR
  ." BETWEEN DISK AND SCREEN YOU MUST" CR
  ." INVOKE ONE OTHER FORTH WORD - THE" CR
  ." TI-BOS FILE DESCRIPTOR. THE FILE" CR
  ." DESCRIPTOR PLACES THE DISK FILE NAME" CR
  ." INTO THE PAB AREA. FOR INSTANCE, TO" CR
  ." ACCESS THE FILE 'F-SCR1' ON DISK #2" CR
  ." YOU WOULD TYPE THE FOLLOWING:" CR CR
  ." F-D' DSK2.F-SCR1" CR CR
  ." NOTE: the single-quotes should be" CR
  ." replaced with double-quotes." CR CR
  ." FOR MORE ON THE FILE DESCRIPTOR" CR
  ." SEE CHAP. 8 OF THE TI FORTH MANUAL." CR CR
  BEEP 24 22 GOTOXY ." HIT ANY KEY " KEY DROP CLS 0 0 GOTOXY
```

(-FTRAN2 - FORTH TO DIS/VAR 80 TRANSFER ROUTINES 16AUG84)

(Page 9)

```
CLS 16 0 GOTOXY ." HELPSCR "
  16 1 GOTOXY ." =====" CR CR
  ." THE WORD 'SCR-BSK' WILL CHANGE A" CR CR
  ." FORTH SCREEN INTO A DIS/VAR 80 FILE." CR CR
  ." THE D/V 80 FILE CAN THEN BEEN READ BY" CR CR
  ." TI-WRITER, UPLOADED USING TE II, ETC." CR CR
  ." 'SCR-DSK' EXPECTS FROM AND TO SCREEN #" CR CR
  ." ON THE STACK. YOU WILL THEN WE ASKED" CR CR
  ." FOR THE PROPER FILE DESCRIPTOR. THEN" CR CR
  ." INSTRUCTED TO TYPE 'PUTFILE' ( DON'T" CR CR
  ." TYPE THE [']S ). FOR MORE ABOUT THE" CR CR
  BEEP ." FILE DESCRIPTOR TYPE 'HELPTRAN'" CR CR
  ." HIT A KEY " KEY DROP
  CLS 0 0 GOTOXY -->
```


Title Index Volume 6 1987

Title	Description	Author	Page #
40 column text mode	Demo program	Steven Peacock	1.09
A PIO port	Hardware review	Peter Schubert	8.07
A data base from scratch #1	Software hints	Chris Buttner	8.09
A data base from scratch #2	Software hints	Chris Buttner	11.16
A look at BASIC programs	Software hints	R.A. Green	4.15
A look at GPLLNK	Software hints	R.A. Green	2.13
	Software hints	R.A. Green	9.16
A look at speed	Software hints	R.A. Green	2.13
AND, a crazy way to add?	Software hints	Ben Takach	3.14
AT mini expansion system	Hardware review	Ross Mudie	9.05
AT-99 disk control system	Software review	John Paine and Peter Schubert	6.17
Adventure hints	Mystery funhouse		4.04
	Mystery funhouse		5.22
	Mystery funhouse		6.20
	Return to pirate's isle		7.06
	Return to pirate's isle		8.04
	Return to pirate's isle		9.11
	Return to pirate's isle		10.11
	Return to pirate's isle		11.27
Algorithm design	Software hints		8.18
All about .. MAX/RLE	Software hints	Steven Shraibman	1.24
An 8K-64K/256K PIO print buffer	Hardware review	John Paine	5.20
Analysis of sorting routines #1	Software hints	Jim Peterson	9.10
Analysis of sorting routines #2	Software hints	Jim Peterson	9.10
Another view of Assembler	Software hints	Arto Heino	7.17
Arrays and Sorts	Software hints	Jim Peterson	9.09
Assembler tit bit	Software hints	Ross Mudie	7.18
	Software hints	Shane Ferret	7.18
Back-up and flip!	General interest	Jim Peterson	8.13
Bug in RS232/SA	Hardware hint	Peter Schubert	9.11
CALL PLOT	Demo program	Arto Heino	4.19
Cartridge port information	Software hints	Mack McCormick	3.14
Co-ordinator's report	General news	Chris Buttner	2.03
	General news	Chris Buttner	3.02
	General news	Chris Buttner	4.02
	General news	Chris Buttner	5.02
	General news	Chris Buttner	6.02
	General news	Chris Buttner	7.02
	General news	Chris Buttner	8.02
	General news	Chris Buttner	9.02
	General news	Chris Buttner	10.02
	General news	Chris Buttner	11.02
Comments on Game Loader Program	Software hints	Geoff Trott	8.29
Comments on TI99/4A	General interest	Eric Whelan	6.23
Compound interest in multiplan	Software help	Steven Shraibman	10.26
Creating RLE screens from X/B	Demo program	Arto Heino	3.17
Customize TI-Writer	Word processing	Bob Pass	1.05
DM1000 modifications	Software hints		3.14
Debug and Extended BASIC	Software hints	George Meldrum	10.17
Debugging	Software hints	Jim Peterson	10.19
Debugs in de print	Software corrections	Shane Ferret	10.10
Diagnose hardware malfunctions	Hardware review	Geoff Trott	1.20
Digitizer pad for your 99/4A	Hardware review	Arto Heino	1.04
Disk controllers, how they work	General interest	Jerry Coffey	6.15
Don't sell them at all	General interest	Keir L. Wells	5.09
Double density disk controller	Hardware review	Peter Schubert	6.22
Editor/assembler auto start	Software hints	Mack McCormick	10.24
Enhanced Extended BASIC	Software review	Arto Heino	5.16
Ext/BASIC recovery after crash	Software hints	Arto Heino	5.16
Extended BASIC Tutorial	Software hints	Ross Mudie	7.12
Fix for excessive PEB voltages	Hardware hint		8.08
Flipping single-sided floppies	Hardware review		1.10
Forti music system	Music in Forth	Arto Heino	2.15
Forum	!@P- and !@P+ in X/B		5.14
	A4 paper with TI-Writer	Percy H.	4.12
	CALL INIT, LOAD, LINK		10.18
	COMPACT, MM and V2.2	Kevin Cox	4.12
	Chess module problems		6.20
	Convert TI files to PC	John Kerr	3.19
	Corrections-A4 paper		5.14
	Cursor colour		6.20

Title	Description	Author	Page #
Forum	Find program entry name		10.18
	Incompatible hardware	Tony S.	5.14
	Linking to GPL routines	George Meldrum	6.20
	Minimemory battery	D. S.	5.14
	Modem problems	Les Andrews	3.19
	Postage costs from shop	H.Hull	4.12
	TI-Writer commands		8.25
	Transfer from TI-Writer	Peter C.	5.14
	Uploading files to BBS	Joseph P.	3.20
Freeware version of TI99-OPOLY	Update1-6	Ross Mudie	1.22
From BASIC to Extended BASIC	Software hints	Ben Takach	4.17
GROMs, what are they?	General interest	John Paine	5.21
Graphics	Software hints	Jim Peterson	7.19
Hackers' new year resolutions	Humorous	Robert Brown	1.08
Hardware news	Club news	Peter Schubert	10.03
Hardware products	Hardware review	Peter Schubert	8.06
Horizon RAM disk hints	Hardware hint	Ross Mudie	5.02
How to condense Assembly in X/B	Software hints	Ross Mudie	8.20
I am computer, hear me roar	Software hints	Barb Berg	5.17
IBM compatibility for the 99/4A	Hardware review		2.03
Illawarra coordinators report	Club affairs		3.20
Imaginative programming	Software hints	Jim Peterson	10.16
Joystick convertor	Hardware hint	Laurie Marsh	11.11
Joystick sampling	Demo program	Arto Heino	6.22
Keyboard matrix	Correction	Ben Takach	4.18
	Hardware hint	Ben Takach	3.23
Let's round up the mavericks!	General interest	Jim Peterson	9.02
Letter	Auction	Lou Amadio	10.18
	Brisbane TI-Faire	G. Christensen	8.02
	Geoff Trott's travels	Lou Amadio	8.03
	Publicity	Shane Andersen	11.03
	Thanks for help	Bill Walker	11.03
	Thanks to John Paine	Noel Richards	8.03
	Users in Philadelphia	Geoff Trott	8.03
	Club affairs		1.19
Letters to Editor	Peeker for DSR ROMs	Ross Mudie	1.06
Link-it #10	Force garbage collect'n	Ross Mudie	2.12
Link-it #11	Scan joystick quickly	Ross Mudie	6.07
Link-it #12	Control simple devices	Ross Mudie	10.08
Link-it #13	Transfer tape and disk	Ross Mudie	11.09
Link-it #14	Software hints	George Meldrum	3.23
Loading large m. 1. files	Software hints	George Meldrum	10.17
Loading memory image files			
Machine code in BASIC programs	Software hints	George Meldrum	10.09
Mechatronic advertisement	Hardware review	Ben Takach	1.23
Micro examination	New products	Shane Andersen	1.03
Midi, music and the TI99/4A	Hardware proposal	Arto Heino	1.14
More on modules	General interest	D.N. Harris	10.24
More plots	Demo program	Arto Heino	6.22
Multicolor mode	Demo program	Steven Peacock	2.16
Multifunction card	Hardware review	Peter Schubert	11.11
Music hardware	Hardware review	Arto Heino	5.16
New flight simulator program	Software review		3.18
No apple for this teacher	General interest	Phil Thomson	4.02
Number speaker	Software hints	Jim Peterson	9.23
ON ERROR from SUBroutine	Software hints	Ross Mudie	9.16
On modules for the TI99/4A	General interest	D.N. Harris	6.21
PIO expansion port	Hardware review	Peter Schubert	5.22
PIO port for mini-PE system	Hardware review	Peter Schubert	9.07
Picasso	Software review	Arto Heino	5.16
Picasso publisher version 1.3	Software hints		9.04
Picasso tips - version 2.0	Software help	Arto Heino	10.11
Piracy	General interest	Jim Peterson	6.02
Pretty please, p m d a S r	Software hints	Jim Peterson	11.26
Producing the TND	Word processing		3.04
Program to type in	28-column converter	Jim Peterson	7.10
	28-column printer	McGovern	10.12
	48K joy draw	Marcello Zannini	2.09
	99 artist	Marcello Zannini	2.08
	Astronomy		9.12
	At the beach	Milo Tsukroff	1.139
	Automatic mouse maze	Jim Peterson	1.137
	Basic number facts	Col Christensen	10.13
	Beam headings for Hams	Arto Heino	5.19
	Bombs away		2.09

Title	Description	Author	Page #
Program to type in	Bowling champ		4.08
	Bridge guard	Mark Smetana	4.08
	Bubble	R.S. Kemmetmueller	4.11
	Cassette sleeve maker	C.A. Provance	1.138
	Caverns of carnage		3.11
	Chase		5.12
	Compressor/REM remover		2.20
	CrazyfrazeeII	Paul Yorke	3.10
	CreateDBF	Chris Buttner	8.14
	Disk label printer		11.15
	Disk sleeve printer	Randy Baxter and Cecil Crowder	1.138
	Don Quixo-TI	John Clulow and Norma Clulow	3.12
	Frogger		10.15
	Frogling		4.10
	Game loader		8.08
	Grumplin		6.10
	Guess a letter/number	Vincent Maker	2.10
	Hammurabi		5.10
	Hangman	Colin Lee	7.08
	Hyphen. fill and adjust	Jim Peterson	7.10
	Jaw breaker		11.13
	Label printer		8.16
	Menu maker		11.15
	Multiplication invaders	David Perkovic	6.11
	Musical kaleidoscope		9.15
	One-line mortgage calc.	Sam Morabito	1.15
	PT109	George Yarmoluk	8.15
	Phantom5		6.12
	RTTY on the TI99/4A	G. Wilson and D. Crawford	8.11
	Re-act		9.14
	Sea diver		5.13
	Sea diver		6.13
	Shaded velvet	Jeff Gatlin	1.140
	Sheepdog	S. Bonnett	8.17
	Spider bop		10.13
	The boogens	W.M. Johnson	11.12
	The file manager		2.07
	Tigercub anagrammer	Jim Peterson	1.137
	Tigercub printall	Jim Peterson	7.11
	Two line mazamund	Dennis Hodgson	1.15
	XB screen colors		2.07
	Zapper zone	Iain Johnson	11.14
Programmers	General interest		4.15
Programming music	Software hints	Bob Montgomery	1.12
Programs that write programs #1	Software hints	Jim Peterson	8.18
Programs that write programs #2	Software hints	Jim Peterson	9.25
Programs that write programs #3	Software hints	Jim Peterson	10.25
Publications report	Club news	Terry Phillips	3.03
	Club news	Warren Welham	11.05
Putting it all together #1	Software hints	Jim Peterson	9.24
Putting it all together #2	Software hints	Jim Peterson	10.20
Putting it all together #3	Software hints	Jim Peterson	11.23
RS232 interface box with 32K	Hardware review	Peter Schubert	2.05
Record Length Encoded files	Software hints	Arto Heino	1.15
Redefine cursor shape in X/B	Software hints	Ross Mudie	10.24
Regional group news	General interest		1.22
	General interest		2.20
	General interest		3.09
	General interest		4.19
	General interest		5.23
	General interest		6.18
	General interest		7.02
	General interest		7.07
	General interest		8.31
	General interest		9.27
	General interest		10.27
	General interest		11.27
Removing protection	Software hints	George Meldrum	11.17
Repair services for TI99/4A	Hardware hints	Ross Mudie	5.06
SAVE tutorial	Software hints	Scott Darling	6.19
Secretary's notebook	Club news	Terry Phillips	2.04
	Club news	Terry Phillips	3.03
	Club news	Terry Phillips	4.03
	Club news	Terry Phillips	5.03
	Club news	Terry Phillips	6.03
	Club news	Terry Phillips	7.03
	Club news	Terry Phillips	8.03
	Club news	Terry Phillips	10.03
	Club news	Terry Phillips	11.03



TISHUG NEWS DIGEST

Title	Description	Author	Page #
Sending mail on Texpac BBS	BBS information	Ross Mudie	3.19
Software column	Club software	Terry Phillips	1.16
	Club software	Terry Phillips	2.06
	Club software	Terry Phillips	3.08
	Club software	Terry Phillips	4.07
	Club software	Terry Phillips	5.08
	Club software	Terry Phillips	6.06
	Club software	Terry Phillips	7.03
	Club software	Terry Phillips	8.04
	Club software	Terry Phillips	10.06
	Club software	Terry Phillips	11.06
Some formatter tricks	Word processing	Jean Wilcox	11.25
Sprites part 1	Software hints	Jim Peterson	10.23
Sprites part 2	Software hints	Jim Peterson	11.18
Stringing and unstringing	Software hints	Jim Peterson	11.24
Super widget	Hardware review		2.11
TI-Writer bugs	Word processing	Jim Peterson	8.24
TI-Writer commands	Word processing		8.27
TI-Writer file printer	Word processing	George Steffen	11.25
TI-Writer help	Word processing	Tom Kennedy	8.21
TI-Writer spricht auch deutsch!	Word processing	Ben Takach	2.19
TI99-opoly updates of fairware	Software hints	Ross Mudie	9.04
TI99/4A Intern	Book review	Ben Takach	1.04
Ticom	Comic strip	Arto Heino	1.11
	Comic strip	Arto Heino	5.19
	Comic strip	Arto Heino	11.27
TisHUG shop	For sale	Chris Buttner	1.17
	For sale	Cyril Bohlsen	2.11
	For sale	Cyril Bohlsen	3.07
	For sale	Cyril Bohlsen	4.04
	For sale	Cyril Bohlsen	5.04
	For sale	Cyril Bohlsen	6.04
	For sale	Cyril Bohlsen	7.07
	For sale	Cyril Bohlsen	8.05
	For sale	Cyril Bohlsen	9.03
	For sale	Cyril Bohlsen	10.05
	For sale	Cyril Bohlsen	11.05
TI99 software competition	Club news		9.19
	Club news		10.02
	Club news		11.02
TND index	Club news	Brian Graham	1.21
	For sale	Brian Graham	2.14
Techo time	Audio amplifier	John Paine	11.04
	Cleaning disk drives?	John Paine	6.05
	Cleaning disk drives?	John Paine	7.06
	Console p. sup. upgrade	John Paine	10.04
	Corrections to V5.8	Robert Peverill	2.04
	Fault of the month	John Paine	3.06
	Fault of the month	John Paine	4.05
	Fault of the month	John Paine	5.07
	Fault of the month	John Paine	6.05
	Fault of the month	John Paine	7.05
	Fault of the month	John Paine	11.04
	Hardware review	John Paine	8.07
	Using the LX86 printer	Robert Peverill	1.18
	New modules	John Paine	10.04
	Plans for 1987	John Paine	2.05
	RAMdisk constr. problem	John Paine	5.07
	RAMdisk constr. tip	John Paine	6.05
	RAMdisk progress	John Paine	3.05
	RAMdisk software hint	John Paine	11.04
	RAMdisk software tips	John Paine	10.04
	RAMdisk with one floppy	John Paine	7.06
	Remedy for alphalock	John Paine	4.06
	Speech synth in console	John Paine	7.05
Texpac BBS, an overview	General interest	Ross Mudie	11.08
Text manipulation	Software hints	D.N. Harris	9.20
The 4A EPROM experiment	General interest	Arto Heino	6.14
The Mechatronic 80 column card	Hardware review	Ben Takach	8.28
The TI99/4A scene in the USA	General interest	Ben Takach	7.13
The communicators	BBS information	Shane Andersen	1.09
	BBS information	Ross Mudie	2.04
	BBS information	Ross Mudie	4.14
	BBS information	Ross Mudie	5.06
	BBS information	Ross Mudie	6.09
	BBS information	Ross Mudie	7.14
	BBS information	Ross Mudie	8.19
	BBS information	Ross Mudie	9.06
	BBS information	Ross Mudie	10.07
	BBS information	Ross Mudie	11.07
The console tester	Hardware review	Geoff Trott	3.15
The matchbox tester	Hardware review	Ben Takach	3.22
The music corner	Software hints	Jeff Gatlin	3.21

Title	Description	Author	Page #
Those confusing variable names	Software hints	Jim Peterson	10.26
Tidbits five	Software hints	Wade Bowmer	9.17
Tidbits six	Software hints	Wade Bowmer	11.19
Tigercub tips #42	Software hints	Jim Peterson	7.15
Tigercub tips #43	Software hints	Jim Peterson	9.21
Tigercub tips #44	Software hints	Jim Peterson	10.21
Tigercub tips #45	Software hints	Jim Peterson	11.21
Tips from the Tigercub	Software hints	Jim Peterson	1.15
Using CTRL(2) with TEII	Software hints		11.26
Using Pilot 99	Software hints	Arto Heino	4.13
Windows for your 99/4A	Sample program	Arto Heino	1.14
Word wrap and fill	Demo program		9.26
Workshop tutorial	Club affairs		5.05
Younger set	Adventure hints	Richard Martin	1.11
	Adventure hints	Vincent Maker	9.08
	Games hints	Kresho Sprem	1.11
	Games scores, program	David Perkovic	5.15
	Picture	Adam Sutherland	4.16
	Program/adventure hints	Vincent Maker	2.10
	Program/adventure hints	Vincent Maker	6.04
	Program computer dice	Peter Mudie	1.10
	Program guesstheno	Peter Mudie	10.11
	Software hints	Joshua Rust	9.08

Author Index Volume 6 1987

Author - source	Title	Description	Page #
Adam Sutherland	Younger set	Picture	4.16
Arto Heino	Another view of Assembler	Software hints	7.17
	CALL PLOT	Demo program	4.19
	Creating RLE screens from X/B	Demo program	3.17
	Digitizer pad for your 99/4A	Hardware review	1.04
	Enhanced Extended BASIC	Software review	5.16
	Ext/BASIC recovery after crash	Software hints	5.16
	Forti music system	Music in Forth	2.15
	Joystick sampling	Demo program	6.22
	Midi, music and the TI99/4A	Hardware proposal	1.14
	More plots	Demo program	6.22
	Music hardware	Hardware review	5.16
	Picasso	Software review	5.16
	Picasso tips - version 2.0	Software help	10.11
	Program to type in	Beam headings for Hams	5.19
	Record Length Encoded files	Software hints	1.15
	TIcom	Comic strip	1.11
	TIcom	Comic strip	5.19
	TIcom	Comic strip	11.27
	The 4A EPROM experiment	General interest	6.14
	Using Pilot 99	Software hints	4.13
	Windows for your 99/4A	Sample program	1.14
Barb Berg	I am computer, hear me roar	Software hints	5.17
Ben Takach	AND, a crazy way to add?	Software hints	3.14
	From BASIC to Extended BASIC	Software hints	4.17
	Keyboard matrix	Correction	4.18
	Keyboard matrix	Hardware hints	3.23
	Mechatronic advertisement	Hardware review	1.23
	TI-Writer spricht auch deutsch!	Word processing	2.19
	TI99/4A Intern	Book review	1.04
	The Mechatronic 80 column card	Hardware review	8.28
	The TI99/4A scene in the USA	General interest	7.13
	The matchbox tester	Hardware review	3.22
Bill Walker	Letter	Thanks for help	11.03
Bob Montgomery	Programming music	Software hints	1.12
Bob Pass			
99'er ONLINE	Customize TI-Writer	Word processing	1.05
Brian Graham	TND index	Club news	1.21
	TND index	For sale	2.14
C.A. Provance			
N.H. 99ers	Program to type in	Cassette sleeve maker	1.138
Cecil Crowder	Program to type in	Disk sleeve printer	1.138



TISHUG NEWS DIGEST

Author - source	Title	Description	Page #	
Chris Buttner	A data base from scratch #1	Software hints	8.09	
	A data base from scratch #2	Software hints	11.16	
	Co-ordinator's report	General news	2.03	
	Co-ordinator's report	General news	3.02	
	Co-ordinator's report	General news	4.02	
	Co-ordinator's report	General news	5.02	
	Co-ordinator's report	General news	6.02	
	Co-ordinator's report	General news	7.02	
	Co-ordinator's report	General news	8.02	
	Co-ordinator's report	General news	9.02	
	Co-ordinator's report	General news	10.02	
	Co-ordinator's report	General news	11.02	
	Program to type in	CreateDBF	8.14	
	TiSHUG shop	For sale	1.17	
Col Christensen	Program to type in	Basic number facts	10.13	
Colin Lee	Program to type in	Hangman	7.08	
Cyril Bohlsen	TiSHUG shop	For sale	2.11	
	TiSHUG shop	For sale	3.07	
	TiSHUG shop	For sale	4.04	
	TiSHUG shop	For sale	5.04	
	TiSHUG shop	For sale	6.04	
	TiSHUG shop	For sale	7.07	
	TiSHUG shop	For sale	8.05	
	TiSHUG shop	For sale	9.03	
	TiSHUG shop	For sale	10.05	
	TiSHUG shop	For sale	11.05	
	David Perkovic	Program to type in	Multiplication invaders	6.11
		Younger set	Games scores, program	5.15
D. Crawford	Program to type in	RTTY on the TI99/4A	8.11	
D.N. Harris	More on modules	General interest	10.24	
	On modules for the TI99/4A	General interest	6.21	
	Text manipulation	Software hints	9.20	
D. S.	Forum	Minimemory battery	5.14	
Dennis Hodgson	Program to type in	Two line mazamynd	1.15	
Eric Whelan	Comments on TI99/4A	General interest	6.23	
G. Christensen	Letter	Brisbane TI-Faire	8.02	
G. Wilson	Program to type in	RTTY on the TI99/4A	8.11	
Geoff Trott	Comments on Game Loader Program	Software hints	8.29	
	Diagnose hardware malfunctions	Hardware review	1.20	
	Letter	Users in Philadelphia	8.03	
	The console tester	Hardware review	3.15	
George Meldrum	Debug and Extended BASIC	Software hints	10.17	
	Forum	Linking to GPL routines	6.20	
	Machine code in BASIC programs	Software hints	10.09	
	Loading memory image files	Software hints	10.17	
	Removing protection	Software hints	11.17	
	Loading large m. l. files	Software hints	3.23	
George Steffen				
L.A. 99ers	TI-Writer file printer	Word processing	11.25	
George Yarmoluk	Program to type in	PT109	8.15	
H.Hull	Forum	Postage costs from shop	4.12	
Iain Johnson				
Channel 99	Program to type in	Zapper zone	11.14	
Jean Wilcox				
SunCoast 99ers	Some formatter tricks	Word processing	11.25	
Jeff Gatlin	Program to type in	Shaded velvet	1.140	
NE Tarrant UG	The music corner	Software hints	3.21	
Jerry Coffey				
NW Ohio 99er	Disk controllers, how they work	General interest	6.15	
Jim Peterson	Analysis of sorting routines #1	Software hints	9.10	
	Analysis of sorting routines #2	Software hints	9.10	
Tigercub soft.	Arrays and Sorts	Software hints	9.09	
	Back-up and flip!	General interest	8.13	
	Debugging	Software hints	10.19	
	Graphics	Software hints	7.19	
	Imaginative programming	Software hints	10.16	
	Let's round up the mavericks!	General interest	9.02	
	Number speaker	Software hints	9.23	
	Piracy	General interest	6.02	
	P p p m d a s r	Software hints	11.26	
	Program to type in	28-column converter	7.10	
	Program to type in	Automatic mouse maze	1.137	
	Program to type in	Hyphen. fill and adjust	7.10	
	Program to type in	Tigercub anagrammer	1.137	
	Program to type in	Tigercub printall	7.11	
	Programs that write programs #1	Software hints	8.18	
	Programs that write programs #2	Software hints	9.25	
	Programs that write programs #3	Software hints	10.25	



TISHUG NEWS DIGEST

Author - source	Title	Description	Page #
Jim Peterson	Putting it all together #1	Software hints	9.24
Tigercub soft.	Putting it all together #2	Software hints	10.20
	Putting it all together #3	Software hints	11.23
	Sprites part 1	Software hints	10.23
	Sprites part 2	Software hints	11.18
	Stringing and unstringing	Software hints	11.24
	TI-Writer bugs	Word processing	8.24
	Those confusing variable names	Software hints	10.26
	Tigercub tips #42	Software hints	7.15
	Tigercub tips #43	Software hints	9.21
	Tigercub tips #44	Software hints	10.21
	Tigercub tips #45	Software hints	11.21
	Tips from the Tigercub	Software hints	1.15
John Kerr	Forum	Convert TI files to PC	3.19
John Paine	An 8K-64K/256K PIO print buffer	Hardware review	5.20
	GROMs, what are they?	General interest	5.21
	Techo time	Audio amplifier	11.04
	Techo time	Cleaning disk drives?	6.05
	Techo time	Cleaning disk drives?	7.06
	Techo time	Console p. sup. upgrade	10.04
	Techo time	Fault of the month	3.06
	Techo time	Fault of the month	4.05
	Techo time	Fault of the month	5.07
	Techo time	Fault of the month	6.05
	Techo time	Fault of the month	7.05
	Techo time	Fault of the month	11.04
	Techo time	Hardware review	8.07
John Paine	Techo time	New modules	10.04
	Techo time	Plans for 1987	2.05
	Techo time	RAMdisk constr. problem	5.07
	Techo time	RAMdisk constr. tip	6.05
	Techo time	RAMdisk progress	3.05
	Techo time	RAMdisk software hint	11.04
	Techo time	RAMdisk software tips	10.04
	Techo time	RAMdisk with one floppy	7.06
	Techo time	Remedy for alphalock	4.06
	Techo time	Speech synth in console	7.05
	AT-99 disk control system	Software review	6.17
John Clulow	Ottawa	Program to type in	3.12
Joseph P.	Forum	Don Quixo-TI	3.20
Joshua Rust	Younger set	Uploading files to BBS	9.08
Keir L. Wells	Don't sell them at all	Software hints	9.08
Kevin Cox	Forum	General interest	5.09
Kresho Sprem	Younger set	COMPACT, MM and V2.2	4.12
		Games hints	1.11
Laurie Marsh	Joystick convertor	Hardware hints	11.11
Les Andrews	Forum	Modem problems	3.19
Lou Amadio	Letter	Auction	10.18
	Letter	Geoff Trott's travels	8.03
Mack McCormick	Cartridge port information	Hardware hints	11.11
U.S.A.	Editor/assembler auto start	Software hints	3.14
Marcello Zannini	Program to type in	Software hints	10.24
	Program to type in	48K joy draw	2.09
	Program to type in	99 artist	2.08
Mark Smetana	Program to type in	Bridge guard	4.08
McGovern	Funnelweb Farm	Program to type in	28-column printer
Milo Tsukroff	Program to type in	At the beach	10.12
			1.139
Noel Richards	Letter	Thanks to John Paine	8.03
Norma Clulow	Ottawa	Program to type in	Don Quixo-TI
			3.12
Paul Yorke	Florida	Program to type in	CrazyfrazeeII
Percy H.	Forum	A4 paper with TI-Writer	3.10
Peter C.	Forum	Transfer from TI-Writer	4.12
Peter Mudie	Younger set	Program computer dice	5.14
	Younger set	Program guesstheno	1.10
			10.11
Peter Schubert	A PIO port	Hardware review	8.07
	AT-99 disk control system	Software review	6.17
	Bug in RS232/SA	Hardware hints	9.11
	Double density disk controller	Hardware review	6.22
	Hardware news	Club news	10.03
	Hardware products	Hardware review	8.06
	Multifunction card	Hardware review	11.11
	PIO expansion port	Hardware review	5.22
	PIO port for mini-PE system	Hardware review	9.07
	RS232 interface box with 32K	Hardware review	2.05
Phil Thomson	No apple for this teacher	General interest	4.02

Author - source	Title	Description	page #
R.A. Green	A look at BASIC programs	Software hints	4.15
Ottawa TI99/4A	A look at GPLLNK	Software hints	2.13
User's Group	A look at GPLLNK	Software hints	9.16
	A look at speed	Software hints	2.13
R.S.Kemmetmueller	Program to type in	Bubble	4.11
Randy Baxter	Program to type in	Disk sleeve printer	1.138
Richard Martin	Younger set	Adventure hints	1.11
Robert Brown	Hackers' new year resolutions	Humourous	1.08
Robert Peverill	Techo time	Corrections to V5.8	2.04
	Techo time	Using the LX86 printer	1.18
Ross Mudie	AT mini expansion system	Hardware review	9.05
	Assembler tit bit	Software hints	7.18
	Extended BASIC Tutorial	Software hints	7.12
	Freeware version of TI99-OPOLY	Updatel-6	1.22
	Horizon RAM disk hints	Hardware hints	5.02
	How to condense Assembly in X/B	Software hints	8.20
	Link-it #10	Peeker for DSR ROMs	1.06
	Link-it #11	Force garbage collection	2.12
	Link-it #12	Scan joystick quickly	6.07
	Link-it #13	Control simple devices	10.08
	Link-it #14	Transfer tape and disk	11.09
	ON ERROR from SUBroutine	Software hints	9.16
	Redefine cursor shape in X/B	Software hints	10.24
	Repair services for TI99/4A	Hardware hints	5.06
	Sending mail on Texpac BBS	BBS information	3.19
Ross Mudie	TI99-opoly updates of fairware	Software hints	9.04
	Texpac BBS, an overview	General interest	11.08
	The communicators	BBS information	2.04
	The communicators	BBS information	4.14
	The communicators	BBS information	5.06
	The communicators	BBS information	6.09
	The communicators	BBS information	7.14
	The communicators	BBS information	8.19
	The communicators	BBS information	9.06
	The communicators	BBS information	10.07
	The communicators	BBS information	11.07
S. Bonnett	Program to type in	Sheepdog	8.17
Sam Morabito	Program to type in	One-line mortgage calc.	1.15
Scott Darling	SAVE tutorial	Software hints	6.19
Shane Andersen	Letter	Publicity	11.03
	Micro examination	New products	1.03
	The communicators	BBS information	1.09
Shane Ferret	Assembler tit bit	Software hints	7.18
	Debugs in de print	Software corrections	10.10
Steven Peacock	40 column text mode	Demo program	1.09
West JAX 99ers	Multicolor mode	Demo program	2.16
Steven Shraibman	All about .. MAX/RLE	Software hints	1.24
	Compound interest in multiplan	Software help	10.26
Terry Phillips	Publications report	Club news	3.03
	Secretary's notebook	Club news	2.04
	Secretary's notebook	Club news	3.03
	Secretary's notebook	Club news	4.03
	Secretary's notebook	Club news	5.03
	Secretary's notebook	Club news	6.03
	Secretary's notebook	Club news	7.03
	Secretary's notebook	Club news	8.03
	Secretary's notebook	Club news	10.03
	Secretary's notebook	Club news	11.03
	Software column	Club software	1.16
	Software column	Club software	2.06
	Software column	Club software	3.08
	Software column	Club software	4.07
	Software column	Club software	5.08
	Software column	Club software	6.06
	Software column	Club software	7.03
	Software column	Club software	8.04
	Software column	Club software	10.06
	Software column	Club software	11.06
Tom Kennedy			
Boston C.S.	TI-Writer help	Word processing	8.21
Tony S.	Forum	Incompatible hardware	5.14
Vincent Maker	Program to type in	Guess a letter/number	2.10
	Younger set	Adventure hints	9.08
	Younger set	Program/adventure hints	2.10
	Younger set	Program/adventure hints	6.04
W.M. Johnson			
Channel 99	Program to type in	The boogens	11.12
Wade Bowmer	Tidbits five	Software hints	9.17
Wade Bowmer	Tidbits six	Software hints	11.19
Warren Welham	Publications report	Club news	11.05



TISHUG NEWS DIGEST

Author - source	Title	Description	Page #	
Unknown author Source:-	Adventure hints	Mystery funhouse	4.04	
	Adventure hints	Mystery funhouse	5.22	
	Adventure hints	Mystery funhouse	6.20	
	Adventure hints	Return to pirate's isle	7.06	
	Adventure hints	Return to pirate's isle	8.04	
	Adventure hints	Return to pirate's isle	9.11	
	Adventure hints	Return to pirate's isle	10.11	
	Adventure hints	Return to pirate's isle	11.27	
	Algorithm design	Software hints	8.18	
	Houston Users	DM1000 modifications	Software hints	3.14
	Sutherland R.G.	Fix for excessive PEB voltages	Hardware hints	8.08
Flipping single-sided floppies		Hardware review	1.10	
Forum		!@P- and !@P+ in X/B	5.14	
Forum		CALL INIT, LOAD, LINK	10.18	
Forum		Chess module problems	6.20	
Forum		Corrections-A4 paper	5.14	
Forum		Cursor colour	6.20	
Forum		Find program entry name	10.18	
Forum		TI-Writer commands	8.25	
GENie BBS		IBM compatibility for the 99/4A	Hardware review	2.03
		Illawarra coordinators report	Club affairs	3.20
	Letters	Club affairs	1.19	
	New flight simulator program	Software review	3.18	
	Picasso publisher version 1.3	Software hints	9.04	
	Producing the TND	Word processing	3.04	
	Program to type in	Astronomy	9.12	
	Program to type in	Bombs away	2.09	
	Program to type in	Bowling champ	4.08	
	Program to type in	Caverns of carnage	3.11	
	Program to type in	Chase	5.12	
	Program to type in	Compressor/REM remover	2.20	
	Program to type in	Disk label printer	11.15	
	ATICC	Program to type in	Frogger	10.15
		Program to type in	Frogling	4.10
	Clubline mag.	Program to type in	Game loader	8.08
		Program to type in	Grumplin	6.10
Program to type in		Hammurabi	5.10	
Program to type in		Jaw breaker	11.13	
Program to type in		Label printer	8.16	
HCM	Program to type in	Menu maker	11.15	
	Program to type in	Musical kaleidoscope	9.15	
	Program to type in	Phantom5	6.12	
	Program to type in	Re-act	9.14	
	Program to type in	Sea diver	5.13	
	Program to type in	Sea diver	6.13	
	Program to type in	Spider bop	10.13	
Inter-8 Tacoma 99ers	Program to type in	The file manager	2.07	
	Program to type in	XB screen colors	2.07	
	Programmers	General interest	4.15	
	Regional group news	General interest	1.22	
	Regional group news	General interest	2.20	
	Regional group news	General interest	3.09	
	Regional group news	General interest	4.19	
	Regional group news	General interest	5.23	
	Regional group news	General interest	6.18	
	Regional group news	General interest	7.02	
	Regional group news	General interest	7.07	
	Regional group news	General interest	8.31	
	Regional group news	General interest	9.27	
	Regional group news	General interest	10.27	
	Regional group news	General interest	11.27	
	Victoria 99ers	Super widget	Hardware review	2.11
	Illinois 99ers	TI-Writer commands	Word processing	8.27
		TiSHUG software competition	Club news	9.19
		TiSHUG software competition	Club news	10.02
TiSHUG software competition		Club news	11.02	
Using CTRL(2) with TEII		Software hints	11.26	
Subfile99	Word wrap and fill	Demo program	9.26	
	Workshop tutorial	Club affairs	5.05	

For sale - contact Richard Exton on (02)727 7061 after 6pm.
Hardware (with manuals)
 TI power supply
 TI external drive with transformer
 TI internal drive for PE box
 TI disk controller card
 TI speech synthesizer
 Cicada 300 baud modem with phone
 Dual RS232 adapter cables
 2 trigger style joysticks
 TI joystick converter
 TI white keyboard

Cartridge - games
 Video chess; Alpina; Microsurgeon
 Return to pirate's isle
 Congo bongo; Buck Rogers
 TI-Invaders (NWO); Parsec; Munchman
 Indoor soccer; Saturday night bingo
Cartridge - utility
 TI-Writer; Mini Memory
 Microsoft Multiplan
 Editor - assembler
 Personal Record Keeping
 Disk Manager II

Disks
 Programming aids I
 Checkbook manager; Text to speech
 Editor - assembler
 Graphing package
 Microsoft Multiplan; TI-Writer
 Auto Spell checker; Mini Memory
 Graphs; Reformatter; Lasso
 Diskfixer

Regional Group News

STOP PRESS

The information presented here is obtained from the BBS. Some Regional groups are not advising of their meetings, which makes the maintenance of this file on the BBS very difficult.

Meeting summary.

TISHUG AGM	6/2/88	Burwood RSL
Glebe	11/2/88	Glebe
Illawarra	15/2/88	Keiraville
Carlingford	17/2/88	?

TISHUG.

ANNUAL GENERAL MEETING, Election of 5 directors, 1.30pm on 6th February 1988, at BURWOOD RSL, 96 Shaftesbury Rd, Burwood.

Other Regular meetings are normally at 2pm on the first Saturday the month, except January, at the Woodstock Community Centre, Church St., Burwood.

GLEBE Regional Group.

The next meeting is on 18th February, 8pm, 43 Boyce St, Glebe. Contact Mike Stattery; (02 692 0559).

Regular meetings are normally on the Thursday evening following the first Saturday of the month.

The activities at these regional meetings are rather informal and include looking at new hardware, hardware repairs, looking at new software and having a general chat.

LIVERPOOL Regional Group

Contact Arto Heino (02 603 8956) for more information.

Regular meeting date is the Friday following the TISHUG general meeting (first Saturday) at 7.30pm.

CENTRAL COAST Regional Group.

Meetings are normally held on the second Saturday of each month at 6.30pm at

Toukley Tennis Club hall,
Header St, Toukley.

Contact Russell Welham (043 92 4000)

CARLINGFORD Regional Group.

The next meeting is on Wednesday 17th February at 7.30pm

Contact Chris Buttner, 79 Jenkins Rd, Carlingford, (02 871 7753), for more information.

Regular meetings are usually on the third Wednesday of each month.

ILLAWARRA Regional Group.

The next meeting is on Monday 15th February at 7.30pm, Keiraville Public School, Gippe Rd, Keiraville, opposite the Keiraville shopping centre.

Contact Bob Montgomery on (042 28 6463) for more information.

Regular meetings are normally on the third Monday of each month, except January.

N. K. SUBURBS Regional Group.

Contact Mike Norman on (02 452 3920) or Dick Burton on (02 418 8132) for further information.

Regular meetings are on the third or fourth Thursday of the month.

SUTHERLAND Regional Group.

Contact Gary Wilson, user name VK2YGW on this BBS or ring Gary on (02 543 1314) for information.

BANANA COAST Regional Group

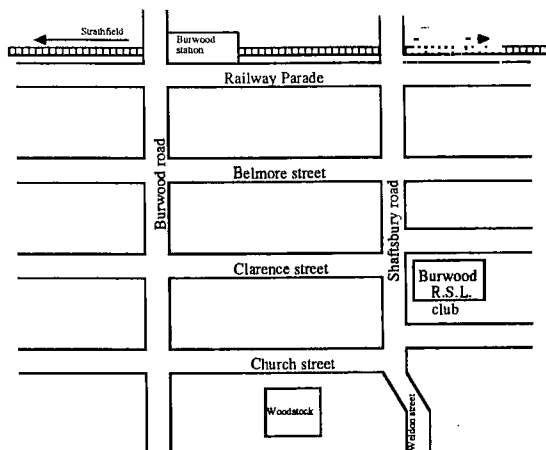
(Coffs Harbour area)

For information on meetings of the Banana Coast group, contact Keir Wells at 9 Tamarind Drive, Bellingen, phone (066 55 1487).

All TISHUG Regional groups are invited to submit items for this department. Send details as system mail to SYSOP.

A. G. !!! A. G. !!!

The Burwood RSL Club has been booked from 1pm until 4pm. The directors would like the actual meeting to start promptly at 1.30pm, so everyone should aim to be at the club by 1pm to sign in and be seated before 1.30pm. Please note that the RSL club has a set of dress rules which you will have to obey if you want to enter the club. These include no shorts, thongs or sandals without socks, and shirts must have a collar (this applies to men only I assume ED).



Please come to the AGM

In this issue of the TND is a letter from Les Tomlinson about increasing the advertising of TISHUG in the community. Now is the time that all members can tell the directors how they want their club to be run for the next year. We should also take the opportunity to thank the current directors for all they have done over the year to provide us with quality software at a good price, to provide encouragement and financial support to the makers of the exciting new hardware which appeared last year, to disseminate quickly the information that comes in from overseas and other sources, and for doing all these things at a bargain basement subscription rate.

If we are happy with the operation of the club we should become visible and vocal and say so. This can best be done at the AGM, so come along and join in. Of course, if you have some different views, the AGM may be the only place in which you can air them, and see if your views have any support with anyone else. If you want your club to be a democratic body, YOU must participate, and the AGM is the starting point.

