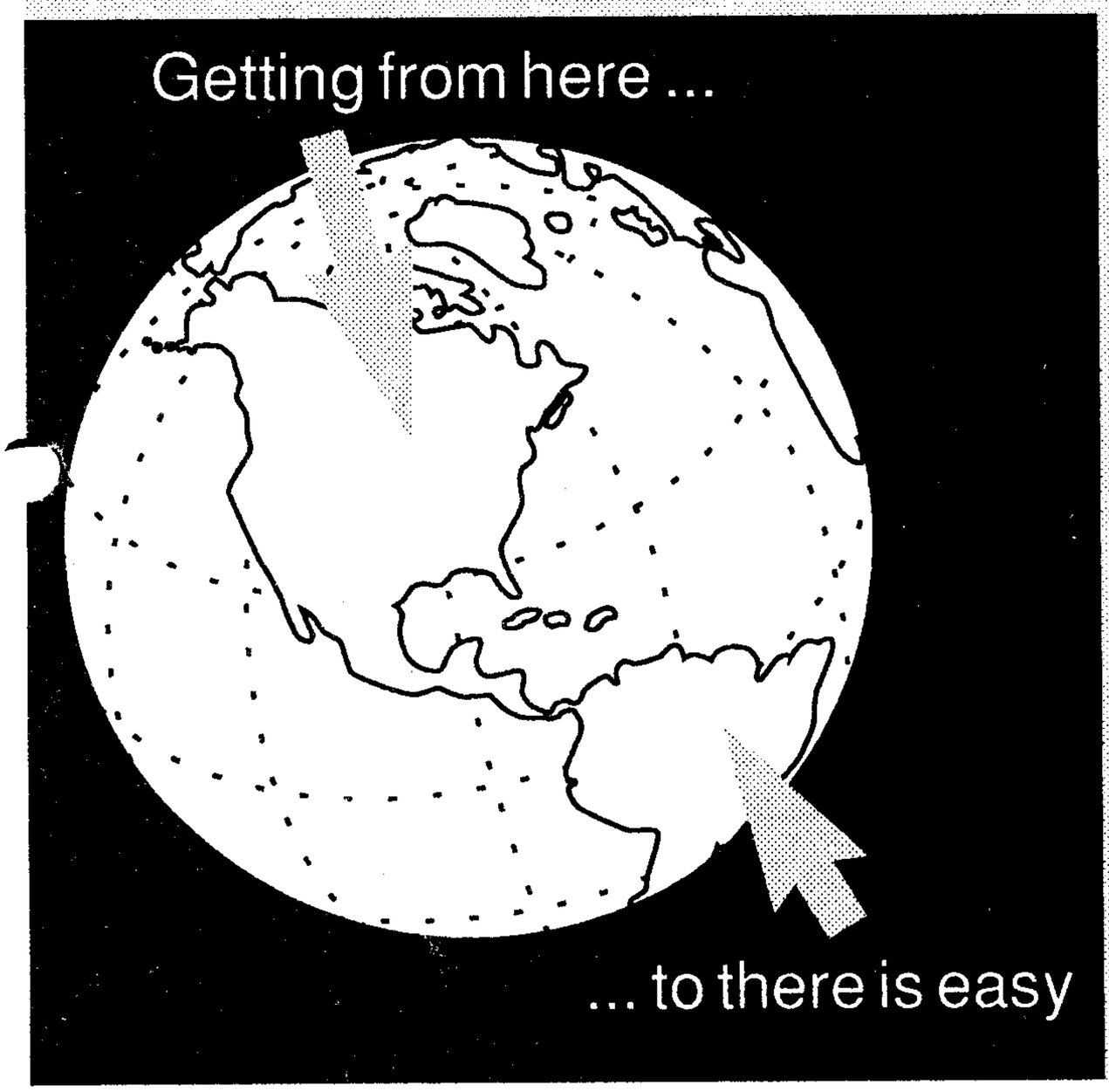
Covering the TI99/4A and the Myarc 9640

MICROpendium

Volume 10 Number 5 | June 1993 | $2.50

The Internet lets you travel the globe without leaving the comfort of your computer room. It puts you in touch with libraries and databases around the world and gives you access to information not available anywhere else. And you can do it for a few dollars an hour!



Page 20

MICROpendium Micro-Reviews columnist Stan Krajewski dies at 38.

Regena on key signatures

Bruce Harrison gets into another sort of assembly language sort

A two-column formatter for Page Pro in Extended BASIC

Contents

MICROpendium

MICROpendium (ISSN 10432299) is published monthly for $25 per year by Burns-Koloen Communications Inc., 16506 Terrace Dr., Austin, TX 78728-1156. 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.

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

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

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

Display advertising deadlines and rates are available upon request.

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

Foreign subscriptions are $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

Regena on BASIC

Major key signatures.....Page 8

The Art of Assembly

Another sort of sort.....Page 10

Page Pro formatter

An Extended BASIC program that produces two-column text for Page Pro 99.....Page 15

Internet and the TI

You don't have to leave your console to travel the globe.....Page 20

Newsbytes

Flash memory for the Geneve, music transcription from Harrison Software, and MDOS V1.50.....Page 22

The Lima tapes

Products and networking highlight conference.....Page 23

Reviews

Extended BASIC III Super Module.....Page 23

User Notes

Lithium batteries and RAMdisks, improvements to Screen Dump, and another routine for The Missing Link.....Page 27

Classified .....Page 31

*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

In memory of Stan Krajewski

I regret to inform our readers that Stan Krajewski, who wrote our MicroReviews column, is dead.

Stan died on May 21at the age of 38. The cause of death , according to his mother, was a blood clot to the brain.

It's difficult to write about someone one has never met. Conversations between Stan and me occurred over a phone, and occasionally by mail. I learned about his death from Don Walden, president of the Chicago TI User Group. At times like this I wish I'd known him better, but I think I got to know him well enough by reading his articles and through our infrequent conversations.

He was as reliable as anyone I've ever worked with. His columns always appeared on time, and in this business that's saying a lot. I liked the way that he approached the task of reviewing the scores of non-commercial programs that came his way. He was out front with his opinions, though I didn't always agree with his conclusions. He definitely wasn't afraid of having an opinion and letting others know about it. But I guess that's what a reviewer is expected to do.

But Stan went further than that. He encouraged readers and programmers to call him, whether they agreed with him or not. He didn't hide behind the facade of a byline. If you wanted to reach him, all you had to do was look at the end of his monthly column. That's where he listed his address and his phone number. How many writers are willing to expose themselves to that extent. Not many.

Stan leaves three sons, ages 18, 16 and four. His obituary is on page 7

TYPO IN AD

A typographical error in MICROpendium's back page ad last month suggested that we had lowered the subscription price for Canadian subscribers to $25, from the current $32.50. Although we'd like to do it, we can't without losing money. Many believe the price difference the two rates is due to the difference in the exchange rate. That's not exactly true. The biggest difference lies in the postage rates. We mail via second class mail in the U.S. at a much lower price than the first class rate we have to pay for Canadian delivery. It costs us 3-4 times what it costs to mail MICROpendium in the states. Basically, what Canadian subscribers are paying extra for is delivery. And there's just nothing we can do about that. Though we are always open to suggestions.

Anyway, the typo occurred because we redid the ad last month and we just didn't proof it well enough. Sorry for any inconvenience this may have caused.

WE'D LIKE TO SEE READER-TO-READER SOLUTIONS

Our Reader-to-Reader column has become a popular way for readers to get answers to their trickiest problems. However, the answers to many of the questions might be of interest to more than one person. So, I'm asking that anyone who receives a response forward a copy to us so that we can include it for use by other readers.

TI FAIR TO MOVE SITE

There's going to be a change of location for the 1993 Chicago TI Faire, according to Don Walden, president of the Chicago TI User Group. Details and dates are still being worked out,but the group has been looking at a community college site to save expenses for attendees and for vendors. (The first TI fair ever was Chicago's, and it was held at Triton Community College, so a site like that would be a return to traditional roots.) "We're going to tighten our belts, and we're going to have a fair where the vendors will pay the least amount that they have to," Walden said.

Another change will be the elimination of the pre-fair party and the post-fair banquet. Expect the date to be Oct. 30, Nov. 6 or Nov. 13.

–JK

Subscription price to increase

After agonizing over a price increase for the past six months, it's become unavoidable. As of July 31, 1993, the annual subscription rate to MICROpendium will increase to $35 in the U.S. Canadian and overseas subscriptions will increase an additional $10 from the current rates. Bulk subscription rates and the rate for individual copies will also increase appropriately. The rate for single copies will increase to $3 each. Although we would like everyone to renew at the new rate, we will continue to accept renewals and back issue orders at the current rate until July 31.

Subscribers may renew or extend their subscriptions at the current rates until July 31, after which the new rates will go into effect.

The rate increase is the result of reduced advertising revenues and increased costs. I'd also like to think that it wouldn't be necessary at this point if everyone who reads MICROpendium actually had a subscription instead of borrowing someone else's copy. It should be noted that MICROpendium has not had an increase in its subscription price since October 1989.

Unfortunately, this price increase does not necessarily mean we will be able to increase the number of pages in MICROpendium. At the very least, however, it means we will be able to hold our ground.

Readers who are on fixed incomes or otherwise may not be able to afford the increase in price are asked to notify us when renewing so that we can do what we can to accommodate them. We hope the majority of readers will understand if not embrace the rate increase. There is simply no other option.

I feel terrible about this, but I know I have no choice. I hope that you will continue to support us now just as you have in the past.

—John Koloen

Feedback

'Good' news

On April 8 I mailed a package to Dr. Good of the Lima, Ohio, Users' Group, taking him up on his kind offer in MICROpendium to provide the Funnelweb V5 Editor. I presumed upon his kindness further by asking if he could include Funnelweb V4.4, which I had been unable to find. Not knowing its size, I sent five DSSD disks with my request.

Within 10 days the mailer returned with everything I had asked for and more. Besides the programs mentioned, Dr. Good included his own article on the Editor and some extra Help files, with valuable instructions on the disk labels. He also filled the surplus fifth disk with a set of programs he thought I might find useful.

As a matter of interest, I also ordered programs from Texaments and Asgard on the same day that I mailed my package to Dr. Good. The Texaments program, held up by Canada Customs, arrived a week after Dr. Good's reply, and I am still (as of May 14) waiting for the order from Asgard.

The Funnelweb programs themselves are yet another example of the McGoverns' excellent contributions to the TI world.

I was about to write to thank Dr. Good for his exceptional kindness when it occurred to me that a public expression of appreciation would be more appropriate. After all, without the selfless efforts of people like Dr. Good, our TI world would have fallen apart long ago. Thanks a lot, Doctor.

Al McLellan
Halifax, Nova Scotia, Canada

Broken spokes need fixing sometimes

I have been subscribing to MICRO for almost four years with keen interest sponging ideas and thoughts for my 4A. I, like many others who started, acquired this "toy" for a fraction; some have even gotten theirs for free!

After acquiring past copies dating back to day one and continuously reading and rereading for a glimmer of light as to "How do I do ... ?" "What is this for or do ... ?" "How do I get this or that ... ?", etc. I have found when going to 4A meetings, the talk has been about GROM, GRAM, VDP, XB, EA, SS/SD, DS/DD, DS/SD, modem, SIG, etc. I was reluctant to ask or inquire what they were talking about or even to abruptly stand up and holler, "*Stop, I am confused. What are you talking about, now?*" Over the years, I have met club members who would offer thoughts and ideas and would sympathize, but would also add "We had to learn by reading and talking when there was no one around like now to ask for help. You can't hurt the 4A, besides, there are quite a few around for next to nothing compared to what we had to pay for it." *They have been right!*

I have found you need to spend time beating the keys, asking and re-asking questions to people who are probably tired of telling you over and over "how to," but are still willing to listen and offer help and in many instances cables, drives, disks and probably the four most valuable things a TIer can offer: compassion/sympathy, understanding, help and help!

While you may not want to reinvent the wheel, sometimes a spoke or two gets broken or is missing and a refresher course or simple suggestion(s) may help to align the bewildered user.

I just acquired a modem and am learning. I have talked to boards or have tried to, but have found some non-existent and in some instances a private party — *oops!* This may not seem bad, but since I don't get home until o'dark-thirty — *oops-oops!*

Mike Scheller
Casa Grande, Arizona

'Albatross' still flies

I have saved every copy of your publication since the beginning of 1985 and refer to them quite often. When an article appears that refers to a specific issue for reference, I have the advantage of reviewing that article to refresh my memory. As I am approaching 80 years of age, my memory is not quite as sharp as it used to be.

When I first became acquainted with the TI99/4A in 1983 and learned it was to become an albatross, I bought four of them plus TV monitors, external drives, printers, expansion systems and all the software TI had available at that time. This was to be my total exposure into the computer age. I have enjoyed every minute of it. My 99/4A is now my friend and a big part of my life.

Keep up the good work to bring us the rest of the TI story. The best is yet to be!

Oscar G. Hook
Boulder City, Nevada

Get together again

While many TI99/4A user groups have ceased to exist, and many others struggle to survive, thousands of TIers are going it alone, without a user group and without a bulletin board.

Many are old-timers who have lost track of their local group. Many, many others are "new kids on the block." They acquired cast off computers during the past few years, and don't know a user group or a BBS for TIers exists. I hope a good percentage of these people read MICROpendium. (All of them should!)

I believe MICROpendium would welcome the opportunity to bring these "Ic ers" together with their local user groups and bulletin boards *if* the editor was provided the date, time and location of meetings, and the name and phone number of BBSes where available.

The Western New York 99ers Users Group (serving the Buffalo, New York, area and southern Ontario, Canada) is alive and well, meeting the second Tuesday of every month (except July and August), 7-9 p.m., at the Williamsville Library, 5571 Main St., Williamsville, New York.

Our BBS is "The AM-CAN FRIENDS" at (716) 835-5316, running 24 hours, seven days a week, at 300-1200-2400 baud, 8N1. In the past 12 months we have logged about 4,000 calls and 5,000 messages.

Come to our meetings and call our BBS. New people are always welcome. We'll try to help you!

Editor, please note!

Last week, at least four members of the WNY 99ers UG traveled to Lima for the Faire. It's a long drive from Buffalo to Lima!

While there, they met several Buffalo TIers who didn't know our group or our BBS existed. They *do* now! Imagine how

(See Page 7)

Feedback

Continued from Page 6)

many "loners" didn't get to Lima, from all over the U.S. and Canada.

Please urge your readers to write you, as I have, making you aware of user groups and BBSes. Then publish this info.

It may also be a good idea to ask if other people are interested in forming a user group, publishing their phone number so others may contact them.

Restoration of the TI community won't just happen! We have to make it happen!

In Buffalo, we have worked hard to promote our group and BBS, with some success, but we can always use more help. Brad Snyder (whom you know, or know of), in the Allentown, Pennsylvania, area, really needs help with his "The First Floor BBS" (215) 760-0527. Brad set up this BBS about 11 months ago. To date, he has logged only about 400 calls, compared to our 4,000.

You have *always* helped *all* of us! I believe you can really turn this thing around and put new life into the TI community with this little effort. We'll all appreciate it. Thank you.

James P. Cavanaugh
SysOp, AM-CAN FRIENDS BBS
Eggertsville, New York

Warning about power supply project

You better know what you are doing before jumpering voltage regulators as noted in Al Beard's article in the April MICROpendium ("Gaining peace of mind").

The main problem is the statement "The modification involves jumpering out of *all* the voltage regulators on each board." This is not correct for the TI Cards P Code, 32K Memory and Disk Controller. There is a -5V regulator on each of these cards that must not be jumpered out! The saving grace is that, on at least most of these cards that were made, the -5V regulators are 79L05 and look like a plastic transistor and would, therefore, be overlooked by the instructions in this article. Anyone doing modifications on third party cards should be aware that only 7805, 78M05, 7812, 78M12, 7912 and 79M12 voltage regulators may be jumpered out. *No 79l05, 79M05 or 7905 regulators should be jumpered out!*

Another problem is that on many of the TI RS232 cards there is a 79L12 regulator that should be jumpered out and this is one of those plastic transistor look-alikes whose exact pinout may vary — get help on this one!

Repaired cards may also be a problem as someone may have substituted a 7905 for a 79L05 which works well, except it must not be jumpered out.

I hope this saves some unnecessary repairs.

Jack Miller
Trenton, Michigan

Big Foot extender with CorComp

Since I haven't contributed any new items to MICRO in quite some time, I thought I'd pass along my latest discovery

While at the Boston Faire in April, I met Frank Billeri who had a fine assortment of TI items. I stocked up on many "spare parts." Everything worked fine except for the "Big Foot" extender. When it worked as it should in my back-up system, I knew I was in trouble.

By now, I'm sure most of us have mongrelized our systems with TI, Myarc, CorComp and specialized chips.

Which one was not compatible with the extender cable?

After many frustrating hours, it turned out to be the CorComp 32K Memory Card, which otherwise works perfectly.

My growing box of mysterious parts had an old but good TI 32K card, and all is well again in Naples.

It would be interesting to know if anybody else has had this problem.

Bob Zink
Naples, Florida

***Feedback is a reader forum. The editor may condense excessively lengthy submissions if necessary. We ask that writers limit themselves to one subject per submission. 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.**

MICRO-Reviews columnist Stan Krajewski, 38, dies

Stanley J. Krajewski, age 38, of Live Oak, Florida, died May 21 at Tallahassee Memorial Hospital of a blood clot to the brain.

Krajewski wrote the MICRO-Reviews columnist for MICROpendium. He took over the column begun by Harry Brashear in August 1991. His final column appeared in the May 1993 issue. Krajewski had formerly been active in the West Palm Beach 99ers.

Funeral services were held under the auspices of the Charles T. Hall Funeral Home. Burial was in the Florida National Cemetery at Bushnell.

Krajewski was a United States Air Force veteran who served in the Korean and Vietnam conflicts.

He is survived by his wife, Nancy Krajewski; three sons, ages 18, 16 and 4; his mother, Helen Bucchieri of Margate, Florida; and one sister, Mila Mazzara of Flushing, New York.

BASIC

Major key signatures

By REGENA

This month's program is a quiz to help music students learn the names of the major key signatures when a certain number of sharps or flats are shown. For example, if there are three sharps (F#, C#, G#) in the signature, the corresponding major key is "A."

One quiz consists of 10 questions with possible sharps in the signature, and the other quiz consists of 10 questions with possible flats in the signature. The key of C Major is a possibility in both quizzes.

After the key signature is shown, the student must press the correct letter for the major key. If the major key is F-sharp, the sharp sign is added. If the major key contains the word "flat," only the letter must be pressed, and a flat is printed automatically.

When an incorrect letter or symbol is pressed, there is an "uh-oh" sound and the student must keep trying. When the correct letter is pressed for the major key, the major scale for that key is played.

Lines 120-310 print the title screen and define graphics characters. The DATA statements contain character definitions for the treble clef and the staff lines. The underline key, Character 95, is redefined as a line in the middle of the character and is used for drawing staff lines. L$ contains several of the underline characters. Character 35 is a sharp, and Character 36 is a flat.

Lines 320-350 print the options for the two quizzes or ending the program, then Line 360 branches appropriately. Lines 370-480 are a subroutine that finish printing the instructions, read in values for B(C), which are the keys in ascending order of number of sharps or flats, and print the treble staff. The DATA statements for B(C) are in lines 510 and 1190.

Lines 530-900 contain the quiz for the sharps. A random number A is chosen for the number of sharps. The variable PA is used so the same key signature will not be shown twice in a row. Lines 580-630 read in row and column variables and place the appropriate number of sharps. Lines 640-700 wait for a key to be pressed. Lines 710-740 are the procedure if the letter pressed is incorrect. Lines 750-770 print the correct name of the major key, adding a sharp if necessary for the key of F#.

Line 780 uses subroutines to RESTORE the proper DATA statement containing the frequencies for playing the scale. Lines 790-800 print a red symbol for each correct answer. Lines 810-840 play the scale. Lines 850-900 erase the sharps and the key and go to the next problem.

Lines 910-950 print the option when the quiz is finished. The student will either go back to the main menu screen to choose a quiz again or will end the program.

Lines 960-1160 are the subroutines containing the frequencies for the major scales for sharps.

Lines 1170-1590 contain the procedure for the quiz for flats and are similar to Lines 530-900. Lines 1610-1810 are the subroutines containing the frequencies for the major scales for key signatures that have flats.

Lines 1820-1830 clear the screen and end the program.

If you wish to save typing effort, you may have a copy of this program by sending $4 to *REGENA, 918 Cedar Knolls West, Cedar City, UT 84720*. Be sure to specify that you need "Major Key Signatures" for the TI and whether you need cassette or diskette.

MAJOR KEY SIGNATURES

```
100 REM  MAJOR KEY    !220
110 REM BY REGENA !071
120 CALL CLEAR !209
130 PRINT "   MAJOR KEY SIGN
ATURES" !129
140 CALL CHAR(136,"000028107
C1028")!177
150 CALL COLOR(14,7,1)!229
160 PRINT :TAB(8);"BY REGENA
" !172
170 CALL CHAR(36,"202C342424
28302")!185
180 CALL CHAR(35,"2424FF2424
FF2424")!044
190 L$="_____
" !117
200 FOR C=91 TO 126 !214
210 READ C$ !254
220 CALL CHAR(C,C$)!081
230 NEXT C !217
240 DATA 071F3F78F0E0C08,00C
0F0F078381818,80808080808080
8,181818101020202,000000FF,8
08040FF4040404 !172
250 DATA 202040FF4080808,212
1212122321212,000000FF000001
03,1C1C18FF78E8C884,00000000
00010307,070F1E3CF8F0E08 !17
6
260 DATA 8404040404020202,0F
1E1EFF3C387878,020202FF03070
F1F,000000FFFFFFC,000000FFFC
FE0F03,000000FF000080C !013
270 DATA 70F0F0E0E0E0E0E,3D3
97160E0C0C0C,000000808080808
,C0E060303010101,E0E0E0FF707
03838,C0C0E0FF6060301 !173
280 DATA 808080FF4040404,101
010FF1020202,3C1C1E0F070301,
0000000080C0F0F8,40402020202
0202,000000010306081 !093
290 DATA 40C0808,3C0F03FF,00
00C0FF,20213EFF2020202,60800
0FF,101010101010101 !195
300 CALL CHAR(37,"1010103020
408")!051
310 CALL CHAR(38,"3C7EFEFEFC
7C391E")!160
320 PRINT : : :TAB(6);"1  SH
ARPS": :TAB(6);"2  FLATS": :
TAB(6);"3  END PROGRAM": : :
!038
330 CALL KEY(3,K,S)!190
340 IF (K<49)+(K>51)THEN 330
 !095
350 CALL CLEAR !209
360 ON K-48 GOTO 490,1170,18
20 !201
370 PRINT :"NAME THE MAJOR K
EY THE      SIGNATURE REPRES
ENTS BY     PRESSING THE LET
TER NAME.": : :!016
```

(See Page 9)

REGENA ON BASIC—

(Continued from Page 8)

```
380 FOR C=0 TO 6 !054
390 READ B(C)!137
400 NEXT C !217
410 PRINT :"PRESS <ENTER> TO
 START.";!039
420 CALL KEY(0,K,S)!187
430 IF K<>13 THEN 420 !154
440 CALL CLEAR !209
450 SC=0 !077
460 PRINT "    [\":"    ]^":
" ___`a__";L$:"    b":" __cd
___";L$:"  efg":" _h_ijkl";L
$ !025
470 PRINT "  m no p":" _q_rs
_t";L$:"  uv wxy":" __z{|}_"
;L$:"     ~":"     ~":"    &
%": : : : : : :!222
480 RETURN !136
490 PRINT "YOU WILL SEE A TR
EBLE STAFF WITH SHARPS." !04
3
500 RESTORE 510 !092
510 DATA 67,71,68,65,69,66,7
0 !059
520 GOSUB 370 !195
530 FOR C=1 TO 10 !099
540 RANDOMIZE !149
550 A=INT(7*RND)!213
560 IF A=PA THEN 550 !181
570 PA=A !144
580 RESTORE 590 !173
590 DATA 6,11,9,13,5,15,8,17
,11,19,7,21 !141
600 FOR J=1 TO A !128
610 READ X,Y !251
620 CALL HCHAR(X,Y,35)!134
630 NEXT J !224
640 CALL HCHAR(23,13,75)!054
650 CALL HCHAR(23,14,69)!058
660 CALL HCHAR(23,15,89)!061
670 CALL HCHAR(23,17,63)!055
680 CALL SOUND(100,1497,3)!1
91
690 CALL KEY(0,K,S)!187
700 IF S<1 THEN 690 !189
710 IF K=B(A)THEN 750 !230
720 CALL SOUND(80,330,2)!084
730 CALL SOUND(80,262,2)!088
740 GOTO 680 !249
750 CALL HCHAR(23,17,K)!079
760 IF K<>70 THEN 780 !007
770 CALL HCHAR(23,18,35)!055
780 ON A+1 GOSUB 960,990,102
0,1050,1080,1110,1140 !158
790 SC=SC+1 !165
800 CALL HCHAR(2,15+SC,136)!
138
810 FOR Y=1 TO 8 !079
820 READ T !235
830 CALL SOUND(100,T,2)!113
840 NEXT Y !239
850 FOR X=6 TO 12 STEP 2 !04
4
860 CALL HCHAR(X,11,95,15)!0
66
870 CALL HCHAR(X-1,11,32,15)
!245
880 NEXT X !238
890 CALL HCHAR(23,13,32,6)!2
25
900 NEXT C !217
910 PRINT "GOOD WORK!" !026
920 PRINT :"PRESS 1 TO TRY A
GAIN":TAB(7);"2 TO END PROGR
AM";!237
930 CALL KEY(0,K,S)!187
940 IF K=49 THEN 320 !126
950 IF K=50 THEN 1820 ELSE 9
30 !071
960 RESTORE 970 !042
970 DATA 262,294,330,349,392
,440,494,523 !176
980 RETURN !136
990 RESTORE 1000 !072
1000 DATA 392,440,494,523,58
7,659,740,784 !199
1010 RETURN !136
1020 RESTORE 1030 !103
1030 DATA 294,330,370,392,44
0,494,554,587 !184
1040 RETURN !136
1050 RESTORE 1060 !133
1060 DATA 440,494,554,587,65
9,740,831,880 !198
1070 RETURN !136
1080 RESTORE 1090 !163
1090 DATA 330,370,415,440,49
4,554,622,659 !175
1100 RETURN !136
1110 RESTORE 1120 !193
1120 DATA 494,554,622,659,74
0,831,932,988 !203
1130 RETURN !136
1140 RESTORE 1150 !223
1150 DATA 370,415,466,494,55
4,622,698,740 !191
1160 RETURN !136
1170 PRINT "YOU WILL SEE A T
REBLE STAFF WITH FLATS." !21
1
1180 RESTORE 1190 !007
1190 DATA 67,70,66,69,65,68,
71 !059
1200 GOSUB 370 !195
1210 FOR C=1 TO 10 !099
1220 RANDOMIZE !149
1230 A=INT(7*RND)!213
1240 IF A=PA THEN 1230 !095
1250 PA=A !144
1260 RESTORE 1270 !087
1270 DATA 10,12,7,14,11,16,8
,18,12,20,9,22 !229
1280 FOR J=1 TO A !128
1290 READ X,Y !251
1300 CALL HCHAR(X,Y,36)!135
1310 NEXT J !224
1320 CALL HCHAR(23,13,75)!05
4
1330 CALL HCHAR(23,14,69)!05
8
1340 CALL HCHAR(23,15,89)!06
1
1350 CALL HCHAR(23,17,63)!05
5
1360 CALL SOUND(100,1497,3)!
191
1370 CALL KEY(3,K,S)!190
1380 IF S<1 THEN 1370 !104
1390 IF K=B(A)THEN 1430 !145
1400 CALL SOUND(80,330,2)!08
4
1410 CALL SOUND(80,262,2)!08
8
1420 GOTO 1360 !164
1430 CALL HCHAR(23,17,K)!079
1440 IF K=70 THEN 1470 !250
1450 IF K=67 THEN 1470 !000
1460 CALL HCHAR(23,18,36)!05
6
1470 ON A+1 GOSUB 1610,1640,
1670,1700,1730,1760,1790 !11
7
1480 SC=SC+1 !165
1490 CALL HCHAR(2,15+SC,136)
!138
1500 FOR Y=1 TO 8 !079
1510 READ T !235
1520 CALL SOUND(100,T,2)!113
1530 NEXT Y !239
1540 FOR X=6 TO 12 STEP 2 !0
44
1550 CALL HCHAR(X,12,95,15)!
067
```

(See Page 8)

THE ART OF ASSEMBLY — PART 24

Another sort of sort

By BRUCE HARRISON

Last month we showed an entire program to sort a self-contained list of 75 strings. Toward the end of that column, we introduced the idea that we call "pre-sort" for sorting things as they are received from a source. It occurred to us that this left another "shoe to drop" on the subject, so today we're dropping that other shoe.

In the sidebar is another complete Option-3 E/A program that you can type in, assemble, and run for yourself. It will prompt for an input file name, then will take the named input file, treating each record as a string, and sort the entire file as received. When it's done, it will prompt for an output file name, and given one it will save the sorted contents of the input file to this output file.

The sort process is fairly simple. As each record is gotten from the file, it will be compared to each record already in memory. When a record is found that's bigger than the incoming record, all of the records currently in the array from there to the end are moved downward by one more than the length of the incoming record, then the incoming record is slipped into its proper place in the array. A zero byte is placed just after the last string in the array to mark the end. If no "bigger" string is found in the array, the incoming one is placed at the end of the array.

We have of course tested this program extensively, and it seems to work exactly as advertised. Error traps are built in to protect the user from errors of almost any kind, including trying to sort a file that's too big for the 24K of high memory. We'll get into what happens on errors in just a bit, but first let's discuss the performance of the program.

PERFORMANCE

The first test we performed was to take a file made from the 75 strings that were in last month's program. From a normal floppy disk drive, it takes about six to seven seconds to sort that list of

(See Page 11)

SIDEBAR 24

```
* FILE SORTING PROGRAM
* BY B. HARRISON
* 22 JUN 1992
* PUBLIC DOMAIN
*
* REQUIRED REFERENCE VECTORS
*
      REF  VMBW,VMBR,VSBW,VSBR
      REF  DSRLNK,KSCAN
*
* REQUIRED EQUATES
*
STATUS EQU  >837C
WS     EQU  >20BA
GPLWS  EQU  >83E0
SCRWID EQU  32
PAB1   EQU  >1000
BUF    EQU  >1050
PABPNT EQU  >8356
KEYADR EQU  >8374
KEYVAL EQU  >8375
*
      DEF  START        DEFINE ENTRY POINT
*
      AORG >2678        SET ORIGIN IN LOW MEMORY
*
* CODE SECTION - MAIN PROGRAM
*
START
      LWPI WS           LOAD USER WORKSPACE
      LI   R15,RTNSTK   SET STACK FOR HIGH LEVEL SUBROUTINE
      LI   R0,3         ROW 1, COLUMN 4
      LI   R4,3*SCRWID  THREE ROWS TO CLEAR
      BL   @CLRFLD      CLEAR THEM
      LI   R1,INFSTR    SET FOR INPUT FILE PROMPT
      BL   @DISSTR      DISPLAY PROMPT
      AI   R0,SCRWID    DOWN TO ROW 2
      LI   R4,15        15 BYTES
      BL   @CRSIN       USE CRSIN SUBROUTINE
      CI   R8,15        HAS F-9 BEEN STRUCK?
      JNE  PLCFN        IF NOT, GO ON
      B    @EXIT        ELSE EXIT PROGRAM
PLCFN
      LI   R9,TEMSTR    POINT AT TEMPORARY STRING
      LI   R10,PAB1DT+9 POINT R10 AT FILE DESCRIPTOR LENGTH BYTE
      BL   @MOVSTR      MOVE STRING FROM TEMSTR TO PAB DATA
      LI   R4,>A000     SET R4 TO >A000
      MOV  R4,@ENDSTR   MOVE THAT TO END OF ARRAY LOCATION
      CLR  *R4          MAKE >A000 EQUAL 0
OPNF1
      LI   R0,2*SCRWID+2 SET FOR ROW 3, COL 3
      LI   R1,SFSTR     "SORTING FILE" MESSAGE
```

REGENA ON BASIC—

(Continued from Page 9)

```
1560 CALL HCHAR(X-1,12,32,15
)!246
1570 NEXT X !238
1580 CALL HCHAR(23,13,32,6)!
225
1590 NEXT C !217
1600 GOTO 910 !224
1610 RESTORE 1620 !183
1620 DATA 262,294,330,349,39
2,440,494,523 !176
1630 RETURN !136
1640 RESTORE 1650 !213
1650 DATA 349,392,440,466,52
3,587,659,698 !207
1660 RETURN !136
1670 RESTORE 1680 !243
1680 DATA 233,262,294,311,34
9,392,440,466 !172
1690 RETURN !136
1700 RESTORE 1710 !017
1710 DATA 311,349,392,415,46
6,523,587,622 !181
1720 RETURN !136
1730 RESTORE 1740 !047
1740 DATA 208,233,262,277,31
1,349,392,415 !169
1750 RETURN !136
1760 RESTORE 1770 !077
1770 DATA 277,311,349,370,41
5,466,523,554 !177
1780 RETURN !136
1790 RESTORE 1800 !108
1800 DATA 185,208,233,247,2
7,311,349,370 !172
1810 RETURN !136
1820 CALL CLEAR !209
1830 END !139
```

THE ART OF ASSEMBLY—

(Continued from Page 10)

words, including the time to open the file. The sort produces an output file that has all 75 of those records in alphabetical order.

Having succeeded with that short file, we decided to try something bigger. We took the D/V 80 file of one of these columns (No. 3) which has 231 records, and occupies 65 sectors on a DS/SD disk. This took about one minute to read and sort. The result was a rather interesting output file. All the blank lines (blank lines are records consisting of a single space) came first. These were followed by a group of lines indented by 26 spaces, then those indented by less than 26 spaces, then those that had the five character indent for paragraph beginnings, and finally a large group of lines which had no indent.

Within each group, the sorting was done as you'd expect. Lines that had characters like parentheses or numbers came first, then those starting with capital letters, then those starting with lower case letters. In other words, the sort worked exactly as it should. One minute didn't seem like much time to read and sort this file. For comparison purposes, we went into the E/A editor and loaded the same file from the floppy disk. It took about 28 seconds to simply load in for editing, or nearly half the time our program took to load and sort the file. Doing this sort operation on a file from RAMdisk takes much less time.

THE ERROR REPORTS

There are two major kinds of errors that will be trapped by this program. First, of course, are file errors. These are reported to the user on rows 23 and 24 of the screen. After the error report, pressing any key takes the user back to the prompt for the appropriate file name. If an error occurs on the input file, you return to that prompt to try again. If an error occurs on the output file, you return to that file name input field. The sorted file in such cases is still in memory, so you get a second try at saving the sorted file to disk.

The other possible error is running out of memory. As the program builds the array of sorted strings in high memory, it checks with each added string for the end of usable memory at >FFE7. If that would be exceeded, the error is reported, and after a keypress you're at the output file name prompt so you can save what's in memory. In general, any D/V 80 file that's small enough to be edited by E/A's editor will not run this program out of memory. The capacity we allow is 24,550 bytes. This is because the program itself is all in low memory, so that all the high memory from >A000 through >FFE6 can be used for the string array.

USER GUIDANCE

The user interface in this program is very simple. At the file name prompts, function key presses may be used to delete (Fctn-1) or toggle into insert (FCTN-2), or move the cursor left and right (FCTN-S or FCTN-D). Function-9 is reserved for two purses. If you're at the output name prompt, F-9 takes you back to the input name prompt. If you're at the input name prompt, F-9 takes you out of the program. When a sort has finished being saved to disk, the program returns to the input file name prompt,

(See Page 12)

```
      BL   @DISSTR      DISPLAY THAT
      MOVB @INMD,@PAB1DT+1 OPEN WILL BE INPUT MODE
      LI   R0,PAB1      SET WRITE ADDRESS IN R0
      MOVB @PAB1DT+9,R2 GET DESCRIPTOR LENGTH BYTE INTO LEFT BYTE R2
      SRL  R2,8         RIGHT JUSTIFY SO R2 IS A WORD OF LENGTH
      AI   R2,10        ADD 10 TO INCLUDE THE PAB1DT LINE PLUS DESCRIPTOR
      LI   R1,PAB1DT    POINT R1 AT PAB DATA
      BLWP @VMBW        WRITE BYTES TO PAB LOCATION IN VDP RAM
      AI   R0,9         ADD NINE TO ADDRESS IN R0
      MOV  R0,@PABPNT   PLACE THAT ADDRESS AT >8356
      CLR  @STATUS      CLEAR GPL STATUS
      BLWP @DSRLNK      USE DSRLNK UTILITY
      DATA 8            REQUIRED DATA
      STST R14          STORE STATUS REGISTER IN R14
      ANDI R14,>2000    MASK ALL BUT BIT #2 IN R14
      JEQ  RDF1         IF ZERO, GO AHEAD TO READ FILE
      BL   @OPNERR      ELSE TO OPNERR
      B    @START       THEN BACK TO START
RDF1
      MOVB @READF,R1    MOVE READ OPCODE INTO LEFT BYTE R1
      LI   R0,PAB1      PAB ADDRESS IN VDP
      BLWP @VSBW        WRITE ONE BYTE INTO PAB
      AI   R0,9         ADD NINE
      MOV  R0,@PABPNT   MOVE TO >8356
      CLR  @STATUS      CLEAR GPL STATUS
      BLWP @DSRLNK      USE DSRLNK
      DATA 8            REQUIRED DATA
      LI   R0,PAB1+1    SET TO SECOND BYTE OF PAB IN VDP
      BLWP @VSBR        READ INTO LEFT BYTE R1
      SRL  R1,13        SHIFT R1 RIGHT BY 13 BITS
      JEQ  READON       IF ZERO, NO ERROR IN DSR OPERATION
      CI   R1,5         IF ERROR = 5, END OF FILE HAS BEEN REACHED
      JEQ  CLSF1        IF SO, CLOSE THE FILE
      BL   @FILERR      ELSE SOME OTHER ERROR, REPORT THAT TO USER
      B    @START       BACK TO START
READON LI   R0,PAB1+5   POINT AT PAB+5 IN VDP RAM
      BLWP @VSBR        READ THAT BYTE INTO LEFT BYTE R1
      MOVB R1,R2        MOVE BYTE INTO R2
      SRL  R2,8         RIGHT JUSTIFY LENGTH IN R2
      MOVB R1,@TEMSTR   MOVE BYTE TO TEMSTR
      LI   R0,BUF       POINT TO BUFFER LOCATION IN VDP
      LI   R1,TEMSTR+1  CONTENT GOES TO TEMSTR+1
      BLWP @VMBR        READ CONTENT OF RECORD FROM VDP BUFFER
      MOV  R2,R8        STASH STRING LENGTH IN R8
      INC  R8           INC TO INCLUDE LENGTH BYTE
      LI   R10,>A000    POINT AT START OF ARRAY MEMORY
CMPSTR
      LI   R9,TEMSTR    POINT AT INCOMING STRING
      MOV  R10,R14      SAVE R10 ADDRESS IN R14
      C    R10,@ENDSTR  COMPARE R10 TO END OF ARRAY
      JLT  CMPON        IF LESS, PROCEED WITH COMPARISON
      JMP  NOSORT       ELSE NO SORT NECESSARY
CMPON  MOVB *R9+,R4     GET INCOMING STRING LENGTH IN R4
      MOVB *R10+,R5     GET AN ARRAY STRING'S LENGTH
      SRL  R4,8         RIGHT JUSTIFY R4
      SRL  R5,8         AND R5
CMP910 CB   *R9+,*R10+  COMPARE ONE BYTE
      JGT  BIG          IF R9'S IS GREATER, JUMP
      JLT  SMALL        IF R9'S IS LESS, JUMP
      DEC  R4           ELSE DECREMENT COUNT
      JEQ  SMALL        IF ZERO, R9'S STRING IS LESS
      CI   R5,1         COMPARE R5 TO 1
      JNE  DEC5         IF NOT EQUAL, JUMP
      JMP  BIG          ELSE R9'S STRING
DEC5   DEC  R5          DECREMENT OTHER COUNT
      JNE  CMP910       IF NOT ZERO, COMPARE ANOTHER
BIG    MOV  R14,R10     GET ORIGINAL ADDRESS BACK IN R10
      MOVB *R10+,R7     TAKE LENGTH BYTE INTO R7
      SRL  R7,8         RIGHT JUSTIFY
      A    R7,R10       ADD LENGTH TO R10
      JMP  CMPSTR       THEN COMPARE TO NEXT STRING
SMALL
      MOV  @ENDSTR,R10  POINT AT END OF ARRAY
      MOV  R10,R9       POINT R9 AT PRESENT END
      MOV  R10,R4       MOVE PRESENT END TO R4
      S    R14,R4       SUBTRACT START OF HIGH STRING
      A    R8,R10       ADD LENGTH OF STRING TO BE ADDED
      CI   R10,>FFE7    ARE WE AT END OF MEMORY
      JLT  DEC9         IF NOT, PROCEED
      LI   R0,22*SCRWID+2 ELSE SET FOR ROW 23, COL 3
      LI   R1,OOMSTR    "OUT OF MEMORY"
      BL   @DISSTR      DISPLAY THAT
      BL   @KEYLOO      WAIT FOR KEYSTROKE
      JMP  GETOFN       THEN MOVE ON
DEC9   DEC  R9          DECREMENT R9
      DEC  R10          AND R10
MOVREV MOVB *R9,*R10    MOVE ONE BYTE
```

THE ART OF ASSEMBLY—

(Continued from Page 11)

so you can sort a series of files without exiting the program, if you wish.

EMBELLISHMENTS

Since we've provided all the source code in today's sidebar, you can add some "touches" of your own to this rather primitive program. For example, just before the BL @CRSIN lines, you could put in a "beep" sound via GPLLNK, and you could add a "boop" sound in the error traps just before the BL @KEYLOO lines. You could also expand the input file name fields to allow for hard disk path names.

PROGRAM CONSTRUCTION

Those who've followed this column will see a lot of familiar stuff in the sidebar. There's CRSIN from number 5, for example. That was lifted "as is" for use here. The error trapping for file operations came right out of number 8, and other subroutines were "imported" from other sources. The opening, reading, writing and closing of files came from number 9. We mention all this just to show that this is something you too can do, taking old subroutines from previous programs or from our "Sidebars" and re-cycling them to make a new program. Thus the bulk of this "new" program is re-cycled subroutines, not newly written source code. Doing it this way allowed this whole program to be put together and tested in a single afternoon. This column to accompany the source code took only a couple of hours the next morning. (Hope John and Laura don't notice this last.)

We keep a "battery" of such routines here on dusty old floppy disks, and in some cases in multiple versions, so that routines for linkage from Extended BASIC, which are slightly different, coexist with those written for E/A use. If we had time, we'd make some kind of index of all those floppy disks, so a particular routine would be easier to find, but it seems we can always find time to search through all the disks, but never find the time to index them. It reminds me of another expression from my days in Civil Service, concerning "rush" projects: "We never have enough time to do it right, but we always have time to do it over."

We hope you'll find today's program useful. It's written rather crudely, but is completely functional. You can make changes in it to your heart's content, for different kinds of files, and so on. You could also adapt it to work from Extended BASIC, or make it an "Option-5" program file. Next month we promise we'll write about something other than sorts. Two columns in succession on that topic is enough.

```
       DEC  R9           DECRMENT POINTER
       DEC  R10          AND OTHER POINTER
       DEC  R4           DECREMENT BYTE COUNT
       JNE  MOVREV       IF NOT ZERO, REPEAT
       JMP  MOVIN        ELSE JUMP AHEAD
NOSORT MOV  @ENDSTR,R14  MOVE END OF ARRAY ADDRESS INTO R14
MOVIN  LI   R9,TEMSTR    POINT AT INCOMING STRING
       MOV  R14,R10      DESTINATION ADDRESS IN R10
       BL   @MOVSTR      MOVE STRING INTO ARRAY
       A    R8,@ENDSTR   ADD LENGTH TO END ADDRESS
       MOV  @ENDSTR,R4   MOVE THAT TO R4
       MOVB @ONE,*R4     PUT A ZERO BYTE THERE
       B    @RDF1        THEN READ NEXT RECORD
*
CLSF1  BL   @CLSF2       USE SUBROUTINE
*
GETOFN
       LI   R0,3         SET TO ROW 1, COL 3
       LI   R4,3*SCRWID  THREE ROWS
       BL   @CLRFLD      CLEAR
       LI   R1,OUTSTR    "OUTPUT NAME" PROMPT
       BL   @DISSTR      DISPLAY
       AI   R0,SCRWID    DOWN ONE ROW
       LI   R4,15        15 BYTES
       BL   @CRSIN       GET NAME
       CI   R8,15        F-9 STRUCK?
       JNE  GETOF1       IF NOT, MOVE ON
       B    @START       ELSE BACK TO START
GETOF1 LI   R9,TEMSTR    POINT AT TEMPORARY STRING
       LI   R10,PAB1DT+9 AND FILE NAME LOCATION
       BL   @MOVSTR      MOVE THE STRING INTO PLACE
*
OPNF2
       MOVB @OUTMD,@PAB1DT+1 OPEN WILL BE OUTPUT MODE
       LI   R0,PAB1      SET WRITE ADDRESS IN R0
       MOVB @PAB1DT+9,R2 GET DESCRIPTOR LENGTH BYTE INTO LEFT BYTE R2
       SRL  R2,8         RIGHT JUSTIFY SO R2 IS A WORD OF LENGTH
       AI   R2,10        ADD 10 TO INCLUDE THE PAB1DT LINE PLUS DESCRIPTOR
       LI   R1,PAB1DT    POINT R1 AT PAB DATA
       BLWP @VMBW        WRITE BYTES TO PAB LOCATION IN VDP RAM
       AI   R0,9         ADD NINE TO ADDRESS IN R0
       MOV  R0,@PABPNT   PLACE THAT ADDRESS AT >8356
       CLR  @STATUS      CLEAR GPL STATUS
       BLWP @DSRLNK      USE DSRLNK UTILITY
       DATA 8            REQUIRED DATA
       STST R14          STORE STATUS REGISTER IN R14
       ANDI R14,>2000    MASK ALL BUT BIT #2 IN R14
       JEQ  WRTF2        IF ZERO, GO AHEAD TO WRITE FILE
       BL   @OPNERR      ELSE TO OPNERR
       JMP  GETOFN       THEN JUMP BACK
WRTF2
       LI   R9,>A000     POINT AT START OF ARRAY
WRTNXT LI   R10,TEMSTR   POINT R10 AT TEMSTR
       BL   @MOVSTR      MOVE THE STRING
       MOVB @TEMSTR,R1   GET LENGTH OF RECORD IN LEFT BYTE R1
       JEQ  CLSFO        IF ZERO LENGTH, WE ARE AT END OF ARRAY
       LI   R0,PAB1+5    POINT TO RECORD LENGTH BYTE OF PAB
       BLWP @VSBW        WRITE LENGTH TO PAB
       MOVB R1,R2        PLACE LENGTH IN LEFT BYTE R2
       SRL  R2,8         RIGHT JUSTIFY LENGTH IN R2
       LI   R1,TEMSTR+1  POINT TO STRING CONTENT
       LI   R0,BUF       POINT AT BUFFER IN VDP
       BLWP @VMBW        WRITE RECORD CONTENTS TO VDP
       MOVB @WRITEF,R1   GET WRITE OPCODE IN R1
       LI   R0,PAB1      POINT TO START OF PAB
       BLWP @VSBW        WRITE THE OPCODE BYTE TO VDP
       AI   R0,9         ADD 9
       MOV  R0,@PABPNT   MOVE TO >8356

       CLR  @STATUS      CLEAR GPL STATUS BYTE
       BLWP @DSRLNK      CALL DSR LINKAGE
       DATA 8            REQUIRED DATA
       LI   R0,PAB1+1    POINT TO SECOND BYTE OF PAB
       BLWP @VSBR        READ THAT BYTE INTO R1
       SRL  R1,13        SHIFT R1 RIGHT 13 BITS
       JEQ  WRTNXT       IF ZERO, NO ERROR, SO GO ON
       BL   @FILERR      ELSE BRANCH TO ERROR HANDLING
       BL   @CLSF2       CLOSE FILE
       B    @GETOFN      THEN BRANCH
CLSFO  BL   @CLSF2       CLOSE THE FILE
       B    @START       THEN BACK TO START
EXIT   LWPI GPLWS        LOAD THE GPL WORKSPACE
       B    @>6A         THEN BACK TO E/A
*
* CODE SECTION - SUBROUTINES
*
OPNERR
```

GEnie cuts connect charges in half

Subscribers to GEnie, the General Electric telecommunications service, will see a new rate structure starting in July.

New rates will lower the hourly connect fee from the current $6 during off-peak hours to $3 ($4 in Canada). The monthly connect charge will increase from $4.95 ($5.95 Canada) to $8.95 ($10.95 Canada). Included in the monthly connect charge will be four hours of free usage, according to GEnie.

GEnie is eliminating its so-called basic service menu and allowing members to access virtually all of its services for the $3 per hour charge.

THE ART OF ASSEMBLY—

```
      MOV  R11,*R15+    STACK R11
      LI   R0,22*32+2   ROW 23, COL 3
      LI   R1,FNOMSG    FILE NOT OPENED
      BL   @DISSTR      DISPLAY THAT
      LI   R0,PAB1+1    POINT AT PAB + 1 IN VDP
      BLWP @VSBR        READ THAT BYTE
      SRL  R1,13        SHIFT RIGHT 13 BYTES
      JMP  FILER1       THEN JUMP
FILERR MOV  R11,*R15+    STACK R11
FILER1 SLA  R1,1          DOUBLE NUMBER IN R1
      AI   R1,LUT       ADD LOOKUP TABLE ADDRESS
      MOV  *R1,R1       GET ERROR MESSAGE ADDRESS INTO R1
      LI   R0,23*32+2   POINT AT ROW 24, COL 3
      BL   @DISSTR      DISPLAY ERROR MESSAGE
      BL   @KEYLOO      STOP AT KEY LOOP
      LI   R0,22*SCRWID LOAD R0 FOR ROW 23
      LI   R4,2*SCRWID  TWO ROWS
      BL   @CLRFLD      CLEAR THOSE
      B    @SUBRET      THEN RETURN
*
DISSTR  MOVB *R1+,R2     GET LENGTH BYTE
       SRL  R2,8        RIGHT JUSTIFY
       JEQ  DISX        IF ZERO LENGTH, SKIP IT
       BLWP @VMBW       ELSE WRITE STRING TO SCREEN
DISX    RT              RETURN TO CALLER
*
CLSF2  LI   R0,PAB1      POINT TO PAB ADDRESS
      MOVB @CLOSEF,R1   GET CLOSE OPCODE IN LEFT BYTE R1
      BLWP @VSBW        WRITE OPCODE TO PAB
      AI   R0,9         ADD NINE
      MOV  R0,@PABPNT   PLACE AT >8356
      CLR  @STATUS      CLEAR STATUS
      BLWP @DSRLNK      CALL DSRLNK
      DATA 8            REQUIRED DATA
      RT                RETURN
*
*
CRSIN
      MOV  R11,*R15+    STACK RETURN ADDRESS
      CLR  @INSFLG      CLEAR OUR INSERT FLAG
      MOV  R0,@PGNUM    STASH R0 IN MEMORY LOCATION
      DEC  R0           DECREMENT R0
      MOVB @EDGE,R1     PLACE EDGE CHARACTER IN LEFT BYTE R1
      BLWP @VSBW        WRITE EDGE CHARACTER TO SCREEN
      INC  R0           RESET R0 TO ORIGINAL VALUE
      A    R4,R0        ADD NUMBER OF CHARACTERS TO ACCEPT
      BLWP @VSBW        WRITE AN EDGE CHARACTER TO SPOT BEYOND FIELD
      MOV  R0,@ENDOC    SAVE THIS LOCATION IN MEMORY
      S    R4,R0        RESET R0 TO ORIGINAL VALUE
      MOV  R4,@SAV4     STASH R4 IN MEMORY
CRSI0A BLWP @VSBR       READ THE CHARACTER POINTED TO BY R0
      MOVB R1,@ALTKEY   STASH THAT CHARACTER AT LOCATION ALTKEY
CRSI0  BL   @CURFRC     FORCE THE CURSOR ONTO THE SCREEN
      BL   @KI2         USE THE SCANNING SUBROUTINE WITH FLASHING CURSOR
      CI   R8,9         HAS RIGHT ARROW BEEN STRUCK?
      JEQ  CRSRT        IF SO, JUMP
      CI   R8,8         HAS LEFT ARROW BEEN STRUCK?
      JEQ  CRSBK        IF SO, JUMP
      CI   R8,10        DOWN ARROW?
      JLT  CRSC4        IF LESS, JUMP
      CI   R8,15        HAS FUNCTION-9 BEEN STRUCK?
      JEQ  CRSDMY       IF SO, JUMP
      CI   R8,13        HAS ENTER KEY BEEN STRUCK?
      JLT  CRSDMY       IF LESS, JUMP
CRSC4  CI   R8,4        HAS FUNCTION-2 (INSERT) BEEN STRUCK?
      JNE  CRSENT       IF NOT, JUMP
      INC  @INSFLG      ELSE SET INSERT FLAG
      JMP  CRSI0        THEN JUMP BACK
CRSENT CB   @KEYVAL,@ENTERV HAS ENTER BEEN STRUCK?
      JEQ  CRSDMY       IF SO, JUMP
      CI   R8,3         HAS FUNCTION-1 (DELETE) BEEN STRUCK?
      JEQ  CRSDEL       IF SO, JUMP
      CI   R8,32        SPACE BAR
      JLT  CRSI0        IF LESS, JUMP
* THE FOLLOWING FIVE LINES ARE NEEDED ONLY IF ONE WANTS LOWER CASE
* CHARACTERS CONVERTED TO UPPER CASE.  IF NOT, OMIT THESE FIVE LINES
      CI   R8,122       COMPARE TO LOWER CASE Z
      JGT  CRSI0        IF GREATER, JUMP
      CI   R8,97        COMPARE TO LOWER CASE A
      JLT  CRSI1        IF LOWER, JUMP
      SB   @ANYKEY,@KEYVAL ELSE SUBTRACT >20 FROM KEYSTROKE
CRSI1
      MOV  @INSFLG,R1   TEST IF INSERT FLAG ON
      JEQ  CRSI1A       IF NOT, JUMP
      MOVB @ALTKEY,R1   ELSE WRITE CURRENT CHARACTER
      BLWP @VSBW        TO CURRENT SCREEN POSITION
      MOV  @ENDOC,R2    MOVE LIMIT ADDRESS INTO R2
      S    R0,R2        SUBTRACT CURRENT R0 POSITION
      LI   R1,TEMSTR    POINT TO TEMSTR LOCATION
      BLWP @VMBR        READ CHARACTERS FROM SCREEN
      DEC  R2           DECREMENT CHARACTER COUNT
      JEQ  CRSI1A       IF R2 IS ZERO, NO INSERT - WE'RE AT LAST POSITION
      INC  R0           INCREMENT SCREEN POSITION
      BLWP @VMBW        WRITE CHARACTERS BACK
      DEC  R0           POINT BACK ONE SPOT
CRSI1A MOVB @KEYVAL,R1   MOVE THE KEY STRUCK INTO LEFT BYTE R1
      BLWP @VSBW        WRITE KEY VALUE TO SCREEN
      INC  R0           POINT AT NEXT CHARACTER POSITION
      BLWP @VSBR        READ CHARACTER THAT'S THERE
      CB   R1,@EDGE     IS THIS AN EDGE CHARACTER?
      JNE  CRSI0A       IF NOT, JUMP
      DEC  R0           ELSE BACK UP ONE CHARACTER
      JMP  CRSI0A       THEN BACK FOR ANOTHER KEY INPUT
CRSRT  MOVB @ALTKEY,R1   TAKE CURRENT SCREEN CHARACTER INTO LEFT BYTE R1
      BLWP @VSBW        WRITE CHARACTER TO SCREEN
      CLR  @INSFLG      CLEAR THE INSERT FLAG
      INC  R0           MOVE TO NEXT SPOT
      BLWP @VSBR        READ THE CHARACTER THERE
      CB   R1,@EDGE     IS THAT EDGE CHARACTER?
      JEQ  CRSRT1       IF SO, JUMP
      MOVB R1,@ALTKEY   ELSE STASH CURRENT SCREEN CHARACTER
      BL   @CURFRC      FORCE CURSOR ONTO SCREEN
      BL   @KI2A        GO SCAN KEYBOARD
      CB   @KEYVAL,@RITEV IS RIGHT ARROW STILL HELD DOWN?
      JEQ  CRSRT        IF SO, KEEP GOING RIGHT
      CB   @KEYVAL,@NOKEY HAS NO KEY BEEN STRUCK?
      JEQ  CRSRT2       IF SO, JUMP
CRSRT1 DEC  R0          BACK TO PREVIOUS SPOT
CRSRT2 MOVB @ONOFF,@KI2A+2 RESTORE DELAY CONSTANT
      MOVB @ALTKEY,R1   GET CHARACTER INTO LEFT BYTE R1
      BLWP @VSBW        WRITE TO SCREEN
      JMP  CRSI0        THEN JUMP BACK FOR ANOTHER KEY
CRSBK  MOVB @ALTKEY,R1   GET CURRENT CHARACTER IN R1
      BLWP @VSBW        WRITE TO SCREEN
      CLR  @INSFLG      CLEAR INSERT FLAG
      DEC  R0           BACK ONE SPOT
      BLWP @VSBR        READ CHARACTER FROM SCREEN
      CB   R1,@EDGE     IS THAT EDGE CHARACTER?
      JEQ  CRSBK1       IF SO, JUMP
      MOVB R1,@ALTKEY   ELSE STASH CHARACTER AT ALTKEY
      BL   @CURFRC      FORCE CURSOR ONTO SCREEN
      BL   @KI2A        GO GET KEYSTROKE
      CB   @KEYVAL,@LEFTV  IS LEFT ARROW STILL HELD DOWN?
      JEQ  CRSBK        IF SO, GO BACK AGAIN
      CB   @KEYVAL,@NOKEY HAS NO KEY BEEN STRUCK
      JEQ  CRSRT2       IF SO, JUMP
CRSBK1 INC  R0          MOVE TO NEXT SPOT
      JMP  CRSRT2       THEN JUMP
CRSDMY JMP  CRSIX       THIS IS A DUMMY JUMP TO KEEP JUMPS IN RANGE
CRSDEL MOV  R0,R7       STASH R0 IN R7
      CLR  @INSFLG      CLEAR INSERT FLAG, SINCE WE'RE DELETING
      MOV  @ENDOC,R2    END OF FIELD ADDRESS IN R2
      S    R0,R2        SUBTRACT CURRENT CHARACTER ADDRESS
      INC  R0           POINT TO NEXT CHARACTER
      DEC  R2           DECREMENT R2 COUNT
      JEQ  CRSD1        IF R2 ZERO, PRINT SPACE - WERE AT LAST POSITION
      LI   R1,TEMSTR    POINT R1 AT TEMSTR FOR TEMPORARY STORAGE
      BLWP @VMBR        READ CHARACTERS INTO LOCATION TEMSTR
      MOV  R7,R0        PUT BACK R0
      BLWP @VMBW        WRITE CHARACTERS FROM TEMSTR TO SCREEN
CRSD1  MOVB @ANYKEY,R1   PUT A SPACE IN LEFT BYTE R1
      MOV  @ENDOC,R0    GET LIMIT SPOT INTO R0
      DEC  R0           DECREMENT BY ONE
      BLWP @VSBW        WRITE A SPACE TO SPOT JUST BEFORE LIMIT
      MOV  R7,R0        GET R0 BACK AGAIN
CRSD0  B    @CRSI0A      BRANCH BACK TO BEGINNING
CRSIX  MOVB @ALTKEY,R1   MOVE CURRENT CHARACTER TO R1
      BLWP @VSBW        WRITE TO SCREEN
      MOV  @ENDOC,R0    SET LIMIT POSITION IN R0
      DEC  R0           DECREMENT BY ONE
      MOV  @SAV4,R2     MOVE MAX NUMBER OF CHARACTERS INTO R2
CRSIX1 BLWP @VSBR       READ THE CHARACTER AT CURRENT R0 POSITION
      CB   R1,@ANYKEY   IS THAT A SPACE?
      JNE  CRSIXX       IF NOT, WE'VE REACHED CONTENT OF STRING
      DEC  R0           ELSE MOVE BACK ONE SPOT
      DEC  R2           DECREASE CHARACTER COUNT BY ONE
      JGT  CRSIX1       IF GREATER THAN ZERO, JUMP BACK
CRSIXX MOV  @PGNUM,R0    GET ORIGINAL R0 POSITION BACK
      SWPB R2           PUT CHARACTER COUNT IN LEFT BYTE R2
      MOVB R2,@TEMSTR   PLACE THAT AT TEMSTR
      SWPB R2           REVERSE R2 AGAIN
      JEQ  CRIX         IF R2=0, JUMP
      LI   R1,TEMSTR+1  ELSE SET R1 TO POINT TO STRING CONTENT STORAGE
CRSIX2 BLWP @VMBR       READ THE STRING FROM THE SCREEN
```

THE ART OF ASSEMBLY—

```
CRIX
SUBRET DECT R15          DECREMENT STACK POINTER BY TWO
       MOV  *R15,R11     GET RETURN ADDRESS BACK
       RT                RETURN
*
*
KI2    CLR  @STATUS      KEY-IN WITH ALTERNATING
       BLWP @KSCAN       CHARACTER AND CURSOR
       LIMI 2            ACTIVATE INTERRUPTS
       LIMI 0            SHUT OFF INTERRUPTS
       DEC  R4           ENTER AFTER R4 SET TO >0200
       JEQ  CHNG         AND R1 TO >1E00 AND VSBW
       CB   @ANYKEY,@STATUS HAS A KEY BEEN STRUCK?
       JNE  KI2          IF NOT, RE-SCAN KEYBOARD
       MOV  @KEYADR,R8   ELSE PUT KEY'S VALUE IN R8
       RT                THEN RETURN
CHNG   CI   R1,>1E00     IS R1 SET TO CURSOR CHARACTER?
       JEQ  L1           IF SO, JUMP
       LI   R1,>1E00     ELSE SET LEFT BYTE R1 TO CURSOR
       BLWP @VSBW        WRITE CURSOR TO SCREEN
       MOVB @ONOFF,R4    PLACE TIMING IN LEFT BYTE R4
       JMP  KI2          GO BACK TO SCANNING KEYBOARD
L1     MOVB @ALTKEY,R1   PLACE ALTERNATING CHARACTER IN LEFT BYTE R1
       MOVB @ONOFF+1,R4  PLACE ALTERNATE DELAY IN LEFT BYTE R4
       BLWP @VSBW        WRITE CHARACTER TO SCREEN
       JMP  KI2          GO BACK TO SCANNING KEYBOARD
*
* THE FOLLOWING IS A SPECIAL KEY INPUT FOR REPEATING OPERATION OF
* THE RIGHT AND LEFT ARROW KEYS
* THIS SUBROUTINE INCLUDES SELF-MODIFYING CODE
*
KI2A   LI   R5,>0280     LOAD R5 WITH DELAY FACTOR
KI2B   CLR  @STATUS      CLEAR GPL STATUS
       BLWP @KSCAN       SCAN KEYBOARD
       CB   @KEYVAL,@NOKEY  HAS NO KEY BEEN STRUCK?
       JEQ  KI2C         IF SO, JUMP
       LIMI 2            SET INTERRUPTS ON
       LIMI 0            SET INTERRUPTS OFF
       DEC  R5           DECREMENT DELAY COUNTER
       JNE  KI2B         IF NOT ZERO, SCAN AGAIN
       MOVB @ONE,@KI2A+2 ELSE MODIFY DELAY COUNT
KI2C   RT                THEN RETURN
*
* THE FOLLOWING SUBROUTINE FORCES THE CURSOR CHARACTER ONTO THE SCREEN
*
CURFRC LI   R1,>1E00     PUT CURSOR CHARACTER IN LEFT BYTE R1
       LI   R4,>0100     SET DELAY FACTOR IN R4
       BLWP @VSBW        WRITE CURSOR TO SCREEN
       RT                RETURN
*
* FOLLOWING SUBROUTINE CLEARS AN INPUT FIELD
* BEGINNING AT R0 POSITION, EXTENDING NUMBER OF CHARACTERS IN R4
*
CLRFLD
       MOV  R4,R2        PLACE VALUE OF R4 IN R2
       MOV  R0,R3        SAVE R0
       MOVB @ANYKEY,R1   PUT SPACE CHARACTER IN LEFT BYTE OF R1
CLRFL1 BLWP @VSBW        WRITE ONE SPACE IN FIELD
       INC  R0           POINT TO NEXT CHARACTER SPOT
       DEC  R2           DECREMENT COUNT OF SPACES
       JNE  CLRFL1       IF NOT ZERO, REPEAT WRITING OPERATION
       MOV  R3,R0        REPLACE ORIGINAL VALUE OF R0
       RT                RETURN
KEYLOO CLR  @STATUS      CLEAR GPL STATUS BYTE
       BLWP @KSCAN       SCAN KEYBOARD
       LIMI 2            ALLOW INTERRUPTS
       LIMI 0            THEN SHUT THEM OFF
       CB   @ANYKEY,@STATUS HAS KEY BEEN STRUCK?
       JNE  KEYLOO       IF NOT, SCAN AGAIN
       RT                ELSE RETURN
MOVSTR
       MOVB *R9,R4       MOVE LENGTH BYTE TO R4
       SRL  R4,8         SHIFT RIGHT
       INC  R4           INCREMENT TO INCLUDE LENGTH BYTE
MOVBTS MOVB *R9+,*R10+   MOVE ONE BYTE, INC POINTERS
       DEC  R4           DECREMENT COUNT
       JNE  MOVBTS       IF NOT ZERO, REPEAT
       RT                RETURN
*
* REQUIRED DATA SECTION
* THE FOLLOWING DATA SOURCE LINES ARE REQUIRED BY THESE SUBROUTINES
*
ONE    DATA 1            ONE AS A WORD
ENDOC  DATA 0            END OF INPUT FIELD
INSFLG DATA 0            INSERT ACTIVE FLAG
PGNUM  DATA 0            STORAGE FOR ONE WORD
SAV4   DATA 0            STORAGE FOR ANOTHER WORD
ONOFF  DATA >0201        ON-OFF BYTES FOR CURSOR
ENDSTR DATA 0            ADDRESS OF END OF STRING ARRAY
* THIS PAB DATA AND MODE BYTES APPLY TO D/V 80 FILES
PAB1DT DATA >0014,BUF,>5000,>0000,>000F
       BSS  15
BADDEV BYTE 15
       TEXT 'BAD DEVICE NAME'
WRPROT BYTE 15
       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'
       EVEN
LUT    DATA BADDEV,WRPROT,BADATT
       DATA ILLOP,OUTSP,ENDFIL
       DATA DEVERR,FILBAD
FNOMSG BYTE 17
       TEXT 'FILE DID NOT OPEN'
EDGE   BYTE >1F
INMD   BYTE >14          BYTE FOR INPUT OF DISPLAY/VARIABLE FILE
OUTMD  BYTE >12          BYTE FOR OUTPUT OF DISPLAY/VARIABLE FILE
APPMD  BYTE >16          BYTE FOR APPEND OF DISPLAY/VARIABLE FILE
UPDAMD BYTE >10          BYTE FOR UPDATE MODE OF D/V FILE -NOT RECOMMENDED
WRITEF BYTE 3            OPCODE FOR WRITE OPERATION
READF  BYTE 2            OPCODE FOR READ OPERATION
CLOSEF BYTE 1            OPCODE FOR CLOSE OPERATION
ANYKEY BYTE >20          KEYSTROKE COMPARE BYTE
NOKEY  BYTE >FF          NO KEY STRUCK
ALTKEY BYTE 0            STORAGE FOR SCREEN CHARACTER
ENTERV BYTE 13           "ENTER" KEY CODE
RITEV  BYTE 9            RIGHT ARROW
LEFTV  BYTE 8            LEFT ARROW
INFSTR BYTE 15
       TEXT 'INPUT FILE NAME'
OUTSTR BYTE 16
       TEXT 'OUTPUT FILE NAME'
SFSTR  BYTE 23
       TEXT 'SORTING FILE - STAND BY'
OOMSTR BYTE 19
       TEXT 'OUT OF MEMORY SPACE'
TEMSTR BSS  81           TEMPORARY STORAGE LOCATION FOR RECORD
       EVEN
RTNSTK DATA 0            RETURN ADDRESS STACK
       END
```

Ottawa group gets new address

The Ottawa TI Users' Group has changed its address to 222 Guigues Ave., Apt. 603, Ottawa, Ontario, Canada K1N 5J2, according to Bill Gard, president and treasurer.

Gard is moving to Halifax, Nova Scotia, but says he will still have access to the Ottawa club through its BBS, (613) 738-0617.

MS Express offers free catalog

MS Express Software is offering copies of its latest TI99/4A software product catalog free to anyone who requests one.

Write MS Express Software, P.O. Box 498, Richmond, OH 43944.

Tacoma fair slated

The Tacoma 99ers plan to hold a TI fair sometime in September, according to group spokesman Jim Tompkins. For more information, contact Tompkins at (206) 756-0934.

PagePro formatter

XBASIC program produces 2-column text for PagePro 99

By BILL GASKILL

PAGEFORM is an Extended BASIC program designed to produce two-column text for use by PagePro 99. It allows up to 114 lines of text to be entered into a single file, which is the number of lines that will produce a single page of two-column text for a PagePro 99 page. Each line may contain up to 28 characters maximum length.

To produce a newsletter, you may type in the text for each page free-hand, or it may be imported from a TI-Writer file. PAGEFORM loads and saves all documents in D/V80 format, just as the TI-Writer program does.

When all text for each page has been entered into a file it may be printed with ragged right margins, or you may "run it through" the justification process prior to printing to enhance the appearance of the Page Pro page.

Justification is selective in that it allows you to select only those lines that you want to justify. For example, one would not normally justify the last line in a paragraph. Lines that have only one word in them, or blank lines are ignored by the justification routine.

Pages may be numbered at the bottom of the page during the printing process. You will be prompted for the number to print and it will appear in the middle of the page.

PROGRAM OPERATION

PAGEFORM 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. The 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 "C" to Clear the screen.

PAGEFORM provides six text screens that are capable of holding the 114 text lines available in a 1 page file. Paging from one screen to another is done by the program, based upon cursor position, and is thus automatic.

Cursor movement is done via the arrow keys and the Enter key. FCTN-E takes the cursor back one line at a time and Enter advances the cursor one line at a time. FCTN-S and FCTN-D move the cursor horizontally within a text line.

COMMAND MENU OPTIONS

• Clear the current screen

(See Page 16)

PAGEFORM—

(Continued from Page 15)

- Delete a line of text
- End of file access
- File name and path display
- Justification of text
- Insert a blank line
- Load a file
- Output/input path change
- Pages prints file in Page Pro format
- Quit the program
- Save a file in TIW format
- Top of file access

Options available from the command mode that are not listed on the menu include the ability to toggle the over write protect mode on/off (F4), the ability to purge the current file (F8) and the the ability to catalog a disk in the active drive (FCTN-E/X).

COMMAND MENU OPTIONS

• **Clear** allows the current screen to be erased in a single key press. A cleared screen may be restored by selecting the Top command, or by pressing the number 1 key to display screen one.

• **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. Then press D to Delete and the line will be removed.

• **End** displays screen six and places the cursor on line 114, in the text mode.

• **File** displays the path/file name of the last file that was either loaded or saved. If neither operation has been performed, such as when you are working with a new file, the display will be blank.

• **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.

• **Justify** allows you to right justify selectively. When J is pressed from the command mode the file is placed at screen 1 and the first line of text to be justified is displayed at the bottom of the screen.

You are prompted to press FCTN-X to skip the line if desired, Enter to allow it to be justified or FCTN-9 to abort and return to the text entry mode. Blank lines, lines that contain only one word, and lines that are already justified are ignored by the routine.

• **Load** retrieves a document from disk and then displays it on screen. FCTN-E/X may be used to catalog the disk that you are retrieving a document from. See *Special Features* below for more detail.

• **Output** provides pathing options for loading and saving of files. When O (not zero) is pressed you may specify DSK2, 3 or even WDS1 as the path where your files will be read from or written to.

• **Pages** first prompts you for the name of the PagePro file that will be printed to disk in two-column format. The default is DSK1.PAGE1. Type over it to suit your needs. A prompt for the page number then appears and finally the number of tabs spaces from the left margin to begin printing at. FCTN-9 will abort file printing and the space bar will suspend it until the space bar is released.

• **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.

• **Save** writes an existing file to disk for later access. An overwrite protect mode is built into Save that protects existing files from being destroyed. See Special Features for more detail.

• **Top** displays screen 1 and places the cursor on line one regardless of where you are at in the file. It is the compliment to the End command.

SPECIAL FEATURES

A disk cataloging option may be used from the Load prompt by pressing the FCTN-E or FCTN-X keys before the file name to load is entered. This will cause the contents of the disk in the active drive to be displayed at the base of the screen, one file at a time. Pressing any key will advance the cataloging to the next file name. Pressing F9 will abort the catalog routine and return to the Load prompt where the last file listed in the catalog will be displayed, ready to load.

PAGEFORM also provides an overwrite protect feature in the Save command. It will warn you when an existing file is about to be overwritten. The overwrite protect feature may be disabled or enabled with FCTN-4. It is enabled as the default. When it is disabled an asterisk is displayed to the right of the word "Load" on the command line, to let you know that you don't have the protection available.

Should you wish to totally purge the current document, you may press the REDO, FCTN-8, keys while in the command mode to clear out the text buffer. This option will re-run the program, so once it is chosen, there is no way to recover a purged document. Be careful!

PAGE PRO PAGES

Once you have printed your text using the Pages option, load Page Pro. To import the text, first place the cursor in the upper left corner of the PagePro screen. Next, press Ctrl0 F and and then 3 to import the PAGES file. At the prompt simply type in the path and file name where the PAGES file was written to and it will be read into PagePro 99 in two-column format.

PAGEFORM

```
100 ON BREAK NEXT :: CALL CL
EAR :: CALL SCREEN(5):: LN=2
8 :: FOR I=0 TO 14 :: CALL C
OLOR(I,16,5):: NEXT I !061
110 DR$="DSK1." :: PR$="DSK1
.PAGE1" :: CALL CHAR(124,"00
10F88484F81000",126,"00FF"):
: CALL HCHAR(20,1,126,32)!18
1
120 ON WARNING NEXT :: CALL
KEY(5,K,S):: OPTION BASE 1 :
: DIM A$(117):: ON ERROR 150
 !135
130 DISPLAY AT(24,1):"Press
ENTER to begin typing" :: I,
M,R=1 :: GOSUB 720 :: GOSUB
1120 !099
140 CALL KEY(5,K,S):: DISPLA
Y AT(23,24):I :: ACCEPT AT(R
,1)SIZE(-LN):A$(I):: M=I !0
5
150 GOSUB 1140 :: CALL KEY(0
,G,H):: IF G=11 THEN 160 ELS
```

(See Page 17)

PAGEFORM—

(Continued from Page 16)

```
E IF G=10 THEN 320 ELSE 230
!101
160 IF I<=1 THEN 140 ELSE I=
I-1 :: R=R-1 :: G=0 !246
170 IF I=95 THEN RS=77 :: RE
=95 :: RO=1 :: GOSUB 1060 ::
 GOTO 140 !079
180 IF I=76 THEN RS=58 :: RE
=76 :: RO=1 :: GOSUB 1060 ::
 GOTO 140 !076
190 IF I=57 THEN RS=39 :: RE
=57 :: RO=1 :: GOSUB 1060 ::
 GOTO 140 !073
200 IF I=38 THEN RS=20 :: RE
=38 :: RO=1 :: GOSUB 1060 ::
 GOTO 140 !061
210 IF I=19 THEN RS=1 :: RE=
19 :: RO=1 :: GOSUB 1060 ::
GOTO 140 !009
220 GOTO 140 !219
230 IF I=19 THEN RS=20 :: RE
=38 :: R=1 :: GOTO 290 !138
240 IF I=38 THEN RS=39 :: RE
=57 :: R=1 :: GOTO 290 !150
250 IF I=57 THEN RS=58 :: RE
=76 :: R=1 :: GOTO 290 !153
260 IF I=76 THEN RS=77 :: RE
=95 :: R=1 :: GOTO 290 !156
270 IF I=95 THEN RS=96 :: RE
=114 :: R=1 :: GOTO 290 !199
280 GOTO 320 !144
290 GOSUB 1130 :: RO=1 :: FO
R I=RS TO RE :: DISPLAY AT(R
O,1):A$(I):: RO=RO+1 :: NEXT
 I !216
300 IF K=69 THEN I=114 ELSE
I=RS !011
310 GOTO 140 !219
320 IF G=10 THEN GOSUB 830 :
: GOSUB 720 :: GOTO 140 !246
330 IF I=114 THEN 350 !196
340 R=R+1 :: I=I+1 :: GOSUB
1120 :: GOTO 140 !061
350 DISPLAY AT(24,1):"     E
scape Print Save" :: CALL KE
Y(3,S,K)!123
360 IF K=0 THEN 350 ELSE IF
S=69 THEN 710 ELSE IF S=80 T
HEN 370 ELSE IF S=83 THEN 51
0 ELSE 350 !142
370 CALL KEY(3,K,S):: GOSUB
1150 :: DISPLAY AT(24,1):"Pa
gePro:DSK1.PAGE1" :: ACCEPT
AT(24,9)SIZE(-20):PR$ :: IF
PR$="" THEN 720 !058
380 CALL KEY(3,K,S):: GOSUB
1140 :: GOSUB 1150 :: DISPLA
Y AT(24,1):"Page Number:1" :
: ACCEPT AT(24,13)SIZE(-2):P
N$ !212
390 DISPLAY AT(24,1):"Tabs f
rom left margin:1" :: ACCEPT
 AT(24,23)SIZE(-1)VALIDATE(D
IGIT):T :: GOSUB 1140 :: GOS
UB 1150 !069
400 DISPLAY AT(24,1):"Are al
l choices correct? Y/N" :: C
ALL KEY(3,P,Q):: IF Q=0 THEN
 400 ELSE IF P<>89 THEN 370
ELSE GOSUB 1140 !201
410 ON ERROR 880 :: OPEN #1:
PR$,OUTPUT :: DISPLAY AT(24,
1):"Printing line:" !238
420 FOR I=1 TO 57 !116
430 IF LEN(A$(I))<>LN THEN A
$(I)=A$(I)&RPT$(" ",LN-LEN(A
$(I)))!248
440 IF LEN(A$(I+57))<>LN THE
N A$(I+57)=A$(I+57)&RPT$(" "
,LN-LEN(A$(I+57)))!212
450 IF LEN(A$(I))=0 THEN A$(
I)=RPT$(" ",LN)!164
460 IF LEN(A$(I+57))=0 THEN
A$(I+57)=RPT$(" ",LN)!146
470 PRINT #1:TAB(T);A$(I)&"
 "&A$(I+57);!019
480 CALL KEY(0,G,H):: IF G=3
2 THEN 480 ELSE IF G=15 THEN
 500 !232
490 DISPLAY AT(24,15):I :: N
EXT I :: PRINT #1 :: PRINT #
1 :: PRINT #1:TAB(LN+T+1);PN
$ !174
500 CLOSE #1 :: I=M :: GOTO
140 !202
510 CALL KEY(3,K,S):: GOSUB
1150 :: DISPLAY AT(24,1):"Sa
ve:          (Enter=Exit)" :
: ACCEPT AT(24,6)SIZE(10):P$
 :: IF P$="" THEN 720 !145
520 IF OP=1 THEN 600 ELSE OP
EN #4:DR$,INPUT ,RELATIVE,IN
TERNAL :: INPUT #4:E$,E,E,F
!224
530 FOR H=1 TO 127 :: INPUT
#4:E$,D,E,F !182
540 IF P$=E$ THEN CLOSE #4 :
: GOTO 580 !128
550 IF ABS(D)=0 THEN 570 !10
8
560 NEXT H !222
570 CLOSE #4 :: GOTO 600 !19
7
580 GOSUB 1140 :: DISPLAY AT
(23,1):"Existing file. Overw
rite Y/N" :: ACCEPT AT(23,28
)SIZE(-1):YN$ !000
590 IF YN$<>"Y" THEN GOSUB 1
140 :: GOTO 140 !196
600 ON ERROR 860 :: P$=DR$&P
$ :: GOSUB 1140 !076
610 OPEN #3:P$,OUTPUT,DISPLA
Y ,VARIABLE :: FOR I=1 TO 11
4 !135
620 PRINT #3:A$(I):: DISPLAY
 AT(23,25):I :: NEXT I :: CL
OSE #3 :: I=M !073
630 GOSUB 1140 :: GOTO 140 !
037
640 CALL KEY(3,Z,Y):: GOSUB
1150 :: DISPLAY AT(24,1):"Lo
ad:          (Enter=Exit)" :
: DISPLAY AT(24,6)SIZE(-10):
E$ !002
650 ACCEPT AT(24,6)SIZE(-10)
:P$ :: CALL KEY(0,Z,Y):: IF
Z=10 OR Z=11 THEN 890 !224
660 IF P$="" THEN 720 ELSE P
$=DR$&P$ :: GOSUB 1130 !161
670 ON ERROR 870 :: OPEN #2:
P$,INPUT ,DISPLAY ,VARIABLE
!244
680 FOR I=1 TO 114 :: LINPUT
 #2:A$(I)!147
690 I$=STR$(I):: DISPLAY AT(
23,25):I :: NEXT I !175
700 CLOSE #2 :: I=1 :: GOSUB
 1130 !091
710 GOSUB 1130 :: FOR I=1 TO
 19 :: DISPLAY AT(I,1):A$(I)
:: NEXT I :: GOTO 130 !173
720 DISPLAY AT(21,1):"Clr De
l End Fil Ins Jus Load Outpu
t Pages Quit Save Top" :: CA
LL KEY(3,K,S)!086
730 IF S=0 THEN 720 ELSE IF
K=6 THEN 1070 ELSE IF K=67 T
HEN 1330 ELSE IF K=68 THEN 9
70 ELSE IF K=69 THEN 840 ELS
E IF K=70 THEN 1040 !233
740 IF K=14 THEN 1340 ELSE I
F K=84 THEN 850 ELSE IF K=79
 THEN 1050 ELSE IF K=73 THEN
```

(See Page 18)

PAGEFORM—

(Continued from Page 17)

```
 980 ELSE IF K=76 THEN 640 !
057
750 IF K=80 THEN 370 ELSE IF
 K=81 THEN 1030 ELSE IF K=6
THEN 1070 ELSE IF K=83 THEN
510 ELSE IF K=2 THEN 1090 EL
SE IF K=74 THEN 1160 !242
760 IF K=49 THEN RS=1 :: RE=
19 :: R=1 :: GOTO 290 !092
770 IF K=50 THEN RS=20 :: RE
=38 :: R=1 :: GOTO 290 !135
780 IF K=51 THEN RS=39 :: RE
=57 :: R=1 :: GOTO 290 !147
790 IF K=52 THEN RS=58 :: RE
=76 :: R=1 :: GOTO 290 !150
800 IF K=53 THEN RS=77 :: RE
=95 :: R=1 :: GOTO 290 !153
810 IF K=54 THEN RS=96 :: RE
=114 :: R=1 :: GOTO 290 !196
820 RETURN !136
830 DISPLAY AT(24,1):"Press
C,D,F,I,J,L,O,P,Q,S,T" :: RE
TURN !165
840 R=19 :: RS=96 :: RE=114
:: GOTO 290 !142
850 I,R=1 :: RS=1 :: RE=19 :
: GOTO 290 !228
860 GOSUB 1140 :: DISPLAY AT
(23,1):" * Can't Save File!
*" :: GOTO 140 !228
870 GOSUB 1140 :: DISPLAY AT
(23,1):" * File not found! *
" :: GOTO 140 !020
880 GOSUB 1140 :: DISPLAY AT
(23,1):" * Printer error! *"
 :: GOTO 140 !020
890 GOSUB 1130 :: ON ERROR 9
60 :: OPEN #1:DR$,INPUT ,REL
ATIVE,INTERNAL :: INPUT #1:E
$,E,E,F !057
900 FOR H=1 TO 127 :: INPUT
#1:E$,D,E,F !179
910 DISPLAY AT(24,1):"Press
any key...<>";E$ :: CALL KEY
(0,AA,BB):: IF BB=0 THEN 910
 !089
920 GOSUB 1150 :: IF AA=15 T
HEN 950 !122
930 IF ABS(D)=0 THEN 950 !23
3
940 NEXT H !222
950 CLOSE #1 :: GOTO 640 !23
4
960 GOSUB 1140 :: DISPLAY AT
(23,1):" * Device Error! *")
:GOTO 140 !072
970 GOSUB 1010 :: FOR I=I TO
 115 :: A$(I)=A$(I+1):: DISP
LAY AT(23,1):"<Del>" :: NEXT
 I :: I=I-1 :: GOSUB 1130 ::
 GOSUB 990 :: GOTO 140 !147
980 GOSUB 1010 :: FOR I=117
TO I STEP -1 :: A$(I)=A$(I-1
):: DISPLAY AT(23,1):"<Ins>"
 :: NEXT I :: A$(M)=" " :: G
OSUB 1130 :: GOSUB 990 :: GO
TO 140 !210
990 IF RS=0 THEN RS=1 :: RE=
19 !248
1000 RO=1 :: FOR I=RS TO RE
:: DISPLAY AT(RO,1):A$(I)::
RO=RO+1 :: NEXT I :: CALL HC
HAR(R,1,32,1):: I=M :: GOTO
140 !024
1010 IF I=1 AND K=73 THEN 10
20 ELSE CALL HCHAR(R,2,124,1
):: RETURN !127
1020 GOSUB 1140 :: DISPLAY A
T(23,1):" * Can't insert lin
e *" :: GOTO 140 !242
1030 DISPLAY AT(24,1):" Are
you sure? (Y/N)" :: CALL KEY
(3,K,S):: IF K=78 THEN 130 E
LSE IF K<>89 THEN 1030 :: EN
D !CALL PEEK(2,A,B):: CALL L
OAD(-31804,A,B) !046
1040 GOSUB 1140 :: DISPLAY A
T(23,1):"FileName:";P$ :: RE
```

(See Page 19)

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

Lima Multi User Group Conference, May 14-15, Ohio State University Lima Campus, Lima, Ohio. Contact Dave Szippl, 4191 Patterson Haplin, Sidney, OH 45365; phone (513) 498-9713 (evenings).

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

SEPTEMBER

TACOMA 99ERS Fair, contact 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.

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.

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.

PAGEFORM—

(Continued from Page 18)

```
TURN !023
1050 DISPLAY AT(24,1):"Data
Path:";DR$ :: ACCEPT AT(24,1
1)SIZE(-18):DR$ :: IF DR$=""
 THEN 1050 :: GOTO 140 !184
1060 GOSUB 1130 :: FOR I=RS
TO RE :: DISPLAY AT(RO,1):A$
(I):: RO=RO+1 :: NEXT I :: I
=RE :: R=19 :: RETURN !237
1070 DISPLAY AT(24,1):"Purge
? Are you sure? (Y/N)" :: CA
LL KEY(3,Z,Y):: IF Y=0 THEN
1070 ELSE IF Z<>89 THEN 130
!248
1080 GOSUB 1150 :: GOSUB 113
0 :: DISPLAY AT(24,1):"  Ini
tializing memory..." :: RUN
!157
1090 IF OP=0 THEN 1100 ELSE
1110 !071
1100 DISPLAY AT(23,1):"Overw
rite disabled." :: CALL HCHA
R(21,31,42,1):: OP=1 :: GOTO
 140 !074
1110 DISPLAY AT(23,1,"Overwr
ite enabled." :: CALL HCHAR(
21,31,32,1):: OP=0 :: GOTO 1
40 !034
1120 DISPLAY AT(24,1):"Use F
ctn X to exit text mode" ::
RETURN !085
1130 CALL HCHAR(1,1,32,608):
: RETURN !236
1140 CALL HCHAR(23,1,32,32):
: RETURN !231
1150 CALL HCHAR(24,1,32,32):
: RETURN !232
1160 GOSUB 1130 :: CALL HCHA
R(21,1,32,128):: RS=1 :: RE=
19 :: GOSUB 1320 !059
1170 FOR Z=1 TO 114 :: Y$=A$
(Z):: Z$=" " :: CALL HCHAR(2
1,25,32,3):: DISPLAY AT(21,2
5):Z !065
1180 IF Z=20 THEN RS=20 :: R
E=38 :: GOSUB 1320 :: GOTO 1
230 !181
1190 IF Z=39 THEN RS=39 :: R
E=57 :: GOSUB 1320 !039
1200 IF Z=58 THEN RS=58 :: R
E=76 :: GOSUB 1320 !042
1210 IF Z=77 THEN RS=77 :: R
E=95 :: GOSUB 1320 !045
1220 IF Z=96 THEN RS=96 :: R
E=114 :: GOSUB 1320 !088
1230 DISPLAY AT(22,1):Y$ ::
DISPLAY AT(24,1):" Fctn X-sk
ip, ENTER-justify" !013
1240 CALL KEY(0,P,Q):: IF Q=
0 THEN 1240 ELSE IF P=10 THE
N 1310 ELSE IF P=13 THEN 125
0 ELSE IF P=15 THEN 710 !031
1250 P,Q=0 :: X=LEN(Y$):: DI
SPLAY AT(21,2):X :: IF X=LN
OR X=0 THEN 1310 !230
1260 T=POS(Y$,Z$,1):: IF T=0
 THEN 1310 ELSE G=T !058
1270 S=POS(Y$,Z$,T):: C$=SEG
$(Y$,1,S):: D$=SEG$(Y$,S+1,(
LN-S))!034
1280 IF C$="" THEN E$=D$ ::
T=G+1 :: GOTO 1270 !016
1290 E$=C$&Z$&D$ !228
1300 DISPLAY AT(22,1):E$ ::
Y$=E$ :: IF LEN(Y$)=LN THEN
1310 ELSE T=S+G :: GOTO 1270
 !144
1310 A$(Z)=Y$ :: CALL HCHAR(
22,1,32,32):: NEXT Z :: GOTO
 710 !208
1320 GOSUB 1130 :: RO=1 :: F
OR I=RS TO RE :: DISPLAY AT(
RO,1):A$(I):: RO=RO+1 :: NEX
T I :: RETURN !226
1330 GOSUB 1130 :: GOTO 130
!017
1340 DISPLAY AT(24,1):"Enter
 new line length:28" :: ACCE
PT AT(24,23)SIZE(-2):LN :: I
F LN=0 THEN 1340 :: GOTO 140
 !077
```

Asgard shipping First Draft V.2.0

First Draft V.2.0 by Art Gibson has been released by Asgard Software:

The new version now allows program installation with a single disk drive. According to the manufacture, the new version improves disk access speed and offers advanced keyboard buffering. It provides an 80-column page with a standard 99/4A in which the screen window scrolls over as a user types.

The manufacturer says the new version provides the fastest spelling checker available for the 99/4A or Geneve and takes advantage of AMS and AEMS compatible memory cards. According to the manufacturer, with a 512K AMS or AEMS, more than 450K of text can be placed in memory, or more than WordPerfect 5.1 permits on a 640K PC.

First Draft 2.0 allows the user to define up to 11 keyboard macros and load or save them to disk. The new version has been modified and verified to work fully on a Geneve with 1.14F of MDOS and 1.04 of the GPL Interpreter.

"Because we thought this was the case in version 1.0, we are giving *all* registered owners of First Draft 1.0 a free upgrade to 2.0," says Chris Bobbitt of Asgard.

A new manual has been produced from scratch for V.2.0, Bobbitt says. He says other changes include faster search and replace, faster line deletion, paragraph reformatting that can be controlled, enhancements to the Final Copy formatter, and access to all control codes and character graphics in a document.

First Draft 2.0 has a suggested retail price of $39.95, which includes two disks and two manuals. Until Aug. 1, registered users of Spell It! can purchase the program for $29.95 with a photocopy of their program disk. Updates from V.1.0 will be mailed free to registered First Draft owners. Until Sept. 1, First Draft 2.0 can be purchased with the AMS card for $129.95.

To order, send a check or money order, plus $3 shipping and handling (overseas air mail, $7) to Asgard Software, 1423 Flagship Dr., Woodbridge, VA 22192.

Internet and the TI

You don't have to leave your console to travel the globe

By JOHN KOLOEN

Imagine dialing up a modem in your hometown and, in a matter of a few minutes, finding yourself accessing a computer in Holland or New Zealand. Sounds expensive, doesn't it? And it sounds like such globe-trotting would be complicated. Doesn't it?

It is neither expensive nor complicated, but you do have to know a little bit about what you're doing, not to mention having a modem connected to your computer.

So, what kind of BBS system lets you hook up to a computer on the other side of the globe? Certainly not CompuServe, or GEnie, or most of the other commercial online services. (The exception is Delphi, which for a modest monthly charge, allows its members to directly access an Internet gateway.) No, the system that gives you access to the world is called The Internet, or simply Internet.

You've probably heard of Internet. It's a hot topic lately in newspapers and magazines. Articles focusing on the "electronic super highway of the future" frequently mention Internet as today's electronic highway. Although its not really like a highway. It's more like a grid of city streets connecting tens of thousands of computers of all types and descriptions. Trying to describe it as a highway is misleading, since what conjures when describing a highway is a broad, extremely long roadway intersected by thousands of smaller roads. On the Internet there is no interstate, no main highway, just thousands and thousands of computers linked to each other by a patchwork of telephone lines and relays.

Despite the fact that many of the computers on the Internet are mainframes or minis — there are even super computers on the Internet — virtually anyone can become a part of the Internet. Millions of people who are involved with universities or businesses, have easy access to Internet through their organization's computers. But even the hobbyist with a TI99/4A in his study can become a participating member on the Internet by buying access to the network. This can be done through businesses that belong to the Internet and sell this access to the public, through commercial services such as Delphi, or through networks, such as Holonet, that offer access through hundreds of dial-in Bulletin Board Services throughout the country.

I initially got access to Internet through HoloNet, which is operated by Information Access Technologies. I access HoloNet

(See Page 21)

INTERNET AND THE TI—

(Continued from Page 20)

through a node in Austin, Texas and from there can gain access to the Internet via a series of menus.

TELCO IS MOST FLEXIBLE EMULATOR

TI and Geneves can use these services, and I find the best terminal program to use is Telco. That's because Telco supports a number of emulations, including VT100. When I log on to HoloNet, one of the first things I have to do is identify a terminal type. Since VT100 is used by many of the universities and libraries that are accessible on the Internet, I use VT100. Other terminal programs, such as Terminal Emulator II and Mass Transfer, support only ANSI terminals. Although you can still use the Internet using ANSI, it's a little trickier moving cursors up and down menus and selecting options. With VT100 you can use arrow keys to move the cursor. Without it, you'll have to use letter keys that are supported by the host computer you access. Typically, the "J" and "K" keys will move the cursor up and down in ANSI mode.

Beyond the convenience of using Telco, I also recommend using a 2400 baud or faster modem. No one will deny you access because you use a 1200 baud modem, but because of the enormous volume of traffic on many of the computers on the Internet, it's simply more efficient and it lets more people use the system.

HOW EXPENSIVE IS IT?

What are the costs? The answer depends on where you go to gain access to an Internet gateway. HoloNet, for example, charges $6/month, or $60/year, to become a member. What you get in return is an Internet address. Mine is jkoloen@holonet.net. Having an Internet address means that anyone who can access the Internet E-mail system can send me messages electronically. Most of the commercial electronic services offer their members access to Internet E-mail. Which means that anyone on CompuServe or GEnie, for example, can send me E-mail whether I subscribe to those services or not. And I can likewise send them E-mail whether I'm a subscriber or not.

Additional feels are phone connect charges. HoloNet charges me $2 per hour during non-prime time and $4 during prime time to connect to its services. A surcharge of 95 cents during off-peak or $1.95 during peak hours is also assessed. This brings the average non-peak cost to $2.95 per hour.

Other charges include $1 per megabyte per hour for data that comes across my computer screen. However, the first megabyte per hour is free of charge. And, because it isn't possible to transfer even a megabyte per hour at 2400 baud, I pay nothing.

I can also store up to 256 kilobytes of data on the HoloNet computer at no charge. Additional storage is billed at $1 per megabyte.

Data transfer between the HoloNet computer and my computer is billed at $1 per megabyte during off-peak hours and $2 per megabyte during peak hours, exclusive of E-mail, which is always free. While many of the files and programs that I download go directly to my computer, some are downloaded at very fast rates from the host computer to the HoloNet computer, which then transfers it to me at my slower 2400 baud rate. However, over the several weeks that I've been on Internet, I've had only two files transferred this way. And both came from a computer at the University of Michigan, and both were programs. Most text goes directly from the host to my computer.

Charges may vary from service to service, but I find them to be reasonable given the access I get to information. If you're a student, check to see if your school has Internet access. If it does, chances are you can obtain an address and use the system for free. Similarly, if the company you work for as Internet access, check to see if you can use it's computers to access Internet. Going to a service such as HoloNet, while convenient and relatively inexpensive, is something you should do only if other options are closed.

LIMITED VS FULL ACCESS

If you are a frequent user of BBSes, you may have logged onto some that offer limited access to the Internet. I know of one in Austin that charges $48 per year in exchange for which you get access to Internet E-mail and the ability to download computer programs and other files, provided that you know the name of the program or file you want to download. However, you cannot directly access Internet. You simply post your message or request for a program or file and then wait for it to be forwarded to the appropriate receiver. In the case of requesting a program, it will either be forwarded to the BBS you use, or you'll be told it couldn't be located on the network. Given the $48 per year cost for this limited access, I opted for full access at a slightly higher fee with HoloNet. And I'm glad I did. Because, with full access, I can go anywhere in the world. On my own. With my TI.

Among the things I've done is to join three of the thousands of "news groups" on the Internet. News groups are special interest groups to which you may subscribe. Suppose you have an interest in the Fox TV show Beverly Hills, 90210. You can subscribe to the news group that focuses on this program. (It's run out of a computer at the University of California-Santa Barbara.) And after subscribing, any messages or files that are concerned with this show will automatically be forwarded to you at your E-mail address for perusal at your leisure. Remember, there are thousands of groups, and more being created every day. The subjects are as varied as the people on Internet.

Experimenting with the Internet can consume hours, and there's a lot to be learned, including commands, etiquette, and protocols.

(See Page 22)

For more information

Where can you turn for more information about the Internet? Here are a few ideas for further reading and, for those who'd like to get their feet wet, a phone number for HoloNet. Other companies provide a similar service.

Boardwatch Magazine, 800-933-6038, $3.95/copy

The Whole Internet User's Guide & Catalog, by Ed Krol, O'Reilly & Associates Inc., $24.95

Zen and the Art of Internet, A Beginner's Guide to the Internet, by Brendan P. Kehoe, downloadable from most commercial electronic services (available on Internet SIGs), Free

HoloNet, 800-638-4656 (modem, 8N1), 510-704-0160 (voice)

User Notes

Flash memory offered to Geneve users

Cecure Electronics is offering Geneve users who have boot ROMs a chip that will allow them to change the operating system in the boot ROM without rebooting the computer.

The new EPROM is a programmable flash memory chip (PFM). A program for a flash EPROM which allows the user to change the operating system at any time without removing the Geneve card has been developed by Cecure Electronics.and is near release, and a program dealing with ANSI graphics is available.

PFM, which was developed by Jim Schroeder, is an upgrade to the boot ROM sold by Cecure. The chip allows users to boot either from the permanent system memory, change to an alternate operating system, or install a new operating system in the permanent system memory, according to Don Walden, of Cecure.

The EPROM is not user installable, since it is a 32-pin chip while the Geneve uses a 28-pin chip. Cecure installs the chip as part of the purchase price. The tentative price is $75, including installation. Those who want the 128K RAMdisk option may have the 128K installed for an additional $25 if it is done at the same time as the PFM is installed.

PFM also eliminates the need for LOAD.SYS with a hard drive controller, he says. If the user selects the option to load from the hard drive, the program asks whether he wants to load from the hard drive itself or the floppy portion of the hard drive. Floppies on Myarc HFDC start at disk drive No. 5. The program requires no switches because flash memory can be reprogrammed by the user.

Currently, Walden says, the company is closing up any way which through malice or accident any programs running through this could corrupt a person's EPROM.

Tim Tesch has combined the color ANSI viewer for IBM files for Geneve in MDOS with Mike Maksimik's mouse key and a couple of special batch files by Maksimik for Cecure.

According to Tesch, the batch files will let a user make different menus for the Geneve and make menu choices with a mouse or keypress. It will allow the user to make color menus and run them out of MDOS for "a different look."

The program sells for $19.95 and includes demo files plus an additional program to dump color screens to the X-1000 Rainbow printer, he says.

For further information, contact Cecure Electronics Inc., P.O. Box 132, Muskego, WI 53150-0132 or (414) 679-4343.

READER TO READER

❑ Ralph E. Rees, 18815 N. 13th Ave., Phoenix, AZ 85027 writes:

I am looking for help from anyone who might know how to repair or get the Myarc HFDC to recognize and use a third hard drive. My card refuses to find "WDS3" and came that way new. I understand there may be a problem with the third 20-pin connector. I wrote Cecure Electronics but have received no reply.

Does anyone know of a better, *faster* way to back up a hard drive with 5¼ floppies? The Myarc MDM5 is too slow!I purchased HARDBACK from T&J Software, but it backs up only to another hard drive, not floppies. I was told Al Beard might have written something better? Thanks for the help.

Write him at the address above, or call the VAST Users Group BBS at (602) 233-0790 and leave E-Mail.

❑ Quinton Diggs, Route 1, Box 34, Xenia IL 62899 wants to know where he might find a copy of Land on Mars by American Software.

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.

Harrison offers music transcription, releases utility as PD

Harrison Software is offering a music transcription service for owners of MIDI-Master 99 and has released a new utility for Extended BASIC programmers, The Ultimate Accept At, as Public Domain software.

According to Bruce Harrison of the company, for the transcription service the customer sends the sheet music and a check, and the company provides a complete SNF source file on disk for that piece of music.

"In other words, our musician will do the hard part for you," he says, "reading the sheet music and typing the instructions in SNF format."

He describes the product as a "piano" arrangement which can be played on any make or model of MIDI instrument, which the user can tailor to other instrument arrangements.

The base price for this service is $5 per piece plus $2 shipping and handling for U.S. and Canadian customers. This price covers up to 70 measures of four parts or "tracks." Additional tracks, if required, cost 50 cents each, and measures beyond 70 are priced at 10 cents per measure. The buyer must supply the score or sheet music,

(See Page 23)

INTERNET—

(Continued from Page 21)

Most large bookstores should carry several books dedicated to the Internet. It's a hot subject right now, and one of the best is The Whole Internet by Ed Krol.

I'm only scratching the surface of Internet here. There's even a newsgroup for the TI, called "comp.TI," though it doesn't seem to be as busy as those on Delphi or GEnie. But there are thousands of other newsgroups that open up an entire world of information that is simply unavailable anywhere else.

It's a vast network, with millions of members around the world. It's endlessly varied and it's where the future of telecommunications is being made.

Newsbytes

(Continued from Page 22)

which will be returned along with the completed disk. Special arrangements (tailored to the user's instrument) will be provided on a "call for price" basis.

Partial measures count only if two or more of them occur in the piece. (Any "pair" of partial measures count as one measure.) Measures are counted "as printed" rather than "as played." Thus, *da capo* or *dal signo* repeats do not count as additional measures.

Harrison notes that customer Gene Bohot has advised the company that its Font Dumper for the Star NX-1020 will also work for the Star XR-1000 printer.

The Ultimate Accept At is available through user groups and through Tigercub Software, or from Harrison Software at $2 per copy.

Also, Harrison notes, the company has made a slight upgrade to its Easy Data product, so that program lines will never be confused with DATA lines in the Extended BASIC program.

"Most users of Easy Data won't ned this, but the update is available to existing owners at $2," Harrison says.

For further information or to order, contact Harrison Software, 5705 40th Place, Hyattsville, MD 20781, or (301) 277-3467.

New LOAD/SYS, MDOS V1.50 is released

Beery Miller of 9640 News has released a new version of LOAD/SYS on GEnie, Delphi and on his BBS. Miller has also released MDOS V1.50H, GPL 1.50H and MDM5 V1.50. Phone number for the 9640 News BBS in Memphis, Tennessee, is (901) 368-0112.

Those who are without modems may order all the above programs from MICROpendium for $4 each, including shipping. See the ad on page 32 for more information.

According to Miller, James Schroeder has modified LOAD/SYS so MDOS can now be loaded from Hard and Floppy Disk Controller floppies by users who have only one controller card.

Miller says support for SCSI for GPL mode (and WDS support for GPL mode on the HFDC) has been built into MDOS and the GPL interpreter. Exec will not be able to use WDS until a new version is released, he notes.

Miller lists the following as fixed or improved.

• Improved hard drive speed (2X-3X improvement, format at interlace 7 if you want the extra speed).

• HFDC formatting problem from MDOS CLI fixed relative to sector 0 problem. Miller notes, "The aborted verification problem is still there as it is a hardware problem, just format the disk a second time and the disk will format properly."

• Case insensitive CLI now available; type lowercase filenames and uppercase filenames will be found.

• Video display routines have been speeded up, many by 2X.

• Fixed mouse driver conflicts with using character definitions higher than 128.

• Fixed a small "quirk" in the Horizon RAMdisk support.

• Added user toggle to activate use of WDS on HFDC/SCSI at CRU >1200 while in GPL mode (allows Gentri/MDM5/HFSECTOR/SECTOR one for GPL/etc.) to function properly.

• Modified time/date function not to query for input if in batch mode.

GPL V1.50 has added support for CRU >1200 powerup if toggle active and the previous page forwarding scheme has been removed.

For MDM V1.50, Miller says, several files have been modified to allow full compatibility on a TI99/4A and tested on a Geneve 9640 with no problems.

Miller says his next project is to see what is needed on ABASIC.

Send information about your products and services to MICROpendium Newsbytes, P.O. Box 1343, Round Rock, TX 78680.

Products, networking featured on Lima fair videotapes

By LAURA BURNS

Presentations of products by Don O'Neil and Mike Maksimik lead off the videotapes of the Lima Multi Users Group Conference held in Lima, Ohio, May 15.

The seven-hour set of two videotapes is available to any users group, dealer or paid member of the Lima Users Group. For ordering information, contact the Lima Users Group, c/o Charles W. Good, Box 647, Venedocia, OH 45894.

Lima's videotapes have never been anything but amateur quality, and this year they have more than their share of glitches. The label notes that the last three presentations have no sound. A good chunk of the video portion of O'Neil's presentation is taken up with a title screen for James Schroeder's REDISKIT program, and a similar chunk of Maksimik's presentation shows a flickering, blank screen.

O'Neil demonstrated Horizon's SCSI, several prototype models of which were on sale at the fair. However, the DSRs (device service routines) have still not been completed. The SCSI will allow the user to attach up to seven devices to his TI.

O'Neil says he has had trouble finding chips to produce his Memex card

(See Page 24)

LIMA FAIR —

(Continued from Page 23)

He says he has been able to obtain chips from National Semiconductor. Memex allocates RAM to peripherals up to 16megabytes and allows the user to avoid cluttering the Horizon RAMdisk. It can have up to four 256K SIMs, O'Neil says. Price is $210 with a single 1-meg SIM, he said, which he called "plenty for most users."

Horizon produces a PAL which replaces the Geneve PAL, he says, and decrease the time it takes to access a video chip. A kit with chip and socket installation instructions is $25. Installation is $15 extra.

In July, according to O'Neil, Horizon plans to introduce a 99/4A EPROM AVPC interrupt system which will eliminate the need for an SOB card.

O'Neil discussed his Digiport, a digital sound player which runs out of the printer port.

Horizon products are available from Bud Mills Services, 166 Dartmouth Dr.., Toledo, OH 43614-2911. Phone number is (419) 385-5946 (voice) or (419) 385-7484 (BBS). Western Horizon Technologies (O'Neil) is at 10225 Jean Ellen Dr., Gilroy, CA 95020, (408) 848-5947.

Maksimik discussed Cecure Electronics" ANSI tools and demonstrated a tower case which can containing a Geneve, 1 meg. memory, two RS232 cards, HFDC and SCSI cards, hard drives and a tape drive. He says he is working on tape backup software for the TI and Geneve. Tape backup requirements for TI include upgraded memory in the hard and floppy disk controller

Maksimik also demonstrated his MIDI interface, sold by Cecure Electronics. He says V3.0 will allow recording of PC songs. Cecure Electronics is at P.O. Box 132, Muskego, WI 53150, (414) 679-4343.

At a meeting of users group officers, Charles Good of the Lima group announced that the fair had more registered participants than last year. Participants discussed the importance of telecommunications services They also noted that TI groups can provide equipment and service for school enrichment and for the children's departments of public libraries at low cost.

Chris Bodenmiller demonstrates his games Astromania, War on the Sea and Who's Behind the Mexican UFO? on the next section of tape. Bodenmiller Computers is at 43 Monroe St., Berea, OH 44017 He is followed by Ken Gilleland of Notung Software, 7647 McGroarty St., Tujung CA 91042, who demonstrates his Disk of the Ancient Ones and discusses his book How to Use The Printers Apprentice. Then Charles Good of the Lima Group demonstrates the V.50 40-column editor of Funnelweb.

Rick Kellogg acts as a "roving reporter" for the next section of tape, which shows various users groups, vendors and other attendees at the fair. A brief clip of an interview with Charles Good broadcast on WLIO-TV May 15 ends the first tape.

The second tape begins with a tutorial on Asgard's AMS development system,by Joe Delekto. The program TI-Nopoly by Jon Dyer and Delekto is demoed. The system, Delekto says, allows a programmer to develop large programs without knowing how the paging process works by using overlays.

Silent portions of the tape include Art Gibson demonstrating First Draft V.2, now compatible with Asgard's AMS; Bruce Harrison with The Ultimate "ACCEPT AT" and the Harrison Word Processor;and Gary Bowser of OPA, demonstrating a Pop-CArt with Rich GKXB, Graphic Editor V.5 and other software on one cartridge.

Extended BASIC 3 Super Module

Souped up programming and multi-function module

By JOHN KOLOEN

The Extended BASIC 3 Super Module is one of those cartridges that can't help but get good grades from a reviewer. It features a beefed up version of Extended BASIC, and a host of other programs that you can access from a menu, including Version 5.0 of TI-Writer, Terminal Emulator II, Editor/Assembler, Disk Manager 1000, Archiver 3.03, Mass-Transfer and Remind Me, Hang Man and a Hang Man demo. All of this has been stuffed into a big 256K module.

Obviously, the strength of this module depends on how much Extended BASIC has been improved. But before getting to that, the module is unique in its use of the speech synthesizer. When you make a selection from one of the menu screens, command is repeated by the speech synthesizer. Of course, if you don't have a speech synthesizer you won't hear this. But it's a nice touch.

When you plug in the module, you'll see a main menu screen that includes selections for Extended BASIC 3, the cartridge menu and TI-BASIC. By selecting the XB3 option, you are brought to the XB3 command line. Like Extended BASIC, DSK1 is searched for a LOAD program. If you want to be certain that you're using XB3 and not standard XB, simply use the SIZE command. You'll see a listing of program memory, stack memory and low memory. And you'll notice that lowercase and uppercase letters are used, unlike XBASICs uppercase only character set.

(See Page 25)

Review

REPORT CARD

Performance	**A**
Ease of Use	**A**
Documentation	**A**
Value	**A**
Final Grade	**A**

Cost: $99.95
Publisher: Asgard Software, 1423 Flagship Dr., Woodbridge, VA 22192; 703-491-1267
Requirements: TI99/4A console, monitor, expansion memory, cassette recorder; disk drive and speech synthesizer are recommended

EXTENDED BASIC 3 SUPER MODULE—

(Continued from Page 24)

If you select the option for the cartridge menu, you'll get a list of the following programs: Expanded BASIC (Extended BASIC 3), Editor/Assembler, TI-Writer, TI-BASIC, Terminal Emulator II, Hang Man game, and Hang Man demo. To select a program, you highlight it with the cursor using the arrow keys, and press Enter.

If you press the space bar, rather than Enter, a second menu will be displayed with the following contents: Disk Manager 1000, Archiver, Mass-Transfer (terminal emulator), Remind Me!(scheduler), TI-Writer editor and TI-Writer formatter.

From either of these menus, you also have the option to bring up another screen that brings up the OPA Memory Manager. This screen is used to manage the contents of the Extended BASIC 3 cartridge. It works as a simple disk manager and program loader. By selecting a device to catalog and pressing Enter, a catalog is displayed. You can select a program to run from the catalog by moving the highlight r to it and pressing Enter.

So much for the menus. The Extended BASIC 3 portion of the cartridge features a version of Extended BASIC that was written by Winfried Winkler. According to Asgard's Chris Bobbitt, Winkler corrected numerous bugs found in Extended BASIC, rewriting the code in GPL with critical routines written in assembly language. He says the cartridge's primary virtue is an increase of speed in running programs, anywhere from 10 to 200 percent, depending on the program.

Comparing the speed of programs running in Extended BASIC with programs running in Extended BASIC 3 is problematical. And the manual that comes with XB3 suggests as much. According to Asgard, the speed gains come mostly from more efficient use of memory, such as loading a program from disk. XB3 reduces the number of steps required to load a program from 6 to 3. In addition, users can expect improvements in speed due to improved memory usage with respect to string functions, "garbage collection" routines, and loading, saving and deleting lines of code from the command line.

Other functions that may be faster include floating-point math functions and random number generation because they were entirely rewritten. Also, GPL and ROM-based subroutines, included most sprite routines and graphic CALLs have been optimized.

The overall effect , collectively, of these improvements can't be generalized from program to program. So it's pointless for me to try to compare them.

What I regard to be among the major changes from XBASIC to XB3 are the use of control and function keys. Basically, all keys, when pressed while holding down the CTRL key, becomes a macro. For example, CTRL-S puts the following on the screen:
CALL SOUND(

CTRL-D puts this on the screen:
DISPLAY AT(

This goes on for every key on the console, which is very handy for anybody programming in XB3. Fortunately, many of the keys are mnemonic, which makes them easier to remember.

If you want to make the CTRL keys inactive, press FCTN-0. This turns the cursor white to remind you that the CTRL keys are inactive. The control keys remain inactive only until you press Enter. However, as long as the cursor is white, you can press the control and have its control character being displayed on the screen. Only a few of these will be visible.

Another handy keypress is FCTN-7. This is used after entering one of those long program lines that go beyond the 5-line limit of Extended BASIC. Normally, to complete one of these extra long lines you need to press Enter at the point where input won't be accepted anymore and then do a FCTN-REDO and cursor all the way to the end of the line to continue entering code. With XB3 all you do when you reach the end of the fifth line is press FCTN-7 and the cursor automatically advances to the sixth line where you continue entering data. Ain't that sweet?

Want to go back to the beginning of a program line after you've gotten to the end? Just press FCTN-5 and you'll jump back to the space after the line number. You can do this from anywhere in a program line. You can also move up or down a line at time with FCTN-6.

The MERGE command is replaced by an OUTPUT command and, if you protect a program using the protection option, you will not be able to list the program without causing any version of Extended BASIC to crash.

New command line functions include:

#"Devicename — produces a catalog of the specified device.

APPEND — all control characters (ASCII 128 and above) are redefined as inverse video.

ERASE Startline-Endline — deletes a specified range of program lines.

OUTPUT "DSKx.FILENAME" Startline-Endline — saves the specified lines to the specified file.

PERMANENT ON — this is the default mode of XB3. The TI character set is replaced with the XB3 character set that includes true lower/uppercase.

PERMANENT OFF — activates the TI upper/lowercase character set.

PERMANENT UALPHA — deactivates lowercase characters.

USING — lists all the CALL statements included in the internal table, if a running program is interrupted. Used for inserting pre-scan commands.

VARIABLE — lists all user DEFined functions and variables, if a running program is interrupted.

Additionally, the ACCEPT AT VALIDATE statement supports LALPHA, which restricts input to lowercase characters.

The following Extended BASIC functions have also been modified:

ASC(character string) — can now handle empty strings.

CLOSE #ALL — closes all open files.

DEF — user DEFined functions now may be used outside a running program. (The DEF statement itself may be used only inside a program.)

VAL(>"Hexadecimal-string") — converts a hex string of up to four characters into the corresponding numerical value.

The following control statements have been modified:

CALL GOSUB(line #) — line number can be a non-zero variable.

CALL GOTO(line#) — line number can be a non-zero variable.

These are the new functions added to XB3:

(See Page 26)

EXTENDED BASIC 3 SUPER MODULE—

(Continued from Page 25)

CALL BYE — used within a program.

DATE$,TIME$ — works with CorComp Triple Tech card, BWG Disk Controller with Clock and Austrian Hardware Clock Card.

HEX$(Number,Length) — supplies a hexadecimal number of length digits corresponding to he number given.

CALL NEW — works within a program.

Here are the changes in subroutine CALLS:

CALL CHAR(character_code,"hex_definition") — string definition may be more than 64 characters long.

CALL COLOR(character_set,foreground,background) — may use ALL to define the color of all character sets at once, ie. CALL COLOR(ALL,2,15).

CALL INIT — loads a shorter version of the known subroutines.

CALL PEEK(...) — Editor/Assembler syntax is now supported.

CALL VERSION(X) — the value of X is 150.

CALL LOAD("access-name"[,address,byte1[,...],file-field,...]) — used to load and link assembly routines. It runs at the same speed as the XBASIC LOAD command. It supports compressed object code format with references and the END/START auto-start feature of the E/A.

New subroutine calls include:

CALL ALL(character-code) — fills the screen with the ASCII code specified.

CALL ALLSET — like CALL CHARSET but includes lowercase TI characters.

CALL ALOCK(value) — the number of value is set to one if the Alpha-Lock is down, zero if it is not.

CALL BEEP — issues beep sound.

CALL CHAR ALL — like CALL CHARSET but includes lowercase XB3 characters.

CALL CHIMES — issues bell sound.

CALL CLRS — like CALL CLEAR but clears only columns 3-28, leaving remaining columns for borders.

CALL FIND(string-1,string-array2(),index) — sets "index" to the first occurrence of string "string-1" within one dimensional "string array2()".

CALL GPEEK(address,value) — same as PEEK but for GRAM/GROM.

CALL GPOKE(address,value) — same as CALL LOAD with bytes, used with GRAM.

CALL HONK — issues honk sound.

CALL PRNTPAT(number-char,character-data-string) — similar to CALL CHARPAT but the "character-data-string" contains data for "number character" appropriate for printing.

CALL QUIT OFF/ON — turns QUIT key on and off.

CALL RESTORE(line number) — same as RESTORE but used with numerical variables.

CALL RND(variable) — sets "variable" to a random number. Fast, but not as "random" as the RND function.

CALL SCREEN OFF/ON — when the screen is off, only the background color is displayed.

CALL VPEEK(address,value) — same as CALL PEEK but used with VDP RAM.

CALL VPOKE(address,value) — same as CALL LOAD but used with VDP RAM.

CALL WAIT(time) — causes computer to pause for specified time.

As you can see, this is not TI Extended BASIC. While it is not 100 percent compatible with TI XBASIC, Asgard says "the vast majority of TI Extended BASIC programs should function fine." I ran a number of programs and found no deficiencies. You can expect to see different error messages generated by some routines. For example, EOF(0) issues "bad value" instead of "file error." The biggest difference may occur with game programs that depend on a certain speed. XB3 runs faster and so CALL COINC loops may have to be adjusted. Of course programs that rely on undocumented techniques may encounter problems. An example is the use of a CALL PEEK to generate random numbers.

More favorably, XB3 offers more graphics support for CALL VCHAR, HCHAR, etc. Character codes from 143 to 159 can be used in XB3, though there are caveats which might limit their use. Using these characters to run in XB3 doesn't create a problem unless you decide to run the program using TI XBASIC. Obviously, if you use any of the new commands and functions supported by XB3, you can't expect the resulting program to run on TI XB.

Ease of Use: XB3 is more efficient than XB, particularly in its line editing functions. Programming with it is no more difficult than programming in XBASIC once you've learned how the modified commands, functions and statements work. The modifications certainly give programmers more flexibility.

Documentation: I did not have the complete documentation when reviewing XB3. What I had was a supplement covering all the new commands and functions in available in XB3, and a second supplement covering the cartridge and its myriad of programs. Asgard is finalizing a rewritten version of the TI XBASIC manual that will incorporate the new commands as well as those from TI XBASIC. I found that the supplements adequately covered all the new features, with programming examples for each.

Value: XB3 is an expensive module, but you get a lot for the money. Not only do you get an upgraded version of Extended BASIC, you also get two terminal programs (TEII also supplies the speech support), a word processor, a full-function disk manager, an archiver/dearchiver and a scheduling program. This combination of programs means that many would-be users could get along just fine without ever having to remove the cartridge from the GROM port. The only thing missing are a database program and spreadsheet.

You may ask, would it be cheaper if XB3 were sold separately from the other programs? According to Bobbitt, the initial expense of the module itself — including parts, components, boards and design — is what makes it expensive. Adding on other cartridge titles to use up available memory is cheap. Then, too, there are only two companies that even produce cartridges for the TI, one is in Germany and the other is OPA in Canada, which produces the Asgard cartridge. Also, producing cartridges in small volumes doesn't provide any volume price breaks.

Final Grade: If you enjoy programming in Extended BASIC, you will probably find XB3 to be an improvement. You can make that decision just by considering the list of features outlined here. In addition, if you use TI-Writer frequently, this becomes a better deal. And, if you have a modem, this cartridge begins to look like an outstanding value.

First Draft

BY Arthur Gibson

TI-99/4A owners use their computers for *word processing* more then anything else. Over the years, Asgard Software has tried, and usually succeeded in making word processing easier and more powerful. Our *Spell It!* spelling checker is the standard by which all others are measured, and our *Screen Preview* has helped many see what they have written before they print it.

Now, for the first time, we have combined all of the useful add-on utilities and nice features found in all of the *TI-Writer* compatible programs into one, advanced program. Written entirely from scratch, *First Draft* brings word processing for the 99/4A and Geneve into the 90's.

This *TI-Writer* compatible programs allows you to create documents as large as your disk in 40 or 80 columns. Spell-check a word or the entire document any time you want with our *Spell It!* compatible dictionary. Use the pull-down menus to access a wide array of commands for every-thing from merging documents to managing your disks. You can even add *D I S K U* comments to your files. Use the advanced formatter to print out multi-column documents with *Page Pro 99* graphics, and to take advantage of your printer's features just select the right "printer driver". The list of features and firsts goes on and on.

$39.95

Shipping & Handling:

U.S. Add $3.00
Canada Add $4.00
Airmail Add $7.00

Spell It! users only $29.95 plus S&H till 8/1/93

From *First Draft* to Final Copy, no other word processor for the 99/4A comes close.

New Version 2.0 Now Available! Key macros, improved 40-column mode with smooth scrolling on 80 columns, full 9640 support, faster disk access, and more!

Features

- The program may be fully customized
- Includes a 25,000 word spelling checker - check a single word or the whole document while writing
- Includes 40 and 80 column support
- TI-Writer compatible - load and save in both *First Draft* and TI-Writer formats
- Create, edit and print documents as large as your available disk space.
- Full built-in disk manager accessible from the editor
- Menu driven - no command line
- Support for AMS Extended Memory - swap between *First Draft* and Final Copy at a key press with your document in memory at the same time!
- Parallel columns - create and edit multi-column documents in the editor and the formatter
- Extended formatter featuring over 20 new commands - include *Page Pro 99* pictures in your document, multi-line footers/headers, etc.
- Printer drivers - use simple commands to access any printer's features. Never worry about codes again
- Built-in screen preview function
- Compatible with Funnelweb, Horizon RAM-disk, TIM Mechatronics, Dijit, and the Geneve 9640

Asgard Software
1423 Flagship Dr.
Woodbridge, VA 22192

User Notes

Lithium batteries and RAMdisks?

This item was written by Glenn Bernasek of the TI-CHIPS of Cleveland, Ohio. It was taken from a user group newsletter.

First of all, allow me to qualify myself on the subject of lithium batteries and portable power systems in general. I am employed as a Senior Laboratory Technologist specializing, for the past eight years, in rechargeable lithium fundamentals. Twenty years of my career has been devoted to the research and development of lithium battery systems.

I would like to present the following information to clear up some misconceptions.

• Most commercially available lithium batteries are essentially not rechargeable. Some rechargeable lithium batteries have been marketed, but they are few and far between.

• Primary (dischargeable only) lithium batteries have been developed for two reasons. They have the highest energy level per weight and an extremely long shelf life. More than 95 percent of the original power remains after storage of more than three years. This means that a lithium battery has an expected useful life of 5-10 years, depending on the level of usage.

• It takes a special charging unit, yet to be designed and marketed, to recharge lithium batteries. Without such a device, recharging lithium batteries is highly inefficient at best.

• The manufacturers of some RAMdisks include in their assembly instructions direction on how to wire the board one way for rechargeable nickel-cadmium (NiCd) batteries and another way for a lithium battery. This is because a voltage divider type circuit is present on the board and, if it is wired for NiCd installation, it will provide a charging bias voltage to the rechargeable NiCd batteries. However, if a lithium battery is used, the properly wired divider circuit will bypass the lithium battery and allow the AC converter to power the board. *Do not replace nickel-cadmium batteries with a lithium battery without changing the circuitry per manufacturer's instructions!* The lithium battery may be subject to shortened life if operated in a circuit not set for this system. This could also be hazardous to you or your computer — lithium metal batteries can burn under certain abuse conditions.

• Nickel-cadmium batteries have what is called "charge memory." The system will only recharge to approximately 75 percent of the last discharge. Therefore, the effective usable charge will be limited by the depth of discharge. If a NiCd cell isn't discharged very deeply, it will eventually lose its ability to recharge. This is why it is recommended that a NiCd battery be fully discharged periodically, and then recharged overnight to maximum capacity.

If I may, I'd like to offer a possible solution to some of the "memory loss" problems experienced with battery backed RAMdisks. I know that I have harped on the card edge contact problems. Well, I've found a new problem in the operation of my RAMdisk. It seems the chips in the sockets tend to come loose. I found that when I pushed them back into the sockets, my ROS didn't disappear. I actually felt them "click" into place.

In short, the power failures of the lithium battery may have been the fault of incorrect usage, and the RAMdisk's "forgetfulness" may be caused by dirty, corroded or loose card edge or chip contacts.

Comparing sector editors

This item, by Jan Alexandersson, of Sweden, has appeared in several user group newsletters.

Four sector editors were compared: Disk Patch (with Funnelweb), Disk Utilities by John Birdwell, Sector One by Randy Moore and Hard Master from Asgard. See the chart comparing features.

NOTES

Disk Utilities can copy only between two disks if you choose the same sector number.

Sector One cannot step to other sectors directly from the editor, so you must return to the menu and choose Read. Sector One writes a sector without double-checking that you really want to. This is a potential risk because the memory buffer and sector number are not always the same. Back and Forward don't read, but change the sector number only and not the memory. Double-sided, quad-density (720-kilobyte) disks show the wrong map sector. Sectors Used is sometimes incorrect — the last sector is sometimes missing — but despite this the number of used sectors is okay. The "=" seems to be misplaced in the 80-column version.

Hard Master shows a map of used sectors for floppy and hard disks. Hard Master can read and show Forth screens and also print them to a printer or a disk. All non-standard ASCII characters will be replaced by

(See Page 29)

Program Features

	Disk Patch	Disk Util	Sector One	Hard Master
Sector				
Read	Yes	Yes	Yes	Yes
Edit	Yes	Yes	Yes	Yes
ASCII/Hex	Yes	Yes	Yes	Yes
Step	Yes	Yes	(Yes)	Yes
Write	Yes	Yes	(Yes)	Yes
Printer		Yes	Yes	Yes
Print DSK		Yes	Yes	Yes
Compare			Yes	
Copy		(Yes)	Yes	Yes
Search Text		Yes	Yes	Yes
Set/Mark		Yes		
Set/Free		Yes		
File				
Search	Yes	Yes		Yes
Disk				
DSK catalog	Yes	Yes		Yes
Directory				Yes
Forth				
Show scrn				Yes
Print scrn				Yes
Medium				
Floppy	Yes	Yes	Yes	Yes
Hard disk			Yes	Yes
Display				
40 col.	Yes	Yes	Yes	Yes
80 col.			Yes	Yes

User Notes

(Continued from Page 28)

ASCII 32 (blank space). I may use this to convert Companion and Pascal text to D/V 80 files. (Companion is a word processor.—Ed.) On page 8 of the manual, I am not sure that the author is correct in his discussion of sector >20 - >3F. I think these are meant for showing bad sectors on the hard disk, so don't use CO (Copy) 0 20 20. This information could be used when deciding which sectors contain map information instead of the way that is done now. The CO command works only within the same device, so you cannot copy from a floppy to a hard disk, or WDS to DSK.

Both Sector One and Hard Master, although intended for use with hard drives, may be useful to a floppy disk user.

Improvements to screen dump

This comes from Oliver Hebert, of Brewton, Alabama. He writes:

Vern Jensen's Extended BASIC screen dump program is an excellent example of the progress that the younger programmers are making. However, there is a typo in line 32060. — 7 AND 2 should be Y AND 2.

Further, look at the following lines:

```
32010 ...CALL GCHAR(R,C,A)
32030 CALL CHARPAT(MIN(MAX,A
,32),143),H$)...
```

These statements limit the CALL CHARPAT...H$ (but not the variable A) by changing numbers <32 to 32, and numbers >143 to 143. The limits for CALL CHAR and CALL CHARPAT are 32 to 143. Therefore, limits placed in the program may seem like overkill, but they aren't. Character set 0 can be useful in creating a screen. CHR$(30) and CHR$(31) (cursor and border) can't normally be redefined, but you can change their color with CALL COLOR. The border character is often present on an XB screen, and can be used as a space if you need to redefine the space.

BASIC programs can have character sets 15 and 16 (CHR$(144-159). Actually, characters from 0 to 32767 can be sent to the screen, so programming to set limits of 32-143 is important. If CHR$(32), the space, has been redefined, changing <32 to 32 might be undesirable.

Try the following changes (Insert at the beginning of line 32010):

```
GR(31,0)=1 ::
```

This tells the program that CHR$(31) has already been defined (to all zeros), so we can point all of the out of range characters here and bypass the CALL CHARPAT and its calculations. Now, insert at the beginning of line 32020:

```
A=MIN(MAX(A,31),144)::
A=A+113*(A=144)::
```

Next, simplify the first part of statement 32030 to:

```
CALL CHARPAT(A,H$)::
```

The first statement limits the characters to the 31-144 range (<32 becomes 31, >143 becomes 144), and if the original was >143, the second statement changes it from 144 to 31. Now, all out of range numbers will print as a space, and be independent of the space, which may have been redefined.

Actually, I wanted to use these changes (at the same places as above):

```
GR(0,0)=1 ::
A=A+A*(A<32)+A*(A>143)::
```

This would have pointed the out of range characters to 0 rather than to 31. As stand-alone statements, they function properly. However, using my test screen, CHR$(0) doesn't give A=0 as expected: it results in A=-6400!^Perhaps the putting of unauthorized characters on the screen causes this.

If your printer supports 72 dpi plotter graphics, change ;CHR$(75); in statement 32010 to ;"*";CHR$(5); and your printout will be in true perspective.

If your printer supports setting of the left margin, change PRINT #7:CHR$(27); in statement 32000 to PRINT #7:CHR$(27);"l";CHR$(27);CHR$(27); and your printout will be centered. Type a lowercase "L" rather than a 1.

The program for my test screen is included in lines 100-130 at the end of this user note. The screen produced contains all characters from 0 to 383. Note that the characters beginning with 256 are repeats of 0-255.

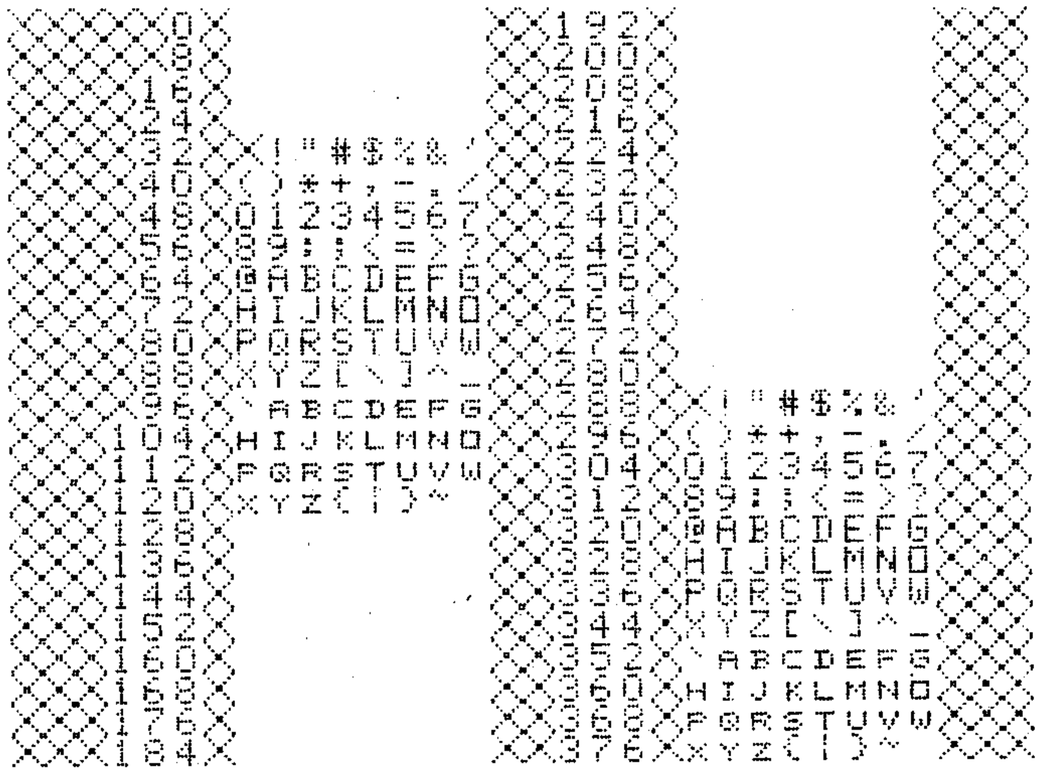
CHR$(0) PROBLEM

Adding A=A*1 to the end of statement 32010 will make the program run! GCHAR(R,C,A) :: A=A*1 produces the value 6400 for A, and A=A+A ... produces a zero. Without adding A=A*1, GCHAR(R,C,A) produces the value zero, and A=A+A ... produces -6400. Strange!

SCREEN DUMP

```
100 ! SCDUMP_TST After the
 first beep, press any key
 and start your timer (get
 second beep).  Stop your
 timer upon the third beep.
110 CALL VCHAR(1,1,32,768)::
 FOR S=0 TO 47 :: DISPLAY AT
(S+1+24*(S>23),2-14*(S>23)):
USING "###":S*8 :: FOR C=0 T
O 7
120 DISPLAY AT(S+1+24*(S>23)
,6-14*(S>23)+C):CHR$(S*8+C):
: NEXT C :: NEXT S :: CALL C
HAR(32,"8142241818244281")::
 CALL SOUND(-2E2,14E2,0)
130 CALL KEY(3,C,S):: IF C<0
 THEN 130 ELSE CALL SOUND(-2
E2,14E2,0):: CALL SCREENDUMP
 :: CALL SOUND(-2E2,14E2,0):
: END
```

SAMPLE SCREEN DUMP



(See Page 30)

User Notes

(Continued from Page 29)

```
31980 ! SCREENDUMP
      by Vern Jensen,
      Middletown, RI,
      from the Apr '93
      MICROpendium, page 29
31990 ! The typo (7 AND 2)
 in 32060 has been changed
 to (Y AND 2), and modifica-
 tions made to 32000-32030
 by Ollie Hebert.
32000 SUB SCREENDUMP :: DIM
GR(143,8):: OPEN #7:"PIO.CR"
 :: PRINT #7:CHR$(27);CHR$(6
5);CHR$(8);CHR$(27);"1";CHR$
(23):: B$="0123456789ABCDEF"
32010 GR(31,0)=1 :: FOR R=1
TO 24 :: PRINT #7:CHR$(10);C
HR$(13);CHR$(27);"*";CHR$(5)
;CHR$(0);CHR$(1):: FOR C=1 T
O 32 :: CALL GCHAR(R,C,A)
32020 A=MIN(MAX(A,31),144)::
 A=A+113*(A=144):: IF GR(A,0
)=1 THEN 32070
32030 CALL CHARPAT(A,H$):: F
OR P=1 TO 15 STEP 2 :: X=POS
(B$,SEG$(H$,P,1),1)-1 :: Y=P
OS(B$,SEG$(H$,P+1,1),1)-1
32040 Z=2^((15-P)/2):: GR(A,
1)=GR(A,1)+Z*SGN(X AND 8)::
GR(A,2)=GR(A,2)+Z*SGN(X AND
4):: GR(A,3)=GR(A,3)+Z*SGN(X
 AND 2)
32050 GR(A,4)=GR(A,4)+Z*SGN(
X AND 1):: GR(A,5)=GR(A,5)+Z
*SGN(Y AND 8):: GR(A,6)=GR(A
,6)+Z*SGN(Y AND 4)
32060 GR(A,7)=GR(A,7)+Z*SGN(
Y AND 2):: GR(A,8)=GR(A,8)+Z
*SGN(Y AND 1):: NEXT P :: GR
(A,0)=1
32070 PRINT #7:CHR$(GR(A,1))
;CHR$(GR(A,2));CHR$(GR(A,3))
;CHR$(GR(A,4));CHR$(GR(A,5))
;CHR$(GR(A,6));CHR$(GR(A,7))
;CHR$(GR(A,8))
32080 NEXT C :: NEXT R :: PR
INT #7:CHR$(27);CHR$(65);CHR
$(12):: CLOSE #7 :: SUBEND
```

Missing Link routine

This comes from Jim Lesher, of Dallas, Texas. He writes:

Look what you can do with The Missing Link employing only six lines of programming.

Since circles are easy to work with, let's experiment with some more of them. This program will show the simplicity of programming with TML and, at the same time, incorporate an illusion of animation.

Line 140 gives us the parameters and location of the circle, row 96 is in the center of the screen vertically, and 120 is in the middle of the screen horizontally. Now, the radius of the circle is replaced by a letter (a variable), so we are going to vary the size of the circle in steps of six dots, as indicated in line 130. We select 90 because it just fills the screen.

To get an even better idea of how this works, acquire some graph paper — I use one-quarter inch squares on 8 ½ x 11 sheets and number the squares 0-192 in 6 dot increments. The square in the left hand corner is zero. The next square is 6, the next 12, and so on. For me, the numbers come out perfectly (192 x 240). Then you can draw a form on the graph paper and know exactly where to place the dots.

```
100 ! CIR2
110 CALL LINK("CLEAR")
120 CALL LINK("COLOR",16,07)
130 FOR A=1 TO 090 STEP 6
140 CALL LINK("CIRCLE",96,12
0,A)
150 NEXT A
160 GOTO 110
```

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

Printing pillars

This item is by Morton Dworshak, a member of the Mid-South 99er User Group, Memphis, Tennessee. It appeared in the group's newsletter.

Most of us have seen great temples with lofty pillars reaching up toward the sky, but we have never been a part of building any of these great structures. Now, through an instrument called a computer, we can make pillars for the temple of our dreams.

I was working with programs to make graph paper, and as I dealt with the commands to create different distances between lines, I accidentally thought about progression from lines very close together to lines farther apart. This led to going from small to large distances, and then back again to small.

This program will allow you to make all sizes of pillars, and all on one sheet. (The pillars are down horizontally using a dot-matrix printer.—Ed.)

In order to draw correctly proportioned pillars, input the same number in lines 105 and 360. Both ask for the number of lines. As a start, try the number 5 in both inputs. The printer I have is a Star Micronics 10X. In line number's 220 and 390, CHR$(241) works better than CHR$(55). Check it out.

```
5 ! PILLARS (EXTENDED BASIC)
 !254
7 ! PRINT PILLAR PATTERNS BY
 MORTON DWORSHAK MID-SOUTH 9
9, JAN. 1993 !119
10 OPEN #1:"PIO" !253
15 PRINT #1:CHR$(27);CHR$(56
)!234
20 ! LN 15 ALLOWS ONE TO PRI
NT A SINGLE SHEET WITHOUT TH
E BELL AND RED LIGHT. !075
50 INPUT "DARKER LINES DESIR
ED? Y OR N? ":D$ !078
60 IF D$="Y" THEN 70 ELSE 90
 !046
70 PRINT #1:CHR$(27);CHR$(69
)!238
80 CALL CLEAR !209
90 DISPLAY AT(10,1):"PRINT T
HE FIRST HALF OF THE PILLAR.
" !175
100 ! 5 LINES MAKE NICE PILL
ARS! !072
105 INPUT "NO.  OF LINES DES
IRED?(MAX = 15) ?":N !145
120 FOR D=1 TO N STEP .5 !10
2
210 PRINT #1:CHR$(27);CHR$(5
1);CHR$(D)!032
220 PRINT #1:RPT$(CHR$(95),8
0)!246
250 NEXT D !218
260 CALL CLEAR !209
270 DISPLAY AT(10,1):"NEXT,
PRINT THE OTHER HALF OF THE
PILLAR." !058
290 ! FIRST CHANGE LN 120 TE
MPORARILY TO: FOR D=34 TO N
```

(See Page 31)

User Notes

(Continued from Page 30)

```
!019
350 ! NOW PRINT THE OTHER HA
LF OF THE PILAR. !158
360 INPUT "NO.  OF LINES? SA
ME AS ABOVE? ":N !218
370 FOR D=N TO 1 STEP -.5 !0
40
380 PRINT #1:CHR$(27);CHR$(5
1);CHR$(D)!032
390 PRINT #1:RPT$(CHR$(95),8
0)!246
400 NEXT D !218
410 PRINT #1:CHR$(27)&CHR$(6
4)!237
```

Digitizing software

Don Walden, of Cecure Electronics, is stocking Tim Tesch's digitizing software for the MVP card. It sends speech or music to Digiport or Polyport. Contact Cecure at 414-679-4343 for pricing and information.

MICROpendium pays $10 for items sent in by readers that appeared in the User Notes column. Send them to MICROpendium User Notes, P.O. Box 1343, Round Rock, TX 78680.

Classified

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.

Send classified advertising to: MICROpendium, P.O. Box 1343, Round Rock, TX 78680.

FOR SALE

GAMES!EDUCATIONAL!HARDWARE!—TI99/4A
CALL OR WRITE FOR FREE CATALOG:
JOY ELECTRONICS, INC; P.O. BOX 542526
DALLAS, TEXAS 75354-2526
(800) 527-7438, OUTSIDE DALLAS AREA
(214) 243-5371, DALLAS AREA
3/5

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

TAKE A BREAK THIS YEAR — ATTEND A TI FAIR

MICROpendium disks, etc.

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

Name _____

Address _____

City _____

State_____ ZIP_____

Check box for each item orderedand enter total amount here:

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

Credit Card # _____

Exp. Date _____

Signature _____

The ONLY monthly devoted to the TI99/4A

Subscription Fees

❑ 12 issues, USA, **$25**

❑ 12 issues, Mexico, **$30.25**

❑ 12 issues, Canada **$32.50**

❑ 12 issues, other countries surface mail, **$30.00**

❑ 12 issues, other countries, air mail, **$42.00**

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 ❑ MasterCard ❑ (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**, $2.50 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, $1.60 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 PUBLIIC DOMAIN DISKS

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

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

SECOND CLASS