Covering the TI99/4A and the Myarc 9640

MICROpendium

Volume 12 Number 3 April 1995 $3.50

Tex-Comp's 4A Division

Changing hands

See Page 5

Drawing a Straight Line

See Page 6

Reviews of DOS utilities for TI-Artist files and Mailing List Manager

Graphics programswith the Horizon RAMdisk

CONTENTS

MICROpendium

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

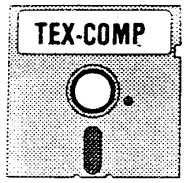
Mailing address: P.O. Box 1343, Round Rock, TX 78680.
Telephone & FAX: (512) 255-1512
Delphi TI NET: MICROpendium
GEnie: J.Koloen
Internet E-mail: jkoloen@io.com
John Koloen.....Publisher
Laura Burns.....Editor

*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.



ANNOUNCING NEW LOW PRICES ON FREEWARE DISKS FROM $2.95 TEX COMP LTD. $2.95

TEX-COMP HAS SLASHED PRICES ON THE BIGGEST & BEST COLLECTION OF FREEWARE FOR THE 99/4A. NOW ONLY $2.95 PER DISK WITH A 5 DISK MINIMUM. CHOOSE FROM HUNDREDS OF GREAT PROGRAMS. NO EXTRA CHARGE ON PROGRAMS THAT REQUIRE MORE THAN ONE DISK SIDE. ALL PROGRAMS HAVE BEEN TESTED AND ALMOST ALL HAVE BEEN PROVIDED WITH EXBASIC AUTOLOAD. CHOOSE FROM THE BEST IN GAMES, UTILITIES, GRAPHICS, HOME & BUSINESS AND DISK BACKUPS OF DOZENS OF YOUR FAVORITE MODULES THAT ARE NOW OUT OF PRODUCTION.

GAMES • BUSINESS • GRAPHICS • WORD PROCESSING • UTILITIES • DATABASE • MUSIC • COMMUNICATIONS • HOME

GRAPHICS, MUSIC & ANIMATION
#1 THE SINGING TI VOL. 1 (S)(2)
#4 PRINTART (2)(P)
#5A MUSIC/GRAPHICS
#6 EXBASIC MUSIC (2)
#7 SPACE SHUTTLE MUSIC/GRAPHICS
#9 MONA LISA PRINTOUT (P)
#11 ANIMATED XMAS (WOODSTOCK)
#14 FIGURE STUDIES (P)(MG)
#32 EXBASIC XMAS MUSIC (2)
#41 VIDEO GRAPHS (M)
#52 ANIMATION 99' (2)
#69 PLAYER PIANO/KEYBOARD ANALYSIS
#93 KBGB GIRLIE CALENDAR (4)(P)
#103 SORGAN THE TI ORGAN
#107 STARTREK MUSIC ALBUM
#111 POP MUSIC & GRAPHICS
#114 PANORAMA
#115 GRAPHICS DESIGN SYSTEM
#120 BITMAC (2)(P)
#230 THE SINGING TI VOL. 2 (S)(2)
#231 THE SINGING TI VOL. 3 (S)(2)
#246 THE SINGING TI VOL. 4 (S)(2)
#313 3-D WORLD (CAD for the TI)
COMPUTER UTILITIES, PRINTER UTILITIES & PROGRAMMING LANGUAGES
#3 DUMPIT (E/A)
#15 STAR/EPSON PRINTER DEMO (P)(2)
#16 SIDEWAYS PRINTOUT (P)(2)
#18 TI DIAGNOSTIC (MM)(2)
#8 LOADERS & CATALOGERS
#30 HOUSEHOLD BUDGET PRINTOUT (P)
#35 PROGRAMMING AIDS & UTILITIES I
#42 FUNNELWEB FARM (SHELL UTILITY)
#53 HACKER CRACKER
#55 SCREEN DUMP (P)
#62 DISK MANAGER II (M)
#75 DISK CATALOGER
#76 PROGRAMMING AIDS & UTILITIES II
#78 ARICON GRAPHIC CONVERSION
#79 DISK MANAGER 1000
#80 BIRDWELL DISK UTILITY (2)
#85 AUTOBOOT UTILITY
#86 COLUMN TEXT III (P)
#87 ARCHIVER III
#89 PROCALC DECIMAL/HEX CONVERTER
#96 STATISC & SORT ROUTINES
#97 MEMORY MANIPULATOR
#101 ENHANCED DISPLAY PACKAGE
#108 FUNLPLUS (PLUS!)
#110 DISK+ AID
#117 UNIVERSAL DISASSEMBLER
#119 RAG LINKER CONVERSION
#127 PIX GRAPHICS UTILITY
#243 OS/99 (GD)
#251 PC TRANSFER TI/IBM (DD)
#253 THE EXPLORER/DM1000
#254 NIBBLER/TURBO
#260 TI FORTH (DISK ONLY- add $8 for manual)
#17 TI FORTH DEMO
#116 TI FORTH TUTORIAL
#5079 FORTH SOURCE CODE
#104 C99 COMPLIER & LIBRARY
#5007 TEACH YOURSELF TI BASIC
#5019 TEACH YOURSELF EX-BASIC
#5067 BEGINNING BASIC TUTOR
#307 GRAPHICS CODE GENERATOR
#3912 SUPER BUGGER
#3913 BETTER BANNERS
#3914 CERTIFICATE 99
#3915 HOROSCOPE MAKER
#3916 GRAPHX+ PRINT SHOPPE
#3917-3720
GRAPHX COMPANIONS 1-4
#3721 MAC FLICK (G)
#3722 PRINTING TO GO (G)
#3723 GRAPHX DINOSAURS (G)

BUSINESS, ACCOUNTING, WORD PROCESSING, DATA BASE HOME OFFICE & HOME
#10 GOTHIC PRINTOUT (P)
#19 TI WRITER/MULTIPLAN UPGRADE
#20 ACCOUNTS RECEIVABLE (2)
#21 DATA BASE DEMO
#23 WILL WRITER
#29 LABEL MAKER I (P)
#36 STRICTLY BUSINESS (2)
#56 SPREAD SHEET
#58 PR BASE (database)
#59 GRAPH MAKER
#74 LABEL MAKER II (P)
#77 MICROdex 99 (database)
#81 HOME ACCOUNTING SYSTEM
#83 HOME APPLICATION PROGRAMS (2)
#90 JET CHECKBOOK MANAGER
#92 HOUSEHOLD INVENTORY
#109 TI WRITER MINI MANUAL
#112 INVOICE PACK
#113 LABEL MAKER III
#129 CASH DRAWER (point of sale)
#130 THE ORGANIZER
#147 CALENDAR-NOTEPAD
#177 HOUSEHOLD BUDGET MANAGEMENT (M)
#30 HBM DATA PRINTOUT (P)
#221 PERSONAL REAL ESTATE (M)
#249 MAPMAKER
#252 99 WRITER II (TI WRITER)(P)
#253 AMA MAILING LIST
#255 TRAK-A-CHECK
#257 DAILY DIARY
#257A EXER-LOG
#23 WILL WRITER
#310 SELF HELP TAX CUT
TELECOMMUNICATIONS (MODEM)
#57 TELCO
#57D TELCO (FOR SYSTEMS WITH DS DRIVES)
#118 FAST TERM
EDUCATION & PERSONAL DEVELOPMENT
#22 ASTROLOGY
#24 ENGINEERING CALCULATIONS (2)
#25 MEDICAL ALERT
#27 KIDS LEARNING I (2)
#31 MORSE CODE TRAINER
#37 LAPD COOKBOOK (2)
#40 ARTIFICIAL INTELLIGENCE (ELIZA)
#54 ASTRONOMY
#66 HEBREW TYPEWRITER
#67 GENEALOGY (2)
#71 KIDS LEARNING II (2)
#95 WEATHER FORECASTER +
#138 FIREHOUSE COOKBOOK
#142 TOUCH TYPING TUTOR (M)
#184 FACE MAKER
#194 ST. NICK/GHOSTLY SPELLING (M)
#195 TINY LOGO
#199 MILLIKEN ADDITION (M)
#200 MILLIKEN DECIMALS (M)
#203 MILLIKEN FRACTIONS (M)
#204 MILLIKEN INTEGERS (M)
#205 MILLIKEN LAWS OF MATH (M)
#211 MIND CHALLENGERS (M)
#212 MINUS MISSION (M)
#213 MILLIKEN PERCENTS (M)
#214 STORY MACHINE (M)
#215 BEGINNING GRAMMAR (M)
#216 METEOR MULTIPLICATION (M)
#217 HANGMAN (M)
#222 MUSICMAKER (M)
#223 PHYSICAL FITNESS (M)
#225 ALIEN ADDITION (M)
#226 ALLIGATOR MIX (M)
#227 DEMOLITION DIVISION (M)
#228 DRAGON MIX (M)
#270 TRIGONOMETRY
#271 CALCULATORS & CONVERSIONS (2)
#272 HIGHER MATH (2)
#273 ASTRONOMY 2
#306 SPEAK & SPELL II (S)(EX)

GAMES
#2 WH. OF FORTUNE, BLKJACK, POKER
#3 TI IRPK
#8 LOTTO PICKER
#13 STRIP POKER (PG)
#26 R-RATED NOVELTY GAME
#33 CHECKERS & BACKGAMMON
#34 SOLITAIRE & SCRABBLE
#38 GREAT II GAMES VOL 1
#39 GREAT II GAMES VOL 2
#43 BEST OF BRITAIN GAMES VOL 1
#45 BEST OF BRITIAN GAMES VOL 2 (LEGEND OF CARFAX ABBY GRAPHIC-INTERACTIVE ADVENTURE)
#46 SUPER TRIVIA 99
#47 INFOCOM RAPID LOADER
#48 GHOSTMAN (from U.K.)
#49 DEMON DESTROYER (from FRANCE)
#50 OH MUMMY! (HIT from GERMANY)
#51 BERLIN WALL (from CANADA)
#60 FREDDY (HIT from GERMANY)
#61 THE MINE (from GERMANY)
#63 ASTROBLITZ & MAZOG
#64 MAJOR IOM & SPACE STATION FHETA
#65 PERFECT PUSH (HIT)
#68 CHESS (SARGON)
#70 TI RUNNER II (HIT)
#72 CEBERUS (HIT SPACE GAME)
#73 CRYTO (GRAM)
#82 CROSSWORD (PUZZLES)
#84 GALACTIC BATTLE & SPY ADVEN.
#88 AUSSIE GAME COLLECTION VOL 1
#91 THE MAZE OF GROG (WOODSTOCK II)
#94 GREAT TI GAMES VOL 3
#98 DAYS/DOORS OF EDEN (BIBLE ADV)(AM)
#99 GREAT TI GAMES VOL 4
#100 ASSAULT THE CITY (TD)
#102 COLOSSAL CAVES (ADVENTURE)
#105 KINGS CASTLE (M)
#106 QUEST (D&D)
#121 SUPER YAHTZEE & WHEEL II
#122 ADULT ADVENT & GAMES (PG)
#123 GREAT TI GAMES VOL 5 (2)
#124 GREAT TI GAMES VOL 6 (2)
#125 BLACKJACK & POKER (M)
#126 VIDEO CHESS (M)
#128 TETRIS (HIT from RUSSIA)
#131 COMPUTER CRAPS
#132 AMBULANCE (M)
#133 DRIVING DEMON (M)
#134 ROTO-RAIDER (M)
#135 ARTURUS (HIT-ZARGON)
#136 ANT-EATER (M)
#137 CROSS-FIRE (M)
#139 MOONMINE (M)
#140 MASH (M)
#141 MOONSWEEPER (M)
#143 CONGO BONGO (M)
#144 STAR TREK (M)
#145 BUCK ROGERS (M)
#148 KENO & SLOTS
#149 GREAT TI GAMES VOL 7 (with HIT BLOCKBUSTER)
#150 ULTIMATE TRIVIA
#151 JUNGLE HUNT (M)
#152 POLE POSITION (M)
#153 DONKEY KONG (M)
#154 PROTECTOR II (M)
#155 PAC MAN (M)
#156 CENTIPEDE (M)
#157 DEFENDER (M)
#158 SHAMUS (M)
#159 MS. PAC MAN (M)
#160 DIG-DUG (M)
#161 PICNIC PARANOIA (M)
#162 MOON PATROL (M)
#3711 ARCADE SPECIAL (4 GAMES)
#3712 THREE GREAT GAMES
#3713 PRO TENNIS+

INFOCOM ADVENTURE BACKUPS
#163 ZORK I
#164 ZORK II
#165 ZORK III
#166 HITCHIKER'S GUIDE
#167 WITNESS
#168 ENCHANTER
#169 INFIDEL
#170 PLANETFALL
#171 SORCERER
#172 DEADLINE
#173 CUTTHROATS
#174 SUSPENDED
#175 STARCROSS
****
#176 AMAZING (M)
#178 DEMON DESTROYER (M)
#179 POPEYE (M)
#180 QUEBERT (M)
#181 METEOR BELT (M)
#182 BLASTO (M)
#183 CAR WARS (M)
#184 FACE MAKER (M)
#185 SUPER FLY (M)
#186 SPACE BANDITS (M)
#188 KILLER CATERPILLER (M)
#190 BLACK HOLE & SPACE AGRESS. (M)
#191 GREAT II GAMES VOL 8
#192 GREAT II GAMES VOL 9
#193 SPY'S DEMISE (HIT) (M)
#194 ST. NICK+ (M)
#196 JOTTO
#197 PRO TENNIS+ (HIT)(M)
#198 TI INVADERS/TOMBSTONE CITY (M)
#202 CONNECT FOUR (M)
#205 HOPPER (M)
#206 TREASURE ISLAND (M)
#206B SLYMOIDS (M)
#207 OTHELLO (M)
#208 FARSEC (M)
#209 SOCCER (M)
#210 SEWERMANIA (M)
#218 HUSTLE/FOOTBALL (M)
#219 CHISOLM TRAIL (M)
#220 ZERO ZAP (M)
#224 ATTACK! (M)
#229 4A FLYER (FLIGHT SIM.)(M)
#232 TUNNELS OF DOOM (MOD BACKUP PLUS 2 NEW ADVENTURES)(M)
#233 MS ADVENTURES (3 ADV-EXBASIC)
#248 STRIKE THREE BASEBALL (M)
#248 GREAT TI GAMES VOL 10
#250 BARRAGE/SPOTSHOT
****
#318 THREE GREAT GAMES
#319 ARCADE SPECIAL
#317 BEANSTALK ADVENTURE
****
SCOTT ADAMS ADVENTURES (USE WITH ADVENTURE MODULE)
#349 II ADVENTURES 1-13+
#350 TI ADVENTURES 14-16+

P = Printer required
G = Graphx required
S = Speech required
M = Module Backup
MM = Mini Memory req.
E/A = Editor Assem. Exbasic and 32k mem req. for most programs.

Authorized Dealer

TEX COMP LTD.

America's Number One TI computer retailer

425 E. Arrow Hwy. #732
Glendora, CA 91740-5684

ORDER BY PHONE

(818) 339-8924 FAX (818) 858-2785

24 HOURS A DAY

TERMS: MINIMUM ORDER FOR $2.95 PRICE IS 5 DISKS (REG. 4.95) ADD $4.00 PER ORDER FOR SHIPPING (U.S.)
ALL PRICES ARE FOR CASH/CHECK, ADD 3% FOR CREDIT CARD ORDERS. INCLUDE STREET ADDRESS FOR U.P.S.

COMMENTS

Jerry Price bows out

It's going to be hard to imagine the TI99/4A without the involvement of Jerry Price and Tex-Comp. Tex-Comp and Jerry were among the first of the third-party companies supporting the TI. For a long time Tex-Comp has been the biggest.

I remember in 1984 when Jerry Price first called me about placing advertising in MICROpendium. We'd just given one of his programs a somewhat disappointing review. I'd written that it was fun to play but had a lot of bugs. I never expected to get advertising from Tex-Comp. But he called and said he wanted three pages of advertising. I was incredulous. Despite the bad review? He said that after the review came out sales of the adventure game soared.

Tex-Comp was our largest advertiser. Period. Nobody else even came close. But while some advertisers wanted special consideration, which we never gave, Jerry never even asked for it. There was a time when some TIers thought we were in Tex-Comp's pocket, but it never happened. Jerry is and was a pro and a very honorable man.

I wish Jerry and Tex-Comp all the luck in the future. They've been very good to the TI market and very good to MICROpendium. I'm sorry to see him go. The TI market is losing a very valuable friend.

For more information about the company that will be taking over Tex-Comp's TI99/4A business, see page 5.

MISTAKES WERE MADE

Did we make a mistake last month. In the article entitled "Relative files — Extended BASIC programs focus on languages," several of the programs printed out with misprints. One of the programs, called INTDATA, we are still unable to correct, though its function can be handled by TI-Writer. And we assume virtually all of our readers has access to TI-Writer or one of its clones.

I'm not sure how the problem was created, except that it was an emergency replacement for another article and program. We're running corrected versions of the programs along with the second installment of the article this month. The programs on the March MICROpendium disk were corrected prior to shipment, except for the INTDATA program.

CAN'T WIN

I noted last month that we had gotten the numbering wrong on Bruce Harrison's column in the February edition. So, what did I do? I put the wrong number on his March column as well. This month's column is Part 46. Last month's was Part 45, which I called Part 43. I can only imagine how confusing this will be in the MICROpendium index.

SCSI PROGRESS

I've got my SCSI card and a power supply and a 40-megabyte SCSI hard drive and cables and software. But it's not up and running — yet. You see, I've got a GENMOD and the SCSI doesn't like it. According to Mike Maksimik, the problem has to do wi timing signals. I won't bore you with the details, but I will say that anyone with a GENMOD probably shouldn't order a SCSI card until a fix is ready. Mike says he'll work on it as soon as he gets a GENMOD from Bud Mills. I'll keep you posted.

—JK

FEEDBACK

Second the motion

I wish to second Mickey Cendrowski's suggestion (Feedback, Jan. '95) that you include some kind of user group listing in your publication.

May I suggest that, as space is available, that you publish the address of the groups which contact you and want to be listed. You might list the name, addr3ess and a telephone number for a contact person. You might also list the day and time of that group's meeting.

The M.U.N.C.H. group can be contacted at my address, my telephone number is (508) 869-2704 and we meet on the second Tuesday of the month.

I also want everyone to know that our group is in the planning stage for a TI fair in the Central Massachusetts area to be held in mid-October of this year. I hope to have a formal announcement next month.

In conclusion I want to thank you for an excellent publication and any help you can give the users' groups will be appreciated.

James W. Cox
905 Edgebrook Dr.
Boylston, Massachusetts 01505

Add him to list

Just looking through my February issue of MICROpendium and noticed the addresses of the people in the TI community. My name as it is used and on which net is listed below.

Name: Arthur Jr., Richard C.
Address:
R.C.ARTHUR@GENIE.GEIS.COM
and Internet:
FLBOY@FREENET.FSU.EDU
FLBOY@FREENET.TLH.FL.US

Richard C. Arthur
Tallahassee, Florida

Program errors create challenge

I enjoy your magazine very much, been getting it since 1986. I type in all of the programs, that way I can see how the programs are constructed and learn some pr gramming. Once in a while a printing er makes getting the programs to run a challenge. In the February issue, I think I have

(See Page 5)

New company takes over Tex-Comp TI99/4A division

By LAURA BURNS

Tex-Comp's TI99/4A division has new owners as of April 1, according to Jerry Price, Tex-Comp vice president.

The new company, Tex-Comp Ltd., is operated by Carey Hoffman. The 24-year-old Hoffman has been working with the TI for 17 years, he says.

"I like the TI," he says. "It comes up a lot quicker than an IBM, that's for sure."

He will be aided in the venture by his father, Larry Hoffman of the West Covina 99ers. The elder Hoffman programmed TI's Speak and Spell to work with Extended BASIC in a project sponsored by Tex-Comp and approved by Texas Instruments, according to Price.

Address of the new company is Tex-Comp Ltd., 425 East Arrow Highway, Suite 732, Glendora, CA 91740-5684. Voice phone is (818) 339-8924 and Fax is (818) 858-2785. In addition, callers can contact the company under the user name TEX-COMP on the bulletin board of the West Covina 99ers, (818) 339-1134.

Carey Hoffman says one project for the new company is "bringing the Databiotics line back to life." The new company has bought out the remaining DataBioTics stock and will be selling its products, he says.

DataBioTics produced numerous cartridges for the TI99/4A, including word processors, multi-screen games, spreadsheets and printer interfaces.

Price says he began negotiating the transfer at Fest-West, held in San Diego, California in February. During the transition months, he will consult with the new firm, he says.

"I feel very comfortable with these people taking over," Price says. "I would not have sold it to just anybody."

Price began his mail order company for the TI99/4A in 1981, when he noticed a need for a source of software other than the popular titles then carried in department stores.

At the time he entered the mail order business, "a couple of distributors and stores had mail order divisions," but he was one of the first TI dealers to make it his major division. Tex-Comp's TI business was in the millions-a-year category at one time.

A tornado that took the roof off a company storage building in December 1991. Later the company sustained minor damage in the Los Angeles-area January 1994 earthquake.

FEEDBACK

(Continued from Page 4)

found two errors. First, on page 28, line 40 belongs to the HIDE-64 program; can't have a program line after a subroutine. Also, HIDE-64 should have different line numbers. The way it is typed you would think it is all one program. Second error is on page 30, line 480 is not complete and lines 490-550 are missing.

Keep up the good work, yours is the only magazine we still have for the TI computer.

Harold Panzer
West Covina, California

You're correct about line 40. It should be the first line of HIDE-64, not the last line of FIND-64. The line numbering is the that of the programmer's, Don Steffen. We do not usually change line numbering.

Good catch on APTITUDE. The apparent explanation for the missing lines is that line 550 was corrupted and when we trasferred the text file listing from the Geneve to the Mac, which is what we use to lay out MICROpendium, the glitch in 550 caused the lost of the previous lines. I went back to the original program, and line 550 is corrupt there. Here is line 550:

```
550 DISPLAY AT(9,8):"pr ss
 y    y" !236
```

All I can say about line 550 is that it didn't interfere with the program when we ran it. When we originally ran the program, we apparently never accessed line 550, which is why it loads and seems to run properly, at least for a while. The original program is named TCX-1129 in Jim Peterson's numbering system. Anyone with an uncorrupted version of this line is invited to provide it to us so that we can pass it on to our readers.—Ed.

```
480 CALL HCHAR(3,30,K+79)::
CALL COLOR(K-47,7,7):: DISPL
AY AT(6,24):CHR$(137)&CHR$(1
40)&CHR$(127)!014
490 CALL HCHAR(6,30,30)!252
500 FOR D=1 TO 10 :: NEXT D
:: CALL HCHAR(6,30,32):: CAL
L KEY(0,K2,ST):: IF (ST=0)+(
K2<49)+(K2>57)+(K2=K)THEN 49
0 !046
510 CALL HCHAR(6,30,K2+79)::
 CALL COLOR(K2-47,7,7):: IF
((K-48=M)*(K2-48=N))+((K-48=
N)*(K2-48=M))THEN 570 !039
520 IF (K=N)*(K2=M)THEN 570
!181
530 CALL SOUND(500,30000,30,
30000,30,400,30,-4,0):: CALL
 COLOR(K-47,2,2):: CALL COLO
R(K2-47,2,2):: FOR D=1 TO 50
0 !175
540 NEXT D :: CALL COLOR(M+1
,7,7):: CALL COLOR(N+1,7,7)!
103
```

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

THE ART OF ASSEMBLY — PART 46

Drawing a Straight Line

By BRUCE HARRISON

While we were developing our drawing program, we found the need for some way to make a straight line between any two places on the bit-map screen. That's not so easy as it might seem, especially if one wants the drawing to happen very quickly. One could use any of a host of methods, including those using sines and cosines of angles, but the burden in those methods is that there are numerous floating point calculations required, and thus the operation is slow, even when Assembly is used.

It's been a long time since high school trigonometry, and much of that sine-cosine stuff has been lost in foggy old memories. In such cases we try to follow a rule we heard, attributed to Albert Einstein, to wit : "Never bother to memorize anything that you can look up in a book." We keep lots of books handy, and now and then find something really valuable in them. One of these is a book on PC Assembly language, which contains some interesting ideas for graphics programming. It's called *Assembly Language Primer for the IBM PC & XT*, by Robert Lafore, 1984, The Waite Group, a Plume/Waite book, published by NAL Penguin Inc., New York, NY and Scarborough, Ontario. We pulled that book out of the mess here in our computer chamber, and looked at the graphics chapter. Sure enough, this had the answer to our need.

A very clever guy named Bresenham devised an algorithm for making straight lines on a computer screen. This algorithm takes full advantage of the discrete nature of the "pixel" structure, and uses only very simple math (add, subtract, compare) to perform all its steps. Thus it works very quickly to draw the "ideal" straight line on the screen.

SIMPLICITY ITSELF

We enter the algorithm with four numbers, these being the X and Y coordinates of the start point, and the X and Y of the ending point. These four numbers are in "pixel" coordinates, of course, so that on the TI the X numbers range from zero through 255, and Y from zero through 191. Our drawing program doesn't use all of that range, but the concept is the same. The algorithm calculates where each point in the line should be drawn, pixel-by-pixel.

There was some adaptation required, since the version in the book was for PC Assembly language. We found that the book version used the SI and DI registers to store some of the variables, and of course the TI doesn't have such registers. We made up for that by setting aside two words in our DATA section called ESSEYE (for SI) and DEEEYE (for DI). Also, since our drawing program already had a PLOT subroutine to place the pixels on-screen, we simply BL to that instead of using the method for plotting the points as given by the book.

TODAY'S SIDEBAR SHOWS...

The code in this month's sidebar is not complete, but merely a fragment that you could surround with your own code. As shown, it starts with the TI already in bit-map mode. In parts 42 and 43 of this series, you'll find the code to get in and out of bit-map. The code shown uses the position of a sprite to supply the coordinates for the start and end of the line. You could substitute some other method of setting these coordinates, and would not necessarily need to add the "offsets" that we show in our code. Those were necessary because the crossing of the two lines in the "+" that we use as a cursor is not at the reported sprite position.

In short, modify to your heart's content, so long as all the required variables get supplied to the algorithm. Those variables are:

X1 = start horizontal position
Y1 = start vertical position
X2 = ending horizontal position
Y2 = ending vertical position
DELX = absolute value of X2-X1
DELY = absolute value of Y2-Y1
ESSEYE = sign of DELY (+1 or -1)
DEEEYE = sign of DELX (+1 or -1)
HALFX = half of DELX
HALFY = half of DELY

In the sidebar, you'll see that we've used registers R3, R4, R5, and R6 to hold these numbers while math was being performed, but put them all into the variables before the line starts being drawn. We've then put X1 and Y1 into R7 and R8, since those are used by our PLOT subroutine as the X and Y positions for plotting a point.

THE HARD PART

The real work of the algorithm starts at label STALG, where we compare @DELX,@DELY. If DELX is lower than DELY, then we have a "steep" line to draw (above 45 degrees), else we have an "easy" line. The main difference between LEASY and STEEP is which variable determines how many points must be plotted. For a steep line, we need to plot DELY points, while for an easy one we must plot DELX points. In other words, the algorithm plots the required number of points, that being the number in DELX or DELY, whichever is the greater number. As we enter the algorithm, R3 still contains DELY, and R4 still contains DELX, so we've taken a shortcut at labels LEASY and STEEP by simply shifting right by one bit to make the appropriate HALF value.

In our implementation, we've used R10 as temporary storage, R12 to count the number of pixels we'll plot, and R9 for the color of the line to be drawn. Thus for an "easy" slope, we put DELX in R12 to count points, while for steep lines, we put DELY in R12. The key to this whole process is the introduction of what Bresenham calls an "error term". This is a variable that tracks how far the current plotted point strays from what would be the ideal position. In our implementation, we cleared R10 at the outset, so R10 serves as the "error term" variable.

STEP-BY-STEP

Let's just for the moment assume that we're going to make a line starting at 10,10 and going eight pixels to the right horizontally and six pixels upwards vertically from that point. This is a slope of the "easy" variety, so we'll start our step by step at label LEASY in the sidebar. Since we're going up, ESSEYE will be -1, while DEEEYE will be +1, going to the right.

Our implementation is slightly complicated by the fact that we can use the samepart of the code to either draw a line or erase one, but we'll assume that we're

(See Page 7)

THE ART OF ASSEMBLY —

(Continued from Page 6)

not erasing, so ERSFLG will be zero. At label LEASY, R4 still contains what we stored at DELX, so we shift that right one bit, cutting its value in half, then move that value to HALFX. Now DELY contains 6, DELX contains 8, and HALFX contains 4. We'll move DELX into R12, so we'll plot 9 points total for this line (including the start point and end point). At LDOT1, we find ERSFLG is zero, so we will MOVB @LINCLR,R9. Next we'll BL @PLOT to put a pixel on-screen at 10,10, then jump over label LEASE to LDOT2. Now we'll move to the next dot-column by adding what's at DEEEYE to R7.

Now the trick of Bresenham's algorithm. We add DELY to R10, so R10 contains 6. We compare that to HALFX, which contains 4. If R10 is greater than HALFX, (it is this time) then we subtract DELX from the error term, and add ESSEYE (-1) to R8, so our next pixel will be plotted at 9,11. Our error term in R10 is now at -2.

After plotting that second pixel, we can add 6 to R10, so it's now equal to 4. Since that's not greater than HALFX, we don't change R8, and we don't subtract anything from the error term. Thus our next point gets plotted at 9,12.

This gets tedious, doesn't it! For those who want to see the rest of this process played out, there is an Extended BASIC program called BRESH which is listed in the sidebar. This program will make a line of "pixels" using cursors for each dot, and will report at the top of the screen the variables as they change. Just press a key to make each "dot". To make it look better on the screen, we started the line at 24,10, but otherwise it's just as we described above.

For those who want to experiment beyond the one line, there's a second XB program called BRESH4, which allows you to enter your own Deltas, and then makes the line for you, starting at 24,1. Notice here that the algorithm doesn't get confused even if either Delta is zero, or even if both are zero. The limits are 27 for DELX and 23 for DELY, so we won't try doing an illegal DISPLAY AT. When BRESH4 finishes a line, it will wait for a keypress. Pressing R will take you back to do another line, while any other key will exit the program. You can of course do the same kind of experiments with the BRESH program, changing the parameters to see what happens, except that BRESH is only set up to handle "easy" lines, not steep ones. The variables are named to coincide with the way the Assembly version works, so that it'll be easier to follow.

MATH WHIZ NEEDED

We've explained what Bresenmham's algorithm is, what it does, and how it does what it does, but that doesn't mean we understand why it works. The book that we took it from says the error term "... is related to the difference between where the pixel should go, if it could be drawn right on the line, and where it must go, since it can only occupy integer pixel locations." Yes, that's fine as far as it goes, but it still doesn't say why adding and subtracting and comparing in this way accomplishes the desired goal. We don't really know either. Perhaps somebody in our worldwide readership will be able to explain this to the rest of us! For the time being, we're very happy that it does work, producing a near-ideal straight line with discrete pixels. We've passed it along to you in the sidebar, so others may get some use from it, and that's the best we can do.

That's all for this month. Next month's topic is undecided at this point, so you'll just have to wait and see what's in next month's column.

Below is a test program (BRESH) written in Extended BASIC, to show how Bresenham's algorithm would work for a line of cursors from 24,10, going eight units right and 6 upward. The listing is in 28 columns. R10 is the "error term."

BRESH

```
10 CALL CLEAR
20 DELX=8 :: DELY=6
30 HALFX=4 :: HALFY=3
40 R12=8 :: R10=0
50 DEEEYE=+1 :: ESSEYE=-1
60 R8=24 :: R7=10
70 DISPLAY AT(1,1):"R12=";R1
2:"R10=";R10:"R8 (ROW)=";R8:
"R7 (COL)=";R7: : : :
80 DISPLAY AT(R8,R7):CHR$(30
)
90 R7=R7+DEEEYE
100 R10=R10+DELY
110 IF R12=0 THEN DISPLAY AT
(6,1):"FINISHED" :: GOTO 130
120 DISPLAY AT(5,1):"R10+DEL
Y=";R10:"HALFX=";HALFX
130 CALL KEY(0,K,S)
140 IF S<1 THEN 130
150 IF R10<=HALFX THEN 180
160 R10=R10-DELX
170 R8=R8+ESSEYE
180 R12=R12-1
190 IF R12>=0 THEN 70
```

Following is an XBprogram (BRESH4) that emulates the action of the assembly version so you can see more clearly what happens.

BRESH4

```
10 CALL CLEAR
20 INPUT "DELTA X (0 - 27) "
:DELX :: IF DELX<0 OR DELX>2
7 THEN 20
30 INPUT "DELTA Y (0 - 23) "
:DELY :: IF DELY<0 OR DELY>2
3 THEN 30
40 CALL CLEAR
50 HALFX=INT(DELX/2):: HALFY
=INT(DELY/2):: R10=0 :: DEEE
YE=+1 :: ESSEYE=-1 :: R8=24
:: R7=1 :: IF DELX<DELY THEN
 110
60 R12=DELX
70 DISPLAY AT(R8,R7):CHR$(30
)
80 IF DELX<DELY THEN 110
90 R7=R7+DEEEYE :: R10=R10+D
ELY :: IF R10<=HALFX THEN 10
0 ELSE R10=R10-DELX :: R8=R8
+ESSEYE
100 R12=R12-1 :: IF R12>=0 T
HEN 70 ELSE 150
110 R12=DELY
120 DISPLAY AT(R8,R7):CHR$(3
0)
130 R8=R8+ESSEYE :: R10=R10+
DELX :: IF R10<=HALFY THEN 1
40 ELSE R10=R10-DELY :: R7=R
7+DEEEYE
140 R12=R12-1 :: IF R12>=0 T
HEN 120
150 CALL KEY(0,K,S):: IF S<1
 THEN 150 ELSE IF K=ASC("R")
THEN 10
```

(Sidebar on Page 8)

THE ART OF ASSEMBLY—

```
0001  * SIDEBAR 46
0002  * A SIMPLE AND FAST METHOD
0003  * FOR DRAWING PIXEL LINES
0004  * BETWEEN TWO POINTS
0005  * THIS IS A FRAGMENT FROM OUR
0006  * DRAWING PROGRAM, NOT COMPLETE CODE
0007  *
0008  * AT THE ENTRY POINT, THE USER HAS PLACED
0009  * SPRITE #0, A + SIGN ACTING AS THE DRAWING
0010  * CURSOR, AT THE SPOT WHERE THE LINE IS TO START
0011  * THE KEY HAS BEEN PRESSED TO INDICATE THAT,
0012  * SO THE CODE BELOW STASHES AWAY THE COORDINATES
0013  * AFTER ADDING OFFSETS SO THE POINT STORED IS
0014  * THE CENTER OF THE + CHARACTER
0015  *
0016        LI   R0,>3800     POINT AT START OF SPRITE POSITION
TABLE
0017        BLWP @VSBR        READ Y OF SPRITE 0
0018        SRL  R1,8         RIGHT-JUSTIFY
0019        AI   R1,5         ADD OFFSET FOR CENTER OF +
0020        MOV  R1,@Y1       Y1=START Y POSITION
0021        INC  R0           POINT TO NEXT VDP BYTE
0022        BLWP @VSBR        READ X OF SPRITE 0
0023        SRL  R1,8         RIGHT-JUSTIFY
0024        AI   R1,3         ADD OFFSET FOR CENTER OF +
0025        MOV  R1,@X1       X1=START X POSITION
0026  *
0027  * BETWEEN HERE AND THE CODE BELOW,
0028  * THE "CURSOR" SPRITE HAS BEEN MOVED
0029  * BY THE USER TO THE DESIRED END-POINT
0030  * FOR THE LINE SEGMENT, AND A KEY HAS BEEN
0031  * PRESSED TO INDICATE THIS IS THE ENDPOINT
0032  * THE CODE BELOW, THEN, STARTS BY CAPTURING
0033  * THE END-POINT COORDINATES
0034  *
0035        LI   R0,>3800     POINT AT Y OF SPRITE 0
0036        BLWP @VSBR        READ THAT
0037        SRL  R1,8         RIGHT JUSTIFY
0038        AI   R1,5         ADD OFFSET
0039        MOV  R1,@Y2       Y2=END Y POSITION
0040        MOV  R1,R3        R3 HAS Y2
0041        INC  R0           POINT AHEAD TO NEXT VDP BYTE
0042        BLWP @VSBR        READ X OF SPRITE 0
0043        SRL  R1,8         RIGHT JUSTIFY
0044        AI   R1,3         ADD OFFSET
0045        MOV  R1,@X2       X2=END X POSITION
0046        MOV  R1,R4        R4 HAS X2
0047        S    @Y1,R3       R3 HAS DELTA Y
0048        JLT  LD40         IF BELOW ZERO, JUMP
0049        LI   R5,1         ELSE R5=1
0050        JMP  LD41         THEN JUMP
0051  LD40  LI   R5,-1        R5 = -1
0052  LD41  ABS  R3           R3 HAS ABS. VAL. DELTA Y
0053        MOV  R5,@ESSEYE   SAVE R5 TO ESSEYE
0054        MOV  R3,@DELY     SAVE R3 AT DELY
0055  *
0056  * AT THIS POINT, DELY HAS THE DIFFERENCE IN Y POSITIONS
0057  * ESSEYE HAS THE SIGN (+1 OR -1) OF THAT DIFFERENCE
0058  *
0059        S    @X1,R4       R4 HAS DELTA X
0060        JLT  LD50         IF BELOW ZERO JUMP
0061        LI   R6,1         ELSE LOAD R6 WITH 1
0062        JMP  LD51         THEN SKIP AHEAD
0063  LD50  LI   R6,-1        LOAD R6 WITH -1
0064  LD51  ABS  R4           R4 HAS ABS. VAL. DELTA X
0065        MOV  R6,@DEEEYE   SAVE R6 AT DEEEYE
0066        MOV  R4,@DELX     SAVE R4 AT DELX
0067  *
0068  * AT THIS POINT, DELX HAS THE DIFFERNCE IN X POSITIONS
0069  * DEEEYE HAS THE SIGN (+1 OR -1) OF THAT DIFFERENCE
0070  *
0071        MOV  @X1,R7       PUT START X POINT IN R7 (DOT-COL-
UMN)
0072        MOV  @Y1,R8       PUT START Y POINT IN R8 (DOT-R
0073        CLR  R9           CLEAR R9 (USED FOR COLOR OF LIN
0074        CLR  R10          AND R10
0075  STALG C    @DELX,@DELY  COMPARE DELX,DELY
0076        JL   LSTEEP       IF DELX LOW, JUMP TO STEEP
0077  LEASY SRL  R4,1         ELSE SLOPE <45 , CUT R4 IN HALF
0078        MOV  R4,@HALFX    SAVE AT HALFX
0079        MOV  @DELX,R12    MOVE DELTA X TO R12
0080  LDOT1 MOV  @ERSFLG,R3   CHECK FOR ERASE STATUS
0081        JNE  LEASE        IF ERASE, JUMP
0082        MOVB @LINCLR,R9   PUT LINE DRAW COLOR IN R9
0083        BL   @PLOT        PLOT A POINT AT COORDINATES IN
R8,R7
0084        JMP  LDOT2        THEN JUMP AHEAD
0085  LEASE BL   @UNPLOT      ERASE A POINT AT R8,R7 LOCATION
0086  LDOT2 A    @DEEEYE,R7   ADD + OR - 1 TO R7
0087        A    @DELY,R10    ADD DELTA Y TO R10
0088        C    R10,@HALFX   COMPARE TO HALF OF X
0089        JLT  LDOT3        IF LESS, JUMP
0090        JEQ  LDOT3        OR IF EQUAL, JUMP
0091        S    @DELX,R10    SUBTRACT DELTA X FROM R10
0092        A    @ESSEYE,R8   ADD PLUS OR MINUS 1 TO R8
0093  LDOT3 DEC  R12          DECREMENT DELTA X IN R12
0094        JGT  LDOT1        IF POSITIVE, REPEAT
0095        JEQ  LDOT1        OR IF EQUAL, REPEAT
0096        JMP  LDREX        ELSE LINE FINISHED
0097  LSTEEP SRA R3,1         CUT R3 (DELTA Y) IN HALF
0098        MOV  R3,@HALFY    SAVE AT HALFY
0099        MOV  @DELY,R12    GET DELY INTO R12
0100  LDOT4 MOV  @ERSFLG,R3   CHECK FOR ERASE
0101        JNE  LSTPE        IF ERASE, JUMP AHEAD
0102        MOVB @LINCLR,R9   ELSE PUT LINE DRAW COLOR IN R9
0103        BL   @PLOT        PLOT ONE POINT
0104        JMP  LDOT5        THEN JUMP AHEAD
0105  LSTPE BL   @UNPLOT      ELSE "UNPLOT" TO ERASE A POINT
0106  LDOT5 A    @ESSEYE,R8   ADD + OR - 1 TO R8
0107        A    @DELX,R10    ADD DELTA X TO R10
0108        C    R10,@HALFY   COMPARE TO HALF OF DELTA Y
0109        JLT  LDOT6        IF LESS, JUMP AHEAD
0110        JEQ  LDOT6        IF EQUAL, JUMP AHEAD
0111        S    @DELY,R10    SUTRACT DELTA Y FROM R10
0112        A    @DEEEYE,R7   ADD DEEEYE TO R7
0113  LDOT6 DEC  R12          DEC COUNT OF POINTS
0114        JGT  LDOT4        IF POSITIVE, REPEAT
0115        JEQ  LDOT4        OR IF EQUAL, REPEAT
0116  LDREX  (END OF LINE DRAWING PROCESS)
0117        B    @KJSCAN      RETURN TO "NORMAL" DRAWING MODE
0118  *
0119  * SUBROUTINES PLOT AND UNPLOT - TO DRAW OR ERASE
0120  * SELECTED PIXEL POSITION
0121  *
0122  * FOLLOWING WRITES ONE PIXEL TO SCREEN AT LOCATION POINT-
ED BY
0123  * R8 (DOT ROW) AND R7 (DOT COLUMN)
0124  *
0125  PLOT   MOV  R7,R3       MOVE DOT COLUMN TO R3
0126        MOV  R8,R4        AND DOT ROW TO R4
0127        MOV  R4,R5        DOT ROW ALSO IN R5
0128        ANDI R5,7         R5 HAS DOT ROW MODULO 8
0129        SZC  R5,R4        SO DOES R4
0130        SLA  R4,5         MULTIPLY R4 BY 32
0131        A    R5,R4        ADD R5, SO R4 HAS DR MOD. 8 * 32 +
DR MOD 8
0132        MOV  R3,R0        MOVE DOT COL TO R0
0133        ANDI R0,>FFF8     R0 HAS DC - DC MOD 8
0134        S    R0,R3        R3 HAS DC MOD 8
0135        A    R4,R0        ADD R4
0136        SWPB R0           SWAP BYTES
0137        MOVB R0,@>8C02    WRITE LOW ADDRESS BYTE
0138        SWPB R0           SWAP
0139        MOVB R0,@>8C02    WRITE HIGH ADDRESS BYTE
0140        NOP               WASTE TIME
```

(See Page 9)

THE ART OF ASSEMBLY—

(Continued from Page 8)

```
0141        MOVB @>8800,R1    READ THE BYTE
0142  PLOTF  SOCB @M(R3),R1    OVERLAY MASK FROM TABLE M
0143  PLOTF0 ORI  R0,>4000     SET THE 4000 BIT IN R0
0144        SWPB R0           SWAP
0145        MOVB R0,@>8C02    WRITE LOW BYTE OF ADDRESS
0146        SWPB R0           SWAP
0147        MOVB R0,@>8C02    WRITE HIGH BYTE OF ADDRESS
0148        NOP               WASTE TIME
0149        MOVB R1,@>8C00    WRITE MODIFIED BYTE BACK TO VDP
0150        MOV  R9,R9        IS COLOR TO BE SET?
0151        JEQ  PLOTX        IF NOT, JUMP AHEAD
0152        ANDI R0,>3FFF     STRIP OFF "4" FROM R0
0153        AI   R0,>2000     ADD >2000 TO POINT AT COLOR TABLE
ENTRY
0154        BLWP @VSBR        READ THAT BYTE INTO R1
0155        MOVB R1,R2        MOVE THE BYTE TO R2
0156        ANDI R2,>F000     STRIP ALL BUT LEFT NYBBLE
0157        CB   R2,R9        COMPARE TO LEFT BYTE R9
0158        JEQ  PLOTX        IF EQUAL, COLOR ALREADY SET
0159        ANDI R1,>0F00     ELSE STRIP OFF LEFT NYBBLE R1
0160        AB   R9,R1        REPLACE WITH LEFT NYBBLE R9
0161        BLWP @VSBW        THEN WRITE COLOR BYTE BACK
0162  PLOTX  RT               RETURN
0163  *
0164  * FOLLOWING ERASES ONE PIXEL AT DOT ROW IN R8,
0165  * DOT-COLUMN IN R7
0166  *
0167  UNPLOT MOV  R7,R3       MOVE DOT COLUMN TO R3
0168        MOV  R8,R4        AND DOT ROW TO R4
0169        MOV  R4,R5        DOT ROW ALSO IN R5
0170        ANDI R5,7         R5 HAS DOT ROW MODULO 8
0171        SZC  R5,R4        SO DOES R4
0172        SLA  R4,5         MULTIPLY R4 BY 32
0173        A    R5,R4        ADD R5, SO R4 HAS DR MOD. 8 * 32 +
DR MOD 8
0174        MOV  R3,R0        MOVE DOT COL TO R0
0175        ANDI R0,>FFF8     R0 HAS DC - DC MOD 8
0176        S    R0,R3        R3 HAS DC MOD 8
0177        A    R4,R0        ADD R4
0178        SWPB R0           SWAP BYTES
0179        MOVB R0,@>8C02    WRITE LOW ADDRESS BYTE
0180        SWPB R0           SWAP
0181        MOVB R0,@>8C02    WRITE HIGH ADDRESS BYTE
0182        NOP               WASTE TIME
0183        MOVB @>8800,R1    READ THE BYTE
0184        INV  R1           INVERT ALL BITS IN R1
0185        SOCB @M(R3),R1    OVERLAY MASK FROM TABLE M
0186        INV  R1           RE-INVERT WITH ONE BIT CHANGED
0187        ORI  R0,>4000     SET THE 4000 BIT IN R0
0188        SWPB R0           SWAP
0189        MOVB R0,@>8C02    WRITE LOW BYTE OF ADDRESS
0190        SWPB R0           SWAP
0191        MOVB R0,@>8C02    WRITE HIGH BYTE OF ADDRESS
0192        NOP               WASTE TIME
0193        MOVB R1,@>8C00    WRITE MODIFIED BYTE BACK TO VDP
0194        RT
0195  *
0196  * DATA SECTION
0197  *
0198  M     DATA >8040,>2010,>0804,>0201  MASK DATA
0199  X1    DATA 0            STORAGE FOR START X
0200  X2    DATA 0            STORAGE FOR END X
0201  Y1    DATA 0            STORAGE FOR START Y
0202  Y2    DATA 0            STORAGE FOR END Y
0203  DELX  DATA 0            STORAGE FOR DELTA X
0204  DELY  DATA 0            STORAGE FOR DELTA Y
0205  HALFX DATA 0            HALF VALUE DELTA X
0206  HALFY DATA 0            HALF VALUE DELTA Y
0207  DEEEYE DATA 0           STORES SIGN DELTA X
0208  ESSEYE DATA 0           STORES SIGN DELTA Y
0209  ERSFLG DATA 0           ERASE MODE OFF - NON ZERO MAKES
ERASE ON
0210  LINCLR BYTE >10         DEFAULT COLOR - BLACK
0211  *
```

MYARC ADVANCED BASIC

Labeler program begets a sleeve printer

By JIM UZZELL
©1995 DDI Software

This month's program was inevitable. When you create a labeler program, published in January, the next logical step is to use that code and add to it and make it into a sleeve. This sleeve program will allow you to add comments to the lower portion of the sleeve, where the sliding door is, at the time the sleeve is being printed.

SLEEVE

```
100 !3 1/2" SLEEVE
110 !DDI SOFTWARE
120 !(C) 1995
130 CALL GRAPHICS(3,3)
140 F$=".    | |    " :: G$=
"   | |    ." :: C$="######
#### " :: DIM B$(135),Z$(32)
,B(32,32),M$(18) :: Y$="N"
150 E$=RPT$("-",47)
160 F1$=RPT$(" ",9)&"|"&RPT$
(" ",59)&"|"
170 H$=SEG$(F$,1,8)&"_______
___________DDI SOFTWARE_(C)1
995_____________________|
 |    ."
180 GOSUB 720
190 XT=135 :: DISPLAY AT(23,
1)BEEP :"WHICH DRIVE i.e DSK
1."
200 ACCEPT AT(24,1):D$
202 DISPLAY AT(23,1):"COMMEN
TS Y/N  N" :: ACCEPT AT(23,1
5)SIZE(-1):Y$
210 OPEN #1:D$,INPUT ,RELATI
VE,INTERNAL
220 INPUT #1:A$,Z1,Z2,Z3
230 I=I+1
240 INPUT #1:B$(I)
250 IF LEN(B$(I))=0 OR EOF(1
) THEN 270
260 GOTO 230
270 I=I-1
280 CLOSE #1
290 OPEN #1:"PIO",VARIABLE 1
36
310 PRINT #1:CHR$(15);CHR$(2
7);CHR$(65);CHR$(5);E$&SEG$(
E$,1,32)
320 PRINT #1:F$&CHR$(27)&CHR
$(71)&"VOL "&A$;TAB(36);DATE
$ &CHR$(27)&CHR$(72);TAB(72)
;G$
330 PRINT #1:CHR$(27)&CHR$(6
5)&CHR$(9)&F$&RPT$(" ",55);G
```

(See Page 10)

SLEEVE—

(Continued from Page 97)

```
$;
340 PRINT #1:CHR$(27)&CHR$(8
3)&CHR$(0)&CHR$(27)&CHR$(65)
&CHR$(5)
350 PRINT #1:F$&"Free: ";STR
$(Z3)&RPT$(" ",5-LEN(STR$(Z3
)));"    Used: ";STR$(Z2-Z3)
&RPT$(" ",5-LEN(STR$(Z2-Z3))
);RPT$(" ",29);G$
360 FOR X=1 TO XT
370 IF X=1 OR X=6 OR X=11 OR
 X=16 OR X=21 OR X=26 OR X=3
1 OR X=36 OR X=41 OR X=46 OR
 X=51 OR X=56 OR X=61 OR X=6
6 OR X=71 THEN PRINT #1:F$;
:: GOTO 400
380 IF X=76 OR X=81 OR X=86
OR X=91 OR X=96 OR X=101 OR
X=106 OR X=111 OR X=116 OR X
=121 THEN PRINT #1:F$;
390 IF X=126 OR X=131 THEN P
RINT #1:F$;
400 PRINT #1,USING C$:B$(X);
410 IF X=5 OR X=10 OR X=15 O
R X=20 OR X=25 OR X=30 OR X=
35 OR X=40 OR X=45 OR X=50 O
R X=55 OR X=60 OR X=65 OR X=
70 OR X=75 THEN PRINT #1:G$
:: GOTO 440
420 IF X=80 OR X=85 OR X=90
OR X=95 OR X=100 OR X=105 OR
 X=110 OR X=115 OR X=120 OR
X=125 THEN PRINT #1:G$
430 IF X=130 OR X=135 THEN P
RINT #1:G$
440 NEXT X
450 PRINT #1:H$
460 IF Y$="Y" THEN 580
470 FOR X=1 TO 72
490 IF X<20 THEN PRINT #1:".
    | |"&RPT$(" ",62)&"| |
   ." :: GOTO 520
500 IF X=20 THEN PRINT #1:"_
____|_|"&RPT$("_",62)&"|_|__
____" :: GOTO 520
510 IF X=22 THEN PRINT #1:"
       |"&RPT$("_",59)&"|"
ELSE PRINT #1:F1$
520 NEXT X :: PRINT #1:"
     |"&RPT$("-",59)&"|"
530 PRINT #1:CHR$(12);CHR$(2
7);"@" :: CLOSE #1
540 DISPLAY AT(23,1):"ANOTHE
R DISK  Y/N" :: DISPLAY AT(2
4,1):" "
550 CALL KEY(0,K,S) :: IF S<
1 THEN 550
560 IF K=78 OR K=110 THEN 57
0 ELSE I=0 :: CALL MEMSET(B$
(),RPT$(" ",28)) :: CALL MEMSET(M$(),
"") :: CALL GRAPHICS(3,3) ::
 GOSUB 1700 :: GOTO 190
570 CALL RESETPLT :: CLS ::
END
580 CALL GRAPHICS(4)
590 ROW=1 :: COL=1 :: ULIM=1
 :: LLIM=18 :: ELMT=1
600 DISPLAY AT(21,1):"Up and
 Down Arrows active":"Max 18
 lines":"MAX 60 CHARACTERS"
610 ACCEPT AT(ROW,COL)SIZE(-
60):M$(ELMT)
620 IF TERMCHAR=10 AND ROW<L
LIM THEN ROW=ROW+1 :: ELMT=R
OW :: GOTO 610
630 IF TERMCHAR=11 AND ROW>U
LIM THEN ROW=ROW-1 :: ELMT=R
OW :: GOTO 610
640 IF TERMCHAR=13 AND ROW>=
LLIM THEN 670
650 IF TERMCHAR=13 THEN ROW=
ROW+1 :: ELMT=ROW
660 GOTO 610
670 DISPLAY AT(19,1):"FINISH
ED  Y/N " :: CALL KEY(0,K,S)
 :: IF S<1 THEN 670
680 IF K=78 OR K=110 THEN DI
SPLAY AT(19,1):"" :: GOTO 59
0
685 DISPLAY AT(19,1):"":"":"
":"":""
690 PRINT #1:".    | |"&RPT$
(" ",62)&"| |     ." :: FOR
X=1 TO 71
700 IF X<19 THEN PRINT #1:".
    | |"&" "&M$(X)&RPT$(" ",
61-LEN(M$(X)))&SEG$(G$,4,9)
:: GOTO 714
710 IF X=19 THEN PRINT #1:"_
____|_|"&RPT$("_",62)&"|_|__
____" :: GOTO 714
712 IF X=21 THEN PRINT #1:"
       |"&RPT$("_",59)&"|"
ELSE PRINT #1:F1$
714 NEXT X :: PRINT #1:"
     |"&RPT$("-",59)&"|" ::
GOTO 530
720 CALL PALETTE(1,7,7,7) ::
 CALL PALETTE(13,8,3,7) :: C
ALL PALETTE(9,7,5,4) :: C
 PALETTE(11,3,8,8)
730 !Converted by ASM2MYB fr
om DDI SOFTWARE
740 CALL INIT750 CALL LOAD(-
8352,83,76,69
,69,86,69,36,248)
760 CALL LOAD(8194,44,224,22
3,96)
770 CALL LOAD(9460,0,6,36,78
,200,11,36,246,2,224,240,0,2
,4,0,0,2,0,0,14,192,96)
780 CALL LOAD(9482,37,94,192
,160,37,88,193,96,37,92,193,
160,37,90,2,7,37,96,208,247,
11,131)
790 CALL LOAD(9504,208,247,4
4,32,36,244,16,1,208,247,6,1
95,11,67,44,32,36,244,5,129,
129,65)
800 CALL LOAD(9526,19,7,11,1
95,44,32,36,244,5,129,129,65
,19,1,16,241,192,96,37,94,5,
130)
810 CALL LOAD(9548,129,130,1
9,1,16,235,194,224,36,240,
91,0,82,0,130,1,52,0,228,14,
238)
820 CALL LOAD(9570,238,238,2
38,238,238,238,238,238,238,2
38,238,238,238,238,238,238,2
38,238,238,238,238,238,238)
830 CALL LOAD(9592,238,238,2
38,238,238,238,238,238,238,2
38,238,238,238,238,238,238,2
38,238,238,238,238,238,238)
840 CALL LOAD(9614,238,238,2
38,238,238,238,238,238,238,2
38,238,238,238,238,238,238,2
38,238,238,238,238,238,238)
850 CALL LOAD(9636,238,238,2
38,238,238,238,238,238,238,2
38,238,238,238,238,17,238,23
8,238,238,238,238,238)
860 CALL LOAD(9658,238,238,2
38,238,238,238,238,238,238,2
38,238,238,238,238,238,238,2
38,238,238,238,238,238,238)
870 CALL LOAD(9680,238,238,2
38,238,238,238,238,17,238,238,17
,238,238,238,238,238,238,
,238,238,238,238,238)
880 CALL LOAD(9702,238,238,2
38,238,238,238,238,238,238,2
```

(See Page 11)

SLEEVE—

(Continued from Page 10)

```
,238,238,238,238,238,238,2
38,238,238,238,238,238)
890 CALL LOAD(9724,238,238,1
7,238,238,238,238,238,238,23
8,238,238,238,238,238,238,23
8,238,238,238,238,238)
900 CALL LOAD(9746,238,238,2
38,238,238,238,238,238,238,2
38,238,238,238,238,238,238,2
38,238,238,238,238,238)
910 CALL LOAD(9768,238,238,2
38,238,238,238,238,238,238,2
38,238,238,238,238,238,238,2
38,238,238,238,238,238)
920 CALL LOAD(9790,238,238,2
38,238,238,238,238,238,238,2
38,238,238,238,238,238,238,2
38,238,238,238,238,238)
930 CALL LOAD(9812,238,238,2
38,238,238,238,238,238,238,2
38,238,238,238,238,238,238,2
38,238,238,238,238,238)
940 CALL LOAD(9834,238,238,2
38,238,238,238,238,238,238,2
38,238,238,238,238,238,238,2
238,238,238,238,238)
950 CALL LOAD(9856,238,238,2
38,238,238,238,238,238,238,2
38,238,238,238,238,238,238,2
38,238,238,238,238,238)
960 CALL LOAD(9878,238,238,2
38,238,238,238,238,238,238,2
38,238,238,238,238,238,238,2
38,238,238,238,238,238)
970 CALL LOAD(9900,238,238,2
38,238,238,238,238,238,238,2
38,238,238,238,238,238,238,2
38,238,238,238,238,238)
980 CALL LOAD(9922,238,238,2
38,238,238,238,238,238,238,2
38,238,238,238,238,238,238,2
38,238,153,153,158,238)
990 CALL LOAD(9944,238,153,1
53,238,238,153,153,238,153,1
58,233,158,238,238,238,238,2
38,238,238,238,238,238)
1000 CALL LOAD(9966,238,238,
238,238,238,238,238,238,238,
238,238,238,238,238,233,158,
153,238,238,233,158,238)
1010 CALL LOAD(9988,233,158,
153,158,233,158,233,158,238,
238,238,238,238,238,238,238,
238,238,238,238,238,238)
1020 CALL LOAD(10010,238,238
,238,238,238,238,238,238,238
,238,233,158,233,158,238,233
,158,238,233,153,238,238)
1030 CALL LOAD(10032,233,158
,153,238,238,238,238,238,238
,238,238,238,238,238,238,238
,238,238,238,238,238,238)
1040 CALL LOAD(10054,238,238
,238,238,238,238,233,158,233
,158,238,233,158,238,238,153
,158,238,233,153,158,238)
1050 CALL LOAD(10076,238,238
,238,238,238,238,238,238,238
,238,238,238,238,238,238,238
,238,238,238,238,238,238)
1060 CALL LOAD(10098,238,238
,233,158,233,158,238,233,158
,238,238,238,153,158,233,158
,153,238,238,238,238,238)
1070 CALL LOAD(10120,238,238
,238,238,238,238,238,238,238
,238,238,238,238,238,238,238
,238,238,238,238,233,158)
1080 CALL LOAD(10142,153,238
,238,233,158,238,233,158,233
,158,233,158,233,158,238,238
,238,238,238,238,238,238)
1090 CALL LOAD(10164,238,238
,238,238,238,238,238,238,238
,238,238,238,238,238,238,238
,153,153,158,238,238,153)
1100 CALL LOAD(10186,153,238
,238,153,153,238,153,158,233
,158,238,238,238,238,238,238
,238,238,238,238,238,238)
1110 CALL LOAD(10208,238,238
,238,238,238,238,238,238,238
,238,238,238,238,238,238,238
,238,238,238,238,238,238)
1120 CALL LOAD(10230,238,238
,238,238,238,238,238,238,238
,238,238,238,238,238,238,238
,238,238,238,238,238,238)
1130 CALL LOAD(10252,238,238
,238,238,238,238,238,238,238
,238,238,238,238,238,238,238
,238,238,238,238,238,238)
1140 CALL LOAD(10274,238,238
,238,238,238,238,238,238,238
,238,238,238,238,238,238,238
,238,238,238,238,238,238)
1150 CALL LOAD(10296,238,238
,238,238,238,238,238,238,238
,238,238,238,238,238,238,238
,238,238,238,238,238,238)
1160 CALL LOAD(10318,238,238
,238,238,238,238,238,238,238
,238,238,238,238,238,238,238
,238,238,238,238,238,238)
1170 CALL LOAD(10340,238,238
,238,238,238,238,238,238,238
,238,238,238,238,238,238,238
,238,238,238,238,238,238)
1180 CALL LOAD(10362,238,238
,238,238,238,238,238,238,238
,238,238,238,238,238,238,153
,153,238,233,158,238,238)
1190 CALL LOAD(10384,233,153
,153,158,233,153,153,158,233
,158,233,158,233,153,153,158
,238,238,238,238,238,238)
1200 CALL LOAD(10406,238,238
,238,238,238,238,238,238,238
,238,233,158,233,158,233,158
,238,238,233,158,238,238)
1210 CALL LOAD(10428,233,158
,238,238,233,158,233,158,233
,158,238,238,238,238,238,238
,238,238,238,238,238,238)
1220 CALL LOAD(10450,238,238
,238,238,238,238,233,153,238
,238,233,158,238,238,233,158
,238,238,233,158,238,238)
1230 CALL LOAD(10472,233,158
,233,158,233,158,238,238,238
,238,238,238,238,238,238,238
,238,238,238,238,238,238)
1240 CALL LOAD(10494,238,238
,238,153,158,238,233,158,238
,238,233,153,158,238,233,153
,158,238,233,158,233,158)
1250 CALL LOAD(10516,233,153
,158,238,238,238,238,238,238
,238,238,238,238,238,238,238
,238,238,238,238,238,238)
1260 CALL LOAD(10538,153,158
,233,158,238,238,233,158,238
,238,233,158,238,238,233,158
,233,158,233,158,238,238)
1270 CALL LOAD(10560,238,238
,238,238,238,238,238,238,238
,238,238,238,238,238,238,238
,233,158,233,158,233,158)
1280 CALL LOAD(10582,238,238
,233,158,238,238,233,158,238
,238,238,153,153,238,233,158
,238,238,238,238,238,238)
1290 CALL LOAD(10604,238,238
```

(See Page 12)

SLEEVE—

(Continued from Page 11)

```
,238,238,238,238,238,238,238
,238,238,238,238,153,153,238
,233,153,153,158,233,153)
1300 CALL LOAD(10626,153,158
,233,153,153,158,238,233,158
,238,233,153,153,158,238,238
,238,238,238,238,238,238)
1310 CALL LOAD(10648,238,238
,238,238,238,238,238,238,238
,238,238,238,238,238,238,238
,238,238,238,238,238,238)
1320 CALL LOAD(10670,238,238
,238,238,238,238,238,238,238
,238,238,238,238,238,238,238
,238,238,238,238,238,238)
1330 CALL LOAD(10692,238,238
,238,238,238,238,238,238,238
,238,238,238,238,238,238,238
,238,238,238,238,238,238)
1340 CALL LOAD(10714,238,238
,238,238,238,238,238,238,238
,238,238,238,238,238,238,238
,238,238,238,238,238,238)
1350 CALL LOAD(10736,238,238
,238,238,238,238,238,238,238
,238,238,238,238,238,238,238
,238,238,238,238,238,238)
1360 CALL LOAD(10758,238,238
,238,238,238,238,238,238,238
,238,238,238,238,238,238,238
,238,238,238,238,238,238)
1370 CALL LOAD(10780,238,238
,238,238,238,238,238,238,238
,238,238,238,238,238,238,238
,238,238,238,238,238,238)
1380 CALL LOAD(10802,238,238
,238,238,238,238,238,238,238
,238,238,238,238,238,238,238
,238,238,238,238,238,238)
1390 CALL LOAD(10824,238,238
,238,238,238,238,238,238,238
,238,238,238,238,238,238,238
,238,238,238,238,238,238)
1400 CALL LOAD(10846,238,238
,238,238,238,238,238,238,238
,238,238,238,238,238,238,238
,238,238,238,238,238,238)
1410 CALL LOAD(10868,238,238
,238,238,238,238,238,238,238
,238,238,238,238,238,238,238
,238,238,238,238,238,238)
1420 CALL LOAD(10890,238,238
,238,238,238,238,238,238,238
,238,238,238,238,238,238,238
,238,238,238,238,238,238)
1430 CALL LOAD(10912,238,238
,238,238,238,238,238,238,238
,238,238,238,238,238,238,238
,238,238,238,238,238,225)
1440 CALL LOAD(10934,17,17,1
7,17,17,17,17,17,17,17,17,17
,17,17,17,17,17,17,17,17,17,
17)
1450 CALL LOAD(10956,17,17,1
7,17,17,17,238,238,238,238,2
38,238,238,238,238,238,238,2
25,17,17,17,17)
1460 CALL LOAD(10978,17,17,3
0,238,238,238,238,238,17,17,
17,17,17,17,17,17,17,17,17,1
7,17,17)
1470 CALL LOAD(11000,17,17,2
38,238,238,238,238,238,238,2
38,238,238,238,225,17,17,17,
17,17,17,30,238)
1480 CALL LOAD(11022,238,238
,238,238,17,17,17,17,17,17,1
7,17,17,17,17,17,17,17,17,17
,238,238)
1490 CALL LOAD(11044,238,238
,238,238,238,238,238,238,238
,225,17,17,17,17,17,17,30,23
8,238,238,238,238)
1500 CALL LOAD(11066,17,17,1
7,17,17,17,17,17,17,17,17,17
,17,17,17,17,238,238,238,238
,238,238)
1510 CALL LOAD(11088,238,238
,225,30,30,225,17,17,17,17,1
7,17,30,238,238,238,238,238,
17,17,17,17)
1520 CALL LOAD(11110,17,17,1
7,17,17,17,17,17,17,17,17,17
,238,238,238,238,238,238,238
,238,30,30)
1530 CALL LOAD(11132,30,225,
17,17,17,17,17,17,30,238,238
,238,238,238,17,17,17,17,17,
17,17,17)
1540 CALL LOAD(11154,17,17,1
7,17,17,17,17,17,238,238,238
,238,238,238,238,225,238,30,
30,225,17,17)
1550 CALL LOAD(11176,17,17,1
7,17,30,238,238,238,238,238,
17,17,17,17,17,17,17,17,17,1
7,17,17)
1560 CALL LOAD(11198,17,17,1
7,17,238,238,238,238,238,238
,238,225,238,17,30,225,17,17
,17,17,17,17)
1570 CALL LOAD(11220,30,238,
238,238,238,238,17,17,17,17,
17,17,17,17,17,17,17,17,17,1
7,17,17)
1580 CALL LOAD(11242,238,238
,238,238,238,238,238,238,225,238
,30,30,225,17,17,17,17,17,17
,30,238,238,238)
1590 CALL LOAD(11264,238,238
,17,17,17,17,17,17,17,17,17,
17,17,17,17,17,17,17,17,238,238
,238,238)
1600 CALL LOAD(11286,238,238
,238,238,30,30,30,225,17,17,
17,17,17,17,30,238,238,238,2
38,238,17,17)
1610 CALL LOAD(11308,17,17,1
7,17,17,17,17,17,17,17,17,17,17
,17,17,238,238,238,238,238,2
38,238,238)
1620 CALL LOAD(11330,225,30,
30,225,17,17,17,17,17,17,30,
238,238,238,238,238,17,17,17
,17,17,17)
1630 CALL LOAD(11352,17,17,1
7,17,17,17,17,17,17,17,238,2
38,238,238,238,238,238,238,2
38,238,238,225)
1640 CALL LOAD(11374,17,17,1
7,17,17,17,17,17,17,17,17,17
,17,17,17,17,17,17,17,17,17,
17)
1650 CALL LOAD(11396,17,17,1
7,17,17,17,238,238,238,238,2
38,238,238,238,238,238,238,2
25,17,17,17,17)
1660 CALL LOAD(11418,17,17,1
7,17,17,17,17,17,17,17,17,17
,17,17,17,17,17,17,17,17,17,
17)
1670 CALL LOAD(11440,17,17,2
38,238,238,238,238,238,14,23
8,238,238,238,225,17,17,17,1
7,17,17,17,17)
1680 CALL LOAD(11462,17,17,1
7,17,17,17,17,17,17,17,17,17
,17,17,17,17,17,17,17,17,238
,238)
1690 CALL LOAD(11484,238,2
,238,224,69,69)
1700 CALL LINK("SLEEVE")
1710 RETURN
```

EXTENDED BASIC

INTSCRAM, INTHANG, INTMENU complete work with relative files

By LUCIE DORAIS

This is the second of a two-part series. Last month's installment contained several program errors. See User Notes for corrections.

We start with INTSCRAM; it uses the two user-def subs from last month, SUB LANG and SUB SHOW (lines 1010-1110). Lines 120-170 initialize the program (lines 130-140 are almost identical as in INTLEARN, but with two more characters: 37 shows a real "#" on the screen, and 136 will be a small green square (color set 14 added to CALL COLOR). The instructions for the game are a subprogram, lines 410-450.

First thing to do before we play is open our INTWORD file as INPUT (to prevent accidental writing to it), line 190. Then CALL LANG to display the first two letters of the five languages involved, as in INTLEARN. Some magenta lines are then displayed. Line 230 picks a record R at random; the DONE() array makes sure that you never get the same problem more than once during a session. Record R is then read into memory as R$, and the English word put at the top of the screen. The four translations are then read in turn into A$, their trailing spaces (if any) removed, and the foreign word is kept into the array GA$(), the good answer that will be used to check against your own.

The word is then sent to the user-def sub SCRAMBLE, our trusted shuffling routine from last fall's game CANFIELD. If A$ as sent back is scrambled in such a way that it is not (IF A$=GA$(X): bound to happen with small words, the routine is called again. When the word is really scrambled, it is displayed in the left half of the screen besides the language initials.

UNSCRAMBLE THE WORDS

To play, you can unscramble all four foreign words, or only the ones you want. Press the first letter of the language (menu in line 300), and Tex will accept your answer in the right half of the screen, on the same line as the scrambled word. The array U$() is used to keep track of the languages already done.

If your answer ANS$ equals the good one kept in GA$(), Tex CALLs YES to sound appropriate trumpets, and your points PT are calculated for this foreign word. BONUS keeps track of the number of good answers, because you get 20 extra points if you have all four right. If your answer is not good, line 370 will produce a scratching noise and your number of TRYs will be incremented. When it reaches four, a green square flag will appear at the left of the good answer (so you can learn it!), which is CALL SHOWed by line 380. This good answer is also displayed when you get the translation right, because CALL SHOW puts the accents at their proper place over the word. If you have absolutely no clue of the good answer, and want to learn it, enter a period as your answer. The other keys you can press from the playing menu are "N" for a new word (line 380), and "Q" to quit (line 400).

HANGMAN WITHOUT A VICTIM

The second game, INTHANG, is Hangman, but without a victim (some foreign words being only two letters long made it hard to draw a little man). You will notice that lines 130-140 and 180-190 are exactly the same as in INTSCRAM, and that line 120 is slightly different. Here again I used the user-def SUB SHOW (and added two parameters to it, W$ and AW$, see line 1050), but not the SUB LANG, so it was deleted from the file I MERGEd in memory before I typed the rest of the program.

Line 200 defines characters 132-135 as one big 64-character string, a sprite MAGNIFYed four times to hide the solution. The BOX called by line 210 (to frame the square where you will enter your character or menu choice) is found in line 1120 — it has no corners but you can add them if you wish. For each problem, the word to guess is replaced by a line of green squares (character 136) and you are given a clue, for example "FRENCH for LEG." You can enter the characters one by one, guess the word whenever you feel confident enough (only one try though), or you can give up at any point. In each case the correct solution will be shown.

We first initialize flags GS (guess a complete word) and ACC (the word contains an accent), then the whole alphabet and the six accents are displayed at the top of the screen. Line 250 picks a problem at random: first a record R, then a word W from that record (remember, each record contains an English word and its four translations). The DONE(,) array is here bi-dimensional, so you can have a total of 400 problems, and never do the same one twice. The random record R is read into memory and the random foreign word put into P$. If its first character is the accent flag "*" (ASCII 42), the ACC flag it set.

In this game, it is the strings W$ and AW$ from SUB SHOW that will be used as check controls (therefore their addition as parameters). Line 270 displays the menu. Line 280 colors the mask sprites green to hide the solution displayed by CALL SHOW and framed by CALL BOX. Line 290 deletes any trailing spaces from the foreign word W$ sent back to us from the CALL SHOW sub, since L=LEN(W$) is used to derive the value of column C (to center the word and its box) and the number of tries allowed (total of letters and accents in the word plus three more tries).

In line 320, the hidden word is displayed (green squares) and the strings A$ and AA$, that will hold your answer as you enter it, are emptied. P$ will be reused later, so we empty it too (save bytes...). Line 330 displays the clue mentioned above and line 340 tells you how many tries you have left. If none, CALL NO will sound some sad notes before sending you to the solution and a new menu. You enter one character at a time, in line 350. Here you can also enter a period to give up (more sad notes) or a "/" to warn

(See Page 14)

INTSCRAM, INTHANG, INTMENU—

(Continued from Page 13)

Tex that you know the complete word. You can earn extra points by using that option, but only if your guess is right of course. Remember, only one try here.

Lines 370-380 deal with any character or accent that you enter. The SUB CHECK has many parameters:

- The solution W$/AW$, your answer up to now A$/AA$.
- The top row of alphabet/accents and their character column (to check and mark the character you have entered).
- The starting column for the hidden word (the green squares).
- The ASCII value A of the character you just entered.
- A flag F to tell Tex if you checked OK or not.

In the SUB, the flag is set to one, and the character you entered is rebuilt as C$ from its ASCII value (one less parameter to pass). The position of C$ at the top of the screen is CALL GCHARed. If it is a green square (character 136), you have already used that character — no harm done, except it costs you one try (flag F=1 will send you to line 440 for a beep upon returning from the sub). If the character is used for the first time, it is replaced by a green square in the alpha/accent rows at the top (line 1150). Then Tex looks for the presence of your character C$ in the word string W$, which is the word W$ or the accent string AW$, depending from the character you entered, of course. If not found, you exit from the sub. The flag F is still one, so off to line 440 (see above). If the character is found, line 1160 will add it to your answer, first on the screen, where it will replace a green square, then in your answer string A$ or AA$. Now the same letter might be present more than once, so line 1160 is repeated until Tex does not find the character any more in the word. When all is okay, the flag F is set to zero and back in the main program you are sent to line 410 to check your answer up to now with the good one.

Lines 390-400 are used when you decide to guess the complete word. The guessing flag GS is set to one, and your word is accepted on the screen (the green squares line) into A$. The negative SIZE makes sure that the letters you already entered will be included. If the word has an accent (Clue: the box framing the word has an extra row), Tex will now ask you to enter it into AA$. Here you need extra caution, as you need to position the accent EXACTLY over the letter it accents, the only way than Tex can tell you if you are right or wrong. Your accent answer is then padded with spaces because the good answer kept into AW$ (as sent back by SUB SHOW) does have them.

Whether you enter only one character or the complete word, Tex now checks how your answer(s) A$ (and optionally AA$) compare with the solutions W$ and AW$. If you don't have it right, you are sent back to enter another character but, if you have opted to guess the complete word (GS=1), you are given a beep and sent to the end of the problem: a boxed menu to try a N)ew word or Q)uit (lines 450-500). If you have the right answer(s), CALL YES will sound the same trumpets as INTSCRAM, give you points (extra ones for accent and complete guess) and tell you a well earned "BRAVO" (lines 420-430).

Finally, INTMENU puts everything together. In the three programs, "END" has been replaced with RUN "DSK1.INTMENU": in line 360 of INTLEARN, 400 of INTSCRAM, 500 of INTHANG. If you have only the INT set of programs on your diskette, you can call INTMENU as LOAD, and use RUN "DSK1.LOAD" in the three long programs; alternatively, type this small LOAD program:

```
100 RUN "DSK1.INTMENU"
140 RUN "DSK1.INTHANG" !198
150 RUN "DSK1.INTSCRAM" !031
160 END !139
```

INTSCRAM

```
100 ! ** INTSCRAM ** - L.Dor
ais/Ottawa UG/Apr 92 !109
110 !!131
120 CALL CLEAR :: CALL SCREE
N(16):: DIM DONE(100),U$(4),
GA$(4):: CALL MAGNIFY(2):: E
$=RPT$(" ",140):: RANDOMIZE
!224
130 A$="00000000" :: CALL CH
AR(39,A$&"10204",96,A$&"4020
1",94,A$&"205088",35,A$&"005
05",126,A$&"40A81",37,"00505
0F850F85050")! accents !1
140 CALL CHAR(128,"181818181
8181818",129,"000000FFFF",13
6,"00FCFCFCFCFCFCFC":: CALL
 COLOR(13,14,1,14,3,1)! line
s/squares !135
150 GOTO 170 :: ANS$,AW$,BON
US,K,P,PT,R,R$,S,S$,TRY,W$,X
 !203
160 CALL KEY :: CALL HCHAR :
: CALL VCHAR :: CALL SOUND :
: !@P- !058
170 S$="       " :: GOSUB 41
0 ! instructions !040
180 ! ** display screen ** !
033
190 CALL CLEAR :: OPEN #1:"D
SK1.INTWORDS",DISPLAY ,RELAT
IVE,FIXED 56,INPUT !080
200 CALL LANG(1,17):: DISPLA
Y AT(1,19):"PTS: 0" :: CALL
HCHAR(2,9,129,9):: CALL HCHA
R(2,21,129,10):: CALL HCHAR(
19,1,129,32)!168
210 ! ** scramble ** !212
220 BONUS=0 :: CALL VCHAR(
19,128,18):: DISPLAY AT(1,
):PT !161
230 R=INT(100*RND)+1 :: IF D
ONE(R)THEN 230 ELSE DONE(R)=
1 !101
240 INPUT #1,REC R:R$ :: DIS
PLAY AT(1,7)SIZE(-10):SEG$(R
$,1,10)! English word !078
250 FOR X=1 TO 4 :: A$=SEG$(
R$,11*X+1,10)! get foreign w
ords & scramble !040
260 P=POS(A$," ",1):: IF P T
HEN A$=SEG$(A$,1,P-1)! del t
railing spaces !049
270 GA$(X)=A$ :: U$(X)="" ::
 CALL SCRAMBLE(A$):: IF A$=G
A$(X)THEN 270 !005
280 DISPLAY AT(4*X+1,7)SIZE(
-10):A$ :: NEXT X !084
290 ! ** unscramble ** !183
300 DISPLAY AT(20,1):"PICK:"
:S$&"G)erman    N)ew word":S
$&"F)rench    Q)uit":S$&"I)t
alian":S$&"S)panish" !129
310 CALL KEY(0,K,S):: IF S=
 THEN 310 ELSE P=POS("GFIS.
",CHR$(K),1):: ON P+1 GOTO 3
10,320,320,320,320,390,400 !
```

(See Page 15)

INTSCRAM, INTHANG, INTMENU —

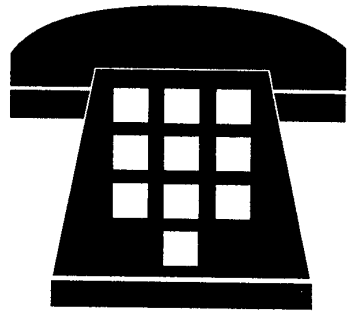
(Continued from Page 14)

```
133
320 IF U$(P)="*" THEN CALL S
OUND(150,200,5):: GOTO 310 E
LSE TRY=1 :: R=4*P+1 :: CALL
 HCHAR(R+1,21,129,10)!221
330 DISPLAY AT(20,1):"":"":"
Enter your answer, or a .":"
to see the translation.":"(A
ccent after letter, #=%)" !0
23
340 DISPLAY AT(20,19):"TRY";
TRY :: ACCEPT AT(R,19)SIZE(-
10):ANS$ !255
350 IF ANS$="" THEN 340 ELSE
 IF ANS$="." THEN TRY=3 :: G
OTO 370 !073
360 IF ANS$=GA$(P)THEN CALL
YES :: BONUS=BONUS+1 :: PT=P
T+2*LEN(ANS$)-20*(BONUS=4)::
 GOTO 380 !192
370 CALL SOUND(100,-5,5):: T
RY=TRY+1 :: IF TRY<4 THEN 34
0 ELSE CALL HCHAR(R,20,136)!
111
380 U$(P)="*" :: CALL SHOW(R
,19,R$,11*P+1):: CALL HCHAR(
R+1,21,32,10):: DISPLAY AT(1
,23):PT :: GOTO 300 !174
390 DISPLAY AT(20,1):E$ :: C
ALL HCHAR(1,9,32,10):: FOR X
=3 TO 17 :: DISPLAY AT(X,7):
"" :: NEXT X :: GOTO 220 ! n
ew word !021
400 CLOSE #1 :: RUN "DSK1.IN
TMENU" !246
410 DISPLAY AT(1,9):"UNSCRAM
BLE" :: CALL HCHAR(2,11,129,
10)!177
420 DISPLAY AT(5,2):"You are
 given an English":" word a
t random, with its":" four
translations in":" scramble
d form." !201
430 DISPLAY AT(10,2):"Pick a
 language and enter":" the
unscrambled word. You":" ha
ve three tries." !185
440 DISPLAY AT(15,2):"Points
 for good answer:": :"    2
* length of word": :"   + 2
 bonus if you get":"
 all four." !101
450 DISPLAY AT(24,9):"PRESS
A KEY" :: CALL KEY(0,K,S)::
IF S=0 THEN 450 ELSE RETURN
!244
1000 !@P+ ! user-def subs !1
51
1010 SUB LANG(R,C)!068
1020 A$="EN14GE11FR05IT03SP0
9" :: FOR X=1 TO 5 :: P=4*X-
3 :: B$=SEG$(A$,P,1):: C$=SE
G$(A$,P+1,1):: K=VAL(SEG$(A$
,P+2,2))!197
1030 CALL SPRITE(#X,ASC(B$),
K,R,C,#X+5,ASC(C$),K,R,C+16)
!116
1040 R=R+32 :: NEXT X :: SUB
END !236
1050 SUB SHOW(R,C,R$,P):: W$
=SEG$(R$,P,10):: AW$=RPT$("
",10):: IF P=1 THEN 1110 !21
5
1060 IF SEG$(R$,P-1,1)<>"*"
THEN 1110 ! accent flag? !24
9
1070 FOR X=1 TO 10 :: A$=SEG
$(W$,X,1):: IF POS("'` #~",A
$,1)=0 THEN 1100 ! find acce
nt !074
1080 W$=SEG$(W$,1,X-1)&SEG$(
W$,X+1,10)! rewrite word wit
hout accent !204
1090 AW$=SEG$(AW$,1,X-2)&A$&
SEG$(AW$,X+1,10)! accent str
ing !050
1100 NEXT X !238
1110 DISPLAY AT(R,C):W$ :: D
ISPLAY AT(R-1,C):AW$ :: SUBE
ND !151
1120 SUB SCRAMBLE(W$):: L=LE
N(W$)!235
1130 FOR X=1 TO L :: SH(X)=X
 :: NEXT X !001
1140 A$="" :: C=L :: FOR X=1
 TO L :: P=INT(RND*C)+1 :: P
W=SH(P)!119
1150 A$=A$&SEG$(W$,PW,1):: S
H(P)=SH(C):: C=C-1 :: NEXT X
 :: W$=A$ :: SUBEND !0381200
 SUB YES :: CALL SOUND(70,20
00,0):: CALL SOUND(100,1500,
0):: SUBEND !182
```

INTHANG

```
100 ! ** INTHANG ** - L.Dora
is/Ottawa UG/Apr 92 !021
```

(See Page 16)



Want to talk to someone at MICROpendium? You'll need to called between the hours of 9 a.m. and noon Saturdays. If you call at other times, you will probably get an answering machine. But don't let that bother you. We listen to the answering machine at least once a day and return calls as soon as possible, usually that day.

Call us at 512-255-1512

INTSCRAM, INTHANG, INTMENU —

(Continued from Page 15)

```
110 !!131
120 CALL CLEAR :: CALL SCREE
N(16):: DIM DONE(100,4),L$(4
):: CALL MAGNIFY(4):: E$=RPT
$(" ",168):: RANDOMIZE !202
130 A$="00000000" :: CALL CH
AR(39,A$&"10204",96,A$&"4020
1",94,A$&"205088",35,A$&"005
05",126,A$&"40A81",37,"00505
0F850F85050")! accents !116
140 CALL CHAR(128,"181818181
8181818",129,"000000FFFF",13
6,"00FCFCFCFCFCFCFC"):: CALL
 COLOR(13,14,1,14,3,1)! line
s/squares !135
150 GOTO 170 :: A,AA$,ACC,AW
$,C,C$,F,GS,K,L,P,P$,PT,R,R$
,S,TRY,W,W$ !240
160 CALL KEY :: CALL HCHAR :
: CALL SPRITE :: !@P- !041
170 L$(1)="GERMAN" :: L$(2)=
"FRENCH" :: L$(3)="ITALIAN"
:: L$(4)="SPANISH" !158
180 ! ** display screen ** !
033
190 CALL CLEAR :: OPEN #1:"D
SK1.INTWORDS",DISPLAY ,RELAT
IVE,FIXED 56,INPUT !080
200 A$=RPT$("F",16):: AA$=RP
T$("0",16):: CALL CHAR(132,A
$&AA$&A$&AA$)!059
210 DISPLAY AT(1,1):RPT$(CHR
$(132),112):: DISPLAY AT(16,
3):"SOLUTION       CHAR" ::
CALL BOX(15,26,1,1)!121
220 CALL SPRITE(#1,132,1,137
,25,#2,132,1,137,57,#3,132,1
,137,73)!063
230 ! ** get/display problem
 ** !001
240 GS,ACC=0 :: DISPLAY AT(2
,2)SIZE(-26):"ABCDEFGHIJKLMN
OPQRSTUVWXYZ" :: DISPLAY AT(
3,9)SIZE(-11):" ' `   # ~ "
!251
250 R=INT(100*RND)+1 :: W=IN
T(4*RND)+1 :: IF DONE(R,W)TH
EN 250 ELSE DONE(R,W)=1 !178
260 INPUT #1,REC R:R$ :: P$=
SEG$(R$,11*W,11):: IF ASC(P$
)=42 THEN ACC=1 !190
270 DISPLAY AT(20,18):"":"":
TAB(18);"% = #":"  Tries:";T
AB(18);". = give up":" Point
s:";PT;TAB(18);"/ = guess" !
047
280 CALL COLOR(#1,3,#2,3,#3,
3):: CALL SHOW(19,2,P$,2,W$,
AW$):: CALL BOX(17,4,10,2)!0
76
290 P=POS(W$," ",1):: IF P T
HEN W$=SEG$(W$,1,P-1)! del t
rail.spaces !067
300 L=LEN(W$):: C=INT((28-L)
/2)+2 :: AW$=SEG$(AW$,1,L)!0
89
310 TRY=L+3+ACC :: CALL BOX(
8-ACC,C,L,1+ACC)!066
320 CALL HCHAR(9,C,136,L)::
A$,AA$=RPT$(" ",L):: P$="" !
250
330 DISPLAY AT(12,5):L$(W)&"
 for "&SEG$(R$,1,10)! clue !
027
340 IF TRY>0 THEN DISPLAY AT
(23,9)SIZE(-4):TRY ELSE CALL
 NO :: GOTO 460 !189
350 ACCEPT AT(16,24)SIZE(1):
C$ :: IF C$="" THEN 350 !077
360 IF C$="." THEN CALL NO :
: GOTO 460 ELSE IF C$="/" TH
EN 390 !018
370 TRY=TRY-1 :: A=ASC(C$)::
 IF A<65 OR A>90 THEN 380 EL
SE CALL CHECK(W$,A$,2,A-61,C
,A,F):: IF F THEN 440 ELSE 4
10 !236
380 P=POS("' ` #~",C$,1):: IF
 P=0 THEN 440 ELSE CALL CHEC
K(AW$,AA$,3,2*P+10,C,A,F)::
IF F THEN 440 ELSE 410 !025
390 GS=1 :: ACCEPT AT(9,C-2)
SIZE(-L):A$ ! guess word !17
6
400 IF ACC THEN ACCEPT AT(8,
C-2)SIZE(-L):AA$ :: AA$=AA$&
RPT$(" ",L-LEN(AA$))!148
410 IF A$<>W$ OR AA$<>AW$ TH
EN IF GS THEN 440 ELSE 340 !
 good guess? !239
420 CALL YES :: PT=PT+5*L+5*
ACC+10*GS !065
430 DISPLAY AT(24,9)SIZE(-7)
:PT :: P$="! BRAVO !" :: GOT
O 460 !117
440 CALL SOUND(100,200,5)::
IF GS THEN 460 ELSE 340 ! wr
ong letter/word !109
450 ! ** end of problem ** !
200
460 CALL COLOR(#1,1,#2,1,#3,
1)! reveal solution !106
470 DISPLAY AT(20,18):P$:"":
TAB(18);"N)ew word":TAB(18);
"Q)uit" :: CALL BOX(21,20,9,
2)!006
480 CALL KEY(0,K,S):: IF S=0
 OR(K<>78 AND K<>81)THEN 480
 !007
490 IF K=78 THEN DISPLAY AT(
7,1):E$ :: GOTO 240 ! new wo
rd !202
500 CLOSE #1 :: RUN "DSK1.IN
TMENU" ! quit !092
1000 !@P+ ! * user-def subs
1050 SUB SHOW(R,C,R$,P,W$,AW
$):: W$=SEG$(R$,P,10):: AW$=
RPT$(" ",10):: IF P=1 THEN 1
110 !116
1060 IF SEG$(R$,P-1,1)<>"*"
THEN 1110 ! accent flag? !24
9
1070 FOR X=1 TO 10 :: A$=SEG
$(W$,X,1):: IF POS("' ` #~",A
$,1)=0 THEN 1100 ! find acc
nt !074
1080 W$=SEG$(W$,1,X-1)&SEG$(
W$,X+1,10)! rewrite word wit
hout accent !204
1090 AW$=SEG$(AW$,1,X-2)&A$&
SEG$(AW$,X+1,10)! accent str
ing !050
1100 NEXT X !238
1110 DISPLAY AT(R,C):W$ :: D
ISPLAY AT(R-1,C):AW$ :: SUBE
ND !151
1120 SUB BOX(R,C,W,H):: CALL
 HCHAR(R,C,129,W):: CALL HCH
AR(R+H+1,C,129,W):: CALL VCH
AR(R+1,C-1,128,H):: CALL VCH
AR(R+1,C+W,128,H):: SUBEND !
239
1130 SUB CHECK(W$,A$,R,CC,CW
,A,F):: F=1 :: C$=CHR$(A)!23
9
1140 CALL GCHAR(R,CC,K):: IF
 K=136 THEN SUBEXIT !144
1150 CALL HCHAR(R,CC,136)::
P=POS(W$,C$,1):: IF P=0 THEN
 SUBEXIT ! char. found? !07
1160 CALL HCHAR(11-R,CW-1+A
A):: A$=SEG$(A$,1,P-1)&C$&SE
G$(A$,P+1,10)!099
```

(See Page 17)

INTSCRAM, INTHANG, INTMENU —

(Continued from Page 16)

```
1170 P=POS(W$,C$,P+1):: IF P
>0 THEN 1160 !187
1180 F=0 :: SUBEND !039
1190 SUB NO :: CALL SOUND(70
,250,0):: CALL SOUND(100,150
,0):: SUBEND !004
1200 SUB YES :: CALL SOUND(7
0,2000,0):: CALL SOUND(100,1
500,0):: SUBEND !182
```

INTMENU

```
100 CALL DELSPRITE(ALL)!115
110 DISPLAY AT(3,1)ERASE ALL
:"INTERNATIONAL WORDS": : :"
1 - LEARN": :"2 - Play HANGM
AN": :"3 - Play UNSCRAMBLE":
 :"4 - QUIT" !203
120 CALL KEY(0,K,S):: IF S=0
 OR K<49 OR K>52 THEN 120 EL
SE ON K-48 GOTO 130,140,150,
160 !019
130 RUN "DSK1.INTLEARN" !027
```

TI speech

Simple to use with powerful capabilities

(This article was originally published in the March 1995 issue of On Cue, *the newsletter of the Computer Users of Erie.)*

By DON GRIM

The availability for speech on the TI computer is remarkable in many ways. The speech has a very clear human tone.

The equipment needed for speech is minor, consisting of a console (the TI99/4A), speech synthesizer and one software cartridge (either a TI Extended BASIC, Speech Editor or Terminal Emulator II). A cassette recorder or disk drive can store any program written for speech.

The basic routine for speech is as simple as typing a CALL SAY command. The manual for each software cartridge is well written for learning with examples to enhance understanding. My first speech program was written as follows:

```
10 CALL CLEAR
20 PRINT "ENTER LETTER (A-M)"
30 CALL KEY (0,KEY,STATUS)
40 IF STATUS=0 THEN 30
50 X=KEY-64
60 ON X GOTO
70,80,90,100,110,120,130,140
,150,160,170,180,190
70 CALL SAY("I AM A COMPUT-
ER")
75 GOTO 10
80 CALL SAY("WHO ARE YOU")
85 GOTO 10
90 CALL SAY("SAY WHAT")
95 GOTO 10
```

The pattern continues down to line 195 with words in each CALL SAY command as follows: DO YOU WANT TO START SOME THING, UHOH, HELLO, GOOD BY, I DO NOT UNDERSTAND, #WHAT WAS THAT#, #TRY AGAIN#, #THAT IS RIGHT#, #THAT IS INCORRECT#, YOU ARE FINISHED NOW. The # symbol surrounding some phrases distinguishes them as special phrases reserved in the speech vocabulary.

I used the above program in a telephone conversation authorizing the computer to do all the talking for me by entering a letter from A through M. The program could be expanded with menus and submenus to aid those who are limited (even from laryngitis)in their regular human speech. A similar system was written for Stephen Hawking. He was disabled in his speech and wrote the scientific best-seller book titled *A Brief History of Time.*

Speech routines are helpful in a spelling test program allowing the computer to ask what words to spell. Speech routines in a program permit the blind to use a computer. So, computer speech can help the blind to read with their ears and help the deaf to talk automatically and independently.

The regular speech routines allow a TI computer to be up and talking in no time. The TI99/4A is also flexible if you want to specialize speech. You can use special symbols (+ - , ; .) to set a pause between 0 and 1 seconds by fractional time increments. You can assign words to variables to save time repeating words or phrases. You can use the SPGET command for viewing the internal speech code storing speech data, and for a slightly more natural sounding voice.

If you want the ultimate in speech flexibility, experiment with the Terminal Emulator II cartridge. It increases the regular vocabulary of 373 words and phrases to any number of imaginable words. You can also adjust the pitch (highness) and slope (pitch rate) to fine tune the speech even to a realistic whisper.

There is an option to adjust for inflection by stressing certain words or syllables so that you can even force the phrase "What time is it" to sound exactly like a question. It will do a good job of interpreting and pronouncing any word by using the allophones option to access any kind of original sounds for words with fine distinction, such as the difference between "skull" and "pull."

TI speech matches human speech nicely, for it abides by the philosophy that it is not always just "what you say" but "how you say it." It also passes a human quality with the word UHOH included in the regular vocabulary.

To ask the TI99/4A to deliver any phrases you type in when prompted, type in this program (the Speech Synthesizer and Terminal Emulator II must be connected).

```
10 OPEN #1;"SPEECH",OUTPUT
20 INPUT "PHRASE-":A$
30 PRINT #1:A$
40 GOTO 20
RUN
```

As an added bonus, experiment with repetitive keys such as GGGGGGGGGG or IIIIIIIIIII or QQQQQQQQQ or MAMAMAMAMAMAMA. Your children will have a ball with this program!

The making of the Corcomp 9900 disk controller card

By W. R. MOSEID

This article appeared in VAST News, the newsletter of the Valley of the Sun TI99ers of Phoenix, Arizona.

When the decision was made to provide a new disk controller card to the TI99/4A world, we began what turned out to be a long and arduous trip. The metal clamshell case was easy. Just use the one that had been designed for the CorComp 32K and RS232 cards. Then modify it slightly for a slot to accommodate the new circuit board that would project through the back of the case. This part would hold the connectors for the cables to the internal and external floppy disk drives.

Since all disk controller integrated circuits can control up to four floppy disk drives, we decided not to restrict people to three. With some careful planning, design and care a card could be produced which would support up to four disk drives with any combination from the following list of options:

Single- or double-sided
Single- or double-density
35 or 40 tracks per diskette side
Any combination of the above
Automatic density recognition

In order to allow the use of a variety of disk drive models, a feature was selected that allowed the owner to set the *head step time*. "Head step time" is the time it takes the disk drive read/write heads to "step" from one track to another. You can select one of four step times for each disk drive in your system. The times supported were 15 milliseconds, 10 ms., 6 ms., and 3 ms. This timing is set by positioning a set of DIP (dual in-line pack) switches to various on/off settings. The decision was made to place the DIP switches inside the case. Even though this meant the user would have to remove the case to set all the switches, this approach was selected for the following reasons:

• Safety for the card — Makes sure power is off when the DIP switch is set.

• Safety for the DIP switch — The chance of something hitting it or changing the settings were minimized.

In order to allow the use of a variety of disk drive models, a feature was selected that allowed the owner to set the *head step time*.

• Lower costs to the consumer — Assembly time and material costs were lower.

• The 10 ms. factory setting works with most drives.

When the 99/4A power-up sequence was examined, several interesting things were discovered:

• Plato, Terminal Emulator II and other command modules have a special sequence that runs at that time. Once they get control, they do not give it up and the power-up scan is not completed.

• In order to allow the user the ability to select the CorComp Disk Manager from the title screen, a special power-up screen had to be made to allow our rapid loader to execute on a single keypress.

• Because of the way Plato, TEII and other modules operate during power-up, a choice of two different menu screens had to be provided.

The early timing studies were done with direct I/O using an assembly language program (NO GROM). At that time we calculated that the CorComp disk controller in double-density could run two to four times faster than the TI disk controller. This was demonstrated by the speed at which the 98-sector disk manager program loads into expansion memory. This is based on the way that the CorComp controller card accesses the diskette and transfers the information into the computer.

When TI designed its disk memory system, it was decided that a memory expansion would not be required. This way, with the controller — the old standalone sidecar version — and a console, you could utilize BASIC with a disk system. To do this, the TI disk system and all of the modules that use floppy disks expect the information read from the disk to be in the console (VDP RAM) memory.

For example, when BASIC, Extended BASIC or the Editor/Assembler load from disk, they expect the information to be in console memory. If a memory expansion is attached, then Extended BASIC will move the information to the memory expansion after

(See Page 19)

Fig. 1 Time to load (in seconds)

File type	No. of sectors	TI controller SD		CorComp controller SD		CorComp controller DD	
		L/O	Ready	L/O	Ready	L/O	Ready
Console BASIC program	47	7.2	24.5	7.2	24.5	5	22.3
Extended BASIC program	39	6.3	8.8	6.3	8.8	3.6	6.3
Extended BASIC I/V254 program	52	NA	18.2	NA	18.2	NA	14.9
Editor/Assembler program	25	NA	6.3	NA	6.3	NA	4.2
Editor/Assembler D/F80 file	181	NA	55.8	NA	54.3	NA	47.8

L/O — Controller out
Ready — Cursor back on screen

CORCOMP—

(Continued from Page 18)

it is loaded. This moving process is very time consuming. Remember, each sector (256 characters) read or written must be passed through VDP RAM to be compatible with TI firmware/software. The table in Fig. 1 contains timing tests using GROM and VDP RAM.

The tests in Fig. 1 were conducted with default interlace selections. Timing may be improved with different interlace selections for the various modules and languages.

With the speed increase indicated in the previous table, we naturally were curious how time is consumed having to use the VDP RAM as an intermediate storage area. The table below shows the time required to copy a disk. When the CorComp disk controller copies a disk, VDP RAM does not have to be used as an intermediate storage place, thus saving time.

Time to copy 360 sectors (3 files) in seconds

		CorComp controller	
Type of copy	**TI controller**	**Wo/Turbo**	**W/Turbo**
SD to SD	151	143	70
SD to DD	NA	135	61
DD to DD	NA	123	51

Measuring performance increase figures is always a challenge. This is due to the fact that the "statistics people" can make them do what they want. But you can see a performance increase in using CorComp disk controller and disk manager of up to about 296 percent, depending on several of the following factors:

- Diskette density
- The operation being done
- Diskette file type (Display/Variable is the worst)
- Language used (GROM — worst case)
- Kind of loader (Extended BASIC, Editor/Assembler, etc.)
- File sector location on the diskette

CORCOMP DISK MANAGER UTILITIES

When the CorComp utilities were designed we learned that TI changed Extended BASIC to prevent its scanning the peripheral DSRs for CALLs, such as CALL FILES, from running a program. To allow the utilities to function with Extended BASIC, a list of the utilities had to be provided in the Link Table in extended memory. Thus the syntax for using the CorComp utilities in Extended BASIC is CALL LINK(utility name)(etc.....).

Console BASIC did not possess this constraint. In BASIC, the utility syntax is CALL(utility name, etc.).

The CorComp disk manager was written especially for the CorComp disk controller. We tried to think of all the features one would like in a disk manager. Just about all of the ideas are in the current disk manager. A decision was made to distribute the disk manager on diskette because it would be easy to easy to release an update and the cost to customers would be lower, compared to a module.

Figuring out all the technical details of how to achieve compatibility with the TI hardware, software and firmware was a hard and time-consuming effort. At times, some of the issues seemed almost too much to overcome. In the end, our perseverance and determination were rewarded and the disk controller card reached the market. All the known problems had been resolved by November 1984. All other third-party cards are compatible with the CorComp 9900 Disk Controller Card.

THE MANUAL

While writing the manual, we decided to try to present the material in a manner that would allow the beginner to follow the guide in a logical, step-wise manner. This allowed the user to learn how to use the card in a straightforward manner. The manual was also designed for the technically inclined. It required some 700 hours of effort before these objectives were achieved.

Run Artist Cardshop, Page Pro 99, and TI Artist Plus! from the Horizon RAMdisk

By MARY PHILLIPS

This article originally appeared in the Ozark 99er News, the newsletter of the Ozark 99ers of Springfield, Missouri.

Wanting to run Paul Coleman's CARDSHOP off a Horizon RAMdisk, I coped the files onto a section designated as DSK9 and tried to run it. The little red light on DSK1 kept coming on as the computer searched for files which it thought were located on DSK1. Suggestions were given to use the CALL DN command to temporarily change the RAMdisk to DSK1. The program still wouldn't work.

I was finally successful by using the Diskreview feature of Funnelweb. Catalog DSKn, press "I" for Inspect, 3 for Disk Search, 1 for ASCII string, press Enter at ASCII Wildcard ?, and type DSK1.???. A window appears telling you that the search is beginning at sector >000. Press Enter and the program starts looking for any instances of DSK1.(anything) — that's what "wildcard" means.

The first DSK1 was in file DSK9.1_F in sector >030, according to the information box in the lower right corner, and bytes @>00 is >02. Now I don't know exactly what that means, but highlighted on the screen are two lines that say DSK1.CARDCONFIG and No DSK1.CARDC. The box in the lower left corner offers options to:

1. Continue
2. Edit
3. Quit

I choose option 2, and a cursor appears on the first "D" and the byte now says

(See Page 20)

RAMDISK—

(Continued from Page 19)

@>E0 is >44. As I move the cursor with c-D, the byte changes. When I get to the 1, I type a 9 to replace it. Control-W tells the program to write the change back to disk and f-6 to proceed. F-9 gives me choices:

1. More
2. Restart
3. Abandon

I choose the first option and notice that the search begins at sector >001. The next instance is found at sector >03A and byte @>00 is A1. I kept repeating this procedure until a box appeared which says "Search done, Press any key." I press Enter and the search choices are there again. Function-9 backs me up to the disk catalog.

CARDSHOP is written in c99 and Rodger Merritt told me I would need the Extended BASIC LOAD program to run it. So, on my RAMdisk menu, I type DSK9.LOAD as the program to run. I hit FCTN-9 and Enter and I choose CARDSHOP from my menu — yes! I'm in business. Immediately, I made a backup of DSK9 on a floppy so that, if and when I have to reload the program, I won't have to repeat the task. There, perhaps is an easier way, but this worked.

Page Pro 99 was another program I wanted to run off my RAMdisk, and I did the same procedure in another section of my RAMdisk I called DSK8, using DSK8.LOAD in the RAMdisk menu as the program to run.

Then I tried TI-Artist Plus!. The procedure didn't work as well this time, as there were many sections which seemed to repeat the same phrase. When I was finished, it wouldn't run. Ahhhh, the manual — check the manual. Right there, on page 5, "Note that the drive which the Load and Run and Program File loaders use may be modified with the @NEWPATH program described below."

Let me mention the minimum Artist Plus! files that I copied to the RAMdisk: ARTIST, ARTIST1, ARTPT1, ARTPT2, ARTPT3, ARTPT4, CONPT1, ENHPT1, ENHPT2, ENHPT3, EXTDSR, FONPT1, FONPT2, LOADART (name changed from LOAD so I could put it on DSK8 with Page Pro 99), MECHA, PRINTER, PRNPT1, PRNPT2, PRNPTH, PRNPTV, SELECT, TITLE_M, VECPT1, VECPT2

Page Pro 99 was another program I wanted to run off my RAMdisk, and I did the same procedure in another section of my RAMdisk I called DSK8....

and VECPT3. These are the files needed to run TI-Artist Plus! on a double-sided, single-density disk. They use 645 sectors. (Thanks to Rick McWilliams of Macon, Georgia, for his help.)

Put TIAP-2 in drive one. In TI BASIC, type OLD DSK1.@NEWPATH and press Enter. Type RUN and press Enter. Questions are as follows:

What peripheral contains TI-Artist Plus!? — DSK1 <Enter>.

The current path used with this TI-Artist Plus! — DSK.INSCEBOT <Enter>.

New path for this TI-Artist Plus! — (Press Enter for no change) DSK8 <Enter>.

Happily, now I use these three programs with much improved processing speed.

A few more tips to make your TIA-Plus! work from a RAMdisk

By JERRY KEISLER

Mary did a good job of changing TI-Artist to operate on drive 9, however, there are a few enhancements I would like to add.

The TITLE_M file takes 167 sectors that could be used for a lot of other things. TI-Artist Plus! will not load the TITLE_M file if it cannot find it. If you have only one drive, just leaving TITLE_M off the disk will do the trick. However, if you have several drives, TI-Artist Plus! will hunt all drives for TITLE_M before it continues with TI-Artist Plus!. To avoid the drive hunting routine, you can change the drive number in ARTIST1 and load ARTIST1 using an Editor/Assembler loader. I use BOOT for my loader because it can also load Extended BASIC and other E/A type programs.

I keep all of my most used programs on drive 5. I store most of my temporary files on drive 4. I went through TI-Artist Plus! and changed all the Program file loads to 5 and all the working file saves to drive 4. I also took the liberty of converting the long names given to the file saves for Instances, Pictures, Slides and Vectors to their first initial. Now when I want to save something, I am prompted for drive 4 and the first initial of what I am saving. I can add to the file name if I want a better description. I no longer have to erase the end of the file name. I also save the most used font, which is also the smallest all-purpose font I would find, to the program disk and called it FONT1_F.

I keep the following files on my RAMdisk (DSK5.PROGRAM) to run TI-Artist Plus!: BOOT, BOOU, ARTIST1, ARTPT1, ARTPT2, ARTPT3, ARTPT4, CONPT1, ENHPT1, ENHPT2, ENHPT3, EXTDSR, FONPT1, FONPT2, FONT1_F, SELECT, PRINTER, PRNPT1, PRNPT2, PRNPTH, PRNPTV, VECPT1, VECPT2 and VECPT3. I did not load MOVPT1 or MOVPT2 as I do not use those files.

Decide where you want your Artist program and where you want to keep the temporary files you make. Make the changes to TI-Artist Plus! as shown below. Use the number of your program disk in place of 5 and the number of your temporary disk in place of 4. **Make all changes on a back-up of TI-Artist Plus!.**

I used Funnelweb 4.4 to make the changes. Load Disk Review, select the drive TI-Artist is loaded on, move the cursor to ARTIST1 and press "I." Select 1 for Sector Edit, then 2 for Offset in file, then change the file offset ? >000 to 000. Now move the cursor to byte 11 using the byte counter in the lower right corner of

(See Page 21)

RAMDISK—

(Continued from Page 20)

your screen and change it to 5 or the drive number containing TI-Artist Plus!.

Save the change by pressing CTRL-W. Continue this routine for the other files listed (see Fig. 1). When "To" is shorter than "From," erase the excess From using the space bar.

You can eliminate several loaders by using BOOT. Following is a description of part of my BOOT file:

TI-ARTIST PLUS	DSK5.ARTIST1.
FW4.4—5.01	DSK5.FW
TI SORT 1.02	DSK5.TISORT0
TI-BASE 3.02	DSK5.TIBASEP
ARCHIVE 303	DSK5.ARC303
MASS COPY	DSK5.MCOPY
TETRIS	DSK6.TETRIS

Fig. 1 TI-Artist Plus!

Filename	File Sect	Byte	From	To
ARTIST1	000	11	DSK*	DSK5
ARTPT1	008	D1	DSK2	DSK4
ARTPT1	008	AD	DSK.INSCEBOT.	DSK5.
ENHPT1	002	6B	DSK2	DSK4
ENHPT1	002	87	DSK2.SLIDES	DSK4.S
ENHPT1	002	AB	DSK2.INSTANCE	DSK4.I
FONT1	002	47	DSK2.FONT1	DSK5.FONT1
SELECT	001	0D	DSK2	DSK5
SELECT	001	AD	DSK.INSCEBOT.	DSK5.
VECPT1	002	1D	DSK2	DSK4
VECPT1	002	39	DSK2.VECTOR	DSK4.V

Crazy XB

Secrets of Extended BASIC

By WESLEY R. RICHARDSON

Richardson is a member of the Northcoast 99ers of Cleveland, Ohio. This article has appeared in several user group newsletters.

The purpose of this article is to describe how an Extended BASIC program is stored on disk and how a program can have line numbers out of sequence, or even have hidden lines, yet still run properly. The intent is to inform programmers so they can attempt to restore programs which have been altered. I recommend the use of formats in any programming language which conform to the specified protocols.

The program CRAZY-XB1 is a very simple program that prints "LINE 40," "LINE 50" and so on to the screen. The listing for CRAZY-XB2 shows how the program can be altered to have descending line numbers. Note that line number seven is for two different instructions. From the listing for CRAZY-XB3 it would appear that only line 10 is in the program yet, when it is run, it will function exactly like -XB1 and -XB2.

The purpose of this article is to describe how an Extended BASIC program is stored on disk and how a program can have line numbers out of sequence, or even have hidden lines, yet still run properly.

CRAZY-XB1

```
10 REM CRAZY-XB1
20 REM WESLEY R. RICHARDSON,
FEB 1990
30 REM NORTHCOAST 99ERS,
CLEVELAND, OH
40 PRINT "LINE 40"
50 PRINT "LINE 50"
60 PRINT "LINE 60"
70 PRINT "LINE 70"
80 PRINT "LINE 80"
90 PRINT "LINE 90"
100 END
```

CRAZY-XB2

```
10 REM CRAZY-XB2
20 REM WESLEY R. RICHARDSON,
FEB 1990
30 REM NORTHCOAST 99ERS,
CLEVELAND, OH
9 PRINT "LINE 40"
8 PRINT "LINE 50"
7 PRINT "LINE 60"
7 PRINT "LINE 70"
6 PRINT "LINE 80"
5 PRINT "LINE 90"
100 END
```

CRAZY-XB3

```
10 REM CRAZY-XB3
```

To understand how these programs work, we must first look at the Extended BASIC representation for the program. If you refer to the CRAZY-XB1 ASCII code sector listing (Fig. 1), you will see that the lines are listed in reverse order. The listing has line 90, then 80 and so on, ending with the CRAZY-XB1 statement. Note that if you edit a line or add a line, that line gets moved to the beginning of the file. If line 40 is edited, it will be in the file ahead of line 90. If you edit a program and simply save it, the lines as listed on the screen will be in proper order, but internally they will be quite mixed. If you have a program, for example PROGNAME1, in which you have made several changes, the lines can be re-ordered by the following steps:

1. SAVE DSK1.PROGNAME2,MERG

(See Page 22)

CRAZY-XB—

(Continued from Page 21)

E

2. NEW
3. MERGE DSK1.PROGNAME2
4. SAVE DSK1.PROGNAME3

I suggest using different filenames in case you make an error, then you can recover using the original file. When creating a program, do all of your debugging and modifications and when your program is finished, use the MERGE routine to reorganize the internal program lines.

Now that we understand that the BASIC lines can be out of order in the file, how do we modify the line numbers? If you refer to the CRAZY-XB1 hex code sector listing (Fig. 2), you will see how Extended BASIC keeps track of the line numbers. In the first row, locate 0064, which is line 100. Also in the first row is 005A, which is line 90. You can see the old line numbers in hexadecimal and decimal.

Old line #		New line #	
Hex	Dec	Hex	Dec
0064	100		
005A	90	0005	5
0050	80	0006	6
0046	70	0007	7
003C	60	0007	7
0032	50	0008	8
0028	40	0009	9
001E	30		
0014	20		
000A	10		

Using a sector editor, I changed the old line number hex values to those indicated under "New." If you examine rows one, two and three in the CRAZY-XB2 hex code sector listing (Fig. 3), you will see these changes. But wait — how can the program still work? Extended BASIC executes instructions according to memory location, not according to line number. When we list the CRAZY-XB2 program, it appears on the screen as I listed it previously. If you try to edit the program by typing 10 then FCTN-X, you will be able to see lines 10, 20 and 30. But when you go to line 9 (the old line 40), Extended BASIC will tell you "Line not found." However, the program will still run correctly.

HIDING LINES

If we make one more change, we can hide some lines. By changing the sector row one value of 0064 for line 100 to a value like 0001, you will produce CRAZY-XB3. Now only line 10 can be viewed when listed, but the program still works fine.

Line numbers in Extended BASIC range from 1 to 32767, or hex 0001 to 7FFF. If we change the line number to a value in the range of 8000 to FFFF, it will cause a break in the program when that line is executed. For example, if the program reached the line number 83E8, the line number would then have the value of 8000 subtracted, leaving 03E8. The message "Breakpoint in 1000" would be displayed.

In Fig. 2, in lines 1 to 3, there are two-byte or four-digit number, such as 373B, 373E, 374A and 3756, after each line number. These refer to the memory locations for the Extended BASIC instruction. The difference between adjacent values is the number of bytes used for the Extended BASIC instruction. The format for each instruction is XXYYY..YYY000. The "XX" is the number of bytes used for the instruction, not including the 00. Since the maximum value which can be represented is FF, the longest line length in Extended BASIC is 255 bytes. Depending upon the statements you use, this 255-byte length can have different ASCII lengths, which you can see when entering an Extended BASIC program. The Extended BASIC statements are stored in token format. For example, PRINT is 9CC7. The word PRINT takes 5

(See Page 23)

Fig. 1

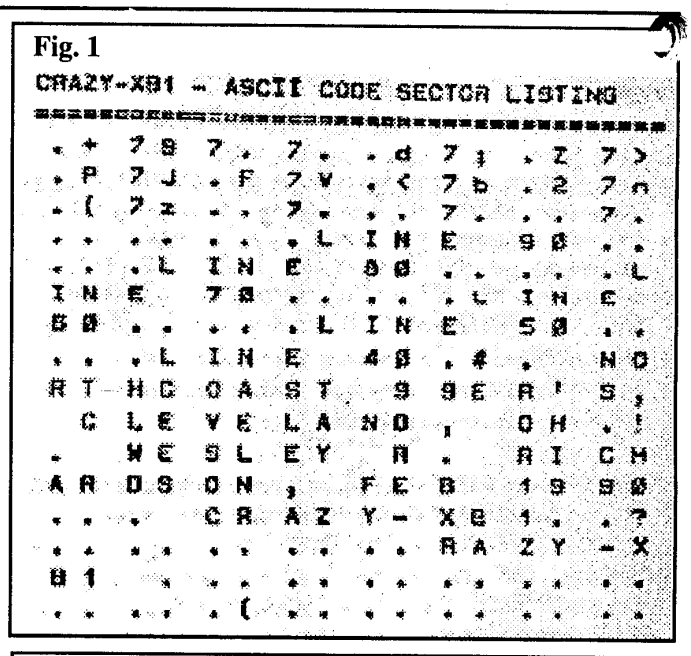
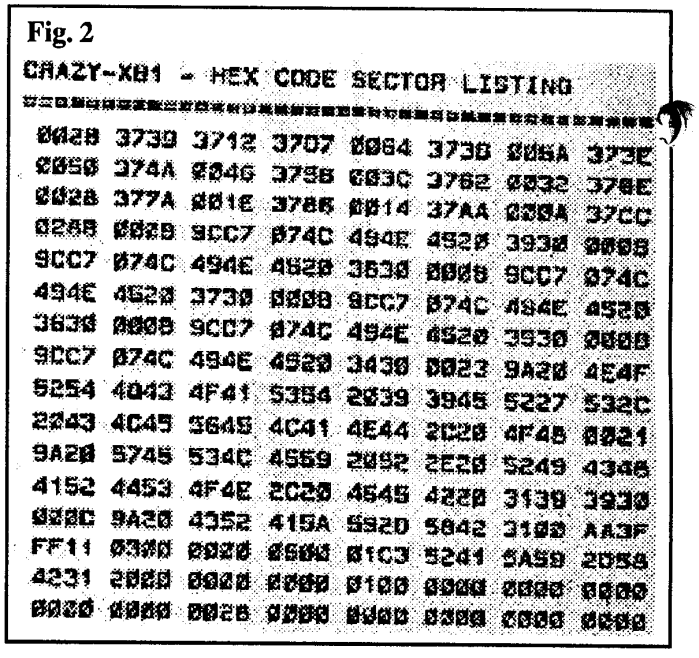


Fig. 2



CRAZY-XB—

(Continued from Page 22)

ASCII bytes, but Extended BASIC requires only two bytes to store 9CC7.

Some information, such as the text contained in print statements, is in the same format when saved to disk. For example, the characters "LINE 50" are stored on disk in the readable form as shown in line 7 of Fig. 1.

The third format that Extended BASIC uses on disk for program files concerns CALL statements. Memory must be reserved for variables and CALL statements. One way to find the tokens for each of the Extended BASIC commands is to write a program using each of the commands on a separate line, and then look at the hex codes using a sector editor. Be sure to use the MERGE technique listed above if you wish to keep the sequence of lines in order when the program is saved to disk.

As I indicated earlier, I do not agree with using hidden instructions or hidden machine language code in Extended BASIC programs. If you encounter one of the modified programs, perhaps now you will have some idea about how they were modified and the meaning of the values of an Extended BASIC program stored on disk.

Fig. 3

```
CRAZY-XB1 - HEX CODE SECTOR LISTING
0020 3730 3712 3707 0084 3730 006A 373E
0050 374A 0046 3786 003C 3762 0032 378E
0028 377A 001E 3786 0014 37AA 000A 37CC
0288 0020 9CC7 074C 494E 4520 3930 0000
9CC7 074C 494E 4520 3630 0000 9CC7 074C
494E 4520 3730 0000 9CC7 074C 494E 4520
3630 0000 9CC7 074C 494E 4520 3530 0000
9CC7 074C 494E 4520 3430 0023 9A20 4E4F
5254 4043 4F41 5354 2039 3945 5227 532C
2243 4C45 5645 4C41 4E44 2020 4F46 0021
9A20 5745 534C 4559 2052 2E20 5249 4348
4152 4453 4F4E 2020 4645 4220 3139 3930
982C 9A20 4352 415A 5920 5842 3100 AA3F
FF11 0300 0020 0500 01C3 5241 5A59 2059
4231 2020 0020 0000 0100 0000 2000 0000
0020 0020 0020 0000 0000 0200 2000 0000
```

NEWSBYTES

Swedish newsletter quits

Programbiten, the quarterly Swedish TI newsletter, ended publication at the end of 1994. Jan Alexandersson of Trånsgund, Sweden, sent the last issue to MICROpendium.

Great Lakes get new address

New mailing address for the Great Lakes Computer Group is 236 Wendy Lane, Bloomfield Hills, MI 48302-1178.

TR Software moves

New mailing address for TR Software is c/o Gerald D. Turner, 4009 Twilight Ave., Enid, OK 73703.

AMS schematics in AutoCad format

The SUPERams SRAM design schematics are now available on AutoCad format and DXF picture format, according to Jim Krych.

Krych also has the original, hand-drawn schematics on three letter-size paper sheets.

Price for the AutoCad schematics is $20 U.S. with three double-sided, single-density 5.25-inch floppy disks. Krych will also supply orders using 3.5-inch diskettes. For more information, contact him via email at ab453@cleveland.freenet.edu.

1995 TI FAIRS

APRIL

Lima Multi Users Group Conference, April 29, Reed Hall, Ohio State University at Lima. Contact Lima Users Group, P.O. Box 647, Venedocia OH 45894, or call Charles Good (evenings) at (419) 667-3131 or Internet cgood@osulima1.lima.ohio-state.edu

SEPTEMBER

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

1996 TI FAIRS

FEBRUARY

Fest West '96, Feb.17, Quality Inn, 1601 Oracle Dr., Tucson, Arizona. Contact SouthWest Ninety Niners User Group by sending e-mail to twills@primenet.com. Or call the Cactus Patch BBS at (520) 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.

DOS utilities for TI-Artist files

PC users have new way to use TI artwork

By JOHN KOLOEN

TI users who also use PCs can display and print graphics created with TI-Artist using a set of programs developed by Jeffrey A. Kuhlmann.

The programs, which run on an IBM-compatible PC in DOS mode, allow users to view TI-Artist instances and pictures in monochrome. See the sample below. I outputted the graphic to a laser printer, which I don't normally have connected to my Geneve. This saved time and was a relatively painless procedure.

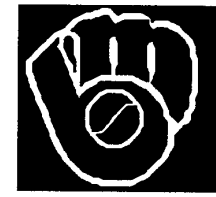
Kuhlmann notes that his software isn't user-friendly. What this means is that not only does it operate only through DOS, but you activate the programs from the command line, rather than a menu. However, this is not difficult to anyone familiar with DOS. Here are some of the commands and how they are implemented:

REVIEW

• showtia — allows IBM to view TI-Artist Instances — usage: showtia file_name graphic_mode (graphic mode is optional)
• showall — allows IBM to view all TI-Artist Instances on drive — usage: showall wildcard graphic_mode (graphic mode is optional)
• showtip — allows IBM to view TI-Artist _P file — usage: showtip file_name header_length (usually 128)
• sidetiai — a program to view TI-Artist Instances sideways, so <shift><printscreen> will work

Also included are routines to view TI CSGD graphics. These work similarly to the TI-Artist programs so I won't list them here.

How you get the TI-Artist files into your PC is up to you. There are numerous ways, from using programs like PC-Transfer to uploading them to an electronic service from your TI and downloading them into your PC.



Sample of TIA file printed from a PC.

As would be expected from a barebones set of programs, the documentation for these programs consist of two small text files on disk. Since it's a PC program, you'll need a PC to read them. But, brief as the docs are, they're enough to get you going.

For more information about the programs, write Kuhlmann at CMR 4/6, Box D, APO, AE 09140.

MICRO-REVIEWS

Mailing List Manager like a dream come true

By CHARLES GOOD

Mailing List Manager by Bill Gaskill

Ever since I got my first 99/4A back in 1982 I have been looking for the perfect mailing list (name, address, phone number) software. I started out years ago with Personal Record Keeping and have made extensive use of three other name and address programs, each a little better for me than the previous software. My quest has been similar to searching for the Holy Grail in that I often appear close to my goal, but never quite get there. Features of existing software just don't quite seem to fit my needs. Some of the problems I have encountered over the years with such software include:

• Kludgy data entry screens and difficulty in correcting incorrectly entered data.

• Inability to enter enough data of the type I desire, such as entering a "country" and very long phone number for foreign addresses.

• Limited number of names one can put in a data file. This happens if an entire data file has to be put into memory each time you use the program.

• Difficulty in deleting a single name from a large data file. You'd be surprised how difficult it is to accomplish this very basic task with some existing mail list software.

• No ability to send special codes to my Gemini 10X printer so it will print mail labels in dark "emphasized" letters rather than thin dot matrixy "draft mode" letters.

• Slow, or no sorting ability. If sorting ability exists, you sometimes are forced to sort your list in memory each time you print sorted mailing labels since the software can't create a sorted data file from an unsorted data file.

• Etc.

NEARLY PERFECT

The closest I have come to my perfect mail list software is Mailing List Manager (MLM). The problem with switching to a new mail list program is that you have to start from the beginning. You have to manually type in all your data into the new program, data you have already long ago typed into the old program. After experimenting with MLM I was so impressed with MLM's features that I decided to transfer my user group membership and newsletter exchange mailing list from the software I had been using to MLM. This means that I spent 3 hours at the keyboard typing into MLM and checking over 150 names and associated data. I consider this time well spent.

I am very happy with MLM, and that is why I am devoting my entire column this month to this one product. I only review

(See Page 25)

MICROREVIEWS—

(Continued from Page 24)

good software for MICROpendium, but that doesn't mean that I personally have a use for the software I review. Unlike many of the other software packages I review here, I am now actually using MLM on a regular basis.

MLM can handle name and address files of unlimited length, subject only to the physical limitations of the media where the file is stored. Each name and address data group takes one disk sector. Although sorting within MLM is limited to files with no more than 1,000 names, an alternative means of sorting larger files is provided. The fully functional fairware version of MLM comes on a SSSD disk and will work very nicely on systems with only one SSSD drive. You can use a separate SSSD data disk with up to 358 names. The program prompts you to insert the program disk into a drive whenever that is necessary.

MLM is set up to have its system files run automatically out of any floppy drive or RAMdisk because it looks for a volume name, not a drive number, when it loads parts of itself into memory. It also works with hard drives and allows you to have up to 23 characters in path names.

MLM is written in Extended BASIC, with various assembly CALL LINKs to speed things up. Default printer names, printer control codes, label spacing, and data file paths can be changed on the fly from within MLM, and you can also permanently change these defaults by changing the Extended BASIC code.

The following data fields are found in each name and address record:

lname
fname
address
city
state
zipcode
nation
group code
home phone
work phone
fax bbs phone
dates
notes 1
notes 2

You can leave any of these fields blank and update them later. Group code might be used for user group affiliation. Dates might be used to indicate when a user group membership expires. The date entry is printed on mailing labels. I particularly appreciate the nation field, something rarely found in mailing list software. This makes it easier to deal with airmail mailings to international locations.

MLM can handle name and address files of unlimited length, subject only to the physical limitations of the media where the file is stored.

ROOM FOR NOTES

Actually you can put any text you want into any of these fields. Thus, you have room to put lots of notes and comments into the last seven fields listed above. You might, for example, put XMAS in the Group Code field to identify those who are on your Christmas card list. It is not necessary, for example, to put numerical digits in all three phone number fields. You can just as easily enter some text.

When you finish typing your data in all the fields for a particular new or updated name and address and press "C" to continue, the data is immediately written to the data file. There are no complicated "exit the program" procedures required to make sure all files are closed. Wherever you are in MLM just press FCTN-9 (Back) a few times for a quick exit to the title screen. All files are safely closed when you do this, so your data remains secure.

You can sort an entire data base on any one of the data fields. When a list is sorted the computer reads the list into memory, sorts the list, and creates a new sorted file on disk. This leaves the original unsorted file intact. Doing all this to a list of 140 names stored on a Horizon RAMdisk (sorting an unsorted list by ZIP code) took me only 1 minute 35 seconds. By 99/4A standards that is fast. An assembly language sort is used.

You can search a data file using either one or two key words in either one or two data fields. If you sort by two key words you only get a hit if both words are found. Searching for only one key word requires that you specify the same key word and data field twice when asked for the first and second key word to be used in the search. This takes some getting used to.

Partial strings can be used in these searches. For example, if you can't remember if L.L. Conner is spelled "Connor" or "Conner" you can search for "CONN". Data entry is automatically in uppercase, so you don't have to worry about what is and is not in uppercase when you do a search.

THREE WAYS TO DELETE

There are three ways to delete names form a file. In each case a new file is written without the deleted names, leaving the original file intact.

1. To delete a single name or a few manually selected names from a file, first display on-screen each name to be deleted and mark it with a caret (Shift-6) in the first space of the lname field. You can mark any number of names this way. Then select "Delete Names" from the main menu and a new file will be written leaving out the marked names. This procedure is easy, safe (you still have the old file), and fast.

2. You can also do a global delete, creating a new file that has all the records from the old file except those containing a text string you specify. For example, you could create a file that omits Christmas list people.

3. Finally, you can create a new file that contains only records from the original file that have a text string you specify. This is sort of the opposite of No. 2 above. You can make a file that contains only Christmas list people. In creating subsets of files based on text strings, you can use either

(See Page 26)

MICRO REVIEWS—

(Continued from Page 25)

one or two text strings as described above for searching by key word.

If you can't remember a data file name you can, from within the program, display a disk directory. Then you can optionally delete any file from the disk.

Reports can be printed in either of two formats. You can also print mailing labels of the entire file or a single label of only the name currently displayed. It is possible to print labels or reports of subsets from larger lists. To do this the software lets you create an index file of a larger data file with pointers to specific records in the larger file. For example, you can make an index of all your names you have marked XMAS in one of the comment fields. Later you can select "Print using an index" to print mailing labels to your Christmas card list.

You also get some unusual software extras with MLM, all of which can be run within MLM:

• There are two different free-form mailing label editors They let you compose mailing labels on the fly and print multiple copies of these onto fan-fold labels. Examples would be return address labels or "Do Not Bend" labels for packages containing floppy disks. You can also load in templates of previously composed labels and print these. One of the two label makers also lets you print disk labels and automatically advances to the next label to continue printing if all the disk's file names won't fit on one label.

• MLM also has a 40-column text editor, which is great for writing short letters or keeping records of your correspondence in your database. This text editor has many features of the TI-Writer editor and is compatible with TI-Writer's text files. You are limited to one page of text at a time. There is no word wrap and no automatic margins.

There are two different free-form mailing label editors They let you compose mailing labels on the fly and print multiple copies of these onto fan-fold labels.

The software that compares most closely with MLM is Asgard's Mail Room, by Larry Tippitt. Both products have their advantages and disadvantages.

I reviewed Mail Room in one of my earlier MicroReviews columns. Mail Room has an 80-column version and allows you to use a modem to dial any phone number stored in its database, features not found in MLM. Advantages of MLM compared to Mail Room include MLM's "nation" and generous comment fields and the ease of permanently creating sorted data files and of deleting names from MLM data files. All Mail Room users should have a look at MLM.

Send me $1 and I will send you MLM on a DSSD disk ($2 for two SSSD disks). The author asks $15 to register your copy of the program. Registered owners will receive an expanded hard copy of the instructions and an update with even more features than those described here for the fairware version. You might consider saving the dollar and immediately send Bill $15 with a request that he send you the most recent version of MLM.

QUICK SEARCH OF ANY DATABASE

Lets say you have a data file created with MLM, TI-Base, First Base, PR Base, PRK, or just about any other database software usable on the TI. If all you want to do is display a particular name, address, and phone number (or some other data within a data file) on-screen, there is a quick and easy way. This method is usually much faster than loading software such as MLM and using the software's internal search engine. Instead of doing that, use an assembly language "find string" to quickly display a data file sector with the desired text you are looking for. Funnelweb's Disk Review or John Birdwell's DSKU can be used to do this. These are programs you probably already have on your computer's menu system.

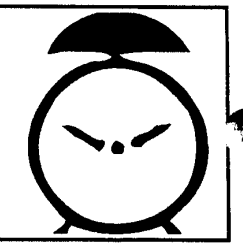
From a Disk Review disk directory, move the cursor next to a file you want to search and press I (for inspect). Select 2 (File search) and then 1 (ASCII string). Press Enter when the cursor appears over the first question mark and enter your search string over the second group of question marks.

From DSKU, select File Utilities from the first menu and then select Find String. Enter the file name, drive number, "A" (for ASCII), and the text you want to find. If your file is on a Horizon RAMdisk, such string searches usually take less than 10 seconds. MLM and many other databases automatically store text only in uppercase, which takes much of the guesswork out of a string search.

ACCESS

Bill Gaskill, 10 Cypress Court, Grand Junction, Colorado, 81506.

Charles Good, P.O. Box 647, Venedocia, Ohio, 45894. Evening phone: 419-667-3131. Internet email: good.6@osu.edu



USER NOTES

Printer control codes

The following was written by Bill Sheridan and appeared in the newsletter of the K-Town 99ers of Knoxville, Tennessee.

Jim Lesher is writing a series of articles called Beginner Printer that are appearing in the Dallas 99 Interface newsletters. Part 6 of the series was printed in November. In it I found something new and interesting.

First, he says that his articles are written using Funnelweb 5.0 and all printer commands are for the Gemini 10X printer. The commands listed below also work on my Epson-compatible Panasonic KX-P1091.

The part of the article that I was most interested in had to do with using superscript and subscript for scientific notation in a text file. He gives the examples of raising a number to a power and a chemical formula.

Here are the key presses to print the formula for water (H_20):

H
CTRL-U
FCTN-R
CTRL-U
S
CTRL-U
SHIFT-A
CTRL-U
2
CTRL-U
FCTN-R
CTRL-U
TO

Here are the key presses to print 5 raised to the third power (5^3):

5
CTRL-U
FCTN-R
S
CTRL-U
SHIFT-2
CTRL-U
3
CTRL-U
FCTN-R
CTRL-U
T

Ottawa newsletter ends publication

The Ottawa TI99/4A Users Group has produced the final issue of its newsletter. The newsletter was in its fourteenth year of publication. The club's BBS, Texlink, is available "for an undetermined period," according to the final issue. The BBS number is (613) 738-0617 (8N1; users need their Extended BASIC module number to log on).

28COLIST for MYBASIC

The following item comes from Jim Uzzell of DDI Software. He writes:

The following is a MY-BASIC version of 28COLIST published in the August 1994 MICROpendium.

PGMLIST28

```
100 !PGMLIST28
110 !28COLIST MODIFIED
120 !MICROPENDIUM AUG 94 PG
18
130 !MYBASIC VERSION
140 !BY DDI SOFTWARE
150 CALL GRAPHICS(2,1)
160 DISPLAY AT(4,1)ERASE ALL
 :"1st LIST your program to
disk then": :"      RUN PGM
LIST28"
170 DISPLAY AT(11,1):"INPUT
FILENAME?":"ex:DSKn.LIST80":
 :"DSK1": :"OUTPUT FILENAME?
```

(See Page 28)

BUGS AND BYTES

Term 80 progress report

Jeff Brown, author of the terminal emulator Term 80, reports progress on the latest version. Here is an updated he posted on the Internet.

"The last release of Term 80 is actually, compared to the current unreleased development version, a really poor program. Kinda funny to put down an older version of my own program. Likewise, I could say that V1.5.3 released on GEnie several months ago is *ancient*! I'd say since then, I have reworked *every* segment of the program (not just the major ones) and added tremendously.

"The GEnie release allowed only a Supercart and needed it. This one doesn't. Under development is an extended memory handler. Term 80 already employs an interesting memory handler routine. However, it only functions on directly accessible CPU RAM, whereas this one should prove well on indirectly accessible memory such as GRAM, VDP (tentative, have to find 9938 docs first), and all memory cards with paging.

A real arTIst

Ken Gilleland, known for his disks of graphics and computer games, recently had an exhibit of paintings (for those who like looking at pictures the old-fashioned way) in a gallery in California.

Fest success

Woody Wilson, writing in the Computer Voice, official newsletter of the Southern California Computer Group, notes that Fest-West was a success both for the club and the hotel, which wanted the event to come there again in 1996 "but Tucson has the honors for next year." The group needed 20 rooms rented to get the convention space free, and 44 room rentals were attributed to Fest -West.

USER NOTES

(Continued from Page 27)

```
":"ex:PIO or HDSn.LIST28": :
"DSK"
180 ACCEPT AT(14,1)SIZE(-40)
BEEP :F$ :: IF LEN(F$)<3 THE
N 180 :: ACCEPT AT(19,1)SIZE
(-15)BEEP :P$
190 OPEN #1:F$,INPUT ,DISPLA
Y ,VARIABLE 80 :: OPEN #3:P$
,OUTPUT
200 ON ERROR 300
210 C=28
220 IF EOF(1)=1 THEN 310 ::
LINPUT #1:A$ :: IF LEN(A$)<8
0 THEN 270
230 LINPUT #1:B$
240 M$=VAL(SEG$(A$,1,POS(A$,
" ",2))) :: Z$=VAL(SEG$(B$,1
,POS(B$," ",2)))
250 IF M$<Z$ THEN F=1 :: GOT
O 270
260 A$=A$&B$ :: IF LEN(B$)>=
80 THEN 230
270 A=LEN(A$) :: L=A/C
280 FOR I=0 TO L :: PRINT #3
:SEG$(A$,1+I*C,C) :: NEXT I
290 IF F=1 THEN F=0 :: A$=""
 :: GOTO 260 ELSE 220
300 ON ERROR 300 :: RETURN 2
60
310 CLOSE #1 :: CLOSE #3 ::
DISPLAY AT(23,4):"Do another
? (Y/N) N" :: ACCEPT AT(23,2
2)SIZE(-1)VALIDATE("YN"):Y$
:: IF Y$="Y" THEN 160
320 END
```

Getting the code

James Murta of Glendale, California, wrote the following tip in response to Richard C. Arthur Jr.'s letter (Feedback, March 1995), concerning his trouble with the line:

.LM 1;RM 132:F1;AD

The line rang a bell with me and I think you will find the answer in the TI-Writer Word Processor Manual starting at page 159 and continuing to page 171. When you got to the command mode on page 162, refer back to page 9 for explanations and help. By following the instructions for letter writing in the manual the program should work for you as it does for me, for I have the same equipment (Epson RX80/FT printer with 1 meg memory and Dots-Perfect chip installed) and I have been able to write the instruction letter from the TI-Writer Manual as well as a couple of personal letters to friends.

Hooking a TI99/4A up to your VCR

This is by Andy Frueh of the Lima Users Group. It's reprinted from the PUG (Pittsburgh Users Group) Peripheral:

You can hook up the computer to a VCR using a standard 300-ohm ro 75-ohm TV antenna adaptor or a composite monitor cable. Adapters are found with almost all home video game systems or at radio supply stores. They have a cable TV male connector and two screw terminals. The male plug goes into the "Cable In" jack of the VCR. The screw terminals go to the modulator's "To TV" wire. You can then hook the VCR to a stereo's "Aux in" jack using a standard aduio/video cable. Plug the other end into the "Audio Out" on the VCR. You can then hear improved sound and record the computer's output. Note that this isn't as good as using a monitor cable into the "Audio In" and "Video In" of the VCR, but it works for those without monitors.

Finally, for those without a monitor, here is another use for the adaptor mentioned ablve. I have the following display set up: a TI with two separate RF modulators. Each one is constantly hooked up to a TV. Only one of the DIN end plugs is connected at a time. I use a small black-and-white TV on the PE-box (with adequate ventilation) whenever abnyone needs to use the larger color TV (which is also connected to cable and a Nintendo). The problem is, when I'd use the color TV, either with the computer or without (i.e., I'd be using the B&W TV), I would get interference from the computer. Placing the antenna adaptor between the color TV and its modulator clears up the interference.

TI-Tips

Here are some simple tips from various sources. We saw them in the newsletter of the West Penn 99ers of Russellton, Pennsylvania.

• If you have a speech synthesizer and the Terminal Emulator II cartridge, here is a trick for debugging programs: All you have to do is enter your program, type LIST "SPEECH" and press Enter. Your computer will read your listing back to you as you check it with your original.

• If you want to disable the Quit key (FCTN-+) type in the following: `CALL INIT :: CALL LOAD(-31806,16)` and press Enter. This is done using Extended BASIC.

• Have you ever pressed ERASE by mistake and lost the whole line? Don't panic. And don't hit Enter. Instead, press FCTN-?, then press Enter. Your line will be intact.

• If you are going to save a program to tape and type OLD CS1, instead of SAVE CS1, don't panic. Press FCTN-E, then press Enter. This will take you out of the tape loop.

• If you have Extended BASIC and 32K, type this in as the last line of your program: `CALL INIT :: CALL LOAD(2,A,B) :: CALL LOAD(-31804,0,A,B)`. This will return you to the title screen when you the program ends.

• When your TI99/4A is hooked up to a black and white TV, use CALL SCREEN(15). This will disable the color generator and remove the vertical lines you may have seen.

Match-A-Patch offers challenge

The following program, aMatch-A-Patch, was written by Jim Peterson. It is written in Extended BASIC.

The object of this entertaining and challenging program is to match a swatches of color and patterns. It requires concentration to do successfully. There are five levels of play. Beginners might want to start out at the easiest level before moving on.

MATCH-A-PATCH

```
1 DATA 96,97,98,99,104,105,1
06,107,112,113,114,115,120,1
21,122,123,128,129,130,131,1
32 !236
2 DIM A$(16),KR(21),P(42),TR
```

(See Page 28)

USER NOTES

(Continued from Page 28)

```
(42),Z(42):: DEF N$=STR$(N)!
136
10 GOTO 100 !179
11 PL$(),SET,D,K,ST,PL,DF,J,
A$(),L,X,B$,C$,CH,X@,R,C,JJ,
NUL$,KR(),Z(),P(),N,N$,SX,W,
NR,TR(),CK(),CX(),RR(),CC(),
TX(),PY(),COUNT !233
30 CALL CHAR :: CALL COLOR :
: CALL CLEAR :: CALL TITLE :
: CALL DELSPRITE :: CALL SCR
EEN :: CALL KEY :: CALL HCHA
R :: CALL SOUND !150
40 !@P- !064
100 RANDOMIZE :: CALL CHAR(3
7,"0010301010101038",38,"003
844040810207C",40,"00")!071
110 CALL COLOR(9,2,16,10,5,1
1,11,12,14,12,12,13,13,7,16,
14,12,2):: PL$(1)="PLAYER #%
'S TURN" :: PL$(2)="PLAYER #
&'S TURN" !048
120 CALL CLEAR :: CALL TITLE
(2,"MATCH A PATCH"):: FOR SE
T=2 TO 9 :: CALL COLOR(SET,1
5,1):: NEXT SET !074
130 CALL CHAR(94,"3C4299A1A1
99423C"):: DISPLAY AT(1,10):
"TIGERCUB SOFTWARE" :: DISPL
AY AT(3,12):" ^ TCX-1102" ::
 GOSUB 290 !082
140 FOR D=1 TO 200 :: NEXT D
 !242
150 CALL DELSPRITE(ALL):: CA
LL CLEAR :: CALL SCREEN(4)::
 FOR SET=2 TO 9 :: CALL COLO
R(SET,2,1):: NEXT SET !102
160 ! Programmed by Jim Pete
rson 4/83, XBasic version 7/
85 !233
170 ! COPYRIGHT 1983 Tigercu
b Software  156 Collingwood
Ave., Columbus Ohio 43213 !0
90
180 ! REPRODUCTION PROHIBITE
D. DELETION OF COPYRIGHT NOT
ICE PROHIBITED. !149
190 DISPLAY AT(8,1):"TRY TO
MATCH UP THE PAIRS": :"OF CO
LORED SQUARES." :: DISPLAY A
T(13,1):"IF YOU MAKE A MATCH
, YOU": :"GET ANOTHER TURN."
 !100
200 DISPLAY AT(17,1):"HOW MA
NY PLAYERS - 1 OR 2 ? " !041
210 CALL KEY(0,K,ST):: IF (S
T<1)+(K<49)+(K>50)THEN 210 !
004
220 PL=K-48 :: DISPLAY AT(19
,1):"PICK SKILL LEVEL -  1 T
O 5 " !234
230 CALL KEY(0,K,ST):: IF (S
T<1)+(K<49)+(K>53)THEN 230 !
027
240 DF=K-48 !139
250 IF (PL<1)+(PL>2)THEN 200
 !012
260 CALL CLEAR :: CALL COLOR
(2,16,16,3,2,16,4,2,16):: GO
TO 340 !021
```

(See Page 30)

USER NOTES

(Continued from Page 29)

```
270 !!131
280 DATA 18,24,3C,42,5A,66,7
E,81,99,00,A5,BD,C3,DB,E7,FF
 !244
290 RESTORE 280 :: FOR J=1 T
O 16 :: READ A$(J)!157
300 NEXT J :: RESTORE 1 :: F
OR J=1 TO 21 :: FOR L=1 TO 4
 :: X=INT(16*RND+1):: B$=B$&
A$(X):: C$=A$(X)&C$ !166
310 NEXT L :: READ CH :: CAL
L CHAR(CH,B$&C$)!191
320 X@=4*RND+3 :: R=25-X@-IN
T(10*RND+1):: C=INT(24*RND+1
):: FOR JJ=R TO R+X@ :: CALL
 HCHAR(JJ,C,CH,X@):: NEXT JJ
 :: B$,C$=NUL$ :: KR(J)=CH !
007
330 NEXT J :: RETURN !234
340 FOR J=1 TO 21 :: FOR L=1
 TO 2 :: X=INT(42*RND+1)!131
350 IF Z(X)=0 THEN 380 !080
360 X=X+1 :: IF X<>43 THEN 3
50 !015
370 X=1 :: GOTO 350 !064
380 P(X)=KR(J):: Z(X)=1 !128
390 NEXT L !226
400 NEXT J :: CALL CLEAR ::
IF DF>1 THEN 420 !155
410 GOSUB 760 !074
420 N=1 :: FOR R=1 TO 21 STE
P 4 :: FOR C=3 TO 27 STEP 4
:: GOSUB 430 :: GOTO 450 !09
4
430 CALL HCHAR(R,C,40,3):: C
ALL HCHAR(R+1,C,40,3):: CALL
 HCHAR(R+2,C,40,3):: FOR J=1
 TO LEN(N$):: DISPLAY AT(R,C
-2):N$;!144
440 NEXT J :: RETURN !234
450 N=N+1 !021
460 NEXT C !217
470 NEXT R :: DISPLAY AT(24,
4):"TYPE NUMBER AND ENTER" !
150
480 SX=SX+1 :: IF PL=1 THEN
500 !151
490 W=W+1+(W=2)*2 :: DISPLAY
 AT(4,1):" " :: DISPLAY AT(8
,1):" " :: DISPLAY AT(W*4,1)
:PL$(W)!053
500 FOR L=1 TO 2 !060
510 ACCEPT AT(24,26)VALIDATE
(DIGIT)BEEP:NR :: IF (NR>0)*
(NR<43)THEN 530 !198
520 GOTO 510 !078
530 IF TR(NR)=0 THEN 550 !14
3
540 CALL SOUND(1000,30000,30
,30000,30,413,30,-4,0):: GOT
O 510 !119
550 CK(L)=NR :: IF CK(2)=CK(
1)THEN 540 !235
560 D=INT(NR/7)+INT(NR/7-INT
(NR/7)+.99):: R=D*4-3 :: C=(
NR-(D-1)*7)*4-3 :: CALL HCHA
R(R,C+2,P(NR),3):: CALL HCHA
R(R+1,C+2,P(NR),3)!164
570 CALL HCHAR(R+2,C+2,P(NR)
,3):: CX(L)=P(NR):: RR(L)=R
:: CC(L)=C :: TX(L)=NR :: NR
=0 !124
580 NEXT L :: CK(2)=0 :: IF
CX(1)=CX(2)THEN 630 !203
590 FOR D=1 TO (5-DF)*100 !0
15
600 NEXT D :: FOR L=1 TO 2 :
: R=RR(L):: C=CC(L)+2 :: N=T
X(L)!248
610 GOSUB 430 !255
620 NEXT L :: GOTO 480 !148
630 TR(TX(1))=1 :: TR(TX(2))
=1 :: CALL SOUND(100,392,5):
: CALL SOUND(100,440,5):: CA
LL SOUND(100,494,5):: CALL S
OUND(300,523,3,392,3,330,3)!
242
640 CALL SOUND(200,30000,30)
:: IF PL=1 THEN 660 !121
650 PY(W)=PY(W)+1 :: W=W-1 !
253
660 COUNT=COUNT+1 :: IF COUN
T=21 THEN 670 ELSE 480 !073
670 IF PL=2 THEN 690 !006
680 DISPLAY AT(4,1):"COMPLET
ED IN ";STR$(SX);" TRIES" ::
 GOTO 710 !014
690 IF PY(2)>PY(1)THEN 750 !
206
700 DISPLAY AT(4,1):"PLAYER
#1 WINS";PY(1);"TO";PY(2)!12
6
710 DISPLAY AT(8,1):"TO PLAY
 AGAIN PUSH ANY KEY" !045
720 CALL KEY(0,K,ST):: IF ST
<1 THEN 720 !192
730 PY(1)=0 :: PY(2)=0 :: CO
UNT=0 :: RESTORE 1 :: RESTOR
E 280 :: FOR X=1 TO 42 :: Z(
X)=0 :: TR(X)=0 !166
740 NEXT X :: CALL CLEAR ::
CALL COLOR(2,2,1,3,2,1,4,2,1
):: GOTO 120 !205
750 DISPLAY AT(4,1):"PLAYER
#2 WINS";PY(2);"TO";PY(1)::
GOTO 710 !024
760 X=1 :: FOR R=1 TO 21 STE
P 4 :: FOR C=3 TO 27 STEP 4
:: CALL HCHAR(R,C,P(X),3)::
CALL HCHAR(R+1,C,P(X),3):: C
ALL HCHAR(R+2,C,P(X),3):: X=
X+1 !239
770 NEXT C !217
780 NEXT R :: DISPLAY AT(4,1
):"TOUCH ANY KEY WHEN READY"
 !034
790 CALL KEY(0,K,ST):: IF ST
<1 THEN 790 !007
800 CALL HCHAR(4,3,32,28)::
RETURN !188
809 !@P+ !062
810 SUB TITLE(S,T$):: CALL S
CREEN(S):: L=LEN(T$):: CALL
MAGNIFY(2)!239
814 GOTO 820 !134
816 S,T$,L,J !122
818 CALL SCREEN :: CALL MAGN
IFY :: CALL SPRITE !232
819 !@P- !064
820 FOR J=1 TO L :: CALL SPR
ITE(#J,ASC(SEG$(T$,J,1)),J+1
-(J+1=S)+(J+1=S+13)+(J>14)*1
3,J*(170/L),10+J*(200/L))::
NEXT J !118
821 !@P+ !062
822 SUBEND !168
```

MDOS 2.0 warning

Gary R. Moore of Neosho, Missouri, passes along this tip:

I have a word of warning for all Geneve computer owners. Don Walden of Cecure Electronics told me not to use the file copy functions of MDOS 2.0. Instead use Clint Pulley's Directory Manager. It is great. I had MDOS totally corrupt just about half of my disks.

If I hadn't had several backups, I would have been a mess, as I use it in m mail order business. As it was, it took me a week to finally get all my files correct again!

(See Page 31)

USER NOTES

(Continued from Page 30)

Correction to relative files article

Here are program corrections from last month's article about relative files. Only the lines that were damaged are included.

INTCOPY

```
130 OPEN #2:"DSK1.INTWORDS",
DISPLAY ,RELATIVE,FIXED 56
140 PRINT #2,REC 0:100
150 FOR X=1 TO 100 :: DISPLAY
AT(6,20):X :: INPUT #1:R$
```

INTDATA

```
90 REM warning: This program
does not run properly. We ha
ve not been able to correct
it at this point.
91 REM Use TI-Writer to crea
INTWORDS database.
160 L=11 :: IF ASC(W$)<>42 T
HEN W$=" "
   W$=W$ :: RPT$(" ",L-LEN(
W$))
180 R$=R$ :: NEXT Y :: PRINT
#2,REC X:R$
```

WORDS

```
1000 ! ** words **
1001 DATA ALL,GANZ,TOUT,TUTT
O,TODO
1002 DATA APPLE,APFEL,POMME,
MELA,MANZANA
1005 DATA BACK,*RU#CKEN,DOS,
DORSO,ESPALDA
```

INTINDEX

```
100 ! ** INTINDEX: build Eng
lish index from INTWORDS **
110 DISPLAY AT(6,2)ERASE ALL
:"BUILDING ENGLISH INDEX:"
120 OPEN #2:"DSK1.INTWORDS",
DISPLAY ,RELATIVE,FIXED 56
```

MICROpendium pays $10 for items used in this column. Send them to MICROpendium User Notes, P.O. Box 1343, Round Rock, TX 768680.

Long-time TIer Joseph Turk dies

Joseph Turk, a long-time active member of the Sheboygan Area 99ers User Group died unexpectedly April 7 at the age of 77 of an apparent heart attack.

Turk was born on May 2, 1917, in Calumet, Michigan. He was married to the former Loraine Hoff of Milwaukee. He is survived by his wife, two sons, a daughter-in-law, four grandchildren, three great grandchildren, and a sister. His funeral was on April 11 at St. Dominic Parish in Sheboygan. Turk was buried at the Arlington Cemetery in Milwaukee.

Tom Wills, of the Southwest 99ers user group, said, "Joe was a real TIer. He was always happy, and smiling. I am personally very saddened by his death. He was a good friend. The real shock here is that Joe had not been ill before his death. In fact, he was trying to get onto the internet to converse with other TIers."

CLASSIFIEDS

FOR SALE

BUY-SELL-TRADE

Hardware — Software. TI Buyers Guide $2. Dual 1/2 HT Kit complete with cab les, screws, drill template & instructions $59. PE Box complete $149. Corcomp DD Cont $180. Star NX-1001 Printer $175. 2400 Baud Modem $79. TI99/4A Console $45. Rare — Myarc Hard & Floppy Controller with 10 Meg Hard Disk & Case $350. 800-471-1600 (Nationwide & Canada ORDERS ONLY). 414-672-1600 Local/tech Support. Huge Genuine TI Inventory Since 1982. Competition Computer, 2219 S. Muskego, Milwaukee, WI 53215. Fax 414-672-8977. Open Daily 9-5, Sat 10-3. Bankcards, Discover, Checks & UPS/COD.

TI99/4A

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

T.O.D. FANS!

All new Tunnels of Doom Adventure for your TI! A major revision with new graphics, spells, ploys, and surprises! It's called **Halls of Lost...MORIA!** and takes place in the dawn of the Fourth Age. Listen to these rave reviews: Justin V (age 10) Neat! It looks more like a dungeon, the fireballs are awsome! Amy V. (age 6) I think it's pretty! Diane V. (age ??) Hmm... if you cut wood, like that Dwarf chops Orcs, we wouldn't be worried about running short this winter.... Send $4.95 ($5.95 outside USA) for disk and booklet to: Michael Veprauskas, 32 Vaillancourt Dr., New Ipswich, NH 03071. v12n5

Classified ads are 10 cents per word

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

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

The ONLY monthly devoted to the TI99/4A

Subscription Fees

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

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

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

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

Address Changes

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

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

Check/MO ❑ MasterCard ❑ VISA ❑ (check one)

Card No. _____

Expiration Date _____

(Minimum credit card order is $9)

Signature _____

(Required on credit card orders.)

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

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

Name _____

Address _____

City _____

State _____ **ZIP** _____

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

Disks, Etc.

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

_____

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

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

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

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

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

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

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

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

GENEVE DISKS (SSSD unless specified)

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

❑ **GPL 1.5** $4.00

❑ **Myarc Disk Manager 1.50** $4.00

❑ **Myarc BASIC 3.0** $4.00

❑ **MY-Word V1.21** $4.00

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

GENEVE PUBLIC DOMAIN DISKS

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

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

SECOND CLASS

A T EAPINES 1/96
CHARLES GOLD
P.O. BOX 647
VENEDOCIA OH 45894