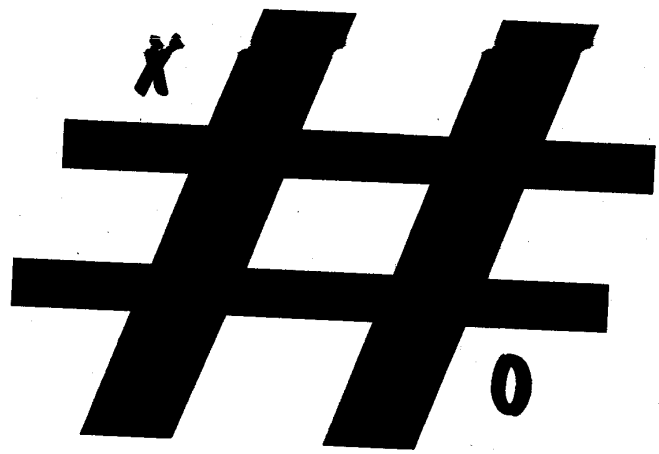
Covering the TI99/4A and the Myarc 9640

MICROpendium

Volume 12 Number 1 February 1995 $3.50



Tic-tac-toe, Philadelpha-style

See Page 13

PC-99 Stage 3 released

Also:

An assembly program by popular demand
Find fellow TIers in cyberspace
A hard drive untility for the Geneve
Identifying 'mystery files'
An XBASIC program that writes sentences

And much more as we begin our 12th continuous year

CONTENTS

MICROpendium

MICROpendium (ISSN 10432299) is published monthly for $35 per year by Burns-Koloen Communications Inc., 502 Windsor Rd., Round Rock, TX 78664-7639. Second-class postage paid at Round Rock, Texas. POSTMASTER: Send address changes to MICROpendium, P.O. Box 1343, Round Rock, TX 78680-1343.

No information published in the pages of MICROpendium may be used without permission of the publisher, Burns-Koloen Communications Inc. Only computer user groups that have exchange agreements with MICROpendium may excerpt articles appearing in MICROpendium without prior approval.

While all efforts are directed at providing factual and true information in published articles, the publisher cannot accept responsibility for errors that appear in advertising or text appearing in MICROpendium. The inclusion of brand names in text does not constitute an endorsement of any product by the publisher. Statements published by MICROpendium which reflect erroneously on individuals, products or companies will be corrected upon contacting the publisher.

Unless the author specifies, letters will be treated as unconditionally assigned for publication, copyright purposes and use in any other publication or brochure and are subject to MICROpendium's unrestricted right to edit and comment.

Display advertising deadlines and rates are available upon request.

All correspondence should be mailed to MICROpendium at P.O. Box 1343, Round Rock, TX 78680. We cannot take responsibility for unsolicited manuscripts but will give consideration to anything sent to the above address. Manuscripts will be returned only if a self-addressed stamped envelope is included.

Foreign subscriptions are $40.25 (Mexico); $42.50 (Canada); $40.00, surface mail to other countries; $52 airmail to other countries.

All editions of MICROpendium are mailed from the Round Rock (Texas) Post Office.

Mailing address: P.O. Box 1343, Round Rock, TX 78680.

Telephone & FAX: (512) 255-1512

Delphi TI NET: MICROpendium

GEnie: J.Koloen

Internet E-mail: jkoloen@io.com

John Koloen.....Publisher

Laura Burns.....Editor

PC-99

CaDD releases Stage 3 .....Page 5

The Art of Assembly

CaDD releases Stage 3 .....Page 6

Myarc Advanced BASIC

Hard drive utility finds, records formatting information .....Page 9

Identifying Files

Finding out what those "mystery programs" are .....Page 10

BASIC

Tic-tac-toe, Philadelphia-style .....Page 13

An E-mail Directory

TI users in cyberspace .....Page 17

Freeloader III

Not your average load program.....Page 19

Extended BASIC

A program that writes sentences.....Page 23

Reviews

MICRO-Reviews: Ant Eaters, Hen Pecked, Rotor Raiders, Princess and the Frog.....Page 25

User Notes

Hide and seek in XBASIC, an aptitude for shapes, and using Telco to upload ASCII files.....Page 28

*READ THIS

Here are some tips to help you when entering programs from MICROpendium:

1. Most BASIC and Extended BASIC programs are run through Checksum, which places the numbers that follow exclamation points at the end of each program line. Do not enter these numbers or exclamation points. Checksum is available on disk from MICROpendium for $4.

2. Long Extended BASIC lines are entered by inputting until the screen stops accepting characters, pressing Enter, pressing FCTN REDO, cursoring to the end of the line and continuing input.

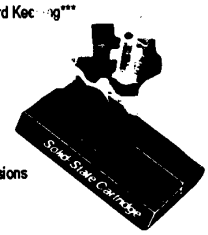
TEX★COMP

Modules 99¢ each—NO LIMIT!

Subject to availability. Effective 3-1-94

PHM	3027	Addition & Subtraction I	PHM	3016	Tax Investment Record Keeping***
PHM	3028	Addition & Subtraction II	PHM	3022	Personal Real Estate
PHM	3051	Numeration II	PHM	3031	The Attack
PHM	3094	Milliken Integers	PHM	3057	Munchman
PHM	3025	Mind Challenger	PHM	3219	Super Demon Attack
PHM	3021	Weight & Nutrition	PHM	3006	Home Financial Decisions
PHM	3002	Early Learning Fun	PHM	3035	Terminal Emulator II
PHM	3003	Beginning Grammar	PHM	3052	Tombstone City
PHM	3112	Parsec	PHM	3109	Logo II*
PHM	3007	Household Budget Management	PHM	3055	Editor Assembler
PHM	3044	Pers. Report Generator**	PHM	3010	Physical Fitness
PHM	3053	TI Invaders	PHM	3194	Jawbreakers II

Plus $4 additional shipping & handling



ALSO INCLUDED ARE THE FOLLOWING ARARISOFT MODULES:

DEFENDER, PROTECTOR II, PICNIC PARANOIA

*ADD AN ADDITIONAL $4.00 EACH ON THESE TITLES DUE TO WEIGHT
**FOR USE WITH PERSONAL RECORD KEEPING ***DISK DRIVE REQUIRED
PERSONAL RECORD KEEPING AVAILABLE AT $4.95

Special Purchase

Personal Record Keeping
PHM 3013
$4.95 (Regular $9.95)

99¢ BOOK BLOWOUT!!!

NO LIMIT!

PUBLISHER'S CLEARANCE ON BOOKS FOR THE 99/4A

GAMES TIs PLAY
THE ELEMENTARY TI
COMPUTER PLAYGROUND (TI)
INTRO TO ASSEMBLY LANGUAGE
PROGRAMS FOR THE TI COMPUTER

TI-99/4A GAME PROGRAMS
STARTER PACK I**
STARTER PACK II**
GAME WRITERS PACK I**
GAME WRITERS PACK II**

*Add $3 for Games TIs Play disk or cassette
** Add $3 for disk or cassette for disk or cassette for Starter Pack Game Writer Series.

→ **SPECIAL**

Kids and the TI — $2.95

The TI in Bits and Bytes with disk — $4.95

SPECIAL PACKAGE AND PRICING ON *Adventure*

THE ADVENTURE SERIES...

SPECIAL PACKAGE AND PRICING ON
THE ADVENTURE SERIES... ON DISK OR CASSETTE (specify)

THE COMPLETE SCOTT ADAMS SERIES I (ADVENTURELAND, PIRATE ISLAND, THE COUNT MISSION IMPOSSIBLE, VOODOO CASTLE, STRANGE ODYSSEY, MYSTERY FUN HOUSE, PYRAMID OF DOOM, GHOST TOWN, SAVAGE ISLAND I, SAVAGE ISLAND II, AND GOLDEN VOYAGE PLUS KNIGHT IRONHEART BONUS ADVENTURE

NEW LOWER PRICE ONLY $4.95 FOR ALL 13

THE SCOTT ADAMS ADVENTURE SERIES II (THE HULK, SPIDERMAN, BUCKAROO BANZAI, AND THE SORCERER OF CLAYBORGUE CASTLE PLUS 2 BONUS ADVENTURES

NEW LOWER PRICE ONLY $4.95 FOR ALL6

COMMAND MODULE FOR ABOVE ADVENTURES (reg $8.98) **ONLY 95 CENTS** WITH EITHER OF THE ABOVE SERIES

HINT BOOK FOR ALL OF THE ABOVE SCOTT ADAMS ADVENTURES ONLY **95 CENTS** WITH EITHER OF THE ABOVE SERIES.

NEW First time on disk — Return to Pirate's Island with Graphics. XBASIC or Editor/Assembler Load

$4.95 Includes disk version of Editor/Assembler

ADVENTURE EDITOR

NOW WRITE AND EDIT YOUR OWN GAMES FOR THE TI ADVENTURE MODULE!

TEX-COMP is proud to announce the 99/4A Adventure Editor — an all new program which allows you to write your own adventure programs — Scott Adams move over! This disk or cassette based program allows

1. Any existing adventure for the Adventure command module to be edited, altered listed and copied from any storage medium to any other storage medium i.e. Tape to disk or vice versa!
2. New adventure games can be created freely using a template, game as a start up step. These games can be conceived using the full power and capabilities of the TI Adventure command module.

The program is written for Mini Memory or Editor Assembler command modules (cassette or disk) and offers a full screen editor and a special easy to use mnemonic language called A.P.L. (Adventure Program ming Language) and comes with a complete and detailed 75 page manual.

TEX-COMP is also organizing a new TI Adventuring Users Group, which will allow low cost exchange of user written adventures between members

Available NOW From TEX-COMP **NEW LOW PRICE**

Specify Mini Memory (Cassette) or Editor Assembler (Disk) versions

SUPER BUY!

$4.95

HOME ENTERTAINMENT

DISKETTE PROGRAMS

PHD 5002	TI-TREK (TE-II req. for speech)	4.95
PHD 5010	Mystery Melody	4.95
PHD 5015	Oldies But Goodies I	4.95
PHD 5017	Oldies But Goodies II	4.95
***SPECIAL	Oldies But Goodies I & II	7.95
PHD 5025	Sat. Night Bingo (Ex-Basic & Speech)	4.95
PHD 5037	Draw Poker (Ex-Basic)	4.95

CASSETTE PROGRAMS

PHT 6002	TI-TREK (TE11 req. for speech)	4.95
PHT 6010	Mystery Melody	4.95
PHT 6015	Oldies But Goodies I	4.95
PHT 6017	Oldies But Goodies II	4.95
***SPECIAL	Oldies But Goodies I & II	7.95
PHT 6026	Sat. Night Bingo (Ex-Basic & Speech)	4.95
PHT 6037	Draw Poker (Ex-Basic)	4.95

COMPUTER PROGRAMMING AIDS

DISKETTE PROGRAMS

PHD 5007	Teach Yourself 99/4A Basic	4.95
PHD 5019	Teach Yourself Extended Basic	4.95
PHD 5004	Programming Aids I	4.95
PHD 5005	Programming Aids II	4.95
PHD 5077	Programming Aids I,II,III	9.95
PHD 5067	Beginning Basic Tutor	4.95
PHD 5076	Text to Speech (Ex-Basic Speech)	4.95
PHD 5098	TI Forth & manual (Ed/Assem req.)	19.95
PHD 5078	TI Forth Demo Disk (Ed/Assem)	4.95
PHD 5079	TI Forth Source Code (2 disks)	4.95

CASSETTE PROGRAMS

PHT 6006	Programming Aids I	4.95
PHT 6007	Teach Yourself 99/4A Basic	4.95
PHT 6019	Teach Yourself Extended Basic	4.95
PHT 6067	Beginning Basic Tutor	4.95

MATH AND ENGINEERING

specify disk or tape with order

Math Routine Library	4.95
Electrical Engineering Library	4.95
Graphing Package	4.95
Structural Engineering Library	4.95
AC Circuit Analysis	4.95
***SPECIAL--ALL 5 OF THE ABOVE ON DISK OR TAPE---	17.95

$4.95 Per Disk

SPECIAL Buy one get one free Any disk or cassette below

EDUCATION

DISKETTE PROGRAMS

PHD 5009	Music Skills Trainer	4.95
PHD 5011	Computer Music Box	4.95
PHD 5018	Market Simulation	4.95
PHD 5030	Speak & Spell II (Ex Basic req.)	9.95
PHD 5031	Speak & Math (TE-II req.)	4.95
PHD 5042	Spell Writer (TE-II req.)	4.95
PHD 5026	Bridge Bidding I	4.95
PHD 5039	Bridge Bidding II	4.95
PHD 5041	Bridge Bidding III	4.95
SPECIAL!!	BRIDGE BIDDING I,II&III	9.95
PHD 5070	Music Maker Demo (use with module)	4.95
PHD 6010	Mystery Melody (stop the music quiz)	4.95

CASSETTE PROGRAMS*
*see disk versions for requirements i.e. TE-II

PHT 6009	Music Skills Trainer	4.95
PHT 6010	Mystery Melody	4.95
PHT 6011	Computer Music Box	4.95
PHT 6018	Market Simulation	4.95
PHT 6031	Speak & Math	4.95
PHT 6042	Spell Writer	4.95
PHT 6026	Bridge Bidding I	4.95
PHT 6039	Bridge Bidding II	4.95
PHT 6041	Bridge Bidding III	4.95
SPECIAL!!	BRIDGE BIDDING I,II&III	9.95
PHT 6070	Music Maker Demo (use with module)	4.95

MANAGEMENT AND BUSINESS

DISKETTE PROGRAMS

PHD 5001	Mailing List (upgraded version)	4.95
PHD 5003	Personal Financial Aids	4.95
PHD 5021	Checkbook Manager	4.95
PHD 5022	Finance Manager	4.95
PHD 5024	Inventory Management	4.95
PHD 5027	Invoice Management	4.95
PHD 5029	Cash Management	4.95
PHD 5038	Lease/Purchase Decisions	4.95
PHD 5075	TI/Multiplan upgrade disk	4.95

CASSETTE PROGRAMS

PHT 6003	Personal Financial Aids	4.95
PHT 6038	Lease/Purchase Decisions	4.95

TERMS: Add $3 S&H to all orders. California orders add 8.25% sales tax. Canada, Hawaii and overseas shipping extra.

TEX★COMP
P.O. Box 33084, Granada Hills, CA 91344

VISA and MASTERCARD HOLDERS CALL DIRECT:
(818) 366-6631

ORDER BY PHONE 24 HOURS A DAY 7 Days a Week!
(818) 366-6631

AUTHORIZED DEALER

Send $2 for our catalog

COMMENTS

PC99 Stage 3 an achievement

CaDD Electronics hasn't produced many products, but those that it has have been ambitious and unique. The company's PC99 TI99/4A emulator, which runs on a PC, is a case in point. Those who have been watching the development of this product over the years may have wondered whether it would ever be "finished." Of course, has any software ever been finished? Judging from an announcement in this month's issue, Mike Wright and company are very close.

Quoting from the article: "The emulation includes the TMS9900 processor; the TMS9918A Video Display Processor (VDP); the TMS9901 Programmable Systems Interface; and the TMS9919 sound chip in the 99/4A console. In addition the emulation includes the TI RS-232 card and its TMS9902 Asynchronous Communications Controller; the TI Disk Controller and its WD-1771 FDC chip; and the TI p-Code card."

I'm not sure what else one would want from an emulator. This is an achievement of a high order, not only because of the utility of this software but that it was done by a handful of programmers working largely in their spare time. My hat is off to Mike and Mark Van Coppenolle for seeing through to the end a truly amazing programming job.

We've also seen a notice, by the way, that V.6.0 of the Edward Schwartz emulator is coming in March.

SCSI IS ON THE WAY

A reader has taken me to task about the SCSI card. As you may recall, I've asked in several recent columns for readers to report on their experiences with the card. The Florida reader says he is more than satisfied with its operation. Well, the only reason I asked for information from readers is that I did not have a card myself. I will now be getting one. Bud says that he's got one with my name on it and I should have it in the near future. I'm looking forward to getting it and will report on it directly.

EXAMINE YOUR MAILING LABELS

We had something of a problem with our mailing labels in February. It seems the hard disk they were on crashed and the backups we'd been making were somehow corrupted. Not to panic. We've recovered most of the label information intact. That which was corrupted, we restored from renewal cards sent in by subscribers. However, we do ask that you compare the mailing label on this issue of MICROpendium with labels on issues from January or December. If there is a discrepancy in the expiration date, please let us know by mail, phone call or Delphi, GEnie or Internet. It's not likely that there will be a problem with the address information, but if there is let us know about that too.

—JK

FEEDBACK

Publicize those user groups

Lately I have heard a great deal of talk about the survival of TI user groups and what to do to keep TI user groups going.

One way that I have found to keep our group going is finding new members. You would be surprised at how many people are out there who know nothing about the existence of TI user groups; thus, advertising is very important.

Many sources of free advertisement exist such as the local newspaper which often will list (free of charge) local meetings in the paper. Your group can be listed in the Computer Shopper User Group listings by entering information about your group into their BBS. The Computer Shopper BBS number is (913) 478-3088. We had someone join our group (Mid-South 9934 User Group) from St. Petersburg, Russia, who saw the listing four our group in Computer Shopper.

An organization called the User Group Connection maintains a list of user groups on which your group can be listed for free. Contact UGC by writing to the User Group Connection, P.O. Box 67249, Scotts Valley, CA 95067-7249 or call (408) 461-5700.

Another good source for advertising is in the local PCuser group newsletter, as our local PC user group maintains a contact list in their newsletter of other user groups in the area. I've received quite a few calls from people who have seen this listing.

If you do not advertise for the purpose of obtaining more members, do so for the reason of receiving free equipment. I have had a great deal of equipment donated to our group by people who have seen our group listed. This equipment we have in turn sold, given away or set up for local schools to use.

So get the word out, you never know who is out there! Advertising is everything!

Gary Cox
President, Mid-South User Group
P.O. Box 38522
Germantown, TN 38183

Error noted

There is an error in the object code file published in the November 1994 My-BASIC column using BREAD XOP.

This portion of the first line:
0009EDDIRED
should be changed to:
0009EDISMRD

Note: This month's column also uses this object code file.

Jim Uzzell
Key West, Florida

Send your letters and comments to *MICROpendium Feedback*, P.O. Box 1343, Round Rock, TX 78680.

CaDD releases Stage 3 of PC99

Program emulates full TI system on a PC

CaDD Electronics has released PC99 Stage 3, according to Mike Wright of the company. PC99 is a program that runs in protected mode under DOS on an IBM or compatible 386 (or higher) PC and emulates the Texas Instruments TI99/4A Home Computer. The emulation includes the TMS9900 processor; the TMS9918A Video Display Processor (VDP); the TMS9901 Programmable Systems Interface; and the TMS9919 sound chip in the 99/4A console. In addition the emulation includes the TI RS-232 card and its TMS9902 Asynchronous Communications Controller; the TI Disk Controller and its WD-1771 FDC chip; and the TI p-Code card.

PC99 Stage 3 is a full upgrade that contains all features available in previous stages, and now includes, according to Wright:

• Sound Blaster support: PC99 can play TI sound channels 1-3 and the TI noise channel on a Creative Labs Sound Blaster, or compatible.

• p-Code: PC99 supports the TI p-Code card. You can run UCSD P-system programs such as the Freeform spreadsheet and Pilot interpreter.

• Multiple GROM banks: PC99 supports up to 16 GROM modules using the Review Module Library feature built into the 99/4A console.

• Mini-Screen debugger: A 256x192 TI screen is displayed in the upper left corner of a VGA screen (640x480). The rest of the screen is filled with editable objects that give you total control of the 4A environment. It is possible to interactively change memory locations and immediately see their effect on the executing TI application. The Mini-Screen debugger replaces the text mode debugger.

According to Wright, improved and new utilities include:

• DSKDIR.EXE, which can catalog p-Code disks from DOS. New switches allow for displaying Plato directories, as well as only filename and disk name so that the user can build a library file.

• DSKNAME.EXE, which allows the user to copy and rename a TI "disk" from DOS.

• DSKOUTF.EXE, which extracts Forth screens from a TI "disk" to DOS.

• DSKOUTP.EXE, which extracts p-System files from a TI "disk" to DOS.

Two products are available:

• PC99 — the full product: Includes PC99.EXE (standard version), PC99A.EXE (accelerated version), and a range of utility programs.

° PC99L — the "light" product: Includes PC99L.EXE (light version), and a range of utility programs.

According to the company, both PC99 products include files representing: the TI99/4A console ROMs and GROMs; TI Disk Controller ROM; TI RS-232 ROM; p-Code ROMs and GROMs; and Extended BASIC, Editor/Assembler, and Tombstone City modules. CaDD is licensed by Texas Instruments to distribute this copyrighted material.

The full product configuration is the equivalent of a TI-99/4A with Peripheral Expansion Box containing 32K Memory Expansion; TI Disk Controller; TI RS-232 card; and TI p-Code card; three DSSD disk drives; together with TI Extended BASIC, TI Editor/Assembler and Tombstone City modules, Wright says.

The full product configuration is the equivalent of a TI-99/4A with Peripheral Expansion Box containing 32K Memory Expansion; TI Disk Controller; TI RS-232 card; and TI p-Code card; three DSSD disk drives; together with TI Extended BASIC, TI Editor/Assembler and Tombstone City modules, Wright says.

For an existing user upgrade from PC99 Stage 2A to PC99 Stage 3 full product, cost is $7 if CaDD supplies new disks ,$5 if the user returns the original disks. For a new purchaser, the full product is $95, the light product, $47.

For further information contact CaDD Electronics, 45 Centerville Dr., Salem, NH 03079, (603) 893-1450, 603-895-0119.

Product comparison:

	PC99 Full		PC99 Light
	PC99.EXE	PC99A.EXE	PC99L.EXE
Sound Blaster support	Yes	Yes	Yes
p-System	Yes	Yes	No
16 GROM banks	Yes	Yes	No
Mini-screen debugger	Yes	No	No

PC99A.EXE (or PC99L.EXE) Performance:

CPU	99/4A	486/33	486/50	486/100	Pentium 90
Norton SI	N/A	72	108	180	286
MG Clock	30.00	30.84	21.04	12.57	8.11 secs
1-10000	34.4	39.50	26.70	15.20	10.7 secs
1-500 PRINT	26.3	35.80	23.10	13.49	8.64 secs

Comparisons furnished by CaDD Electronics.

THE ART OF ASSEMBLY — PART 43

By popular request

By BRUCE HARRISON

Many moons ago, we got a request from one reader for a little help getting through such things as the prompts given by the TI Assembler. There was only one letter, from one reader, but by our standards that constitutes "popular request." This took some experimenting, since we never use that assembler. We got out the necessary disks and refreshed our tired old memory. Here's how the prompts go, and how to answer them:

LOAD ASSEMBLER? (Y/N) —This means "are you ready to load the files ASSM1 and ASSM2 from the disk in drive 1?" The E/A module expects to find those files on drive 1, and since you might not have a disk with those files in place, this prompt gives you a chance to put one there before suffering the error report that will happen if E/A can't find those files on that drive. On our own system, we don't generally use the module, but have E/A on our P-Gram, and we've modified it to look on drive 5 of our RAMdisk for any of its standard files.

SOURCE FILE NAME? — Fill in the complete name, as in DSK1.MYPROG/S, then press ENTER. The Assembler will check for the existence of this file, lighting the appropriate drive briefly.

OBJECT FILE NAME? — Fill in the name for the output file, as in DSK2.MYPROG/O, then press ENTER. This, too, will briefly light up a disk drive, thus assuring a disk is available.

LIST FILE NAME? — Most of the time, you'll want to leave this blank. On those occasions when you need a listing, you'll most likely use the PIO port in answer to this prompt, so a listing will be printed.

OPTIONS? — This causes probably the most grief. There are four possible options, each represented by one uppercase letter, and they may all be entered, in any order. R means that the Assembler will expect registers to include the R in their names (e.g., LI R0,35). In our own work, we always use this option. L in the options means a listing is to be produced on the device named as LIST FILE NAME above. A listing can be produced only if both the name and the L are present. S means that the listing will include a symbol table at its end. This can be very useful, as it will list all labels used or referenced in the source file, with their addresses. Finally, there's the C option, which will make the object file in Compressed format. That saves disk space, but we never use it. First, this makes a non-readable object file, and second, compressed files can't be loaded except by the E/A or Mini-Memory loaders.

In our own work, we never use the TI Assembler anyway, mainly because its error reporting is lousy. We prefer using the RAGASM by Art Green, and keep that ever at hand on our RAMdisk. Here are the prompts for Art Green's Assembler:

MACROS: — We always leave this blank, since we don't use MACROs.

SOURCE: — Same as for the first prompt on TI's Assembler.

OBJECT: — Same as for the second prompt on TI's Assembler.

ID/DATE: — Can be used to date the object file for identification. We never use this, but just leave the entry field blank.

OPTIONS: — Accepts the same four letters as the TI version, and they mean the same thing. The List file name normally defaults to PIO. This can be changed by an INSTALL procedure. In that same procedure, one may also designate options to be used by default. For example, in our own use, we've set up RAGASM so that it defaults to R option, thus saving us the trouble of typing that in for each assemble. If, however, we do put in any options, then we must include R, since any entry in this field voids the default entries.

MORE: — When RAGASM finishes an Assemble, it will prompt with MORE. Any answer other than Y <ENTER> will exit the program. Y <ENTER> will take you back to the MACROS: prompt for another assembly operation.

CALLs REVISITED

Okay, so your author is just plain stubborn about some things. Some time back, we reported making attempts to use CALLs directly from assembly code. We'd tried to get some understanding of the process, particularly for CALL FILES, as can be used from BASIC and Extended BASIC. We'd learned where the code was located, at >5D5A in the TI disk controller's ROM. That information did not, however, help us with the problem of passing along the one parameter that this call needs. Being so stubborn, we just couldn't give up this idea, and so resorted to the brute force method of disassembling the entire ROM program, then tracing painfully through its contents. This is difficult to do, as the code contains many BLs and BLWPs, so one must plod through a mountain of code to determine what's really happening. One must also make some educated guesses about what the registers will contain at various places, since the code has lots of things like this: MOV @>0070(R9),@>0034(R9). Our guess (which proved correct) was that R9 must contain >8300 at this point, so that the resulting addresses in the example above would be >8370 and >8334.

After a lot of slogging through, including a step-by-step run with Miller's Explorer, we found that the answer to the riddle was >832C. That is, when the call process starts, that word in RAM Pad had to point to the location in VDP RAM where the length of the name FILES was located. The actual number parameter had to be encoded in tokenized form as >B7,>C8,>01,>3x,>B6. (The x is a number from 1 through 9.) This had to be written to VDP immediately following the PAB.

THE SOLUTION

Today's sidebar is a complete program, written to allow us to pick a number between 1 and 9, and have CALL FILES execute to make room for that number of files in VDP RAM. By itself, this

(See Page 7)

THE ART OF ASSEMBLY —

(Continued from Page 6)

isn't really useful except to prove a point. It proves only that we can CALL FILES from assembly, and have that execute correctly. One could, however, take pieces of today's sidebar, incorporate them into his own program, and thus have the number of files set up correctly for that program's needs. The three essential elements are the inclusion of REF DSRLNK, the data starting with PABDT and ending with ANYKEY, and the code section between label MKCALL and the entry DATA >A, a few lines below. The labels CALPNT and PABPNT are defined as >832C and >8356, respectively.

You'll note that this PAB data does not need to contain the usual 10 bytes that are required in most PABs. Just the length of the name, followed by the name FILES, then the data for the number in tokenized form. >B7 is the token for "(";C8 means that what follows is an unquoted string; >01 is the length of this string; and >3x is the number of files desired, where x is any number from 1 through 9. Finally, the token >B6 stands for ")" (the signal for the end of the parameter string.

As shown in the sidebar, this little program includes a little "window dressing," in the form of prompts, a routine to display those, and a routine to display a number in decimal form. It starts with a prompt at the beginning for the number of files you want capable of being open simultaneously. This can be any number from 1 through 9. Just a single keypress answers this prompt, and any key outside the range 1 through 9 will be rejected. The selected number will appear on the screen below this prompt, then a couple more numbers will appear. The numbers displayed are taken from location >8370. The first number displayed shows the address for the highest available byte in VDP RAM after the space for file processing has been set aside. On a "cold" start, this number will be >37D7, or 14295 in decimal. That indicates that the default number (3) has been set for FILES. The number shown below that is the number after the CALL FILES has been performed. If the number of files selected is above 3, then this number will be less than 14295. If the number selected was less than 3, then this will show a number greater than 14,295.

When it's done, there will be a "press any key" prompt at the bottom of the screen. Pressing any key will get you out of the program. You can then re-run this little exercise by selecting 4 from the E/A menu, then pressing ENTER at the Program Name prompt. If you do this, you can see by the numbers shown that the previous run has indeed left the correct number at address >8370. The limits on that number are 11,187 for 9 files through 15,331 for 1 file. Each file accounts for 518 bytes of space in VDP RAM.

As we said, you can excerpt the needed stuff from today's sidebar to allow a program of your own to CALL FILES. Just put the number of files you want between the single quotes where our sidebar shows '9' in the tokenized data following PABDT. For example, if the desired number is five, you'd change that tokenized line to look like this:

```
BYTE >B7,>C8,>01,'5',>B6
```

Putting the 5 between single quotes like this will cause the Assembler to insert >35 at that place in memory, and that's the desired value for the CALL FILES routine. For your own application, of course, that number can be anything from 1 through 9, just as in our little demo program. You must, however, take some care in the rest of your program not to write into the area of VDP reserved for the five files. CALL FILES will leave the highest available address at >8370, so you can compare to that word within your program to set a limit on your own "writes" to VDP. In many of our own programs, we limit our VDP writes to >37D7 and below, which is the "normal" number found in >8370. If we've used CALL FILES, we'll have to be more careful about this.

Finally, you should know that this little trick only works in limited cases. It won't work correctly for CALL SCREEN, for example. We don't know exactly why, but take our word for it, because we've tried that and got an endless repeating INCORRECT STATEMENT for our trouble. That's not too surprising, since one can't expect a call from BASIC to work like one from the disk controller's ROM. This trick has been tested with only the TI disk controller, so we've no idea whether it will work correctly with Cor-Comp or Myarc disk controllers. Some brave souls among our readers may want to try this with those "third-party" disk controllers, but we guarantee nothing in those cases. (Attempt at your own risk.) Should you need our help in applying this little trick, please write or phone, as we're always glad to help. That address and phone number are: Bruce Harrison, 5705 40th Place, Hyattsville, MD 20781, U.S.A. Phone (301) 277-3467

SIDEBAR 44

```
0001  * SIDEBAR 44
0002  * A COMPLETE PROGRAM THAT SOLVES
0003  * THE OLD CALL FILES PROBLEM
0004  *
0005  * CALLS FILES (1-9) FROM E/A OPT 3
0006  * 17 APR 94
0007  * by Bruce Harrison
0008  * Public Domain
0009  *
0010        REF  VMBW,KSCAN,VMBR,VSBW,DSRLNK  REF'D
UTILITIES
0011        DEF  START        DEFINE ENTRY POINT
0012  *
0013  * REQUIRED EQUATES
0014  *
0015  PAB    EQU  >1000        PERIPHERAL ACCESS BLOCK
VDP ADDRESS
0016  CALPNT EQU  >832C        POINTER FOR CALL FILES
0017  PABPNT EQU  >8356        POINTER FOR DSRLNK
0018  GPLWS  EQU  >83E0        GPL WORKSPACE
0019  *
0020  * MAIN CODE SECTION
0021  *
0022  START  LWPI WS           LOAD OUR WORKSPACE
0023         CLR  @>8374       CLEAR KEY-UNIT
0024         LI   R0,34        POINT AT ROW 2
0025         LI   R1,PROMPT    PROMPT FOR USER CHOICE -
```

(See Page 8)

SIDEBAR44—

(Continued from Page 7)

```
NUMBER OF FILES
0026         BL   @DISSTR      DISPLAY THE PROMPT
0027  GTNF   BLWP @KSCAN       SCAN THE KEYBOARD
0028         LIMI 2            ALLOW INTERRUPTS
0029         LIMI 0            THEN STOP
0030         CB   @>837C,@ANYKEY HAS A KEY BEEN PRESSED
0031         JNE  GTNF         IF NOT, RE-SCAN
0032         MOV  @>8374,R8    MOVE KEY AS WORD INTO R8
0033         CI   R8,'1'       COMPARE TO "1"
0034         JLT  GTNF         IF LESS, REPEAT SCAN
0035         CI   R8,'9'       COMPARE TO "9"
0036         JGT  GTNF         IF GREATER, REPEAT SCAN
0037         LI   R0,3*32+14   ROW 4, COL 15
0038         MOVB @>8375,R1    GET THE KEY'S VALUE INTO
R1
0039         BLWP @VSBW        WRITE THE CHOICE TO SCREEN
0040         MOVB R1,@PABDT+9  AND PLACE IT IN DATA FOR
THE CALL
0041         LI   R0,6*32+6    ROW 7, COL 7
0042         LI   R1,ONSTR     "OLD NUMBER"
0043         BL   @DISSTR      DISPLAY THAT
0044         LI   R0,8*32+12   ROW 9,COL 13
0045         BL   @NUMDIS      SHOW NUMBER FROM >8370
0046  MKCALL LI   R0,PAB       POINT AT PAB LOCATION IN
VDP
0047         LI   R1,PABDT     AND DATA FOR PAB
0048         LI   R2,ANYKEY-PABDT INCLUDE THE TOKENIZED
STUFF
0049         BLWP @VMBW        WRITE THAT TO VDP
0050         MOV  R0,@PABPNT   MOVE R0 TO >8356
0051         MOV  R0,@CALPNT   AND TO >832C
0052         BLWP @DSRLNK      USE DSR LINK
0053         DATA >A           WITH DATA FOR A "CALL" SER-
VICE
0054         MOV  @>8350,@>835E  CHECK FOR ERROR
0055         JEQ  NEWNUM       IF NONE, JUMP
0056         BL   @ERRDIS      ELSE DISPLAY ERROR CODE
0057  NEWNUM LI   R0,13*32+6   ROW 14, COL 7
0058         LI   R1,NNSTR     "NEW NUMBER"
0059         BL   @DISSTR      DISPLAY THAT
0060         LI   R0,15*32+12  ROW 16, COL 13
0061         BL   @NUMDIS      DISPLAY NUMBER FROM >8370
0062         LI   R0,22*32+5   ROW 23, COL 6
0063         LI   R1,PAK       "PRESS ANY KEY"
0064         BL   @DISSTR      DISPLAY THAT
0065  SCAN   BLWP @KSCAN       SCAN KEYBOARD
0066         CB   @ANYKEY,@>837C  KEY STRUCK?
0067         JNE  SCAN         IF NONE, REPEAT
0068         LWPI GPLWS        LOAD GPL WORKSPACE
0069         B    @>6A         BRANCH TO GPL INTERPRETER
0070  *
0071  * SUBROUTINE SECTION
0072  *
0073  DISSTR MOVB *R1+,R2      GET LENGTH BYTE
0074         JEQ  DISX         IF ZERO, EXIT
0075         SRL  R2,8         RIGHT JUSTIFY
0076         BLWP @VMBW        WRITE TO SCREEN
0077  DISX   RT                RETURN
0078  ERRDIS LI   R0,12*32+12  ROW 13, COL 13
0079         JMP  NDIS1        THEN JUMP
0080  NUMDIS MOV  @>8370,@>835E  GET WORD FROM >8370 TO
>835E
0081  NDIS1  CLR  @>837C       CLEAR GPL STATUS
0082         BLWP @GPLLNK      USE GPL LINKAGE
0083         DATA >2F7C        DATA FOR INTEGER TO STRING
CONVERSION
0084         MOVB @>8361,R2    GET STRING LENGTH
0085         SRL  R2,8         RIGHT JUSTIFY
0086         JEQ  NODIS        IF ZERO, EXIT
0087         MOVB @>8367,R1    GET LOW BYTE OF POINTER TO
STRING
0088         SRL  R1,8         RIGHT JUSTIFY
0089         AI   R1,>8300     ADD 8300 HIGH BYTE
0090         BLWP @VMBW        DISPLAY THE NUMBER STRING
0091  NODIS  RT                RETURN
0092  *
0093  *  GENERAL PURPOSE GPL LINK
0094  *  BY DOUG WARREN AND CRAIG MILLER
0095  *
0096  GR4    EQU  GPLWS+8
0097  GR6    EQU  GPLWS+12
0098  STKPNT EQU  >8373
0099  LDGADD EQU  >60
0100  XTAB27 EQU  >200E
0101  GETSTK EQU  >166C
0102  *
0103  GPLLNK DATA GLNKWS
0104         DATA GLINK1
0105  RTNAD  DATA XMLRTN
0106  GXMLAD DATA >176C
0107         DATA >50
0108  GLNKWS EQU  $->18
0109         BSS  >08
0110  GLINK1 MOV  *R11,@GR4
0111         MOV  *R14+,@GR6
0112         MOV  @XTAB27,R12
0113         MOV  R9,@XTAB27
0114         LWPI GPLWS
0115         BL   *R4
0116         MOV  @GXMLAD,@>8302(R4)
0117         INCT @STKPNT
0118         B    @LDGADD
0119  XMLRTN MOV  @GETSTK,R4
0120         BL   *R4
0121         LWPI GLNKWS
0122         MOV  R12,@XTAB27
0123         RTWP
0124  *
0125  * DATA SECTION
0126  WS     BSS  32           WORKSPACE
0127  PABDT  BYTE 5            CALL NAME LENGTH
0128         TEXT 'FILES'      NAME
0129         BYTE >B7,>C8,1,'9',>B6  TOKENIZED CODE FOR
(9)
0130  ANYKEY BYTE >20          NUMBER FOR KEYSTROKE
CHECKING
0131  PROMPT BYTE 27
0132         TEXT 'HOW MANY FILES? PRESS (1-9)'
0133  PAK    BYTE 21
0134         TEXT 'PRESS ANY KEY TO EXIT'
0135  ONSTR  BYTE 19
0136         TEXT 'OLD NUMBER AT >8370'
0137  NNSTR  BYTE 19
0138         TEXT 'NEW NUMBER AT >8370'
0139         END
```

MYARC ADVANCED BASIC

Hard drive utility finds, records formatting information

By JIM UZZELL
©1995 DDI Software

This month's program is special. Special only to the extent that it is for hard drive users only.

This program requires the object code file "BREAD-O" published in the November 1994 issue of MICROpendium, page 11, and is set up to load from drive No. 1 in line 170. Some users will have to change the printer name in line 990.

This program was designed to return information based on MDM5's hard drive formatter and may not work with third-party software.

Those of you who have had the unfortunate task of reformatting your hard drive recently will recognize the information provided by this program.

If you have been lucky and your drive has never crashed, did you write down your format information? This program will provide you with a hard copy for safekeeping (I keep mine in the front of the hard drive manual).

HIGHEST/LOWEST LINE NUMBER

To round out this article so all MY-BASIC users will get something out of this article, here is a tidbit you may have forgotton about; with a program in memory, from the prompt type list 1 or list 32767 to find the lowest or highest line number of the program.

HARDDRIVE INFO

```
100 !DDI SOFTWARE
110 !COPYRIGHT 1994
120 !HARDDRIVE INFO
130 CALL GRAPHICS(2,2)
140 CLS
150 CALL TCOLOR(2,2)
160 CALL INIT
170 CALL LOAD("DSK1.BREAD-O"
)
180 CALL LOAD(9481,1)
190 CALL TCOLOR(4,2)
200 DISPLAY AT(7,17):" HDS1.
 ";
210 ACCEPT AT(7,21)SIZE(-1)V
ALIDATE("123"):X$(1)
220 X$(1)="HDS"&X$(1)&"."
230 CALL LINK("GO",X$())
240 FOR X=1 TO 10
250 READ A$(X)
260 NEXT X
270 CALL TCOLOR(2,2)
280 DISPLAY AT(9,8):A$(10);
290 CALL TCOLOR(4,2)
300 FOR X=1 TO 9 STEP 2
310 DISPLAY AT(X+9,8):A$(X);
320 NEXT X
330 CALL TCOLOR(10,2)
340 FOR X=2 TO 8 STEP 2
350 DISPLAY AT(X+9,8):A$(X);
360 NEXT X
370 CALL TCOLOR(2,2)
380 DISPLAY AT(19,8):A$(10);
390 B$="" :: FOR X=0 TO 9
400 CALL PEEK(9618+X,A)
410 IF A=32 THEN 430
420 B$=B$&CHR$(A)
430 NEXT X :: C$(0)=B$
440 CALL TCOLOR(4,2)
450 DISPLAY AT(10,31-LEN(B$)
):B$;
460 CALL PEEK(9630,A)
470 CALL TCOLOR(10,2)
480 DISPLAY AT(11,28):A;
490 C$(1)=A$(2)&STR$(A)
500 CALL PEEK(9632,A)
510 CALL TCOLOR(4,2)
520 DISPLAY AT(12,29):A;
530 C$(2)=A$(3)&STR$(A)
540 CALL PEEK(9633,A)
550 CALL TCOLOR(10,2)
560 DISPLAY AT(13,27):A*8;
570 C$(3)=A$(4)&STR$(A*8)
580 CALL PEEK(9634,A)
590 CALL TCOLOR(4,2)
600 DISPLAY AT(14,29):A-47;
610 C$(4)=A$(5)&STR$(A-47)
620 CALL PEEK(9635,A)
630 CALL TCOLOR(10,2)
640 DISPLAY AT(15,27):A*16;
650 C$(5)=A$(6)&STR$(A*16)
660 CALL PEEK(9645,A)
670 CALL TCOLOR(4,2)
680 IF A<1 THEN B$="NO" :: C
L=29 ELSE B$="YES" :: CL=28
690 C$(6)=A$(7)&B$
700 DISPLAY AT(16,CL):B$;
710 SU=0 :: FOR X=1 TO 228 S
TEP 2
720 CALL PEEK(9646+X,A)
730 IF A>0 THEN SU=SU+1
740 NEXT X
750 CALL TCOLOR(10,2)
760 DISPLAY AT(17,28):SU;
770 C$(7)=A$(8)&STR$(SU)
780 GOSUB 1060
790 CALL TCOLOR(6,5)
800 DISPLAY AT(20,12):"1994
DDI SOFTWARE";
810 CALL TCOLOR(4,2)
820 DISPLAY AT(24,1):" P=HAR
DCOPY & Exit "
830 CALL TCOLOR(10,2)
840 DISPLAY AT(24,32):"ESC=E
XIT";
850 CALL KEY(0,K,S) :: IF S<
1 THEN 850
860 IF K=80 THEN GOSUB 990 :
: GOTO 880
870 IF K<>155 THEN 850
880 CALL TCOLOR(16,6) :: END
881 REM DO NOT TYPE #'s IN (
 )
882 REM BELOW THESE ARE # OF
 SPACES
890 DATA " VOL NAME
  (16) "
900 DATA " SECTORS/TRACK
  (11) "
910 DATA " STEP RATE
  (15) "
920 DATA " REDUCED WRITE CUR
R (6)  "
930 DATA " NO. OF HEADS
  (12) "
940 DATA " WRITE PRECOMP
  (11) "
950 DATA " EMULATE ON
  (14) "
960 DATA " SUBDIR IN MAIN DI
R (6)  "
970 DATA " DRIVE SIZE-MEGS
```

(See Page 10)

MBASIC—

(Continued from Page 9)

```
  (9)  "
980 DATA "
  (25) "
990 OPEN #1:"PIO"
1000 PRINT #1:TAB(38);X$(1)
1010 PRINT #1:TAB(34);"DATE
";DATE$
1020 PRINT #1:TAB(28);A$(1)&
C$(0)
1030 FOR X=1 TO 8
1040 PRINT #1:TAB(28);C$(X)
1050 NEXT X :: CLOSE #1 :: R
ETURN
1060 CALL PEEK(9628,A,B)
1070 CALL PEEK(9634,C)
1080 C1$=HEX$ (C)
1090 C1=(VALHEX(SEG$(C1$,3,1
))+1)
1100 D=(((A*256)*C1)+(B*C1))
*256
1110 IF LEN(STR$(D))=7 THEN
1140
1120 IF LEN(STR$(D))=8 THEN
1150
1130 IF LEN(STR$(D))>8 THEN
1160
1140 D$=LEFT$ (STR$(D),1) ::
 CL=30 :: GOTO 1170
1150 D$=LEFT$ (STR$(D),2) ::
 CL=29 :: GOTO 1170
1160 D$=LEFT$ (STR$(D),3) ::
 CL=28 :: GOTO 1170
1170 CALL TCOLOR(4,2)
1180 DISPLAY AT(18,CL):D$;
1190 C$(8)=A$(9)&D$ :: RETUR
N
```

Identifying files

Uncovering the identify of 'mystery' programs is a process of elimination

By DEAN HANCOCK

This article appeared in the newsletter of the British Columbia 99er Users.—Ed.

Well it must be something. It's on my disk. But what is it?

How many times have you asked yourself that question — especially after you carefully typed in the correct filename for a program only to have the computer make some rude noises at you and put a meaningless message on the screen. The question of what different filenames and their cryptic types will NOT be completely solved here, but I will attempt to shed some light on it.

The reason I decided to type up some information on files and file types is because somewhere along the way, at one of our Thursday night meetings, someone asked the question about filenames, types and how do you load them. Of course, numbers of answers were given to various questions and more than likely the answers were forgotten (mostly due to the fact that there was almost too much data to consume in one evening and if you didn't write all of the information down you forgot it by the time you got home — or misplaced the paper you wrote it on).

Now let it be known that 99.9 percent of the information that follows is not mine. I have found articles written by Earl Raguse in the Los Angeles 99er TopIcs newsletters, plus some information again from L.A. 99er TopIcs by Tom Freeman that was in the Ozark 99 User Group newsletter and sent to L.A. by Steve Langguth. To make matters even more interesting, one of the articles is a compilation of three other articles appearing in other newsletters. One article was written by R.A. Green of the Ottawa 99/4 Users Group. It originally appeared in the May 1986 New Jersey Users Group newsletter and was reprinted in the September 1986 newsletter of the Kansas City TI 99/4A Users Group. Another article was written by Jerome Trinkl and was somewhat more detailed. It appeared first in the April 1986 newsletter of the Atlanta 99/4A Computer Users Group and was reprinted in the Greater Akron (Ohio) 99ers September 1986 newsletter. And the more TI club newsletters I read, the more times I see it appearing in their newsletters in various write-ups. Are we confused yet? So one can see that the articles have been around.

SEVEN WAYS

There are seven different ways to store programs in the TI99/4A. I'll try to list the program files and what to expect. The biggest problem we have is which cartridge do we need in the console. Let's try to stick with the Extended BASIC and the Editor/Assembler cartridges. There are numerous variations of both those cartridges, but let's stick with the stock TI ones, since most users has those.

Most users are familiar with the form used by TI BASIC to store programs on cassette or disk. It is identified as "PROGRAM" when a disk is cataloged. It is created or stored by the BASIC "SAVE" command and loaded into the computer by the Basic "OLD" command. This is the only way that TI BASIC uses to store programs.

Extended BASIC can, and usually does, use the same format as console BASIC to store programs. There are, however, two other forms that Extended BASIC can use to store programs on a disk (but not on cassette). If you have the 32K Memory Expansion (or almost any setup that gives you 32K memory), you can write an Extended BASIC program that is too large to store in the usual format. Extended BASIC will store these large programs in an "INTERNAL/VARIABLE 254" file. (Depending on what method you use to read a disk, the file will show up as I/V 254 or INT/VAR 254.) The usual "SAVE" and "OLD" commands are used to store and load these programs. The third form of storing programs used by XB is the "merge format" as a "DISPLAY/VARIABLE 163" file. Again, it could show as a D/V 163 or DIS/VAR 163. This form is

(See Page 11)

IDENTIFYING FILES—

(Continued from Page 10)

created when the "MERGE" option is specified with the "SAVE" command. If you are not familiar with the merge format, see pages 122 and 123 of the TI Extended BASIC manual. The beauty of merge format is that when it is loaded, it does not necessarily overwrite the program in memory. The "MERGE" command does just that — it merges the new program (or program segment) with the program in memory according to the line numbers. A line from a file being "MERGEd" that has the same number as one already in memory will overwrite the old line.

So let's quickly review what just transpired (Is that like sweat?) You've cataloged a disk and found three files that are listed as "PROGRAM," two files listed as "INT/VAR 254" and one file showing up as "DIS/VAR 163." The INT/VAR 254 files can be loaded via "OLD DSKn.Filename" and then run. With any degree of luck you will probably be looking at a new game, or whatever, doing its thing on your TI99/4A. Doing the same with the "PROGRAM" files could get you the same results and, if the program runs, go for it. You can load the DIS/VAR 163 file into the computer and edit it to see if it is an update to one of the other files — or it might not even be related to any of them, but was on that particular disk.

If you type in wrong filenames, you will get an error message on the screen. If you typed everything in perfectly and your "PROGRAM" file still gives you an error message, do not despair. There are other possibilities as to what that "PROGRAM" file is. Nobody said the computer would be perfect. There's more to come.

ASSEMBLY PROGRAMS

Now, we get to the good stuff, assembly language programs. There are three formats for assembly language programs:

1. tagged object
2. compressed tagged object
3. memory image.

Tagged object code files are stored in "DISPLAY/FIXED 80" files on disk only. All program data are in hexadecimal code so that they can be edited by the E/A editor. Tagged object code can be loaded by "CALL LOAD" in XB, by option 3 ("Load and Run") on the E/A menu, by option 1 on the Mini-Memory module menu, or by "CALL LOAD" in TI BASIC when either the Editor/Assembler or Mini-Memory cartridge is plugged in. The programs can have *absolute addresses* or be *relocatable*. A program with an absolute address is always loaded into the same place in memory. A relocatable program can be loaded into any place in memory. A tagged object program can have references to other programs or subroutines.

Compressed tagged object code is very nearly the same as tagged object code except that the program data is stored as bytes rather than as hexadecimal digits.

The loader will resolve these *external references*, except for the XB loader.

Compressed tagged object code is very nearly the same as tagged object code except that the program data is stored as bytes rather than as hexadecimal digits. Compressed tagged object code loads faster than regular tagged object code, as you would expect. The XB loader cannot load compressed object code.

Tagged object code, in either regular or compressed form (compressed if the "C" option was chosen while assembling), can be produced by the assembler when it "assembles" source code.

The *memory image* format of assembly language program is the most compact and the fastest loading. It can be stored on cassette or disk. It is identified as "PROGRAM" in the disk catalog (just like a BASIC or XB program). Memory image programs can be loaded by option 5 on the E/A cartridge menu or by option 3 on the TI-Writer menu. It should be noted that there is one small difference between how the E/A calls a memory image program and how TI-Writer does it. TI-Writer blanks the screen just before calling the program and E/A does not. This means the program being loaded must turn the screen back on or nothing will show. Memory image programs are created by a utility program, like the one called "SAVE" that is provided on the E/A disk.

There is a limit on the size of an assembler memory image file of >2000 bytes. The reason for this limit is that the various loaders use the 16K VDP memory to transfer the data during the Device Service Routine (DSR), and only that number of bytes have been allocated in the transfer buffer. However, the E/A and TI-Writer cartridges will load multiple memory files to make up a program of any length. They use the convention that the name of the second and following files is obtained by incrementing the last digit or letter of the previous file name. For example, the TI-Writer editor consists of two memory fields — EDITA1 and EDITA2.

As a matter of interest, the Adventure, Tunnels of Doom, Personal Record Keeping, Statistics, and Personal Report Generator cartridges all use memory image files to store data. The fact that memory image files can be saved or loaded with single input/output operation makes them attractive for such uses.

MEMORY IMAGE

Now, let's take a closer look at memory image format. Assembler memory image files have a three-word "header" followed by the program data to be placed in memory. this three-word header is not "loaded," but rather directs the loading process. The three words are used as follows:

1. The first word is a "flag." If it is "FFFF," then this flag is not the last in a multi-file program. If it contains "0000," then it is the last file.

2. This word contains the length of the file in bytes (approximately >2000), including the six-byte header.

3. The third word is the CPU memory address where the memory image is to be loaded.

Execution of a memory image program always begins at the first byte of the segment loaded.

Okay, enough already. Actually, there

(See Page 12)

IDENTIFYING FILES—

(Continued from Page 11)

is one little bit left after this. But first, let's take a look at what we have. Basically, if you can't get a PROGRAM file to run with the XBASIC cartridge in place, you are going to have to turn off the unit, and install the Editor/Assembler cartridge in its place. Restart the computer. Get E/A on screen and try Option 5 — especially if this program is 33 sectors and happens to have another program file of almost any size up to 33 following that has the last letter or digit incremented by one. I'm going to put a list at the end of all this that will hopefully do something for you. As always, if all else fails - pick up the phone and talk to another computer club member — he or she might have the answer and save you lots of grief.

Finally, there is a seventh form for program files. This form is created and loaded by the "Easy Bug" of the Mini-Memory cartridge. It can only be written to cassette and is memory image, but it is slightly different from the E/A memory image files. The Easy Bug memory image program can consist of only one segment. The header for Easy Bug format consists of only two words, as follows:

1. This word is the CPU memory address at which the memory image is to be loaded.

2. This word is the length of the memory data, not including the four header bytes.

Okay, now that we have gone through all that, let's try to sort a bit of it out. Again, I am back to reading articles from the various newsletters that we exchange with other TI computer clubs throughout Canada and the States. A lot of thanks goes to Earl Raguse for information I have used as well as Tom Freeman, and I can't guess how many others have put their little bit of info in as well. As Raguse writes - "A partial list of some newsletters which have published good file info are: the UGOC ROM, LA99ers TopIcs, Long Island 99er, St Louis Computer Bridge, Greater Akron 99er, Columbus' Spirit of 99, Cleveland Area UG newsletter, Birmingham BUG, etc."

Other sources of information on files are indeed the TI Extended BASIC and TI Editor/Assembler manuals.

Down to the nitty gritty. BASIC means TI BASIC, XB means Extended BASIC, and A/L means assembly language.

If the file is 25 sectors, you could have files that relate to any number of graphic programs that are on the market - too numerous to list. A hint would be if the filename is followed by "_C" or "_P." A hint to the nature of the files on a disk can also be gleaned from the disk name — presuming the disk has a name — like ARTISTPGMS, indicating the disk might have something to do with one of those numerous graphic programs.

IDENTIFICATION GUIDELINES

If the file is less than 33 sectors, try BASIC, then XB, then A/L.

If the file is 33 sectors, try A/L.

If the file is 34 sectors, then it is probably a GRAM-U-LATOR or GRAM Kracker file. They tend to end in numbers from 1 to 7. If you don't have a GRAM device, you will not get them to load or work or do anything using BASIC, XB or A/L.

If the file is over 34 sectors, try BASIC, then XB. You may need to do a CALL FILES, NEW OLD DSKn.Filename, RUN. See the XB manual for CALL FILES if you are unfamiliar with it. Also, it could be a FORTHSAVE file, which requires the Forth kernel.(Another story unto itself.)

If the file is 52 sectors, you may be looking at a Tunnels of Doom file.

If the file is 54 sectors, you are may be looking at a Scott Adams Adventure file.

If the file is listed as a "Program" but still won't run, don't erase it immediately as a lost cause - it could be data for another program on the disk, which will not run if the erased file is not on the disk. Did anybody say this was going to be easy?

Now we'll go from the PROGRAM file to some of the other files that can appear on your disk.

If the file is an INT/VAR 254, it's probably XB - you'll need 32K memory. If the file happens to be called LOAD, you're in luck, since XB looks for a file called LOAD on the disk and automatically loads it from the main menu. (Although you do have to press 2, to get XB.)

If the file is a DIS/VAR 163, it's most likely an XB merge file.

If the file is a DIS/FIX 80, it's most likely an A/L program.

If the file is a DIS/VAR 80 file, it probably is a text file for TI-Writer or any of the numerous word processors programs available to TI users—again almost too numerous to list. Look at the file.

If the file is a DIS/FIX 128, it may be an archived file which will need a program that can un-arc the information into separate usable files.(This is yet another story.) We are also looking at the possibility that it might even be Forth.

You are quite likely somewhat confused by now, and there is still more.

GRAPHIC PROGRAMS

Quickly looking at graphics programs, we can have some of the following goodies to deal with:

If the file is a 25-sector _C or _P, we are dealing with the color and pattern portions of TI-Artist.

If the file is any sector length DIS/VAR 80 with _F as the ending, we probably have a character font.

If the file is an 18-sector DIS/VAR 80 _S, it most likely is a TI-Artist Slide.

If the file is a DIS/VAR 80 _I, take it to be an Instance.

If the file is a DIS/FIX 12 _V, take it to be a vector by TI-Artist Plus.

If the file is a DIS/FIX 254 _M, take it to be a Movie by TI-Artist Plus.

If you happen to be looking at TIPS (Texas Instruments Print Shop) then we have some more files:

I/F 53 is a Picture (TXT).

I/F 16 is name text (XXX).

D/V 250 is spooled graphic.

I/V 254 is a TIPS Font.

I'm sure if I tried I could probably go on ad infinitum, as file types tend to go on ad infinitum. Just because you can't get a file to load the first time, don't give up on it. I would hesitate to guess how many pages one could fill of different file types available in the TI world. If all else fails, pick up the telephone and call a club member, or bring it to a club meeting.

But you can see that not all needs to be lost because the file won't load.

BASIC

Tic-Tac-Toe Philadelphia-style

The following program is a variation of Tic-Tac-Toe written by Barry Traver. Documentation is included in the program, which can run from BASIC or Extended BASIC. The "Philadelphia-style" version of the game is more challenging than standard varieties of the game. The game doesn't permit draws, for example.

TICTACTOE

```
90 REM TI BASIC OR EXTENDED BASIC (BUT SEE LINE 130 IF YOU HAVE VERSION 100) !233
100 REM TIC/TAC/TOE (PHILADELPHIA STYLE) BY B.A. TRAVER !237
110 REM WARNING: THIS GAME IS REALLY A CREAM-CHEESE PUT-ON, FOR--PLAYED STRICTLY--THE 2ND PLAYER CAN'T WIN! !121
120 REM FOR FURTHER INFORMATION, SEND S.A.S.E. TO B.A. TRAVER, 835 GREEN VALLEY DR., PHILA., PA 19128 !175
130 REM FOR EXTENDED BASIC VERSION 100 THIS LINE SHOULD READ GOSUB 3800 !225
140 CALL SCREEN(14)!199
150 CALL CLEAR !209
160 PRINT TAB(5);"T I C - T A C - T O E": :TAB(5);"(Philadelphia Style)": :TAB(6);"An Original Program": :!058
170 PRINT :TAB(3);"Released to Public Domain": : :TAB(14);"by": :TAB(10);"B.A. Traver": :TAB(4);"835 Green Valley Drive" !040
180 PRINT :TAB(7);"Phila., PA 19128": : :TAB(2);"TI BASIC or Extended BASIC" !110
190 FOR I=1 TO 10 !105
200 READ F(I)!147
210 NEXT I !223
220 GOSUB 3680 !190
230 FOR I=1 TO 9 !064
240 READ RW(I),CL(I)!238
250 NEXT I !223
260 W=0 !014
270 L=0 !003
280 GOSUB 3680 !190
290 CALL CHAR(42,"3366CC9933 66CC99")!066
300 CALL CHAR(128,"")!212
310 CALL CHAR(129,"")!213
320 CALL CHAR(130,"")!205
330 CALL CHAR(131,"")!206
340 CALL CHAR(132,"1818181818181818")!039
350 CALL CHAR(133,"181818FFFF181818")!110
360 CALL CHAR(134,"000000FFFF")!019
370 CALL CHAR(136,"0060703810E0703")!046
380 CALL CHAR(137,"03070E1C38706000")!047
390 CALL CHAR(138,"00060E1C3870E0C0")!078
400 CALL CHAR(139,"C0E0703810E0600")!079
410 CALL CHAR(140,"030F3E306060E0C0")!070
420 CALL CHAR(141,"C0E0606030E0F03")!071
430 CALL CHAR(142,"C0F07C0C06060703")!076
440 CALL CHAR(143,"0307060C7CF0C0")!077
450 CALL CLEAR !209
460 CALL SCREEN(12)!197
470 P$="  Press any key when ready." !103
480 PRINT "    Tic-Tac-Toe is usually a": :"very  B-O-R-I-N-G  game for": :"two reasons:" !134
490 PRINT :"     1) The best the second": :"player can ordinarily hope": :"for is a tie." !030
500 PRINT :"     2) The game is rather": :"predictable, with the first": :"player almost always opening" !145
510 PRINT :"in the center square." !229
520 PRINT : :P$ !122
530 GOSUB 3560 !069
540 CALL CLEAR !209
550 CALL SCREEN(4)!149
560 PRINT "TIC-TAC-TOE (Philadelphia  Style) is like standard  Tic-Tac-Toe with two major  differences:": :!225
570 PRINT " 1) The first player must  win to win (i.e., a 'tie'  is a win for the second  player!)." !189
580 PRINT " 2) Neither player can move  in the center square  unless that move blocks" !190
590 PRINT "  the opponent from three  in a row (or results in  three in a row for the  player!)." !217
600 PRINT :"This entertaining variation was invented to give the  second player a more equal" !099
610 PRINT "opportunity to win.  Try it!": :P$ !118
620 GOSUB 3560 !069
630 CALL CLEAR !209
640 CALL SCREEN(8)!153
650 PRINT "Do you want me to be strict about your not occupying thecenter square other than" !126
660 PRINT "blocking (or obtaining)  three-in-a-row?": :!124
670 PRINT "(By the way, to give you  every chance, I will myself follow that rule even if I" !173
680 PRINT "allow you to disregard it.  After all, you do have the  handicap of trying to match" !251
690 PRINT "wits with a computer!)": :!224
700 PRINT "Your choice (1 for strict,  2 for not so strict)?": : : :!212
710 GOSUB 3560 !069
720 IF K=50 THEN 860 !148
730 IF K<>49 THEN 710 !198
740 PRINT TAB(10);"Number one?":TAB(8);"You ARE brave!": :!088
750 CALL SOUND(300,131,0)!125
```

(See Page 14)

TIC-TAC-TOE PHILADELPHIA STYLE—

(Continued from Page 13)

```
760 CALL SOUND(100,147,0)!13
0
770 CALL SOUND(200,156,0)!13
1
780 CALL SOUND(250,131,0)!12
9
790 CALL SOUND(125,131,30)!1
82
800 CALL SOUND(300,131,0)!12
5
810 CALL SOUND(100,147,0)!13
0
820 CALL SOUND(200,156,0)!13
1
830 CALL SOUND(200,131,0)!12
4
840 CALL SOUND(600,185,0)!13
7
850 GOTO 880 !194
860 PRINT TAB(10);"Number tw
o?":TAB(12);"Coward!": :!144
870 CALL SOUND(750,-3,0)!228
880 G=K-48 !072
890 GOSUB 2130 !170
900 CALL CLEAR !209
910 CALL SCREEN(12)!197
920 A$="      *" !219
930 B$=A$&"**************" !
098
940 C$=A$&"            *", !
225
950 D$=A$&"   *   *    *" !
246
960 E$=A$&" ********** *" !
081
970 PRINT "Here's how the pl
aying boardis numbered:": :!
166
980 PRINT B$:C$:D$:D$:D$:E$:
D$:D$:D$:E$:D$:D$:D$:C$:B$:
:!000
990 K=49 !064
1000 FOR I=11 TO 19 STEP 4 !
083
1010 FOR J=12 TO 20 STEP 4 !
077
1020 GOSUB 2110 !150
1030 CALL HCHAR(I,J,K)!129
1040 K=K+1 !015
1050 NEXT J !224
1060 NEXT I !223
1070 PRINT P$: :!122
1080 GOSUB 3560 !069
1090 GOSUB 1820 !115
1100 GOSUB 2110 !150
1110 Y=1 !017
1120 R=19 !068
1130 GOSUB 2350 !135
1140 M=1 !005
1150 GOSUB 3640 !150
1160 IF (G=1)*(M=5)THEN 1200
 !086
1170 IF M=1 THEN 2170 !132
1180 GOSUB 2350 !135
1190 ON M-1 GOTO 1240,1260,1
280,1300,1320,1340,1360,1380
 !109
1200 GOSUB 2110 !150
1210 GOSUB 2000 !039
1220 GOSUB 2110 !150
1230 GOTO 2220 !003
1240 GOSUB 3610 !120
1250 ON M GOTO 2170,2200,262
0,1400,2620,2620,2170,2620,2
620 !150
1260 GOSUB 3630 !140
1270 ON M GOTO 2170,2640,220
0,2640,1430,2640,2640,2640,2
170 !024
1280 GOSUB 3590 !100
1290 ON M GOTO 2170,1460,217
0,2200,2580,2580,2580,2580,2
580 !010
1300 GOSUB 3630 !140
1310 GOTO 1490 !038
1320 GOSUB 3610 !120
1330 ON M GOTO 2170,2620,262
0,1680,2620,2200,2170,2620,2
620 !175
1340 GOSUB 3630 !140
1350 ON M GOTO 2170,2640,264
0,2640,1720,2640,2200,2640,2
170 !059
1360 GOSUB 3590 !100
1370 ON M GOTO 2170,1750,217
0,2580,2580,2580,2580,2200,2
580 !045
1380 GOSUB 3590 !100
1390 ON M GOTO 2170,1790,217
0,2580,2580,2580,2580,2580,2
200 !085
1400 GOSUB 2350 !135
1410 GOSUB 3630 !140
1420 ON M GOTO 2170,2200,270
0,2200,2700,2700,2170,2640,2
170 !249
1430 GOSUB 2350 !135
1440 GOSUB 3610 !120
1450 ON M GOTO 2170,2620,220
0,2700,2200,2620,2170,2620,2
170 !069
1460 GOSUB 2350 !135
1470 GOSUB 3630 !140
1480 ON M GOTO 2170,2200,217
0,2200,2660,2640,2660,2660,2
170 !129
1490 IF (M=1)+(M=5)+(M=9)THE
N 1510 !204
1500 GOSUB 2350 !135
1510 ON M GOTO 2170,1520,155
0,1570,2200,1600,1630,1650,2
170 !014
1520 M=8 !012
1530 GOSUB 3640 !150
1540 ON M GOTO 2170,2200,268
0,2680,2200,2680,2460,2170,2
170 !008
1550 GOSUB 3610 !120
1560 ON M GOTO 2170,2620,220
0,2700,2200,2620,2170,2620,2
170 !069
1570 M=6 !010
1580 GOSUB 3640 !150
1590 ON M GOTO 2170,2600,249
0,2200,2200,2170,2600,2600,2
170 !054
1600 M=4 !008
1610 GOSUB 3640 !150
1620 ON M GOTO 2170,2680,268
0,2170,2200,2200,2520,2680,2
170 !068
1630 GOSUB 3590 !100
1640 ON M GOTO 2170,2660,217
0,2580,2200,2580,2200,2580,2
170 !165
1650 M=2 !006
1660 GOSUB 3640 !150
1670 ON M GOTO 2170,2170,255
0,2600,2200,2600,2600,2200,2
170 !114
1680 GOSUB 2350 !135
1690 M=5 !009
1700 GOSUB 3640 !150
1710 ON M GOTO 2170,2600,272
0,2200,2170,2200,2170,2600,2
600 !029
1720 GOSUB 2350 !135
1730 GOSUB 3590 !100
1740 ON M GOTO 2170,2660,217
0,2580,2200,2580,2200,2580,2
170 !165
1750 GOSUB 2350 !135
1760 M=5 !009
```

(See Page 15)

TIC-TAC-TOE PHILADELPHIA STYLE—

(Continued from Page 14)

```
1770 GOSUB 3640 !150
1780 ON M GOTO 2170,2200,217
0,2680,2170,2680,2720,2200,2
680 !013
1790 GOSUB 2350 !135
1800 GOSUB 3610 !120
1810 ON M GOTO 2170,2200,217
0,2640,2620,2620,2170,2620,2
200 !009
1820 CALL CLEAR !209
1830 CALL SCREEN(8)!153
1840 CALL COLOR(14,8,8)!237
1850 CALL COLOR(13,8,8)!236
1860 FOR I=2 TO 17 !113
1870 CALL HCHAR(I,9,128,16)!
056
1880 NEXT I !223
1890 CALL HCHAR(7,10,134,14)
!019
1900 CALL HCHAR(12,10,134,14
)!064
1910 CALL VCHAR(3,14,132,14)
!031
1920 CALL VCHAR(3,19,132,14)
!036
1930 CALL HCHAR(7,14,133)!05
2
1940 CALL HCHAR(7,19,133)!05
7
1950 CALL HCHAR(12,14,133)!0
97
1960 CALL HCHAR(12,19,133)!1
02
1970 CALL COLOR(14,2,16)!023
1980 CALL COLOR(13,2,16)!022
1990 RETURN !136
2000 P=128 !116
2010 GOTO 2050 !089
2020 P=136 !115
2030 GOTO 2050 !089
2040 P=140 !110
2050 CALL SOUND(-175,F(M),0)
!233
2060 CALL HCHAR(RW(M),CL(M)+
2,P)!091
2070 CALL HCHAR(RW(M),CL(M)+
3,P+2)!024
2080 CALL HCHAR(RW(M)+1,CL(M
)+3,P+3)!212
2090 CALL HCHAR(RW(M)+1,CL(M
)+2,P+1)!209
2100 RETURN !136
2110 D=45 !053
2120 GOTO 2140 !179
2130 D=135 !102
2140 FOR DELAY=1 TO D !168
2150 NEXT DELAY !005
2160 RETURN !136
2170 GOSUB 2110 !150
2180 GOSUB 2020 !059
2190 GOTO 2220 !003
2200 GOSUB 2110 !150
2210 GOSUB 2040 !079
2220 R=20 !060
2230 C=5 !255
2240 M$="Illegal move--I win
!" !205
2250 GOSUB 2790 !064
2260 GOTO 2940 !214
2270 R=19 !068
2280 C=6 !000
2290 M$="What's your move?"
!058
2300 GOSUB 2790 !064
2310 C=24 !049
2320 GOSUB 2870 !145
2330 GOSUB 2110 !150
2340 RETURN !136
2350 C=9 !003
2360 M$="My move is" !153
2370 GOSUB 2790 !064
2380 C=20 !045
2390 RETURN !136
2400 GOSUB 2820 !095
2410 GOSUB 2110 !150
2420 GOSUB 2020 !059
2430 IF Y=0 THEN 2450 !168
2440 GOSUB 2270 !054
2450 RETURN !136
2460 GOSUB 2350 !135
2470 GOSUB 3590 !100
2480 ON M GOTO 2170,2200,217
0,2660,2200,3440,2200,2170,2
170 !234
2490 GOSUB 2350 !135
2500 GOSUB 3610 !120
2510 ON M GOTO 2170,2700,220
0,2200,2200,2170,2170,3420,2
170 !254
2520 GOSUB 2350 !135
2530 GOSUB 3590 !100
2540 ON M GOTO 2170,3480,217
0,2170,2200,2200,2200,2580,2
170 !194
2550 GOSUB 2350 !135
2560 GOSUB 3610 !120
2570 ON M GOTO 2170,2170,220
0,3460,2200,2620,2170,2200,2
170 !214
2580 M=2 !006
2590 GOTO 2730 !003
2600 M=3 !007
2610 GOTO 2730 !003
2620 M=4 !008
2630 GOTO 2730 !003
2640 M=5 !009
2650 GOTO 2730 !003
2660 M=6 !010
2670 GOTO 2730 !003
2680 M=7 !011
2690 GOTO 2730 !003
2700 M=8 !012
2710 GOTO 2730 !003
2720 M=9 !013
2730 M$="My move is "&STR$(M
)&" and I win!" !220
2740 Y=0 !016
2750 GOSUB 2780 !054
2760 GOSUB 2410 !195
2770 GOTO 2940 !214
2780 C=15-LEN(M$)/2 !101
2790 IF R<>19 THEN 2810 !006
2800 GOSUB 2130 !170
2810 CALL HCHAR(R,3,32,28)!0
07
2820 FOR I=1 TO LEN(M$)!241
2830 CALL HCHAR(R,C+2,ASC(SE
G$(M$,I,1)))!156
2840 C=C+1 !255
2850 NEXT I !223
2860 RETURN !136
2870 GOSUB 3560 !069
2880 IF (K<49)+(K>57)THEN 28
70 !091
2890 M=K-48 !078
2900 M$=STR$(M)!196
2910 GOSUB 2820 !095
2920 GOSUB 2110 !150
2930 RETURN !136
2940 GOSUB 2110 !150
2950 GOSUB 3680 !190
2960 W=W+1 !039
2970 GOTO 3010 !028
2980 GOSUB 2110 !150
2990 GOSUB 3740 !250
3000 L=L+1 !017
3010 M$="I have now won "&ST
R$(W)&" game" !175
3020 IF W=1 THEN 3050 !001
3030 M$=M$&"s." !194
3040 GOTO 3060 !078
3050 M$=M$&"." !078
3060 R=21 !061
```

(See Page 16)

TIC-TAC-TOE PHILADELPHIA STYLE—

(Continued from Page 15)

```
3070 GOSUB 2780 !054
3080 M$="You have now won "&
STR$(L)&" game" !154
3090 IF L=1 THEN 3120 !061
3100 M$=M$&"s." !194
3110 GOTO 3130 !149
3120 M$=M$&"." !078
3130 R=22 !062
3140 GOSUB 2780 !054
3150 M$="Another game (yes/n
o)?" !164
3160 R=23 !063
3170 C=2 !252
3180 GOSUB 2790 !064
3190 GOSUB 3560 !069
3200 IF K=78 THEN 3330 !078
3210 IF K<>89 THEN 3190 !132
3220 M$="Yes" !042
3230 C=25 !050
3240 GOSUB 2820 !095
3250 FOR M=1 TO 9 !068
3260 GOSUB 2000 !039
3270 NEXT M !227
3280 CALL SOUND(350,F(10),0)
!000
3290 FOR I=19 TO 24 !168
3300 CALL HCHAR(I,3,32,28)!2
54
3310 NEXT I !223
3320 GOTO 1110 !169
3330 M$="No" !181
3340 C=26 !051
3350 GOSUB 2820 !095
3360 M$="Thank you for playi
ng!" !027
3370 R=24 !064
3380 GOSUB 2780 !054
3390 GOSUB 2130 !170
3400 GOSUB 3740 !250
3410 GOTO 3860 !114
3420 M=2 !006
3430 GOTO 3490 !254
3440 M=4 !008
3450 GOTO 3490 !254
3460 M=6 !010
3470 GOTO 3490 !254
3480 M=8 !012
3490 M$="My move is "&STR$(M
)&", but you ""win."" !138
3500 GOSUB 2780 !054
3510 GOSUB 2110 !150
3520 GOSUB 2020 !059
3530 GOTO 2980 !254
3540 PRINT TAB(7);"*    *
*    *" !151
3550 RETURN !136
3560 CALL KEY(0,K,S)!187
3570 IF S=0 THEN 3560 !252
3580 RETURN !136
3590 M=3 !007
3600 GOTO 3640 !149
3610 M=7 !011
3620 GOTO 3640 !149
3630 M=9 !013
3640 M$=STR$(M)!196
3650 GOSUB 2400 !185
3660 GOSUB 2040 !079
3670 RETURN !136
3680 FOR J=1 TO 4 !060
3690 FOR I=1 TO 10 !105
3700 CALL SOUND(75,F(I),0)!2
41
3710 NEXT I !223
3720 NEXT J !224
3730 RETURN !136
3740 FOR I=10 TO 1 STEP -1 !
215
3750 CALL SOUND(400,F(I),0)!
026
3760 NEXT I !223
3770 RETURN !136
3780 DATA 131,165,196,262,33
0,392,523,659,784,1047 !025
3790 DATA 4,9,4,14,4,19,9,9,
9,14,9,19,14,9,14,14,14,19 !
047
3800 FOR I=65 TO 90 !172
3810 CALL CHARPAT(I,B$)!062
3820 S$="0000"&SEG$(B$,1,4)&
SEG$(B$,7,4)&SEG$(B$,13,4)!0
58
3830 CALL CHAR(I+32,S$)!087
3840 NEXT I !223
3850 RETURN !136
3860 CALL CLEAR !209
3870 PRINT "I hope you enjoy
ed the game":"TIC-TAC-TOE (P
hiladelphia" !213
3880 PRINT "Style).  If so,
you may want":"to order the
disk CONEYGAMES" !176
3890 PRINT "from Barry Trave
r, 835 Green":"Valley Drive,
 Philadelphia," !063
3900 PRINT "PA 19128 ($10).
 Included on":"that disk are
 _eight_ games" !069
3910 PRINT "and winning stra
tegies for":"each:  3 differ
ent versions" !209
3920 PRINT "of the classic c
on game ""31""":"(using matc
hsticks, playing" !124
3930 PRINT "cards, or dice),
 ""The Game"":"of Gale"" (a.k
.a. ""Bridg-it"")" !100
3940 PRINT """Nimrow"" (chec
kerboard game)":"""Shutout,"
" and ""Tic-Tac-Toe" !153
3950 PRINT "(Philadelphia St
yle).""  All":"of these game
s are copyright" !146
3960 PRINT "by Barry Traver
(except for":"TICTAC/PHI, wh
ich I released" !210
3970 PRINT "into public doma
in).":"":"(Press any key to
conclude.)" !160
3980 CALL KEY(0,K,S):: IF S<
1 THEN 3980 ELSE CALL CLEAR
:: STOP !077
```

Erie 99ers members join umbrella group

The eight members of the former Erie 99ers user group have joined the Computer Users of Erie. Dave Howell, formerly of the Erie 99ers, is the CUE secretary. He notes, "When our membership dropped below 10, we decided to 'run for cover' rather than just drop 'out of sight' as happened to so many computer groups over the years.

According to On CUE, the group's newsletter, CUE has rewritten its constitution and by-laws to allow the group to accommodate different computer types.

Address for CUE is P.O. Box 1975, Erie, PA 16507-0975. Newsletters on the Erie 99ers mailing list should get copies of On CUE, but the mailing addresses for exchange newsletters to the group should be sent to Erie 99ers Department, Computer Users of Erie, 3904 Myrtle St., Erie, PA 16508-3016.

Email directory

Address list keeps TIers in touch in Cyberspace

By TOM WILLS

The following list of email addresses was compiled by Tom Wills, vice president and SysOp of the Southwest 99ers TI User Group, Tucson, Ariz. It was made available on Jan. 30. MICROpendium is publishing the list in the interest of expanding communications among TI users.—Ed.

The listing is organized so that the Internet address is followed by the name of the addressee in parenthesis. The listing includes addresses from North America, Europe and Australia. All are current or former TI users. There are 74 names.

Name	Address
Arnold, Oliver	oliver@thorin.swb.de
Begiers, Vincent	v.begiers@genie.geis.com
Bernasek, Glenn W.	dd314@cleveland.Freenet.Edu
Bishop, Gary D.	gdbishop@crems.rockwell.com
Bobbitt, Chris	72561.3241@compuserve.com
Boone, Barry	boone@galstar.com
Bowser, Gary	af670@torfree.net
Bray, Eric	ebray@pacs.pha.pa.us
Brent, Mike	mbrent@proton.com
Briscoe, John	john@wknott.lakes.trenton.sc.us
Butcher, Audrey	74367.3206@compuserve.com
Cavanaugh, James	j.cavanaugh2@genie.geis.com
Cox, Gary	cox@mbcf.stjude.org
Dehaney, Judy	ehaneyj@ersys.edmonton.ab.ca
Dorais, Lucie	l.dorais@genie.geis.com
Eggers, Steve	SEggers617@AOL.com
Gerut, Ron	rgerut@casbah.acns.nwu.edu
Gilpin, Robert D.	rdg@po.CWRU.Edu
Good, Charles	cgood@osulima1.lima.ohio-state.edu
Hayter, Steve	shayter@montreal.cbc.ca
Heckert, Curt	75136.1527@compuserve.com
Hoffman, Harrison W.	cz429@cleveland.Freenet.Edu
Horn, Jim	76702.603@compuserve.com
Jakabfy, Tom	tomjaka@village.ca
Knightes, Greg	gregrph@delphi.com
Koloen, John	jkoloen@io.com
Koplien, Henry	ti6hk@tick.ti6.tu-harburg.de
Kraemer, Markus	kraemer@cip.informatik.uni-wuerzburg.de
Kram, Chaim	chaimk@umich.edu
Krych, Jim	ab453@cleveland.freenet.edu
Kuehn, Gary	75227.2370@compuserve.com
Lanman, Jim	JIMLAN@delphi.com
Lucid, Bill	lucid@indy.net
Mac Leod, Peter	004352m@dragon.acadiau.ca
Mah, Dean	mah@cs.ualberta.ca
Mahler, Donald L.	d24@delphi.com
Mathis, BJ	b.mathis3@genie.geis.com
McGovern, Tony	phpam@cc.newcastle.edu.au
Meier, Roland	meier@rbg.informatik.th-darmstadt.de
Miller, Beery	bw.miller@genie.geis.com
Nieters, David	David.Nieters@seag.fingerhut.com
Nouspikel, Thierry	nouspike@cmu.unige.ch
O'Neil, Don	doneil@delphi.com
Olson, Erik G.	olsone@marcus.its.rpi.edu
Ormand, David	dlormand@CCGATE.HAC.COM
Reed, Tracy R.	treed@ucssun1.sdsu.edu
Reiss, Jim	jim@accelr8.com
Renda, Lee	ab594@yfn.ysu.edu
Rodenkirch, Bruce	bruce@marconi.w8upd.uakron.edu
Rottland, Joerg	rottland@fb9-ti.uni-duisburg.de
Schultz, Arne	Arne.Schulz@arbi.informatik.uni-oldenburg.de
Sheldon, Jeffery	jeff@tiger3.ocs.lsu.edu
Slicer, Shirley	SSLICER@delphi.com
Slomer, Howard	hms@sei.cmu.edu
Snider, Brian	brians@ibmoto.com
Snyder, Brad	bradsnyder@delphi.com
Squires, Burke	BurkeS@eWorld.com
Stasiowski, Scott	Scott@wknott.lakes.trenton.sc.us
Swartz, Edward	swartze@ralph.txswu.edu
Tesch, Tim	t.tesch1@genie.geis.com
Tippett, Larry	chatter@delphi.com
Tipton, David	6500dtpt@ucsbuxa.ucsb.edu
Traver, Barry	GENIAL.AL@GENIE.GEIS.COM
Trott, Geoff	Geoff_Trott@uow.edu.au
Turner, Dee	gdturner@ix.netcom.com
Underwood, George	gunderwo@sun.cis.smu.edu
Wacholtz, Mark	mark@wknott.lakes.trenton.sc.us
Webster, Andrew	andy-w@worldgate.edmonton.ab.ca
White, Jeff	jhwhite@delphi.com
Wills, Tom	twills@primenet.com
Wilson, Kent	kwilson@crosby.phx.mcd.mot.com
Wright, Mike	mjmw@xyvision.com
Zapf, Michael	zapf@rbi.informatik.uni-frankfurt.de

TIPS ON SENDING MESSAGES

To send messages from one online service to abother, read the following. But, be aware that only Compuserve, Delphi, GEnie and Internet are included. Services such as MCI, BIX, Prodigy, and America On-line are not included.

From CompuServe

TO Delphi : >internet:userlogon@delphi.com
Genie : >internet:userlogon@genie.geis.com
Internet : >internet:username@internet.domain

From Delphi

TO Compuserve: internet"user numeric id@compuserve.com"
Genie : internet"userlogon@genie.geis.com"
Internet : internet"username@internet.domain"

From GEnie

TO Compuserve: numericID@compuserve.com@inet#

(See Page 18)

INTERNET ADDRESSES—

(Continued from Page 17)

Delphi : userlogon@@delphi.com@inet#
Internet : username@internet.domain@inet#

From Internet

TO Compuserve: numericDI@compuserve.com
Delphi : userlogon@delphi.com
Genie : userlogon@genie.geis.com

NOTES

When using the CompuServe ID number, substitute a period for the comma. For example, if you were to send a message to me on Compuserve (my ID is 71550,3213), you would send a message to me as 71550.3213@compuserve.com.

When sending a message to the Internet from Compuserve, if you are using Wincim or Doscim, omit the ">" from the address.

If a user has more than one address, which is not uncommon (I have four, of which one is a CompuServe address), use a non-CompuServe address unless the user says otherwise. The reason for this is that CompuServe charges a small fee for receiving Internet mail. There is no charge for sending Internet mail.

DISCLAIMER

I have tried to verify all the addresses listed above. This was done by sending messages to all the addresses I was able to locate. I did not get replies from all the users in the list, but neither did any of the addresses reject. Some of the addressees listed above may not access their mail very often, and replies will be very slow in coming. Also, some addresses may become obsolete as the addressee drops that service provider and goes to another service. I will try to keep the list as current as possible.

ADDITIONS, CHANGES, AND UPDATES

If there are any changes to the above address list, or someone wants to add their name to the list, please contact me at twills@primenet.com. Include your Internet mailing address along with your first and last name, or that of someone who you know who also wants to be on the list.

Thanks to all who have given me their address. I hope this proves to be a good way, and cheap way, for TIers to keep in touch with each other. I put no claims on this list, which means you can use it as you want. Things likes TI fair announcements would be an example. But, please, do not abuse this listing.

Until the next update, keep TI'ing!

Peterson disks at Lima MUG conference

The Central Ohio Ninety-Niners (CONNI) are scheduled to have a complete set of the Jim Peterson public domain software library and three computers available for copying this material at the Lima Multi User Group Conference scheduled April 29.in Lima, Ohio.

The conference, sponsored by the Lima 99/4A Users Group, will be held at Reed Hall at the Ohio State University in Lima and is free to vendors and attendees.

According to the Lima group's newsletter, anyone who brings his own disks can copy anything from the Peterson library at no charge during the show. The only limitations are time (show hours only) and the physical setup of one set of master disks and three copy stations.

The library includes all of Peterson's software that has been released to the public domain as well as hundreds of disks of public domain software from other sources. Most is in SSSD format.

After the conference, CONNI intends to donate the set of Peterson disks to the Lima User Group, so the group has requested its members to order Peterson library disks after the conference, to give copying time to non-members.

Peterson was a prolific author of TI programs who distributed them through his company, Tigercub Software, for many years. He died Jan. 12, 1994.

The Lima user group will have all its library's software added since last year's conference available for free copying, also. According to the group's newsletter, this material includes about 150 disks, mostly in DSSD format. Some contain public domain versions of classic books, including the complete works of William Shakespeare.

For further information, contact the Lima Users Group, P.O. Box 647, Venedocia OH 45894, or call Charles Good (evenings) at (419) 667-3131 or Internet cgood@lima.ohio-state.edu.

READER TO READER

❑ Oscar A. Ros, 14007 Hubbard St., Sylmar, CA 91342, (818) 362-6387, writes:

I would like James W. Krych's address and phone and I would like J&KH Software or James B. Hollender's address and phone. I would also like to know if anyone has information on TI half-bit graphics.

Reader to Reader is a column to put TI and Geneve users in contact with other users. Address questions to *Reader to Reader, c/o MICROpendium, P.O. Box 1343, Round Rock, TX 78680.* We encourage those who answer the questions to forward us a copy of the reply to share with readers.

Firm offers recycled products

Quill Corporation has released its latest 26-page Recycled Products Catalog, which lists more than 370 different items.

According to the company, these include paper products, such as laser printer paper, multipart forms, file folders and storage boxes, as well as printer cartridges, desk protectors, binders and packing materials.

The firm strives to offer only those recycled products that contain at least 50 percent recycled material with a guaranteed minimum of at least 10 percent post-consumer recycled content, according to a company news release. For a catalog, send your name and address on company letterhead to Quill Corporation, Public Relations Dept. #95171, 100 Schelter Rd., Lincolnshire, IL 60069-3621.

Freeloader III

Not your average load program

Freeloader III was written by Karl Romstedt and is in the public domain. Send comments, programs or contributions to Karl Romstedt, 2543 Cranford Rd., Columbus, OH, 43221-1105.

An "X" at the end of a file name indicates an object file designed to be used from Extended BASIC. The source files are documented and end with "-S". The BASIC programs begin with "USE" and illustrate the use of each object file.

Freeloader III is a colorful Extended BASIC load program that contains object code embedded by Todd Kaplan's ALSAVE program. The source code is called LOADX-S. To use it, save it under the name LOAD. It will automatically be loaded when you execute Extended BASIC.

There are lots of load programs around, but you're sure to like this one! It can shift directly to any page of a disk's programs and selected files are highlighted using inverse video. A touch of one key selects any drive or it defaults to drive 1 after about five seconds. It can run Extended BASIC programs, read D/V80 files, delete and print catalogs. The read function uses a 40-column screen and will properly parse 80-column text. Freeloaded III also restores standard colors and characters (except lower case) prior to running an Extended BASIC program. All functions are designed for ease of use. Press AID (FCTN-7) for instructions.

```
100 CALL INIT :: CALL LOAD(8
196,63,248):: CALL LOAD(1637
6,65,32,32,32,32,32,255,48):
: CALL LINK("A"):: CALL CHAR
(127,"0")!113
110 P$="PIO" :: PC$=CHR$(15)
&CHR$(27)&"S0"&CHR$(27)&"A"&
CHR$(6)!PRINTER & CODES !228
120 DATA DIS/FIX,DIS/VAR,INT
/FIX,INT/VAR,PROGRAM !116
130 DIM A$(127),C$(127),E(12
7):: FOR I=1 TO 5 :: READ T$
(I):: NEXT I :: CALL SCREEN(
2):: CALL LINK("COLORS",1,12
,16,5)!057
140 ER,N,PG,TS=0 :: DISPLAY
AT(6,7)ERASE ALL BEEP:"Free
Loader III": : : :TAB(11);"
Drive ?" !by Karl Romstedt,
10/89 !187
150 DISPLAY AT(23,5):"AID (F
CTN-7) Active" !217
160 FOR I=1 TO 99 :: CALL KE
Y(0,K,S):: IF S>0 THEN 180 !
246
170 NEXT I !223
180 IF K=1 THEN 420 ELSE IF
K<49 OR K>55 THEN K=49 !212
190 D$="DSK"&CHR$(K)&"." ::
DISPLAY AT(10,17):K-48 !079
200 ON ERROR 560 :: OPEN #1:
D$,INPUT ,RELATIVE,INTERNAL
:: INPUT #1:N$,J,T,F !051
210 FOR I=1 TO 127 :: INPUT
#1:A$(I),A,E(I),S :: IF LEN(
A$(I))=0 THEN 250 !179
220 K=ABS(A):: S$=T$(K):: L$
=STR$(E(I)):: S$=RPT$(" ",4-
LEN(L$))&L$&" "&S$ :: IF K<>
5 THEN L$=STR$(S):: S$=S$&RP
T$(" ",4-LEN(L$))&L$ !019
230 IF A<0 THEN S$=S$&RPT$("
",16-LEN(S$))&" P" !023
240 C$(I)=S$ :: N=N+1 :: NEX
T I !074
250 CLOSE #1 :: ON ERROR STO
P :: M=MIN(1,N):: PP=INT(N/1
8.01):: S1=1 :: S2=MIN(18,N)
!189
260 TS=1 :: CALL CLEAR :: CA
LL LINK("NUMCOP"):: CALL CHA
R(141,"0"):: CALL LINK("COLO
RS",1,8,16,5):: CALL LINK("C
OLORS",9,12,2,4)!140
270 CALL LINK("COLORS",13,14
,2,9):: CALL HCHAR(1,1,141,9
6)!047
280 DISPLAY AT(2,1):N$;:: GO
SUB 390 !219
290 CALL LINK("CLRWIN"):: FO
R S=S1 TO S2 :: K=S+4-PG*18
:: DISPLAY AT(K,1):A$(S);TAB
(11);C$(S):: NEXT S !070
300 IF N>0 THEN DISPLAY AT(2
4,1):"PAGE";PG+1;"OF";PP+1;"
- DIGIT FOR PAGE" :: GOSUB 4
00 !203
310 IF N=0 THEN DISPLAY AT(2
4,1):"NO FILES. PRESS REDO O
R AID." !223
320 CALL KEY(0,K,S):: IF S=0
THEN 320 ELSE IF K=1 THEN 4
20 ELSE IF K=13 THEN 600 ELS
E IF K=144 THEN 410 !064
330 IF K>48 AND K<50+PP THEN
PG=K-49 :: S1=PG*18+1 :: S2
=MIN(S1+17,N):: M=MIN(S1,N):
: GOTO 290 !076
340 IF K=6 THEN CALL LINK("C
OLORS",1,12,16,5):: GOTO 140
!094
350 IF K=132 THEN 450 !042
360 DISPLAY AT(M+4-PG*18,1):
A$(M);:: IF K=11 OR K=69 OR
K=101 THEN M=M-2 :: IF M<S1-
1 THEN M=S2-1 !248
370 M=M+1 :: IF M>S2 THEN M=
S1 !234
380 GOSUB 400 :: GOTO 320 !2
43
390 F$="FREE "&STR$(F)&"/"&S
TR$(T):: DISPLAY AT(2,29-LEN
(F$)):F$ :: RETURN !122
400 CALL LINK("LOWER",A$(M),
Y$):: DISPLAY AT(M+4-PG*18,1
):Y$;:: RETURN !211
410 OPEN #1:P$ :: PRINT #1:P
C$;RPT$("*",28):N$;TAB(29-LE
N(F$));F$:RPT$("*",28):: FOR
S=1 TO N :: PRINT #1:A$(S);
TAB(11);C$(S):: NEXT S :: PR
INT #1 :: CLOSE #1 :: GOTO 3
20 !210
420 DISPLAY AT(5,1):"DIGITS-
SELECT PAGE": :"ARROWS SEL
ECT FILE": :"ENTER RUNS X
B PROGRAM":TAB(9);"DISPLAYS
DV80 FILE": :!255
430 DISPLAY AT(12,1):"CTRL-P
PRINTS CATALOG":TAB(9);"(C
ODES IN LINE 110)": :"CTRL-D
DELETES UNPRO. FILE": :"FC
TN-8 CATALOGS NEW DISK": :
: : : :"PRESS ANY KEY" !24
2
440 CALL KEY(0,K,S):: IF S<1
THEN 440 ELSE IF TS=0 THEN
140 ELSE 290 !212
450 IF LEN(C$(M))>16 THEN CA
LL SOUND(99,222,0):: GOTO 32
0 !246
```

(See Page20)

FREELOADER III —

(Continued from Page 19)

```
460 F=F+E(M):: DELETE D$&A$(
M):: N=N-1 :: FOR K=MAX(M,1)
TO N :: J=K+1 :: A$(K)=A$(J)
:: C$(K)=C$(J):: E(K)=E(J)::
 NEXT K :: PP=INT(N/18.01)::
 K=MIN(PG,PP)+49 :: GOSUB 39
0 :: GOTO 330 !156
470 CALL LINK("UNCOP"):: ER=
2 :: ON ERROR 570 :: OPEN #2
:D$&A$(M),INPUT :: ON ERROR
STOP !154
480 CALL CLEAR :: CALL LINK(
"TEXT40"):: L$="Press any ke
y to scroll text (Ctrl-C to
cancel)." :: GOSUB 550 :: GO
SUB 540 :: GOSUB 540 :: GOTO
 500 !182
490 LINPUT #2:L$ :: IF EOF(2
)<>0 THEN 510 ELSE GOSUB 550
 !232
500 CALL KEY(0,K,S):: IF K=1
31 THEN 530 ELSE IF S=0 THEN
 500 ELSE 490 !105
510 GOSUB 540 :: GOSUB 540 :
: L$="FINISHED. Ctrl-C TO AB
ORT." :: GOSUB 550 !015
520 CALL KEY(0,K,S):: IF S=0
 OR K<>131 THEN 520 !050
530 CLOSE #2 :: CALL LINK("T
EXT32"):: GOTO 260 !107
540 L$=" " !022
550 CALL LINK("PARSE",L$)::
RETURN !120
560 CALL ERR(K,S):: ON ERROR
 570 :: CLOSE #1 !239
570 CALL ERR(K,S):: DISPLAY
AT(10,2)ERASE ALL:"DISK ERRO
R. PRESS ANY KEY." :: GOSUB
590 !235
580 CALL KEY(0,K,S):: IF S<1
 THEN 580 ELSE IF ER=2 THEN
RETURN 260 ELSE RETURN 140 !
069
590 CALL SOUND(99,222,0):: R
ETURN !102
600 IF POS(C$(M),T$(2)&"  80
",5)>0 AND E(M)>1 THEN 470 !
215
610 IF POS(C$(M),T$(5),5)+PO
S(C$(M),T$(4)&" 254",5)=0 TH
EN GOSUB 590 :: GOTO 320 !24
7
620 CALL SCREEN(8):: CALL CL
EAR :: CALL CHARSET :: D$=D$
&A$(M):: PRINT "Running ";D$
 :: CALL LINK("UNCOP")!172
630 CALL PEEK(-31952,K,S)::
CALL PEEK(K*256+S-65534,K,S)
:: S1=K*256+S-65534 !023
640 CALL INIT :: S2=LEN(D$):
: CALL LOAD(S1,S2):: FOR M=1
 TO S2 :: CALL LOAD(S1+M,ASC
(SEG$(D$,M,1)))):: NEXT M ::
CALL LOAD(S1+M,0)!004
650 RUN "DSK1.0123456789" !2
05
```

FREELOADER SOURCE CODE

```
0B0FAx0001  *BY KARL ROMSTEDT, 10/89 FOR USE WITH FREELOADER
III XB PROGRAM.
0002
0003        DEF  LOWER,COLORS,CLRWIN,PARSE
0004        DEF  TEXT40,TEXT32,NUMCOP,UNCOP
0005  STRASG EQU  >2010       STRING ASSIGNMENT UTILITY (SEND
STRING TO BASIC)
0006  STRREF EQU  >2014       STRING REFERENCE (GET STRING FROM
BASIC)
0007  GPLWS  EQU  >83E0       GPL WORKSPACE
0008  STATUS EQU  >837C       GPL STATUS BYTE
0009  VSBW   EQU  >2020       DEFINE LOCATION OF UTILITIES,
ETC. (NO REF'S ALLO
WED)
0010  VMBW   EQU  >2024       VDP MULTI BYTE WRITE
0011  VMBR   EQU  >202C       VDP MULTI BYTE READ
0012  VWTR   EQU  >2030       VDP WRITE REGISTER
0013  NUMREF EQU  >200C       NUMERIC REFERENCE (GET NUMBER
FROM BASIC)
0014  FAC    EQU  >834A       FLOATING POINT ACCUMULATOR FOR
PASSING NUMBERS
0015  COLTBL EQU  >080F       LOCATION  OF XB COLOR SET 0 FOR
CHARACTERS 30 & 3
1
0016
0017  MYREG  BSS  >20         REGISTER SPACE
0018  SAV11  BSS  2           SAVE RETURN
0019  INPUT  BSS  82          INPUT FROM OR OUTPUT TO BASIC +
LENGTH BYTE
0020  SHIFT  BSS  40          BUFFER FOR SCREEN SCROLL
0021  SAV32  BSS  192         SAVE 32 COLUMN SCREEN AREA ERASED
BY 40 COLUMN SC
REEN
0022  FIRST  BSS  1           FIRST COLOR SET TO CHANGE
0023  LAST   BSS  1           LAST  COLOR SET TO CHANGE
0024  FCOL   BSS  1           FOREGROUND COLOR
0025  BCOL   BSS  1           BACKGROUND COLOR
0026        EVEN
0027  F2     BSS  2           ADDRESS OF FIRST SET TO CHANGE
0028  L2     BSS  2           ADDRESS OF LAST  SET TO CHANGE
0029  LCASE  BYTE 32          OFFSET FOR CONVERSION OF UPPER TO
LOWER CASE
0030  LOLIM  BYTE 65          ASCII FOR "A"
0031  HILIM  BYTE 90          ASCII FOR "Z"
0032  LCASE2 BYTE 80          OFFSET FOR CONVERSION OF "0"-"?"
0033  LOLIM2 BYTE 48          ASCII FOR "0"
0034  HILIM2 BYTE 63          ASCII FOR "?"
0035  DEL    BYTE 127
0036        EVEN
0037
0038  *CHANGE UPPER TO LOWER CASE
0039
0040  LOWER  LIMI 0           DISABLE INTERUPTS
0041        MOV  R11,@SAV11   SAVE RETURN LOCATION
0042        LWPI MYREG        SET WORKSPACE POINTER
0043        BL   @GETSTR      GET INPUT STRING FROM BASIC
0044
0045        CLR  R5           RESET R5 (BYTE POINTER WITHIN
STRING)
0046  NEXT1  INC  R5          LOOK AT NEXT BYTE
0047        C    R5,R6        FINISHED WITH ALL BYTES?
0048        JH   OUTPUT       IF SO, SEND RESULT TO BASIC
0049        CB   @INPUT(5),@LOLIM  IS CHARACTER < "A" ASCII
0050        JL   NUMBER       IF SO, NEXT CHARACTER BYTE
0051        CB   @INPUT(5),@HILIM  IS CHARACTER > "Z" ASCII
0052        JH   NUMBER       IF SO, NEXT CHARACTER
0053        AB   @LCASE,@INPUT(5)  SHIFT CHARACTER FROM UPPER
TO LOWER CASE
0054        JMP  NEXT1        GOTO NEXT CHARACTER IN STRING
0055
0056  NUMBER CB   @INPUT(5),@LOLIM2 IS CHARACTER < "0" ASCII
0057        JL   DELB         IF SO, NEXT CHARACTER BYTE
0058        CB   @INPUT(5),@HILIM2 IS CHARACTER > "?" ASCII
0059        JH   DELB         IF SO, NEXT CHARACTER
0060        AB   @LCASE2,@INPUT(5) SHIFT CHARACTER
```

(See Page 21)

FREELOADER III—

(Continued from Page 20)

```
0061         JMP  NEXT1          GOTO NEXT CHARACTER IN STRING
0062  DELB   MOVB @DEL,@INPUT(5)    CHANGE TO CHR$(127)
0063         JMP  NEXT1
0064
0065  OUTPUT INC  R1             PASS TO SECOND VARIABLE
0066         BLWP @STRASG        RETURN STRING TO BASIC
0067         JMP  QUIT2
0068
0069  GETSTR LI   R1,>5000       PUT 80 INTO MSB OF R1
0070         MOVB R1,@INPUT      SET MAXIMUM STRREF INPUT LENGTH
TO 80
0071         CLR  R0             CLEAR ELEMENT POINTER
0072         LI   R1,1           PREPARE TO PASS FIRST VARIABLE
0073         LI   R2,INPUT       POINTER TO CPU RAM BUFFER FOR BA-
SIC STRING
0074         BLWP @STRREF        GET STRING FROM BASIC
0075  GS1    MOVB @INPUT,R6      PUT LENGTH IN MSB OF R6
0076         JEQ  QUIT2          ABORT IF LENGTH IS 0
0077         SRL  R6,8           PUT LENGTH IN LSB; CLEAR MSB (WORD
AT R6 NOW = LE
NGTH)
0078         RT                  RETURN FROM SUBROUTINE
0079
0080  PARSE  LIMI 0              DISABLE INTERUPTS
0081         MOV  R11,@SAV11     SAVE BASIC RETURN ADDRESS
0082         LWPI MYREG          CHANGE WORKSPACE POINTER
0083         BL   @GETSTR        GET INPUT FROM BASIC
0084
0085  *CHECK FOR "." IN FIRST POSITION
0086
0087         LI   R3,>2E00       PUT ASCII FOR "." IN MSB OF R3
0088         CB   R3,@INPUT+1    IS FIRST CHARACTER "."?
0089         JEQ  QUIT2          IF SO, QUIT
0090
0091  *CHECK FOR CHR$(13) AT END
0092
0093         CLR  R8             R8 IS PRINT FLAG
0094         LI   R3,>0D00       PUT HEX FOR CHR$(13) IN MSB OF R3
0095         CB   R3,@INPUT(6)   IS LAST CHARACTER CHR$(13)?
0096         JNE  P1             IF NOT, GO TO P1
0097         DEC  R6             DECREMENT R6 (INPUT LENGTH)
0098         JEQ  EXITP          IF R6 = 0, QUIT
0099
0100  *ADD BASIC OFFSET TO ALL CHARACTERS
0101
0102  P1     LI   R3,>6000       PUT OFFSET IN MSB OF R3
0103         MOV  R6,R4          PUT INPUT LENGTH IN R4
0104  P2     AB   R3,@INPUT(4)   ADD OFFSET TO POSITION R4 OF IN-
PUT
0105         DEC  R4             DECREMENT R4
0106         JNE  P2             IF R4 <> 0, GO TO P2
0107
0108  *PARSE INPUT ONTO 40 COLUMN SCREEN LINES
0109
0110         LI   R0,920         VDP SCREEN ADDRESS FOR 1ST COL OF
LAST LINE FOR V
SBW
0111         LI   R3,>8000       ASCII + OFFSET FOR SPACE IN MSB
0112         CLR  R4             R4 IS PRESENT INDEX POSITION
0113         INC  R6             MAKE R6 INDEX FOR LAST CHARACTER IN
BUFFER + 1
0114  P7     INC  R4             INCREMENT R4 TO NEXT CHARACTER ON
RIGHT
0115         C    R4,R6          IS R1 > LAST CHARACTER POSITION
0116         JHE  EXITP          IF YES, REST OF LINE IS BLANK, SO
QUIT
0117         CB   R3,@INPUT(4)   IS THIS CHARACTER A BLANK?
0118         JEQ  P7             IF SO, GO TO P7 TO LOOK AT NEXT
CHARACTER
0119         BL   @SCROLL        SCROLL SCREEN
0120  P3     LI   R1,INPUT       R1 IS CPU RAM START FOR VMBW
0121         A    R4,R1          ADD INDEX TO R1 FOR START OF PART
OF BUFFER TO WR
ITE
0122         MOV  R6,R2          PUT END OF LINE + 1 IN R2
0123         S    R4,R2          CALCULATE REMAINING LENGTH OF
BUFFER (-1)
0124         CI   R2,40          IS LENGTH IN R2 > 40?
0125         JH   P4             IF SO, GO TO P4
0126         BLWP @VMBW          PRINT REMAINDER OF LINE
0127         JMP  QUIT2          EXIT
0128
0129  P4     LI   R2,40          PUT MAXIMUM LENGTH FOR VMBW IN R2
0130         AI   R4,39          R4 IS INDEX FOR END OF BUFFER PART
TO WRITE
0131  P5     CB   R3,@INPUT(4)   IS CURRENT CHARACTER A BLANK?
0132         JEQ  P6             IF SO, GO TO P6 TO PRINT LINE
0133         DEC  R4             PREPARE TO LOOK AT CHARACTER TO
LEFT
0134         DEC  R2             DECREMENT LENGTH OF LINE TO SEND
0135         JNE  P5             IF LENGTH NOT 0, GO TO P5
0136         LI   R2,40          NO SPACES, SO SEND LINE AS IS
(LENGTH = 40)
0137         AI   R4,40          NO SPACES, RESET R4 TO END OF CUR-
RENT LINE
0138  P6     BLWP @VMBW          PRINT LINE TO BOTTOM OF SCREEN
0139         INC  R8             SET PRINT FLAG
0140         JMP  P7             GO TO P3 TO PRINT MORE OF CPU INPUT
BUFFER
0141
0142  SCROLL LI   R0,40          L1 IS VDP ADDRESS OF 1ST POSTION
OF 2ND SCREEN LI
NE
0143         LI   R1,SHIFT       R1 WILL USE CPU SHIFT BUFFER FOR
VMBR AND VMBW
0144         LI   R2,40          R2 IS NUMBER OF BYTES TO READ OR
WRITE
0145  S1     BLWP @VMBR          READ LINE FROM VDP SCREEN
0146         AI   R0,-40         POINT ONE LINE HIGHER ON SCREEN
0147         BLWP @VMBW          WRITE SHIFT BUFFER TO NEW LINE
```

(See Page 22)

1995 TI FAIRS

FEBRUARY

Fest West '95, Feb. 18, Fabulous Inn, San Diego, California. Contact Southern California Computer Group, P.O. Box 152535, San Diego,CA 92195, or call the SCCG BBS, (619) 263-9135, User No. 25, password FEST

APRIL

Lima Multi Users Group Conference, April 29, Reed Hall, Ohio State University at Lima. Contact Lima Users Group, P.O. Box 647, Venedocia OH 45894, or call Charles Good (evenings) at (419) 667-3131 or Internet cgood@lima.ohio-state.edu.

SEPTEMBER

10th International TI-Meeting, Sept. 22-24, Wohlfahrtsgebäude der Wiener E-Werke (Welfare Building of the Vienna Electricity Board), Wachaustr. 28, A-1020 Vienna, Austria. For information write Kurt Radowisch, TI- and Geneve User Group Vienna, Fugbachgasse 18/17, A-1020 Vienna, Austria.

This TI event listing is a permanent feature of MICROpendium. User groups and others planning events for TI/Geneve users may send information for inclusion in this standing column. Send information to MICROpendium Fairs, P.O. Box 1343, Round Rock, TX 78680.

FREELOADER III—

(Continued from Page 21)

```
0148          AI   R0,80         POINT TO NEXT LOWER SCREEN LINE
0149          CI   R0,959        HAVE WE PASSED THE LAST SCREEN
LINE?
0150          JL   S1            IF NOT, GO TO S1
0151          LI   R0,959        PUT LAST VDP SCREEN ADDRESS IN R0
FOR VSBW
0152          LI   R1,>8000      SPACE + OFFSET FOR VSBW TO CLEAR
LAST LINE OF SCR
EEN
0153  S2      BLWP @VSBW         PRINT SPACE
0154          DEC  R0            DECREMENT SCREEN ADDRESS
0155          CI   R0,920        FINISHED WITH LAST LINE?
0156          JHE  S2            IF NOT, GO TO S2
0157          INC  R0            RESET R0 TO FIRST POSITION OF LAST
LINE
0158          RT                 RETURN
0159
0160  EXITP   MOV  R8,R8         IS R8 0?
0161          JNE  QUIT2         NO, EXIT
0162          BL   @SCROLL       PRINT BLANK LINE
0163
0164  QUIT2   B    @QUIT
0165
0166  COLORS LIMI 0              DISABLE INTERUPTS
0167          MOV  R11,@SAV11    SAVE RETURN LOCATION
0168          LWPI MYREG         SET WORKSPACE POINTER
0169
0170  *       ACCESS VALUES FROM XB
0171
0172          CLR  R0            NO DIMENSION ELEMENT
0173          LI   R1,1          LOOK AT FIRST PASSED VALUE/VARI-
ABLE
0174  NEXT2   BLWP @NUMREF       PUT VALUE IN FAC
0175          MOV  @FAC,R3       GET VALUE
0176          ANDI R3,>00FF      DISCARD EXPONENT
0177          SWPB R3            PUT VALUE IN MSB
0178          DEC  R1            SUBTRACT 1 FROM R1 AS INDEX
0179          MOVB R3,@FIRST(1) SAVE VALUE TO FIRST + R1 RAM LO-
CATION
0180          INCT R1            ADD 2 TO R1
0181          CI   R1,5          HAVE ALL VALUES BEEN PASSED?
0182          JL   NEXT2
0183
0184          LI   R3,>0100
0185          SB   R3,@FCOL      CONVERT FROM XB COLOR (1-16) TO EA
(0-F)
0186          SB   R3,@BCOL      CONVERT FROM XB COLOR TO EA
0187          MOVB @FCOL,R1      MOVE FCOL BYTE TO MSB OF R1
0188          SLA  R1,4          CHANGE FCOL BYTE FROM 0? TO ?0 IN
R1
0189          AB   @BCOL,R1      COMBINE COLORS INTO SINGLE BYTE IN
R1
0190
0191          CLR  @F2           ZERO F2 SINCE MOVB DOES NOT AFFECT
LSB
0192          CLR  @L2           SAME FOR L2
0193          LI   R3,COLTBL
0194          MOVB @FIRST,@F2    MOVE FIRST SET TO MSB OF F2
0195          SWPB @F2           SWAP FIRST SET TO LSB OF F2
0196          A    R3,@F2        STORE ADDRESS OF FIRST SET TO
CHANGE IN F2
0197          MOVB @LAST,@L2     MOVE LAST  SET TO MSB OF L2
0198          SWPB @L2           SWAP LAST  SET TO LSB OF L2
0199          A    R3,@L2 .      STORE ADDRESS OF LAST  SET TO
CHANGE IN L2
0200
0201
0202          MOV  @F2,R0        POINT TO FIRST COLOR SET TO CHANGE
0203  NEXT3   BLWP @VSBW         WRITE COLOR CHANGE TO VDP RAM
COLOR TABLE
0204          INC  R0            POINT TO NEXT COLOR SET
0205          C    R0,@L2        HAS LAST DESIRED SET BEEN CHANGED?
0206          JLE  NEXT3
0207          JMP  QUIT
0208
0209  CLRWIN LIMI 0              DISABLE INTERUPTS
0210          MOV  R11,@SAV11    THIS PART CLEARS SELECTED PORTION
OF SCREEN
0211          LWPI MYREG
0212
0213          LI   R0,129        STARTING LOCATION TO CLEAR
0214          LI   R1,>8000      MSB IS SPACE + BASIC OFFSET
0215  NEXT4   BLWP @VSBW
0216          INC  R0
0217          CI   R0,701        LAST LOCATION TO CLEAR
0218          JLE  NEXT4
0219          JMP  QUIT
0220
0221  TEXT40 LIMI 0              DISABLE INTERUPTS
0222          MOV  R11,@SAV11    SAVE BASIC RETURN ADDRESS
0223          LWPI MYREG         SET CPU WORKSPACE ADDRESS
0224
0225          BL   @T1           SET UP TO READ END OF TEXT MODE
SCREEN
0226          BLWP @VMBR         STORE END OF TEXT MODE SCREEN AREA
0227
0228          LI   R0,959        LOAD WORKSPACE REGISTER 0 WITH 767
(LAST SCREEN L
OC)
0229          LI   R1,>8000      LOAD ASCII + BASIC OFFSET FOR
SPACE INTO R1 FIRST
 BYTE
0230  CL      BLWP @VSBW         WRITE BLANK TO VDP SCREEN ADDRESS
0231          DEC  R0            DECREMENT R0
0232          JNE  CL            JUMP TO CLEAR IF EQUAL BIT OF STA-
TUS REGISTER NOT
 ZERO
0233          BLWP @VSBW         WRITE SPACE TO FIRST LOCATION (0)
0234
0235
0236          LI   R0,>0713      SET TEXT COLORS IN VDP REG 7 TO
BLACK ON CYAN
0237          BLWP @VWTR
0238
0239          LI   R0,>F001      PUT COPY OF NEW VDP REG 1 CONTENTS
IN >83D4
0240          MOVB R0,@>83D4
0241
0242          SWPB R0            SET VDP REG 1 TO TEXT MODE
0243          BLWP @VWTR
0244
0245          JMP QUIT
0246
0247  T1      LI   R0,768        END OF 32 COLUMN SCREEN
0248          LI   R1,SAV32      CPU AREA TO SAVE END OF 40 COLUMN
SCREEN (SPRITE
AREA)
0249          LI   R2,192        LENGTH OF PART OF EXTRA AREA OVER-
WITTEN BY TEXT M
ODE
0250          RT                 RETURN
0251
0252  TEXT32 LIMI 0              DISABLE INTERUPTS
0253          MOV  R11,@SAV11    SAVE BASIC RETURN ADDRESS
0254          LWPI MYREG         SET CPU WORKSPACE ADDRESS
0255
0256          BL   @T1           RESTORE SPRITE TABLES
0257          BLWP @VMBW
0258
0259          LI   R0,>0711      SET SCREEN COLORS IN VDP REG 7 TO
BLACK ON CYAN
0260          BLWP @VWTR
0261.
0262          LI   R0,>E001      PUT COPY OF NEW VDP REG 1 CONTENTS
```

(See Page 23)

FREELOADER III—

(Continued from Page 22)

```
IN >83D4
0263        MOVB R0,@>83D4
0264
0265        SWPB R0           SET VDP REG 1 TO TEXT MODE
0266        BLWP @VWTR
0267
0268
0269  QUIT   LWPI GPLWS        RESET WORKSPACE FOR GPL
0270        MOV  @SAV11,R11   PUT RETURN LOCATION IN R11
0271        CLR  @STATUS      CLEAR GPL STATUS BYTE
0272        RT                RETURN TO XBASIC
0273
0274  NUMCOP LIMI 0           DISABLE INTERUPTS
0275        MOV  R11,@SAV11   SAVE BASIC RETURN ADDRESS
0276        LWPI MYREG        SET CPU WORKSPACE ADDRESS
0277
0278        LI   R0,>480      VDP ADDRESS FOR VMBR
0279        LI   R1,INPUT     CPU DESTINATION ADDRESS FOR VMBR
0280        LI   R2,64        NUMBER OF BYTES TO TRANSFER
0281        BLWP @VMBR        READ CHARPATS FOR ASCII 48-55
0282
0283        LI   R0,>700      COPY CHARPATS TO ASCII 128-135
0284        BLWP @VMBW
0285
0286        LI   R0,>4C0      READ CHARPATS FOR ASCII 56-63
0287        BLWP @VMBR
0288
0289        LI   R0,>740      COPY CHARPATS TO ASCII 136-143
0290        BLWP @VMBW
0291
0292        JMP  QUIT
0293
0294  UNCOP LIMI 0            DISABLE INTERUPTS
0295        MOV  R11,@SAV11   SAVE BASIC RETURN ADDRESS
0296        LWPI MYREG        SET CPU WORKSPACE ADDRESS
0297
0298        LI   R0,>700      START OF CHARPAT FOR ASCII 128
0299        CLR  R1
0300  UN1   BLWP @VSBW        CLEAR CHARPATS ABOVE 128
0301        INC  R0           INCREMENT R0
0302        CI   R0,>780      END OF CHARPAT TABLE?
0303        JL   UN1
0304        JMP  QUIT
0305        END
```

Extended BASIC

A program that writes sentences

By W. LEONARD TAFFS

Taffs' column, Feedforth, appears regularly in the newsletter of the SouthWest 99ers of Tucson, Arizona.—Ed.

Here is a program that could be creative fun for those so inclined. What it does is construct a sentence composed of randomly selected phrases. The results can be very funny if you plan your data in the right order. (Space limitations for this article oblige me to leave out the data information needed to run this program.) Once you understand what is needed, all you have to do is exercise your creative ability to create phrases, or select random passage phrases from any book that happens to be lying around.

First, the program, and then some explanation. Some of these lines were excerpted from a program I copied years ago, but unfortunately I did not make note of what program they were extracted from. There are several similar type programs such as "JIBBERISH" that appeared in the September 1985 issue of Suncoast Beeper's newsletter.

SENTENCE

```
10 REM RANDOM SENTENCE GENER
ATOR !183
20 RANDOMIZE !149
30 DIM X$(40)!161
40 FOR I=1 TO 40 :: READ X$(
I):: NEXT I !024
50 INPUT "# OF PARAGRAPHS: "
:P !170
60 INPUT "# OF LINES PER PAR
A: ":S !142
70 CALL CLEAR !209
80 FOR I=1 TO P !142
90 PRINT " ";!056
100 FOR J=1 TO S !146
110 A=INT(RND*10+1):: B=INT(
RND*10+11):: C=INT(RND*10+21
):: D=INT(RND*10+31)!017120
PRINT X$(A);X$(B);X$(C);X$(D
);" " !002
130 NEXT J :: PRINT !254
140 NEXT I !223
150 REM DATA "EXAMPLE ","Sec
ond Entry, ","Third Entry, E
tc." !225
```

Line 30 sets up an array of 40 elements for the 40 phrases included in your data statements if you are going to use the LOOP in line 40 as is. Which is to say, you can have fewer data statements (but change your LOOP variable figure if you do). The same applies if you wish to increase the number of data items. The advantage of more data items is that you have more variety possible in the results.

Lines 50 and 60 give you the options of printing a number of paragraphs and how many lines you would like in each paragraph.

Once you understand what is needed, all you have to do is exercise your creative ability to create phrases, or select random passage phrases from any book that happens to be lying around.

Note line 110. You don't have to understand the expression used here. (Just be sure you copy each expression carefully!) The variable A will select from your first group of data items, B will likewise set the random choice for the second group of data items, C, the third, and D, the fourth.

(See Page 24)

Computer hazards

Computers can be a pain

By DAVE HOWELL

This appeared in the June 1994 newsletter of the Erie 99ers.—Ed.

We've been reading about certain hazards in using computers for extended periods of time. One of the concerns focused on the excessive radiation allegedly emitted from monitors. Newer models have apparently minimized that threat — if one can believe the "LOW RADIATION" stickers displayed on many new monitors in our stores.

According to Madonna Behen in the March issue of *Good Housekeeping*, injuries through constant use of computer terminals are on the increase. She says that an alarming number of office workers are suffering injuries from using computers on the job. Studies have shown that employees experience strain to their backs, eyes, legs and necks as well as wrist disorders.

If you use computers for several hours at a time, you might be advised to follow the suggestions listed in the box below.

Problem	Possible reasons	Solutions
Back Strain	Workstation surface is too low, causing you to hunch over the desk and put undue pressure on your back. Chair does not properly support back	Consider an adjustable-height work surface, especially if you share your station with others. Choose a chair that allows you to move the back and seat separately.
Eyestrain	Monitor is too close to user or is not at the proper height.	Locate VDT screen at least an arm's distance away and slightly below eye level.
Leg Strain	Placing feet flat on the floor and sitting in one position for a long time can cause stiffness and fatigue.	Use a movable footrest, which will elevate your feet and provide support. Get up and move around periodically.
Wrist Disorders	Resting wrists or arms on the work surface puts undue pressure on the nerves in the wrist. Extending wrists up or down can lead to fatigue and inflammation.	Keep wrists straight; don't rest them on the keyboard. Consider using wrist supports.

*Always seek medical attention for persistent discomfort.

SENTENCE—

(Continued from Page 23)

You will need to understand these variables to change from selection out of each 10 DATA statements per group. The data entries you make will decide what topics or phrases you wish the computer to randomly select. The format for this DATA organization (as the above program indicates) should be planned in four separate groups. Enter 10 phrases in each group in this order:

Group 1: (Antecedent Phrase) such as "Once upon a time, " (10 phrases)

Group 2: (Subject) such as "a great Emperor of China " (10 phrases)

Group 3: (Verb) such as "decreed "

Group 4: (Adverb Ending) such as "must wear burnt Levis!" (10 phrases)

Best you enter each DATA phrase or word inside quotation marks, which allows you freedom in adding punctuation. Note also the space left after each item in the first three groups. Remember if the LOOP in line 40 and the DIM in line 30 are set at 40, and variables in Line 110 are as is, you will need 10 DATA statements in each group. How funny your results will be will depend upon what material you chose for DATA, though even with non-humorous DATA, the random choices made by randomized variables chosen by the computer, called by Line 110, can still result in amusing juxtapositions.

This program opens up many possibilities such as a Bible quiz, where the random selection produces a result the player must unscramble. If it produced: "In the time of King Herod, St. Paul slew Goliath with water from a Macedonian well," you'd open up a can of worms. This would take some skill on your part in your choice of DATA, or, if the result was the previous statement, then the challenge would be for the player to point out how much of the sentence is not correct.

Support Our Advertisers

MICRO-REVIEWS

Ant Eater, Hen Pecked, Rotor Raiders, Princess and the Frog

By CHARLES GOOD

When was the last time you purchased shiny *new* cartridge games for your 99/4A. I don't mean the "recycled and comes in a zip lock bag" cartridges you get at TI shows and by mail from TI dealers, I mean virgin "still in their original manufacturer's shrink wrap container" cartridges containing games you and the kids have probably never played. L.L. Conner sells Romox game cartridges that meet these criteria.

Romox was a Campbell, California, company that sold third party games for the 99/4A and other computers in 1983 and 1984. These games could be loaded into reprogrammable cartridges that were loaded at retail stores that had Romox's game center. Pay the clerk, insert your reusable cartridge into the machine, and load your game. Games were available for Atari game systems and Commodore computers as well as for the 99/4A.

Romox also sold stand alone (not reprogrammable) cartridges, and the rights to distribute these cartridges were licensed to Navarone Industries in February 1984. The cartridges reviewed here are original Romox products, purchased from the current L.L. Conner Enterprise catalog. These are unusually shaped and really good-looking cartridges that were "made in Philippines." Each cartridge has a sloping top-front that contains a full color artist's picture illustrating part of the game action. They are certainly superficially different from the usual "official" TI cartridge, but, like the TI products, Romox cartridges plug into the 99/4a's cartridge port. Romox cartridges come in a shiny copper-colored box with holes in the box that reveal the artwork of the cartridge itself. These are not generic boxes, since each has the name of its particular game cartridge imprinted on the box along with specific game instructions printed on the back of the box. If you lose the box you lose the only printed game docs. However, most games come with plenty of on-screen help as well as an automatic demonstration mode.

All the Romox games make good use of the 99/4A's bells and whistles. Some games use speech. Each has a catchy musical tune and various additional sound effects that play throughout the game. You

(See Page 26)

Program comparison

Loaders: Boot vs. Funnelweb

By TED PETERSON

Peterson's article originally appeared in Wordplay, *the newsletter of the Portland Users of Ninety-Nines — Ed.*

This article will be on two of the loaders that our user group has in the library. They are Boot and Funnelweb. Also, I will mention Menu, which is not in our library. Everything said about Boot can apply to Menu.

The main difference between these two loaders and other load or loader programs is the regular load programs will run in Extended BASIC only, and these two loaders will run in regular BASIC, XBASIC or E/A (Editor/Assembler). Without these loaders, you would have to use the E/A module to load or run an E/A program.

Both of these loaders can be set up to hold a large number of your favorite utilities. With XB cartridge installed and Boot or FW in your disk drive, when XB is selected, a menu will pop up showing a listing of the different program sets. All you do is key your selection and that program will run.

The main difference between these two loaders and other load or loader programs is the regular load programs will run in Extended BASIC only, and these two loaders will run in regular BASIC, XBASIC or E/A (Editor/Assembler).

The difference between Boot and FW are as listed below:

1. FW comes with a DM-1000 (Disk Manager) type utility included and a good TI-Writer like word processor installed.

In Boot you would have to load in your own DM-1000, BA-Writer, DSKU, etc., programs.

2. Boot can be set up with 24 different sets of programs. FW can have about 14.

3. Boot is quicker to set up. To add a new set to the Boot menu, you just add the disk number the new program is on and its name. This takes about 30 seconds. FW uses programs to load a new program. After some practice you can set up a new program on the FW menu an about two to three minutes.

The Menu program that comes with the Horizon RAMdisk is the same as Boot but comes up in a different color. You can use Boot, but the FW program has to be changed to work on a RAMdisk.

Both FW and Boot have a lot of other options that can be used. The options differ, but both have a good set of instructions.

Try both of them and you will not need to pick up an E/A module unless you are going to do assembly programming. Some members like Boot better and some like FW better. That is your choice to make.

MICRO REVIEWS—

(Continued from Page 25)

have the option of turning off the music. All games can use joysticks or the keyboard for control.

I gave these four cartridges to my 14- and 16-year-old boys to play with. They turned off our 386 clone with CD-ROM, fired up the TI system next to the 386, and played with these cartridges for a week of afternoons before they got bored. At least one of the boys had the TI system going whenever they were home during this time. This is an indication of the entertainment value of these Romox games. Even though the games were made in 1983, they were new and interesting to the boys, for a while anyway.

L.L. Conner Enterprise will sell these *new* game cartridges to you for $10 each, or all four for $30. If you want, you can order the set of four wrapped up in a nice red ribbon with bow for gift giving.

ANT EATER
By Romox

You are an ant (actually three ants) in a nest in the ground. You are supposed to tunnel up to the surface, grab some food, and bring it back to the underground nest. Once all the surface food is obtained, you go on to the next level. On the surface is the deadly anteater, who will follow you on the surface and back down into your tunnels trying to eat you.

The ant is armed with five eggs that it will lay and leave behind in a tunnel at the push of the fire button. The egg explodes seconds later, destroying a pursuing ant eater if things go as you hope. Also, ants can tunnel under rocks in the ground that may then fall into the tunnel, squishing the pursuing anteater. When you advance to another level, you get an additional ant (life) and an additional anteater appears. At level 2 there are two anteaters, etc. The speed of the game increases with each of the nine levels.

HEN PECKED
By Romox

Navarone's title for the exact same game is Chicken Coop. This is my least favorite of the bunch. It has no demo mode and few on-screen instructions. Some aspects of the game seem hard for my little mind to figure out. In this one- or two-player game the rooster competes against the hens. You press the joystick button (or Q key) to flap your wings and go up, and you move the stick (or arrows) in the desired direction of movement.

Apparently the object of the game is for the rooster to get onto the back of a hen. I wonder what he does once he's there? Anyway, once contact is made in this manner the hen is captured and additional hens appear. If contact between rooster and hen is any other way (from the front or back or bottom), the rooster dies. That's because the hen pecked him, instead of vice versa. Things become tricky for the rooster because after the first hen is captured two hens appear, when one of those is captured two more appear, etc. There may be as many as four hens on screen simultaneously. With all these hens floating around randomly or chasing the rooster it is hard for the rooster to approach the hens in exactly the right orientation. In the two-player game, players take turns being the rooster and compete against each other for the highest score.

ROTOR RAIDERS
By Romox

No, the word rotor does not refer to a helicopter. This is not a helicopter rescue game. In this game the word rotor refers to a remote control drainage pipe auger designed to clean gunk out of drains. The object is to move your rotor around in the sewer maze eating all the droppings (the docs call them footprints) left by mice. Of course, you have to catch the mice too, and this isn't easy because the mice can run as fast as your rotor. You have to trap a mouse in a dead end and then roto him to mouse heaven, but meanwhile more mice appear.

This is a maze game somewhat resembling Pac-Man. Instead of energy dots there are mice droppings, and unlike PAC-Man new droppings are continuously deposited by the mice as they run around the sewers. Sometimes your flashlight will go out and you can't see the maze. All you can see are the mice and your rotor. You can still move under these conditions, feeling blindly for barriers. Soon the lights come on again. There are speed levels, which accelerate the speed of both the rotor and mice. Of the group of games reviewed here this is my favorite. I like the music, and the fast action just goes on forever. It is hard to lose but it is also hard to win, because you can never quite keep up with the mice. Finally time runs out and the game displays your score.

PRINCESS AND FROG
By Romox

This is your typical one-player "frogger" game with speech and some interesting variations. You move your frog across a field of jousting knights and into a moat filled with alligators and snakes. You jump from one moving animal to another until you get across the moat, but watch out! If you are on the back of an alligator it may submerge, taking the frog with it. In either the joust field or moat it is possible to jump both up/down and left/right to avoid obstacles. Left/right jumping over obstacles is, I believe, unusual in a "frogger" type of game.

On the other side of the moat is a castle with several open gates. Reaching any gate gives you bonus points, but in one gate is a pair of big red lips, the lips of the princess. If your frog manages to jump off a snake into this gate and kiss the princess, then the frog turns into a prince. Neat! And then there is this little extra, as quoted from the game box: "Bonus points are gained by mating with the female frog of the moat on the journey to the castle gates."

This is my second favorite of these four games. It is easily winnable, which is something an arcade game bimbo like me appreciates. This cartridge does not work on my 80 column AVPC system. The other cartridges do.

ACCESS:

L.L. Conner Enterprise (sells Romox cartridges), 1521 Ferry St., Lafayette IN 47904. Phone 317-742-8146

Charles Good, P.O. Box 647, Venedocia OH 45894, Phone 419-667-3131, internet email cgood@osulima1.lima.ohio-state.edu

Old tricks for new (and not so new) users

By JOSEPH COHEN

This article has appeared in a number of newsletters. The author is from the Lima Users Group and the article appeared in 1983 or before.—Ed.

Though many of us tend to ignore most of the cartridge software for our computer, with the exception of Extended BASIC and possibly Multiplan, Logo, Editor/Assembler and Terminal Emulator II (for speech), many of the cartridges are very enjoyable. In order to give you an excuse for searching your closets and basements looking for those hidden modules, I'd like to point out that many of them have undocumented features ranging from useful to interesting to amusing. Here are a few examples.

PERSONAL RECORD KEEPING

Many are probably familiar with "The Secret of Personal Record Keeping: Implementing DISPLAY AT and ACCEPT AT without XBASIC," published way back in *99er Magazine* and reprinted in *The Best of 99er*, page 76. Briefly, TI BASIC with the PRK module contains the commands CALL D( ) and CALL A( ) (similar to DISPLAY AT and ACCEPT AT). Presumably this also works with the Statistics module, but I do not have this one and could not verify it.

I have been told that this is a result of the hybrid nature of certain modules, containing both GPL and BASIC coding. Perhaps someone knows or could discover additional undocumented features of this cartridge, and other cartridges with CALL console BASIC routines (e.g. Tax/Investment Record Keeping).

DISK MANAGER

Next, the Disk Manager cartridge offers a proprietary protection feature that does not allow the Disk Manager to copy a protected diskette. To use it, press the FCTN X key 10 times while on any menu screen. You will hear a beep. If your monitor has sound, an >< will appear at the center top of the screen. Any diskette initialized as this point will be proprietary protected. Each time you address them using the Disk Manager cartridge (e.g., to catalog such a diskette), a low-tone beep will sound (not present for unprotected diskettes), informing you that the diskette is protected.

Press the FCTN X key 10 times while on any menu screen. You will hear a beep. If your monitor has sound, an >< will appear at the center top of the screen. Any diskette initialized as this point will be proprietary protected.

The protection information is stored in sector 0 on the diskette. This type of protection is ineffective against the sector disk copiers and has been discussed in the past. I wonder if anybody knows more about it. The DM-1000 offers protection and unprotection of diskettes. Is it the same kind as the Disk Manager cartridge?

GAME CARTRIDGES

Now to a few game cartridges. Moonmine, Alpiner, Munchman, Munchmobile and Hooper have a test mode, where you can select the starting level. So if you wanted to see what it is like to play at those levels you could never reach, here is a good reason to plug those cartridges into your 99/4A! The test mode is obtained by pressing Shift B, 3, 8 at the game title screen (Shift 8 only, for Hooper), and on Burgertime, pressing Shift 8 gives a message: "Code modifications by John M. Phillips."

Have you always played Parsec as a one-player game? Here is something different, for a two-player team. If the fire buttons on both joysticks are pressed simultaneously, Spaceship Parsec will not overheat. Horizontal lines will appear on the screen, but they do not disturb the game and would allow, in fact, one to see the Bynites when they turn invisible.

This is certainly not an exhaustive list. If anybody knows about other "tricks" please let me know.

Gaskill releases Mailing List Manager

Bill Gaskill has announced the release of Mailing List Manager, effective Feb. 1. The new program is a shareware product. The author says he will make it available through user group libraries, or users can obtain it by sending either $15 or a prepaid return mailer and two SSSD disks to William Gaskill, 2310 Cypress Ct., Grand Junction, CO 81506.

Gaskill describes the program as offering unlimited file sizes as well as add, browe, delete, find, global delete, index, merge, report, sort and subfile creation features. He says it also provides for envelope addressing, columnar, label or Rolodex-type reports, and it has a label editor that lets the user design and print labels in a TI-Writer-like environment without having to learn to use any other part of the program.

MLM also offers a text editor that lets the user attach up to 144 lines of free-form text to any record for use in contact management or for just keeping track of mail correspondence, according to Gaskill.

MLM requires 32K RAM, Extended BASIC, at least one disk drive of any density and, if hard copy output is desired, a printer.

USER NOTES

Bug fix reported

A program published in the November 1994 MICROpendium contained a small bug. The bug had no significant effect on operation of the program. The program, Forget Me Knots, was written by Alfred Malcolm. The bug is in line 950.

To correct the bug, change CALL KEY(_,D,E) to CALL KEY(@@,D,E).

This fix permits the use of lowercase characters.

Hide and seek in Extended BASIC

Don Steffen, of Silverton, Oregon, writes:

These are two short programs named HIDE-64 and FIND-64 that enable you to make use of a feature of the TI99/4A that may not be available on other computers.

The HIDE-64 program takes a decimal number that you input up to over 18 quintillion which it converts to a binary number up to 64 bits long. Then it converts that to a 16-character hexadecimal character pattern for any ASCII character number from 128 to 143. It then shows a sprite (magnify 2) with that character pattern. You can then stop the program, load and run the FIND-64 program and it will display the decimal value of the number you hid. This occurs as long as you do no turn the power off on your TI. This is done without using any file.

This program could be used to transfer numbers from one program to another or to store your Social Security number of driver's license or credit card numbers where they can be viewed when needed. Up to 16 numbers can be hidden and retrieved this way. The program restores the hidden number as a string, rather than as scientific notation, up to the largest number that can be hidden.

FIND-64

```
10 ! *** FIND 64 BIT NUMBER
022
12 ! SAVED AS FIND-64 ON DST
EFEN381 !036
14 K=127 :: N=1 :: CALL CLEA
R :: CALL SCREEN(5):: CALL M
AGNIFY(2)!194
16 ! *** PEEK AT 64 BIT CODE
D NUMBER AND CONVERT TO DECI
MAL VALUE. !221
18 K=K+1 :: N=N+1 :: CALL CH
ARPAT(K,K$):: CALL SPRITE(#N
,K,16,N*4+50,240,0,-10)!224
20 L$=SEG$(K$,1,8):: R$=SEG$
(K$,9,8):: CALL HTD(L$,D$)::
 CALL HTD(R$,D2$):: D2$=RPT$
("0",9-LEN(D2$))&D2$ :: D$=D
$&D2$ !249
22 L=LEN(D$):: IF SEG$(D$,1,
1)="0" THEN D$=SEG$(D$,2,L-1
):: GOTO 22 !074
24 PRINT "":D$ :: D$="" :: I
NPUT "FIND ANOTHER? Y/N ":Z$
 :: IF Z$="Y" THEN CALL HCHA
R(23,1,32,32):: GOTO 18 ELSE
 STOP !162
26 !!131
28 ! ### SUBPROGRAMS ### !01
0
30 SUB HTD(H$,D$):: FOR X=7
TO 0 STEP -1 :: Z=16^X :: E=
ASC(SEG$(H$,8-X,1))-48 :: E=
E+(E>9)*7 :: D=D+Z*E !003
32 NEXT X :: IF D>0 THEN D$=
STR$(D)ELSE D$="" !193
34 D=0 :: SUBEND !037
40 B$="" :: K=127 :: N=1 !09
9
```

HIDE-64

```
42 ! *** HIDE-64 ******
 BY DON STEFFEN 13DEC94 !082
44 ! SAVED AS HIDE-64 ON DST
EFEN381 !029
46 ! THIS PROGRAM CODES ANY
DECIMAL NUMBERS UP TO 19 DIG
ITS LONG INTO CHARACTERS 128
-143 !020
48 ! ~~~~ CONVERT DECIMAL VA
LUE TO HEX STRING !071
50 K=K+1 :: N=N+1 :: H$="" !
025
52 DISPLAY AT(2,2)ERASE ALL:
"ENTER A DECIMAL NUMBER UP T
O 19 DIGITS " :: ACCEPT AT(6
,4)SIZE(19)VALIDATE(DIGIT):D
$ :: IF LEN(D$)>19 THEN 52 !
147
54 L=LEN(D$):: D$=RPT$("0",1
9-L)&D$ :: L$=SEG$(D$,1,10):
: R$=SEG$(D$,11,9):: CALL DT
H(L$,H$):: CALL DTH(R$,G$)::
 K$=H$&G$ :: PRINT K$ !242
56 ! ~~~ STORE CODE IN CHARA
CTER !105
58 K$=RPT$("0",16-LEN(K$))&K
$ :: CALL CHAR(K,K$):: CALL
SCREEN(5):: CALL MAGNIFY(2):
: CALL SPRITE(#N,K,16,N*4+40
,240,0,-10):: D$,H$,K$="" !1
07
60 ! ~~~ DO ANOTHER? !064
62 L$,R$,G$="" :: INPUT "DO
ANOTHER? Y/N ":Z$ :: IF Z$="
Y" THEN 50 ELSE STOP ! OR RU
N"DSK1.FIND-64 !014
64 ! ### SUBPROGRAMS ### !01
0
66 SUB DTH(D$,H$):: D=VAL(D$
):: FOR X=7 TO 0 STEP -1 ::
Z=16^X !151
68 IF D>=Z THEN H=INT(D/Z)::
 D=D-H*Z :: H=H+48-(H>9)*7 E
LSE H=48 !099
70 H$=H$&CHR$(H):: NEXT X ::
 PRINT H$ :: SUBEND !253
```

Do you have the aptitude?

The following program was written by Jim Peterson. Peterson, also known as Mr. Tigercub, died last year.

The program tests your ability to match shapes. It is not as easy as it sounds.

APTITUDE

```
1 DATA 0000007824242478,0010
301010101038,003844040810207
C,0038440418044438,000818284
87C0808,007C407804044438,001
8204078444438 !053
2 DEF RD=INT(5*RND+1)!108
10 GOTO 100 !179
11 SET,D,Q,ST,CH,T,J,X,L(),L
L(),M,N,RX,CX,R,TT,C,RD,K,K2
,F,CH$ !134
30 CALL CLEAR :: CALL CHAR :
: CALL TITLE :: CALL COLOR :
: CALL DELSPRITE :: CALL SCR
EEN :: CALL KEY :: CALL HCHA
R :: CALL VCHAR :: CALL SOUN
```

(See Page 29)

USER NOTES

(Continued from Page 28)

```
D !246
40 !@P- !064
100 CALL CLEAR :: CALL CHAR(
94,"3C4299A1A199423C"):: CAL
L TITLE(2,"MECHANICAL APTITU
DE TEST"):: FOR SET=2 TO 12
:: CALL COLOR(SET,15,1):: NE
XT SET !126
110 DISPLAY AT(1,10):"TIGERC
UB SOFTWARE" :: DISPLAY AT(3
,16):"^ TCX-1129" :: FOR D=1
 TO 500 :: NEXT D :: CALL DE
LSPRITE(ALL)!254
120 FOR SET=2 TO 12 :: CALL
COLOR(SET,2,1):: NEXT SET ::
 CALL CLEAR :: CALL SCREEN(1
6)!208
130 ! programmed by Jim Pete
rson 2/84, XBasic version 7/
85 !008
140 ! COPYRIGHT 1984 TIGERCU
B SOFTWARE 156 COLLINGWOOD A
VE. COLUMBUS OHIO 43213 !143
150 ! REPRODUCTION PROHIBITE
D. DELETION OF COPYRIGHT NOT
ICE PROHIBITED !103
160 DISPLAY AT(3,3):"MECHANI
CAL APTITUDE TEST": : :"TCX-
1129 ^ Tigercub Software": :
" Nine broken blocks will be
":"placed on the screen." !0
38
170 DISPLAY AT(11,1):" Two o
f them can be rotated":"or f
lipped to fit together":"int
o a perfect square." !240
180 DISPLAY AT(15,1):" Type
the numbers of the two":"tha
t go together.": : :" Do you
 want a grid?(Y/N)" !061
190 CALL KEY(3,Q,ST):: IF ST
=0 THEN 190 !179
200 CALL CLEAR :: GOSUB 630
:: CALL CLEAR :: CALL SCREEN
(11)!180
210 CALL CLEAR :: FOR SET=2
TO 10 :: CALL COLOR(SET,5,1)
!130
220 NEXT SET :: FOR CH=43 TO
 107 STEP 8 :: CALL CHAR(CH,
"FFFFFFFFFFFFFFFF")!188
230 NEXT CH :: IF Q<>89 THEN
 250 !165
240 CALL CHAR(32,"FF80808080
808080")!017
250 CALL CLEAR :: T=1 :: RAN
DOMIZE :: FOR J=1 TO 6 :: X=
INT(5*RND+1):: L(J)=X :: LL(
J)=6-X !160
260 NEXT J !224
270 M=INT(9*RND+1)!158
280 N=INT(9*RND+1):: IF N=M
THEN 280 !144
290 FOR RX=3 TO 19 STEP 8 ::
 FOR CX=3 TO 19 STEP 8 :: CA
LL HCHAR(RX-1,CX+2,T+127)::
ON INT(4*RND+1)GOSUB 320,340
,360,380 :: T=T+1 !109
300 NEXT CX !049
310 NEXT RX :: GOTO 450 !212
```

(See Page 30)

MICROpendium disks, etc.

- ❑ **Series 1994-1995 mailed monthly** (April 1994-March 1995)..... $40.00
- ❑ **Series 1993-1994 mailed monthly** (April 1993-March 1994)..... $25.00
- ❑ **Series 1992-1993** (Apr 1992-Mar 1993, 6 disks).. $25.00
- ❑ **Series 1991-1992** (Apr 1991-Mar 1992, 6 disks).. $25.00
- ❑ **Series 1990-1991** (Apr 1990-Mar 1991, 6 disks) ..$25.00
- ❑ **Series 1989-1990** (Apr 1989-Mar 1991, 6 disks) ..$25.00
- ❑ **Series 1988-1989** (Apr 1988-Mar 1989, 6 disks)...$25.00
- ❑ **110 Subprograms** (Jerry Stern's collection of 110 XB subprograms, 1 disk) $6.00
- ❑ **TI-Forth** (2 disks, req. 32K, E/A, no docs).....$6.00
- ❑ **TI-Forth Docs** (2 disks, D/V80 files) $6.00
- ❑ **1988 updates of TI-Writer, Multiplan & SBUG** (2 disks) $6.00
- ❑ **Disk of programs** from any one issue of MICROpendium between April 1988 and present $4.00
- ❑ **CHECKSUM and CHECK** programs from October 1987 issue (includes docs as D/V 80 file).....$4.00

Name _____

Address _____

City _____

State _____ ZIP _____

Texas residents add 7.75% sales tax.. Credit card orders add 5%.
Check box for each item ordered and enter total amount here:

Check/MO Visa M/C
(Circle method of payment)

Credit Card # _____

Exp. Date _____

Signature _____

USER NOTES

(Continued from Page 29)

```
 FOR CX=3 TO 19 STEP 8 :: CA
LL HCHAR(RX-1,CX+2,T+127)::
ON INT(4*RND+1)GOSUB 320,340
,360,380 :: T=T+1 !109
300 NEXT CX !049
310 NEXT RX :: GOTO 450 !212
320 J=0 :: FOR R=RX TO RX+5
:: J=J+1 :: GOSUB 400 :: CAL
L HCHAR(R,CX,35+T*8,TT)!077
330 NEXT R :: RETURN !242
340 J=0 :: FOR C=CX TO CX+5
:: J=J+1 :: GOSUB 400 :: CAL
L VCHAR(RX,C,35+T*8,TT)!046
350 NEXT C :: RETURN !227
360 J=0 :: FOR R=RX TO RX+5
:: J=J+1 :: GOSUB 400 :: CAL
L HCHAR(R,CX+5-TT,35+T*8,TT)
!118
370 NEXT R :: RETURN !242
380 J=0 :: FOR C=CX TO CX+5
:: J=J+1 :: GOSUB 400 :: CAL
L VCHAR(RX+5-TT,C,35+T*8,TT)
!087
390 NEXT C :: RETURN !227
400 IF M<>T THEN 420 !194
410 TT=L(J):: RETURN !115
420 IF N<>T THEN 440 !215
430 TT=LL(J):: RETURN !191
440 TT=RD :: RETURN !006
450 DISPLAY AT(3,24):CHR$(14
0)&CHR$(141)&CHR$(32)&CHR$(1
42)!025
460 CALL HCHAR(3,30,30)!249
470 FOR D=1 TO 10 :: NEXT D
:: CALL HCHAR(3,30,32):: CAL
L KEY(0,K,ST):: IF (ST=0)+(K
<49)+(K>57)THEN 460 !195
480 CALL HCHAR(3,30,K+79)::
CALL COLOR(K-47,7,7):: DISPL
y y" !236
560 CALL KEY(0,K,ST):: IF ST
=0 THEN 560 ELSE 210 !058
570 F=262 :: FOR J=1 TO 12 :
: CALL SOUND(-99,F*1.0594630
94^J,5)!189
580 NEXT J :: GOTO 550 :: RE
TURN !227
590 !!131
600 DATA 007C040810202020,00
38444438444438,003844443C040
830 !047
610 DATA 00000038447C4444,00
00007C4078407C,0000002428302
824,0000004464544C44 !241
620 DATA 0000007C4444447C,00
28287C287C2828,0038440408100
010 !084
630 FOR CH=127 TO 143 :: REA
D CH$ :: CALL CHAR(CH,CH$)!1
21
640 NEXT CH :: RETURN !043
649 !@P+ !062
650 SUB TITLE(S,T$)!219
654 GOTO 660 !229
656 S,T$,L,J !122
658 CALL SCREEN :: CALL MAGN
IFY :: CALL SPRITE !232
659 !@P- !064
660 CALL SCREEN(S):: L=LEN(T
$):: CALL MAGNIFY(2)!146
670 FOR J=1 TO L :: CALL SPR
ITE(#J,ASC(SEG$(T$,J,1)),J+1
-(J+1=S)+(J+1=S+13)+(J>14)*1
3,J*(170/L),10+J*(200/L))::
NEXT J !118
680 !@P+ !062
681 SUBEND !168
```

Using Telco to upload ASCII text files

The following item has appeared in several user group newsletters. The author is unknown.

Those who have place messages or full-fledged works of prose onto a BBS with the Telco line editor o the editor provided by the BBS have experienced the ultimate pain in the you know what. Keep in mind that the line editor is just that. It could never be a complete word processor by any stretch of the imagination.

So what do you do when you want to put a message longer than two or three lines on a BBS? You upload the text file from a disk drive, that's how. Thus, you create a finished text, spell-checked and all, save it to a disk and then upload it directly from the disk to the BBS. This happens all without the headache of fumbling with a line editor.

How is this accomplished? It's simple, and here are instructions:

First, you must configure the file transfer setup routine in Telco. Choose "Setup Options" from Telco's main menu. Then choose "File Transfer Setup" from that menu and set the file transfer defaults. The defaults that work well for me are as follows:

A)	Abort downloads	: Discard
B)	Default error check	: CRS
C)	Echo locally	: Off
D)	Blank lines	: Off
E)	Character pacing	: 000
F)	Line pacing	: 010
G)	Pace character	: 000
H)	Strip leading space	: Off
I)	Line by line feed	: Off
J)	Send at end of line	: CR
K)	CR translation	: Strip
L)	LF translation	: Strip

Save these changes before continuing.

Once the file transfer defaults are set and saved, you can upload any text file with the greatest of ease using the following five steps:

1. Whether you are contributing an article or an answer to a BBS or writing to somebody by email and are cued on the screen to enter your file, just press Fctn-6.
2. Choose ASCII option by pressing Enter.
3. Type the full disk and filename into the filename block — DSKn.FILENAME — and press Enter.
4. You text file will then be uploaded and you can monitor the progress by the uploaded line numbers shown at the bottom of the screen. The process ends when the "Transfer Complete-Press a Key" message appears.
5. The last line of your text file will be shown at the bottom of the screen, and the cursor will appear on the next blank line. To signal End-of-File, type in "###" and press Enter.

That's all there is to uploading files with Telco. It's quick and it's simple.

Using TI-Writer to transfer text over the serial port

We're not sure who the author of this item is, but we picked it up Hocus Focus

(See Page 31)

USER NOTES CLASSIFIEDS

(Continued from Page 30)

99, the newsletter of the Milwaukee Area User Group.

You can hook two different kinds of computers together with a cable linking the RS232 ports of both computers. The TI serial printer cable will do the trick. You can then load text files directly into TI-Writer (or the Funnelweb editor) from a word processor program on any other computer. You don't need a modem or a terminal program, and the other computer doesn't have to be compatible with the TI.

After cabling the two computers' RS232 ports together, boot TI-Writer, type LF (Load File) and press Enter. Then type RS232.CR for the file name and press Enter. The TI's screen will appear to be locked up as the TI waits to receive the file from the RS232 port. It may be necessary to specify a baud rate in the RS232.CR file name if the default 300 baud is not satisfactory. However, TI-Writer (and Funnelweb) will not accept a baud rate higher than 600.

With the other computer, save or send a text file already in memory, specifying RS232 as the save file name. PC users may have to specify the COM port rather than RS232. Text will then flow into TI-Writer. When the transfer is complete, press Fctn-4 on the TI and the received text file will be displayed.

This is how I transport text from my CC40 to my TI for processing with Funnelweb and printing with my Star printer.

Have a tip or program that might be of use to other TI and Geneve users? MICROpendium pays $10 for items submitted by readers that appear in the User Notes column. Send them to: MICROpendium User Notes, P.O. Box 1343, Round Rock, TX 78680. Or email them to jkoloen@io.com.

Support MICROpendium Advertisers!

FOR SALE

TI99/4A

TI software & hardware. Joy Electronics Inc., P.O. Box 542526, Dallas, TX 75354-2526. 214-243-5371. Call or write for free catalogue. v12n5

FOR SALE

TI99/4A computer, power supply & video modulator. Guaranteed to work. I have no peripherals. $70.00. Call 704-497-7534.

BUY-SELL-TRADE

Hardware — Software. TI Buyers Guide $2. Dual 1/2 HT Kit complete

SERVICES

with cab les, screws, drill template & instructions $59. PE Box complete $149. Corcomp DD Cont $180. Star NX-1001 Printer $175. 2400 Baud Modem $79. TI99/4A Console $45. Rare — Myarc Hard & Floppy Controller with 10 Meg Hard Disk & Case $350. 800-471-1600 (Nationwide & Canada ORDERS ONLY). 414-672-1600 Local/tech Support. Huge Genuine TI Inventory Since 1982. Competition Computer, 2219 S. Muskego, Milwaukee, WI 53215. Fax 414-672-8977. Open Daily 9-5, Sat 10-3. Bankcards, Discover, Checks & UPS/COD.

LOOK

What a deal!

Prices slashed on MICROpendium Classifieds!

Classified ads are now available for 10 cents per word, a reduction of more than 50 percent.

If you want to sell or buy something,advertise it in MICROpendium Classifieds.
Simply write your ad on a separate sheet of paper, count the word (a phone number counts as one word) and send it, along with payment, to:

MICROpendium Classifieds
P.O. Box 1343
Round Rock, TX 78680.

MasterCard VISA

The ONLY monthly devoted to the TI99/4A

Subscription Fees

❑ 12 issues, USA, **$35** ❑ 12 issues, Mexico, **$40.25**

❑ 12 issues, Canada **$42.50** ❑ 12 issues, other countries surface mail, **$40.00**

❑ 12 issues, other countries, air mail, **$52.00**

Outside U.S., pay via postal or international money order or credit card; personal checks from non-U.S. banks will be returned.

Address Changes

Subscribers who move may have the delivery of their most recent issue(s) delayed unless MICROpendium is notified six weeks in advance of address changes. Please include your old address as it appears on your mailing label when making an address change.

Check each item ordered (or list on separate page) and enter total amount here:_____

Check/MO ❑ ❑ ❑ (check one)

Card No. _____

Expiration Date _____
(Minimum credit card order is $9)

Signature _____
(Required on credit card orders.)

No sales tax on magazine subscriptions. Texas residents add 7.75% sales tax on other items, including back issues and disk subscriptions. Credit card orders add 5%.

Mail to: MICROpendium, P.O. Box 1343, Round Rock, TX 78680

Name _____

Address _____

City _____

State _____ **ZIP** _____

The set of numbers at the top of your mailing label indicates the cover date of the last issue of your subscription.

Disks, Etc.

❑ **Back Issues**, $3.50 each. List issues:_____

_____

No price breaks on sets of back issues. Free shipping USA. Add 30 cents, single issues to Canada/Mexico. Other foreign shipping 50 cents single issue surface, $1.50 airmail. Write for foreign shipping on multiple copies.

OUT OF STOCK: Vols. 1, No. 1-2; Vol. 2, No. 1

❑ **MICROpendium Index** (2 SSSD disks, 1984-1992), Extended BASIC required $6.00

❑ **MICROpendium Index II** (9 SSSD disks — 1 for each year — 1984-1992), XB required $30.00

❑ **MICROpendium Index II with MICROdex 99** (11 SSSD disks), XB required.....$35.00

❑ **MICROdex 99** (for use with MP Index II, 2 SSSD disks), XB required $10.00

❑ **MICROpendium Index II** annual disks ordered separately (1 disk per year, 1984-1992); each $6.00

MICROdex 99, by Bill Gaskill, is a collection of programs that allow users of MP Index II to modify their index entries, as well as add entries. MICROdex 99 supports many other functions, including file merging, deletion of purged records, record counting and file browsing.

GENEVE DISKS (SSSD unless specified)

❑ **MDOS 2.0** (req. DSSD or larger (for floppy & hard drive systems) $4.00

❑ **GPL 1.5** $4.00

❑ **Myarc Disk Manager 1.50**.....$4.00

❑ **Myarc BASIC 3.0**.....$4.00

❑ **MY-Word V1.21** $4.00

❑ **Menu 80** (specify floppy or hard disk versions(s); includes SETCOLR, SHOWCOLOR, FIND, XUTILS, REMIND $4.00

GENEVE PUBLIC DOMAIN DISKS

These disks consists of public domain programs available from bulletin boards. If ordering DSDD, specify whether Myarc or CorComp.

	SSSD	DSSD	DSDD
❑ **Series 1**	$9.00	$7.00	$5.00
❑ **Series 2**	$9.00	$7.00	$5.00
❑ **Series 3**	$9.00	$7.00	$5.00
❑ **Series 4**	$9.00	$7.00	$5.00
❑ **Series 5**	$9.00	$7.00	$5.00
❑ **Series 6**	$9.00	$7.00	$5.00

SECOND CLASS

A T EXPIRES 11/95
LEONARD CUMMINGS
815 NORTH BLACKHOOF
WAPAKONETA OH 45895