Covering the TI99/4A and the Myarc 9640

MICROpendium

Volume 10 Number 9 October 1993 $3.50

Halloween Treats!

Chicago Faire
Oct. 30

Milwaukee Faire
Oct. 31

Regena reviews

Assembly warning: Never do this!

XBASIC card file, plus the Battle of Bull Run

Thoughts on competition

REVIEWS:
The Asgard Memory System
ALEX/FORMA
Maze Mania
3.5" Disk Label Program

And much more!

TEX-COMP

$$ DOLLAR DAYS

WATCH YOUR SAVINGS GROW DURING DOLLAR DAYS

SOFTWARE SUPER SALE!

PHM 3027 ADDITION & SUBTRACTION I
PHM 3028 ADDITION & SUBTRACTION II
PHM 3093 MILLIKEN DIVISION
PHM 3051 NUMERATION II
PHM 3094 MILLIKEN INTEGERS
PHM 3118 MINUS MISSION
WEIGHT & NUTRITION
PHM 3002 EARLY LEARNING FUN
PHM 3003 BEGINNING GRAMMAR
PHM 3112 PARSEC
PHM 3007 HOUSEHOLD BUDGET MANAGEMENT
PHM 3044 PERS. REPORT GENERATOR**
PHM 3053 TI INVADERS
PHM 3194 JAWBREAKERS II

PHM 3018 TAX INVESTMENT RECORD KEEPING***
PHM 3022 PERSONAL REAL ESTATE
PHM 3028 HUNT THE WUMPUS
PHM 3031 TIC ATTACK
PHM 3220 MICROSURGEON
PHM 3067 MUNCHMAN
PHM 3110 CHISHOLM TRAIL
PHM 3219 SUPER DEMON ATTACK
PHM 3006 HOME FINANCIAL DECISIONS
PHM 3035 TERMINAL EMULATOR II
PHM 3052 TOMBSTONE CITY
PHM 3109 LOGO II* →
PHM 3065 EDITOR ASSEMBLER
PHM 3010 PHYSICAL FITNESS
PHM 3140 MUNCHMOBILE

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 THREE TITLES DUE TO WEIGHT
** FOR USE WITH PERSONAL RECORD KEEPING, *** DISK DRIVE REQUIRED
PERSONAL RECORD KEEPING AVAILABLE AT $14.95

Special Purchase

DONKEY KONG featuring Mario $4.95 (regular $9.95)

AS LOW AS $1.95 EACH!!

WHEN BUYING TEN OR MORE
OR BUY ANY AMOUNT AT ONLY $2.95 EACH.

99¢ BOOK BLOWOUT!!!

PUBLISHER'S CLEARANCE ON BOOKS FOR THE 99/4A
BUY FIVE OR MORE BOOKS FOR ONLY 99 CENTS EACH + s&h.
BUY ANY AMOUNT LESS THAN 5 FOR ONLY $1.49 EACH + s&h.

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

****

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 ORC/HEART BONUS ADVENTURE

ONLY $12.95 FOR ALL 13.

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

ONLY $9.95 FOR ALL 8

COMMAND MODULE FOR ABOVE ADVENTURES (reg $9.95) ONLY 95 CENTS WITH EITHER OF THE ABOVE SERI

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

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 style! This disk or cassette based program allows:

1. Any existing adventure for the Adventure command module(s) to be edited, altered, listed and copied to 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 Programming 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 SOON From TEX-COMP **NEW LOW PRICE**

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

SUPER BUY! $4.95

$4.95 Per Disk

NEW BONUS SAVE AS MUCH AS 33%
BUY FIVE DISKS GET ONE FREE
BUY TEN DISKS GET THREE FREE
BUY FIFTEEN DISKS GET FIVE FREE

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

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 5023 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

****

TEX-COMP

P.O. Box 33084, Granada Hills, CA 91344

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

TERMS: All prices F.O.B. Los Angeles. For fastest service use cashiers check or money order. Personal checks take ten days to clear. Add 3% shipping & handling (with a $3 minimum). Add 4.5% east of Mississippi. All prices shown include a 3% cash discount. Add 3% if paying by Visa or Mastercharge. Prices and availability subject to change. Include street address for UPS delivery. California orders add 8.25% sales tax. Canada, Hawaii, Alaska and overseas shipping extra.

CHARGE IT WE ACCEPT VISA or MASTERCARD

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

Send $2 for our new 1993 catalog

AUTHORIZED DEALER

CONTENTS

MICROpendium

MICROpendium (ISSN 10432299) is published monthly for $25 per year by Burns-Koloen Communications Inc., 502 Windsor Rd., Round Rock, TX 78664. Second-class postage paid at Austin, Texas, and additional mailing offices. POSTMASTER: Send address changes to MICROpendium, P.O. Box 1343, Round Rock, TX 78680-1343.

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 $30.25 (Mexico); $32.50 (Canada); $30.00, surface mail to other countries; $42 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: (512) 255-1512
CompuServe: 75156,3270
Delphi TI NET: MICROpendium
GEnie: J.Koloen
Internet E-mail: jkoloen@holonet.net
John Koloen.....Publisher
Laura Burns.....Editor

DEPARTMENTS

*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.

COMMENTS

Barry plans return to MICROpendium

I'm happy to write that Barry Traver will be writing for MICROpendium again. We expect to have material from Barry for the November edition.

As readers know, Barry has been suffering from Carpal Tunnel Syndrome since last spring and has not been able to use a computer keyboard on a regular basis. However, recently he reported that his condition was improving and that it would allow him to work at his computer with greater regularity. I'm not certain how frequently Barry will be submitting articles, but it's certainly good news that we can look forward to seeing his byline in the near future.

BLACK FRIDAY ANNIVERSARY

Just a reminder, not that you need it, that Black Friday is just around the corner. Black Friday, of course, is the day in 1983 that Texas Instruments announced that it was abandoning the home computer market and the TI99/4A. So mark Oct. 28 on your calendar.

CECURE RELEASES HARD DISK FORMATTER

Cecure Electronics has released a hard disk formatter that runs out of MDOS. The program, called CForm, allows users to increase the number of sectors per cylinder from 32 to 34, creating an extra 1.2 megabytes of storage space on a 20-megabyte drive. Larger drives will have proportionate increases.

CForm was written by Mike Maksimik. A version for the TI99/4A is expected in the near future. CForm is priced at $15. Contact Cecure Electronics at P.O. Box 132, Muskego, WI 43150.

Maksimik is also writing software that will back-up hard or floppy disks to tape. Called Quick 40 and Quick 80, the programs are in the beta test stage. The programs will support tape drives used on PCs, such as those manufactured by Mountain Systems. These drives are the same size as half-height floppies and use one-quarter inch tape data cartridges.

The backup software will be distributed as freeware through 9640 News, P.O. Box 752465, Memphis, TN 38175.

LOOKING FORWARD TO CHICAGO

Laura and I are looking forward to visiting with fellow TIers again this year at the Chicago TI fair. I hope to see many of you there.

—JK

1993 TI FAIRS

APRIL

Northeast TI Fair, April 17, Waltham High School, Waltham, Massachusetts. Contact Ron Williams, 14 East St., Avon, MA 02322.

Canadian TI Fest, April 24, Merivale High School, Nepean, Ontario, Canada. Contact Bill Gard, 3489 Paul Anka Dr., Ottawa, Ontario, Canada K1V 9K6 or (613) 523-9396 or Fax (819) 997-2194 Attn: DMES 2.

MAY

Fourth Annual TI Orphans Reunion, May 15, Zurich Insurance Claims Centre, 9715 Ottewell Rd., Edmonton, Alberta, Canada. Contact Ron Hohman, (403) 456-0862.

SEPTEMBER

Western Washington TI Fair, Sept 18, Tacoma Waterworks, 3506 S. 35th, Tacoma, Washington. Jim Tompkins, (206) 756-0934.

OCTOBER

Annual International TI-Faire, Oct. 8-10, Evangelisches Ferienwaldheim Weidachtal, 7000 Stuttgart 80 (Mörhingen), Weidach Gewann 8, Germany. Contact Hans Huben, Berberitzenweg 6, 7033 Herrenberg, Germany; Wolfgang Bertsch, Helenenburgweg 61, 7120 Bietiigheim-Biss, German; or Dierk Warburg, Lilienweg 12, 7141 Benningen, Germany.

Chicago International World Faire, Oct. 30, Holiday Inn, Gurnee, Illinois. Contact Cecure Electronics, P.O. Box 132, Muskego, WI 53150, or Don Walden, (414) 679-4343.

Milwaukee TI Fair, Oct. 31, Quality Inn, 5311 Howell Ave, Milwaukee, Wisconsin. Contact Gene Hitz, 4121 North Glenway, Wauwatosa, WI 53222.

1994 FAIRS

FEBRUARY

Fest-West, Feb. 19-20, Santa Rita Park Inn, Tucson, Arizona. Contact Tom Wills, Fest-West '94 Committee, Southwest 99ers Users Group, P.O. Box 17831, Tucson, AZ 85731 or (602) 886-2460; BJ Mathis, (602) 747-5046; or the Cactus Patch BBS, (602) 290-6277.

MAY

Lima Multi User Group Conference, May 14-15, Ohio State University Lima Campus, Lima, Ohio. ContactLima Oho Users Group, P.O. Box 647, Venedocia, OH 45894.

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.

FEEDBACK

Another screen dump

I keep noticing screen dump programs written in Extended BASIC, but I don't recall any mention of the "Danny Michaels" screen dump written in assembly language that can be loaded via Extended BASIC. This program can dump the screen of an Extended BASIC program as it runs, or cartridge (with a load interrupt switch) in a matter of seconds. It also is able to rotate and print in full page mode. The program was designed for the use of Epson graphics type printers. It has been modified, by me, for use with an Axiom or C. Itoh type graphic printer thanks to Danny's excellent documented source code. A copy of these programs can be obtained through the SouthWest 99ers library, P.O. Box 17831, Tucson, AZ 85731.

I am presently working on DM1000 V.6.2 and correcting and improving on 6.1 with such features as quad density support and high density formatting, displaying the number of read/write errors while sector copying.

Jack C. Mathis
South Korea

Just like clockwork

Just to let you know, I am a long time subscriber to your wonderful magazine — since October '84. (I have all the back issues, too!)

MICROpendium has always been a source of pleasure for me. It's kept me informed and up-to-date with the latest software and hardware. Like clockwork, once a month, your publication manages to find my mailbox (I've moved four times), and in just a few minutes, lets me know what's going on in the TI world I really enjoy reading it!

Roland Chapman
Reno, Nevada

***Feedback is a reader forum. The editor may condense excessively lengthy submissions if necessary. We ask that writers limit themselves to one subject persubmission. Our only requirement is that submissions be of interest to those using the TI99/4A, the Geneve 9640 or compatibles. Send items to* MICROpendium Feedback, P.O. Box 1343, Round Rock, TX 78680.**

BUGS & BYTES

Do not collect $200

Look for TI-Nopoly to become available in the near future. The TI game, which is similar to a famous board game by Parker Brothers, was demonstrated at the Lima Multi User Group Conference last spring. It is by Joe Delekto and Jon Dyer and takes advantage of Asgard Software's Advanced Memory System.

Not for women only?

The Women's Information & Resource Exchange (WIRE), the first on-line service devoted to women, started Oct. 9. Subscribers pay $15 a month for two hours of use, with each additional hour $2.50.

Women are estimated at 10 to 15 percent of on-line users.

Founders of the new service are Ellen Pack and Nancy Rhine. for information or to subscribe, call (415) 615-8989.

The service is also open to men.

Kid appeal

Summer has ended here, but begun south of the Equator, and we found this comment in the September 1993 newsletter of the TI Brisbane User Group:

School holidays must be here as well because today all the local children who do not own Sega Mega-Drives were at my desk playing TI-Runner and that game with the real annoying music (Burger Time); it sounds like I'm complaining, but I'm not. It's nice to know that the TI is more popular in Willowbank than the other machines.

Attend a TI Fair This Year

MICROpendium

DISK SALE

If you've been waiting for a sale on MICROpendium program disks, this is it! For a limited time (through Nov.15, 1993) Series 1-6 disks are available for a special price. (Series 6 disks are mailed monthly starting with the October 1993 edition, programs from April 1993 through October 1993 will be mailed as soon as the order is placed.)

MICROpendium disks

SERIES #	REGULAR PRICE	SALE PRICE	YOU SAVE	DISCOUNT
Series 1 (Apr. '88-Mar. '89)	$25.00	**$15.00**	*$10.00*	*40%*
Series 2 (Apr. '89-Mar. '90)	$25.00	**$15.00**	*$10.00*	*40%*
Series 3 (Apr. '90-Mar. '91)	$25.00	**$15.00**	*$10.00*	*40%*
Series 4 (Apr. '91-Mar. '92)	$25.00	**$15.00**	*$10.00*	*40%*
Series 5 (Apr. 92-Mar. 93)	$25.00	**$15.00**	*$10.00*	*40%*
Series 6 (Apr. 93-Mar. 94)	$40.00	**$25.00**	*$15.00*	38%

But that's not all, if you order more than one series, you get an additional discount.

Series 1-5	$125.00	**$62.50 for all four**	*$62.50*	*50%*
Any 4 from Series 1-5	$100.00	**$50.00 for three**	*$50.00*	*50%*
Any 3 from Series 1-5	$75.00	**$37.50 for two**	*$37.50*	*50%*
Any 2 from Series 1-5	$50.00	**$25.00 for two**	*$25.00*	*50%*
Any 1 from Series 1-5 + Series 6	$65.00	**$32.50 for two**	*$32.50*	*50%*
Any 2 from Series 1-5 + Series 6	$90.00	**$45.00 for three**	*$45.00*	*50%*
Any 3 from Series 1-5 + Series 6	$115.00	**$57.50 for four**	*$57.50*	*50%*
Any 4 from Series 1-5 + Series 6	$140.00	**$70.00 for four**	*$70.00*	*50%*
Series 1-5 + Series 6	$165.00	**$82.50 for all five**	*$82.50*	*50%*

(Circle the items you want to order)

ACT NOW — THIS SALE ENDS NOV. 15, 1993

Customer information

Name _____

Address _____

City _____

ST_____ **ZIP** _____

Please circle the items above and return this entire page (or a copy of it) with a check or money order in payment.

TOTAL REMITTED_____

(U.S. Funds only; Texas residents add 7.75% sales tax)

Disk shipping information

Postage is included for any disk sales to U.S. addresses. **Canadian delivery**: add $2.00 for each series of disks for airmail delivery, $1.50 for surface. **Overseas delivery**: add $3.50 for each series of disks for airmail delivery; add $2.00 for each series for surface

FOR CREDIT CARD ORDERS

Credit Card No. _____

Credit Card: MC Visa (Circle One) **Exp. Date** _____

Signature _____

(credit card orders only)

BASIC

Time to review

By REGENA

I recently found a job placement form I had filled out when I graduated from college more than 20 years ago. My first choice occupation was teaching at a university, and my second choice was being a statistician for an athletic team. What I actually did was get my master's degree and become a full time mom (six children).

A couple of years ago I did get asked to be on the stats team for football and basketball for our university, so I am doing that job now, but for no pay. This year I will be doing my first choice occupation. Fall Quarter I will teach three math classes at our university, and Winter Quarter I will fill in full time for an engineering professor who is taking a one-quarter sabbatical. I hope that this may open up possibilities for teaching future classes.

My youngest child is now in school all day, so I felt I could leave home a few hours for a "real" job. After much thought, I decided it was time to make a break with the TI world. There is only so much time in the day, and the additional study to review engineering will take a lot of time — plus I am preparing for one daughter's wedding and another daughter's baby due in December.

We got our first TI99/4A for Christmas in 1980. It probably changed my life more than anything else (other than marriage and children). I wrote 16 programs and 14 articles for *99'er Magazine* in 1981-1982, along with editing and revising several programs from other authors. Most of the articles and programs were reprinted in *Home Computer Magazine* and in the book, *The Best of 99'er*.

I discovered other computer magazines paid well, so I got a VIC-20 and a Radio Shack Color Computer and wrote free-lance for several computer magazines. Eventually I added a Commodore 64, an IBM PCjr, a TRS-80 Model I, an Atari ST and an Amiga and wrote books and magazine articles about programming.

I started writing for *COMPUTE!* in 1982 and wrote 53 TI programs and articles for them from 1982 to 1986. I also wrote for *COMPUTE!*'s Commodore and IBM magazines. The TI books I wrote were *Programmer's Reference Guide to the TI99/4A, COMPUTE's First Book of TI Games* and *BASIC programs for Small Computers*. I also had five chapters in *COMPUTE's TI Collection*.

In 1983-1984 I wrote a bi-monthly column for *Enthusiast '99* published by the International Users Group.

I started writing for MICROpendium in January 1987 and have written 82 programs and articles for MICROpendium.

My writing has been how to program in BASIC, and I have tried to show a variety of uses for the TI — music, games, math, physiology, typing, quilting, graphics, languages, quizzes, amortizations, geography, flags, sports, baking, knitting, history, reading, grammar, farming, chemistry, databases, sorting, etc.

I have started you off with ideas — and hope you continue to write your own programs and continue to enjoy your TI.

The TI gave me a chance to have a home business for 13 years. The TI also enabled me to travel to visit users groups throughout the United States (and Canada and Japan) and to meet many wonderful people involved with the TI. Friendships started with the TI and carried on throughout the past several years will never be forgotten.

The TI also opened up doors to other worlds. I learned about other computers along the way. I have been teaching IBM classes and WordPerfect classes, and have been learning Macintosh applications for the high school and university labs, so I am still much involved with computers. I also have been helping our public library get connected to Internet (my address is "WHITELAW_CS@cc.suu.edu" if you use Internet electronic mail).

I have enjoyed programming the TI and writing for MICROpendium. I will still use the TI for many things — especially helping my younger children and my grandchildren. The command modules and Scott, Foresman math and reading programs are still great educational tools for young children. My older children inspired many of the educational programs I wrote for the TI, and the younger children will need the TI to use those programs as they grow.

The mathematics capabilities of the TI have a higher degree of accuracy (number of significant figures) than many other computers. And, of course, the graphics and music are great on the TI. If I have trouble with timing on a piece of music, I can put it into a program on the TI to hear how it should be.

Many times we have had a problem that seemed to fit in the computer world, and it is easier to sit down at the TI and write a quick program than to try to figure it out on a different computer or calculator. I've been trying to learn the HP-48SX programmable calculator with its Reverse Polish notation and programming in RPL, and I use the TI to check my calculator work.

Yes, the TI will continue to be a part of our lives for many more years. I hope you, too, will continue to enjoy it. Thanks to my many friends I have met through the TI!

A complete list of all TI programs and articles I have written is available for $1 from *REGENA, 918 Cedar Knolls West, Cedar City, UT 84720*. I will have one final program for MICROpendium next month.

THE ART OF ASSEMBLY — PART 28

Never do this!

By BRUCE HARRISON
© 1992 Harrison Software

There are a lot of things that could fit the title of today's column. The words come to us from an old friend named Chester Majewski. He would use a hammer to bang a car's battery terminal lug onto the post, all the while repeating "Never do this." This column could also be subtitled "I can't believe I ever did that!"

Today's column was inspired by our going back into the source code for our Word Processor to make some improvements. The source code hadn't been worked on in about a year, and then only for relatively minor adjustments. This time the aim was to make some significant additions after correcting a bug. The bug is a story in itself, but let's tell that very briefly.

My partner Dolores was the chief "guinea pig" for the Word Processor, but rarely uses it now, preferring to do her writing on one of our PC computers. One day a couple of weeks ago she decided to use her TI and my WP to write a letter. "It Crashed!"

These dreaded words reached my ears, but I was busy in the next room, and couldn't investigate just then. Later, she told me he'd made several attempts to create a document, and each time the computer went bonkers. This started me thinking. I asked whether the disk she was using for the document had a name. Back when the WP was being developed, we used only the TI Disk Manager II module to initialize disks, and that won't let the user proceed without having entered a disk name. Now, we both use the DM-1000 program to work with disks, and Dolores almost never assigns a name to the disks she initializes.

A quick check of the source code showed that indeed the Word Processor program would crash if it encountered a disk whose name was a null string. In a couple of hours, the problem was fixed, tests were run, and Dolores had a new version of WP that would tolerate the "un-named" disk. Having thus nibbled on the edges of this mass of source code, we decided that more long-needed improvements should be tackled. Ugh! More long days and nights at the computer, more endless Assembler runs and testing sessions.

THE UGLY TRUTH

There is nothing worse for an Assembly programmer than to have to go back into source code that was written a long time ago. We forget, over time, how crudely made the older stuff really was, and now see hundreds of places where what was done can be materially improved. Just the simple little things like setting a register as a word to a byte value taken from memory can easily be improved. In many places, it was like this:

```
CLR  R4      Clear a register      MOVB @LABL,R4
Move the byte      SWPB R4      Swap the bytes
```

It should be done like this:

```
MOVB @LABL,R4 Move the byte      SRL  R4,8
```

(See Page 10)

* SIDEBAR 28

```
*
* FILE ERROR HANDLING
* THIS IS NOT COMPLETE CODE, JUST SNIPPETS
* FIRST, A SAMPLE FILE READING
*
*
* CODE BY B. HARRISON
* PUBLIC DOMAIN
*
* THIS OPENS AND READS A FILE RECORD
*

OPNF   LI   R0,PAB       POINT TO PAB IN VDP
       LI   R1,PABDT     AND PAB DATA
       LI   R2,25        25 BYTES TO WRITE
       BLWP @VMBW        WRITE PAB DATA
       AI   R0,9         ADD NINE
       MOV  R0,@>8356    STASH ADDRESS
       BLWP @DSRLNK      USE DSR LINKAGE
       DATA 8
       STST R14          STORE STATUS REGISTER
       ANDI R14,>2000    MASK ALL BUT TWO BIT
       JEQ  RDFI         IF ZERO, READ RECORD
GOERR  B    @OPNERR      ELSE REPORT OPEN ERROR
RDFI   MOVB @READF,R1    SET TO READ
       LI   R0,PAB       POINT AT PAB
       BLWP @VSBW        WRITE BYTE
       AI   R0,9         ADD NINE
       MOV  R0,@>8356    STASH ADDRESS
       BLWP @DSRLNK      USE DSR
       DATA 8
       LI   R0,PAB+1     POINT AT PAB PLUS ONE
       BLWP @VSBR        READ A BYTE
       SRL  R1,13        SHIFT RIGHT 13 BITS
       JEQ  READON       IF ZERO, MOVE ON
GOERR2 B    @FILERR      ELSE REPORT ERROR
READON (PROGRAM CONTINUES)
*
* NEXT SECTION IS "OLD" METHOD FOR REPORTING ERRORS
*
OPNERR LI   R0,22*32+3   POINT AT ROW 23, COL 4
       LI   R1,OPNMSG    "FILE NOT OPENED"
       BL   @DISSTR      DISPLAY STRING WITH SUBROUTINE
       LI   R0,PAB+1     POINT AT PAB PLUS ONE
       BLWP @VSBR        READ A BYTE
       SRL  R1,13        SHIFT RIGHT 13 BITS
FILERR SLA  R1,1         SHIFT LEFT ONE BIT (DOUBLE NUMBER)
       MOV  @LUT(R1),R1  GET MESSAGE ADDRESS INTO R1
       LI   R0,23*32+4   POINT AT ROW 24, COL 5
       BL   @DISSTR      DISPLAY SELECTED MESSAGE
       BL   @KEYLOO      STOP FOR A KEYPRESS
       B    (SOMEWHERE ELSE)
*
* DATA SECTION FOR "OLD" METHOD
*
PABDT  DATA >0014,BUF,>5050,>0000,>000F
       TEXT 'DSK1.ANYOLDFILE'
*
READF  BYTE 2
*
* ERROR MESSAGE STRINGS
*
BADDEV BYTE 15
       TEXT 'BAD DEVICE NAME'
WRPROT BYTE 15
```

THE ART OF ASSEMBLY—

(Continued from Page 9)

Right-justify in the register

That takes fewer bytes to accomplish the same job. Of course the Word Processor has thousands of lines of source code (about 150 pages of listing) and perhaps a hundred or so instances where the above could be done. If that were all, it would be easy to track them all down and make the change, but of course that's not all.

When we start looking at old source code, we find truly amazing things. As our programs are developed, we make changes all over the place to fix problems or to head off potential bugs. This process, with its extreme labor-intensive nature, leaves little time for re-checking parts of the code once they seem to work correctly. In a couple of instances, we found that whole processes were being performed twice, because a change had been made in a subroutine, and no checking had been done in the part of the code that called the subroutine. We found that some files were being closed in a subroutine, and then closed again in the main code. No harm was being done, but memory and execution time were being wasted.

There were other problems to be attacked. Most of the Word Processor's code was written while your author was a relative novice in Assembly language. The handling of file errors was done crudely, and some needed error traps for file handling just didn't exist. To make matters worse, the whole matter of handling file operations was scattered all over the place in various source files. Nesting of subroutines was an intricate maze, making it difficult to track down just what subroutines were being called, and from where. Of course the human memory is a truly amazing thing, so after a while the patterns of the original source begin to make some kind of sense, albeit a rather screwball kind of sense.

This sort of thing strikes one as similar to a Marx Brothers movie. The whole thing may be absurd, but there's some consistency to it, so that each ridiculous thing that happens fits into its own "universe."

Every once in a while when revamping this kind of source code, we are tempted to just start over with a clean sheet of paper. We never give in to that temptation, partly because we realize that doing so would mean maybe a year of effort just to get back to where we were when we started. That's impractical in the extreme. Thus we struggle with our old stuff, trying to make it work better, faster, and more efficiently.

The work proceeds, but very slowly. Each small change we make bears the risk of disaster. Thus as each small change is made in the source code, another process of assembling, loading, and testing must be performed. To speed things up, we keep all the main source code and the object file on our Horizon Ramdisk. This way, it takes only about 10 minutes to assemble, but of course the testing can take much longer. We never can be quite sure whether a small "fix" has messed up something that used to work, so each time through the process we exercise all the functions of the program. Sometimes this seems a thankless task, but every now and then we make a discovery that helps our future work. One of those involves the use of the PIO port. We've found a way out of the situation when there's nothing attached to that port or when the printer is turned off. As you all know, in such situations the activity light on the RS-232 card comes on and stays on. What we found, though, is that the machine will still sense the Function-4 keystroke, and this will cause our program to "escape" from servicing the PIO port. On exit, there is an error reported in the Peripheral Access Block in VDP Ram. Getting the byte at PAB+1 and shifting it right by 13 bits gives us an error value of 6. By placing an error trap in our code, we were able to use that Function-4 keystroke as an emergency exit from a printing operation.

```
       TEXT 'WRITE PROTECTED'
BADATT BYTE 13
       TEXT 'BAD ATTRIBUTE'
ILLOP  BYTE 17
       TEXT 'ILLEGAL OPERATION'
OUTSP  BYTE 19
       TEXT 'OUT OF BUFFER SPACE'
ENDFIL BYTE 11
       TEXT 'END OF FILE'
DEVERR BYTE 12
       TEXT 'DEVICE ERROR'
FILBAD BYTE 16
       TEXT 'OTHER FILE ERROR'
*
* LOOKUP TABLE
*
       EVEN
LUT    DATA BADDEV,WRPROT,BADATT,ILLOP
       DATA OUTSP;ENDFIL,DEVERR,FILBAD
OPNMSG BYTE 17
       TEXT 'FILE DID NOT OPEN'
*
* SECOND "NEW" WAY OF REPORTING FILE ERRORS
*
OPNERR
       LI   R0,22*32+6   POINT AT ROW 23, COL 7
       LI   R1,FNOTXT    FILE NOT OPENED
       LI   R2,17        17 BYTES TO WRITE
       BLWP @VMBW        WRITE MESSAGE
       LI   R0,PAB+1     POINT AT PAB PLUS ONE
       BLWP @VSBR        READ A BYTE
       SRL  R1,13        SHIFT RIGHT 13 BITS
FILERR SLA  R1,4         SHIFT LEFT FOUR BITS (MULTIPLY BY 16)
       AI   R1,FERMSG    ADD START OF MESSAGE TABLE
       LI   R2,16        16 BYTES IN EACH MESSAGE
       LI   R0,23*32+7   POINT AT ROW 24, COL 8
       BLWP @VMBW        WRITE MESSAGE
       BL   @KEYLOO      PAUSE FOR KEYSTROKE
       B    (SOMEWHERE ELSE)
*
* DATA SECTION FOR "NEW" METHOD
*
FERMSG TEXT 'BAD DEVICE NAME '  EACH TEXT LINE 16 BYTES
       TEXT 'WRITE PROTECTED '
       TEXT 'BAD ATTRIBUTE   '
       TEXT 'BAD OPERATION   '
       TEXT 'DISK IS FILLED  '
       TEXT 'END OF FILE     '
       TEXT 'DEVICE ERROR    '
       TEXT 'OTHER FILE ERROR'
FNOTXT TEXT 'FILE DID NOT OPEN' 17 BYTES LENGTH
```

ERROR TRAPS REVISITED

Back in number 8 of this series, we showed source code for making "plain English" error reports on the screen when file operations created the error. That code works very nicely, but in our

(See Page 11)

THE ART OF ASSEMBLY—

(Continued from Page 10)

playing around with the Word Processor, we decided to take advantage of a slightly more efficient way of doing the screen reporting. Today's sidebar shows both the old and new ways. The difference in the new way is that, by making all the report texts the same length (16 bytes), and using the fact that 16 is an exact power of two, we could eliminate the lookup table used in our earlier method, and make the reporting more streamlined. This means of course that the error messages in some cases are a bit terse, but still better than "I/O ERROR 26". In the Word Processor, we supplemented this "bare bones" message with additional screen information like "PRINTING FILE DID NOT OPEN".

The method works like this. We set up the whole series of messages as sixteen character text lines with a label at the beginning of the first message. That message corresponds to the situation where the reported error code is zero, and we only know from the status register test that an error has happened. After reading the byte at PAB+1 into R1 with VSBR, we shift it right by 13 bits so that R1 contains the actual error code. Now to index into our table of messages, we shift the register left by four bits, effectively multiplying by sixteen. Adding the address of the beginning of the table of messages makes R1 point to the correct message for the error code. Now setting R2 to the common message length of 16 bytes, and R0 to the desired screen location sets us up to put the correct message on screen with VMBW. That's pretty simple, but effective. Of course if we wanted to apply the same method and make the messages longer, we'd have to make them all 32 bytes long and shift R1 left by five bits instead of four. That would chew up the memory we saved by not having a lookup table, and thus defeat the purpose of using this method in the first place.

Take your choice, as either method will produce excellent results, and will give the user a bit more information than some "error code" report, or the infamous "FATAL FILE ERROR". We always like to tease our friend Mike Maksimik about this last way of treating errors. It's one of the minor annoyances in Mike's MIDI-Master 99 that any file error gets reported on the screen as simply "FATAL FILE ERROR". We like to point out to Mike that at no time, so far as we know, has anyone died because of a file error on his TI. (At least we hope nobody has.)

We mentioned the business of a small fix causing a big problem, and here's an example. We went into the section of code called HOMIT, which allows the user to get to the top of a page with just a single keystroke (Function-H). There was a problem when trying to use this on pages with only one screenful of text. We'll skip the details, but our first attempt at fixing this one little thing caused a really big problem that we didn't discover for several days. During a test run, we tried moving a paragraph from one part of a page up to the top, and used the Function-H to put the cursor at the beginning of the page. When we then executed Function-9 to complete the move, most of the page became scrambled. Words appeared to be scattered in all the wrong places. We got out of edit mode without saving the changes to this garbled mess, and went right back to the "drawing board" in the source code.

After some groping around, we found that we'd skipped a step, so that part of our program didn't know that the cursor was sitting at the top of the page, and thus the completion of the move operation was using the wrong starting point in the screen. We were able to correct the problem, so that "homing" the cursor no longer messed things up, but another lesson had to be learned yet again. Never assume that a quick fix will be okay until all the functions have been exercised.

A FINE BOWL OF SPAGHETTI

Sometimes we find that we've made a real mess of things even when everything works. The freedom that Assembly gives us can be its own pitfall. The source code for the Word Processor is much too big to be kept in one file, so it's "organized" into editable files from WORDA through WORDP. During its development, many subroutines were developed, and sometimes these were simply placed in whatever of the source files was the smallest, regardless of this subroutine's relationships to other parts of the program. That, plus the "nesting," can make tracing through some operations a real nightmare.

For example, the main code for getting into the edit mode is in the WORDG file. This calls a subroutine that's in WORDJ, then that subroutine calls one that's in WORDG, which in turn calls subroutines in WORDD, and so on. In programming parlance, this is called "spaghetti code," since it's all mixed up and twisted upon itself. If we did this in Extended BASIC, where our users could list the program and see how messy it really is, we'd never release it. In Assembly, we can hide behind the fact that it all works correctly, and few of our users will try to disassemble the code once it's working.

As we write this, it's "take a break" time in our assembly work on the Word Processor. Jim Peterson has just issued another of those little "challenges" that we sometimes take up, so maybe we'll put the WP to bed again for a while, release Version 2.9, and start on Jim's challenge. Of course we already had about four other projects waiting in line before starting this round with the WP. Maybe we'll keep trying for Geneve compatibility, but then there's this game that our little boy wants me to create for him, and then again there's ....

Now perhaps you see what we mean by "Never Do This!" Maybe that should apply to the whole idea of being a programmer.

Chicago group makes changes

The Chicago TI Users Group has moved its meeting place to the Illinois Institute of Technology, according to Don Walden, the group's president.

Walden says the group is in the process of becoming certified as a student organization. To do this, the group has to get approximately 20 members from the student body. Mike Maksimik, a member of the group who is a graduate student at Illinois Tech, is recruiting student members, Walden says.

Walden, Maksimik and Don Jones have taken over production of the group's newsletter, the Chicago TImes Walden says. He notes that Jones' Geneve articles, which have been requested by many readers, will resume, and that Cal Zanella has also agreed to write articles for the newsletter.

Walden says he expects the next issue of the newsletter to be of "high quality, both in content and appearance."

Extended BASIC

A card index on your TI

By BILL GASKILL

Have you ever wished that the 99/4A had a computerizedindex card system with enough capacity to store all those User Notes that you have come across over the years? I have, but I could never find a program that was quite right for the job. So, like any loyal 99er I wrote one, and it's called CARD FILE.

CARD FILE will consume only some 33 sectors on a floppy disk, but it requires seven floppies to support the 676 possible index cards you can have. The TI's 127 files per disk limitation is the reason why seven disks are needed. It's not caused by the program, necessarily. Two drives are preferred in order to use CARD FILE, but it can also be used on single disk drive systems. You just have to swap disks more often.

BEFORE YOU BEGIN:

In order to use CARD FILE you must have Brad Snyder's 40-Column Utilities. Send $8 to Brad Snyder, 148 Ave. A, Palmerton, PA 18071 if you need to obtain a copy.

When you have keyed in this program, save it as CARD and then have the 40-Column Util program RUN "DSK1.CARD".

Before running the CARD program there are a couple of setup duties that you will need to attend to.

1. Load your disk manager and format six diskettes using the names
AD EH IL MP QT and UZ
on them. This means that disk 1 will be named AD, disk 2 will be named EH and so on. These disks are the CARD disks that will be used to house the 676 index cards that CARD FILE is capable of storing. *The disk names are important,* so you will want to get them right.

The seventh disk is used to house the CARD program, the 40-Col LOAD program from Brad Snyder and the Insert Files. This is what I refer to as the PROGRAM disk, and it may be named anything you want. CARD FILE assumes that the PROGRAM disk will always be found in DSK1, thus disk name does not matter here.

PROGRAM DESCRIPTION:

CARD FILE is designed to emulate a box of 3x5-inch index cards, complete with the alphabetical inserts that are used to divide each letter of the alphabet. There are 26 inserts per box and enough room in the box to have 26 cards behind each insert.

Before placing any index cards in the box you must first create the inserts. This is done by pressing Fctn 5 at the title screen to the program and then pressing Fctn 6 to confirm creation of a new set of inserts. Two steps are required for this process because you could wipe out any existing inserts if Fctn 5 were pressed and you did not intend to do so. *Do not* press Fctn 5 after inserts have been created or you will overwrite all existing insert data.

If you press a letter key before the inserts are created the disk drive will light up, it will not find the inserts and then it will just restart the program. If this happens, it's because the inserts have not been created.

Besides separating the various index cards into alphabetical categories, the inserts are used to record names or descriptions for the 26 index cards that will be stored behind them. Each index card may have a 15-character "name,"which is the description of what is on the card. This "name" is the same information that you would normally write at the top of a paper index card to describe what the card is all about. Each insert has room for 26 index card "names" on it.

The Insert files are named using the numbers 65-90, so your PROGRAM disk should have 26 new files on it after the Create routine is used. Once you have created the inserts the title screen is redisplayed. You may press Fctn 6 at this point to configure the program to your hardware is you wish. The data drive may be changed from DSK1 to another drive and PIO may be changed to another configuration. If you have a two-drive system, by all means change the DS$ variable to DSK2. and then resave the program.

Once you are ready to create an index card, press a key that corresponds with the insert that you wish to flip to in the box. For example, if you were going to record various User Notes and tips about TI-Writer, you would probably want to file them under T for TI-Writer. So, you would press T. Once you do so, CARD FILE will read the insert for T and display the names of all cards that are on file behind that insert. If all cards are blank then you will see just the letters A-Z displayed, with no names recorded.

To create an index card choose any of the letters displayed by pressing the key for that letter. You *do not* have to organize a card by the letters that you see displayed, meaning you do not have to have something under the A category that begins with the letter A. You can put anything you want under A. The letters only exist to allow single key press access to a card behind the insert. I could have used numbers instead of letters, but then we would have been limited to ten cards behind each insert.

Once you press a letter key, CARD FILE looks to see if an index card is on file for that key by reading the disk in the DATA drive. If a card is not on file, a message is displayed telling you so and the program is placed in the edit mode. If a card does exist then the card is read into memory and side one of the card is displayed. Each index card has a front and back side. Each side allows up to 18 lines of text with 36 characters per line.

CREATING/PRINTING/SAVING:

The CARD FILE editor is the part of the program where cards are created, printed and saved. You cannot delete a card once it has been created, but you can replace it with another card by pressing Fctn 4 to purge the current text, and then typing in the new text to be saved on that card. When you press S to save, a new index name is asked for that will overwrite

(See Page 13)

CARD FILE—

(Continued from Page 12)

the name of the card you just purged the text from.

The index card editor uses a command mode and a text mode to provide all text and file processing features. The text mode is active when the cursor is flashing. The command mode is active when the cursor is not visible on the screen. Fctn X is used to toggle between the two modes.

Text mode is used to enter text that is to be saved or printed. Command mode is used to access any of the commands listed in the menu at the base of the screen. Commands are accessed by pressing the first letter of the command. For example, one would press H to read the Help screen.

Cursor movement is done with the arrow keys and the ENTER key. Fctn E takes the cursor back one line at a time, while ENTER advances the cursor one line at a time. Fctn S and Fctn D move the cursor right and left within a line of text.

MENU OPTIONS:

CLEAR erases the screen.

DELETE physically removes a line of text from the file. You may use Fctn 3 to erase a line, but must use DELETE to actually remove it so that the next lowest line on the screen takes the deleted line's place in the file. To use DELETE, simply place the cursor at the line to be deleted and then press Fctn X to access the command mode. Press D to Delete and the line will be removed.

END displays the back of the card and places the cursor on the last lines

FILE displays the path/file name of the active index card in its native format, meaning its real name. For example, if you are working with the A insert and you selected card A, the path and file name display will read DSK1.6565.

HELP displays a single help screen with explanations for the command menu options. Its display has no effect on the current document screen.

INSERT physically "pushes' each line of text downward to make room for a blank line. You can insert a blank line at any point in a file except line number 1. To use INSERT simply place the cursor at the line where a blank line is to appear, press Fctn X to access the command mode, and then press I to Insert.

Both Insert and Delete affect all lines of text after the line being deleted or inserted. That means that deleting a line from side 1 pulls the contents of side 2 a line closer to the bottom of side 1. CARD FILE sees 36 lines lines of text when it stores information. It does not see a side 1 and 2 to each card.

LOAD restores the title screen so that you may load another insert or another card from the same insert. If an index card has not been saved you will be asked if you wish to save it prior to the exit.

PRINT first prompts you for the Card Title, which is used only if you are going to be printing a report. If you press <ENTER> without entering a title, 40-Column Utilities' resident screen dump is invoked, which will allow the contents of the current screen to be printed on a 3x5-inch index card. If a title is entered, a report will be printed. After Card Title is passed you're asked to enter the number of tab spaces to use from the left edge of the card or paper.

When report printing begins, CARD FILE prints the data in a newspaper-column format with the front and back sides of the card appearing next to each other. Three cards can be printed on a single sheet of paper.

QUIT prompts you for a Y/N key press to confirm that you *do* wish to exit the program. Any key press not a Y will return the program to the point you left when Q was pressed. You will *not* be warned to save an unsaved card.

SAVE first prompts you to enter an index name for the card that is being filed. This name may be up to 15 characters long. Once you enter an index name the program saves that name in the insert file on the disk in drive 1, and then looks at the designated data disk drive. It reads the name of the disk to ensure that it is the correct one to be saving the current index card on. If it is, then the card is written to disk. If not, CARD FILE will tell you the name of the correct disk to insert into the data disk drive. Once you do so and press Fctn 8 the program will read the new disk to verify that it's the correct one and then (assuming it is the right disk) will save the index card. If you wish to override the use of disk names and instead be able to save your index card to any disk, press Fctn 6 when you are prompted to insert the correct disk. This bypasses the routine that is used to do the disk name checking.

TOP displays side 1 of the current card, from wherever the cursor is, and places the cursor on line 1 side 1 of the card.

CARDFILE

```
1 !card 10/27/91
    Bill Gaskill
  Grand Junction, CO 81506
        !113
100 CALL KEY(3,K,S):: CALL L
INK("CLS"):: CALL LINK("TEXT
",16,5):: CALL LINK("SCINIT"
,1,1,40,18):: GOSUB 1000 ::
CALL LINK("LOWCAS")!050
110 OPTION BASE 1 :: DIM A$(
40),B$(40):: DR$="DSK1." ::
DS$="DSK2." :: PR$="PIO" ::
ON BREAK NEXT :: ON WARNING
NEXT :: ID$="" !099
120 CALL CHAR(124,"0010F8848
4F81000",126,"00FF",129,"FF0
0000000000000040404040404040
40000000000FF00FF80808080808
08080")!111
130 GOSUB 1100 :: CALL HORZ(
1,3,129,15):: CALL VERT(1,2,
130,5):: CALL HORZ(5,3,131,1
):: CALL VERT(1,17,132,2)!03
9
140 CALL HORZ(3,5,129,16)::
CALL VERT(3,4,130,6):: CALL
HORZ(8,5,131,1):: CALL VERT(
3,21,132,2)!017
150 CALL HORZ(5,7,129,18)::
CALL VERT(5,6,130,7):: CALL
HORZ(11,7,131,1):: CALL VERT
(5,25,132,2)!079
160 CALL HORZ(7,9,129,20)::
CALL VERT(7,8,130,8):: CALL
HORZ(14,9,131,1):: CALL VERT
(7,29,132,2)!092
170 CALL HORZ(9,11,129,22)::
 CALL VERT(9,10,130,9):: CAL
L HORZ(17,11,131,23):: CALL
VERT(9,33,132,9)!029
180 CALL DISP(10,17,"CARD FI
LE"):: CALL DISP(13,14,"Pres
s the letter")!051
190 CALL DISP(14,14,"of the
```

(See Page 14)

CARD FILE—

(Continued from Page 13)

```
insert to"):: CALL DISP(15,1
4,"display:")!219
200 CALL HORZ(19,1,126,40)::
 CALL HORZ(21,1,32,160):: CA
LL DISP(24,2,"Press Fctn 5-C
reate, Fctn 6-Configure")!23
7
210 FOR I=1 TO 40 :: A$(I)="
" :: NEXT I :: CALL KEY(5,K,
S):: ID$="" !233
220 CALL DISP(20,8,"A B C D
E F G H I J K L M"):: CALL D
ISP(21,8,"N O P Q R S T U V
W X Y Z")!084
230 CALL KEY(3,H,S):: IF H=1
2 THEN 1140 ELSE IF H=14 THE
N 1160 ELSE IF H<65 OR H>90
THEN 230 !185
240 D$=STR$(H):: CALL CHAR(1
35,D$):: CALL CHARPAT(135,D$
):: CALL DISP(15,23,CHR$(H))
!096
250 D$=SEG$(D$,1,2):: F$=DR$
&D$ :: ON ERROR 1250 :: OPEN
 #1:F$,INPUT ,DISPLAY ,VARIA
BLE !154
260 FOR F=1 TO 26 :: LINPUT
#1:B$(F):: NEXT F :: CLOSE #
1 :: GOSUB 1100 :: R=3 :: C=
2 :: G=65 !191
270 FOR F=1 TO 26 :: CALL DI
SP(R,C,CHR$(G)&". "&B$(F)):
 R=R+1 :: IF R>15 THEN C=22
:: R=3 !110
280 G=G+1 :: NEXT F :: GOSUB
 1130 :: CALL DISP(20,7,"Pre
ss letter of card to read"):
: CALL DISP(21,7,"from inser
t:")!094
290 CALL DISP(21,20,CHR$(H))
:: CALL KEY(3,X,Y):: IF Y=0
THEN 290 ELSE IF X=15 THEN 1
00 ELSE IF X<65 OR X>90 THEN
 290 !111
300 I$=STR$(X):: P$=DS$&D$&I
$ :: CALL DISP(21,22,CHR$(X)
):: GOTO 740 !026
310 CALL DISP(24,1,"      Pr
ess ENTER to begin typing ")
:: I,M,R=1 :: GOSUB 780 :: G
OSUB 1090 !110
320 CALL KEY(5,K,S):: CALL D
ISP(23,36,STR$(I)):: CALL LI
NK("ACCEPT",R,2,-38,"",A$(I)
,G):: M=I !025
330 GOSUB 1110 :: IF G=11 TH
EN 340 ELSE IF G=10 THEN 400
 ELSE 360 !161
340 IF I<=1 THEN 320 ELSE I=
I-1 :: R=R-1 :: G=0 !171
350 IF R=0 AND I>=1 THEN CAL
L LINK("SCROLL",2):: R=R+1 :
: CALL DISP(1,2,A$(I)):: RS=
I :: RE=I+17 :: GOTO 320 ELS
E 320 !130
360 IF R=18 AND I<36 THEN CA
LL LINK("SCROLL"):: R=R-1 ::
 CALL DISP(18,2,A$(I+1)):: R
S=I-16 :: RE=I+1 :: GOTO 400
 ELSE 400 !216
370 GOSUB 1100 :: RO=1 :: FO
R I=RS TO RE :: CALL DISP(RO
,2,A$(I)):: RO=RO+1 :: NEXT
I !192
380 IF K=69 THEN I=36 ELSE I
=RS !221
390 GOTO 320 !144
400 IF G=10 THEN GOSUB 850 :
: GOSUB 780 :: GOTO 320 !252
410 IF I=36 THEN 430 !230
420 R=R+1 :: I=I+1 :: GOSUB
1090 :: GOTO 320 !212
430 CALL DISP(24,1,"
 Escape Print Save.
    "):: CALL KEY(3,S,K)!224
440 IF K=0 THEN 430 ELSE IF
S=69 THEN 770 ELSE IF S=80 T
HEN 450 ELSE IF S=83 THEN 55
0 ELSE 430 !228
450 CALL KEY(3,K,S):: GOSUB
1130 :: CALL DISP(21,1,"  Ca
rd Title:"):: CALL DISP(22,1
,"  Tab spaces from left mar
gin:2")!152
460 CALL DISP(24,1,"  Omit c
ard title to print screen on
ly"):: CALL ACCEPT(21,14,20,
"",RT$):: CALL ACCEPT(22,31,
-1,"0123456789",T$)!093
470 IF T$="" THEN 780 ELSE T
B=VAL(T$):: ON ERROR 890 !16
5
480 IF RT$="" THEN 1260 ELSE
 OPEN #1:PR$,OUTPUT :: GOSUB
 1120 :: CALL DISP(24,1,"  P
rinting...")!175
490 PRINT #1:TAB(TB);RT$ ::
PRINT #1 :: FOR I=1 TO 18 !2
05
500 IF LEN(A$(I))<>38 THEN A
$(I)=A$(I)&RPT$(" ",38-LEN(A
$(I)))!046
510 IF LEN(A$(I+18))<>38 THE
N A$(I+18)=A$(I+18)&RPT$(" "
,38-LEN(A$(I+18)))!254
520 IF LEN(A$(I))=0 THEN A$(
I)=RPT$(" ",38)!063
530 IF LEN(A$(I+18))=0 THEN
A$(I+18)=RPT$(" ",38)!039
540 PRINT #1:TAB(TB);A$(I)&"
  "&A$(I+18);:: NEXT I :: PR
INT #1 :: PRINT #1 :: CLOSE
#1 :: I=M :: GOSUB 1130 :: G
OTO 780 !236
550 CALL DISP(24,3,"Enter ca
rd index:[              ]")
:: CALL ACCEPT(24,21,-15,"",
ID$):: IF ID$="" THEN 320 !2
14
560 I$=SEG$(P$,8,2):: I=VAL(
I$):: I=I-64 :: B$(I)=ID$ ::
 ON ERROR 900 !106
570 OPEN #1:F$,OUTPUT,DISPLA
Y ,VARIABLE :: FOR F=1 TO 26
 :: PRINT #1:B$(F):: NEXT F
:: CLOSE #1 !163
580 IF DS$="DSK1." THEN GOSU
B 1240 ELSE GOTO 590 !006
590 IF H>=65 AND H<=68 THEN
H$="AD" :: GOTO 650 !188
600 IF H>=69 AND H<=72 THEN
H$="EH" :: GOTO 650 !195
610 IF H>=73 AND H<=76 THEN
H$="IL" :: GOTO 650 !202
620 IF H>=77 AND H<=80 THEN
H$="MP" :: GOTO 650 !209
630 IF H>=81 AND H<=84 THEN
H$="QT" :: GOTO 650 !216
640 IF H>=85 AND H<=90 THEN
H$="UZ" !134
650 OPEN #5:DS$,INPUT ,RELAT
IVE,INTERNAL :: INPUT #5:N$,
N,N,N :: CLOSE #5 :: IF N$=H
$ THEN 690 !195
660 GOSUB 1130 :: CALL DISP(
21,3,H$):: CALL DISP(21,6,"d
isk is not in the drive. Ins
ert")!175
670 CALL DISP(22,3,"the corr
ect disk, then press F8 or
    press F6 to override."):
: CALL KEY(3,K,S):: IF S=0 T
HEN 670 !066
```

(See Page 15)

CARD FILE—

(Continued from Page 14)

```
680 IF K=12 THEN 690 ELSE IF
 K<>6 THEN 670 ELSE GOSUB 11
30 :: GOTO 650 !106
690 OPEN #2:P$,OUTPUT,DISPLA
Y ,VARIABLE :: FOR I=1 TO 40
 !083
700 PRINT #2:A$(I):: I$=STR$
(I):: CALL DISP(23,36,I$)::
NEXT I :: CLOSE #2 :: I=M !1
76
710 IF DS$="DSK1." THEN 720
ELSE 730 !123
720 CALL DISP(24,3,"  Inser
t program disk, press F6"):
 CALL KEY(3,X,Y):: IF X<>12
THEN 720 !244
730 GOSUB 1100 :: GOTO 780 !
128
740 IF DS$="DSK1." THEN GOSU
B 1240 ELSE GOTO 750 !166
750 ON ERROR 880 :: OPEN #2:
P$,INPUT ,DISPLAY ,VARIABLE
!254
760 FOR I=1 TO 40 :: LINPUT
#2:A$(I):: I$=STR$(I):: CALL
 DISP(23,36,I$):: NEXT I ::
CLOSE #2 :: I=1 !089
770 GOSUB 1100 :: GOSUB 1120
 :: FOR I=1 TO 18 :: CALL DI
SP(I,2,A$(I)):: NEXT I :: GO
TO 310 !127
780 CALL DISP(20,1,"  Clear
 Delete End File Help Insert
          Load Print Quit
Save Top")!098
790 CALL KEY(3,K,S):: IF S=0
 THEN 790 ELSE IF K=188 THEN
 1300 ELSE IF K=2 THEN 1080
ELSE IF K=49 THEN 820 ELSE I
F K=50 THEN 820 !012
800 IF K=13 THEN 320 ELSE IF
 K=10 THEN 320 ELSE IF K=11
THEN 320 ELSE IF K<65 OR K>9
0 THEN 780 ELSE 810 !151
810 ON K-64 GOTO 790,790,730
,910,860,990,790,1060,920,79
0,790,1200,790,790,790,450,9
70,790,550,870,790,790,790,7
90,790,1260 !218
820 IF K=49 THEN RS=1 :: RE=
18 :: R=1 :: GOTO 370 !171
830 IF K=50 THEN RS=19 :: RE
=36 :: R=1 :: GOTO 370 !221
840 RETURN !136
850 CALL DISP(24,1,"  Pres
s: C,D,E,F,H,I,L,P,Q,S or T
  "):: RETURN !250
860 R=18 :: RS=19 :: RE=36 :
: GOTO 370 !170
870 I,R=1 :: RS=1 :: RE=18 :
: GOTO 370 !051
880 GOSUB 1130 :: CALL DISP(
23,1,"      * Card not on
 file!  *"):: I,R=1 :: GOSU
B 1100 :: GOTO 310 !043
890 GOSUB 1110 :: CALL DISP(
23,1,"      * Printer err
or!    *"):: GOTO 320 !093
900 GOSUB 1110 :: CALL DISP(
23,1,"      * Cannot save
 card!  * "):: GOTO 320 !22
0
910 GOSUB 950 :: FOR I=I TO
37 :: A$(I)=A$(I+1):: CALL D
ISP(23,1," <Del>"):: NEXT I
:: I=I-1 :: GOSUB 1100 :: GO
SUB 930 :: GOTO 320 !170
920 GOSUB 950 :: FOR I=37 TO
 I STEP -1 :: A$(I)=A$(I-1):
: CALL DISP(23,1," <Ins>"):
 NEXT I :: A$(M)=" " :: GOSU
B 1100 :: GOSUB 930 :: GOTO
320 !231
930 IF RS=0 THEN RS=1 :: RE=
18 !247
940 RO=1 :: FOR I=RS TO RE :
: CALL DISP(RO,2,A$(I)):: RO
=RO+1 :: NEXT I :: CALL HORZ
(R,1,32,1):: I=M :: GOTO 320
 !175
950 IF I=1 AND K=73 THEN 960
 ELSE CALL HORZ(R,1,124,1)::
 RETURN !030
960 GOSUB 1110 :: CALL DISP(
23,1,"     * Can't insert a
line there. *"):: GOTO 320 !
002
970 CALL DISP(24,1,"  Are yo
u sure?  (Y/N)
  ")!254
980 CALL KEY(3,X,Y):: IF Y=0
 THEN 980 ELSE IF X=78 THEN
320 ELSE IF X<>89 THEN 980 E
LSE END !184
990 GOSUB 1110 :: CALL DISP(
23,1," Active Card:"):: CALL
 DISP(23,15,P$):: RETURN !17
6
1000 CALL CLEAR :: CALL HCHA
R(1,1,42,31):: CALL VCHAR(1,
2,42,23):: CALL HCHAR(23,1,4
2,31):: CALL VCHAR(1,31,42,2
3)!001
1010 DISPLAY AT(6,1):"C -cle
ars the current screen":"D -
deletes a line of text":"E -
goes to end of card" !123
1020 DISPLAY AT(9,1):"F -dis
plays card path/name":"I -in
serts a blank line":"L -disp
lays title screen" !177
1030 DISPLAY AT(12,1):"P -pr
ints the current card":"Q -e
xits Card File program":"S -
saves card to disk" !120
1040 DISPLAY AT(15,1):"T -go
es to top of card":"1 -displ
ays side 1 of card":"2 -disp
lays side 2 of card":"F4-pur
ges current card" !020
1050 DISPLAY AT(24,1):"  Pre
ss <ENTER> to return." :: RE
TURN !210
1060 CALL LINK("NORM"):: CAL
L KEY(0,S,K):: IF S<>13 THEN
 1060 :: CALL LINK("TEXT",16
,5):: GOTO 320 !044
1070 GOSUB 1100 :: FOR I=RS
TO RE :: CALL DISP(RO,2,A$(I
)):: RO=RO+1 :: NEXT I :: I=
RE :: R=18 :: RETURN !212
1080 GOSUB 1120 :: GOSUB 110
0 :: CALL DISP(24,1,"  Purgi
ng...."):: FOR I=1 TO 40 ::
A$(I)="" :: NEXT I :: ID$=""
 :: I,R=1 :: GOSUB 1120 :: G
OTO 780 !093
1090 CALL DISP(24,1,"
Use Fctn X to exit text mode
    "):: RETURN !230
1100 CALL HORZ(1,1,32,720)::
 RETURN !195
1110 CALL HORZ(23,1,32,40)::
 RETURN !194
1120 CALL HORZ(24,1,32,40)::
 RETURN !195
1130 CALL HORZ(20,1,32,200):
: RETURN !238
1140 GOSUB 1130 :: CALL DISP
(22,3,"Data drive :"):: CALL
 DISP(23,3,"Printer:"):: CAL
L DISP(22,15,DS$):: CALL DIS
P(23,11,PR$)!211
```

(See Page 16)

CARD FILE—

(Continued from Page 14)

```
680 IF K=12 THEN 690 ELSE IF
 K<>6 THEN 670 ELSE GOSUB 11
30 :: GOTO 650 !106
690 OPEN #2:P$,OUTPUT,DISPLA
Y ,VARIABLE :: FOR I=1 TO 40
 !083
700 PRINT #2:A$(I):: I$=STR$
(I):: CALL DISP(23,36,I$)::
NEXT I :: CLOSE #2 :: I=M !1
76
710 IF DS$="DSK1." THEN 720
ELSE 730 !123
720 CALL DISP(24,3,"   Inser
t program disk, press F6"):::
 CALL KEY(3,X,Y):: IF X<>12
THEN 720 !244
730 GOSUB 1100 :: GOTO 780 !
128
740 IF DS$="DSK1." THEN GOSU
B 1240 ELSE GOTO 750 !166
750 ON ERROR 880 :: OPEN #2:
P$,INPUT ,DISPLAY ,VARIABLE
!254
760 FOR I=1 TO 40 :: LINPUT
#2:A$(I):: I$=STR$(I):: CALL
 DISP(23,36,I$):: NEXT I ::
CLOSE #2 :: I=1 !089
770 GOSUB 1100 :: GOSUB 1120
 :: FOR I=1 TO 18 :: CALL DI
SP(I,2,A$(I)):: NEXT I :: GO
TO 310 !127
780 CALL DISP(20,1,"   Clear
 Delete End File Help Insert
          Load Print Quit
Save Top")!098
790 CALL KEY(3,K,S):: IF S=0
 THEN 790 ELSE IF K=188 THEN
 1300 ELSE IF K=2 THEN 1080
ELSE IF K=49 THEN 820 ELSE I
F K=50 THEN 820 !012
800 IF K=13 THEN 320 ELSE IF
 K=10 THEN 320 ELSE IF K=11
THEN 320 ELSE IF K<65 OR K>9
0 THEN 780 ELSE 810 !151
810 ON K-64 GOTO 790,790,730
,910,860,990,790,1060,920,79
0,790,1200,790,790,790,450,9
70,790,550,870,790,790,790,7
90,790,1260 !218
820 IF K=49 THEN RS=1 :: RE=
18 :: R=1 :: GOTO 370 !171
830 IF K=50 THEN RS=19 :: RE
=36 :: R=1 :: GOTO 370 !221
840 RETURN !136
850 CALL DISP(24,1,"   Pres
s: C,D,E,F,H,I,L,P,Q,S or T
   "):: RETURN !250
860 R=18 :: RS=19 :: RE=36 :
: GOTO 370 !170
870 I,R=1 :: RS=1 :: RE=18 :
: GOTO 370 !051
880 GOSUB 1130 :: CALL DISP(
23,1,"       * Card not on
 file!   *"):: I,R=1 :: GOSU
B 1100 :: GOTO 310 !043
890 GOSUB 1110 :: CALL DISP(
23,1,"       * Printer err
or!     *"):: GOTO 320 !093
900 GOSUB 1110 :: CALL DISP(
23,1,"       * Cannot save
 card!   * "):: GOTO 320 !22
0
910 GOSUB 950 :: FOR I=I TO
37 :: A$(I)=A$(I+1):: CALL D
ISP(23,1," <Del>"):: NEXT I
:: I=I-1 :: GOSUB 1100 :: GO
SUB 930 :: GOTO 320 !170
920 GOSUB 950 :: FOR I=37 TO
 I STEP -1 :: A$(I)=A$(I-1):
: CALL DISP(23,1," <Ins>"):::
 NEXT I :: A$(M)=" " :: GOSU
B 1100 :: GOSUB 930 :: GOTO
320 !231
930 IF RS=0 THEN RS=1 :: RE=
18 !247
940 RO=1 :: FOR I=RS TO RE :
: CALL DISP(RO,2,A$(I)):: RO
=RO+1 :: NEXT I :: CALL HORZ
(R,1,32,1):: I=M :: GOTO 320
 !175
950 IF I=1 AND K=73 THEN 960
 ELSE CALL HORZ(R,1,124,1)::
 RETURN !030
960 GOSUB 1110 :: CALL DISP(
23,1,"     * Can't insert a
line there. *"):: GOTO 320 !
002
970 CALL DISP(24,1,"  Are yo
u sure?   (Y/N)
   ")!254
980 CALL KEY(3,X,Y):: IF Y=0
 THEN 980 ELSE IF X=78 THEN
320 ELSE IF X<>89 THEN 980 E
LSE END !184
990 GOSUB 1110 :: CALL DISP(
23,1," Active Card:"):: CALL
 DISP(23,15,P$):: RETURN !17
6
1000 CALL CLEAR :: CALL HCHA
R(1,1,42,31):: CALL VCHAR(1,
2,42,23):: CALL HCHAR(23,1,4
2,31):: CALL VCHAR(1,31,42,2
3)!001
1010 DISPLAY AT(6,1):"C -cle
ars the current screen":"D -
deletes a line of text":"E -
goes to end of card" !123
1020 DISPLAY AT(9,1):"F -dis
plays card path/name":"I -in
serts a blank line":"L -disp
lays title screen" !177
1030 DISPLAY AT(12,1):"P -pr
ints the current card":"Q -e
xits Card File program":"S -
saves card to disk" !120
1040 DISPLAY AT(15,1):"T -go
es to top of card":"1 -displ
ays side 1 of card":"2 -disp
lays side 2 of card":"F4-pur
ges current card" !020
1050 DISPLAY AT(24,1):"  Pre
ss <ENTER> to return." :: RE
TURN !210
1060 CALL LINK("NORM"):: CAL
L KEY(0,S,K):: IF S<>13 THEN
 1060 :: CALL LINK("TEXT",16
,5):: GOTO 320 !044
1070 GOSUB 1100 :: FOR I=RS
TO RE :: CALL DISP(RO,2,A$(I
)):: RO=RO+1 :: NEXT I :: I=
RE :: R=18 :: RETURN !212
1080 GOSUB 1120 :: GOSUB 110
0 :: CALL DISP(24,1,"  Purgi
ng...."):: FOR I=1 TO 40 ::
A$(I)="" :: NEXT I :: ID$=""
 :: I,R=1 :: GOSUB 1120 :: G
OTO 780 !093
1090 CALL DISP(24,1,"
Use Fctn X to exit text mode
     "):: RETURN !230
1100 CALL HORZ(1,1,32,720)::
 RETURN !195
1110 CALL HORZ(23,1,32,40)::
 RETURN !194
1120 CALL HORZ(24,1,32,40)::
 RETURN !195
1130 CALL HORZ(20,1,32,200):
: RETURN !238
1140 GOSUB 1130 :: CALL DISP
(22,3,"Data drive :"):: CALL
 DISP(23,3,"Printer:"):: CAL
L DISP(22,15,DS$):: CALL DIS
P(23,11,PR$)!211
```

(See Page 16)

CARD FILE—

(Continued from Page 15)

```
1150 CALL ACCEPT(22,15,-5,""
,DS$):: CALL ACCEPT(23,11,-2
5,"",PR$):: GOSUB 1130 :: GO
TO 200 !126
1160 CALL DISP(24,1,"  Press
 F6 to confirm insert creati
on   "):: CALL KEY(3,K,S)::
IF S=0 THEN 1160 ELSE IF K<>
12 THEN 100 !055
1170 CALL DISP(24,2," Creati
ng insert:
  "):: N=65 :: FOR I=1 TO 26
 :: CALL DISP(24,20,CHR$(N))
!207
1180 OPEN #1:DR$&STR$(N):: F
OR J=1 TO 26 :: PRINT #1:" "
 :: NEXT J :: CLOSE #1 !115
1190 N=N+1 :: NEXT I :: GOTO
 100 !171
1200 GOSUB 1130 :: IF ID$<>"
" THEN 130 ELSE CALL DISP(24
,3,"Save this index card? Y/
N")!061
1210 CALL KEY(3,K,S):: IF S=
0 THEN 1210 ELSE IF K=78 THE
N 1220 ELSE IF K=89 THEN 550
 ELSE IF K=15 THEN 780 ELSE
1210 !069
1220 IF DS$="DSK1." THEN 123
0 ELSE 130 !033
1230 CALL DISP(24,3,"  Inser
t program disk, press F6
  "):: CALL KEY(3,Y,S):: IF
Y<>12 THEN 1230 :: GOTO 130
!232
1240 CALL DISP(24,3,"     In
sert data disk, press F6
  "):: CALL KEY(3,U,V):: IF
U<>12 THEN 1240 ELSE RETURN
!165
1250 RUN !169
1260 E$=PR$&".CR" :: GOSUB 1
130 :: CALL LINK("DUMP",E$,0
,0,1,TB):: GOTO 780 !0671270
 SUB ACCEPT(R,C,S,C$,R$):: C
ALL LINK("ACCEPT",R,C,S,C$,R
$):: SUBEND !142
1280 SUB DISP(R,C,E$):: CALL
 LINK("DISP",R,C,E$):: SUBEN
D !048
1290 SUB HORZ(R,C,CH,RP):: C
ALL LINK("HORZ",R,C,CH,RP)::
 SUBEND !068
1300 SUB VERT(R,C,CH,RP):: C
ALL LINK("VERT",R,C,CH,RP)::
 SUBEND !064
```

Take on the computer in a game of Bull Run

The following article and program were written by Stephen Shaw of the TI99/4A User Group United Kingdom.

This strategic game is played out on a 6x6 grid, with rows identified by the letters A-F and columns identified by the numbers 1-6.

At the start of the game, the human player allocates a total of 100 men around the 36 squares, which may have from zero to 9 men in each. After the men have been allocated, the computer allocates its men.

The computer, having the handicap of no brains, always goes first.

Each player in turn selects one square as the center of a battle, which will rage over all adjacent squares, with a maximum of nine squares in the center. Or, if the battle is centered on one of the four corner squares, only four squares are involved. Six squares are involved for any other edge square.

The total men for each player is calculated for all these actives squares, and the winner is the player with the most men. The loser sees all his men in the active squares wiped out.

Play terminates when one player has more than twice the number of men that his opponent has, or after 18 battles. The winner is the player in control of most squares at game's end. The winner may even end up with fewer men than his opponent.

It is easy to play for a draw on territory. So, as a penalty, in the event each player controls the same territory as the other, the games is awarded to the computer. It doesn't pay to play for a tie.

Your playing strategy takes two parts:

- What is the best arrangement of you 100 men to tackle any computer play?
- What is the best pattern of battle to take territory or to defend territory, or to grab the most computer men?

There is a winning strategy which will usually beat the computer. How fast can you find it?

As written, beating the computer may be difficult. Various adjustments may be made to make the computer play more intelligently. Or less so. I will leave this to the reader.

BASIC programmers may note how the board is placed rapidly on the screen, and how player input is handled. Notice the use of RND*RND as the computer places its men. Can you see why? You may manipulate distribution patterns by using more than one RND — RND*RND or (RND+RND)/2 or even (RND*RND+RND)/2.

Enjoy!

BULLRUN

```
100 REM!154
110 REM based on FIRST BULL
  RUN by Tim hartnell 1984 !
183
120 REM for TI99/4A by s sha
w  october 1990 !086
130 REM winner has most terr
itory at end not most troops
 !007
140 REM!154
150 REM JUST NEEDS EX BAS! !
099
160 REM!154
170 REM PLACE UP TO 9 TROOP
   IN EACH TERRITORY   TROOP
S IN BLOCK OF ARE COUNTED FO
R RESULT !094
180 REM!154
190 CALL CLEAR :: GOSUB 290
:: T=0 :: RANDOMIZE !106
200 GOSUB 350 !175
```

(See Page 17)

BULLRUN—

(Continued from Page 16)

```
210 GOSUB 610 :: IF FLAG=1 T
HEN 1110 !095
220 GOSUB 820 !135
230 GOSUB 610 :: IF FLAG=1 T
HEN 1110 !095
240 GOSUB 730 !044
250 BATL=BATL+1 :: IF BATL=9
 THEN 1110 ELSE 210 !151
260 ,5 !177
270 REM HUMAN SELECTS !102
280 REM!154
290 REM SETUP !075
300 CALL COLOR(11,15,15,12,1
6,16)!240
310 A$="ppppzzzzpppppzzzzpppp
zzzz" :: B$="zzzzppppzzzzppp
pzzzzpppp" !119
320 FOR T=1 TO 14 STEP 6 ::
DISPLAY AT(T,1):A$:A$:A$:B$:
B$:B$ :: NEXT T !182
330 FOR T=1 TO 6 :: DISPLAY
AT(T*3-1,25):CHR$(64+T):: DI
SPLAY AT(19,T*4-2):CHR$(48+T
):: NEXT T !165
340 RETURN !136
350 REM PLACE !031
360 DISPLAY AT(20,1):"YOU MU
ST PLACE 100 MEN" :: TOT=100
 !153
370 DISPLAY AT(21,1):"ROW
 COL     MEN" !061
380 CALL HCHAR(21,7,32):: CA
LL KEY(5,A,B):: CALL HCHAR(2
1,7,30):: IF A<65 OR A>70 TH
EN 380 ELSE ROW=A-64 :: CALL
 HCHAR(21,7,A)!195
390 CALL HCHAR(21,15,32):: C
ALL KEY(5,A,B):: CALL HCHAR(
21,15,30):: IF A<49 OR A>54
THEN 390 ELSE COL=A-48 :: CA
LL HCHAR(21,15,A)!073
400 CALL KEY(5,A,B):: IF A>2
0 THEN 400 !226
410 CALL HCHAR(21,23,32):: C
ALL KEY(0,A,B):: CALL HCHAR(
21,23,30):: IF A<48 OR A>57
THEN 410 ELSE MEN=A-48 :: CA
LL HCHAR(21,23,A)!089
420 IF ARRAY(ROW+1,COL+1)>0
THEN TOT=TOT+ARRAY(ROW+1,COL
+1):: ARRAY(ROW+1,COL+1)=0 !
084
430 IF TOT-MEN<0 THEN 410 !2
33
440 DISPLAY AT(22,1):TOT-MEN
;" LEFT TO PLACE" :: TOT=TOT
-MEN :: ARRAY(ROW+1,COL+1)=M
EN !031
450 CALL HCHAR(ROW*3-1,COL*4
,A)!245
460 IF TOT=0 THEN 480 !140
470 GOTO 370 !194
480 COMP=100 :: FOR ROW=1 TO
 6 !183
490 FOR COL=1 TO 6 !210
500 A=INT(RND*RND+1):: IF CO
MP-A<0 THEN 500 !197
510 COMP=COMP-A :: CALL HCHA
R(ROW*3-1,COL*4+2,A+48):: DI
SPLAY AT(24,1):"COMPUTER HAS
 ";COMP;" LEFT" !105
520 GRID(ROW+1,COL+1)=A :: I
F COMP=0 THEN 590 !070
530 NEXT COL !116
540 NEXT ROW !142
550 IF COMP=0 THEN 590 !051
560 FOR COL=7 TO 2 STEP -1 :
: FOR ROW=2 TO 7 :: IF GRID(
ROW,COL)<9 THEN GRID(ROW,COL
)=GRID(ROW,COL)+1 :: COMP=CO
MP-1 !080
570 CALL HCHAR((ROW-1)*3-1,(
COL-1)*4+2,48+GRID(ROW,COL))
:: IF COMP=0 THEN 590 !138
580 NEXT ROW :: NEXT COL !13
2
590 CALL HCHAR(20,1,32,160)!
013
600 RETURN !136
610 REM CHECK END? !078
620 YOURTOT,COMPTOT,FLAG=0 !
163
630 FOR ROW=2 TO 7 !238
640 FOR COL=2 TO 7 !212
650 YOURTOT=YOURTOT+ARRAY(RO
W,COL)!128
660 COMPTOT=COMPTOT+GRID(ROW
,COL)!231
670 NEXT COL :: NEXT ROW !13
2
680 IF YOURTOT<16 OR COMPTOT
<16 THEN FLAG=1 !012
690 IF YOURTOT<COMPTOT/2 THE
N FLAG=1 !240
700 IF COMPTOT<YOURTOT/2 THE
N FLAG=1 !240
710 DISPLAY AT(21,1):" ":"YO
U NOW=";YOURTOT;" COMP=";COM
PTOT :: CALL DELAY !132
720 RETURN !136
730 REM HUMAN SELECTS !102
740 DISPLAY AT(22,1):"BATTLE
 CENTER: ROW  COL":"":"" !11
4
750 CALL HCHAR(22,21,30):: C
ALL KEY(5,A,B):: CALL HCHAR(
22,21,32):: IF A<65 OR A>70
THEN 750 ELSE ROW=A-64 :: CA
LL HCHAR(22,21,A)!192
760 CALL HCHAR(22,27,30):: C
ALL KEY(5,A,B):: CALL HCHAR(
22,27,32):: IF A<49 OR A>54
THEN 760 ELSE COL=A-48 :: CA
LL HCHAR(22,27,A)!200
770 ROW=ROW+1 :: COL=COL+1 !
032
780 IF BATTLE(ROW,COL)=1 THE
N DISPLAY AT(21,1):"FOUGHT T
HERE ALREADY" :: CALL DELAY
:: GOTO 740 !160
790 BATTLE(ROW,COL)=1 !106
800 GOSUB 930 !245
810 RETURN !136
820 REM COMP SELECTS !028
830 FOR A=1 TO 16 :: ROW=INT
(RND*4+3):: COL=INT(RND*4+3)
:: GOSUB 1210 !109
840 IF CT>UT AND BATTLE(ROW,
COL)=0 AND UT>3 THEN 900 !20
0
850 NEXT A !215
860 FOR ROW=7 TO 2 STEP -1 :
: FOR COL=2 TO 7 :: GOSUB 12
10 !066
870 IF CT>UT AND UT>0 AND BA
TTLE(ROW,COL)=0 THEN 900 !19
7
880 NEXT COL :: NEXT ROW !13
2
890 ROW=INT(RND*6+2):: COL=I
NT(RND*6+2):: IF BATTLE(ROW,
COL)=1 THEN 890 !092
900 DISPLAY AT(22,1):"COMPUT
ER FIGHTS AT ";CHR$(ROW+63);
COL-1:" ":" " !253
910 IF BATTLE(ROW,COL)=1 THE
N 820 !158
920 BATTLE(ROW,COL)=1 :: CAL
L DELAY !197
930 REM LETS COUNT !155
940 GOSUB 1200 !004
950 DISPLAY AT(21,1):" ":"YO
U:";UT;" COMPUTER:";CT !035
```

(See Page 18)

BULLRUN—

(Continued from Page 17)

```
960 CALL DELAY !217
970 IF CT=UT THEN DISPLAY AT
(23,1):" -NO BATTLE-" :: GOT
O 1100 !169
980 IF UT>CT THEN DISPLAY AT
(23,1):"YOU WIN" ELSE DISPLA
Y AT(23,1):"YOU LOSE" !205
990 IF UT>CT THEN GRID(ROW-1
,COL-1),GRID(ROW-1,COL),GRID
(ROW-1,COL+1),GRID(ROW,COL-1
),GRID(ROW,COL+1),GRID(ROW+1
,COL),GRID(ROW+1,COL-1)=0 !0
53
1000 IF UT>CT THEN GRID(ROW,
COL),GRID(ROW+1,COL+1)=0 !07
6
1010 IF CT>UT THEN ARRAY(ROW
-1,COL-1),ARRAY(ROW-1,COL),A
RRAY(ROW-1,COL+1),ARRAY(ROW,
COL-1),ARRAY(ROW,COL+1),ARRA
Y(ROW+1,COL+1)=0 !192
1020 IF CT>UT THEN ARRAY(ROW
,COL),ARRAY(ROW+1,COL-1),ARR
AY(ROW+1,COL)=0 !226
1030 CALL DELAY !217
1040 FOR ROW=2 TO 7 :: FOR C
OL=2 TO 7 !068
1050 CALL HCHAR((ROW-1)*3-1,
(COL-1)*4,ARRAY(ROW,COL)+48)
!114
1060 CALL HCHAR((ROW-1)*3-1,
(COL-1)*4+2,GRID(ROW,COL)+48
)!213
1070 NEXT COL :: NEXT ROW !1
32
1080 DISPLAY AT(22,1):"CALCU
LATED NEW TROOPS":"" :: CALL
 DELAY :: CALL HCHAR(22,1,32
,32)!131
1090 RETURN !136
1100 CALL DELAY :: RETURN !2
27
1110 REM FINAL PRINT OUT !23
3
1120 DISPLAY AT(21,1):"":"FI
NAL TROOPS:":"YOU:";YOURTOT;
" COMPUTER";COMPTOT !039
1130 FOR ROW=2 TO 7 :: FOR C
OL=2 TO 7 !068
1140 IF ARRAY(ROW,COL)=0 THE
N CW=CW+1 !013
1150 IF GRID(ROW,COL)=0 THEN
 UW=UW+1 !216
1160 NEXT COL :: NEXT ROW !1
32
1170 IF UW>CW THEN DISPLAY A
T(24,1):"YOU WON MOST TERRIT
ORY" ELSE DISPLAY AT(24,1):"
COMPUTER WON MOST TERRITORY"
 !222
1180 DISPLAY AT(20,1):"PRESS
 ANY KEY TO RE-RUN" :: CALL
KEY(5,A,B):: DISPLAY AT(20,1
):"PRESS ___ KEY TO RE-RUN"
:: IF B<1 THEN 1180 ELSE RUN
 !108
1190 CALL KEY(5,A,B):: IF B>
0 THEN RUN !255
1200 REM CHECK COMP CHOICE !
050
1210 UT,CT=0 !170
1220 CT=GRID(ROW-1,COL-1)+GR
ID(ROW-1,COL)+GRID(ROW-1,COL
+1)+GRID(ROW,COL-1)+GRID(ROW
,COL)+GRID(ROW,COL+1)+GRID(R
OW+1,COL+1)+GRID(ROW+1,COL)!
207
1230 CT=CT+GRID(ROW+1,COL-1)
!064
1240 UT=ARRAY(ROW-1,COL-1)+A
RRAY(ROW-1,COL)+ARRAY(ROW-1,
COL+1)+ARRAY(ROW,COL-1)!229
1250 UT=UT+ARRAY(ROW,COL)+AR
RAY(ROW,COL+1)+ARRAY(ROW+1,C
OL+1)+ARRAY(ROW+1,COL)+ARRAY
(ROW+1,COL-1)!129
1260 RETURN !136
1270 SUB DELAY :: FOR T=1 TO
 700 :: NEXT T :: SUBEND !16
0
```


No Contest?

(This article is from the March 1993 St. Louis Computer Bridge.—Ed.)

By HAROLD C. HOYT JR.

Another book on the talk circuit is making me think about basic values. The book, *No Contest, The Case Against Competition,* Houghton 1986 (316846)392,14/K78N by Alfie Kohn, questions the notion that Darwinian survival of the fittest should be the modus operandi of our culture. *No Contest* brings a strong case against "competition." Everything is competition, right? Who gets the first chair in the high school orchestra is determined in a fight to the death between contenders. The result is better music, right? In all other fields, the Darwinian survivor is better equipped to beat the competition into the ground, right?

The first area that we might question this idea is in school sports. Little League baseball has a bad reputation for making the kids who strike out feel bad. It may be the parents, but the attitude seems to be "Winning isn't the only thing, it's everything." As a person who has been both on the horse and under it in competition, I have both sets of memories. One that goes with that won-

(See Page 19)

NO CONTEST?—

(Continued from Page 18)

derful satisfying feeling that comes from beating the competition into the ground, and another feeling that goes with being completely beaten into submission. So, I can compare.

Neither feeling is very healthy, and I'm not certain how I feel about competition. On the one hand, we have to be world competitors to keep up our standard of living. On the other hand, one gets the distinct feeling of having been had, emasculated and used by the system. We have been seduced into believing that we should emulate the Japanese because they beat us into the ground for a while, so something that they are doing must be better than something that we are doing, right?

I don't think so. Our system is better. What they did do better, they learned from us by importing an efficiency expert from the Bell Labs so they could learn from us. It would be disaster for us to emulate their ant-hill society because we are a different kind of people. Our workers are supposed to take the blame for failures in manufacturing when the real problem has been a lack of leadership. I don't think their system is even very good for them. Too much competition produces winners who are arrogant and losers who don't feel good about themselves. Feeling good about yourself came into vogue with the hippies. As a puzzled person living in the Clinton era, I wonder about the notion that an automobile is not evaluated on its engineering merit, but whether or not it was made by nice people.

But I digress. Companies that have been ripping off and abusing their engineering talent have been yearning for a way to measure creativity. If they could measure it, they could control it. The most creative people I've run into don't fit into a mold. Overpaid CEOs who have competed in a contest to see who can be the most obnoxious, and won, seem to enjoy beating on their engineering help. Abusing the help enhances their feeling of superiority.

Does this have anything to do with people falling all over each other to accept offers of early retirement? The most creative people I know have to be torn from their work, kicking and screaming, by their spouses, etc., and forced to take a vacation. The traditional (reactionary, conservative?) view of the virtue of capitalism is the creation of real wealth in manufacturing something. The idea of using the lottery as a way of getting rich without producing any useful product is a kind of sick idea. Getting something for nothing by "taxing" compulsive gamblers has something to do with competition, macho and major corporations losing their ability to compete. If I had an answer to what is wrong, I'd be doing more than writing this column. Maybe it has to do with this generation of leaders. I asked a loyal 25-year employee of McDonnell-Douglas who had been laid off: "If J.S. McDonnell were still alive, would his corporation be in this fix, even with the end of the cold war?" "Not in a million years," he replied. Success is an imperative. Failure isn't in the vocabulary of these founders of the giant corporations that made America great. When TI stopped its home computer, the competition didn't win. We all lost. When IBM picked up the home computer market a little later, it didn't win either. IBM still hasn't figured out what they are doing wrong, but let me tell you, it will take more than a new operating system like OS2 to fix it. The people that have a vision of the future that could change everything are orthogonal to competition, and still aren't being encouraged, challenged.

Andy Rooney, of *60 Minutes*, decried sneaky ways to get kids interested in math. Mr. Rooney feels the need to take his math cold turkey, memorize the multiplication tables up to 12x12 before a child would be allowed to touch a pocket calculator. I don't think so. Give the kid the calculator and sneak in the times tables. A kid who doesn't realize that he is learning something and just thinks that he is having fun will beat his grim-faced Asian competition into the ground and do so without exhibiting bad sportsmanship. What to do with the CEO who blamed our problem on Japan is another question.

Statement of Ownership, Management and Circulation
(Required by 39 U.S.C. 3685)

1A. Title of Publication	1B. PUBLICATION NO.	2. Date of Filing
MICROpendium	1 0 4 3 2 2 9 6	9-30-93
3. Frequency of Issue	3A. No. of Issues Published Annually	3B. Annual Subscription Price
Monthly	12	$35/yr

4. Complete Mailing Address of Known Office of Publication (Street, City, County, State and ZIP+4 Code) (Not printers)
502 Windsor Rd. Round Rock Tx 78664-7639

5. Complete Mailing Address of the Headquarters of General Business Offices of the Publisher (Not printer)
P.O. Box 1343 Round Rock Tx 78680-1343

6. Full Names and Complete Mailing Address of Publisher, Editor, and Managing Editor (This item MUST NOT be blank)
Publisher (Name and Complete Mailing Address) John Robert
P.O. Box 1343 Round Rock Tx 78680-1343
Editor (Name and Complete Mailing Address) Laura [redacted] Burns
P.O Box 1343 Round Rock Tx 78680-1343
Managing Editor (Name and Complete Mailing Address)

7. Owner (If owned by a corporation, its name and address must be stated and also immediately thereunder the names and addresses of stockholders owning or holding 1 percent or more of total amount of stock. If not owned by a corporation, the names and addresses of the individual owners must be given. If owned by a partnership or other unincorporated firm, its name and address, as well as that of each individual must be given. If the publication is published by a nonprofit organization, its name and address must be stated.) (Item must be completed.)

Full Name	Complete Mailing Address
Burns-Koleen Communications Inc.	P.O. Box 1343 Round Rock Tx 78680-1343
c/o Jim Kyle	P.O. Box 1343 Round Rock Tx 78680-1343
Laura Burns	P.O. Box 1343 Round Rock Tx 78680-1343

8. Known Bondholders, Mortgagees, and Other Security Holders Owning or Holding 1 Percent or More of Total Amount of Bonds, Mortgages or Other Securities (If there are none, so state)

Full Name	Complete Mailing Address

9. For Completion by Nonprofit Organizations Authorized To Mail at Special Rates (DMM Section 424.12 only)
The purpose, function, and nonprofit status of this organization and the exempt status for Federal income tax purposes (Check one)
(1) ☐ Has Not Changed During Preceding 12 Months (2) ☐ Has Changed During Preceding 12 Months (If changed, publisher must submit explanation of change with this statement.)

10. Extent and Nature of Circulation (See instructions on reverse side)	Average No. Copies Each Issue During Preceding 12 Months	Actual No. Copies of Single Issue Published Nearest to Filing Date
A. Total No. Copies (Net Press Run)	2425	2100
B. Paid and/or Requested Circulation 1. Sales through dealers and carriers, street vendors and counter sales	340	301
2. Mail Subscription (Paid and/or requested)	1271	1166
C. Total Paid and/or Requested Circulation (Sum of 10B1 and 10B2)	1611	1467
D. Free Distribution by Mail, Carrier or Other Means Samples, Complimentary, and Other Free Copies	78	55
E. Total Distribution (Sum of C and D)	1689	1522
F. Copies Not Distributed 1. Office use, left over, unaccounted, spoiled after printing	736	578
2. Return from News Agents	0	0
G. TOTAL (Sum of E, F1 and 2—should equal net press run shown in A)	2425	2100

11. I certify that the statements made by me above are correct and complete
Signature and Title of Editor, Publisher, Business Manager, or Owner

PS Form 3526, January 1991 (See instructions on reverse)

Using a spreadsheet to protect retirement income

By DAVE HOWELL
Erie 99er User Group

Some of you already know that I have been planning to retire from the school district. Well, the time has arrived. I will be entering the rapidly growing world of senior citizens — or whatever you call those wonderfully resourceful people on fixed incomes. Anyway, I've been planning for this day since the early '80s, never realizing that I would actually be looking forward to it! That's when I seriously embarked on building a nest egg — courtesy of the IRA. Actually, I was just looking for a way to reduce my income taxes at the time. Ever since then, I took an interest in where to place the money, wondering if I would ever have enough to live on in retirement and when that day would come.

Five years ago, I began receiving pension estimates from our retirement system with which to do some number crunching. I also needed to estimate what our investments would be worth and how much we would need to live on each year throughout retirement. In other words, what figure should we use as the most prudent "cost of living" factor and "percent of yield" for our investments. How much insurance should I carry, if any, to protect my beneficiary and which pension option should we select? This is a mighty tall order and one that most people of modest means must come to grips with sooner or later.

I've received all kinds of advice from insurance and stockbrokers, from financial planners and tax accountants, from friends, business associates, retirees and family members. When word gets out, there's no stopping the flood of "well-wishers" hoping to benefit in some way.

Many things have been learned:

(1) Everyone's situation is different and doesn't necessarily fit a standard financial mold.

(2) Realize that most of those professionals who seem eager to give you free advice have something to gain in doing so.

(3) Seek as many alternatives and quotes as you need to feel confident in arriving at the most appropriate arrangement you can live with.

(4) Determine the parameters of just what you expect from your retirement years: what do you expect to accomplish, what kinds of activities do you wish to get into, and are those expectations realistic in terms of your financial reserves?

There are no hard, fast, air-tight guidelines to determine the solution for everyone. One must laboriously go through the process to discover the best course of action. To help facilitate the process, I turned to the computer. Gosh, what did people do before the personal computer?

I've received all kinds of advice from insurance and stockbrokers, from financial planners and tax accountants, from friends, business associates, retirees and family members. When word gets out, there's no stopping the flood of "well-wishers" hoping to benefit in some way.

But try to find suitable software to project all of the variables of retirement financing over a period of say — 20 or 30 years! Some "financial planners" and insurance outfits have"in house" computer programs that are able to project your resources against need but, because they were in someone else's hands, I didn't feel comfortable with the results. I needed a process I was familiar with. I found a few programs which accept some of the basic data that determines if there is enough to retire on, but they didn't paint the picture for each year down the road. Nor could the variables be changed to observe the "what if" on your standard of living at any point in time. I even tried to find an MS-DOS program with which to convert for use on the TI. I found nothing suitable.

So I decided to use the spreadsheet format as a number-crunching tool. And a mighty fine decision at that. It's the spreadsheet that can take all the required variables, calculate and display the results over a period of time.

Since the bulk of my experience with computers involved the TI99/4A, I decided to use the TI Multiplan. As a novice to Multiplan, I had to learn to use it first. It was well worth the effort! The resulting spreadsheets were indeed very crude but promising. Over the next few months I succeeded in refining the design to the point of impressing several investment counselor. In preparing for this article, I decided to streamline the spreadsheet to permit changing the outcomes instantly simply by changing one or more variables.

The spreadsheet displayed herein shows the results of a hypothetical situation over a period of 20 years. The display is divided into two sections: The first five columns gives projections for a retired couple. The second four columns attempt to display the spouse's finances beginning with the year of the retiree's death. There are six variables to which I've assigned appropriate values:

40,000—Beginning joint income.

30,000—Beginning income for surviving spouse.

1.050 — (5 percent) Cost of living factor.

150,000 — Initial investment.

0.080 — (8 percent) Investment yield factor

100,000 — Life insurance proceeds on retiree or spouse.

Now that you've had a look at the sample spreadsheet and you are still interested, let's examine the variables in greater detail. Then we will take a look at the column structure and their implications.

Joint Income Needed (INC). The initial amount needed for both the retiree and spouse to live on for the first year of retirement. Much care must be taken in determining a realistic figure. The popular rule of thumb that you could live on approximately 80 percent of your current income is, for all intents and purposes, about as re-

(See Page 21)

USING A SPREADSHEET—

(Continued from Page 20)

liable as your "arthritic thumb." You should gather all of your receipts and records for the past 12 months (a very convenient proposition this time of the year!) and deal with each line item separately. Try to determine if the specific expense will exist or change in retirement. Take into consideration what kinds of activities you hope to be pursuing in retirement — beyond loafing and goofing off! Don't forget medical and/or long-term nursing care insurance. Also remember that a good portion of your monthly pension payments and possibly Social Security payments and any earned income you may have will be subject to income taxes.

Spouse Income Needed (SINC). As with Joint Income Needed, this is the amount estimated for the spouse in the specific year he/she is widowed. Again, each expense category must be examined carefully in terms of anticipated lifestyle changes while living alone.

Cost of Living Factor (COL). Obviously, the expense of living will increase somewhat each year if you wish to maintain your standard of living. The figures shown in both joint and spouse income columns are multiplied by the COL each year. As can be seen, the effects of COL down the road are rather brutal to say the least! The COL is a formidable threat to any and all retirement plans for those on fixed incomes.

Initial Investment (INV). This is the amount of your current next-egg or the nest-egg you expect to have upon retirement. If you are of modest means, as we are, you will undoubtedly need the income from this investment to supplement your fixed income from pensions, Social Security, etc. I found that, initially, our income from pension and Social Security sources accounts for only 60 percent of our total income needed when I retire! Less for each of the years thereafter. That's why this nest-egg is mandatory.

Hopefully, the yield from this investment will be enough to see you through the rest of your life in spite of COL and your lofty retirement dreams. Some say you must try to preserve, or even increase, your nest-egg at all costs *after* your retirement. Others say that there is nothing wrong with gradually depleting your principal investment as the years pass, unless, of course, you want to leave it for your heirs. You should, however, keep in mind

(See Page 22)

FWMXJTSP - RETIREMENT INCOME PROJECTION (JOINT/SPOUSE)

JOINT INCOME NEEDED	=	40000.00	(INC)
SPOUSE INCOME NEEDED	=	30000.00	(SINC)
COST OF LIVING FACTOR	=	1.05	(COL)
INITIAL INVESTMENT	=	150000.00	(INV)
INVESTMENT YIELD FACTOR	=	0.08	(IYF)
LIFE INSURANCE PROCEEDS*	=	100000.00	(LINS)*

PROJECTIONS FOR WIDOWED SPOUSE at time o retiree's death.

YR	JOINT INCOME NEEDED	PLANNED INCOME Pension+	INVESTMT YIELD	EXCESS/ SHORTFAL	INVESTMT BALANCE 150000	SPOUSE INCOME NEEDED	PLANNED INCOME AVAILABLE	INVESTMT YIELD C6*IYF	INVESTMT BALANCE
1	40000	47335	12000	19335	169335	30000	12107	12000	244107
2	42000	37150	13547	8697	178032	31500	13066	13547	264448
3	44100	42100	14243	12983	165789	33075	13024	14243	272223
4	46305	40050	13263	7008	172797	34729	13385	13263	257708
5	48620	40432	13824	5636	178433	36465	13756	13824	263912
6	51051	40825	14275	4048	182481	38288	14139	14275	268558
7	53604	41224	14598	2219	184700	40203	13535	14598	270412
8	56284	41646	14776	138	184838	42213	13939	14776	271202
9	59098	42076	14787	-2235	182602	44324	14357	14787	269658
10	62053	42518	14608	-4927	177675	46540	14788	14608	265459
11	65156	42971	14214	-7971	169705	48867	15232	14214	258255
12	68414	43443	13576	-11394	158311	51310	15689	13576	247660
13	71834	43927	12665	-15242	143068	53876	16159	12665	233259
14	75426	44425	11445	-19556	123513	56569	166445	11445	214588
15	79197	44937	9881	-24379	99133	59398	17143	9881	191139
16	83157	45466	7931	-29760	69373	62368	17658	7931	162354
17	87315	46010	5550	-35755	33618	65486	18187	5550	127623
18	91681	46570	2689	-42421	-8804	68761	18733	2689	86280
19	96265	47147	-704	-49822	-58626	72199	19295	-704	37589
20	101078	47742	-4690	-58026	-116652	75809	19874	-4690	-19250

*Includes $100,000 life insurance on retiree for spouse

USING A SPREADSHEET—

(Continued from Page 21)

that the older you get, the less time you will be around to need it — according to life expectancy projections.

When estimating what your initial investment is going to be, you should be aware of the tax bites on any funds or rollovers when implementing your retirement plans. There are new rules in effect January 1st of this year which may govern the taxes you pay on funds "rolled over" from 401(K) and similar plans.

Investment Yield Factor (IYF). Just as elusive as estimating the COL or your life expectancy is the guess of how well you think your investments will do during your retirement. In addition to a volatile stock market, you must consider the type of investor you are. Are you conservative— unwilling to take sizeable risks? Or can you sleep well at night knowing that the bulk of your funds are in higher risk investments?

Obviously, the conservative investor, at this point in time, is playing footsie with the COL. That's fine if the size of the nest-egg is large enough to see you through the rest of your life — or at least until the yield returns to a level significantly higher than the COL. I have a recently widowed 82-year-old neighbor whose investments are in CDs and money markets. I was very concerned about her future until I realized that at her age she could easily live well on what she has left without earning much of a return on her money.

At the other end of the scale are those who invest aggressively in an effort to stay well ahead of the COL *and* current living expenses before cutting back to safer territory. If the bottom falls out of the economy, well....

Then there are those who seek a balanced portfolio so that no matter which way the market goes, the impact is minimized — in either direction.

More and more investors, these days, are finding it desirable to invest in the rapidly expanding world of mutual funds. A mutual fund, depending on your goals, spreads your money among many markets, thus cushioning the effects of a rapidly swinging market and avoiding the "sudden death" syndrome of investing in individual stocks or bonds.

Life Insurance Proceeds (LINS). Don't forget to include any insurance you might have on your life in the "Investment Balance" for your spouse upon your demise. I've taken this factor into account in the last column of the spreadsheet to arrive at the estimate your spouse will have on hand at any specific year of death. I used this column to help determine how much, if any, life insurance proceeds your spouse will need at any point down the road.

CELL CONTENTS OF SPREADSHEET

Name the following cells:

Row	Column	Name
3	5	INC
4	5	SINC
5	5	COL
6	5	INV
7	5	IYF
8	5	LINS

Use the following formulae:

Col	Formula	Row
2	INC	R14
2	R[-1]C*COL	R15-34
3	Use actual numbers or create your own	
4	IYF*INV	R14
4	R[-1]C[+2]*IYF	R15-34
5	C3+C4-C2	R15-34
6	C5+C6+INV	R14
6	R[-1]C+RC[-1]	R15-34
7	SINC	R14
7	R[-1]C*COL	R15-34
8	Use actual numbers or create your own	
9	IYF*INV	R14
9	R[-1]C[-3]*IYF	R15-34
10	INV-C7+C8+C8+LINS	R14
10	R[-1]C[-4]-C7+C8+C9+LINS	R15-R34

Since I was "insurance poor," I used this information to shop around for insurance. This is a bewildering task made more difficult by aggressive salespersons promoting a plethora of insurance products. They ranged from term life insurance to the more elaborate cash value annuities and insurance trusts. You have to decide whether you want to use insurance as a savings vehicle that could provide a significant monthly payout at some point later in your retirement, or simply for a basic face value payout to your beneficiary. It has been my experience that insurance brokers will try to move you into the more expensive cash value policies. These policies including tax-deferred annuities are more expensive (much of which goes to sales commissions, at least for the first year).

Since I couldn't "hack" spending from $5,000 to $9,000 of my retirement income *every year* during retirement for a cash value $200,000 policy, I opted for a universal life policy at less than half of the above premiums. In the process of looking for suitable policies, I found $100,000 term life insurance policies whose annual $800 premiums were very inviting indeed! But the term is only 10 to 15 years. After that, you will pay close to triple that amount to renew the policy *if* you pass the required physical! I decided not to base my spouse's financial well-being on my health.

The key to finding the right insurance program *at an affordable price* is to shop around. I first went to "big name" companies for estimates, then checked out identical policies with lesser known but equally rated companies whose rates were much more reasonable. When I say "identical policies," I'm referring to things like the same rate of return (cash value) over the identical number of years. One almost has to be an actuarial expert to avoid the pitfalls in buying insurance even if you trust your insurance agent.

THE SPREADSHEET FORMAT

As I mentioned earlier, the spreadsheet

(See Page 23)

USING A SPREADSHEET—

(Continued from Page 22)

is divided into two exhibits. The first five columns show the utilization of current (fixed) income and investment income in financing the needed income for two people. If you are single, there is probably no need to consider the information in the last four columns including the life insurance.

The challenge is set by the "Needed Income" increased each year by the COL. Adjusting the initial amount up or down for the first year will, of course, proportionately affect the amounts for subsequent years.

The second column should show the totals of the regular income you expect from pensions, Social Security and, perhaps, earned income from a part-time job, rental unit or some avocational endeavor. This should include your spouse as well. Don't include income distributions or payouts from investments. Those funds should be included in the "Investment Balance" column.

Important! You may change these amounts for any year at any time in order to adjust your financial plan as experience dictates. That is the beauty of spreadsheets. If your income rises or your expenses decrease during any specific year, you can make the required adjustment to see how it affects the years to come.

The "Investment Yield" is simply the Investment Balance times the Investment Yield Factor (IYF). The Investment Balance for the first year is the original amount prior to retirement (in this case $150,000). For each of the succeeding years, the Investment Yield is calculated on the previous year's Investment Balance.

The Excess/Shortfall column contains the amount that both the Planned Income and Investment Yield combined exceeds or falls short of the Income Needed (first column). A plus figure (numbers with no sign in front — I don't know how to put a + sign in Multiplan) indicates the excess that is added to the Investment Balance. A negative sum indicates that the combined Planned Income and Investment Yield did not meet the Joint Income Needed and the resulting shortage was subtracted from the Investment Balance column. In other words, your nest-egg is being raided! That's all right if you still have enough to last the rest of your life! Why not extend the spreadsheet another 10 or 20 years to make sure? I dare ya!

The Investment Balance column (6) indicates that I would run out of funds by the 18th year of retirement assuming, of course, that all of the variables remained the same throughout. Had this been my projection, I probably would hold off retirement until the Initial Investment increases substantially up to perhaps $200,000. Or perhaps I should find ways to cut our living expenses somewhat below $40,000 ($30,000 for the spouse). In any event, using a spreadsheet like this alerts us to what's involved and how those factors affect our future.

(See Page 28)

READER TO READER

Hans Huben, Berberitzenweg 6, D 71083, Germany, writes:

I'd like to contact readers who work with c99. I wrote machine routines to work in text mode (80 characters). These you can call from c99. It's possible to rewrite your old programs from 40 to 80 characters. These additional routines contain 16 new commands. All C-routines, i.e., printf, scanf, puts, putchar, work again; 255 characters are available. I wrote a window routine in c99 using the characters above 127. You can position your text with locate (x,y) from line 1 to 25 and column 1 to 80, using and deleting colors, etc.

Now I'm writing my first program with this text mode routine. It's called PRINTMANAGER. With some experience you can adapt it to your own printer by altering parameters and text, considering the examples I wrote. PRINTMANAGER is able to select printer codes from a menu, send it to the printer, load and send text to the printer. You look for the result. You can store print codes, too, and load the file to your editor. I'll demonstrate it at Stuttgart at our TI-Faire.

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.*

NEWSBYTES

Racine, Milwaukee groups plan merger

The TI99/4A Users Group of Racine, Wisconsin, is discussing a merger with the Milwaukee Users Group. The group will continue to meet the second Thursday of the month in Racine as a SIG (special interest group) of the Milwaukee group, according to Don Walden, president of the Milwaukee group..

President of the Racine group is Leon Guntly, (414) 639-7633.Walden may be contacted at (414) 679-4343.

Historical Computer Society forms

The Historical Computer Society, described as a non-profit organization dedicated to preserving older and classic computer hardware, software and literature, has been founded and has published its first newsletter, Historically Brewed.

Persons wishing to subscribe and join may send a check or money order for $15 to HCS, 10928 Ted Williams Place, El Paso, TX 79934. For a sample newsletter, send $2.50 to the same address.

David A. Greelish, founder and president of HCS as well as editor of the newsletter, may be contact on CompuServe at ID #100116, 217, or on Delphi under the ID HISTORICAL.

Send your product announcements and other information of interest to TI/Geneve users to MICROpendium Newsbytes, P.O. Box 1343, Round Rock, TX 78680.

ASGARD MEMORY SYSTEMS

A new day for the TI99/4A

By BRUCE HARRISON

Let's start with your author's admission of guilt. Readers of my regular column will readily accuse me of being strongly biased against the addition of various hardware gadgets to the TI99/4A. Guilty as charged!

The folks at Asgard may be worried about such a biased author reviewing this product, but they need not worry. This product is good! Lately, we've become all-too accustomed to having new and great-sounding products announced, touted widely, and then dropped without ever getting to the stage of "production," let alone being placed in our P-Boxes. Not so the Asgard Memory System, which has had a remarkably short period elapse between our first knowledge and the reality of a product. Thanks for this must go to the eagerness of Jim Krych, the backing of Chris Bobbitt and Harry Brashear, and the talents of programming geniuses Joe Delekto and Art Green. Thanks to these folks, there's a real chance that TI users will find a light at the end of the 10-year tunnel.

WHAT IS THIS THING?

To start with, it's not just a memory, but a system that can make the TI into a world-class all-purpose computing machine. Physically, it's a half-height card that goes into the P-Box in place of the 32K card. The only compatibilty problem that we know about is that it can't coexist with either the Cor-Comp or Myarc "RAMdisk" systems, since both of those must supplant the 32K memory, and two things can't occupy the same memory address space at the same time. More modern RAMdisk systems, like the Horizon 3000 and 4000 series can coexist peacefully with the new system, as we proved to our own satisfaction by placing the card in our own P-Box. (Our Horizon 3000 does not have RAMBO, so we can't be certain of compatibility with that feature, but we don't think that would cause any problem.) The System part is a set of software, which includes a new menu driver that allows the use of the larger memory for assembly programs. Programs designed and developed for the system will be able to perform truly wonderful feats. For those of us who like to make our own programs, there are the Macro Assembler and Linker that come with the system. Thus we can "homebrew" programs well over the 32K limit that we've had to live with before. The base size for the memory is 128K, which allows the "system software" and programs of 70K or so to be resident at the same time.

REVIEW

REPORT CARD

Performance.....A
Ease of Use.....A
Documentation.....A
Value*.....?
Final Grade.....A
Cost: $119.95**
Publisher: Asgard Software, 1423 Flagship Dr., Woodbridge, VA 22192 (703) 491-1267
Requirements: TI-99/4A Console, P-Box Monitor, DSDD disk drive.
* Value is hard to assess — see text.

QUICK TESTS

Our first little test, after placing this card in the P-Box, supplanting our TI 32K card, was to simply load up and run our little CALIB test. This test checks the speed at which the machine is running in a manner which reflects the memory-access time as well as the CPU speed. For a "normal" TI with the TI 32K card in place, this program puts the number 199 on the screen twice, once just to show the "norm," and the second reflecting actual measured time for the current configuration. (For example, on a Geneve this second number may be 48, indicating a much faster operating speed.) Our concern here, mainly, was that the speed with this new card be not too different from "normal," so that timing-sensitive operations would behave as expected. The AMS card gave us the hoped-for result, with 199 being displayed twice on the screen. This means that programs you ran on your TI before will behave exactly as you'd expect.

IN THE SYSTEM

Next, we used the program ABOOT (included in the package) to get ourselves into the "System." We were curious about whether the normal type of Option-5 program file would work the same in the AMS "environment." We tried selection A (LOAD AND RUN) from the AMS menu, and tried DSK1.FW (Funnelweb). So far as we could tell, Funnelweb worked perfectly in all respects. It also worked with the card in place operating outside the "system." Then we remembered that this was an old version of F'web, not the latest edition. (The old version happened to be closer at hand.) We pulled out our "working copy" of Version 4.4, and tried that with AMS's system. *No go!* Funnelweb itself would load up, but none of the normal Funnelweb programs (Text Edit, for example) would load. Why? One of the differences between the old version and the new one is the use of our "boot-track" process in the newer one. This might be the problem, so we tried our own WP under the AMS system, and this went bonkers! We know exactly why this is so. Art Green's DSRLNK routine, unlike just about any other DSRLNK, does not leave the data about the CRU address and device name pointer in CPU RAM Pad. (Most DSRLNK routines leave the CRU address in >83D0 and a pointer which can be used to find the device name at >83D2.)

Thus the use of Art's DSRLNK in the "system" has deprived us of information we thought could be depended upon, and caused some otherwise good software not to work. In the docs for Art Green's Linker, he mentions that his DSRLNK does not leave anything behind in RAM PAD as this were an advantage. It's not, if the software we're loading depends on finding out what disk it was loaded from. Tony Mc-

(See Page 24)

ASGARD MEMORY SYSTEMS—

(Continued from Page 23)

Govern's Funnelweb is robust enough to gracefully report errors when trying to load its program files, and didn't crash the system like our own WP did, but still we wish Art Green would make his DSRLNK leave a trail like the others do. Some of us use that information, Art! What possible advantage is there in not providing that data?

We called Joe Delekto, and he confirmed that the system software uses Art Green's DSRLNK, as we suspected. We strongly recommend that this be fixed, not just for us, but for others who may be designing new software for use with AMS. Of course either Funnelweb or our own WP can be run using the card without the AMS system software, and everything will work correctly.

During our rather lengthy conversation with Joe, he pointed out a few facts we hadn't yet discovered. Very skilled Assembly programmers can use routines included on one of the disks that are provided to "home brew" software that takes advantage of the card's memory without using the AMS system software. One author reportedly has come up with an Assembly routine for use with Extended BASIC and the AMS card. Using this routine, one can load more than one XB program into the card's pages, then run the "resident" XB programs selectively. Making this kind of use of the memory will require some study, but basically the job consists of writing through a CRU process into the registers that control paging of the memory, so that different parts of the large memory can be "mapped" into the normal high memory addresses.

HERITAGE

One rather interesting "doc" supplied with the system gives the background of how this new product relates to the TI-99/8. The technology base and architecture of this new memory card are rooted in the TI99/8, which of course was never put into production. In effect, what Asgard has done fulfils the "promise" that was to have been the 99/8. It will take some time for this new product to get established, so early purchasers may need patience to wait for the programmers to take full advantage of the new capability. There's no guarantee that this wait will be worth it, but at least there's a real product being offered, not "vaporware," and that product is firmly rooted in TI's original concepts, not in some half-baked idea.

THE DOCS

As we said in our review of the Assembler and Linker that are included with this package (September 1993 MICROpendium), the docs are voluminous, but clearly written and worth reading. Those who are not familiar with Assembly programming will want to skip over a lot of the material, but for people who plan to write software in really big amounts, a careful reading of the docs is a must. The Linker docs are particularly important for the programmer, as the rules for making full use of the memory are somewhat complex. Once the programmer understands these rules, the Linker will make construction of large programs easier, but lots of studying will be required to get "up to speed" with the system.

EASE OF USE

This topic can be looked at in several ways, so let's begin with the situation for the ordinary non-programmer user. For normal everyday use of the programs in your library, nothing changes. The card will behave as if it's just an ordinary 32K memory expansion, and programs will run just the same as always. As programs designed for the expanded memory become available, things will get a bit more complicated. To run programs that are designed within the system environment, you'll need to load up ABOOT from E/A Option 5 first, then run the programs from that. Some programmers will choose to work outside the system and still make use of the extra memory, and these programs will most likely be runnable from E/A without the ABOOT system software. In either case, the use of programs will not be significantly more difficult.

For those of us doing Assembly programming, things will be different, as there's a new set of rules overlaid on the usual problems of writing and debugging programs. For that matter, just debugging presents a whole new range of things that can go wrong. Maybe the next product developed for this system should be an advanced debugger. So far as we know, there's no existing tool that would allow probing the depths of an 80K program on the TI.

The learning curve for programmers will be steeper for those of us who have done Assembly programming on the PC, where we can make our "modulesÆ in any length from two bytes to 64K bytes, and can assign segment boundaries in any manner we want. Thinking in 4K increments will take some getting used to.

THE GRIPES

Right from the start, there was the fact that the distributed disk containing all the system software and docs was DSDD. Many of us in the community still use the TI disk controller, and can't use DSDD disks. We recommend that the distribution disks be provided as DSSD. This point is made a bit more galling by the fact that the docs state that the distribution is DSSD. Maybe that one extra disk would push the cost of shipping up by a quarter or so, but that's small compared to the irritation that DSDD can cause. A phone call to Asgard got us the two DSSD disks so we could proceed, but that shouldn't be required.

When ABOOT loads up, it reads a "script" D/V-80 file to make its menu on the screen. Several items listed on this menu were not included in the package. We were able to edit the script file so that only included items showed up on the menu, but we think that editing should have been done before the disks were distributed.

The system software provides a "stay resident" quality, so that programs can be loaded and kept just a keystroke away. That worked as long as we were still in the system, but using the re-entry program ASHOE rendered the resident programs unavailable. Joe Delekto told us that this should work, and thought that perhaps we had the wrong version of ASHOE. All distributed copies should have the correct versions of the programs.

The card, as we said at the beginning, is half-height. This makes it more difficult to install in the P-Box, especially for those with aging fingers, like yours truly. Once it's in there, it worked correctly, but the particular model we received has no activity light. Sometimes that light is the only

(See Page 25)

ASGARD MEMORY SYSTEMS—

(Continued from Page 24)

clue we have that something is happening, particularly on some XB programs. We are told that the light has been added to all new production units, so this gripe should go away.

SUMMARY

This is a very good product, with the potential to open a whole new "world" to the TI99/4A user. The card we tested was the AMS with 128K memory. This model can also be ordered with 512K memory, or can be upgraded from 128 to 512. The AMS-2, as we understand it, will be able to go to 1 megabyte capacity, and the AEMS into the hinterlands up to 16 megabytes, all within the original "framework" as intended for the TI99/8. That's an incredible capability. As our report card indicates, the product gets straight A's, but we've made the Value grade a question mark. The true value of this product will be realized only when we have programs to take advantage of the memory capacity now offered. For the sake of all our readers, we hope that one day you'll be running mega-programs on your TI99/4As.

This package is not super-cheap, but it's far less expensive than a PC or Mac, and if the software becomes available, this may be the most valuable product ever to hit the TI market.

MICRO-REVIEWS

ALEX/FORMA, Maze Mania, 3.5" Disk Label Program

By CHARLES GOOD

I have been told by well known personalities in the TI community that the primary programming language these days is assembly, and that little new *good* public domain material is being released for us any more. Based on what people are sending me for review, these statements may not be true. I am receiving some entertaining games and useful applications written in Extended BASIC such as the XB game and utility reviewed this month. Some useful software is currently being created for our public domain including the ALEX/FORMA reviewed here and Bruce Harrison's Extended BASIC compiler project. Bruce sends me on-disk demos of his progress, and there is significant progress.

You may notice that I don't give letter grades or 1-4 star ratings in my reviews. I don't find such quick distinctions meaningful. Almost everything reviewed in MICROpendium gets an A- to A+ rating. I have never seen a D and can't remember the last time I saw a C over-all rating. I feel that a good description of the product, a comparison to similar products, and comments on any specific problems I encounter should be sufficient and meaningful. If I don't consider a product sent to me worthwhile I won't bother doing a review. Send your 99/4A materials for my review and your requests for the $1 software I offer to distribute to me at P.O. Box 647, Venedocia OH 45894. Your $1 bill pays for the disk, return postage, and my Florida vacation fund. My evening phone is (419) 667-3131.

One additional note: I have scheduled the next totally free Lima all TI conference for Friday/Saturday, May 13-14, 1994. Mark your calendars.

ALEX/FORMA
by Jan Alexandersson

In order to utilize its advanced graphic and foreign language features fully, users of the newly released Funnelweb v5 editors now have to consider a bunch of technical questions that did not previously concern most 99/4A users. What is the difference between 7 bit and 8 bit ASCII characters, screen displays, and printer printouts? Why won't Eurowriter (TI's European version of TI-Writer) D/V80 files load into the USA versions of TI-Writer? What is the difference between the tab lines of the USA and European TI-Writers? How do you print the accented vowels created with Funnelweb's Eurowriter option? Which printers will properly print the neat graphics of Funnelweb's All Chars mode (some older dot matrix printers won't)? What are the differences between the USA TI-Writer Formatter and the European TI-Writer Formatter? What does one do with all the transliteration files that are a part of the European TI-Writer Formatter package? How do you use your printer's "national character sets"?

These and many other similar questions have been researched by Jan Alexandersson. The results of this research have been placed on a DSSD public domain disk named ALEX/FORMA. Included is Jan's own review of the v5 Funnelweb editors and his advice on their configuration. Also included are text files that discuss how different printers handle All Chars graphics and descriptions of the USA and European formatter. One text file can be printed to provide a hard copy of how your particular printer prints each of the high ASCII graphics accessible with the Funnelweb v5 editors. Everyone should have this hard copy readily available. The results may differ from the Funnelweb's All Chars help screens appearing on your monitor.

Jan includes a number of Formatter transliteration files. These transliterate Eurowriter to a 7-bit printer or to an 8-bit printer, transliterate All Chars files created with Funnelweb to 7-bit printers with European and USA formatters, etc. etc.

(See Page 26)

USER NOTES

(Continued from Page 27)

should make photocopies of the grid because, aside from errors, you may want to redefine the entire character set. I can redefine up to 128 characters on my printer.

Line 240 is optional for emphasized print.

Line 250 begins the downloading with an ESCape sequence — ESC and zero on my printer — and continues in the same sequence with two characters (line 260) that tell the printer the code for the first CHR$ and the last CHR$ (65 to 67, or A to C).

Line 270 is the attribute character used to tell the printer which set of eight print wires of the nine available to use for this one character. It also determine where the first and last dot columns are.

Lines 280 through 300 describe each dot column by binary weights in a straight binary format.

Line 310 is the attribute for the second character (B or 66), and lines 320 through 340 define its binary equivalents.

Line 350 is the attribute character for the third redefined character (C or 67) and, after the characters that you wish to be defined are taken care of, line 380 enables the use of RAM instead of ROM for character definition during the printing process.

Now run this program. Assuming it is compatible with your printer, using the information in lines 270 to 370, draw the characters that are printed on your grid. The attribute character takes a little study because of the values assigned to the binary positions. If entered incorrectly, it will make the printer output things that are not even close to what you will see when you run this program.

Lines 390 through 450 are there just to print out a listing of the uppercase alpha characters with the first three being shown as the redefined characters.

Just think, if you wanted to encode messages, you could encrypt code to redefine and download new characters to the printer and have the printer create a hard copy that would be difficult to decipher.

```
100 REM DOWNLOAD CHARACTERS
!018
110 REM BY JOHN WILFORTH !05
1
120 REM HOW TO REDEFINE THE
!206
130 REM CHARACTER A PRINTER
!236
140 REM THINKS IT PRINTS. !1
18
150 REM ENABLE ANY HARD- !21
3
160 REM WARE SWITCHES YOU !1
44
170 REM PRINTER REQUIRES TO
!049
180 REM PERMIT THE DOWNLINE
!012
190 REM LOADING FEATURE IN !
155
200 REM YOUR PRINTER FIRST.
!035
210 REM ******************
!216
220 OPEN #1:"PIO" !253
230 PRINT #1:CHR$(27)&":"&CH
R$(0)&CHR$(0)&CHR$(0)!084
240 PRINT #1:CHR$(27)&"E" !1
31
250 PRINT #1:CHR$(27);"&";CH
R$(0);!004
260 PRINT #1:CHR$(65);CHR$(6
7);!162
270 PRINT #1:CHR$(137);!165
280 PRINT #1:CHR$(28)&CHR$(3
4)&CHR$(1)&CHR$(32);!190
290 PRINT #1:CHR$(17)&CHR$(9
6)&CHR$(129)&CHR$(34)&CHR$(2
8);!098
300 PRINT #1:CHR$(0)&CHR$(0)
;!044
310 PRINT #1:CHR$(9);!065
320 PRINT #1:CHR$(8)&CHR$(4)
&CHR$(10)&CHR$(1);!083
330 PRINT #1:CHR$(248)&CHR$(
65)&CHR$(42);CHR$(4);!247
340 PRINT #1:CHR$(8)&CHR$(0)
;!052
350 PRINT #1:CHR$(152)!238
360 PRINT #1:CHR$(28)&CHR$(3
4)&CHR$(65)&CHR$(8)&CHR$(85)
;!253
```

(See Page 29)

USING A SPREADSHEET—

(Continued from Page 23)

The last four columns under "Projections for the Widowed Spouse" are designed to reflect outcomes similar to those seen for"Joint Income." The "Planned Income Available" column (8) would include the recalculated Social Security benefits for the"surviving spouse" and any monthly pension payments if any joint survivor option was selected at the time of retirement. In this sample, I chose to withdraw all of my contributions plus interest and selected the maximum option leaving my spouse with no residual benefits. Was that callous of me? No, not really. I found that we could do much better re-investing my contributions than we could receive had we left my contributions in the pension system for my spouse. The premiums spent for life insurance were more than covered by the increase in benefit under the "maximum" option for me. That, of course, is one situation. Obviously, it is not necessarily applicable to every situation.

Any lump-sum proceeds from life insurance or annuities occasioned by the death of the retiree should be added to the Investment Balance column (10). In the sample spreadsheet, the Life Insurance (LINS) proceeds ($100,000) was added to the "Investment Balance" column (10) for each year as if death occurred during that year. Accordingly, the "Investment Balance" in column 10 begins with the Investment Balance for the previous year in column 6 minus the "Spouse Income Needed " in column 7 plus both the "Planned Income" (8) and the "Investment Yield" (9) plus the Life Insurance proceeds (LINS), which in this case is $100,000 It is *not* a continuous balance from one year to the next. That is the balance that should become the "Initial Investment" for the surviving spouse's new projection for the rest of his or her life.

USER NOTES

(Continued from Page 28)

```
370 PRINT #1:CHR$(8)&CHR$(65
)&CHR$(34)&CHR$(28)&CHR$(85)
;!253
380 PRINT #1:CHR$(27)&"%"&CH
R$(1)&CHR$(0)!076
390 FOR O=1 TO 26 !118
400 X=O+64 !090
410 FOR I=1 TO 20 !106
420 PRINT #1:CHR$(X);!151
430 NEXT I !223
440 PRINT #1:"" !015
450 NEXT O !229
460 CLOSE #1 !151
470 END !139
```

Circles galore

The following item is by Jim Lesher. It is another in a series of small programs that use The Missing Link.

Harry Wilhelm has written a much larger and more complex program as a demo of what The Missing Link is capable of. I will send you a copy if you send me a mailer and postage.

It is helpful to have graph paper to lay out your figures and to have a perspective of how the pixels are displayed on the screen. I use one-quarter inch squares and, for me, the numbers come out perfectly — 192 x 240. Each square equals six pixels. This lets me draw a form on the paper and know exactly where to place the dots.

This time we will create multiple circles, arranged in a beautiful pattern. You will believe squares can be made from circles, along with many other geometric shapes. The forms and shapes are limited only by your imagination. I look at the screen and see a pattern, blink my eyes, and see another.

In the following program we are varying all parameters of the circle — its X,Y position on the screen and its radius. If you prefer a different color, change the numbers in line 160 to suit your tastes.

The patterns here are linear, but in the future we may find we can make then nonlinear as well.

If you would like a copy of the prenumbered graph paper, or if you have questions about TML, send an SASE to Lesher at 722 Huntley, Dallas, TX 75214, or call him at 214-821-9274.

```
100 REM C6B !117
110 R=10 !059
120 Y=15 !071
130 X=00 !064
140 X=X+(R*1)!171
150 IF R>120 THEN 100 !209
160 IF X>220 THEN 200 !060
170 CALL LINK("COLOR",2,11)!
220
180 CALL LINK("CIRCLE",Y,X,R
,0)!075
190 GOTO 140 !219
200 Y=Y+(R*1)!173
210 IF Y>182 THEN 230 !098
220 GOTO 130 !209
230 FOR D=1 TO 500 :: NEXT D
 !245
240 CALL LINK("CLEAR")!055
250 R=R+5 !033
260 GOTO 120 !199
270 FOR D=1 TO 1000 !198
280 NEXT D !218
```

Cleaning contacts

This comes from Vern Jensen of Gretna, Louisiana:

A common problem on the TI is when it locks up, erasing any unsaved program or document you are working on. This is usually caused when the contacts of a cartridge get dirty. I have often taken my computer apart to get to the cartridge slot, so I could clean it. I have also used Q-Tips to clean the cartridge contacts, but this process often leaves tiny hairs on the contacts, which can also cause lock-ups.

In search of a better way, I tried using a Game Boy cleaning kit on the computer and cartridges. It was almost as if it were made for the TI! It comes with a cleaning cartridge that will fit in a Game Boy, but not in the TI. However, if you take the cleaning card out of the cartridge, it will nearly fit in the TI cartridge slot. Just cut off a millimeter on the top and right, and it will slide in smoothly. The card will only clean one side of the slot at once, so move the card in and out several times, then flip it over and repeat the process.

It is also useful for cleaning the speech synthesizer, expansion box cable and anything else you can fit it in. The cleaning kit also contains wands that you can rub back and forth on the kit also contains wands that you can rub back and forth on the cartridge contacts. I use Q-Tips to get most of the dirt off, and then finish the job with the wands. The Game Boy cleaning kits can be found in most stores, including Radio Shack, for about $10.

Load interrupt has many uses

The following item appeared in TopIcs, the newsletter of the Los Angeles 99ers. The author is uncredited.

The load interrupt, when activated, will cause the computer to suspend its current operations. Then it will look at a specific memory location that will tell it where to go for the next set of directions. This switch is useful for several utility programs. You may have a debugger or disassembler loaded in memory, along with the program you plan to check. When your running program cuts up, you can hit the load interrupt and be put into your debugger program. Then you can see what happened to your program in the computer's memory.

Another use is with screen dump routines. You may have a screen dump utility load in memory and your program. When you want a copy of a screen you hit the load interrupt switch, and then the screen dump routine takes over and you end up with a hard copy of what was on the screen.

Users can come up with their own utilities for the load interrupt switch as well.

Examine the diagram on the next page. Have someone competent with computers and electronics wire this and don't want to do it yourself. Carefully count the pins when making the connections. Triple check your work. (Remember, the reader is responsible for any problems which result from this hardware project.)

The switch and its connections may be placed anywhere on the I/O bus. Inside the console, the expansion box, or the speech synthesizer. The speech synthesizer is the most convenient location, and the switch may be glued or mounted under the cover

(See Page 30)

USER NOTES

(Continued from Page 29)

to prevent accidental use.

The best switch to use is a microswitch, such as Radio Shack No. 275-016. However, for console mounting a high quality push button switch will do. Make sure you get a Single Pole Single Throw (SPST) switch with normally open contacts.

Use of the siwtch when DFX-PRINT or another program which uses it is not loaded into memory, the console will lock up and require a reset back to the title screen. Pressing the switch a second time will confuse the computer if it is done before DFX-PRINT has finished the current print operation. Do not use disk file output with Super Sketch or other bitmap video programs.

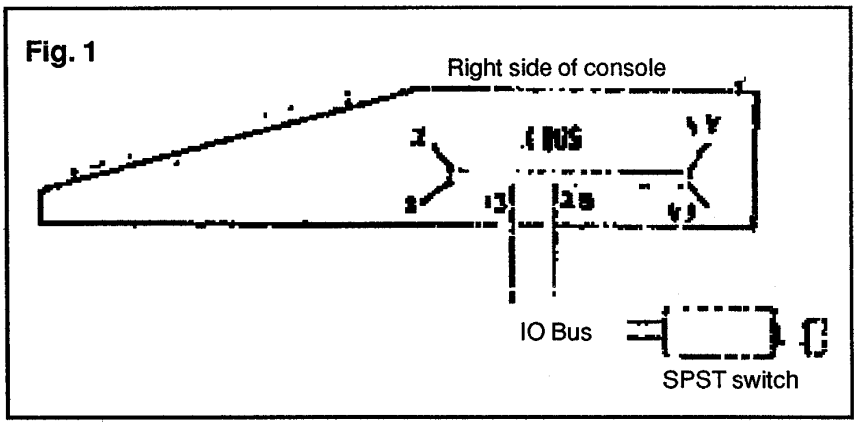

Fig. 1

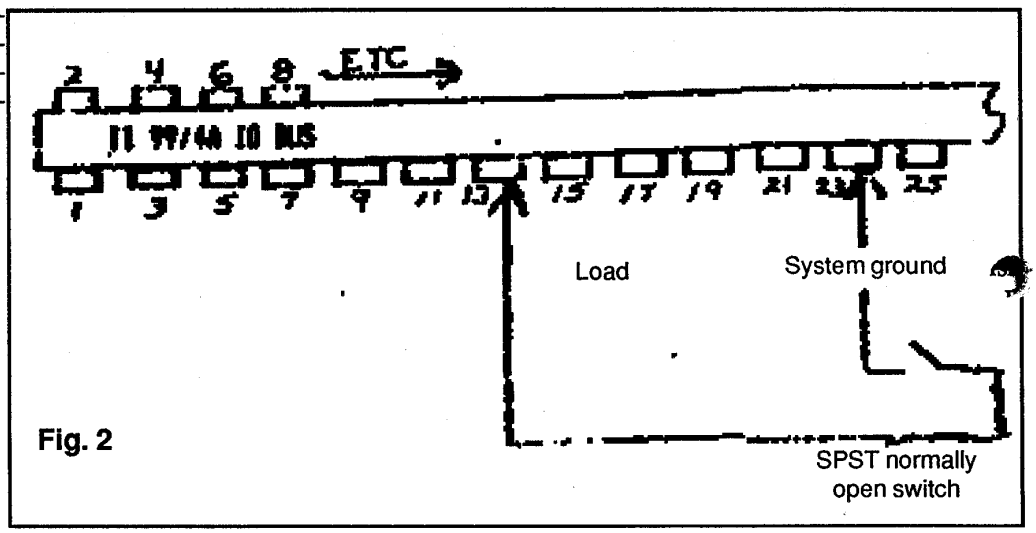

Fig. 2

Program makes graph paper

The following program appeared in the newsletter of the Erie 99er User Group. It requires a printer.

The program produces graph paper on dot matrix printers.

```
100 ! **********************
      *       GRAPH          *
      * from ERIE 99'er      *
      *Newsletter, May 1993*
      **********************
 !232
110 E$=CHR$(27)!157
120 A$=RPT$(CHR$(128),228)!0
52
130 B$=RPT$(CHR$(255)&SEG$(A
$,1,6),8)!145
140 B$=RPT$(B$&CHR$(255),4)!
234
150 A$=E$&"K"&CHR$(228)&CHR$
(0)&A$ !202
160 B$=E$&"K"&CHR$(228)&CHR$
(0)&B$ !204
170 OPEN #1:"PIO.CR" !195
180 FOR I=1 TO 11 !106
190 PRINT #1:E$;"@";E$;"3";C
HR$(24)!096
200 FOR J=1 TO 8 !064
210 PRINT #1:B$;B$;CHR$(10)!
234
220 NEXT J !224
230 PRINT #1:A$;A$;E$;"3";CH
R$(2)!132
240 NEXT I !223
250 PRINT #1:RPT$(CHR$(13)&C
HR$(10),9)!226
260 PRINT #1:E$;"@" !109
```

How random is random?

The following program is by Earl Raguse, of the User Group of Orange County, California. It appeared in the group's newsletter.

After having written a program to pick a random drawing winner, I began to worry about the randomness of its picks. So, I wrote the following program, and I ran it several times, first at 100, then at 5000. You decide if it produces random numbers. I think it does, especially if you ran a million of them.

```
100 ! SAVE DSK1.RANDOM? !051
105 ! Earl Raguse 7/93 !160
110 CALL CLEAR :: DIM Z(10):
: DISPLAY AT(12,1)ERASE ALL:
"How many random 0-9 Num
 bers?": :"           99" :
: ACCEPT AT(14,13)SIZE(-6):R
N !114
120 DISPLAY AT(12,5)ERASE ALL:"  WORKING ON RANDOM NUMBERS         Please Wait" !205
130 FOR I=1 TO RN :: DISPLAY
 AT(24,13):RN-I !208
```

(See Page 31)

USER NOTES

(Continued from Page 30)

```
140 RANDOMIZE :: N=INT(RND*1
0):: Z(N)=Z(N)+1 !073
150 NEXT I !223
155 CALL CLEAR :: DISPLAY AT
(2,2):"NUMBER FREQUENCY" !03
4
160 FOR I=0 TO 9 !063
170 DISPLAY AT(I+5,4):I;"
";Z(I):: NEXT I :: CALL CHAR
GE !079
175 DISPLAY AT(24,1):"Do it
again? or Print it A/P" !142
180 CALL GKEY(Q,24):: IF Q=A
SC("A")THEN 110 ELSE IF Q=AS
C("P")THEN 190 ELSE 180 !050
190 OPEN #1:"PIO" :: PRINT #
1:TAB(10);"NUMBER  FREQUENCY
" !162
200 FOR I=0 TO 9 :: PRINT #1
:TAB(12);I;TAB(20);Z(I):: Z(
I)=0 :: NEXT I :: PRINT #1:
:: CLOSE #1 :: CALL QUIT(K):
: GOTO 110 !091
210 END !139
220 !!131
230 ! SUBPROGRAM AREA !222
240 !!131
4000 SUB GKEY(Q,ROW)!006
4010 CALL KEY(3,K,S):: IF S<
1 THEN 4010 ELSE Q=K :: IF Q
=32 THEN A$="Space" ELSE A$=
CHR$(Q)!228
4020 DISPLAY AT(ROW,1)SIZE(3
0):"       You selected ";A$
 :: SUBEND !162
5000 SUB CHARGE !025
5010 ! By Earl Raguse 9/90 !
122
5020 CALL SOUND(125,554,0)::
 CALL SOUND(125,587,0):: CAL
L SOUND(125,659,0):: CALL SO
UND(125,784,0)!195
5030 CALL SOUND(125,110,30):
: CALL SOUND(200,659,0):: CA
LL SOUND(600,784,0):: SUBEND
 !250
6150 SUB QUIT(K)!104
6160 DISPLAY AT(23,1):" Pre
ss any key to proceed ":"  B
 Q quits, and B breaks" !2
47
6170 CALL KEY(0,K,S):: IF S<
1 THEN 6160 ELSE K=K AND 95
:: DISPLAY AT(24,9):"You pre
ssed ";CHR$(K)!217
6175 IF CHR$(K)="Q" THEN RUN
 "DISK1.DIR" ELSE IF CHR$(K)
="B" THEN STOP !030
6180 SUBEND !168
```

CLASSIFIEDS

Policy

The cost of classified advertising is 25 cents per word. Classified display (i.e., special formatting or graphics) is $9 per column inch. Classified advertisements must be paid in advance. Classified advertisers may request a category under which they would like their advertisements to appear, but the final placement decision is the responsibility of the publisher.

Classified deadlines will be kept open for as long as practical. For the purpose of classified advertising deadlines, any classified ad received later than the first day of any month cannot be assured of placement in the next edition. We will do our best to include every advertisement that is submitted in the earliest possible edition.

The publisher offers no guarantee that any advertisement will be published in any particular issue. Any damages that result either from errors in copy or for failure to be included in any particular edition will be limited to the amount of the cost of the advertisement itself. The publisher reserves the right to reject any advertisement.

The advertiser may elect to publish the advertisement in subsequent editions at the same charge, payable prior to publication. The deadline for carryover classifieds is the same as for new advertising.

In submitting an ad, please indicate whether you would like a refund if it is not published in the requested edition or whether you would like us to hold it for the next edition. Cancellations and refunds cannot be made after the second day of the month.

FOR SALE

FOR SALE

ENORMOUS TI99/4A INVENTORY. CATALOGS $2.00. **BRAATZS COMPUTER SERVICES**, 719 E. BYRD ST., APPLETON, WI 54911. **1-414-731-3478.** 10/12

Attend a TI Fair

Buy/Sell Used Hardware and Software

National Used Software/Hardware Club has buyers looking for TI products, as well as sellers. More than 170 TI items are listed for sale. The membership fee is only $15/year and entitles you to buy or sell computer items via the NUS/HC database. You will also receive FREE our newsletter (mailed every 8-12 weeks). If you don't buy or sell anything during the term of your membership we will refund your annual fee.

For a free, no-obligation information packet, write to NUS/HC, P.O. Box 1343, Round Rock, TX 78680; or call 512-255-1512.

Here are some of the items offered for sale:

Horizon RAMdisk (192K)	$125.00
Advanced Diagnostics	$25.00
Seagate 251 HD (40 meg)	$75.00
MG Explorer	$25.00

Call our BBS for weekly price and product updates — 512-255-1557. Hours are from 5:30 p.m. to 8 a.m. weekdays and from noon Saturday through 8 a.m. Monday weekends. Supports 300-2400 baud.

MICROpendium pays $10 for items submitted by readers and used in the User Notes column. Mail them to MICROpendium User Notes, P.O. Box 1343, Round Rock, TX 78680.

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 ❑ VISA ❑ DISCOVER ❑ (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.

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

Name _____

Address _____

City _____

State _____ **ZIP** _____

The set of numbers on the left 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; Vol. 4, No. 9

❑ **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

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 1.23F** (req. SSDD or larger, for MBASIC).....$4.00

❑ **MDOS 1.50H** (req. for MBASIC).....$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

SECOND CLASS