



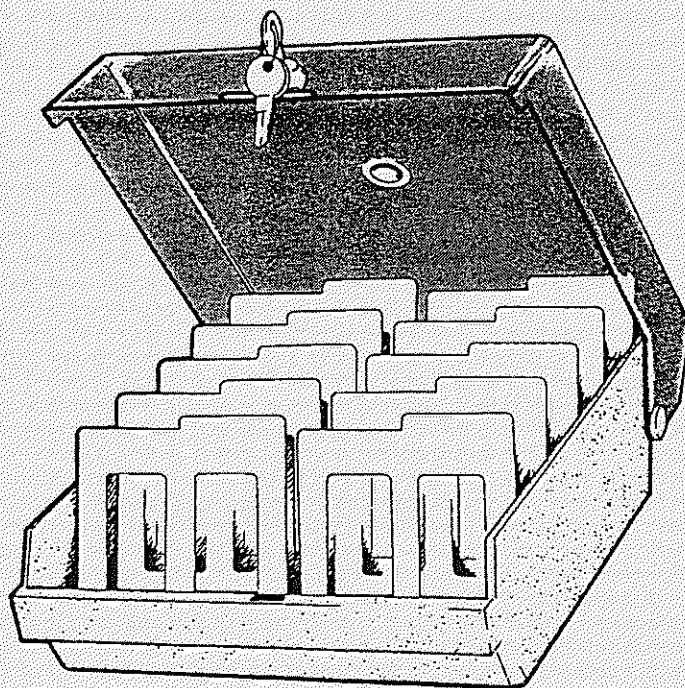
NEWS DIGEST

Focusing on the TI99/4A Home Computer

Volume 12, Number 5

June, 1993

Registered by Australia Post - Publication No. NBH5933



Sydney, New South Wales, Australia

\$3

TisHUG (Australia) Ltd.
A.C.N. 003 374 363

TisHUG News Digest

ISSN 0819-1984

TisHUG News Digest

INDEX

All correspondence to:
C/o 3 Storey St.
Ryde 2112 Australia

The Board

Co-ordinator

Dick Warburton (02) 818 8132

Secretary

Russell Welham (043) 92 4000

Treasurer

Cyril Bohlsen (02) 839 5847

Directors

Percy Harrison (02) 808 3181

Peter Schubert (02) 318 1181

Sub-committees

News Digest Editor

Bob Relyea (048) 57 1253

BBS Sysop

Ross Mudie (02) 458 2122

BBS telephone number (02) 456 4806

Merchandising

Percy Harrison (02) 808 3181

Publications Library

Russell Welham (043) 92 4000

Software Library

Larry Saunders (02) 844 7377

Technical Co-ordinator

Geoff Trott (042) 28 6629

Regional Group Contacts

Central Coast

Russell Welham (043) 92 4000

Coffs Harbour

Kevin Cox (088) 53 2849

Glebe

Mike Slatery (02) 692 8182

Hunter Valley

Geoff Phillips (049) 42 8176

Illawarra

Geoff Trott (042) 28 6629

Liverpool

Larry Saunders (02) 844 7377

Northern Suburbs

Dennis Norman (02) 452 3920

Sutherland

Peter Young (02) 528 8775

Membership and Subscriptions

Annual Family Dues \$35.00

Associate membership \$10.00

Overseas Airmail Dues A\$85.00

Overseas Surface Dues A\$50.00

TisHUG Sydney Meeting

The June Meeting will start at
2.00 pm on the 5th June 1993
at Ryde Infants School,
Tucker Street, Ryde.

Printed by
Kwik Kopy Parramatta

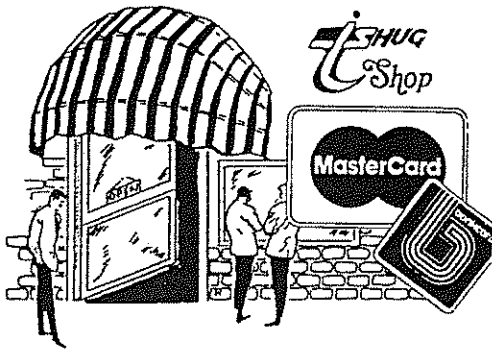
Title	Description	Author	Page No.
Bits and Bites	Tech / Gen Interst	Jim Swedlow	12
Editors Comments	General Interest	Bob Relyea	1
Getting the Most From Your TI	General Interest	Mickey Schmitt	12
Language Drill	General Interest	Don Shorock	13
Learn to Know Your TI No.5	Programming	Percy Harrison	4
Music Pro (v1.4)	Review	Jim Peterson	5
Nostalgia Time	General Interest	Geoff Trott	3
Ramble and Program	Programme	Stephen Shaw	15
Rambles and More Missing Links	Programme	Stephen Shaw	7
Regional Group Reports	Club News		19
Rieman2 the Missing Link	Programme	Stephen Shaw	7
Scudbuster	Review	Harrison Software	2
Smeggit	Review		18
Tips No.69	Programming	Jim Peterson	14
TI Bits No.24	General Interest	Jim Swedlow	6
TI Shop	Club News	Percy Harrison	2
TisHUG Software	Club News	Larry Saunders	4
Tournament Solitaire	Review	Jim Peterson	9
Treasur's Report	Club News	Cyril Bohlsen	13
Word Processing Part 5	Form Letters	Col Christensen	10

Editor's Comments

by Bob Relyea

I found the time at the All Day Tutorial to be beneficial as I attended a well-prepared and executed tutorial about TIPS by Alf Ruggeri. I am learning more and more about this software all the time and really enjoy it. Thanks to Alf and also to Larry for the use of his system. Ross gave an interesting presentation about Facsimilies and Peter Shubert had a well attended display on the latest in the IBM world. Geoff performed a valuable service by taking care of some technical problems. Where would we be without him? Dick was running around trying to keep things organised and running smoothly. The only negative observation was that there are fewer and fewer displays on these tutorial days and fewer and fewer Users attending most of them. It makes me wonder if we are hitting the mark! It was not even necessary for me to unpack my system. At least it gave me an opportunity to clean the dust up from in back of the monitor! Perhaps we need to have a re-think about what we are doing AND/OR each of us could make a bit better effort to support these tutorials before they disappear.

There has been a dearth of articles from the locals just lately except from the usual reliable contributors. We could use some more if you have the time. I like Geoff's articles on 'Nostalgia' and it is nice to see the effort that Percy and Larry consistently put into their articles and work. Keep it up guys. See you on Saturday.



TISHUG SHOP

with Percy Harrison

This month I am not going to list any of the goodies available from the shop as they have been fully covered in the last four issues so if you want to know what is available please refer to the Feb to May TND's.

Instead, I have decided to write about problems which I have encountered with the exchange consoles, etc. in the hope that it will make members aware of some of the things that make life a little more difficult for me when offering, what we believe, a first class service to our members at a fraction of the cost of having the same work carried out commercially.

Firstly, many of the consoles returned to me are in a varying state of cleanliness ranging from slightly soiled to absolutely filthy, being covered with dirt, grime and even a sticky film over the entire outside top surface of the console, including the keypad. The worst unit that I received, besides being in the grossly filthy state mentioned above, had several live brown bugs crawling around inside the console - the type that one does not normally welcome as a pet inside the home.

Those members who have received exchange consoles from me, if observant, would have noticed that the consoles are in a very clean state and certainly free from vermin of any kind. To achieve this degree of cleanliness often takes me many hours as I have to literally scrub the entire surface with a damp cloth and Ajax powder. Members could save me a lot of time if they cleaned the outside of the console before returning it to me for exchange, simply by rubbing the dirty areas with a slightly damp cloth and using a bit of Ajax powder on the stubborn areas. If you do clean the console make sure the cloth you use is not too damp or you will squeeze the moisture out of it when scrubbing the surface and it could run down into the internal works and possibly cause irreparable damage to your system. If you do not think that you can perform this task without getting moisture inside the console then leave well alone - do not do it as I do not want to damage or wreck my test equipment or system by connecting it to a waterlogged unit.

Another problem that is becoming more prevalent is that of missing screws, both on the console case and also inside the console - those that fasten the components to the case, etc. I have found that the self tapping screws used on the case are very hard to come by and would ask members to make sure that they do not misplace them if they open the unit for any reason. Also I have noticed on several consoles that were returned to me some of the screws have been replaced with screws having a different thread - this is a no-no as it results in damage to the plastic screw holes ultimately resulting in the loss of thread grip and holding power. It is best to leave the holes empty rather than fit screws with the wrong type or size of thread.

Members who do not have any knowledge of electronics should not attempt to repair their console or other electronic equipment should it decide to give up the ghost, firstly because you could finish up doing irreparable damage to the unit and secondly, most of the IC's in the unit cannot be purchased from electronic component suppliers as they are dedicated TI IC's. I believe that \$30 is a very small price to pay for a replacement console which carries a 12 months guarantee, in fact the \$30 would not cover the labour costs if you took it to a computer repair serviceman.

This raises another point, do not, under any circumstances, take your console to a commercial repair centre as they cannot obtain the TI chips for this console. The only chips available in Australia are through the TI Clubs who have taken over all the stock that was available from Texas Instruments.

In finishing I would ask those members who have modules, keyboards, etc that are not working and are not repairable to bring them back to the club as we can use the module cases for other projects and salvage spare parts from the other items. In this way we will be able to keep up the servicing of the TI units for a much longer period and hopefully maintain the membership numbers for several more years.

Bye for now.

***** END OF ARTICLE *****

Review: SCUDBUSTER Harrison Software

Enemy missiles fall from above on random courses and you must shoot them down with your missile from screen bottom left. You have one missile to hit each enemy missile, miss and they hit and you get negative points, shoot them down for points.

The aiming system is unique- you use your joystick to move a dot around the screen and your defence missile is launched at the dot. Unlike "Barrage" your missile does not explode at that fixed point, but keeps on going. Thus you do not have to target a specific future location of the enemy missile, merely aim for your defence missile's flight path to cross the flight path of the attack missile so the two missiles coincide.

Bruce Harrison is a first rate machine code programmer, and the quality of programming can be seen in the highest speed of the defence missile! You have an unlimited number of lives- missing a missile produces negative points! Missiles become mildly faster as the score increases until you hit the target score and end the game.

An interesting touch is that from pressing fire to the time your missile arrives at the marked point is virtually a constant time. This makes for interesting strategy- you can move the aiming dot to far screen right to make your defence missile speed really quickly even if the attack missile is at screen left- just so long as the missiles make contact with each other.

Conclusion: this appears at first sight to be a simple but playable shoot-em-down game.

Harrison Software can be contacted at:

5705 40th Place
Hyattsville, MD
USA, 20781.

Nostalgia Time

by Geoff Trott

This series of articles consists of my observations on the contents of the early TNDs starting with the earliest copies of our News Digest that I have in my possession. Assuming that you find that interesting, I am continuing with the series this month. Please stop me if you do not want me to continue. I will repeat my general disclaimer in case anyone reading this article gets the wrong idea. I am attempting to describe the look, layout and content of the newsletters without any critical intent. I will try to avoid using any adjectives which could cause offense and if anyone takes offense, that is purely their interpretation of the words and not my intention. I hope that makes my position clear and that no one will be offended.

As I said in the last article, the colour of the paper used in the November and December issues was a buff colour. The cover of the November issue has a montage of photographs of club members at meetings, while the back page had a membership renewal form, an advertisement for Chopper Attack and a description of the contents of the four new cassettes of club software. The November meeting in 1983 was the Annual General Meeting and all members were urged to attend. At this time, the Acting Co-ordinator was Brian Lewis who was not standing for re-election; Secretary was John Robinson; Treasurer and Acting Librarian was Terry Phillips; Educational Co-ordinator was Peter Lynden; Crisis Line was Graeme Hollis; Publications Editor was Shane Andersen; and General Committee Members were Peter Varga and Manual Constantinidis.

In the editorial, Shane noted that the December meeting will be a "pot luck dinner", where everyone brings along a plate of food and all share. I remember one of those, perhaps it was that one! Shane also started a column called "The Communicators" as a growing number of members were buying modems (10 at this time). TI Australia was looking at setting up a BBS in the Sydney office, perhaps by the middle of 1984. John Robinson's Minutes contain the winner of the monthly software competition, but John was obviously expecting Shane to insert his name before printing it so he is ".....". His program was "Maths Practice". The club had stopped producing monthly software tapes and was producing entire tapes devoted to particular areas of interest.

Andrew Nutting had a small article on adding printers, modems and disk drives to a system. He mentioned the Dick Smith external drives which could be picked up for \$200 to \$400. On the same page was a disk drive for sale for \$500, disk controller for \$250, 32K memory expansion for \$275 and RS232 for \$180 (all stand alone). David Liell's Tips was an introduction to writing programs in BASIC. He used a simple program as an example but unfortunately some characters have been lost in printing. It would be a great learning exercise even now, to get the final program working correctly.

Jenny's Younger set page had two photographs from the all day tutorial with the hall full of machines and people using them. There was a list for the Hall of Fame which showed some quite incredible scores for a number of games (incredible to me at least). There was a BASIC program called Magic Square Music by Stuart Anderson (1982). I do not know what it does apart from play some music. There was also an Extended BASIC program in the newsletter called Bugs in the Gallery by Don Gilchrist from Canberra (1983) which took up two full pages. It came with the recommendation that you would split your sides with laughter as you watch it run. On a more serious note was a program from Chris Ryan which gave a 40 column screen output in BASIC. It was an assembler program callable from BASIC, and a sample BASIC program was included. There was also a BASIC program to take a date in normal form of day, month and year and print out the day number in the year.

The December 1983 issue was the first with colour printing. The paper colour was still buff, but the printing was in green and on the front and back pages there was red also to provide a bright Christmas theme. The committee (with photographs) appeared on the back page. These were (after the AGM): Peter Varga (his name spelled incorrectly in this issue) co-ordinator; John Robinson secretary; Terry Phillips treasurer and librarian; Shane Anderson editor; Peter Lynden educational co-ordinator; Paul Mansel advertising; Graeme Hollis crisis line; Chris Ryan public relations; Russell Welham music; Andrew Nutting assistant librarian.

In the editorial, Shane wrote about the consequences of TI pulling out of the home computer market place. (This took place in October 1983.) He was optimistic about the future except for those who only want to play games. I think he was quite right, although even the games players were looked after quite well. In John Robinson's Minutes, he gave the name of the winner left out in the previous month's magazine (Brian Rutterford) along with a photograph of Peter Varga handing him his prize. The club decided to buy disks from Memorex for re-sale to members. A letter was printed from 99er Home Computer Magazine assuring that they will continue to support the TI99/4A. In October, 46 new members joined the club.

Peter Varga introduced himself as someone in the real estate business who wanted to bring the advantages of bulk buying to the club members. He wanted the club to become agents for overseas software suppliers and to look at buying hardware in large quantities. He was also keen to get special interest groups started. Jenny's Younger Set announced the winner of their software award as Michael Pascoe with "Air Defence". There was also a crossword puzzle and the news that the meeting in February would be a workshop day.

Peter Lynden had two pages of information on educational software. He reviewed the following programs: Mind Challengers; Market Simulation; Music Maker; Oldies but Goodies 1; Speak and Spell; Speak and Math; Addition and Subtraction 1; Computer Math Games 2 and 6; Scholastic Spelling; Milliken Math series. On the same page was a LOGO program to show a Christmas tree with flashing lights by an 11 year old from Texas. The Communicators program gave instructions for using a BBS run by MiCC. Graeme Hollis had a tip to help people work out how to interpret the input from joysticks into motion of a tank (for example). He gave a simple Extended BASIC program to do this. Techo Time from Steven Williams talked about connecting a parallel printer to a serial interface using a "black box" which he offered to other members. He talked about designing cheaper peripherals starting with memory expansion. There was also a photograph of his system.

As for programs, there was a BASIC program to play the tune "Deck the Halls". There were two programs from the Hamilton User Group in Canada (Channel 99). The first one was from Mike Towers and was called Commodore Killer and was in BASIC while the second one was a game by Malcolm Johnson about dodging carnivorous dogs and mines. Some hints were included in the write-up. It was written in Extended BASIC. There was another BASIC program to play the tune "Deck the Halls" in BASIC by Ross Jennens from Victoria.

Well that was the end of 1983. It was an exciting year for the club with an enormous increase in membership numbers but ending in a bit of uncertainty with the news from TI. In hindsight, the exit of TI from the market allowed a great freeing up of the market place for third parties, so that there were a lot of very exciting developments in the years after 1983. I am not sure how many would have predicted that we would still be going strong 10 years later.

***** END OF ARTICLE *****

Learn to Know Your TI

Lesson 5

with Percy Harrison

This lesson concerns the INPUT statement and the idea of a string variable.

INPUT will be taught in its pure form in this lesson: no messages in front. In a later lesson we will again return to the INPUT statement.

String variables are again introduced using the "box" concept. Variable names are restricted to one letter for the time being to avoid confusion in short programs and will allow faster typing.

LESSON 5 THE INPUT STATEMENT, STRING VARIABLES

INPUT STATEMENT

Use INPUT to make the computer ask for something.

```
Enter: 10 REM ::: TALKY-TALK :::
        15 CALL CLEAR
        20 PRINT "SAY SOMETHING"
        25 INPUT A$
        30 PRINT
        35 PRINT "DID YOU SAY"
        40 PRINT A$
```

Run this program. When you see a question mark, type "HI" and press the ENTER key.

The question mark was written by INPUT in line 25. The flashing cursor means the computer is waiting for you to type something in. When you type "HI", the computer stores this word in a box named A\$.

Later, in line 40, the program asks the computer to print whatever is in the box named A\$. Run the program again and this time type something funny.

STRING VARIABLES

A\$ is the name of a "string variable". The name is written on the front of a box and the string is put inside the box.

Rule: A string variable name always ends in a dollar sign, "\$". You can use any letter you like for the name and then put a dollar sign after it.

A\$ is called a variable because you can put different strings into the box at different times in the program. The box can hold only one string at a time.

ERROR MESSAGE FROM INPUT

Run the program again and answer the question with:

HI, THERE

The computer answered:

```
* WARNING:
  INPUT ERROR IN 25
  TRY AGAIN:
```

You should type your answer again, but not have any commas in it.

Rule: Do not put any commas into the string you type in answer to INPUT, unless...

YOU REALLY WANT TO USE COMMAS!

Run the program again and answer the question with:

"HI, THERE"

This time the computer works OK, no nasty messages.

Rule: Put quotation marks around the answer if it has commas in it.

Assignment 5:

1. Write a program which asks for a person's name and then says something silly to the person by name.

2. Write a program which asks you to INPUT your favourite colour and put it into a box called C\$. Now the program asks you to INPUT your favourite animal and also puts this into box C\$. Now tell the program to print C\$. What will be printed? Run the program and see if you are right.

Next month we will do some tricks with the PRINT statement.

ANSWER TO LESSON 4

Assignment Question 4A:2

```
10 REM SMILE
12 CALL CLEAR
20 PRINT
30 PRINT
40 PRINT
50 PRINT "      _ _  "
60 PRINT "      0  0  "
70 PRINT
71 PRINT
72 PRINT "      !!  "
73 PRINT
74 PRINT "      *  *  "
75 PRINT "      *  *  "
76 PRINT "      **** "
77 PRINT
80 END
```

***** END OF ARTICLE *****

TIshUG Software

By Larry Saunders

TIshUG Software for June 1993

PP024

This is the second part of 4 diskettes of Page Pro Templates (Cards) scanned and designed by Alf Ruggeri for the TI-Fair last year. All cards are archived separately. The Archiver is supplied on all 4 disks along with ROOT.

Card 13 to 18 Christmas Cards
Line-Demo Bonus program. Screen blanker.

Used= 355 Free= 3

ARC	33 Prog	CARD13	39*I128
CARD14	54*I128	CARD15	49*I128
CARD16	41*I128	CARD17	39*I128
CARD18	45*I128	LINES-DEMO	5*Prog
LOAD	5 Prog	READ*ME	17 d 80
ROOT	28 Prog		

PP025

Cards 19 and 20 Christmas Cards

- Card 21 Birthday
- Card 22 Anniversary
- Card 23 Bon Voyage
- Card 24 Thank You

Used= 337 Free= 21

ARC	33 Prog	CARD19	39*I128
CARD20	44*I128	CARD21	33*I128
CARD22	53*I128	CARD23	37*I128
CARD24	43*I128	LINES-DEMO	5*Prog
LOAD	5 Prog	READ*ME	17 d 80
ROOT	28 Prog		

G026

Games Disk.

Breakout. Hit the ball with the paddle and try to remove all the tiles above it. It is a very fast game after a hit or two.

Henhouse. Shoot the Ducks to pick-up extra points. Clear the shoots when full of eggs to the car. Kill the thief before he gets of the screen. Kill the fox before he gets to the henhouse.

Space Invaders. A classic game. Popeye. Collect the hearts around the screen. Watchout for Popeyes emiens. Shamus (Atari Game). A maze that has many levels to it.

Used= 343 Free= 15

BREA	33 Prog	BREB	29 Prog
HENHOUS1XB	35 Prog	HENHOUS2XB	18 Prog
INVA	33 Prog	INVADERS	9 D 80
INVE	15 Prog	LOAD	5 Prog
POP/LOAD	2 Prog	POPEYE1XB	43 Prog
POPEYE2XB	44 Prog	ROOT	28 Prog
SHAM/LOAD	2 Prog	SHAMUS	47 Prog

G027

Astro fighter. Is a space shootup style game.

Dragons. A game that you fly over land trying not to get hit or crashing.

Interceptor. A game that I have yet to master. Moon Patrol (ATARI game). Patrol the moon with your car.

Rock Hopper. A game that you will find quite good. Very large Extended Basic game that is now pre-scanned.

Used= 349 Free= 9

ASTRO1XB	31 Prog	ASTRO2XB	40 Prog
DRAG/LOAD	2 Prog	DRAGONS	44 Prog
INTERCP1XB	35 Prog	INTERCP2XB	43 Prog
LOAD	5 Prog	MOON1XB	35*Prog
MOON2XB	33*Prog	R/HOPPERXB	51 i254
ROCKLOAD	2 Prog	ROOT	28 Prog

***** END OF ARTICLE *****

Music Pro (version 1.4)

Reviewed by Jim Peterson

This program, written by David Caron, won first prize in a software contest held by the Ottawa TI-99/4A User Group in 1988. It is now being sold by Asgard Software. I am sure that Asgard is selling it by arrangement with Bruce and the Ottawa group - in fact, Asgard supplies it with a companion disk containing a utility program for use with Music Pro, also written by David Caron, and the documentation file for this utility refers users to Asgard to obtain Music Pro. However, the status of Music Pro itself is confusing because the program still bears a fairware notice requesting a donation to the Ottawa UG!

The program is written in Extended Basic with many links to assembly, which take up a great deal of memory, and VDP memory is completely remapped. The documentation contains warnings about doing things that could corrupt this memory, and the documentation for the utility program contains many more warnings. They seem to be needed - when I tried keying in a short piece of music and tried to save it, the program crashed, and I found that my filename had been replaced by a long string of garbage!

I must first say that this is a truly remarkable piece of programming. Lucie Dorais has contributed some help files which make it even better. The documentation is also very well written.

The Music Editor displays three blank staff lines on the screen, treble and bass. Notes are entered on these staves by pressing the appropriate key on the computer keyboard, from Q for low A to the period for high F - alternate rows on the keyboard provide sharps and flats. The disk contains a file which will print out a chart, but a person working with this program would probably soon memorize the keys. Provision is made for shifting to a higher octave but the bass octaves, which are accessible in Extended Basic programming through the -4 noise, are not available.

The length of the note is controlled by holding down the key while its pattern on the staff cycles from a 32nd note to a whole note - I found myself wishing that it would cycle back around when I held it too long. Many editing keys are available, as well as full control of volume. Notes can be "tied together" for a smoother sound; otherwise, the music has a rather "toot-toot" sound.

Only one voice can be programmed at a time. This is saved as a file, and a second and third voice can then be programmed and saved. The Compiler will read all three files and play them together in three-part harmony. While programming the music, the current voice can be played through the Editor, but the effect of combined voices can only be checked by saving the file and going to the Compiler.

This program was obviously written to enable a person with no knowledge of music to simply copy sheet music onto the screen. It does that extremely well. The person would still need some knowledge of key signatures, accidentals, and the notations used to indicate repetitions. I also think they would find it difficult to key in the three separate files without error, and to track down the errors.

However, this is certainly not the most efficient way of programming music, nor the most creative way, and it does not take advantage of musical effects, such as rapid attack and decay, which could only be achieved in assembly. To do so would have probably taken too much memory, and would have made the program too complex for its intended purpose.

I perhaps did not give the program a fair test but I did key in a melody in two voices. When I played them together, I found that I had made mistake somewhere, so that one voice was a quarter-note ahead of the other. The result was still musical, but quite unusual!

I decided that in the length of time I had spent on this, I could have programmed a couple of songs in all three voices in very elementary Basic, with the option of going to various subroutines for a wide variety of musical effects, and with many other features available.

***** END OF ARTICLE *****

TI Bits No.24

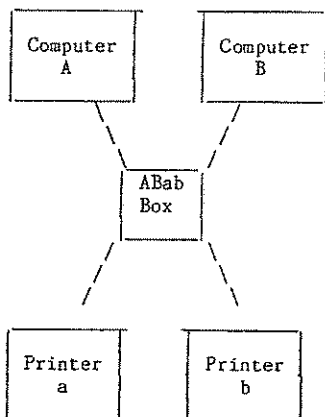
by Jim Swedlow

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

MULTIPLE PRINTERS AND COMPUTERS

Over time you tend to collect lots of stuff. I have two operating computers and four printers. I wanted to set up two printers -- one for regular paper and one for labels (I do lots of labels).

The first thing I found was a ABab switch box. It works something like this:



The box has four standard, 36 wire centronics plugs on the back and a switch with two positions on the front. In one position (AaBb) computer A is attached to printer a and computer B to printer b. In the other (AbBa) computer A is attached to printer b and computer B to printer a. I found a box for under \$25.

You have to buy two more cables with centronics connectors on both ends. Luckily, these can be found for under \$10 each if you shop around.

No problem, or so I thought. I hooked everything up and it did not work. The problem came because standards change. If you look in your RS232 book, it tells you to use pin 16 on the centronics side for the ground. All of my printers accept this as a ground. Unfortunately the cables are not 36 wire -- they just use the common wires and 16 is no longer used as a ground in the IBM world.

Out came my handy soldering iron. I switched a wire from a pin that the TI does not use to pin 16 on both ends. Now everything works like a dream.

It is a joy not to have to unload paper and load labels and then reverse the procedure. Net cost was about \$35.

TI IN ENGLAND

Received a letter from Stephen Shaw, who is the Disk Librarian and VP of the TI 99/4A User Group in the United Kingdom (TIUGUK). He had some interesting things to say:

"TIUGUK is now into its 8th year. Back in 83, JUST before the plug was pulled, TI insisted that the amateur TI HOME group should pass to a commercial organization. Membership was then 4,000 and rising and getting a little heavy to handle on a non commercial basis. Then TI pulled the plug and there was no user group. And we could not get the mailing list. So we managed perhaps 400 members by the end of 83. A small local group in Brighton became the default national group and started from there, presently declined to about 140 odd, which is just about sustainable."

"[Our] annual meet [was] held at Chester Northgate Arena on Saturday, May 26th, 1990. The annual meet again confirmed the strong support of the 4A in the UK by a small handful of users."

"With a membership of between 140 and 170, scattered over the UK and with several living abroad, the meet was visited by maybe 50 odd members, who came together to elect group officials for the year and to see the latest in software and hardware."

"One unexpanded owner went away the proud owner of a Mini Memory, a stand alone 32k RAM, a full Editor Assembler pack (for only five pounds) and the rare Miner 2049er sideways module. Your scribe picked up a Munchman II module to add to the collection. There was even the excellent SuperSketch peripheral on sale! And lots of spare bits and cables and so on."

"One display item was a photograph of young George Shaw (present with his favorite teddy bear Matthew) measuring two inches by two inches, and two printouts produced on a 4A, based on the photos, one measuring 7/8" by 5/8" and the other 8" by 8". These represent an interesting beta test of a possible new service for TI owners which may be offered by a famous US based owner (who at present wishes anonymity) - the photograph was scanned on a PC and cleaned up using Paintbrush (in particular the background was removed). Then the PC picture was translated to MacPaint format and transferred to the TI via RS232. From here it could be printed with MacFlix or PixPro and in fact was translated from MacPaint format to TI Artist format using PixPro."

"The TI Artist pictures was then printed as the 8" x 8" pictures using SmArtcopy by Alexander Hulpke. The smaller picture was made by using SQUEEZER to reduce the TI Artist picture to quarter size. SQUEEZER provides a choice of four densities and is the only really usable reducer for pictures. The small TI Artist pic was printed using Artist Photographic Vn 2, supplied with Harry Brashers Home Publishing on the 4A, Supplement #3. It is to be noted that while the TI programs could reproduce the scanned photo in the correct aspect ratios, the PC could not -- the two computers being used together was a result better than either could do on their own!"

"Our newsletter is 60 pages or longer. On average each issue of TI*MES occupies about 400 sectors, unarchived."

"I continue to spend my computing time writing, trying to establish (and maintain!) order in a very large disk collection and playing with fractal graphics (very time consuming). My chosen language now is THE MISSING LINK. Fractals are very slow of course and I do look forward to seeing the Graphics extension to Turbo Pasc 99 should they ever come out. However, unlike TML, they will not have TI Artist file computability built in."

"Still get lots of pleasure from the machine and have very many more years of exploration to do, at least until the day comes when it breaks down irreparably!"

Best wishes,

Stephen Shaw

Enjoy!

***** END OF ARTICLE *****

Rieman2 the Missing Link

from Stephen Shaw

```

100 ! RIEMANN INVERSE V3
110 ! R CASTLE-SMITH
120 ! 2 MAY 91
130 ! Original in Hewlett Packard Rocky Mountain Basic
    for plotter output
140 ! For TI99/4A + Extended Basic + The Missing Link
    by Stephen Shaw September 1991
150 RANDOMIZE
160 CALL LINK("CLEAR")
170 ! X,Y,Z LOCATION OF VIEWER
180 A1=(RND*2-RND*2)*RND :: A2=(RND*2-RND*2)*RND :: A3=
    RND*RND*2-RND*2-.2
190 A1=INT(A1*1000)/1000 :: A2=INT(A2*1000)/1000 :: A3=
    INT(A3*1000)/1000
200 CALL LINK("PRINT",1,1,STR$(A1)&" "&STR$(A2)&" "&STR$(
    A3))
210 IF A1=0 AND A2=0 AND A3=0 THEN 180
220 MULT=(INT(RND*3)+1)*2+1 ! size of squares. must be
    odd. 7=small sq 3=big sq
230 CALL LINK("PRINT",20,200,STR$(MULT))
240 ! PLOT REVERSE? 1=YES
250 PLOBAC=-(RND>.75)
260 CALL LINK("PRINT",180,1,STR$(PLOBAC))
270 !
280 IF A1=0 AND A2=0 THEN B1,C2=1 :: B2,B3,C1,C3=0 ::
    GOTO 320
290 M=SQR(A1*A1+A2*A2):: B1=-A2/M :: B2=A1/M :: B3=0
300 N=SQR(M*M+A3*A3):: C1=-A1*A3/M*N :: C2=-A2*A3/M*N ::
    C3=M/N
310 A1=A1/N :: A2=A2/N :: A3=A3/N
320 !
330 FOR YPIX=-60 TO 60
340 YV=YPIX/60
350 FOR XPIX=-60 TO 60
360 XV=XPIX/60
370 IF XV*XV+YV*YV<1 THEN GOSUB 400
380 NEXT XPIX :: NEXT YPIX
390 GOTO 390
400 T(1)=SQR(1-XV*XV-YV*YV):: T(2)=-T(1)
410 FOR I=1 TO PLOBAC+1
420 XR=A1*T(I)+B1*XV+CV*YV
430 YR=A2*T(I)+B2*XV+C2*YV
440 ZR=A3*T(I)+B3*XV+C3*YV
450 XC=2*(XR/(1-ZR))
460 YC=2*(YR/(1-ZR))
470 GOSUB 500
480 NEXT I
490 RETURN
500 ! IS THE PIXEL AT LOCATION XC,YC ON OR OFF?
510 ! plot of a chess board
515 IF XC+YC>6 THEN 540 ! reduces chaos pictured near
    south pole due to squares too close to show pattern.
520 PLOT=INT(MULT*XC+99)+INT(MULT*YC+99)! +99 offset
    makes all values +ve. thats all. MULT must be odd
    (3,5,7...)
530 IF PLOT/2=INT(PLOT/2)THEN CALL LINK("PIXEL",YPIX+90,
    XPIX+90)
540 RETURN

```

***** END OF ARTICLE *****

Rambles and More Missing Links

by Stephen Shaw

RIEMANN SPHERES....

One of the difficulties with taking an image and plotting it to a sphere is that the image can spread and distort and gaps left between pixels which then need filling. Another way of tackling this is to take the sphere, and from each point on its surface, calculate if the point is visible from the chosen viewing angle, and then calculate back to the plane (flat) surface for an equivalent x,y point and see if that pixel is on or off, and plot the sphere accordingly. This is a program that does that- although for The Missing Link (a commercial program from Inscebot/Texaments) any pixel addressable graphics program can be used.

Some interesting results can be obtained with this program! Rather than use a specific graphic I have here merely used a "chequer board" pattern test which in theory plots a pattern of black and white squares onto the plane. The results have been very varied. Instead you could perhaps utilise a look up table of some sort - or even use Myarc Extended Basic to look up a graphic on the screen. The sphere image is based on 0,0 at the North pole, extending to infinity, infinity at the South pole. An image from -2,-2 to 2,2 will cover the Northern hemisphere. Note that in the program below multipliers are used to control the size of the squares. The offset prevents having to deal with the way INT affects negative numbers, but does not affect the pattern. While the ability to see through the sphere to the "back" surface has been included, it will usually result in a very untidy plot.

```

100 ! RIEMANN INVERSE V3
110 ! R CASTLE-SMITH
120 ! 2 MAY 91
130 ! Original in Hewlett Packard Rocky Mountain Basic
    for plotter output
140 ! for TI99/4a + Extended Basic + the Missing Link
    by Stephen Shaw September 1991
150 RANDOMIZE
160 CALL LINK("CLEAR")
170 ! X,Y,Z LOCATION OF VIEWER
180 A1=(RND*2-RND*2)*RND :: A2=(RND*2-RND*2)*RND :: A3=
    RND*RND*2-RND*2-.2
190 A1=INT(A11000)/1000 :: A2=INT(A21000)/1000 :: A3=
    INT(A31000)/1000
200 CALL LINK("PRINT",1,1,STR$(A1)&" "&STR$(A2)&" "&STR$(
    A3))
210 IF A1=0 AND A2=0 AND A3=0 THEN 180
220 MULT=(INT(RND*3)+1)*2+1 ! size of squares. must be
    odd. 7=small sq 3=big sq 230 CALL LINK("PRINT",20,
    200,STR$(MULT))
240 ! PLOT REVERSE? 1=YES
250 PLOBAC=-(RND>.75)
260 CALL LINK("PRINT",180,1,STR$(PLOBAC))
270 !
280 IF A1=0 AND A2=0 THEN B1,C2=1 :: B2,B3,C1,C3=0 ::
    GOTO320
290 M=SQR(A1*A1+A2*A2):: B1=-A2/M :: B2=A1/M :: B3=0
300 N=SQR(M*M+A3*A3):: C1=-A1*A3/M*N :: C2=-A2*A3/M*N ::
    C3=M/N
310 A1=A1/N :: A2=A2/N :: A3=A3/N
320 !
330 FOR YPIX=-60 TO 60
340 YV=YPIX/60
350 FOR XPIX=-60 TO 60
360 XV=XPIX/60
370 IF XV*XV+YV*YV<1 THEN GOSUB 400
380 NEXT XPIX :: NEXT YPIX
390 GOTO 390
400 T(1)=SQR(1-XV*XV-YV*YV):: T(2)=-T(1)
410 FOR I=1 TO PLOBAC+1
420 XR=A1*T(I)+B1*XV+CV*YV
430 YR=A2*T(I)+B2*XV+C2*YV
440 ZR=A3*T(I)+B3*XV+C3*YV
450 XC=2*(XR/(1-ZR))
460 YC=2*(YR/(1-ZR))
470 GOSUB 500
480 NEXT I
490 RETURN
500 ! IS THE PIXEL AT LOCATION XC,YC ON OR OFF?
510 ! plot of a chess board
515 IF XC+YC>6 THEN 540 ! reduces chaos pictured near
    south pole due to squares too close to show pattern.
520 PLOT=INT(MULT*XC+99)+INT(MULT*YC+99)! +99 offset
    makes all values +ve. thats all. MULT must be odd
    (3,5,7...)
530 IF PLOT/2=INT(PLOT/2)THEN CALL LINK("PIXEL",YPIX+90,
    XPIX+90)
540 RETURN

```

=====

GAMES PLAYERS SPECIAL...

With thanks to Robert Brown of the Sydney (Australia) group...

Many games modules have a "cheat" mode, reported before but not for a while, and not with such an extensive list. In general, to enter cheat mode, when you have the title screen displayed, you must hold down SHIFT and press 8 then 3 then 8. There is often a time-out which means you have to do this fairly promptly from the time the title screen appears.

Available with:

ALPINER, MUNCHMAN, MUNCHMOBILE, MOONMINE.

After pressing the required keys you are able to enter the game at any level you require. TI released TI Invaders OFFICIALLY on disk, and the disk version only has the cheat mode, allowing you not only to select level but ALSO speed.

STAR TREK will give you unlimited lives after you press the required keys from the title screen.

CHISHOLM TRAIL, instead of starting from the title screen, wait until it asks you for LEVEL 1-9 then quickly make the required key presses.

HOPPER you only need type SHIFT 8.

TI RUNNER was only released on disk. When your man is flashing before the game starts, you can press FCNT 5 to skip to next level; or press SHIFT 3 followed by a two digit number to go to any screen (6 is 06).

CONTROL OF CSI BY PROGRAM:

The following small program will turn your cassette recorder or other device on and off under software control. The program has appeared widely, not always with the authors name attached (rectified here!) - below you will also find some hints on the practicalities of its use!

```
10 ! program by Ed Hall
11 ! to control the
12 ! "CSI" remote control
13 ! put in a music tape
14 ! plug in the remote
15 ! and press keys P and S
16 !
17 ! reverse polarity of
18 ! remote if it will not operate your recorder!
19 !
20 !
100 CALL INIT
110 CALL LOAD(16368,79,70,70,32,32,32,36,252)
120 CALL LOAD(16376,79,78,32,32,32,32,36,244)
130 CALL LOAD(8194,37,4,63,240)
140 CALL LOAD(9460,2,12,0,45,29,0,4,91,2,12,0,45,30,0,4,
91,203,78)
150 PRINT "PRESS:":" P Play":" S Stop"
160 CALL KEY(3,A,B)
170 IF B<1 THEN 160
180 ON POS("PS",CHR$(A),1)+1 GOTO 160,190,200
190 CALL LINK("ON"):: GOTO 160
200 CALL LINK("OFF"):: GOTO 160
210 END
```

This program requires Extended Basic and 32k Ram. It can operate the cassette player to provide audio-visual tuition, with cassette parts triggered at the appropriate stage by the program.

Note 1. The remote control is an electronic switch (for the technically minded, most consoles use a Darlington driver controlled by an optically isolated device). The polarity of its connection to the remote device MATTERS! If it does not work, try reversing the wires to the 2.5mm plug.

Note 2. As an electronic device, it uses up some of the voltage available to the remote device, a minimum of 1 volt. Some devices may be unhappy to have a whole volt removed - use mains supply for your recorders if possible and note that Ni-Cad rechargeable batteries only start with a lower open circuit voltage than other cells! Not too much lower but if the one volt drop counts...

You can use this program together with a clock such as that to be found in Triton Extended Basic, or Enhanced Display Package (library supply) etc etc to turn the cassette on and off at specific times (who needs a time switch! You can turn this device on and off several times a minute all year if you wish!). A simple for-next loop delay program can be used to time your on-off periods.

Of course you are not limited to turning just a cassette recorder on and off!!! You can turn anything on and off... provided you observe the correct polarity and do not try to switch too great a load!

In simple terms the ABSOLUTE maximum you can switch is 40V DC at 400mA, but you can exceed that by using a relay - ensuring that you use a diode to protect the computer circuitry (I gather a diode clamp is often used with relays...). For the technically minded, most consoles use a TIL119 isolator, and add to it a TIS92 NPN transistor, with the collector connected (normally!) to the tip of the plug. Thanks to Ross Mudie of Australia for this detail.

The following program will run with EITHER Mini Memory OR with Extended Basic and 32k Ram. This is by Art Green of Ottawa, and demonstrates the interrupt routine capability. It can also be used in a LOAD program on disk to convince your best mate that his TI really has got a computer virus. The first listing is in Basic for either module - it actually tests to see which module you are using!

```
100 REM interrupt demo
110 REM
120 REM MACHINE LANGUAGE
130 REM ROUTINE LOADED AT
140 REM >2600 XB OR E/A WITH 32K
150 REM >7200 MINI MEM NO 32K
160 REM
170 CALL INIT
180 XM=9728
190 MM=29184
200 LAD=XM
210 REM TEST XB OR MM?
220 CALL LOAD(XM,170)
230 CALL PEEK(XM,X)
240 IF X=170 THEN 270
250 REM NO 32K MUST BE MM
260 LAD=MM
270 A=LAD
280 REM LOAD M/C
290 CALL CLEAR
300 FOR D=540 TO 630 STEP 10
310 CHECK=0
320 FOR N=1 TO 10
330 READ X
340 CALL LOAD(A,X)
350 CHECK=CHECK+X
360 A=A+1
370 NEXT N
380 READ X
390 IF CHECK<>X THEN 490
400 NEXT D
410 REM POKE INTERRUPT
420 REM ROUTINE ADDRESS
430 REM INTO >83C4
440 CALL LOAD(-31804,LAD/256)
450 REM JUST IDLE AWAY TIME
460 FOR N=1 TO 9940
470 NEXT N
```

Tournament Solitaire

Reviewed by Jim Peterson

```

480 STOP
490 PRINT "ERROR IN DATA STATEMENT ";D
500 STOP
510 REM EACH DATA STATEMENT
520 REM HAS 10 DATA BYTES
530 REM PLUS A CHECK SUM
540 DATA 192,236,000,092,004,194,005,131,002,131,987
550 DATA 000,060,026,003,004,195,006,236,000,094,624
560 DATA 203,003,000,092,060,172,000,090,006,002,628
570 DATA 017,015,019,010,006,002,019,004,002,000,94
580 DATA 002,039,010,083,016,002,002,000,002,086,242
590 DATA 096,003,016,007,002,000,000,119,010,083,336
600 DATA 016,002,002,000,000,072,160,003,002,096,353
610 DATA 064,000,006,192,215,192,006,192,215,192,1274
620 DATA 016,000,216,044,000,094,140,000,004,091,605
630 DATA 000,015,000,000,138,128,000,000,000,000,281
640 END
    
```

For those of you interested in machine code, this is the SOURCE CODE for the machine code element contained in the above program...

```

* Interrupt demonstration
* GPL WS >83EO
* R11=RETURN ADDRESS
* R12=ADDRESS OF THIS ROUTINE
* R13 >9800 GROM READ DATA
* R14 SYSTEM FLAGS
* R15 >8CO2 VDP WRITE ADDRESS
* PROGRAM IS SELF RELOCATABLE
*
USRINT  MOVV  @T(R12),R3  GET TICK COUNTER
        CLR   R22        SET R2,R3 FOR DIVIDE
        INC   R3         COUNT TICKS
        CI    R3,60
        JL   TOCK       JUMP NO
        CLR   R3         WRAP FROM 60 TO 0
        SWPB @C(R12)    AND SWAP SCREEN CHARS
TOCK    MOV   R3,@T(R12) SAVE NEW TICK COUNTER
        DIV  @F(R12),R2  DIVIDE INTO 4 QUADRANTS
        DEC  R2         R2 HAS QUADRANT 0 TO 3
        JLT  QZERO      JUMP IF QUAD 0
        JEQ  QONE       JUMP IF QUAD 1
        DEC  R2
        JEQ  QTWO
QTHREE  LI    RO,551     LEFT SIDE GOING UP
        SLA  R3,5       Q3=551-32*REM
        JMP  QS
QTWO    LI    RO,598     BOTTOM GOING LEFT
        S    R3,RO      Q2=598-REM
        JMP  OUT
QONE    LI    RO,119     RIGHT SIDE GOING DOWN
        SLA  R3,5       Q1=119+32*REM
        JMP  QA
QZERO   LI    RO,72     TOP GOING LEFT
        A    R3,RO      Q0=72+REM
OUT     ORI   RO,>4000   VDP WRITE BIT
        SWPB RO
        MOVB RO,*R15    SET VDP ADDRESS FOR WRITE
        SWPB RO
        MOVB RO,*R15
        NOP
        MOVB @C(R12),@>8CO0 * CHAR TO VDP
        B    *R11       RETURN TO ROM ROUTINE
F       DATA 15        # OF CHARS PER QUADRANT
T       DATA 0         TICK COUNTER
C       DATA >8A80    "*" CHARS FOR DISPLAY
        END
    
```

And before we leave this...

You do not have to turn your console off or quit.
Enter in command mode:

CALL LOAD(-31804,0,0).

Tournament Solitaire is a collection of seven different card solitaire games on disk. You can select any of the games from the load menu, or elect to play all seven in sequence as a "tournament", hence the name. The games were programmed by William Reiss in Extended Basic with assembly links, and the disk is available from Asgard Software (P.O. Box 10306, Rockville MD 20849) for \$14.95 plus \$2.50 for shipping and handling (U.S. and Canada; \$7.50 for airmail elsewhere; 7% additional for credit card orders). The disk is accompanied by a very neatly published 7-page manual of instructions.

As a programmer, I can appreciate the skill and the effort that went into writing these seven programs. The graphics are all that can be done on the TI in Extended Basic, colourful and legible. The programming logic appears to be flawless - in none of the games was I able to make an illegal move, nor was any legal move refused. The manual is well written, although a bit sketchy - I still don't quite understand how to play the "Corners" game.

The seven games are Golf, Pyramid, Klondike, Canfield, Calculation, Pile Up and Corners. As far as I know, only two of these have previously been programmed for the TI - Klondike by Schererville and under the British name Patience by Gadget Man, and Pyramid by Regena. Of the others, Canfield was the only one I had ever heard of. To evaluate computerized card solitaire games, one must ask two questions - how do they compare with Walt Howe's Chainlink Solitaire, and are they easier and more enjoyable to play on the computer than with a deck of cards?

The first question is perhaps unfair, because I consider Chainlink Solitaire to be the best "brain game" ever programmed on the TI-99/4A.

As for the second, the shuffling and laying out of the cards is far quicker than could be done manually, thanks to the assembly link. Thereafter, action slows down. Moving cards from one stack to another is accomplished by using the arrow keys to move a cursor to the card to be moved, pressing the space bar to select it, using the arrow keys to move to the position it is to be moved to, and pressing the space bar again. Cards on the stack are turned over by pressing the Enter key, and some games also use other keys. The method of playing is the same for all the games, which makes it easier to play a tournament.

Many people would probably much rather use the joystick than the arrow keys. Personally I would very much prefer to simply select a numbered pile by pressing a number key, as Chainlink Solitaire is played. In spite of the cumbersome method of play, I did find these games to be very entertaining and addictive, and I spent a good deal of time playing them when I should have been doing something more productive.

I liked Pyramid, although it is one of the slowest in play, because it allows some opportunity for strategy. Its rules differ in one respect from Regena's version, which enabled me to actually beat the game once. I also managed to win at Pile Up, a complicated game with 20 piles of cards, which allows two reshuffles and a draw during the game, as well as peeking into stacks. Calculation is an unusual game which might permit considerable strategy, but would require a great deal of study. Klondike is the well-known solitaire game - it could have been improved by automatically turning exposed cards face up. Canfield is a variant of Klondike. Golf is the fastest playing, and very addictive.

Is it worth buying? Absolutely!

***** END OF ARTICLE *****

Word Processing Part 6

by Col Christensen
Brisbane User Group

FORM LETTERS

Form letters are those sent to a number of persons and containing the same information but generally personalised for each recipient. Certain parts of the letter like name, address, salutation change from letter to letter. Such a letter can be created in the normal way on the editor but where the wording has to differ, a variable must be placed. The variable takes the form of 'n' where n is a number from 1 to 99.

A value file of data to replace the variables must be created so that it can be called by the Text Formatter while printing. If the .FI command has been used, the data for each variable will be slotted smoothly into the text. The value file can take two forms. Probably the more common is the Mail List file on disk and the other is the list typed in through the keyboard during the course of printing in response to screen prompts. Which method you use depends on a number of factors.

Mainly, if you have need to send out form letters to the same people on more than one occasion, the mailing list is the more efficient and it is too, if you have many letters to send. With just a few variables in each of a small number of letters the list typed in while printing takes place seems the more attractive.

Here is a sample of a form letter complete with variables more suited to a disk based mailing list file. The =, where it appears represents the carriage return symbol that appears on the screen when the <ENTER> key is pressed.

```
*1* *2* *3*,=,  
*4*,=,  
*5*,=,  
=,
```

```
Dear *1* *3*,=,  
The Weeders' Digest's bonanza prize draw is to take place very soon and there could be much rejoicing at *4* at that time. All you need to do, *1* *3*, is to complete the enclosed form and return it together with the YES sticker and your luck might change.=,
```

MAIL LIST AS A VALUE FILE

A mailing list is compiled with a text editor keeping in mind certain protocols when typing the list. The first character on a line must be a number (1-99) which corresponds to the 'n' in the form letter. Next on the same line is a space, then the data for that variable and lastly a =, symbol by pressing <ENTER>. Other lines of data follow using the same format until all data for one form letter is entered. On the next line type an asterisk and press <ENTER>. This indicates that that is all the data for that letter. Continue on similarly with data for all letters. Here is a sample mailing list saved to disk as DSK1.MLIST1.

```
1 Mr=,  
2 J.=,  
3 Brown=,  
4 15 Redwood Street=,  
5 Forestvale Qld 4076=,  
*=,  
1 Mrs=,  
2 E.=,  
3 Smith=,  
4 28 Brolga Avenue=,  
5 Gumdale Qld 4090=,  
*=,
```

FORM LETTERS USING A MAILING LIST

When the text formatter is loaded, the list of screen prompts allows you to incorporate mailing lists in the form letters printed out. If using a mailing list, these prompts will appear:

ENTER INPUT FILENAME Answer the disk filename of the form letter.

ENTER PRINT DEVICENAME Answer FIO.LF or your usual RS232 name complete with the final .LF or you can send the output to a disk file without the .LF of course. If you happen to print out this disk file, however, you will need to add the .LF to the printer devicename then.

USE MAILING LIST? Answer "Y" this time.

WHICH LETTER(S)? (ALL) Answer "A" for all letters if using the full mailing list or specify by numbers which letters to print. A selection of letters could look like this: 1,5,9-17,19-20,23,25-E. Each individual number or group must be separated by commas and each group tied by a dash. The "E" can be used to signify the end of the mailing list.

PAUSE AT END OF PAGE? Answer as you normally do. "Y" if using single sheet manually fed paper, otherwise answer "N".

MAILING LIST NAME? There are two ways to inform the Text Formatter the location and filename of the data list. A mail list dot command such as .ML DSK1.MLIST1 towards the top of the text file would take care of that and there would be no need to give an answer to this prompt. Just press the <ENTER> key. If no dot command is included in the text file, it's obvious that you would have to input the filename here.

Printing then begins with a screen display of "PRINTING LETTER NUMBER n" tucked in amongst the other screen text. As printing continues the number of letters shown on the screen and yet to be done counts down.

FORM LETTERS USING ALTERNATE INPUT

If you answer "N" to the prompt, USE MAILING LIST?, and there are variables in the form letter the Text Formatter expects input of data from the keyboard. The screen inputs are slightly different too.

ENTER INPUT FILENAME Same as above.

ENTER PRINT DEVICENAME Same as before.

USE MAILING LIST? Answer "N".

WHAT PAGE(S)? (ALL) The input here is ignored anyway so just press <ENTER>.

NUMBER OF COPIES? This prompt takes the place of the previous one. Here enter the number of form letters you are going to do. During the course of printing this number on the screen will count down as each letter is printed.

PAUSE AT END OF PAGE? Same as above.

When the formatter encounters the first variable in the form letter file, another screen prompt appears saying, "ENTER DATA FOR VARIABLE *1*" and invites you to supply the data for it. Up to 28 characters (a whole screen line) are allowed for each data input. Once you press <ENTER>, printing will continue if there is sufficient text to send to the printer before the next variable crops up. So you must be on hand to input all the data values as they arise for every letter to be printed. You need to know the kind of value to type for each variable, like a name for *1*, a street for *2* and a city for *3* etc.

This process of alternate keyboard input through non-descriptive prompts like ENTER DATA FOR VARIABLE *6* leaves a little to be desired. A define prompt dot command (.DP) has been provided to allow you to tailor your own wording for each prompt. Your own prompt would guide you as to the type of information to type in. To achieve this a list of define prompts must be placed in the form letter prior to the first occurrence of a variable. The list might look like this:

```
.DP1 :NAME?=-
.DP2 :STREET?=-
.DP3 :CITY ETC?=-
.DP4 :Dear=-
```

Then as each variable is encountered, the prompt you defined for that variable number pops up to jog your memory on what kind of response to give.

You have seen how variables such as *4* are placed in form letters and treated in a particular way by the Text Formatter. The variable starts with an asterisk followed by a number from 1 to 99 and terminates with another asterisk. If this format is not adhered to, the formatter can do strange things to the text. That is why a printed asterisk must be transliterated from some other character non-sensitive to the formatter.

RANDOM NOTES

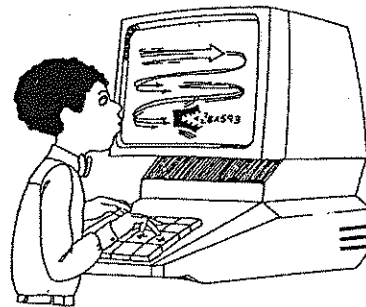
- * The text formatter defaults are NoFill, NoAdjust, LineSpacing=1 and PageLength=60.
- * HHeader, FFooter, DefinePrompt and TransLiterate must be on a line by themselves or be the last command on a line.
- * The ADjust command must be preceded either on the same line or separately by a Fill command to function at all.
- * .IN relative (+,-) is based on the preceding LM setting.
- * Other relative value commands must be preceded by an absolute value for that same command name
- * Your printer must have the automatic perforation skip disabled by setting one of the dip switches accordingly. You are likely to tear your hair out trying to format pages correctly if you do not. Both the formatter and the printer would be issuing form feed commands resulting in blank pages and pages with just a few lines printed on them.

- * Sending print through the RS232 card can be halted by pressing FCTN/4. The printer will stop when its print buffer is empty.
- * Printer problems such as jammed paper can be stopped on most printers by pressing the ON LINE button. Then press FCTN/4 on the computer to cancel its output.
- * When testing the formatter output, you can use a disk filename as the print devicename. You can then inspect the file to see the effect of the formatting.
- * The Text Formatter prints 5 lines less than the .PL command setting probably to allow for a header and a footer. Check this out using a disk filename instead of a print devicename.
- * You can outdent when creating lists of definitions or notes such as these. Set .LM inwards and .IN relatively less with a command such as .LM5;IN-4.
- * FindString will also search text column numbers. e.g. 18 27/REBATE/ will search columns 18 to 27 only. Handy for finding a match in a chart or table.
- * FindString will still search for a word even if no slashes surround the input.
- * ReplaceString can take quite a while in a lengthy text file with word wrap on and reformatting taking place. Don't be tempted to turn word wrap off, though, if the new string is longer than the one to be replaced. You could lose characters spilled off the right of the page.
- * ReplaceString must have word wrap turned off if searching through any columnar text such as lists and tables. Make a habit of doing a SaveFile prior to any RS anyway just in case there is some part of your text that is marked for NoFill.
- * Use TransLiterates to print asterisks and circumflexes.
- * Never print out a listed Basic program through the formatter. The multiply sign (*) and power sign (^) will disappear and the string concatenation (&) and the string variable (@) will produce confusing results.

I must compliment Tony McGovern on the sterling work he has done to manipulate the old TI-Writer program into such a useful and powerful word processor. One thing I have not mentioned is the ShowDirectory routine which Tony has expanded so that it is more like a file management program. Although it forms part of our WP, I feel that information on its use is best left to the maestro himself so look up his Funnelweb documents for all the gen.

Good Luck and Happy Formatting
See ya next month

***** END OF ARTICLE *****



arrow

Getting the Most From Your Cassette System

by Mickey Schmitt

I (Bob) received a nice letter from an American User, Bryant Krause of California who commented on the excellence of our magazine. He states, "I believe that every issue is a gem and that you seem to balance articles between people who know little about the TI as well as the ones who have worked with the TI for many years." He has sent me a copy of the publication-"Getting The Most From Your Cassette System" which is described in the following paragraphs.

What follows is a copy of the propaganda accompanying the publication mentioned above. If anybody is interested in inspecting the publication you can peruse it at one of the club meetings.

The popular series, "Getting The Most From Your Cassette System", written by Mickey Schmitt for the TI cassette-based system, is now available directly from the author in an all-new format.

What Is This New Format?

This all-new format is a 52-page, professionally typeset, loose-leaf booklet (without the holes), containing all of the original articles which first appeared in the West Penn 99'ers newsletters (though all have since been updated, corrected, and improved). In addition, new material that surfaced since the release of the original series has been added, making this booklet as complete as possible for the TI cassette-based user.

Why The Loose-Leaf Pages (without the holes)?

This particular format was chosen specifically so that user groups who purchase a copy of this booklet, directly from the author, could use their copy as a "master copy", making additional copies as needed for their own club's membership. Thus, a user group need only purchase one copy of the cassette booklet, the author will benefit by the sale to the user group, and the user group's members will all be able to benefit by their club's purchase.

What More Can Be Said About This New Cassette Booklet?

This cassette booklet provides an excellent opportunity for all user groups to provide a source of help to those club members who are still using a cassette-based system, as well as providing a source of help to those who are just joining a user group. The future existence of all TI user groups is dependent upon meeting the needs of the membership. This booklet is intended to fulfill one of those such needs.

To order your copy directly from the author, please send \$9.95 plus \$2.50 shipping and handling in the USA, or \$9.95 plus \$4.00 shipping and handling outside the USA (in US funds) to:

Mickey Schmitt
196 Broadway Avenue
Lower Burrell, Pa. 15068

N.B. This copying agreement is not offered to any commercial company, nor are user groups given permission to distribute copies of this booklet outside their own membership.

***** END OF ARTICLE *****

Bits and Bites

by Larry Saunders
from Reflections V2, No4

OPA Acquires TI Canada

OPA acquired a contract to service all 99/4A related products north of the U.S./Canadian border, as well as a huge stock of cartridges, parts and peripherals.

Gofer

If you use Page Pro 99 you are probably familiar with the plethora of utilities and tools available for this program. At last count Asgard Software alone offers over 50 compatible software packages, and they are coming out faster and faster. It is getting harder to separate what you need from what you want.

Gofer is a utility designed to appeal to the "power user" who uses the Page Pro 99 frequently. If you fit into this category, the time saved alone with this utilities in this package will pay for it many times over. Even if you are not, you may find Gofer to be more convenient and more flexible than the alternatives.

Simply put, Gofer is a collection of miscellaneous Page Pro 99 utilities by Dan Gazsy. Written in compiled c99, this package features complete re-writes of almost all the utilities included with Page Pro 99, plus many other additional little programs designed to let you get work done quicker and more efficiently. You will still need Pix Pro, and FX, but they can be added into this program along with TI-Artist Plus.

Do you columnize text with Page Pro Columnizer? The columnizer included with Gofer is around 50 times faster, gives you a much better idea of what kind of results you are going to get, and is loaded with new features. Need to convert art to and from TI-Artist? The conversion utilities allow you to convert files between TI-Artist instance and Page Pro 99 with ease in both directions, in batches. If wanting TI-Artist pictures you still need Pix Pro. Need to modify your Page Pro 99 page files (change picture filenames or drive number)? No problem with Gofer!

I am working on doing a full review of Gofer in the near future.

LinEditor

Do you ever need to look at REALLY large text files? Do you download large ASCII files from BBS's, write really large programs, or are you working on a Great Australian Novel? Do you ever run into TI-Writer's infamous "Out of Memory - Please Save"? If you answer to any of these questions is "yes", then you need a program that will let you load and edit a text file larger than what you can fit into the computer's memory. You need LinEditor.

Pronounced "Line Editor", LinEditor is the first text editor for the TI-99/4A or the Myarc Geneve 9640 that breaks the "what you can fit into memory" limit. This simple to use program makes viewing and changing big files a breeze. Simply load the file, tell the filename you want to give the changed version (you never lose your original - perfect for programmers!), and start typing, editing and deleting as you normally would.

LinEditor allows you to immediately modify what fits into memory, and automatically loads the next batch - never losing its place in your text file. Make as many changes as you like - LinEditor will handle them. It also allows you to open and close at any time input or output text files.

An all-Assembly program, linEditor is fast and flexible - it loads by RAM loader, TI-Writer, Editor/Assembler or even Extended Basic. linEditor even features a built-in help screen so you will never forget a command. Finally, linEditor uses many of the same keys as TI-Writer does for things, so working with it will not mean re-training your fingers.

It is compatible with the HFDC and RAM disks. (20mb text files anyone?).

Subject TI-Artist Plus

For colour instances of fonts, workout the area where you will be placing them, go to TI-Artist. Box the areas that you will be using, and fill each box with the colour that you the Font or picture to be. Exit to Font or Instances section. Press CTRL A, get your font or instance, move it to the area you want, press T (temporarily) to check, press ENTER if right.

You will find it will come up with the colour that you have placed on the screen in TI-Artist and cleared with CTRL A. This is a very easy way to colour your fonts or instances.

Bye for now Larry.

***** END OF ARTICLE *****

Language Drills

by DON SHOROCK
Reviewed by Jim Peterson

Don Shorock has written a shell program to give multiple-choice quizzes in word recognition, using disk files of words, and another program to be used to create these word files. By using the latter program, and by redefining letters peculiar to a specific language, he has created fairware disks of language drills in many languages. Each disk contains one or more word files, and he asks that users key in their own additional files for study, and share them with him.

My TI-PD catalog offers these disks for Modern Greek, Danish, Vietnamese, Czech (two disks), German, French, Polish, Hungarian, Russian, Japanese (three disks), Spanish, Finnish, Swedish, Norwegian, Latin, and Hebrew. Don also offers them in Arabic, Bulgarian, Croatian, Dutch, Esperanto, Gaelic, Icelandic, Indonesian, Italian, Rumanian, Serbian, Slovakian, Swahili, Turkish, Ukrainian and Yiddish, and is willing to consider requests for other languages, although he considers some Asiatic alphabets to be beyond the capabilities of the TI computer.

Most of these languages required only the modification of a few characters - the German umlaut, Spanish tilde, Scandinavian slashed O, etc. The Russian, Greek and Hebrew disks required a complete new alphabet. Vietnamese was a real challenge because their alphabet uses Roman letters with accent marks both above and below, which will not fit within the 8x8 dot matrix of a standard character definition.

Japanese, however, was the most difficult of all. In ancient times, the Chinese developed a most impractical system of using a picture to represent each word in their language, and these pictures evolved into a stylized form which do not even give a clue as to their meaning. One must therefore memorize several thousand of these characters in order to read fluently.

Since China was the Asiatic center of learning, the Chinese ideographs were also adopted in Annam (now Vietnam), Korea and Japan. The Vietnamese and Koreans have had the good sense to abandon them in favor of their own alphabet systems, and even the Chinese Communist regime streamlined them for easier writing and learning.

The Japanese, however, cling to the traditional form, although they are even more impractical for writing Japanese. Chinese verbs are not conjugated, therefore the same pictograph can represent all tenses. Japanese verbs, however, can have many different endings - and so can the adjectives and adverbs. The Japanese therefore had to develop a phonetic alphabet in order to tack these endings onto their words - and for some reason, they invented not one phonetic alphabet but two! Japanese newspapers are now written, usually in vertical columns, in Chinese kanji characters combined with the hiragana alphabet for the conjugations, with the katakana alphabet used for foreign words which have been adopted into Japanese, and often also with Roman letters used for foreign names and foreign words. The Japanese schoolchildren must therefore memorize thousands of kanji characters plus three alphabets in order to read their schoolbooks - and they are still the best-educated people in the world!

Don has written three Japanese drill disks. The one uses the ordinary Western alphabet, with the long vowels overlined. This is the Romaji or Romanji system which is used in phrase books for tourists, etc., and is adequate for learning the spoken language. Since the Japanese have already purchased Hawaii and are negotiating to buy Ohio, it might pay anyone to try out this disk - just in case!

The second disk provides drills in recognizing the characters of both the hiragana and katakana phonetic alphabets, and a considerable number of the less complex kanji ideographs.

The third disk provides multiple choice quizzes in the meaning of words written in either the hiragana or katakana alphabet. This disk also provides the option to run a program in console Basic, using the TE II module and Speech Synthesizer, and take a spoken rather than printed quiz! The speech of the Speech Synthesizer is never very clear, of course, and it was programmed for English rather than Japanese sounds, but the result is not all that bad. In fact, since Japanese is spelled phonetically with far fewer exceptions than English, most of the words are pronounced correctly although not too intelligibly.

***** END OF ARTICLE *****

TREASURER'S REPORT

by Cyril Bohlsen

Income for previous month \$ 1509.90
Expenditure for previous month .. \$ 1165.56

Profit for previous month \$ 344.34
Membership accounted for \$ 840.00 of Income.

On behalf of our Secretary,
Let me welcome our new members :-
Lou Newhouse of West Pennant Hills
Renae Smith of Happy Valley (SA)

Tips No.69

by Jim Peterson

Tigercub Software
156 Collingwood Ave.
Columbus, OH 43213

In Tips #68 I published my solution to Dr. Ecker's challenge to alternately assign X the value of A and B without using IF...THEN or any outside help. Computer Monthly has arrived again and his solution is better than mine. Try it with any two numbers-

```
100 A=2.765 :: B=-10
110 X=A+B-X :: PRINT X :: GOTO 110
```

There has been controversy for years as to whether the TI's pseudorandom number generator is truly random. Dr. Ecker's "Computer Fun & Learning" column in Computer Monthly had a question - if you randomly generate numbers between 0 and 9, how often will you get the same number twice in succession? Three times in succession? And etc. Since there are 10 numbers to choose from, it seems to me you would get 2 in a row 10% of the time, 3 in a row 1% of the time, 4 in a row .1%...etc. I wrote this to prove it-

```
100 RANDOMIZE
110 C=C+1 :: X=INT(RND*10):: PRINT X:: IF X=F THEN FL=F
L+1 :: CL(FL)=CL(FL)+1 :: PRINT "":FL;"=";CL(FL):"C=";C:
"%=";CL(FL)/C :: GOTO 110 ELSE FL=0 :: F=X :: GOTO 110
```

After 10,000 tries, I had 2 in a row 8.75% of the time and 3 in a row .83% and 4 in a row .07%. Does that prove anything? I do not know. (Dr. Ecker points out that those percentages could not ever quite add up to 100%!) Z Here is another of my Extended Basic programs to write assembly source code.

```
100 DISPLAY AT(2,1)ERASE ALL:"ASSEMBLY HELP SCREEN WRITE
R":":": This program will write the":":source code for an
assembly":":routine which can be linked"
110 DISPLAY AT(7,1):"from Extended Basic to dis-":":play
any one of several help":":screens at any designated":":ke
y press or input at any":":point in a program."
120 DISPLAY AT(12,1):" The original source code,":":autho
r unknown, was improved":":by Karl Romstedt and further":":
modified by Bruce Harrison.
"
```

```
130 DISPLAY AT(20,1):"How many help screens?" :: ACCEPT
AT(20,24)SIZE(1)VALIDATE(DIGIT)BEEP:N
140 FOR J=1 TO N :: H$=H$&"HELP"&STR$(J)&" " :: NEXT J :
: H$=" DEF "&SEG$(H$,1,LEN(H$)-1)
150 DATA VMBW EQU 2024,VMBR EQU 202C,KSCAN EQU
>201C,STATUS EQU >837C
160 OPEN #1:"DSK1.HELP/S",OUTPUT :: PRINT #1:H$ :: FOR J
=1 TO 4 :: READ M$ :: PRINT#1:M$ :: NEXT J
170 FOR J=1 TO N :: H$="HELP"&STR$(J):: PRINT #1:H$&" L
WPI WS": LI R13,HELPS"&STR$(J)
180 IF J<N THEN PRINT #1:" JMP AVSCR"
190 NEXT J :: H$=RPTS(" ",7)
200 PRINT #1:"SAVSCR CLR RO ":H$&"LI R1,SAVIT":H$&"LI
R2,768":H$&"BLWP @VMBR":H$ &"LI R9,NEWSCR":H$&"MOV R
9,R1":H$&"MOV R2,R4"
210 PRINT #1:H$&"LI R3,>6000":ADDOFF MOV B *R13+,*R9":
H$&"AB R3,*R9+":H$&"DEC R4":H$&"JNE ADDOFF":H$&"BLWP
@VMBW"
220 PRINT #1:"KEYLOO BLWP @KSCAN":H$&"BLWP @KSCAN":H$&"C
B @ANYKEY,@STATUS":H$&"JNEKEYLOO"
230 PRINT #1:"REPL LI R1 ,SAVIT":H$&"BLWP @VMBW":":RET
N LWPI >83E0":H$&"B @>6A"
240 PRINT #1:"WS BSS 32":SAVIT BSS 768":NEWSCR
BSS 768":ANYKEY BYTE >20":H$&"EVEN"
250 DISPLAY AT(3,1)ERASE ALL:" Enter data just as you":":
want it to appear, in 24":":lines. Press Enter for blank"
:":lines."
```

```
260 FOR J=1 TO N :: DISPLAY AT(12,1):"Ready for screen #
"&STR$(J):":":Press any key"
270 CALL KEY(O,K,S):: IF S=0 THEN 270 ELSE CALL CLEAR
280 ACCEPT AT(1,0):M$ :: PRINT #1:"HELPS"&STR$(J)&" TEXT
' "&M$&RPTS(" ",30-LEN(M$)) &" "'
290 FOR K=2 TO 24 :: ACCEPT AT(K,0):M$ :: PRINT #1:H$&"T
EXT ' "&M$&RPTS(" ",30-LEN(M$))&" "'
300 NEXT K :: NEXT J :: PRINT #1:H$&"END"
310 DISPLAY AT(3,1)ERASE ALL:" Source code has been writ
-":":ten to DSK1 as HELP/S. To":":assemble, insert Editor/
":":Assembler module."
320 DISPLAY AT(7,1):"Insert Assembler disk in drive 1
name DSK2.HELP/S"
330 DISPLAY AT(12,1):"Object file name? DSK2.HELP/O":":Li
st file name? Press Enter":": Options? R"
340 DISPLAY AT(15,1):"Load the resulting object":":file i
nto your program by":":CALL INIT ::":":CALL LOAD("":":DSK1.HE
LP/O") or,"
350 DISPLAY AT(19,1):"much better, imbed it with":":ALSAV
E or SYSTEX."
360 DISPLAY AT(21,1):"Access the screens in your progra
m by":": CALL LINK("":":HELP1")":":CALL LINK("":":HELP2")", etc
370 CALL KEY(O,K,S):: IF S=0 THEN 370 ELSE CALL CLEAR
```

For instance, at any point in a program where keyboard input is required and user may not know what to do - ACCEPT AT(24,1):M\$:: IF M\$= "HELP" THEN CALL LINK("HELP1 ") and the first help screen will pop up to give instructions. Press any key and the previous screen reappears.

This time I am borrowing heavily from the TI*MES news letter of England, which has also borrowed from the REC newsletter. This one is useless, but is a remarkable example of compact complex programming. It shows that there is an algorithm for everything. See if you can figure out how it works -

```
100 CALL CLEAR :: FOR A=1 TO 2 :: FOR B=1 TO 4 :: X=2-AB
S(SGN(B-3)):: FOR C=1 TO X :: PRINT CHR$(84-7*A+5*B-8*X)
:: NEXT C :: NEXT B :: PRINT CHR$(A+31):: NEXT A
```

Another useless one that is easier to figure out-

```
100 DISPLAY AT(1,1)ERASE ALL:"NUMBER OF MONTH(1-12)"
110 ACCEPT AT(2,12)SIZE(2)VALIDATE(DIGIT):A :: IF A<1 OR
A>12 THEN 110
120 DISPLAY AT(3,1):A;"x 4=";A*4 :: A=A*4
130 DISPLAY AT(4,1):A;" +13=";A+13 :: A=A+13
140 DISPLAY AT(5,1):A;"x 25=";A*25 :: A=A*25
150 DISPLAY AT(6,1):A;" -200=";A-200 :: A=A-200
160 DISPLAY AT(8,1):"Input date (1-31):" :: ACCEPT AT(8,
19)SIZE(2)VALIDATE(DIGIT):B :: IF B<1 OR B>31 THEN 160
170 DISPLAY AT(10,1):A;" +";B;"=";A+B :: A=A+B
180 DISPLAY AT(11,1):A;"x 2=";A*2 :: A=A*2
190 DISPLAY AT(12,1):A;" -40=";A-40 :: A=A-40
200 DISPLAY AT(13,1):A;"x 50=";A*50 :: A=A*50
210 DISPLAY AT(15,1):"Input last two digits of year e
g 91:"
220 ACCEPT AT(16,16)SIZE(2)VALIDATE(DIGIT):B
230 DISPLAY AT(18,1):A;" +";B;"=";A+B :: A=A+B
240 DISPLAY AT(19,1):A;" -10500=";A-10500 :: A=A-10500
250 DISPLAY AT(24,1):"ANY KEY FOR ANOTHER"
260 CALL KEY(5,A,B)
270 IF B<1 THEN 260
280 RUN
290 END
```

One for the little ones - change the string to anything you want.

```
1 REM SILLY PROG BY S SHAW MARCH 1991
2 ! did you see COMPUTER WARS-the film? It is said that
the star, who was required to type fast into a computer
3 ! could not type, so a program just like this one was
used to give a good effect!
4 ! now adjust it how you wish and show your friends how
fast you can type
5 ! at end of text string program will just stop with th
is listing but can be modified to do anything you wish!
```

```

b :
100 A$="This is how a non-typist can produce information
on screen quickly, without "
110 A$="Having to look at what keys are being bashed!
Just bash keys and watch how perfect text appears no matter
what you press."
120 CALL CLEAR :: PRINT A$ : : : :
130 CALL KEY(5,A,B):: IF B<1 THEN 130
140 C=C+1 :: PRINT SEG$(A$,C,1):: IF C=LEN(A$) THEN 160
150 GOTO 130
160 GOTO 160

```

And a very fast routine to find prime numbers-

```

100 ! FIRST 100 PRIMES -QUICKLY-
110 ! Dr H B Phillips from THE REC NEWSLETTER
March 1988 Vol 3 #2
120 DIM P(300),X(12)
130 A=0 :: B=1 :: D=0.5 :: E=180
140 M=100 :: L=3 :: F=0
150 ! increase M for more- also increase DIMS.
160 PRINT 2;:: C=B :: IF M=B THEN END
170 L=INT((M/C)*L+F):: N=L+L+B
180 FOR I=B TO INT((SQR(N)-B)*D):: PP=P(I)
190 IF PP=B THEN 230
200 IF PP=A THEN PP=I+I+B :: PRINT PP;:: P(I)=PP :: C=C+
B :: IF C=M THEN END
210 IF X(I)=A THEN X(I)=(PP*PP-B)*D
220 FOR J=X(I) TO L STEP PP :: P(J)=B :: NEXT J :: X(I)=J
230 NEXT I :: IF F=0 THEN S=I
240 FOR I=S TO L
250 IF P(I)=A THEN PP=I+I+B :: PRINT PP;:: P(I)=PP :: C=C+
B :: IF C=M THEN END
260 NEXT I :: F=(M-C)*L/E :: S=L+B
270 GOTO 170

```

And a demonstration of how the INTERRUPT routine works independently of whatever else the computer is doing-

```

100 REM interrupt demo
110 REM
120 REM MACHINE LANGUAGE
130 REM ROUTINE LOADED AT
140 REM >2600 XB OR E/A WITH 32K
150 REM >7200 MINI MEM NO 32K
160 REM
170 CALL INIT
180 XM=9728
190 MM=29184
200 LAD=XM
210 REM TEST XB OR MM?
220 CALL LOAD(XM,170)
230 CALL PEEK(XM,X)
240 IF X=170 THEN 270
250 REM NO 32K MUST BE MM
260 LAD=MM
270 A=LAD
280 REM LOAD M/C
290 CALL CLEAR
300 FOR D=540 TO 630 STEP 10
310 CHECK=0
320 FOR N=1 TO 10
330 READ X
340 CALL LOAD(A,X)
350 CHECK=CHECK+X
360 A=A+1
370 NEXT N
380 READ X
390 IF CHECK<>X THEN 490
400 NEXT D
410 REM POKE INTERRUPT
420 REM ROUTINE ADDRESS
430 REM INTO >83C4
440 CALL LOAD(-31804,LAD/256)
450 REM JUST IDLE AWAY TIME
460 FOR N=1 TO 9940
470 NEXT N
480 STOP
490 PRINT "ERROR IN DATA STATEMENT ";D
500 STOP

```

```

510 REM EACH DATA STATEMENT
520 REM HAS 10 DATA BYTES
530 REM PLUS A CHECK SUM
540 DATA 192,236,000,092,004,194,005,131,002,131,987
550 DATA 000,060,026,003,004,195,006,236,000,094,624
560 DATA 203,003,000,092,060,172,000,090,006,002,628
570 DATA 017,015,019,010,006,002,019,004,002,000,94
580 DATA 002,039,010,083,016,002,002,000,002,086,242
590 DATA 096,003,016,007,002,000,000,119,010,083,336
600 DATA 016,002,002,000,000,072,160,003,002,096,353
610 DATA 064,000,006,192,215,192,006,192,215,192,1274
620 DATA 016,000,216,044,000,094,140,000,004,091,605
630 DATA 000,015,000,000,138,128,000,000,000,000,281
640 END

```

Run that, then press FCTN 4. Enter LIST. Enter NEW. To stop it, enter BYE.

This is an oldie, but well worth repeating. You can use it to turn your cassette recorder on and off, to add speech or music from tape to a running program. With the proper hardware, you could write a program to control almost anything from the cassette port. If it does not work, reverse the polarity of the remote. Ed Hall wrote this-

```

100 CALL INIT
110 CALL LOAD(16368,79,70,70,32,32,32,36,252)
120 CALL LOAD(16376,79,78,32,32,32,36,244)
130 CALL LOAD(8194,37,4,63,240)
140 CALL LOAD(9460,2,12,0,45,29,0,4,91,2,12,0,45,30,0,4,
91,203,78)
150 PRINT "PRESS": " P Play": "S Stop"
160 CALL KEY(3,A,B)
170 IF B<1 THEN 160
180 ON POS("PS",CHR$(A),1)+1 GOTO 160,190,200)
190 CALL LINK("ON"):: GOTO 160
200 CALL LINK("OFF"):: GOTO 160

```

And that is just about -

MEMORY FULL!

Jim Peterson

Rambles and Programs

by Stephen Shaw

DISK DRIVE EXTENDED BASIC AUTO-LOAD

Back in Issue 32, page 24, I reported to you a "bug" in the system such that if you permitted your system to auto-load a program from disk when selecting Extended Basic, you lost the use of randomize. In Issue 32 I gave a short Extended Basic program to cure it. No response from anyone! I tested the bug out on my console and found that the bug was there and the fix cured it! However I am now using a different (cosmetically older) console, and find the bug is NOT there. Disk owners... can you do a little test and report the results please?

Type in this program:

```

100 RANDOMIZE
110 FOR T=1 TO 5
120 PRINT INT(RND*10);
130 NEXT T
140 RUN "DSK1.LOAD"

```

and save this little program onto a new disk in drive one as "LOAD".

Now reboot Extended Basic from the title screen so that your LOAD program is auto-loaded and watch the result. Do you keep getting a repeating pattern of five numbers or is each group of five numbers different? Please let me know, together with the serial numbers of your console (on the base) and your Extended Basic module. These typically are in the form ATA0583 - goods manufactured later in Italy often omit serial numbers, so advise "no serial/Italian" or whatever.

In issue 32 there is a simple Extended Basic cure, but Bruce, who is an assembly programmer, found a similar problem occurring with some ramdisk operating systems booting machine code, so he wrote the following two routines, both intended to be used with Extended Basic. These are available on disk from the disk library ready assembled on a utility disk.

MACHINE CODE PROGRAMMERS- CIF & CFI

I received an enquiry about using CFI in a machine code program to be used from Extended Basic, and was unable to find any source code using it. The second program below uses it and works!

```

* ASSEMBLY SUBROUTINE "SEED"
* FOR USE with EXTENDED BASIC PROGRAMS
* SEEDS RANDOM NUMBERS
* THIS SEEDS RANDOM NUMBER PROCESS AND REPORTS KEY
  PRESSED INTO VARIABLE IN EXTENDED BASIC
* BEHAVES LIKE A "CALL KEY" LOOP
* I.E. "SEED" WILL KEEP LOOPING ITSELF UNTIL A KEY IS
  STRUCK
* CALL LINK("SEED",K) SEEDS THE RANDOMIZE PROCESS
* USE AS CALL LINK("SEED",K) :: RANDOMIZE
* AND REPORTS THE KEY STRUCK BY THE USER INTO THE
  EXTENDED BASIC VARIABLE K
* AFTER THIS LINK, RANDOMIZE WILL WORK REGARDLESS OF HOW
  PROGRAM STARTED
* CODE BY BRUCE HARRISON
* RELEASED TO PUBLIC DOMAIN
* 18 AUG 1991

NUMASG EQU >2008      NUMERIC ASSIGNMENT VECTOR
XMLLNK EQU >2018      XML LINKAGE VECTOR
KSCAN EQU >201C      KEYBOARD SCAN VECTOR
KEYADR EQU >8374      KEY-UNIT LOCATION
KEYVAL EQU >8375      KEY VALUE BYTE
FAC EQU >834A        FLOATING POINT ACCUMULATOR
CIF EQU >20          CONVERT INTEGER TO FLOATING POINT
CFI EQU >12B8        CONVERT FLOATING POINT TO INTEGER
NUMREF EQU >200C     NUMERIC REFERENCE VECTOR
STATUS EQU >837C     GPL STATUS BYTE
DEF SEED             DEFINE ENTRY POINT

SEED
  LWPI WS           LOAD OUR WORKSPACE
  MOV @>8378,R10    TAKE THE VDP INTERRUPT TIMER INTO R10
  ANDI R10,>0001    MASK OFF ALL BUT THE LOWEST BIT
  CLR @KEYADR       CLEAR KEY-UNIT
KEYIN  MOVB @>83D7,@>83C1 TAKE THE SCREEN TIMEOUT'S LOW BYTE INTO SEED + 1
  CLR @STATUS       CLEAR GPL STATUS
  BLWP @KSCAN       SCAN KEYBOARD
  LIM1 2           ALLOW INTERRUPTS
  LIM1 0           DISALLOW INTERRUPTS
  CB @ANYKEY,@STATUS HAS A KEY BEEN STRUCK?
  JNE KEYIN        IF NOT, GO BACK
  XOR @>83C0,R10    NOW XOR SO LOW BIT OF R10 IS LOW BIT TAKEN ABOVE
  MOV R10,@>83C0    PUT R10 AT SEED
  MOVB @>8379,@>83C0 PUT BYTE FROM VDP INTERRUPT INTO HIGH BYTE OF SEED
  CLR RO           CLEAR RO FOR NUMBER ASSIGN
  LI R1,1         FIRST PARAMETER TO PASS
  MOV @KEYADR,@FAC PLACE KEY'S ASCII VALUE AT FAC
  BLWP @XMLLNK     USE XML LINKAGE
  DATA CIF        TO CONVERT INTEGER TO FLOATING POINT NUMBER
  BLWP @NUMASG     ASSIGN NUMBER TO PARAMETER
  LWPI >83E0       LOAD GPL WORKSPACE
  CLR @STATUS      CLEAR GPL STATUS BYTE
  B @>006A         RETURN TO GPL INTERPRETER
WS     BSS 32      OUR OWN WORKSPACE
ANYKEY BYTE >20   SPACE CHARACTER ASCII
END

```

The following program uses both CFI and CIF and works. The comments are very useful too. Note that these listings are for use with the Extended Basic module, and amendments are required for use with EdAs or MiniMem - notably you do not need some of the EQUates.

The source code below is "standalone" and does not require the above code, they are alternates!

```

* QUICK RANDOM
* MAKES RANDOM NUMBERS QUICKLY
* USE WITH EXTENDED BASIC
* TWO CALL LINKS ARE INCLUDED
* CALL LINK("SEED",K) :: CALL LINK("RKWIK",1,10,B)
* randomizes AND sets variable B from 1 to 10.
* CALL LINK("SEED",K) ACTS LIKE A CALL KEY LOOP, BUT
* SETS A RANDOM VALUE IN RANDOM NUMBER SEED
* AND REPORTS THE KEY VALUE INTO A VARIABLE (K)
* AFTER SEED HAS BEEN PERFORMED, CALL LINK ("RKWIK",LOW,
HIGH,VAR) WILL WORK
* GIVE LINK THREE PARAMETERS:
* FIRST THE LOWEST INTEGER IN DESIRED RANGE
* SECOND THE HIGHEST INTEGER IN DESIRED RANGE
* THIRD THE VARIABLE INTO WHICH NUMBER IS TO BE ASSIGNED
* LIMITS FOR LOW AND HIGH ARE (-32768 AND +32767)
* SO LONG AS "SEED" HAS BEEN USED, RANDOMIZE IS
UNNECESSARY
* TO GET RANDOM NUMBERS FROM RKWIK
* IF RND IS USED, RANDOMIZE MUST BE DONE AFTER "SEED"
AND BEFORE RND IS USED
* CODE BY BRUCE HARRISON
* RELEASED TO PUBLIC DOMAIN 30 AUGUST 1991

NUMASG EQU >2008      NUMERIC ASSIGNMENT VECTOR
XMLLNK EQU >2018      XML LINKAGE VECTOR
KEYADR EQU >8374      KEY-UNIT ADDRESS
KEYVAL EQU >8375      KEY VALUE ADDRESS
KSCAN EQU >201C      KEYBOARD SCANNING VECTOR
FAC EQU >834A         FLOATING POINT ACCUMULATOR
CIF EQU >20           CONVERT INTEGER TO FLOATING POINT
CFI EQU >12B8         CONVERT FLOATING POINT TO INTEGER
NUMREF EQU >200C      NUMERIC VARIABLE REFERENCE
STATUS EQU >837C      GPL STATUS BYTE
DEF SEED,RKWIK

SEED
LWPI WS              LOAD OUR WORKSPACE
MOV @>8378,R10      TAKE THE VDP INTERRUPT TIMER INTO R10
ANDI R10,>0001      MASK OFF ALL BUT THE LOWEST BIT
CLR @KEYADR          CLEAR KEY-UNIT
KEYIN MOVB @>83D7,@>83C1 TAKE THE SCREEN TIMEOUT'S LOW BYTE INTO SEED + 1
CLR @STATUS          CLEAR GPL STATUS
BLWP @KSCAN          SCAN KEYBOARD
LIMI 2              ALLOW INTERRUPTS
LIMI 0              DISALLOW INTERRUPTS
CB @ANYKEY,@STATUS  IF ANY KEY HAS A KEY BEEN STRUCK?
JNE KEYIN           IF NOT, GO BACK
XOR @>83C0,R10      NOW XOR SO LOW BIT OF R10 IS LOW BIT TAKEN ABOVE
MOV R10,@>83C0      PUT R10 AT SEED
MOVB @>8379,@>83C0  PUT BYTE FROM VDP INTERRUPT INTO HIGH BYTE OF SEED
CLR RO              CLEAR RO FOR NUMBER ASSIGN
LI R1,1             FIRST PARAMETER TO PASS
MOV @KEYADR,@FAC    PLACE KEY'S ASCII VALUE AT FAC
BLWP @XMLLNK        USE XML LINKAGE
DATA CIF            TO CONVERT INTEGER TO FLOATING POINT NUMBER
BLWP @NUMASG        ASSIGN NUMBER TO PARAMETER
LWPI >83E0          LOAD GPL WORKSPACE
CLR @STATUS          CLEAR GPL STATUS BYTE
B @>006A            RETURN TO GPL INTERPRETER

RKWIK
LWPI WS              LOAD OUR OWN WORKSPACE
CLR RO              CLEAR RO, NOT ARRAY VARIABLE
LI R1,1             SET FOR FIRST PARAMETER
BLWP @NUMREF        GET FIRST PARAMETER (LOW END OF DESIRED RANGE)
BLWP @XMLLNK        USE XML LINKAGE
DATA CFI            TO CONVERT VARIABLE TO INTEGER
MOV @FAC,R12        R12 HAS LOW NUMBER
INC R1              POINT TO SECOND PARAMETER
BLWP @NUMREF        GET SECOND PARAMETER (HIGH END OF DESIRED RANGE)
BLWP @XMLLNK        USE XML VECTOR
DATA CFI            TO CONVERT TO INTEGER NUMBER
MOV @FAC,R13        R13 HAS HIGH NUMBER
INC R13             INCREMENT TO INCLUDE BOTH ENDS
S R12,R13           SUBTRACT LOW LIMIT FROM HIGH LIMIT

```

```

LI R4,28645 PUT A BIG NUMBER IN R4
MPY @>83C0,R4 MULTIPLY BY THE RANDOM NUMBER SEED
AI R5,31417 ADD A BIG NUMBER TO RESULT IN R5
MOV R5,@>83C0 PLACE THAT BACK AT SEED LOCATION
CLR R4 CLEAR R4 SO NUMBER IS RIGHT JUSTIFIED IN R4-R5 PAIR
DIV R13,R4 DIVIDE BY THE RANGE +1
A R12,R5 ADD THE LOWER LIMIT TO REMAINDER FROM INTEGER DIVISION
MOV R5,@FAC MOVE THAT NUMBER TO FAC
BLWP @XMLLNK USE XML LINKAGE
DATA CIF TO CONVERT TO FLOATING POINT FORMAT
INC R1 POINT AT THIRD PARAMETER (VARIABLE FOR RANDOM NUMBER)
BLWP @NUMASG ASSIGN THE VALUE TO THE VARIABLE
LWPI >83E0 LOAD UP GPL WORKSPACE
CLR @STATUS CLEAR STATUS BYTE
B @>006A RETURN TO GPL INTERPRETER
WS BSS 32 OUR OWN WORKSPACE
ANYKEY BYTE >20 THE SPACE CHARACTER VALUE
END

```

***** END OF ARTICLE *****

Sneggit

SNEGGIT is a game requiring fast reactions, quick planning, and a small dose of caution. Those evil forces of chaos have struck the henhouse, scattering all the eggs around where some hungry snakes are ready to make a quick meal if you do not stop them. But watch it! These snakes are not about to just slink around while you rescue all the eggs - they think you taste pretty good, too.

You control the chicken who was left to guard the henhouse. By moving next to an egg and pressing the fire button, you can pick up the egg onto your back, where you can carry it to a nest and drop it off safely. Defend the nest long enough, and your egg hatches into a baby chick, who runs off the screen and leaves you another egg to save. The snakes will leave eggs in the nest as long as you are there to guard the nest, but slip away to get another egg and your nest egg is fair game. If you can put 16 eggs into nests, the chicken in the sky will give you a helper to take over in case of snakebite. There are a lot of weeds and rocks scattered around the barnyard which just get in your way and make it hard to get around. The snakes are smaller than you, so they can sometimes go places you cannot. Snakes can also go down snake holes when they feel like it, and they come up where you least expect them. Watch your nests! You can face down a snake, but turn your back on one and you are his next meal.

The chicken is controlled by either a joystick or the standard keyboard keys for directional control (S/J, D/K, E/I, and X/M for left, right, up, down, and Q/Y for FIRE). Since SNEGGIT can be played by two players, only the active player's keys are enabled. During play, the REDO key will take you back to the secondary title screen where you can restart another game or go back to the main title screen. Pressing the BACK key during play takes you back to the main title screen.

A round ends when all eggs have been eaten or after the snake has you for lunch. Three rounds make a game. Your highest score is kept as long as SNEGGIT is running. The scores of the last game played are shown for comparison.

SNEGGIT may be played by one or two players. There are three levels of play; Novice, Advanced, and Expert. The Novice level is set up for easy learning and a slower playing speed. After you master the Novice level, you can move up to the Advanced level. Advanced gives twice as many points per egg, but is quite a bit faster to play. When even the Advanced level is too easy, you can move on to the Expert level. This scores four times the points as the Novice level, but there are two snakes, and they move much faster.

Each colour egg gives a different number of points when it is picked up; see the help screen for a picture which shows the scoring. The help screen also shows you where you can pick up eggs and where you must stand to drop them into the nest. You may call up the help screen from the main title screen by pressing the 'AID' key. If you are not close enough to the nest, or if you hold the FIRE button too long, you will drop the egg and smash it. The FIRE button needs a lighter touch for each advanced level of play, so you need a very quick touch at the Expert level.

Eggs are scored once when you pick them up, again when you successfully put them into the nest, and one you must protect the eggs until they hatch.

A nest can only hold eight eggs at one time; if you try to put more than eight into a nest, the new eggs will fall out onto the ground. Sometimes eggs can be very hard for you to see, but the snake still knows where they are. This is the chicken in the sky's way of telling you there are too many eggs in a horizontal line. It is not a problem but you can prevent it from happening by not putting all your eggs in one basket.

I cannot tell you any more. If you want to learn more, you will have to play the game! Happy egging!

Supplied by GAMES for TEXPAC BBS.

***** END OF ARTICLE *****

Regional Group Reports

Meeting Summary For JUNE

Banana Coast	13/06/93	Sawtell
Central Coast	12/06/93	Saratoga
Glebe	10/06/93	Glebe
Hunter Valley	12/06/93	
Illawarra	15/06/93	Keiraville
Liverpool	11/06/93	Yagoona West
Northern Suburbs	24/06/93	
Sutherland	18/06/93	Jannali

BANANA COAST Regional Group (Coffs Harbour Environs)

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

CENTRAL COAST Regional Group

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

GLEBE Regional Group

Regular meetings are normally on the Thursday evening following the first Saturday of the month, at 8pm at 43 Boyce Street, Glebe. Contact Mike Slattery, (02) 692 8162.

HUNTER VALLEY Regional Group

The meetings are usually held on the second Saturday of each month at members homes starting at 3:15 pm. Check the location with Geoff Phillips on (049) 428 176. Note that after 9:00 pm this number is used for the ZZAP BBS which includes TI-99 information. Geoff.

ILLAWARRA Regional Group

Regular meetings are normally held on the second Tuesday of each month after the TISHUG Sydney meeting (except January) at 7.30pm, at the home of Geoff & Heather Trott, 20 Robsons Road, Keiraville. A variety of activities accompany our meetings, including Word Processing, Spreadsheets and hardware repairs. Last month we had another look at the TIM Card and some colour graphics for the 80 column users. Contact Geoff Trott on (042) 29 6629 for more information.

* LIVERPOOL Regional Group *

Regular meeting date is the Friday following the TISHUG Sydney meeting at 7.30 pm. Contact Larry Saunders (02) 644-7377 (home) after 9.30 pm or at work (02) 708-1987, Liquorland, Yagoona for more information.

*** ALL WELCOME ***

11th June, 1993 *****
My Place * Picasso and *
34 Colechin St *Utilities for it*
Yagoona West 2199 *****

Bye for now Larry
Liverpool Regional Co-Ordinator

NORTHERN SUBURBS Regional Group

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

SUTHERLAND Regional Group

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

TISHUG in Sydney

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

JUNE MEETING - 5th JUNE

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

July	13th June
August	11th July

These dates are all Sundays and there is no guarantee that they will make the magazine unless they are uploaded by 6:00pm, at the latest. Longer articles should be to hand well before the above dates to ensure there is time to edit them.

