BUG BYTES

NEWSLETTER

of

TIBUG

TI - 99/4A - BRISBANE USER GROUP INC
P.O. BOX 3051
CLONTARF MDC, QLD AUST 4019

MARCH 1992

COMING MEETINGS
27 MARCH and 24 APRIL

7.30 p.m.
EAST BRISBANE STATE SCHOOL
CNR WELLINGTON RD. AND
STANLEY STREETS,
EAST BRISBANE.

COMMITTEE

PRESIDENT
Ralph Nielsen - 378 1237

VICE-PRESIDENT
Col Christensen - 284 7783

SECRETARY
Garry Christensen - 888 4857

TREASURER
Val Jones - 396 1662

DISK LIBRARIAN
John Reynolds - 357 9758

TAPE LIBRARIAN
John Peacock - 074 673 376

BOOK LIBRARIAN
Trevor Campbell - 351 3107

MODULE LIBRARIAN
John Peacock - 074 673 376

EDITOR
Garry Christensen - 888 4857

All items within this newsletter may be reprinted providing the source and author are acknowledged.

The views expressed in articles published in TIBUG are those of the author and do not necessarily reflect the views of the Editor, Committee Members or Members of this User Group.

All items, articles, programs etc in this Newsletter are believed to be public domain.

Contributions to TIBUG are invited from both members and non-members. Articles for inclusion in the succeeding monthly newsletter are required at least two weeks before the monthly meeting and may be included in that newsletter at the discretion of the Editor. If you have a disk system, please supply script on disk with diagrams separately on paper and as clear and black as possible to facilitate photocopying.

Most original articles by members of TIBUG in this newsletter are on disk and are available to other User Groups on request.

Submissions of articles, reviews, comments and letters from mebers is encouraged, however the editor would ask that members keep the following in mind.

Submissions should be about computers, the TI community in particular, or have general interest value.

The preferred media is floppy disk (any format) however cassette tape is most acceptable for those members who do not have expanded systems. Please remember that handwritten submissions have to be retyped into the computer so that they can be reproduced. Typed submissions can also be used directly if the quality of the type is suitable for photocopying.

The newsletter is produced on the weekend preceding the monthly meeting. Any submissions made after the Friday, one week before the meeting will be held over until the following month.

Submissions are best sent directly to the Editor rather than through the PO Box. The address is Garry Christensen, 18 Zammitt St, Deception Bay QLD 4508.

Contact the editor if you have any difficulties with preparing a submission or have any comments about the newsletter.

öööööööööööööööööööööööööööööööööö

CONTENTS

EDITORIAL

For the owners of a Geneve: What do you think about MDOS? Personlly I think that it is about as useful as an ashtray on a motorbike. What an enormous waste of time. A lot of the functions duplicate operations that can be done a lot easier by a disk management program so its only use seems to be to load programs.

Let's face it. How often do you use programs written for MDOS? For myself, I spend most of my time using TI programs. I didn't buy this computer to run new exclusive software (there wasn't any at the time), and I don't expect it to run PC programs (if I had wanted that I would have bought a PC). I use programs for the TI because that is the system that I am familiar with and it does all that I want. I have to put up with PCs at work, but I still look forward to coming home to my TI99/9640. I want a better TI99, not something that tries to pretend it's a PC.

MDOS is strangling the 9640. Software developers are battling with compatability, and trying to force an unstable environment to support their talents. The 9640 already has a stable environment - GPL. Here is a structure that we are all familliar with. It has a flexable memory manager set up by the hardware (let's face it, Myarc really do know about good hardware) and simple access to the video chip. even gpl access if it is needed. I believe that programmers should be using that talent to produce some truely spectacular TI programs using the features afforded by the 9640.

Here is my opinion of the way the 9640 should boot. The EPROM should contain the files loaded by the GPL program, that is the dumps of the TI ROMs and GROMs. When switched on, the computer should initialise and check memory, switch to TI mode, install the ROMs and GROMs into ram, then search ramdisk, hard disk and floppy disk (in that order) for a version of the BOOT program. This version of BOOT would be very similar to the one for ramdisks but it would have the added feature of loading modules into GRAM.

BOOT would also initialise the system for internal ramdisk and print spooler. This phase involves setting up a master DSR, providing SPOOL and IRD (Internal RamDisk) device names only and the master DSR must allow external DSR calls. The TI was designed for all DSRs to be on the cards, why gag the system by limitting them to what is in a master DSR. The ramdisk and print spooler could be configured along with the BOOT menus. There could even be an option to load additional DSR's. Perhaps a package like Text To Speech could be set up so that it would function like and external device. I'm sure that there would be many other applications.

There it is! An easily configured menu system that will load modules as well as files and load a disk manager if you need those functions. It would occupy a fraction of the memory used by MDOS and be far more user friendly. If, on the odd chance that you want to use a MDOS program, SYSTEM/SYS is needed, a small loader would handle that with ease.

Spend a moment thinking about it. It could even go GUI (graphics user interface), using the features of the advanced video chip and the mouse. There is an opportunity here to develope a front end that is really good.

Let's forget about which version of MDOS will work and which won't. I say scrap the lot and start again with something that is not years out of date before it is written.

0000000000000000000000000000000000000

IN THE P.O. BOX

News Digest (Sydney), March: Editor's Comment, Co-Ordinator's Report, Treasurer's Report, Secretarie's Notebook, Notes on DSCAN, the DSR Scanner by Ross Mudie, Shop, Software Column, Techo Time with Geoff Trott (formatting problems), Word Count, GIF Mania reviewed by Alf Raggeri, GIF File Transfer - PC to TI by Rolf Schreiber, Amateur Radio Log and The Communicators by Ross Mudie, Vaporware Slowware and No Ware at All by Jim Peterson, TI-Bits and XB Tips by Jim Swedlow, TI99/4A world News by Jim Peterson, Assembly Class, For Sale, Page Pro 99 by Ed Johnson, Beginning Forth by Earl Raguse, Decoding EPROM Files by Ben Takach.

ADVENTURE COMPENDIUM

(EDITOR'S NOTE: The following has been extracted from the READFIRST file on the TI Adventure Compendium that this group has just received. All the files will be added to the library and will be available to members. There are 10 disks in the set, all full of archived adventure programs. It seems unfair that the disk librarian should have to do all the work of unpacking them and cataloging them so I sure that a volunteer would be appreciated.)

The TI ADVENTURE COMPENDIUM is a joint copyrighted fundraising package of the Western New York 99ers (WNY99ER) [contact Harry Brashear, 2753 Main Street, Newfane, NY 14108] and the Massachusetts Users of the Ninety-nine Computer and Hobbyists (M.U.N.C.H.) [contact Jack Sughrue, Box 459, East Douglas, MA 01516]. This mutual effort is an attempt to provide users with a historical and, in many cases, remarkable collection of fascinating and fun adventure games and utilities (and some bonus games and utilities) that would not be readily available anywhere else.

The disks are a collection of the *COMPLETE* Fairware and Public Domain adventure programs written for the TI-99/4A as of 1989. Over two years and the diligent efforts of many users have gone into the gathering of these items, and every attempt was made to secure the original versions.

The authors of these adventures did wonderfully well and take us from the simple simulation type of all-text adventure of the early 80s to some recent heavy-duty graphic fare. Although some authors have remained anonymous, their work is as greatly appreciated as some of the more famous authors whose works are part of this collection. Without these ingenious and very creative authors, of course, there would be no adventures, none of the fun that only such puzzles and games can reward you with.

Mickey Schmitt (herself the author of the Adventure Module adventure "Oliver's Twist") is the person most responsible for this compilation. Two years ago she spoke to me of her vision to gather up every known adventure for the TI and to write a book which would be a source and reference guide to all users. Her book, THE ADVENTURE REFERENCE GUIDE (The Encyclopedia of TI-99/4A and Geneve Adventure Gaming), is - At Last! - completed. And, in the process, the gathering of these adventures has been made possible.

Mickey's book ($7.95 & $2 S/H from ASGARD Publishing, P.O. Box 10697, Rockville, MD 20850), has already become the adventure afficianado's bible. This classic work is not only a multi-sorted bibliogaphy of EVERY known Commercial, Fairware, and Public Domain adventure, but breaks the adventures into mode (TOD, BASIC, etc.) and type. Each adventure is rated against other adventures of kind. The names of all known authors and addresses are given as are the required hardware needs. Checklists for players are provided along with detailed descriptions. And much more. This book - one of a kind in the entire computer world - is an almost-essential companion to these disks and highly recommended by both user group compilers. About 200 adventures are analyzed. The Fairware and Public Domain items amount to a little less than half of these. All of THESE games (and some utilities and a few other items and a game) are contained on this TI ADVENTURE COMPENDIUM disk collection.

All of these programs were gathered from around the world on hundreds of disks. Mickey sorted through them all, checked for repeats with different names, hunted down the originals, hunted down the authors, tested them all out, and reduced them to about 30 disks which she released to our two groups for distribution and fundraising. These disks, in turn, were packed tightly onto disks and ARChived and, again, repacked tightly onto the present disk form on DSDD. They were then repacked onto DSSD and SSSD forms (which took up more space). To attempt to release this huge package in any other form would be too expensive and costly, and it would not get into the hands of the average user.

You will be able to leap into adventuring with both feet. There is enough here to satisfy the most jaded gamester and programmer, perhaps, for years! There are no age limitations to interactive fiction; no barriers of any kind to imagination. Enjoy thoroughly all these journeys into

your own mind and the challenges that await you there.

Thank you very much for your support of our groups' efforts.

(NOTE: TI ADVENTURE COMPENDIUM is a copyrighted collection by JAHA and may not be distributed in any form except by user groups to their own members and not part of an outside revenue or "distribution cost" fundraiser for M.U.N.C.H. & WNY99ER groups as mentioned above. If you come into possession of this package other than ordering through our clubs, would you be kind enough to send just $6 donations to EACH of the groups above. If you received this through your user group would you be kind enough to take up a user-group collection and send half of that amount to each of our clubs. Please maintain the integrity of our work by keeping these disks in these original forms as you received them. Again, thank you.)

** Copyright 1989 by JAHA **

ADVENTURE PROGRAMS AND UTILITIES.

Adventure or Tunnels of Doom Module Requried:
An Adult Adventure
Great Avocado Adventure
Magic Rings of the Gnomes
Matilda's Dilemma
Ridgemount High
Space Mine
Star Trek Adventure
Tomb of the Grey Elf

Others:
Adventure
Adventure Copier
Adventure of Bigfoot
Big Texas Spy Adventure
Carfax Abbey
Castle David
Castle Hallways
Castle Mygog
Catacombs
Cave Maze
Classic Adventure
Clueless
Colossal Caves I
Colossal Caves II
Computorama
Crystal Cavern
DD Character Generator
Deathship I
Deathship II
Deliver the Cake
Devil's Dungeon
Disaster
Discovery at June Lake
Doors to Eden
Dragon's Lair
Dragon of Hong Kong
Dungeon
Dungeons I
Dungeons II
Escapade
Escape from Mars I
Escape from Mars II
First Days of Eden
Gerry's Place
Get the Gold
Gnome
High Adventure
Hobbit
Infocom's Games Rapid Loader
It's a Jungle Out There
Lost Gold
Monster Combat I
Monster Combat II
Moon Survival Adventure
MS-Adventure Series
Murder in the Mansion
Murder Mystery
Mystery at Raven Mountain
New Quest
Ninja
Nomad I
Nomad II
PS Adventure
Phantom of Blackmoore
Polar Trek
Pyramid
Quest
Quest 99
Quest for the King
Questsword
Raise Quest
Search for Lost Duchman's Gold
Secret Chambers
Shipwrecked
Smurf Adventures
Stellar Explorer
Superman
Swordquest
Swords and Sorcery
Treasure Quest
TOD Monster Editor
TOD Character Editor
Twelve Dungeons of Remzak
Valley
Welcome to Madam Fifi's Whorehouse
Welcome to Sherlock's Home

!!! and BONUS games and utilities

0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000

TRADING POST

FOR SALE - Joystick adaptors. Plugs into computer. Any commercial joystick can plug directly into the J1 or J2 socket without modification. $15. Phone Col 284 7783.

FOR SALE - As a complete unit, TIBUG's fully expanded system including console, modulator, transformer, Expansion box, RS232 card, Memory expansion card, Disk controller card, 2 slimline DSSD disk drives, and Extended Basic module and manual. $450 or $470 with free club membership. Phone Col 284 7783.

FOR SALE - Geneve 9640. Complete with manuals and lots of software. $500. Phone Larry 07 202 1884

WANTED - FOR $ALE - FOR EXCHANGE
What do you have or need that can be listed in this column? Contact Garry or Col with details.

MAGAZINE LIBRARY

MAGAZINES in TI Bug, as of 1 Feb `92

KC CONNECTION	Vol 8#11 - 10#8/9	
RYTE DATA	Ver 1#10/11 - 2#24	
MSP 99	Vol 2#8 -10#5	Oct. `83 - June `87
L.A. 99ers "Topics"	Vol 2#9 -10#11	Nov. `83 - Nov. `91
Asgard News	Vol 1#1 -2#1	
TI Sig		March`85 - April`89
Cin-Day News		Dec. `83 - May `89
Melbourne Times		Sept.`86 - Nov. `89
Byte Monger		Jan. `90 - June `91
Aticc	No. 22 - 43	Aug. `87 - Nov. `91
TI UP	Vol 3#1 -10#3	
Tas. User Group		Nov/Dec`83-May/June`87
Word Play	Vol 7#4 -10#11	April`88 - Dec. `91
Hunter Valley 99'er		June `85 - March`91
TI Shug		Nov. `87 - Dec. `91
Micropendium	Vol 4#5 -8#9	June `87 - Oct. `91

NEWSLETTER MAJOR ARTICLES

MICROPENDIUM

JUNE `87 VOL 4#5 --"Speech synthesizer as a proofreader."(orig.)
JULY `87 VOL 4#6 --"Pastors' User Group".(orig.)
AUG. `87 VOL 4#7 --"B-I-N-G-O, Testing MMM Expansion RAM."(orig.)
SEPT.`87 VOL 4#8 --"Variable Names in C99."(orig.)
OCT. `87 VOL 4#9 --"MDOS update."(orig.)
NOV. `87 VOL 4#10 -"Poker Solitaire."(copy)
DEC. `87 VOL 4#11 -"Putting XB in the console."(orig.)

*

FEB. `88 VOL 5#1 --"Programming in MDOS."(orig.)
MAR. `88 VOL 5#2 --"Signed numbers in BASIC, Text to speech."(orig.)
APR. `88 VOL 5#3 --"Back to the future."(orig.)
MAY `88 VOL 5#4 --"C99 Dictionary, Dallas T.I. Faire."(orig.)
JUNE `88 VOL 5#5 --"Croaker, C99 MINI-MEMORY."(copy.)

*

AUG. `88 VOL 5#7 --"Wipe-out race simulator."(orig.)

*

OCT. `88 VOL 5#9 --"Configuring Funnelweb."(copy)
NOV. `88 VOL 5#10 -"Constructing files in C99."(copy)
DEC. `88 VOL 5#11 -"Assembly Language, Geneve 9640."(copy)
JAN. `89 VOL 5#12 -"Hardware Project - Bypassing the 16 bit bus."(copy)
FEB. `89 VOL 6#1 --"512 colours in XBASIC for Geneve."(copy)

*
*

MAY `89 VOL 6#4 --"Database tutorial."(copy)

*

JULY `89 VOL 6#6 --"Polynominals in C99."(copy)
AUG. `89 VOL 6#7 --"Magnifying Sprites, Portable T.I."(orig.)
SEPT.`89 VOL 6#8 --"Now Performing Sprites."(orig.)
OCT. `89 VOL 6#9 --"Selecting a hard drive."(orig.)
NOV. `89 VOL 6#10 -"Expanding your TI's system 1, 2 meg card."(orig.)
DEC. `89 VOL 6#11 -"Address files in C99, Expanding your TI system.(o)

*
*

MAR. `90 VOL 7#2 --"The Game of YACHT, Do you need a modem?"(orig.)

APR. `90 VOL 7#3 --"Gram Devices, Pyramid solitaire."(orig.)
MAY `90 VOL 7#4 --"Tank Commander, EEPROMs the T.I."(orig.)
*
JULY `90 VOL 7#6 --"Speeding up TIBASE, Guide for peripherals."(o)
AUG. `90 VOL 7#7 --"Orchestration of noises,sounds and silence."(o)
*
OCT. `90 VOL 7#9 --"Knitting in BASIC, Game of Keno."(orig.)
NOV. `90 VOL 7#10 -"Rave PE/2 Expansion Box."(orig.)
DEC. `90 VOL 7#11 -"T.I. In Germany, Solve the puzzle of print codes.
JAN. `91 VOL 7#12 -"Advice on screen displays in Assembly."(orig.)
FEB. `91 VOL 8#1 --"BASIC Assembly."(orig.)
MAR. `91 VOL 8#2 --"Operation Desert Shield(game)."(orig.)
APR. `91 VOL 8#3 --"New XB for the TI - programming with tokens."(o)
*
JUNE `91 VOL 8#5 --"Harnessing the 99105 CPU chip."(orig.)
JULY `91 VOL 8#6 --"TI Accelerator, Deeper into Assembly."(orig.)
AUG. `91 VOL 8#7 --"Digital sound for the TI, Top-down programming." (o)
*
OCT `91 VOL 8#9 --"Fancy Printing, Scanning Graphics into your TI."(o)

ΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩ

WHAT'S NEWS

Remember the Wang monitors that the Sydney group were able to sell? It seems that they have been able to find some more. These are RGB monitors that are suitable for 9640 or 80 column devices (including the TIM) and the price is $110 + postage. They do not have a speaker so you need to arange sound seperately. Limited numbers will be available soon. See me (Garry) at the next meeting if you are interested.

The date of the Sydney Fair has been changed. It will be on the 28th and 29th of November (2 day affair) and at the Ashfield Boy's High School. Things look to be shaping up well.

I have placed some orders for software from the US. The order to Asgard went in last month so I should be arriving soon. I phoned in an order from Tex-Comp for a couple of copies of LOGO and Multiplan. If these arrive without problems, I will look at placing a larger order(you may remember that there is still a couple of modules on back order that we have not received). I was to order some software from Texaments but, just in the nick of time, I was informed that the Sydney group had a stock of TI-Base and TI-Artist Plus, and considerably cheaper them I could get them. I have ordered 3 of each from them, the price will be around $28 each. I will get a much reduced order off to Texaments shortly.

The following news comes from Jim Peterson (courtesy of TI-SHUG):

The following report on the Berlin TI Faire is condensed from a report by Jim Fetzner in Reflections and a letter from Alexander Hulpke published in the Lima newsletter.

* One of the features was a new P-System for the TI-99/4A and Geneve, not requiring the P-Code card. It was rewritten from scratch, supports 80-column cards and more disk drives, and runs 3 or 4 times faster. It is still in the process of final debugging and testing.

* The Berlin User Group was selling, for about $60, a device to install a Speech Synthesizer in the P-Box. Unlike the Rave card, this device contains the ROMs from the TEII cartridge, thereby allowing unlimited text to speech in any programming language and without the memory limitations of the disk-based Text-to-Speech.

* TI Club Leipzig was demonstrating a true video digitizer for the P-box which accepts a video input through standard ports and digitizes it in real time (on the fly!) into a My-Art/YAPP format picture suitable for display on an 80-column card. It uses very expensive new chips and therefore costs about $350.

* The same group demonstrated modified routines for disk access which allowed direct controller programming, avoiding the TI file system overhead, thereby achieving speed comparable to a hard drive. They also discussed an AT-Bus hard disk, but had not written a DSR for it.

* Another group was demonstrating, and expects to have available soon for about $100, a TI Hex-Bus interface which allows the TI-99/4A to be used with the many peripherals which Texas Instruments released for the CC-40 hand-held computer.

* Also displayed, and costing about $14, was a simple chip and motherboard for the Mechatronics 80-column card which expands the number of available colours from 256 to 256 out of 256,000. Alexander Hulpke suggests that it could be adapted for the DIJIT card or the Geneve if an unused 8-byte area could be found to decode.

* Also released was XB3, a complete rewrite of TI Extended Basic which is 16k larger, offers dozens of new

commands, is 100% compatible with existing XBasic programs and runs them an average of 2 to 4 times faster! However, it requires a GRAM device. It will soon be released on disk by Asgard Software, and will later be available as a module. I am not sure if this module consists of 3 ROM banks at >6000 and will therefore run only on the Mechatronics GRAM card, not on the Geneve or Gram-Kracker or other simulators with just two banks.

* New software included a CAD program by Henrik Wedekind, described as very user friendly, for the TI with 80-column device; CRASH by Peter Muys for the Geneve, a stock market program claimed to be equal to very expensive PC programs; and an XBasic compiler for the TI, written in Basic (and not able to compile itself into assembly) which compiles everything but SUBs, DEFs and arrays, but takes hours to do so.

It's amazing how advanced the IBM PC computers. I was reading in the newspaper last week-end that a new video card has just been released that will greatly speed up some programs (Windows etc). The secret is that it has a video chip with it's own memory that will handly the image on the screen, freeing up the CPU for running the program. The TI99/4 was using this system in 1980, 12 years ago. Boy, these guy's are quick.

I have just got off the phone to OPA. God only knows what is happeneing over there, it seems that they don't. The stories so far have been - 1. they will be sent next week (2 months ago), 2. They were sent in groups of 5 since early January, 3.(the latest) they are waiting on some crystals. I hope to make some sense out of it by the next meeting. Stay tuned.

ΘΘΘΘΘΘΘΘΘΘΘΘΘΘΘΘΘΘΘΘΘΘΘΘΘΘΘΘΘΘΘΘΘΘ

LAST MEETING

Opened 8.45pm

Minutes of previous meeting accepted

Business arising from the minutes:

Aditional keys for the room in which we hold the meeting have been cut.

No software from Asgard yet.

Correspondance accepted.

Treasurer reported that the balance was $965.06 and that some active members still had not paid membership fees.

Librarians' Reports - Blank disks available, good usage of the module library.

General Business:

Speech adapter boards start production next week. The group has ordered 9 and they will cost about $30 or so. John Reynolds is taking the orders.

Col is looking for a speech synthasizer.

It was moved that the treasurer issue a cheque to purchase software from Texaments.

The TIMs haven't arrived as yet.

It was agreed that we place a small order from Tex-Comp to make sure it was delivered OK before and substantial order was placed.

The EPROM for the Col's Console expander will also include the Tape Loader and Word Processor programms.

$10.45 refunded to John Peacock for postage.

Meeting closed 9.45 pm.

Larry demonstrated Checkers and TI-Pei that he had received from Asgard. TI-Pei needs an understanding of the game to play properly. Checkers was very popular, we even beat the computer.

ΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩ

FRED'S THINGAMEE PARTS AND MULTIPLAN

by Garry Christensen

Fred's Thingamee Parts is a business that specialises in spare parts for thingamees. Fred needs to keep record of the sales and of his profits. He sets out a page like this:

PART NAME	Frap	Gort
BUYING PRICE	$22.48	$16.75
SELLING PRICE	$25.98	$20.00
PROFIT PER SALE	$ 3.50	$ 3.25
NUMBER SOLD	10	6
NUMBER IN STOCK	25	12
SALES	$259.80	$120.00
PROFIT	$ 35.00	$ 19.50

TOTAL PROFIT $54.50

Fred's business is not really big because a thingamee has only 2 parts so he needs to keep a close eye on sales and profits to see if he will have enough to do the shopping at the end of the week. Each night he updates the figures (he has to work in pencil so they can be rubbed out). Fred adds the number sold in that day to the total sold, subtracts it from the number in stock, calculates the new sales and profit figures and finally produces the total profit so far this week.

Fred is finding that the business in thingamee parts isn't as good as it used to be so he decides to increase the price of the frap and the gort. How much will he have to increase them the make the business viable? Fred gets out the calculator and sets to work.

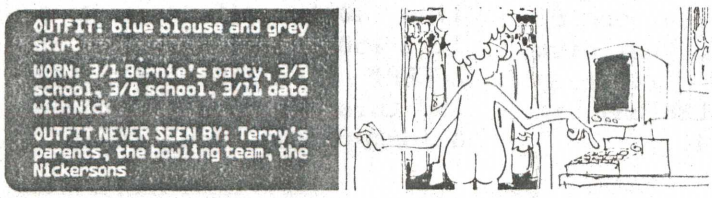
Some time later, and Fred is still in business. Things are actually going quite well now because Joe's Whatzit Bits no longer stocks thigamee parts so Fred has no competition. All is really going well but Fred doesn't like the paperwork that he has to do each night and the cost of pencils and erasers is an expense that he could do without. During a particularly good month, a computer salesman calls and introduces Fred to electronic book keeping. He concentrates on a program called Multiplan and promises that this spreadsheet will reduce paperwork and eraser and pencil consumption.

Fred is quite taken by Multiplan because it works exactly the same way that he keeps his books now. The salesman shows Fred how the screen is divided into rows and column where he can enter either names or a number. He also shows Fred how equasions can be set up so that profits can be calculated by subtracting costs from income. The best thing is that the computer does all the maths, so the deal is signed. Fred's business takes on a new perspective.

Prior to 1980, Microsoft Multiplan was only available on large computer systems. With the coming of the home computer, Texas Instruments collaborated with Microsoft to bring spreadsheets (called electronic worksheets then) to the small computer. Multiplan is still used today, on PCs mostly, and the principles of operation have changed little.

I will be demonstrating the use of this program at the next meeting, but till then I will leave you with a copy of the introduction in the Multiplan manual.

THE "WHAT I WORE" PROGRAM



LETTER TO THE EDITOR

Dear Garry,

I recently finished writing the above program which is intended to accept morse code through the computer keyboard and print the translation to the screen. My ultimate aim, as a novelty, is to combine it with a previous program which does the reverse, viz. creates a random sentence and changes it to morse code, which will hopefully suffice as an input to this latest program. However, this latest effort needs playing around within order to fine tune it and generally get the timing right. This is done through adjusting the values in statements 140 thru 180, a task that I thought you might throw open to any interested members of the user group. I did manage to 'send' the word NORTHERN a couple of times but I've fiddled about with the variables since then and forgotten what they were.

The statement at 400 is a dummy statement which replaces a CALL SOUND. The relevant variables are:- LDOT - length of a dot, LDASH - length of a dash, SDOT - length of time between dots and dashes, SDASH - length of time between letters and SSPACE - length of time between words. I am informed by BROWNS BOOK ON SIGNALLING that the correct ratio for these is 1 unit between dots and dashes, 3 units between letters, and 7 units between each word but that's not to say that these rules should be adhered to in this case.

Anyway, if you are short of material for the next newsletter, give this a whirl. Incidentally, I recently bought a CONSOLE WRITER 2.1 WP from Navarone, a stand alone effort which has no bell at the end of the line - hence the occasional wrap around. Hope you will bear with me on this.

Yours Truely,

A. Lanning

Hi Tony,

Thanks for the letter. As you can see I have included it in the newsletter. I am rarely short of material but I am always short of contributions from TI-BUG members. Thank you for writing and sharing this program with other members. I know that there are others whose interests lie in this area.

Here's a thought for the amateur radio people. As you know, the TI can produce sound so a program to convert text into morse code is not difficult to write, nor would it be difficult to connect the output to a transmitter. With the help of a little assembly program, the signal from your receiver can be put into the cassette port so that the TI could also decode incoming messages and display them on the screen. This system may already exist, I don't know. Maybe you could write and let me know. If you are interested in developing something like that, I could help out with the assembly part.

All the best,

Garry

PS. To all the other members of the group, your contributions would also be welcome.

```
100 CALL CLEAR
110 OPTION BASE 1
120 DIM LETR$(126)
130 Z=1
140 LDOT=2
150 LDASH=6
160 SDOT=3
170 SDASH=9
180 SSPACE=15
190 FOR K=101 TO 126
200 READ LETR$(K)
210 NEXT K
220 RESTORE
230 DATA .-,-...,-.-.,-..,.,
..-,--.,....,..,.---,-.-,.-.
.,--,-.,---,.--.,--.-
240 DATA .-.,...,-,..-,...-,
.--,-..-,-.--,--..
250 PRINT :"READY FOR MESSAG
E"
260 CALL KEY(3,KI,ST)
270 IF ST=0 THEN 260
280 CALL KEY(3,K,S)
290 IF S=-1 THEN 400
300 IF B>SSPACE THEN 510
310 IF A=0 THEN 360
320 IF A>LDOT THEN 360
330 LETR$(Z)=LETR$(Z)&"."
340 A=0
350 GOTO 380
360 LETR$(Z)=LETR$(Z)&"-"
370 A=0
380 B=B+1
390 GOTO 280
400 ALP=0
410 IF B>SDOT THEN 420 ELSE
480
420 IF B<=SDASH THEN 470
430 Z=Z+1
440 LETR$(Z)="32"
450 Z=Z+1
460 B=0
470 A=A+1
480 GOTO 280
490 GOSUB 520
500 GOSUB 630
510 GOTO 250
520 FOR K=1 TO 100
530 IF LETR$(K)="" THEN 620
540 IF LETR$(K)="32" THEN 60
0
550 FOR K1=101 TO 126
560 IF LETR$(K)=LETR$(K1)THE
N 580
570 NEXT K1
580 PRINT CHR$(K1-36);
590 GOTO 610
600 PRINT CHR$(32);
610 NEXT K
620 RETURN
630 FOR K=1 TO 100
640 LETR$(K)=""
650 NEXT K
660 Z=1
670 A=0
680 B=0
690 RETURN
```

TIPS FROM THE TIGERCUB

#38

TIGERCUB SOFTWARE
156 Collingwood Ave.
Columbus, OH 43213

I have discovered a rare bug in the 28-Column Converter, published in Tips #18, which will cause an I/O 25 ERROR if the very last line of the program being converted happens to have exactly 80 characters. You can fix it by adding a line -
215 IF EOF(1)=1 THEN 260

There is also a rare bug in the SIDEWAYS subroutine on my Nuts & Bolts #2 disk, which prevents turning some redefined character sets sideways. If you are one of those who BOUGHT that disk from me, you can fix it by changing the L=LEN(B$) in line 21639 to L=64.

I was in too much of a hurry to go fishing when I put the last couple of Tips together. In the Gordian Knot in Tips #35, I left out some essential instructions. Please add -
131 DISPLAY AT(11,1):" When you cross your track,":"press O to go over, U to go":"under, C to go across."

To make that fit, you will have to change the DISPLAY AT in line 130 to (8,1), in line 140 to (15,1) and in line 150 to (20,1), also the ACCEPT At in 160 to (20,11). And this change will prevent a lockup when you reach a

```
border -
200 D=D-1 :: IF ABS(D-D2)=2
OR R+(D=1)=0 OR R-(D=3)=25 O
R C+(D=4)=2 OR C-(D=2)=31 TH
EN 180 :: GOSUB 510 :: IF D<
>D2 THEN GOSUB 450
```

I wrote the dulcimer music in Tips #36 in Basic, but I forgot to test it in Basic. It actually runs much better in Extended Basic, but will run fairly well in Basic if you delete the delays in lines 280 and 300.

If you liked the ESCHER ART in Tips #37, these modifications will improve it considerably -

```
110 DISPLAY AT(12,1):"Press
-": :" Q for new pattern":"
B to change background":" F
to change foreground":" R to
 reverse colors": : :"Any ke
y to start"
280 A=INT(6*RND+3):: H=INT(2
4/A):: RX=24-H*A :: HC=INT(2
8/A):: CX=28-HC*A :: W=ABS(H
C/2=INT(HC/2))-(RX>0):: DIM
M(8,8):: FOR P=1 TO A
330 IF K<>66 THEN 346
340 BC=BC+1+(BC=16)*15 :: IF
 BC=F THEN 340 ELSE 347
346 IF K<>70 THEN 360 :: F=F
+1+(F=16)*15 :: IF F=BC THEN
 346
347 FOR S=7 TO 14 :: CALL CO
LOR(S,F,BC):: NEXT S :: GOTO
 310
350 ! **DELETED LINE **
360 IF K<>ASC("R")THEN 310 :
: T=F :: F=BC :: BC=T :: GOT
O 347
600 GOSUB 900 :: FOR T=1 TO
A :: DISPLAY AT(R-1+T,C):M$(
V,T):: NEXT T :: NEXT C
601 IF CX>0 THEN AA=A :: GOS
UB 800
605 GOSUB 1000 :: NEXT R
606 IF RX=0 THEN 610
607 GOSUB 1000 :: FOR C=1 TO
 A*HC STEP A :: GOSUB 900 ::
 FOR T=1 TO RX :: DISPLAY AT
(R-1+T,C):M$(V,T):: NEXT T :
: NEXT C
608 IF CX>0 THEN AA=RX :: GO
SUB 800
800 GOSUB 900 :: FOR T=1 TO
AA :: DISPLAY AT(R-1+T,C):SE
G$(M$(V,T),1,CX);:: NEXT T :
: RETURN
900 V=V+1+(V=4)*4 :: RETURN
1000 V=V+W :: V=V+(V>4)*4 ::
 RETURN
```

I had a letter from a teacher who was using the PRK module to keep student grades, and wanted to know how to average them. It can be done, but is so impractical that I wrote this program. While I was at it, I speeded up the loading and saving to cassette greatly by converting the grades to an ASCII string and combin- the student's name and all grades into one record.

```
100 DIM N$(50),T(50,20)
110 CALL CLEAR
120 PRINT "       TEACHER'S
HELPER": : : :
130 REM  - by Jim Peterson
140 PRINT "(1)CREATE A FILE?
":"(2)ADD TO FILE?":"(3)LOAD
 A FILE?":"(4)SAVE A FILE?":
"(5)PRINT A FILE?"
150 PRINT "(6)CORRECT A FILE
?":"(7)COMPUTE AVERAGES?":"(
8)QUIT?"
160 CALL KEY(0,K,S)
170 IF (S=0)+(K<49)+(K>56)TH
EN 160
180 ON K-48 GOTO 190,250,610
,800,380,990,1120,1510
190 X=0
200 INPUT "SUBJECT? ":S$
210 GOSUB 1370
220 INPUT "TEST #? ":N
230 GOSUB 1440
240 GOTO 140
250 PRINT :;:"(1)ADD NAMES?"
:"(2)ADD GRADES?"
260 CALL KEY(0,K,S)
270 IF (S=0)+(K<49)+(K>50)TH
EN 260
280 ON K-48 GOTO 290,310
290 GOSUB 1370
300 GOTO 140
310 INPUT "TEST #? ":Q
320 IF T(1,Q)=0 THEN 350
330 PRINT :;:"TEST #";STR$(Q
);" ALREADY RECORDED"
340 GOTO 140
350 N=Q
360 GOSUB 1440
370 GOTO 140
380 CALL CLEAR
390 PRINT "OUTPUT TO":"(1)SC
REEN?":"(2)PRINTER?"
400 CALL KEY(0,K,S)
410 IF (S=0)+(K<49)+(K>50)TH
EN 400
420 IF K=49 THEN 460
430 INPUT "PRINTER DESIGNATI
ON? ":P$
440 OPEN #2:P$
450 F@=2
460 PRINT "PRESS ANY KEY TO
PAUSE": :
470 PRINT #F@:S$: :
480 FOR J=1 TO X
490 PRINT #F@:"":N$(J)&" ";T
AB(10);
500 FOR K=1 TO HN
510 PRINT #F@:T(J,K);
520 NEXT K
530 CALL KEY(0,K,S)
540 IF S<>0 THEN 530
550 NEXT J
560 PRINT #F@
570 IF F@=0 THEN 140
580 F@=0
590 CLOSE #2
600 GOTO 140
610 PRINT :;:"(1)CASETTE?":"
(2)DISK?"
620 CALL KEY(0,K,S)
630 IF (S=0)+(K<49)+(K>50)TH
EN 620
640 ON K-48 GOTO 650,670
650 OPEN #2:"CS1",INPUT ,FIX
ED
660 GOTO 690
670 INPUT "FILENAME? DSK":F$
680 OPEN #2:"DSK"&F$,INPUT
690 INPUT #2:X,HN,S$
700 FOR J=1 TO X
710 INPUT #2:K$
720 N$(J)=SEG$(K$,1,POS(K$,C
HR$(255),1)-1)
730 K$=SEG$(K$,POS(K$,CHR$(2
55),1)+1,255)
740 FOR K=1 TO HN
750 T(J,K)=ASC(SEG$(K$,K,1))
-50
760 NEXT K
770 NEXT J
780 CLOSE #2
790 GOTO 140
800 PRINT :;:"(1)CASETTE?":"
```

```
(2)DISK?"
810 CALL KEY(0,K,S)
820 IF (S=0)+(K<49)+(K>50)TH
EN 810
830 ON K-48 GOTO 840,860
840 OPEN #2:"CS1",OUTPUT,FIX
ED
850 GOTO 880
860 INPUT "FILENAME? DSK":F$
870 OPEN #2:"DSK"&F$,OUTPUT
880 PRINT #2:X:HN:S$
890 FOR J=1 TO X
900 K$=""
910 FOR K=1 TO HN
920 K$=K$&CHR$(T(J,K)+50)
930 NEXT K
940 PRINT #2:N$(J)&CHR$(255)
&K$
950 K$=""
960 NEXT J
970 CLOSE #2
980 GOTO 140
990 CALL CLEAR
1000 INPUT "STUDENT'S NAME?
":Q$
1010 FOR J=1 TO X
1020 IF N$(J)=Q$ THEN 1060
1030 NEXT J
1040 PRINT :;:"NAME NOT FOUN
D": :
1050 GOTO 140
1060 INPUT "CORRECT WHICH TE
ST? (0 TO QUIT) ":C
1070 IF C=0 THEN 1110
1080 PRINT :;:N$(J);"'S TEST
 #";STR$(T(J,C)): :
1090 INPUT "CORRECT TO? ":T(
J,C)
1100 GOTO 1060
1110 GOTO 140
1120 CALL CLEAR
1130 PRINT "OUTPUT TO":"(1)S
CREEN?":"(2)PRINTER?"
1140 CALL KEY(0,K,S)
1150 IF (S=0)+(K<49)+(K>50)T
HEN 1140
1160 IF K=49 THEN 1200
1170 INPUT "PRINTER DESIGNAT
ION? ":P$
1180 OPEN #2:P$
1190 F@=2
1200 PRINT #F@:S$
1210 FOR J=1 TO X
1220 PRINT #F@:N$(J);" AVERA
GE ";
1230 FOR K=1 TO HN
1240 TT=TT+T(J,K)
1250 NEXT K
1260 AV=TT/HN
1270 TAV=TAV+AV
1280 PRINT #F@:AV
1290 TT=0
1300 NEXT J
1310 PRINT #F@:"CLASS AVERAG
E ";TAV/X
1320 TAV=0
1330 IF F@=0 THEN 1360
1340 F@=0
1350 CLOSE #2
1360 GOTO 140
1370 PRINT :;:"STUDENT'S NAM
ES - ":"type END when finish
ed": :
1380 X=X+1
1390 M$="NAME #"&STR$(X)&" "
1400 INPUT M$:N$(X)
1410 IF N$(X)<>"END" THEN 13
80
1420 X=X-1
1430 RETURN
1440 FOR J=1 TO X
1450 M$=N$(J)&"'S GRADE? "
1460 INPUT M$:T(J,N)
1470 NEXT J
1480 IF N<HN THEN 240
1490 HN=N
1500 RETURN
1510 END
```

The reason that 50 is added to the value in line 920, before saving, and subtracted again in line 750 after loading, is because of a quirk of the computer that I don't recall seeing in print anywhere. Did you know that INPUT will read a string beginning with ASCII 0, 2, 4, 7, 10, 12, 14, 18, 20, 26, 27, 31, 32, or 44 as a null string (a blank), and will drop these characters at the end of a string? And ASCII 32 will be dropped at the beginning or end of a string. And ASCII 0 within a string, or ASCII 34 anywhere, will crash, while ASCII 44 within a string will lose the rest of the string. I should have known what ASCII 0, 32 (the space), 34 (quotes) and 44 (comma) would do, but why the others?

LINPUT will accept anything, of course, but I wanted to keep this in BASIC for the teachers who are struggling along without the XBasic module or disk drive.

Chick De Marti published in LA 99ers TOPICS the surprising discovery that PRINT USING and DISPLAY USING can read the IMAGE format from a variable, array or string!

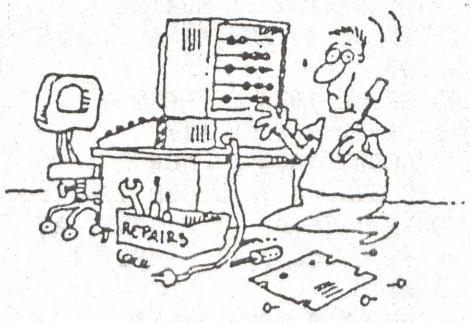
Which led me to some fooling around -

```
100 !PRINT USING DEMO by Jim
 Peterson, based on a discov
ery by Chick De Marti
110 CALL CLEAR :: RANDOMIZE
:: CALL SCREEN(5):: FOR S=2
TO 14 :: CALL COLOR(S,S,S)::
 NEXT S
120 N=INT(13*RND+1):: C$=CHR
$(8*N+32-(N=4)*11)
130 FOR J=N TO 12 :: A$=RPT$
(" ",J)&"#"&RPT$(" ",26-J*2)
&"#" :: PRINT USING A$:C$,C$
 :: NEXT J
140 FOR J=12 TO N STEP -1 ::
 A$=RPT$(" ",J)&"#"&RPT$(" "
,26-J*2)&"#" :: PRINT USING
A$:C$,C$ :: NEXT J :: GOTO 1
20
```

Here is one last Tigercub challenge. What is the longest possible one-liner? And what is the longest possible one-liner that actually does something?

MEMORY FULL

Jim Peterson



BASIC COMPUTERS

The Inside World

Last month we looked at memory and described it like pidgeon holes for storing information. Data can be written into a byte in memory, where it will replace whatever was there before it. During a read, only a copy of the data is sent so that the value of that byte remains unchanged. The address of each byte is 16 bits long and they can be written as a 4 digit code using 0 to 9 and A to F, the '>' sign means that it is a code and not a number. The CPU has a register called the Program Counter that tells it the address in memory where the next instruction is stored. The CPU 'fetches' the instruction and executes it, then prograsses on to the next instruction. The list of instructions is called a program and the language is called machine code. Finally there are 2 main types of memory, RAM and ROM. The CPU can change the values stored in RAM memory but the data in ROM is fixed.

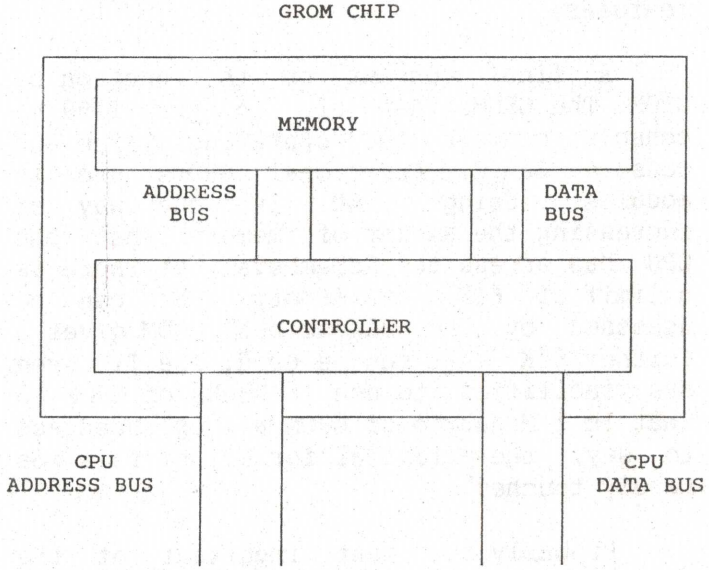
This month I will be talking about some of the other components of a computer. There is obviously more to a computer that CPU and memory. There has to be a way for the computer to communicate to the 'outside world' otherwise there would be no way to load new programs or data, or to display the results of a program. There are also some devices that help or enhanse the operation of the computer, like speech, so these must also have a link to the CPU.

All computers use a method called memory mapping and the TI99/4A is no different but it also has a additional feature call the communications register unit (CRU). I will leave the CRU for next month because it is used for most of the external communication and control. The memory mapped devices are predominantly for internal use.

In general, a memory mapped device is one where the CPU accesses it through a particular memory address. That address has no RAM but the device responds like it was a byte of memory so the CPU knows no different. Back to the pidgeon hole analogy. Memory mapped devices are like someone standing behind the pidgeon holes and when data is placed in a particular hole, that person takes it and performes the required task. These devices use the address and data bus to communicate to the CPU.

When a particular address is placed on the address bus, the device recognises it and looks at the data bus if it a write operation or places data on the data bus if it is a read. Most devices use several addresses, called ports. Some ports can only be written to and some can only be read from. Trying to write to a read only port, or visa-versa, will generally cause no problems, that data will be ignored.

The first of these devices is called GROM, that stands for Graphics Read Only Memory. GROM is a type of ROM in that it contains data that can not be altered by the processor but it is not random access. GROM has an intermediate 'controller' to look after access to the ROM. Consider it like this.



This can be a bit confusing but I'll try to take it slowly and simply. To access data, the CPU will firstly have to tell GROM which byte it wants. GROM address codes are the same as normal memory. Let's say that you want to access GROM address >1000 (usually described as >1000G). The first thing is to write that code to the GROM Write Address Port. It is set to memory address >9C02, so write >1000 to address >9C02. The GROM controller now knows which byte of GROM memory you want to use. Now you need to get the byte by reading from the GROM Read Data Port at address >9800. When the CPU tries to read

from this address, the GROM controller fetches the byte and places it on the data bus for the CPU to read.

The GROM controller has one more function. It increments the value in the address register so that it points to the next byte. That means that if the CPU wants to read successive bytes from GROM, it doesn't have to set the address each time, just keep reading from the Read Data port. There are other ports available for GROM (like Write Data if you have GRAM) but I think that it gives you the idea.

A thought just occurred, do you know what GRAM is. Think back to ROM and RAM, the RAM can be changed by the CPU. Likewise with GROM and GRAM. The CPU can write to and change the values stored in GRAM. There are several GRAM devices around that allow you to copy GROMs into them (load them from disk) and use the values in GRAM instead of GROM. The advantage here is that the data can be edited to change some of the features.

A final comment on the function of GROM. The GROMs that are in the TI99/4A console contain the operating system and console BASIC. Additional GROMs are in modules. Using GROM is one way of increasing the amount of memory that the CPU has access to. Remembers that there is a limit of 64K of memory that can be accessed by the address bus. GROM gives a further 64K that can be used. The TI even has facilities to use 16 banks of GROM so that is 1 Megabyte of GROM all up. Needless to say, the potential for all of this was barely touched.

Probably the most important of the memory mapped devices in the Video Display Processor (VDP). This device is responsible for creating a video signal that can be sent to your monitor or TV. Without it, the computer would be almost useless, imagine using a computer that has no monitor. The VDP uses its own memory to store what characters are to be displayed on the screen and where they are, what the characters look like and what colour they are. The video processor scans its memory many times each second and uses the information to create the video signal. As well as this, it will communicate with the CPU to allow the data in it's memory to be written to or read.

Once again there are ports. The VDP Write Address port (>8C02) indicates which byte in video memory to be accessed, the VDP Read Data port (>8800) allows that byte to be read by the CPU while the VDP Write Data port (>8C00) allows the CPU to write a new value to it. There is only 16K of video memory on the TI99/4A. 80 column devices supply much more but the method of access is still the same.

The address of the byte in video memory automatically increments after each operation, write or read.

There are 2 other memory mapped devices that need to be mentioned here. One is the sound chip and the other is the speech synthasizer. The sound port is at >8400 and is a write only port. I won't go into the structure of sound tables at this time but suffice it to say that the bytes written to this port will control the sound that is generated.

There are 2 ports for the speech synthasizer, the Speech Write port (>9400) and the Speech Read port (>9000). These ports allow the CPU to write data to and read data from the speech synthasizer.

Those are the memory mapped ports defined in the console. All of these devices have the one thing in common, how they communicate with the CPU. All are attached to the address bus and only one particular address code will activate them. Depending on the port being addressed, these devices read data from the data bus or place data on it so the CPU can write or read as appropriate. Of course, the CPU needs a program in memory so that it can be directed to perform the correct functions otherwise the results are called a lock-up. You may have experienced a lock-up yourself where there is a pretty display of flashing lights on the screen and garbled noises coming fron the speaker. This is caused by indiscriminant writing to memory mapped devices. It doesn't harm the computer but the only solution is the on/off switch. Fortunately, instances are fairly rare.

Enough of devices for this month. Next month I will discuss how the computer talks to the outside world through the CRU.

DISK LIBRARY

FILE NAME: TIART

TI-ARTIST PICS,RLE,INSTANCES.

FILENAME.	TYPE.	DSK.
ALICEWO1_P	PRG	276
ALICEWO2_P	PRG	276
ALICEWO3_P	PRG	276
ALICEWO4_P	PRG	277
ALICEWO5_P	PRG	277
ALLEN,W_P	PRG	273
APPLE_I	DV080	256
BARNEY_P	PRG	275
BASEBALL_I	DV080	256
BIPLANE_I	DV080	256
BIRD_I	DV080	256
BLUGOSI_P	PRG	273
BERGMAN_P	PRG	273
BISSET,J_P	PRG	273
BLOOM_P	PRG	273
BOGART,H_P	PRG	273
BRUSH_I	DV080	256
BUGS3_P	PRG	275
CANNON_I	DV080	256
CAR_I	DV080	256
CASSETTE_I	DV080	256
CELTIC_P	PRG	275
CHALANGR_P	PRG	275
CHAMPAGN_I	DV080	256
CHESHIRE_P	PRG	277
CITY_P	PRG	275
CLINT_P	PRG	273
CLOCK_I	DV080	256
CLOWNS_P	PRG	275
COKE_I	DV080	256
COKE_P	PRG	275
COLBERT_P	PRG	275
COMB_I	DV080	256
COPTER_I	DV080	256
COSBY_P	PRG	275
CROSS_I	DV080	256
DAFFY_P	PRG	275
DEAN,J_P	PRG	273
DELF,ANA_P	PRG	273
DENEUVE_P	PRG	275
DIETRICH_P	PRG	275
DISKETTE_I	DV080	256
DISNYCTN_P	PRG	277
DISNYLGO_P	PRG	277
DNGERFD_P	PRG	277
DONALD_P	PRG	275
DONALDUK_P	PRG	277
DOUGLS,K_P	PRG	273
DOUGLS,M_P	PRG	273
DRAGON1_P	PRG	273
DRAGON2_P	PRG	273
DROP_P	PRG	275
DRUM_I	DV080	256
DWARFS_P	PRG	275
EAGLE_P	PRG	275
EARTH_P	PRG	277
EINSTEIN_P	PRG	273
ELEFUNT_P	PRG	275
ELVIRA_P	PRG	273
ENTPOOP_P	PRG	275
ENTRPIRZ_P	PRG	275
EPCOTCTR_P	PRG	277
FAIRBNKS_P	PRG	273
FISH_P	PRG	275,277
FISHBIRD_P	PRG	273
FLAG_I	DV080	256
FLYNN,E_P	PRG	277
FONDA,J_P	PRG	273
FONTART1_P	PRG	298
FONTART2_P	PRG	298
FONTART3_P	PRG	298
FONTREAD	DV080	298
FORK_I	DV080	256
FOX_P	PRG	275
FROG_I	DV080	256
G/BUST_P	PRG	275
GABLE,C_P	PRG	273
GARBO,G_P	PRG	273
GARDNER_P	PRG	273
GIRLS_P	PRG	275,277
GLOBE_I	DV080	256
GOOFY_P	PRG	275
GRANT,C_P	PRG	273
HANDS_P	PRG	275
HELMET_I	DV080	256
HILSTRT1_P	PRG	273
HILSTRT2_P	PRG	273
HORNET_P	PRG	273
HORSE_I	DV080	256
HORSE_P	PRG	275
HOUSE_I	DV080	256
HOWDY_P	PRG	276
HUBBLE_P	PRG	273
INGRAM_I	DV080	256
IRON_I	DV080	256
JET_I	DV080	256
KALIOP_P	PRG	273
KARATE_P	PRG	276
KENT,C_P	PRG	273
KNIFE_I	DV080	256
LABELKAKER	DV080	256
LAUNCH_P	PRG	276
LIBERTY_P	PRG	276
LIGHT_I_P	DV080	256
MADHATTR_P	PRG	277
MCAULIFF_P	PRG	276
MERC7_P	PRG	276
MERLIN_P	PRG	276
MICKEY01	PRG	277
MICKEY02	PRG	277
MICKEY03	PRG	277
MICKEY04	PRG	277
MICKEY05	PRG	277
MICKEY06	PRG	277
MILLO/BC_P	PRG	276
MINI99_I	DV080	256
MINNIE01_P	PRG	277
MINNIE02_P	PRG	277
MOUSLAND_P	PRG	277
MRSPOCK_P	PRG	276
NASA_P	PRG	276
NEUMAN_P	PRG	277
NINJATUR_P	PRG	277
NOTEPAD_IP	DV080	256
OLIVER2_P	PRG	276
OPUS_P	PRG	276
P-KNIFE_I	DV080	256
PAGODA_P	PRG	276
PAW_I	DV080	256
PICKFORD_P	PRG	277
PIN_I	DV080	256
PIRATE_P	PRG	276
R-MAN_P	PRG	277
REAGAN,R_P	PRG	277
RIBBON_I	DV080	256
ROCKWL01_P	PRG	277
ROCKWL02_P	PRG	278
ROCKWL03_P	PRG	278
ROCKWL04_P	PRG	278
ROCKWL05_P	PRG	278
ROCKWL06_P	PRG	278
SAILBOAT_I	DV080	256
SAPPHO_P	PRG	278
SATURN5_P	PRG	276
SGTMAJ_P	PRG	276
SHIP_I	DV080	256
SHUTTLE_I	DV080	256
SHUTTLE_P	PRG	276
SIGNAL_I	DV080	256
SKYLAB_P	PRG	276
SMILE2_I	DV080	256
SMILE_I	DV080	256
SNOOPY_I	DV080	256
SNOOPY_P	PRG	276
SNOWHITE_P	PRG	278
SPACEWAK_P	PRG	276
SPIRTE76_P	PRG	278
SPOON_I	DV080	256
STARTK4_P	PRG	278
STARTK5_P	PRG	278
STARTK7_P	PRG	278
STARTK8_P	PRG	278

STARWAR_P	PRG	276
STAR_I	DV080	256
STEREO2_I	DV080	256
STEREO_I	DV080	256
STOP_I	DV080	256
STRIPE_P	PRG	278
SUPERCAR_P	PRG	276
SWITCH_I	DV080	256
SWORDFIT_P	PRG	276
TANK_I	DV080	276
TAPE_I	DV080	276
TARGET_I	DV080	276
TELESCOP_I	DV080	276
TEMPLE,S_P	PRG	278
TENNIS_I	DV080	256
TINKER_P	PRG	278
TRUCK_I	DV080	256
TRUMPET_I	DV080	256
TV_I	DV080	256
UFO_I	DV080	256
VISA_I	DV080	256
WARGIRL_P	PRG	278
WASHNGTN_P	PRG	278
WAYNE,J_P	PRG	278
WC/MAE_P	PRG	276
WEATHER_P	PRG	276
WINIPOO1_P	PRG	278
WINIPOO2_P	PRG	278
WOOKIE_P	PRG	278

MAXRLE PICS, RLE. ETC. FILENAME MAXRLE

32KMOD	24	DV080	MAXRLE	246
3STOOGES	17	DV080	MAXRLE	246
C-BRINKLEY	123	DV080	MAXRLE	246
CATH/DEN	29	DV080	MAXRLE	247
CAVEGIRL	18	DV080	MAXRLE	247
CELTIC	51	DV080	MAXRLE	247
CHESS	41	PRG	MAXRLE	349
COGNAC	25	DV080	MAXRLE	247
DRAGON2	37	DV080	MAXRLE	247
E-PRESLEY	21	DV080	MAXRLE	246
ELVIRA	16	DV080	MAXRLE	247
FOXY/RLE	25	DV080	MAXRLE	247
FUTURE	32	DV080	MAXRLE	247
HOWDY	25	DV080	MAXRLE	247
INDY	49	DV080	MAXRLE	247
KNIGHTS	20	PRG	MAXRLE	349
MADONNA	33	DV080	MAXRLE	247
MAX-RLE	35	DF080	MAXRLE	246
MAX-RLE	35	DF080	MAXRLE	349
MX-RLE/DOC	27	DV080	MAXRLE	246
MX-RLE/DOC	28	DV080	MAXRLE	349
SNOOPY	92	DV080	MAXRLE	246
STAN&OLLIE	17	DV080	MAXRLE	246
TEMPLE,S	43	DV080	MAXRLE	349
TIGER	92	DV080	MAXRLE	349
WARGIRL	33	DV080	MAXRLE	349
WARRIOR	30	DV080	MAXRLE	349

GRAPHX CLIPBOARDS. FILENAME GRAPHX.

READ-ME	3	DV080	GRAPHX	299
*README	8	DV080	GRAPHX	381
--README--	4	DV080	GRAPHX	300
-DISKDOC	4	DV080	GRAPHX	299,300
??	4	DF080	GRAPHX	299
???	2	IF008	GRAPHX	299
????	16	IV254	GRAPHX	299
?CH	5	IF065	GRAPHX	299
?SC1	4	IF029	GRAPHX	299
?SC2	4	IF029	GRAPHX	299
?SC3	4	IF029	GRAPHX	299
?SC4	4	IF029	GRAPHX	299
CART1DOC	4	DV080	GRAPHX	299
CART2DOC	4	DV080	GRAPHX	300

C_3DLG/A1	14	PRG	GRAPHX	299
C_3DLG/A2	14	PRG	GRAPHX	299
C_3DLG/A3	7	PRG	GRAPHX	299
C_3DLG/B1	14	PRG	GRAPHX	299
C_3DLG/B2	14	PRG	GRAPHX	299
C_3DLG/B3	7	PRG	GRAPHX	299
C_3DLG/C1	14	PRG	GRAPHX	299
C_3DLG/C2	14	PRG	GRAPHX	299
C_3DLG/C3	7	PRG	GRAPHX	299
C_3DSM/1	12	PRG	GRAPHX	299
C_3DSM/B	12	PRG	GRAPHX	300
C_3DSM/L	12	PRG	GRAPHX	300
C_3DSM/R	12	PRG	GRAPHX	300
C_3DSM/T	12	PRG	GRAPHX	300
C_B/BLOCK1	14	PRG	GRAPHX	300
C_B/BLOCK2	14	PRG	GRAPHX	300
C_BLOCK1	14	PRG	GRAPHX	300
C_BLOCK2	14	PRG	GRAPHX	300
C_BLOCK3	5	PRG	GRAPHX	300
C_BOLD/B	14	PRG	GRAPHX	300
C_BOLD/L	14	PRG	GRAPHX	300
C_BOLD/R	14	PRG	GRAPHX	300
C_BOLD/T	14	PRG	GRAPHX	300
C_BROADWY1	14	PRG	GRAPHX	300
C_BROADWY2	12	PRG	GRAPHX	300
C_BROADWY2	12	PRG	GRAPHX	300
C_COMP/1	14	PRG	GRAPHX	299
C_COMP/2	4	PRG	GRAPHX	299
C_COMP/B1	14	PRG	GRAPHX	300
C_COMP/B2	4	PRG	GRAPHX	300
C_COMP/L1	14	PRG	GRAPHX	300
C_COMP/L2	4	PRG	GRAPHX	300
C_COMP/R1	14	PRG	GRAPHX	300
C_COMP/R2	4	PRG	GRAPHX	300
C_COMP/T1	14	PRG	GRAPHX	300
C_COMP/T2	4	PRG	GRAPHX	300
C_FAROUK1	14	PRG	GRAPHX	300
C_FAROUK2	12	PRG	GRAPHX	300
C_HEADLNR1	14	PRG	GRAPHX	300
C_HEADLNR2	14	PRG	GRAPHX	300
C_HEADLNR3	14	PRG	GRAPHX	300
C_LCHR/F1	14	PRG	GRAPHX	299
C_LCHR/F2	14	PRG	GRAPHX	299
C_LCHR/L1	14	PRG	GRAPHX	299
C_LCHR/L2	14	PRG	GRAPHX	299
C_OLDE/1	14	PRG	GRAPHX	299
C_OLDE/2	12	PRG	GRAPHX	299

C_OLDE/3	14	PRG	GRAPHX	299
C_OLDE/4	9	PRG	GRAPHX	299
C_OLDE/B1	14	PRG	GRAPHX	300
C_OLDE/B2	14	PRG	GRAPHX	300
C_OLDE/B3	14	PRG	GRAPHX	300
C_OLDE/B4	7	PRG	GRAPHX	300
C_OLDE/L1	14	PRG	GRAPHX	300
C_OLDE/L2	14	PRG	GRAPHX	300
C_OLDE/L3	14	PRG	GRAPHX	300
C_OLDE/L4	3	PRG	GRAPHX	300
C_OLDE/R1	14	PRG	GRAPHX	300
C_OLDE/R2	14	PRG	GRAPHX	300
C_OLDE/R3	14	PRG	GRAPHX	300
C_OLDE/T1	14	PRG	GRAPHX	300
C_OLDE/T2	14	PRG	GRAPHX	300
C_OLDE/T3	14	PRG	GRAPHX	300
C_SAMPLER1	14	PRG	GRAPHX	300
C_SAMPLER2	14	PRG	GRAPHX	300
C_SAMPLER3	14	PRG	GRAPHX	300
C_STUBBY1	14	PRG	GRAPHX	300
C_STUBBY2	12	PRG	GRAPHX	300
C_TALL/1	14	PRG	GRAPHX	299
C_TALL/2	12	PRG	GRAPHX	299
C_XMAS/1	4	PRG	GRAPHX	299
DL6A	65	DV080	GRAPHX	299
DL6B	32	DV080	GRAPHX	299
FONT_COMP	11	PRG	GRAPHX	299
FONT_GOTH	12	PRG	GRAPHX	299
G1B	54	PRG	GRAPHX	381
G2B	54	PRG	GRAPHX	381
G3B	54	PRG	GRAPHX	381
G4B	54	PRG	GRAPHX	381
LOAD	59	IV254	GRAPHX	299
P_XMAS/1	54	PRG	GRAPHX	299
P_XMAS/2	54	PRG	GRAPHX	299
P_XMASCARD	54	PRG	GRAPHX	299
READXMAS	7	DV080	GRAPHX	299

ΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩ

Sorting part 3

Courtesy of TIsHUG by Ron Brubaker, USA

A very common application of sorting is the alphabetisation of names or of words. Alphabetic sorting in BASIC is particularily easy due to the fact that all of the characters utilised by BASIC have numeric equivalents. Thus, BASIC allows alphabetic characters or even strings to be used in comparisons (e.g. "A"<"B" is a valid comparison as is A$<B$ where A$ = "ABC" and B$ = "BCDE"). Due to this feature of the language it is very easy to modify a program that sorts numbers to sort strings.

Sorting Strings

The following program segment will be used to provide a list of names in the form of a one-dimensional string array.

```
10 REM *** READ IN A LIST OF RANDOMLY ORDERED NAMES ***
20 REM
30 DIM A$(100)
40 READ N
50 FOR I=1 TO N
60 READ A$(I)
70 PRINT A$(I)
80 NEXT I
90 DATA 15
100 DATA "WASHINGTON, GEORGE","JEFFERSON, TOM","FORD,
    GERALD"
110 DATA "KENNEDY, JOHN","FILMORE, MILARD","ARTHUR,
    CHESTER"
120 DATA "ADAMS, JOHN Q","LINCOLN, ABE","ROOSEVELT,
    FRANKLIN"
130 DATA "REAGAN, RONALD","CARTER, JIMMY","WILSON,
    WOODROW"
140 DATA "MONROE,JAMES","ROOSEVELT,THEODORE","ADAMS,
    JOHN"
150 END
```

The output from this program is the following:

```
WASHINGTON. GEORGE
JEFFERSON, TOM
FORD, GERALD
KENNEDY, JOHN
FILMORE, MILARD
ARTHUR, CHESTER
ADAMS, JOHN
LINCOLN, ABE
ROOSEVELT, FRANKLIN
REAGAN, RONALD
CARTER, JIMMY
WILSON, WOODROW
MONROE, JAMES
ROOSEVELT, THEODORE
ADAMS, JOHN
```

The sorting is very similar to the program shown at the end of Part 2.

```
10 REM *** READ IN A LIST OF RANDOMLY ORDERED NAMES ***
20 REM
30 DIM A$(100)
40 READ N
50 FOR I=1 TO N
60 READ A$(I)
70 PRINT A$(I)
80 NEXT I
90 DATA 15
100 DATA "WASHINGTON, GEORGE","JEFFERSON, TOM","FORD,
    GERALD"
110 DATA "KENNEDY, JOHN","FILMORE, MILARD","ARTHUR,
    CHESTER"
120 DATA "ADAMS, JOHN Q","LINCOLN, ABE","ROOSEVELT,
    FRANKLIN"
130 DATA "REAGAN, RONALD","CARTER, JIMMY","WILSON,
    WOODROW"
140 DATA "MONROE,JAMES","ROOSEVELT,THEODORE","ADAMS,
    JOHN"
160 REM ***** IMPROVED BUBBLE SORT *****
170 REM
180 J=1
190 Q=0
200 FOR I=1 TO N-J
210 IF A$(I)<=A$(I+1)THEN 260
220 Q=1
230 T$=A$(I)
240 A$(I)=A$(I+1)
250 A$(I+1)=T$
260 NEXT I
270 IF Q=0 THEN 310
280 J=J+1
290 GOTO 190
300 REM
310 REM **** ROUTINE TO PRINT SORTED LIST OF NAMES ****
320 REM
330 PRINT
340 FOR I=1 TO N
350 PRINT A$(I)
360 NEXT I
```

The changes consist of replacing all references to the array A with the array A$ and replacing the variable T with T$. The only other change is the removal of the trailing semi-colon in the print loop so that the names

are listed on the separate lines.

The resulting list is:

```
ADAMS, JOHN
ADAMS, JOHN Q.
ARTHUR, CHESTER
CARTER, JIMMY
FILMORE, MILARD
FORD, GERALD
JEFFERSON, TOM
KENNEDY, JOHN
LINCOLN, ABE
MONROE, JAMES
REAGAN, RONALD
ROOSEVELT, FRANKLIN
ROOSEVELT, THEODORE
WASHINGTON, GEORGE
WILSON, WOODROW
```

Not a bad job for such a simple program!

Some Difficulties With String Sorts

Occasionally string sorts produce unexpected results. For example, although it is possible to sort numeric information that is in the form of strings, the results may not always be anticipated. For example, if the data in the above program is modified as follows:

```
110 DATA "2","9","7","57"
120 DATA "229","11","41","99","4"
130 DATA "89","199","50","220","1"
```

N.B. Lines 100 and 140 must be deleted and also line 70 if you want to eliminate the printing of the unsorted random numbers.

The results will be:

```
1 100 11 199 2 200 220 4 41 50 57 7 89 9 99
```

The difficulty may not be immediately apparent. However, if you note that the first digit is strictly in order and that the digits that follow are in order of the second digit, etc. the problem becomes apparent. Consider the following data statements:

```
100 DATA "  2","  9","100"," 57"
120 DATA "220"," 11"," 41"," 99","  4"
130 DATA " 89","199"," 50","200","  1"
```

The resulting output is:

```
1 2 4 7 9 11 41 50 57 89 99 100 199 200 220
```

The lesson to be learned from the above is that string representations of numerics must be right justified to ensure that they are sorted properly. Either leading spaces or zeros may be used to accomplish this. In the case of alpha strings it is normally necessary that the strings be left justified in on order to ensure proper sorting. In the case of mixed alpha and numeric strings, which are common for computer file names, precautions should be taken to pad the numeric portions of the name with leading zeros (e.g. SORT01,SORT02,SORT10).

A second type of problem can arise if one attempts to sort mixed upper and lower case alphabetic characters. (Refer user handbook for the numeric equivalents of the ASCII character set). When BASIC is used to sort ASCII coded characters it does so on the basis of these numeric equivalents. Thus all upper case letters come before their lower case equivalents. Note also that the most common punctuation marks and all numbers come before the upper case alphabet. Care must be exercised to avoid situations where this inherent order of the string characters will dictate sorting in a different order than is desired.

The Use Of Pointers For Indirect Sorting

The sorting routines shown so far have actually rearranged the data in the data array variable. Although this can be the fastest method in the case of a simple numeric sort the time required to move lengthly strings about tends to slow down a direct string sort. In addition, it may not be desirable to actually rearrange the data. Since BASIC does permit string data and numeric data to be stored in the same array and there are some extra difficulties associated with sorting string representations of numbers, it is often more convenient to sort by an indirect method using pointers.

A pointer is simply a numeric array of the same length as the array that is to be sorted. It must be initialised by assigning each element a value to the index of the element:

(i.e. P(1)=1, P(2)=2, ----> P(N)=N)

This is accomplished in line 90 of the following program. In line 190 the numeric array to be sorted is referenced indirectly using P(I) in the place of the simple index loop index I. Remember, initially P(I)=I. However, in lines 200-230 which normally swap elements that are out of order, the values in the pointer array are swapped instead.

```
10 REM **** GENERATION OF A LIST OF RANDOM NUMBERS ****
20 REM
30 DIM A(100),P(100)
40 PRINT "HOW MANY NUMBERS DO YOU WANT?"
50 INPUT N
60 PRINT
70 FOR I=1 TO N
80 A(I)=INT(100*RND)+1
90 P(I)=I
100 PRINT A(I);
110 NEXT I
120 PRINT
130 REM
140 REM ***** BUBBLE SORT WITH POINTER *****
150 REM
160 J=1
170 Q=0
180 FOR I=1 TO N-J
190 IF A(P(I))<=A(P(I+1))THEN 240
200 Q=1
210 T=P(I)
220 P(I)=P(I+1)
230 P(I+1)=T
240 NEXT I
250 IF Q=0 THEN 290
260 J=J+1
270 GOTO 170
280 REM
290 REM ***** ROUTINE TO PRINT LIST USING POINTER *****
300 REM
310 PRINT
320 PRINT "  I  A(I)  P(I)  A(P(I))"
330 IMAGE ###  ###   ###   ###
340 PRINT "============================"
350 FOR I=1 TO N
360 PRINT USING 330:I,A(I),P(I),A(P(I))
370 NEXT I
```

The output routine was also modified to show the values of the index, the numeric array, and the sorted results. Note that the letter is achieved by referencing the numeric array using the pointer array.

```
HOW MANY NUMBERS DO YOU WANT? 15

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

I	A(I)	P(I)	A(P(I))
1	64	3	2
2	48	6	5
3	2	14	13
4	79	11	14
5	36	10	24
6	5	15	34
7	66	5	36
8	71	2	48
9	100	13	57
10	24	1	64
11	14	7	66
12	67	12	67
13	57	8	71
14	13	4	79
15	34	9	100

Examine the pointer array carefully. Note that the first element has a value of 3, the smallest number in the numeric array, and the last element has a value of 9, the largest number in the numeric array. Thus, the values found in the pointer array after sorting point to the values in the original array in the order that they must be printed to obtain a sorted list. Pretty neat, huh?

Next month we will look at how this type of sorting can be applied to string arrays. o

ΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩ

SIMPLE PROGRAMS

Courtesy of LA99ers

by Jim Peterson

I like SIMPLE programs that do exactly what I need to do when I need do it, and nothing more - without the bells and whistles and fancy title screens, without going through a series of menus to get to what I want.

I have been asked, what is the best checkbook program? I have two disks full in my TI-PD library, but I don't know which one is best, because I have never used any of them. I have my own little checkbook program. It is so little that I don't even bother to save it - I just key it in whenever I need to do any adding or subtracting. It goes like this -

```
1 INPUT A :: T=T+A :: PRINT .T ::
GOTO 1
```

When I receive a bank statement, I open my checkbook, check off all the debits and credits recorded on the statement, then run that little program. I enter the bank's ending balance, then enter each outstanding check or other outstanding debit in the checkbook as a negative value (i.e., with a minus sign in front) and every deposit not yet recorded by the bank as a positive value. I should end up with my checkbook's ending balance. If I don't, I next enter my checkbook ending balance as a negative value (presuming it is not already negative!) to see how much the discrepancy is, and scan the checkbook and statement for an entry of that amount which may explain the error. If not, I hit FCTN 4, run the program again, enter my checkbook balance the last time I reconciled it, enter each succeeding debit as negative and each credit as postive, and see if I can spot an error in my calculations. If not, I give up - and I don't think that any checkbook program could help me go any farther.

Yes, I know that some of those programs will give me a printout of all my transactions, after I have keyed in a lot of data - but why do I need a hard copy of something that is already recorded in my checkbook.

And I know that some programs will give me a record of bills paid in various categories. I don't need that - at the end of the year I run a simple little program I wrote, Adder-Upper, and in 15 minutes with one pass through the checkbook I can total expenses in as many categories as I want.

I have also been asked for a good mailing list program. I have lots of those too, but I haven't tried them. My own mailing list program isn't even a program - it's a D/V80 file created with Funlweb. Just type in the name, address, etc. in 3 or 4 lines, as you want it to appear on the label; hit Enter 2 or 3 more times to make a total total of 6 lines; and start the next name, etc. when you are finished, make sure that the first blank line above the first line of the last address has a line number evenly divisible by 6; otherwise, you didn't space your address 6 lines apart somewhere. Line up your strip labels in the printer, print the file out of Funlweb editor, and that's all there is to it.

It would be a good idea to set up that file in sequence by persons' last name, so that you can quickly find them for the purpose of changing or deleting - all done with Funlweb editing commands faster than any computer program could do it. To add a name,

just FCTN 4 down to the right spot in sequence, FCTN 8 to open lines, and type it in.

You want something a little more than that? You want to selectively print or skip names? OK, you've got two blank lines after each record, so use the 6th line to code your special requirements - for instance, a C for Christmas cards, BU for business mailing, B11 for birthday cards in November - just don't use B alone for one code and in combination with something else for another code.

Then, instead of printing through Funlweb, use this program -

```
100 DISPLAY AT(12,1)ERASE AL
L:"Filename? DSK" :: ACCEPT
AT(12,14)BEEP:F$ :: OPEN #1:
"DSK"&F$,INPUT :: OPEN #2:"P
IO"
110 DISPLAY AT(14,1):"Print
addresses with code -":"":"(
to print all addresses,    j
ust press Enter)"
120 ACCEPT AT(15,1)BEEP:X$
130 LINPUT #1:A$ :: LINPUT #
1:B$ :: LINPUT #1:C$ :: LINP
UT #1:D$ :: LINPUT #1:E$ ::
LINPUT #1:F$
140 IF POS(F$,X$,1)<>0 OR X$
="" THEN PRINT #2:A$:B$:C$:D
$:"":""
150 IF EOF(1)<>1 THEN 130 EL
SE CLOSE #1
```

For a one-time selective mailing, load your file, go down through it with FCTN X, type an asterisk in the 6th line of each record you want to print, save the file to another disk, run the above program and enter the asterisk as the special code to select on.

You need a home record-keeping system? Do you really really really need to sort your home records by various fields, print them out, etc.? If so, you need a data base program. But, if you just need to file some info in a way that you can find it again, I have the perfect system for you. I use it often. Technically, it is called the File Box Full Of Ruled 3x5 Index Cards. Unlike a data base program, this system offers fields of unlimited length, records of unlimited length (if the front of the card is full, flip it over and write on the back; if that is full, stick a second card behind the first), quick manual alphabetic sequencing (probably faster than the computer could do it), quick manual updating and editing with eraser, quick deletions into the wastebasket, and optional cross referencing (if you're not sure by which of several key words you might try to find something, put in a card in alphabetic sequence for each keyword, with a note on it referring to the card containing the data).

Still think you need a computerized record keeping system? OK, how about the Funlweb Index Card Simulation. Just boot up Funlweb, use (T)ab to put an R at 39, type in whatever you want to keep track of, and save it to disk using a keyword as the filename. They will be automatically sorted into alphabetic sequence. Funlweb's SD will catalog them for you, put whatever you want on the screen, dump it to a printer if you want. You have all the great editing features of Funlweb to update, delete, etc., and you can cross reference keywords with dummy files. You might want to use DM 1000 periodically to protect your files so that you don't accidentally use the same keyword filename over again and overwrite a record.

Of course, you can only get 127 records on a disk. But a deck of 100 index cards costs me $.59 at the drug store, and generic disks cost $.25 or less! For $7 you can set up a filing system with a disk for each letter of the alphabet.

Just one more example of the power of the Funlweb filing system. Before I bought a disk system, I had already acquired or written over a thousand programs, and I had recorded each one on a 3x5 index card, filed by program name, with the author's name and comments. I promptly transferred all those programs to disk, ran the disks through one of those disk catalogers, and was greatly dissatisfied with the results. I DO NOT LIKE DISK FILENAMES! It is much too hard to remember what I was trying to abbreviate when I saved a file as BRFLTZK. Note: this was in the

days before the good catalog programs allowed you to add comments.

I didn't like listings of file names on disk labels either, and I have never used them. I don't want to paw through boxes of full of disks, trying to decipher filenames written in tiny subscript. Obviously, my index card system was better than that, but I wanted a printed catalog, in plain English, in actual program name alphabetic sequence, and with as much other information as possible.

My old Gemini would print 136 characters on a line in condensed print, so I wanted each record to be up to that long. Funlweb (or TI-Writer back then) limited me to 80 characters, but no problem; I'll just use two lines to key them in, and write a little program to combine them into one line for printer output. I set the TI-Writer tabs to give me 25 spaces for full program name, 16 spaces for author's name, 4 spaces for disk number (I number my disks consecutively as I fill them, and file them numerically), 10 spaces for coded language and system requirements, 11 spaces for a 1 to 10 star rating, and the remaining 14 spaces to begin a brief program description which could continue for 56 spaces on the next line.

My index cards were filed in sequence by actual program name, so I keyed in their contents in that sequence. As I acquired more programs, I continued to record them on index cards. Every once in a while I loaded my catalog file into Funlweb, FCTN 4'd to the proper place in alphabetic sequence, opened a line there with FCTN 8, and typed in the new record. The file soon became too large to load, but it was very easy to use Funlweb's features to split it into two files, then into many files, finally into five disks full of files. Each disk is labeled with the beginning and ending letters of each file, so they can be quickly found.

The last time I updated my catalog and printed it out (I have obtained hundreds of programs since) it contained 4041 records on 50 pages at 80 lines per page, nearly half a million bytes of data. I doubt that anyone else in the TI world has anything like it.

If I'm figuring correctly, it would take about 500k of RAM to read all that into memory. Since the PC's seem to be so incredibly wasteful of memory, I wonder if a 640k computer could update that file? If so, could they do it any faster than I can do it with Funlweb?

But, you say, you can't SORT that file! Well, why should I want to? And could that 640k computer do it? Besides, if I wanted to, I CAN DO IT!

I tried it, just for the heck of it. The only worthwhile field to sort on would be the disk number. So, I did a CALL FILES(9) and ran a little 10-line program which opened one file for input and eight for output, read a disk directory, opened each file in sequence, read the records and, if the disk number was less than 9, wrote the record to the file of the same number. When I was through swapping disks, I had separate files for disks 1 through 8, still in alphabetical sequence. It would take many hours sort out over 500 disks 8 at a time, but I could cram all those files onto two DS disks and my ramdisk, modify the program a bit, and all I would have to do is change backup disks when they were full. Or to be more sensible, I could extract records in batches of 10 or more, whatever would be small enough to cram into TI-Sort for a further sort.

The point is - some jobs can be done better without a computer, and many others can be done with a very simple little program!

"How do you like the new super-minicomputers?"

SHOP

All the software listed below is in stock or will be very shortly. Prices are in Australian dollars and include postage. The recent drop in the value of the Aussie dollar has resulted in a small increase in prices.

Rock Runner	$14.00
Waterworks	$14.00
Beyond Video Chess	$11.50
Rattlesnake Bend	$ 7.75
Castle Darkholm	$ 9.75
Doom Games I	$ 7.75
Doom Games II	$ 7.75
Doom Games III	$ 7.75
Page Pro	$27.00
Page Pro Pics #1	$ 8.50
Page Pro Pics #2	$ 8.50
Page Pro Pics #3	$ 8.50
Page Pro Pics #4	$ 8.50
Page Pro Pics #5	$ 8.50
Page Pro Pics #6	$ 8.50
Page Pro Pics #10	$ 8.50
Page Pro Pics #14	$ 8.50
Page Pro Pics #15	$ 8.50
Page Pro Borders #1	$ 8.50
Page Pro Borders #2	$ 8.50
Page Pro Fonts #1	$ 8.50
Page Pro Fonts #2	$ 8.50
Page Pro FX	$16.50
Page Pro Headline Maker	$11.50
Page Pro Headline Fonts #1	$ 8.50
Page Pro Headline Fonts #2	$ 8.50
Page Pro Hradline Fonts #3	$ 8.50
Page Pro Templates #5	$ 7.70
Page Pro Templates #6	$ 7.50
Page Pro Templates #7	$ 7.50
Pix Pro	$16.00
Spell It	$21.00
Quick Run	$11 00
Tournament Solitaire	$16.50

TI BRISBANE USER GROUP
P.O. BOX 3051
CLONTARF M.D.C.
QLD AUST 4019.

43
GRAEME MORRIS
P.O. BOX 371
CLEVELAND QLD 4163