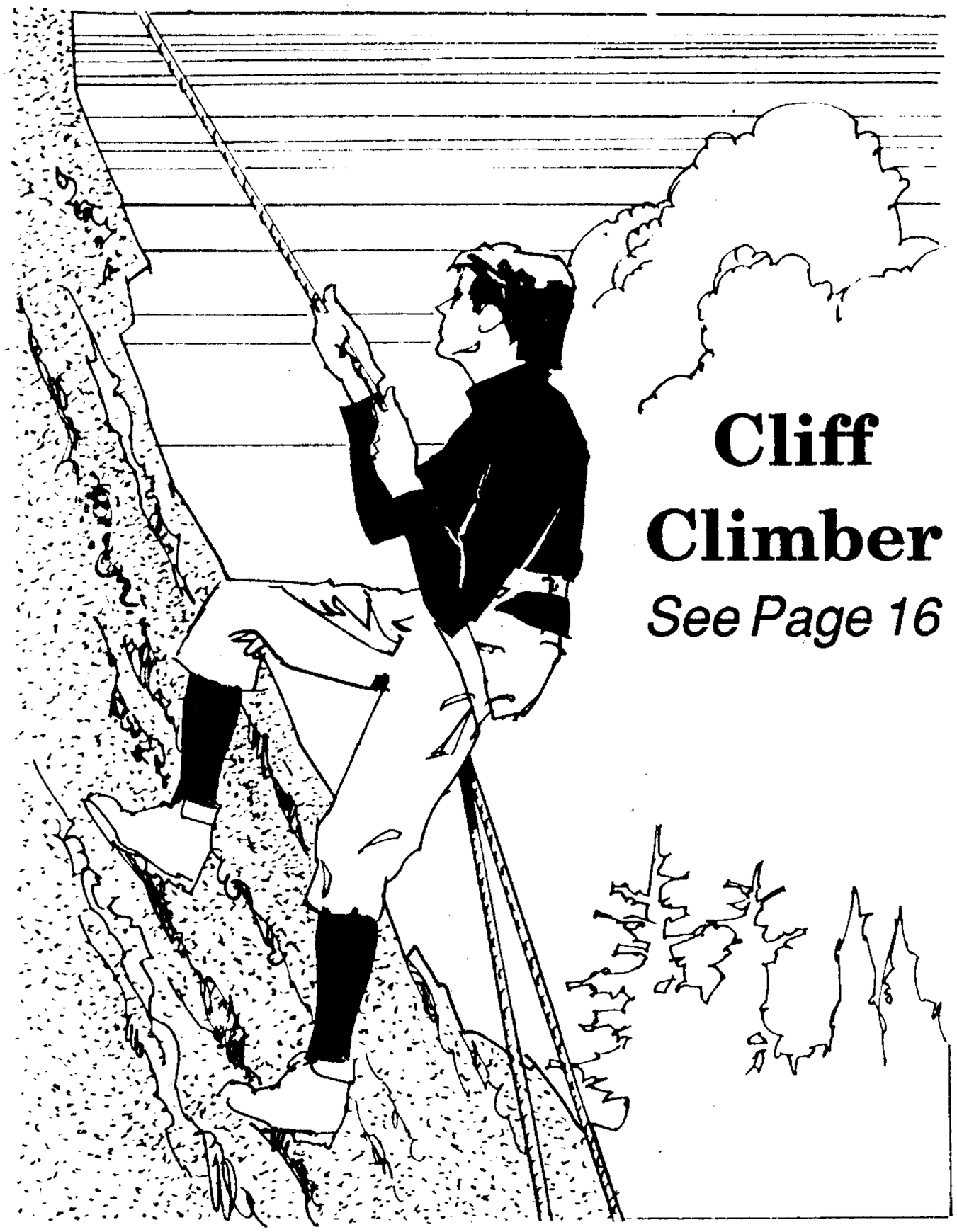
Covering the TI99/4A and the Myarc 9640

MICROpendium

Volume 10 Number 4 | May 1993 | $2.50



Cliff Climber

See Page 16

- International flags in BASIC
- Sorting out sorts in Assembly
- Report on RXB
- Reviews of:
  - Easy Data
  - The Panda
  - Expansion Box
  - Scout Museum Disk
  - O Say Can U See

And more of the news and tips for the TI you have depended on for 10 years.

Turning a TI into a burglar alarm

See page 19

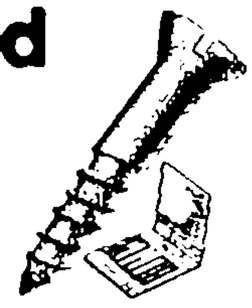
TEX-COMP

$$ DOLLAR DAYS $$

WATCH YOUR SAVINGS GROW DURING DOLLAR DAYS

OUR LOWEST PRICES EVER ON THIS ENTIRE PAGE!

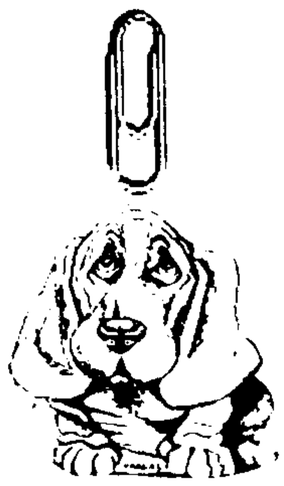
Big Price Reduction for Our Full-Featured COMPLETE PRINT SHOP PACKAGE FOR THE TI-99/4A special price $4.95*



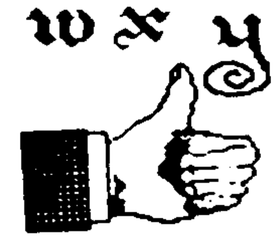
BUICK

featuring

GRAPHX + with Flip & Rotate & Quick Load (REG $19.95)



THE ALL TIME BEST GRAPHICS PROGRAM FOR THE 99/4A HAS JUST GOT EVEN BETTER WITH AN ALL NEW PACKAGE AT A GREAT LOW PRICE. ONLY $19.95 COMPLETE WITH GRAPHX+ AND TWO DISK SIDES FILLED WITH NEW CLIPART AND FONTS. GRAPHX+ OFFERS EVERY FEATURE YOU NEED TO CREATE THE BEST 4A GRAPHICS YOU HAVE EVER SEEN. GRAPHX+ OFFERS FEATURES SUCH AS FREE HAND DRAWING AND ERASING, ZOOM FOR DETAILED WORK, AUTO FILL, COLORS, TEXT AND TITLES, AUTO CIRCLE DRAWING, CLIP STORAGE AND ANIMATION. 32K, EXBASIC, AND A STAR/EPSON COMPATIBLE DOT MATRIX PRINTER IS REQUIRED.



NEW FONTS INCREDIBLE

*BUY ANY 4 ACCESSORY PACKAGES AT THE NEW LOWER PRICE OF $4.95 AND BUY THE COMPLETE GRAPHX+ PACKAGE FOR ONLY $1.00.

GRAPHX DINOSAURS!

$4.95

Asgard Software

DISK OF DINOSAURS is a fantastic collection of real and whimsical dinosaurs from that master 99/4A artist Ron Gilliland. This latest utility for Graphx also contains a bonus disk filled of the Dinosaurs" for creating great scenes along with all the other great clip art

Graphx with printout resolution far superior to TI-Artist along with it's nearly instantaneous "zoom", built-in move/copy function, multi speed cursor, and always available clipboard is, in the words of Chris Bobbitt of Asgard Software, "the best drawing tool available for the 99/4A".

Now you can "dino-saur-ize" anything from calendars to greeting cards to reports. Create dinosaur stickers, logos, and pictures faster than you can say "Brontosaurus". Can also be used with TI Artist with the conversion routine supplied. GRAPHX and 1 DS disk drive system required.

MAC-FLICK

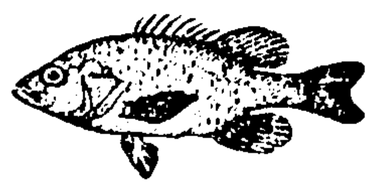
HIGH RESOLUTION CLIP ART FOR GRAPHX AND TI ARTIST

$4.95

TEX-COMP HAS TAKEN THE BEST MAC® CLIPART AVAILABLE AND CONVERTED IT TO GRAPHX CLIPART FILES. NOW YOU CAN PRODUCE GRAPHICS WITH YOUR 4A THAT RIVALS THE BEST ANYWHERE. WE FILLED SIX DISK SIDES WITH OVER 140 PIECES OF CLIPART THAT ARE READY TO BE PRINTED OUT WITH GRAPHX OR CAN EVEN BE CONVERTED TO OTHER GRAPHIC SYSTEMS. THE SUPERIOR PRINTING CAPABILITY OF GRAPHX WILL CREATE ART THAT WILL BE OUTSTANDING!!

SIX DISK SIDES GRAPHX, 32K, AND STAR/EPSON DOT MATRIX PRINTER REQ.

SIX DISK SIDES



ASGARD SOFTWARE HAS PUBLISHED FOUR SETS OF PROFESSIONAL CLIPART AND FONTS FOR USE WITH THE GRAPHX+ PUBLISHING SYSTEM SO YOU CAN CREATE FANTASTIC ARTWORK WITHOUT EVEN HAVING ARTISTIC TALENT. EACH SET IS ONLY $4.95 (VOL I-IV)

GRAPHX SLIDESHOW BY ASGARD IS A CLIPART COLLECTION AND A SPECIAL PROGRAM FOR DISPLAYING YOUR GRAPHX CREATIONS IN AUTOMATIC SEQUENCE. GREAT FOR DISPLAYS, TRADESHOWS AND PRESENTATIONS.....ONLY $4.95

PRINTING TO GO

exciting new pieces of Clip Art

For use with GRAPHX $4.95

OVER 130 PIECES OF CLIPART ON THREE DISK SIDES + FONT BONUS DISK

THE ULTIMATE GRAPHICS SOFTWARE

PRINTING TO GO IS A GREAT COLLECTION OF CLIP ART AND FONTS TO ADD TO YOUR GRAPHX PUBLISHING SYSTEM FOR THE TI-99/4A. THIS GREAT PACKAGE OF FOUR DISK SIDES AT A FANTASTIC PRICE IS ADDED PROOF THAT GRAPHX IS THE BEST BUY IN 4a GRAPHIC PACKAGES AND BY FAR, THE BEST SUPPORTED. THIS COLLECTION CAN BE USED FOR CREATING NEWLETTERS, ANNOUNCEMENTS OR ANY OTHER APPLICATION WHERE YOU NEED PROFESSIONAL LOOKING ART. THESE CLIPS CAN ALSO BE USED FOR CREATING PERSONALIZED STATIONARY OR EVEN FOR MAKING T-SHIRTS WITH THE TEX-COMP T-SHIRT KIT. OVER 130 PIECES OF QUALITY CLIPART + FONTS.

GRAPHX+ SYSTEM (reg. 19.95) special sale price	$4.95*
ASGARD GRAPHX COMPANION I	$4.95
ASGARD GRAPHX COMPANION II	$4.95
ASGARD GRAPHX COMPANION III	$4.95
ASGARD GRAPHX COMPANION IV	$4.95
ASGARD GRAPHX SLIDESHOW	$4.95
PRINTING TO GO (Clipart for GRAPHX reg 9.95)	$4.95
GRAPHX DINOSAURS (Dinosaur clipart reg. 12.95)	$4.95
MAC-FLICK (Hi-res. clipart for GRAPHX reg. 12.95	$4.95

*BUY ANY 4 ACCESSORY PACKAGES AT THE NEW LOWER PRICE OF $4.95 AND BUY THE COMPLETE GRAPHX+ PACKAGE (REG $19.95) FOR ONLY $1.00.

****

TEX-COMP

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

VISA and MASTERCARD HOLDERS CALL DIRECT:

(818) 366-6631

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

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

Send $2 for our new 1993 catalog

TEX-COMP

DOLLAR DAYS

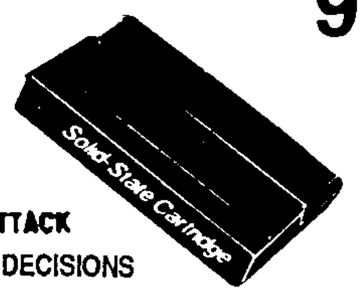
WATCH YOUR SAVINGS GROW DURING DOLLAR DAYS

SOFTWARE SUPER SALE!

AS LOW AS $1.95 EACH!!

PHM 3027 ADDITION & SUBTRACTION I
PHM 3028 ADDITION & SUBTRACTION II
PHM 3093 MILLIKEN DIVISION
PHM 3051 NUMERATION II
PHM 3094 MILLIKEN INTEGERS
PHM 3118 MINUS MISSION
PHM 3029 MULTIPLICATION I
PHM 3002 EARLY LEARNING FUN
PHM 3003 BEGINNING GRAMMAR
PHM 3112 PARSEC
PHM 3007 HOUSEHOLD BUDGET MANAGEMENT
PHM 3044 PERS. REPORT GENERATOR**
PHM 3053 TI INVADERS
PHM 3194 JAWBREAKERS II

PHM 3018 TAX INVESTMENT RECORD KEEPING***
PHM 3022 PERSONAL REAL ESTATE
PHM 3023 HUNT THE WUMPUS
PHM 3031 THE ATTACK
PHM 3220 MICROSURGEON
PHM 3057 MUNCHMAN
PHM 3110 CHISHOLM TRAIL
PHM 3219 SUPER DEMON ATTACK
PHM 3006 HOME FINANCIAL DECISIONS
PHM 3035 TERMINAL EMULATOR II
PHM 3052 TOMBSTONE CITY
PHM 3108 LOGO II* ⟶ Plus $4 additional shipping & handling
PHM 3055 EDITOR ASSEMBLER
PHM 3010 PHYSICAL FITNESS
PHM 3146 MUNCHMOBILE



ALSO INCLUDED ARE THE FOLLOWING ARARISOFT MODULES:
DEFENDER, PROTECTOR II, PICNIC PARANOIA

*ADD AN ADDITIONAL $4.00 EACH ON THESE THREE TITLES DUE TO WEIGHT
** FOR USE WITH PERSONAL RECORD KEEPING, *** DISK DRIVE REQUIRED
PERSONAL RECORD KEEPING AVAILABLE AT $14.95

Special Purchase

DONKEY KONG featuring Mario
$4.95 (regular $9.95)

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

99¢ BOOK BLOWOUT!!!

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

GAMES TIs PLAY
THE ELEMENTARY TI
COMPUTER PLAYGROUND (TI)
INTRO TO ASSEMBLY LANGUAGE
PROGRAMS FOR THE TI COMPUTER
TI-99/4A GAME PROGRAMS
STARTER PACK I
STARTER PACK II
GAME WRITERS PACK I
GAME WRITERS PACK II

****

SPECIAL PACKAGE AND PRICING ON Adventure

THE ADVENTURE SERIES...

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

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

ONLY $12.95 FOR ALL 13.

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

ONLY $9.95 FOR ALL 8

COMMAND MODULE FOR ABOVE ADVENTURES (reg $6.95) ONLY 95 CENTS WITH EITHER OF THE ABOVE SERIES

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

NEW First time on disk — Return to Pirate's Island with Graphics. XBASIC or Editor/Assembler Load $4.95

ADVENTURE EDITOR

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

SUPER BUY!

$4.95

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

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

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

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

Available NOW From TEX-COMP NEW LOW PRICE
Specify Mini Memory (Cassette) or Editor Assembler (Disk) Versions

$4.95 Per Disk

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

EDUCATION

DISKETTE PROGRAMS

PHD 5009 Music Skills Trainer ... 4.95
PHD 5011 Computer Music Box ... 4.95
PHD 5018 Market Simulation ... 4.95
PHD 5030 Spell & Spell II (Ex Basic req.) ... 9.95
PHD 5031 Spell & Math (TE-II req.) ... 4.95
PHD 5042 Spell Writer (TE-II req.) ... 4.95
PHD 5026 Bridge Bidding I ... 4.95
PHD 5039 Bridge Bidding II ... 4.95
PHD 5041 Bridge Bidding III ... 4.95
SPECIAL!! BRIDGE BIDDING I,II&III ... 9.95
PHD 5020 Music Maker Demo (use with module!) ... 4.95
PHD 6010 Mystery Melody (stop the music quiz) ... 4.95

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

PHT 6009 Music Skills Trainer ... 4.95
PHT 6010 Mystery Melody ... 4.95
PHT 6011 Computer Music Box ... 4.95
PHT 6018 Market Simulation ... 4.95
PHT 6031 Spell & Math ... 4.95
PHT 6042 Spell Writer ... 4.95
PHT 6026 Bridge Bidding I ... 4.95
PHT 6039 Bridge Bidding II ... 4.95
PHT 6041 Bridge Bidding III ... 4.95
SPECIAL!! BRIDGE BIDDING I,II&III ... 9.95
PHT 6020 Music Maker Demo (use with module) ... 4.95

MANAGEMENT AND BUSINESS

DISKETTE PROGRAMS

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

CASSETTE PROGRAMS

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

HOME ENTERTAINMENT

DISKETTE PROGRAMS

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

CASSETTE PROGRAMS

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

COMPUTER PROGRAMMING AIDS

DISKETTE PROGRAMS

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

CASSETTE PROGRAMS

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

MATH AND ENGINEERING

specify disk or tape with order

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

****

TEX-COMP

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

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

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

AUTHORIZED DEALER

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

Send $2 for our new 1993 catalog

Contents

MICROpendium

MICROpendium (ISSN 10432299) is published monthly for $25 per year by Burns-Koloen Communications Inc., 16506 Terrace Dr., Austin, TX 78728-1156. Second-class postage paid at Austin, Texas, and additional mailing offices. POSTMASTER: Send address changes to MICROpendium, P.O. Box 1343, Round Rock, TX 78680-1343.

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

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

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

Display advertising deadlines and rates are available upon request.

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

Foreign subscriptions are $30.25 (Mexico); $32.50 (Canada); $30.00, surface mail to other countries; $42 airmail to other countries.

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

Mailing address: P.O. Box 1343, Round Rock, TX 78680.
Telephone: (512) 255-1512
CompuServe: 75156,3270
Delphi TI NET: MICROpendium
GEnie: J.Koloen

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.

Comments

Trying to get RXB on a cartridge

Rich Gilbertson is a TI'er on a mission. Over the past two years he has developed several editions of his Rich GKXB, an extensive update of Extended BASIC. Unlike other versions of XBASIC, Gilbertson's RXB is compatible with all TI hardware and software. Unfortunately, it is available only for use with a GRAM device or a Geneve.

Those who are familiar with Super Extended BASIC know how XBASIC can be improved through the addition of subroutines. Gilbertson's RXB is similar in this regard, but offers far more power than even SXB. Unlike SXB, RXB is not available on cartridge. Not yet, at least.

Gilbertson would like to see his RXB available in cartridge format so that it can be used with any TI99/4A. But the problem has been finding a suitable hardware company to produce it. He says he has not found a single hardware company that is willing to put RXB on a cartridge without adding all kinds of embellishments, presumably to boost the retail price.

Rich says he has been talking with four possible cartridge producers and is hopeful that something will come of it. I hope so, too. Making the power of RXB available to all TI users would be a public service.

D.WRIGHT STUFF SELLS COMPANY

Delbert Wright, owner of D. Wright Stuff, has sold his business to Ricky Bottoms of Shelbyville, Indiana. No other details were immediately available.

NEW LOAD/SYS, MDOS RELEASED

Beery Miller of 9640 News has released a new version of LOAD/SYS on GEnie, Delphi and on his BBS. Miller has also released MDOS V1.50H, GPL 1.5H AND MDM5 V1.50. Phone number for the 9640 News BBS in Memphis, Tennessee, is (901) 368-0112.

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

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

We'll have more on this next month.

—JK

READER TO READER

❑ Harold W. Evans, 293 Circle Hills Dr., Grand Forks, ND 58201, writes:

I was exuberant, well, almost, when I encountered "Output TRACE to a printer" on page 30 of the December 1992 issue. Even with changing the D to a C in line 120, I could print *only* the TRACE line numbers for the subprogram.

I would like to use this subprogram. Would someone please let me know how I may make it print the TRACE line numbers for a program? Does it depend on the location of the GOSUB in the program?

❑ Ron Warfield, 216 10th Ave., New Westminster, British Columbia, Canada V3L 2B2, writes:

I have a Geneve computer and have purchased the IWD software package so I can digitize pictures, etc. My problem is the company that made the digitizer hardware does not make it anymore. Can anyone help with information or used hardware? Thank you in advance.

❑ Martin Zeddies, Haupstr. 26, DW-3180, Wolfsburg, Germany says he has had trouble creating a program with the TASM assembler to read out the MAPPER of the MDOS CPU RAM. He wants to know why he can start the program with the TDEBUG and it works correctly, but when he starts it from the MDOS CLI command line it gives impossible values to the monitor.

Reader to Reader is a column to put TI and Geneve users in contact with other users. Address questions to Reader to Reader, c/o MICROpendium, P.O. Box 1343, Round Rock, TX 78680.

1993 TI FAIRS

APRIL

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

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

MAY

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

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

OCTOBER

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

1994 FAIRS

FEBRUARY

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

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

Feedback

Praise for article

I do not keep up with the computer nearly as much as I used to but I still would like to keep up on what is going on. I especially like the articles like "Undocumented features of ACCEPT AT" by B.V. Takach in the November '92 issue. I used the information in a program I wrote in order to go back up a list and re-enter or correct entries.

Robert G. Wagner
Spring Valley, California

'Apathy,' 'appalling' are overly polite

Mr. Robert Carmany ("Apathy appalls," Feedback, March 1993) must be a nice guy. I say this because I think he was too easy in his remarks.

We, the Danville 99(ers), fully understand his frustrations. Over the years we have written to other groups and vendors and never hear from them. But we here in Danville want him to know that if he or any of his friends are 99/4A users, then we would like to hear from them. It makes no difference to us if he is a member of an active group or just an individual user, we care.

His system can be expanded beyond reason or no more than the basic 16K (with or without even a cassette), to us he is a 99/4A user. To all of the 99/4A world he should be important.

As long as people like Jim Peterson in Ohio, Dallas Phillips in the Louisville, Kentucky, area, Mr. Green of R.A.G. Software and the bunch of great guys in the Lexington, Kentucky, group are around, then Mr. Carmany is needed.

As long as Mr. Jim Krych in Corpus Christi, Texas, and developers like Asgard continue to work to make the 99/4A better, then all 99/4A groups and users need to stand together.

I don't know if Mr. Carmany wrote to the Bluegrass Area 99 group or the Kentuckiana (Louisville) group, but they have both been really nice to this little Danville group.

As a matter of fact, Mr. Carmany may have written to us, but if he used the address in the TI Users Group Guide, then that is why we never got his letter. In five years they have not changed that address. Most of the people that I have mentioned I have never met in person (except Dallas Phillips), but I know they are there keeping the 99/4A alive. It just seems crazy (no, stupid) for a group not to at least "acknowledge" a letter from another group, and more important than that is to take the time to answer letters from individuals who may be in areas where there is no organized group.

Of the 109 "members" of the Danville group, only eight are in the actual Danville area. The rest are our "mail" members and are treated just as if they were here with us. It is not just our group that needs these users, it is the entire 99/4A world. We need each other. We need MICROpendium. We need Jim Krych to be successful with his books on assembly.

We need people like Dallas Phillips, who attends the TI fairs and comes back and informs those of us who are unable to attend. Even though I do not have the means for Bruce Harrison's "music" software (at this time), I feel we need him. I have never met him, but for the first 30 years of my life I was raised on 42nd Place in Hyattsville, Maryland, and my wife is there now visiting her mother who has cancer. We must not continue to discourage people like Mr. Carmany. There is no doubt that this causes many to put their TI aside and forget it. And with this "don't answer the letter" attitude, the TI will surely die. This attitude will not in any way keep developers active, either. So if this sounds like a plea mixed with a little anger, then you are right. And to all of those people whom I did not mention because I do not know you, who are devoted to the 99/4A, I say thank you all.

And to Mr. Carmany, I say, write to us in Danville. You can count on hearing from us "quickly," my friend.

Gene Downs
The Danville 99(ers)
888 Airport Rd.
Danville, KY 40422

Group seeks users

In response to Mr. Robert M. Carmany's letter (Feedback, March 1993), our group (The Southern California Compute Group) based in San Diego is always looking for new members. We are a group dedicated to serving the needs of TI users exclusively. We publish a monthly newsletter, maintain a large library of TI shareware and public domain programs (available by mail to members) and a large document library on TI related topics. Anyone interested in joining our group may write to the following address for an application as well as a free sample issue of our newsletter, for more information: S.C.C.G., P.O. Box 152535, San Diego, CA 92195.

O.E. Peppe
Membership Chairman, S.C.C.G
San Diego, Californi

Numbers needed

How do you expect us to win the lotter if someone doesn't get the error correcte in the Lottery Program? (Joke.) Us user rely on you programmers, and option 1 sti produces the same number twice in some of the lines.

We are waiting with "baited breath (whatever that means) for the next issue of MICROpendium to see if the problem ca be corrected. If it is left up to the likes of me, we'll have to return to birthdays, an niversaries and other "slipshod" method of obtaining numbers.

Jack Lyon
Hearne, Texa

We hope you took note of the correctio in the April issue and that you plan t share your millions with the author an with us. We see the term "baited breath" on boards and in newsletters all the time but it really should be "bated breath"; i means breath held from excitement. Th word "bated" is related to "abate" and ha nothing to do with bait. — Ed.

Taking issue with Micro-Reviews

In the 10 years or so that I have been member of the TI99 community I hav made every effort to avoid becoming em broiled in controversy. With what I have t

(See Page 7)

Feedback

(Continued from Page 6)

say in this letter, that stance may change. I have a grave concern about the fairness, objectivity, thoroughness and usefulness of the Micro-Reviews column since Harry Brashear stopped doing the reviews. I am concerned enough to recommend that anyone reading the column *not* make a purchase decision based upon the ratings and the comment in any Micro-Review.

The people who write software for the 99/4A and 9640 have a right to a review that is more than just the writer's opinion, but even when opinion is all that is offered, it should at least be based upon a complete examination of the product. While I may be criticized for appearing to promote my own interests, I must point to the April 1993 Micro-Review of my Check Plus program as an example of the appalling disservice to supporters of the TI community.

In that Micro-Review the author states that he "*found this program difficult to use. Although there are 31 pages* (of laser printed) *documentation, you need to follow the steps the documentation outlines.*" Do I need to apologize for this? It sure sounds like it. Later the author of the Micro-Review states, "*I was unable to enter any information after the type of transaction and the date.*" Know why? Because the author of the review never read past page 2 in the documentation. It seems pretty obvious that little or no effort was made to read the instructions and understand the mechanics of the program, nor was any real effort made to evaluate the program.

As one of a dwindling number of people who still write software for the 99/4A, I feel that I have been penalized not because my product stinks, but because the author of the Micro-Review likes programs that don't require documentation. He states, "*I like programs that I can get right into and start to work.*" Fine! But don't ask anyone to make a purchase decision based upon your likes or dislikes. Let the product speak for itself.

At the end of the Micro-Review the author states that "*The price (of the program) was not made available to me.*" That is absolutely correct. The reason no price was provided was because I didn't intend for Check Plus to be reviewed by the Micro-Reviews columnist. I sent him Check Plus for comparison purposes only (and made that fact very clear in my letter), after buying another author's checkbook program based upon the glowing Micro-Review it received. The letter that accompanied Check Plus was openly critical of the glowing Micro-Review because the product received was nowhere near the stellar performer that the glowing Micro-Review made it out to be. My offer to the Micro-Review columnist was to compare Check Plus to the other author's program, both of which cost $15, and decide which of the two a user would rather have for his $15 investment. I truly regret providing that copy of Check Plus to this person now, for any future sales of the program will most certainly be nonexistent.

It is true Check Plus requires a second disk to store data files, but the reason is because the program provides 358 sectors of programming and on-disk help files that fill a SS/SD disk. I'm sorry that single drive owners must swap disks, but I'm not sorry I created the disk swapping requirement by offering a program that provides audit trail creation, bar graph creation, checkbook reconciliation, auto-insertion of data for recurring accounts, complete budget management capabilities for up to 10 income and 50 expense accounts, subfile creation and a lot more. Perhaps Check Plus is not for everyone because it offers too many features, but the discriminating buyer ought to be able to decide that based upon a complete evaluation of the product.

My understanding of the original intent of Micro-Reviews was to do a cursory report on a product (as opposed to a full-fledged review), not a cursory examination. If I've lost sight of the changing intent of this column, please let me know. In the meantime, how about hiring a Micro-Review columnist who has a little better understanding of the tremendous responsibility one takes on when doing product reviews? I haven't given up my day job to sell copies of Check Plus, but I still would like a fair shake when the product is reviewed. I, too, have been openly critical of products that I've reviewed for MICROpendium, but I like to think I have some objective rationale behind my ratings.

Bill Gaskill
Grand Junction, Colorado

Geneve's not dead, any more than TI was

To Feedback writer Frank D. Ormonde Jr. (January 1993) and his statement, "I believe that the Geneve is dead...." Isn't this what was said about the TI 10 years ago? To continue with your statement, "...so any article devoted to it does not interest me." Well, I for one, have no interest in Regena BASIC programs, but this magazine tries to support *all users* by providing a variety of topics, which is the glue that binds us together and allows this magazine to survive. Also, if you invested in a used PC/XT with 640K it would set you back about $300, and you would still be behind in the computing world.

Jim Uzzell
Key West, Florida

Function built in to MY-WORD

On page 29 of the January issue you described a Geneve trick which should work in MY-WORD.

I tried out the trick on my Geneve which I boot with MDOS 123F and loading MY-WORD in version 1.22b. In this version I can start MY-WORD with the EXEC13 program direct in MDOS.

I made the entries in the way the article told me and found out that the function is really built in in my version of MY-WORD and works also when you start the word processor with EXEC13.

Martin Zeddies
Wolfsburg, Germany

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

BASIC

International code flags

By REGENA

Years ago I wrote this program for a couple of other computers. With the TI graphics, it was fairly easy to convert to our TI computer. There are code flags that can be used by seamen of all nations. The code pennants and repeater pennants are used for signaling messages. The numbers are a trapezoidal shape. There are 26 square flags that represent the 26 letters of our alphabet, and these are presented in this program.

The first section of the program goes in alphabetic order and simply draws each flag. The letter is shown, and that letter must be pressed to continue. If you press the Enter key, you may get back to the main menu screen.

The second section of the program draws the flag as you press a letter. For example, if you press "W" the flag representing "W" is shown. Continue as long as you wish, and then press the Enter key to get back to the main menu screen.

The third section of the program is a quiz that draws the flag and you need to press the corresponding letter. After two incorrect guesses, the correct letter is printed, and that flag will appear again in the quiz. If you answer correctly, the flag will not be shown again. The quiz ends after all 26 flags have been correctly identified, or you may press the Enter key to end the quiz and get back to the main menu screen.

Many of the flags are made up of blocks of color that can be drawn with a solid square character. Lines 160-190 define the first two characters in Color Sets 9 and 10 to be solid squares. Lines 150-360 print the title screen while defining other characters which are used in diagonal lines and a circle. The characters in Color Sets 11 and 12 are only used for a black on yellow circle for the letter "I," so those colors are defined in Lines 220-230. Other characters are used with different colors.

Lines 370-470 present the main menu screen options and branch appropriately. Lines 480-550 are the subroutine used to press a key to start. Lines 560-630 are a subroutine to branch to a particular subroutine for drawing the flag for the letter, where J is a number from 1 to 26.

Lines 640-750 are the subroutine to draw the flag for each letter of the alphabet in order. Lines 680-740 go through the 26 letters. Lines 760-900 are the subroutine to draw the flag for the letter pressed. A little extra logic is used for detecting a capital letter or a key pressed with the Alpha Lock up.

Line 910-1410 are the subroutine for the quiz. Lines 970-990 initialize N(JJ), an array representing the 26 possible letters. As a flag is correctly identified, N(J) is set equal to zero so the flag will not be used again. Line 1010 initializes the number of guesses, GUESS, and Line 1090 increments GUESS for each trial. FLAG is used to determine if a flag has been incorrectly identified twice. After two incorrect guesses, the correct letter is shown.

Lines 1100-1190 detect the key pressed. Lines 1200-1280 are the procedure for an incorrect answer, and Lines 1290-1330 are the procedure for a correct answer. Lines 1350-1410 print a score, then wait until a key is pressed , before returning to the main menu screen.

Lines 1420-1450 are a subroutine to draw a 20x20 square of one color. This subroutine is used in drawing several of the flags. The rest of the program contains subroutines for drawing each of the flags. The variable M is used as an index in FOR-NEXT loops, and ROW and CC are row and column numbers which may be based on M.[

The first statement in each subroutine sets the colors in Color Set 9 for the two major colors in each flag. A few flags need a third color, so Color Set 10 is used. Some of the flags are alike except for the colors, so only the color set needs to be defined; then the drawing can be a subroutine within a subroutine.

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

INTERNATIONAL FLAGS

```
100 REM INTERNATIONAL FLAGS
110 REM BY REGENA
120 DIM N(26)
130 CALL CLEAR
140 CALL SCREEN(8)
150 PRINT " INTERNATIONAL CO
DE FLAGS"
160 CALL CHAR(96,"FFFFFFFFFF
FFFFFF")
170 CALL CHAR(104,"FFFFFFFFF
FFFFFFF")
180 CALL CHAR(97,"")
190 CALL CHAR(105,"")
200 CALL CHAR(35,"00183C7E7E
3C18")
210 CALL COLOR(1,12,1)
220 CALL COLOR(11,2,12)
230 CALL COLOR(12,2,12)
240 PRINT :"EACH LETTER OF T
HE ALPHABET IS REPRESENTED B
Y A FLAG."
250 CALL CHAR(98,"FFFEFCF8F0
E0C08")
260 CALL CHAR(99,"7F3F1F0F07
0301")
270 CALL CHAR(100,"0080C0E0F
0F8FCFE")
280 CALL CHAR(101,"0103070F1
F3F7FFF")
290 DATA FFFFFEFEFCFCF8F8,F0
F0E0E0C0C0808,00008080C0C0E0
E,F0F0F8F8FCFCFEFE,FFFEFCF8F
0E0C08,7F3F1F0F070301
300 FOR C=106 TO 127
310 READ C$
320 CALL CHAR(C,C$)
330 NEXT C
340 DATA 073FFFFFFFFFFFF,E0
```

(See Page 9)

REGENA ON BASIC—

(Continued from Page 8)

```
FCFFFFFFFFFFFF,00000003070F3
F7F,000000C0E0F0FCFE,0001030
7070F1F1F
350 DATA 0080C0E0E0F0F8F8,3F
3F7F7F7FFFFFFF,FCFCFEFEFFFFF
FFF,FFFFFF7F7F7F3F3F,FFFFFFF
EFEFEFCFC
360 DATA 1F1F0F07070301,F8F8
F0E0E0C08,7F3F1F0703,FEFCF8E
0C,FFFFFFFFFFFF3F07,FFFFFFFF
FFFFFCE
370 PRINT : :"CHOOSE:"
380 PRINT :"1 COMPLETE ALPHA
BET"
390 PRINT "2 CHOOSE LETTERS"
400 PRINT "3 QUIZ"
410 PRINT "4 END PROGRAM"
420 CALL SOUND(100,999,2)
430 CALL KEY(3,K,S)
440 IF (K<49)+(K>52)THEN 430
450 CALL CLEAR
460 CH=K-48
470 ON CH GOSUB 640,760,910,
3920
480 CALL CLEAR
490 GOTO 370
500 PRINT : :"PRESS ANY KEY
TO START."
510 CALL SOUND(100,999,2)
520 CALL KEY(3,K,S)
530 IF S<1 THEN 520
540 CALL CLEAR
550 RETURN
560 CALL CLEAR
570 IF CH=3 THEN 590
580 CALL HCHAR(11,7,J+64)
590 IF J>13 THEN 620
600 ON J GOSUB 1460,1610,165
0,1750,1840,1900,2010,2090,2
150,2290,2360,2390,2460
610 GOTO 630
620 ON J-13 GOSUB 2630,2750,
2820,2880,2910,2980,3010,309
0,3120,3150,3250,3510,3780
630 RETURN
640 PRINT "YOU WILL SEE EACH
 FLAG."
650 PRINT :"PRESS THE CORREC
T LETTER."
660 PRINT :"PRESS <#ENTER> T
O RETURN TO THE MAIN MENU SC
REEN."
670 GOSUB 500
680 FOR J=1 TO 26
690 GOSUB 560
700 PRINT "PRESS THIS LETTER
 OR #";
710 CALL KEY(3,K,S)
720 IF K=13 THEN 750
730 IF (K<>J+64)+(K<>J+96)=-
2 THEN 710
740 NEXT J
750 RETURN
760 PRINT "PRESS A LETTER FR
OM A TO Z."
770 PRINT :"THE CORRESPONDIN
G FLAG WILL BE SHOWN."
780 PRINT :"PRESS <#ENTER> T
O RETURN TO THE MAIN MENU SC
REEN."
790 PRINT : :"PRESS ANY LETT
ER TO START.": :
800 CALL KEY(3,K,S)
810 IF K=13 THEN 900
820 IF (K<65)+(K>122)THEN 80
0
830 IF (K>90)+(K<97)=-2 THEN
 800
840 J=K-64
850 IF J<33 THEN 870
860 J=J-32
870 GOSUB 560
880 PRINT "PRESS ANOTHER LET
TER OR #";
890 GOTO 800
900 RETURN
910 PRINT "YOU WILL SEE A FL
AG."
920 PRINT :"WHAT LETTER DOES
 IT":"REPRESENT?"
930 PRINT :"PRESS THE LETTER
."
940 PRINT : :"ALL 26 FLAGS W
ILL BE":"PRESENTED."
950 PRINT : :"PRESS <#ENTER>
 TO STOP THE  QUIZ AND GO BA
CK TO THE"
960 PRINT "MAIN MENU SCREEN.
"
970 FOR JJ=1 TO 26
980 N(JJ)=1
990 NEXT JJ
1000 GOSUB 500
1010 GUESS=0
1020 FOR T=1 TO 26
1030 RANDOMIZE
1040 J=INT(26*RND)+1
1050 IF N(J)=0 THEN 1040
1060 FLAG=0
1070 CALL CLEAR
1080 GOSUB 560
1090 GUESS=GUESS+1
1100 CALL SOUND(100,999,2)
1110 CALL KEY(3,K,S)
1120 CALL HCHAR(11,7,63)
1130 CALL HCHAR(11,7,32)
1140 IF K=13 THEN 1410
1150 IF (K<65)+(K>122)THEN 1
110
1160 IF (K>90)+(K<97)=-2 THE
N 1110
1170 IF K<91 THEN 1190
1180 K=K-32
1190 CALL HCHAR(11,7,K)
1200 IF K=J+64 THEN 1290
1210 CALL SOUND(100,165,2)
1220 CALL SOUND(150,131,2)
1230 FLAG=FLAG+1
1240 IF FLAG<2 THEN 1090
1250 CALL HCHAR(13,7,J+64)
1260 PRINT "PRESS ANY KEY.";
1270 CALL KEY(3,K,S)
1280 IF S=1 THEN 1040 ELSE 1
270
1290 CALL SOUND(100,262,2)
1300 CALL SOUND(100,330,2)
1310 CALL SOUND(100,392,2)
1320 CALL SOUND(300,523,2)
1330 N(J)=0
1340 NEXT T
1350 CALL CLEAR
1360 PRINT "OUT OF 26 FLAGS"
1370 PRINT :"YOUR GUESSES: "
;GUESS
1380 PRINT : : : :"PRESS ANY
 KEY."
1390 CALL KEY(3,K,S)
1400 IF S<1 THEN 1390
1410 RETURN
1420 FOR M=1 TO 20
1430 CALL HCHAR(M,11,96,20)
1440 NEXT M
1450 RETURN
1460 CALL COLOR(9,5,16)
1470 CALL COLOR(10,5,1)
1480 FOR M=11 TO 20
1490 CALL VCHAR(1,M,97,20)
1500 CALL VCHAR(1,M+10,104,2
0)
1510 NEXT M
1520 FOR M=1 TO 5
1530 ROW=2*(M-1)
1540 CALL HCHAR(9-ROW,25+M,1
```

(See Page 10)

REGENA ON BASIC—

(Continued from Page 9)

```
06)
1550 CALL HCHAR(10-ROW,25+M,
107)
1560 CALL VCHAR(11-ROW,25+M,
32,4*(M-1))
1570 CALL HCHAR(11+ROW,25+M,
108)
1580 CALL HCHAR(12+ROW,25+M,
109)
1590 NEXT M
1600 RETURN
1610 CALL COLOR(9,7,7)
1620 CALL COLOR(10,7,1)
1630 GOSUB 1480
1640 RETURN
1650 CALL COLOR(9,5,16)
1660 CALL COLOR(10,7,7)
1670 FOR M=1 TO 4
1680 CALL HCHAR(M,11,96,20)
1690 CALL HCHAR(M+4,11,97,20
)
1700 CALL HCHAR(M+8,11,104,2
0)
1710 CALL HCHAR(M+12,11,97,2
0)
1720 CALL HCHAR(M+16,11,96,2
0)
1730 NEXT M
1740 RETURN
1750 CALL COLOR(9,5,12)
1760 FOR M=1 TO 6
1770 CALL HCHAR(M,11,97,20)
1780 CALL HCHAR(M+6,11,96,20
)
1790 CALL HCHAR(M+14,11,97,2
0)
1800 NEXT M
1810 CALL HCHAR(13,11,96,20)
1820 CALL HCHAR(14,11,96,20)
1830 RETURN
1840 CALL COLOR(9,7,5)
1850 FOR M=1 TO 10
1860 CALL HCHAR(M,11,97,20)
1870 CALL HCHAR(M+10,11,96,2
0)
1880 NEXT M
1890 RETURN
1900 CALL COLOR(9,16,7)
1910 GOSUB 1420
1920 FOR M=1 TO 10
1930 CALL HCHAR(M,21-M,98)
1940 CALL HCHAR(M,20+M,99)
1950 CALL HCHAR(M,22-M,97,2*
(M-1))
1960 CALL HCHAR(21-M,21-M,10
0)
1970 CALL HCHAR(21-M,20+M,10
1)
1980 CALL HCHAR(21-M,22-M,97
,2*(M-1))
1990 NEXT M
2000 RETURN
2010 CALL COLOR(9,5,12)
2020 FOR M=1 TO 3
2030 FOR CC=6*(M-1)TO 6*(M-1
)+2
2040 CALL VCHAR(1,11+CC,97,1
8)
2050 CALL VCHAR(1,14+CC,96,1
8)
2060 NEXT CC
2070 NEXT M
2080 RETURN
2090 CALL COLOR(9,16,7)
2100 FOR M=1 TO 10
2110 CALL VCHAR(1,10+M,96,20
)
2120 CALL VCHAR(1,20+M,97,20
)
2130 NEXT M
2140 RETURN
2150 CALL COLOR(9,12,2)
2160 GOSUB 1420
2170 RESTORE 2220
2180 FOR M=1 TO 17
2190 READ ROW,CC,G
2200 CALL HCHAR(ROW,CC,G)
2210 NEXT M
2220 DATA 8,20,112,8,21,113,
8,22,115,9,23,117,10,23,119,
11,23,121,12,23,123,13,22,12
5,13,21,127
2230 DATA 13,20,126,13,19,12
4,12,18,122,11,18,120,10,18,
118,9,18,116,8,19,114,8,20,1
12
2240 CALL HCHAR(9,19,97,4)
2250 CALL HCHAR(10,19,97,4)
2260 CALL HCHAR(11,19,97,4)
2270 CALL HCHAR(12,19,97,4)
2280 RETURN
2290 CALL COLOR(9,5,16)
2300 FOR ROW=1 TO 6
2310 CALL HCHAR(ROW,11,96,18
)
2320 CALL HCHAR(ROW+6,11,97,
18)
2330 CALL HCHAR(ROW+12,11,96
,18)
2340 NEXT ROW
2350 RETURN
2360 CALL COLOR(9,12,5)
2370 GOSUB 2100
2380 RETURN
2390 CALL COLOR(9,12,2)
2400 GOSUB 1420
2410 FOR M=1 TO 10
2420 CALL HCHAR(M,21,97,10)
2430 CALL HCHAR(M+10,11,97,1
0)
2440 NEXT M
2450 RETURN
2460 CALL COLOR(9,5,16)
2470 GOSUB 1420
2480 CALL HCHAR(1,11,97)
2490 FOR M=2 TO 20
2500 CALL HCHAR(M,9+M,100)
2510 CALL HCHAR(M-1,10+M,99)
2520 CALL HCHAR(M,10+M,97)
2530 NEXT M
2540 CALL HCHAR(1,30,97)
2550 FOR M=2 TO 20
2560 CALL HCHAR(M-1,31-M,98)
2570 CALL HCHAR(M,32-M,101)
2580 CALL HCHAR(M,31-M,97)
2590 NEXT M
2600 CALL HCHAR(10,20,97)
2610 CALL HCHAR(11,21,97)
2620 RETURN
2630 CALL COLOR(9,5,16)
2640 FOR M=1 TO 11 STEP 10
2650 FOR ROW=1 TO 11 STEP 10
2660 FOR CC=1 TO 5
2670 CALL VCHAR(ROW,9+M+CC,9
6,5)
2680 CALL VCHAR(ROW,14+M+CC,
97,5)
2690 CALL VCHAR(ROW+5,9+M+CC
,97,5)
2700 CALL VCHAR(ROW+5,14+M+C
C,96,5)
2710 NEXT CC
2720 NEXT ROW
2730 NEXT M
2740 RETURN
2750 CALL COLOR(9,7,12)
2760 FOR M=1 TO 20
2770 CALL HCHAR(M,10+M,99)
2780 CALL HCHAR(M,11,97,M-1)
2790 CALL HCHAR(M,11+M,96,2
-M)
2800 NEXT M
2810 RETURN
```

(See Page 11)

REGENA ON BASIC—

(Continued from Page 10)

```
2820 CALL COLOR(9,5,16)
2830 GOSUB 1420
2840 FOR M=8 TO 13
2850 CALL HCHAR(M,18,97,6)
2860 NEXT M
2870 RETURN
2880 CALL COLOR(9,12,12)
2890 GOSUB 1420
2900 RETURN
2910 CALL COLOR(9,7,12)
2920 GOSUB 1420
2930 FOR M=8 TO 13
2940 CALL HCHAR(M,11,97,20)
2950 CALL VCHAR(1,M+10,97,20
)
2960 NEXT M
2970 RETURN
2980 CALL COLOR(9,16,5)
2990 GOSUB 2830
3000 RETURN
3010 CALL COLOR(9,7,16)
3020 CALL COLOR(10,5,16)
3030 FOR M=1 TO 6
3040 CALL VCHAR(1,10+M,96,18
-
3050 CALL VCHAR(1,16+M,97,18
)
3060 CALL VCHAR(1,22+M,104,1
8)
3070 NEXT M
3080 RETURN
3090 CALL COLOR(9,7,16)
3100 GOSUB 2400
3110 RETURN
3120 CALL COLOR(9,16,7)
3130 GOSUB 2470
3140 RETURN
3150 CALL COLOR(9,5,16)
3160 CALL COLOR(10,7,7)
3170 GOSUB 1420
3180 FOR M=5 TO 16
3190 CALL HCHAR(M,15,97,12)
3200 NEXT M
3210 FOR M=9 TO 12
3220 CALL HCHAR(M,19,104,4)
3230 NEXT M
3240 RETURN
3250 CALL COLOR(9,16,5)
3260 GOSUB 2920
3270 RETURN
3280 REM MID
3290 CALL HCHAR(MM,CC,98)
3300 CALL HCHAR(MM,CC+1,97,3
)
3310 CALL HCHAR(MM,CC+4,101)
3320 CC=CC-1
3330 RETURN
3340 REM LEFT
3350 GOSUB 3300
3360 CALL HCHAR(MM+1,11,97,2
)
3370 CALL HCHAR(MM+1,13,101)
3380 CALL HCHAR(MM+2,11,97)
3390 CALL HCHAR(MM+2,12,101)
3400 CALL HCHAR(MM+3,11,101)
3410 RETURN
3420 REM RIGHT
3430 CALL HCHAR(MM-4,30,98)
3440 CALL HCHAR(MM-3,29,98)
3450 CALL HCHAR(MM-3,30,97)
3460 CALL HCHAR(MM-2,28,98)
3470 CALL HCHAR(MM-2,29,97,2
)
3480 CALL HCHAR(MM-1,27,98)
3490 CALL HCHAR(MM-1,28,97,3
)
3500 RETURN
3510 CALL COLOR(9,12,7)
3520 GOSUB 1420
3530 CC=14
3540 FOR MM=1 TO 4
3550 GOSUB 3290
3560 NEXT MM
3570 GOSUB 3350
3580 CC=22
3590 FOR MM=1 TO 12
3600 GOSUB 3290
3610 NEXT MM
3620 GOSUB 3350
3630 MM=5
3640 CC=26
3650 GOSUB 3430
3660 FOR MM=5 TO 20
3670 GOSUB 3290
3680 NEXT MM
3690 CC=26
3700 MM=13
3710 GOSUB 3430
3720 FOR MM=13 TO 20
3730 GOSUB 3290
3740 NEXT MM
3750 MM=21
3760 GOSUB 3430
3770 RETURN
3780 CALL COLOR(9,5,12)
3790 CALL COLOR(10,5,7)
3800 GOSUB 1420
3810 FOR M=1 TO 10
3820 CALL HCHAR(M,10+M,100)
3830 CALL HCHAR(M,11+M,97,20
-2*M)
3840 CALL HCHAR(M,31-M,101)
3850 NEXT M
3860 FOR M=1 TO 10
3870 CALL HCHAR(10+M,21-M,11
0)
3880 CALL HCHAR(10+M,22-M,10
5,2*(M-1))
3890 CALL HCHAR(10+M,20+M,11
1)
3900 NEXT M
3910 RETURN
3920 CALL CLEAR
3930 END
```

Device ends ground-line surges

A new model ZS1800 surge protector by Zero Surge Inc. of Montclair, New Jersey, is said by its manufacturer to eliminate spikes and surges without relying on commonly used MOVs (metal oxide varistors). As a result, the manufacturer says, this surge protector, rated 15 amps, not only eliminates surges on the 120-volt hot wire, but also keeps the power-line ground circuit clean, eliminating diverted surges which MOVs routinely dump onto the ground line.

For information, contact Zero Surge Inc., 215 Glenridge Ave., Montclair, NJ 07042; phone (201) 744-1760; FAX (201) 744-1804. In Canada, contact Datacom Ltd., 32 James Speight Rd., Markham, Ontario, L3P 3G4; phone (416) 472-8128; FAX (416) 471-7384.

WHT recalls Turbo Video PAL

Western Horizon Technologies is recalling all Turbo Video upgrades because of a problem with applications not executing correctly, according to Don O'Neil of the company.

Individuals with upgrades need to supply their addresses and phone numbers to Western Horizon Technologies, 10225 Jean Ellen Dr., Gilroy, CA 95020, to receive a replacement Turbo Video PAL at no cost.

O'Neil says users should replace the Turbo Video PAL with their old PAL until further notice. For those who do not have a known functioning original PAL, WHT will ship one free of charge, O'Neil says. He says the company is working on a solution to the problem.

THE ART OF ASSEMBLY — PART 23

Sorting out sorts

By Bruce Harrison

Sorting things into alphabetical or numerical order is of course one of those things computers do better and faster than humans. At least that's what they're supposed to do. Sometimes, when we have run sorts in Extended Basic, it has seemed that the human could have beaten the machine at this task, particularly for string variables in arrays.

There are two keys to making an effective and efficient sort. First is having a quick and memory-efficient way of determining which of two things is bigger than the other. The second is to have an efficient "algorithm" for using that information to re-order the groups of numbers or strings we're dealing with. Recently, we've done some work on sorting, mainly aimed at helping the Extended Basic programmer who's frustrated with the slow response of Extended Basic in performing sorts.

USE WHAT'S THERE

There are helping routines already built into your TI, and in some instances these can speed things along dramatically. For sorting numbers, there are two possible ways. If the numbers are all integers in the range of 0 through 65535, one can simply use the compare instruction, then be careful about whether the sign of the numbers is taken into account. To include the sign, a JLT or JGT instruction is used after the comparison, while to ignore the sign, one uses JL or JH after the compare. Let's say for example that R9 and R10 are pointing to members of an array of integers, and that we want to find out whether the one pointed by R9 is bigger than the one pointed by R10. We could proceed like this:

```
C    *R9,*R10     Compare two words
JGT  BIGGER       If R9's word is bigger, jump
```

That would take the state of the sign bit into account, so that for example >8001 would not be bigger than >7FFF, but smaller, because it has a sign bit of one. To ignore sign, we'd use this:

```
C    *R9,*R10     Compare the integers
JH   BIGGER       If R9's logically higher, jump
```

In this case, >8001 compared to >7FFF would jump to BIGGER.

Of course this is the most simple example of comparing numbers. If the numbers are floating point numbers instead of simple integers, they can still be handled simply, by using the "Floating Point Compare" routine through the XMLLNK utility vector.

To do that, one must first know the correct address data to supply

(See Page 13)

THE ART OF ASSEMBLY—

(Continued from Page 12)

to the XMLLNK vector. There lies one of the "rubs" in this business. When one is working in "pure" Assembly, to be run under the E/A module, there is the address >0A00 for FCOMP to be supplied to the XMLLNK, which itself is REF'd in the source code. If on the other hand one is working on a routine to be linked from Extended Basic, both the XMLLNK itself and the address for FCOMP are different. In that case XMLLNK is at address >2018, and the FCOMP is at address >0D3A.

Thus in pure Assembly, we'd have:

```
        REF  XMLLNK      Required reference
        BLWP @XMLLNK     Use the utility
        DATA >0A00       Data for FCOMP routine
```

But if we're linked from XB, we'd need:

```
XMLLNK  EQU  >2018       XB's XML link vector ad-
dress

        BLWP @XMLLNK     Use the utility
        DATA >0D3A       Address for FCOMP routine
```

It's maddening, this kind of thing, especially if you're trying to make a program that will run in either XB or E/A environments, as we often do. The XMLLNK utility for E/A is different from the one used by XB. In E/A's version, it uses the >0A00 to access a lookup table, where it gets the address >0D3A for the FCOMP routine. Things get even more complex for some other routines accessed by XMLLNK, but this one is bad enough to illustrate the problem.

In either case, the FCOMP routine requires that one of its two numbers be located at FAC (>834A), and the other at ARG (>835C). Each of course must be eight bytes representing a floating point number in Radix 100 form. Once the comparison is done, one must examine the GPL Status byte at >837C to determine the results of the comparison. Only two bits of that byte are important, and they must be examined separately. One way of doing this is to put the Status byte into two registers, then mask off all but the bits we want. For example:

```
        BLWP @XMLLNK     Use utility
        DATA FCOMP       Floating point comparison
        MOVB @>837C,R5   Move GPL status byte to R5
        MOVB @>837C,R6   And to R6
        ANDI R5,>4000    Mask all but the "greater" bit
        JNE  BIGGER      If not zero, jump
        ANDI R6,>2000    Mask all but "equal" bit
        JNE EQUAL        If not zero, jump
```

This code will jump to BIGGER if the number at ARG is arithmetically bigger than the number at FAC, and will jump to label EQUAL if they're equal. If both tests fail, then the number at ARG is smaller than the number at FAC.

What is done at labels BIGGER and EQUAL is of course another problem, but remember it's important to isolate the individual bits from the STATUS byte as we've shown above. In many instances you can eliminate the business of finding out if the two are equal, since that may not matter, depending how your sort proceeds from there. Often the EQUAL status can be ignored to save time and trouble.

(See Page 14)

SIDEBAR 23

```
* TEST SORT ROUTINE
* BY B. HARRISON
* 18 JUN 1992
* PUBLIC DOMAIN
*
      REF  VMBW,GPLLNK,KSCAN  REFERENCE UTILITY VECTORS
      DEF  SORTEM       DEFINE ENTRY POINT
*
* CODE SECTION
*
SORTEM
      LWPI WS           LOAD OUR WORKSPACE
      CLR  @COUNT       CLEAR COUNTER
      LI   R4,STSTR     GET ADDRESS OF START OF STRING ARRAY
      MOV  R4,@LOWEND   MOVE THAT TO LOW END ADDRESS STORAGE
      MOV  R4,@LEAST    AND FOR NOW INTO VARIABLE LEAST
COUNTM
      MOVB *R4+,R5      GET LENGTH BYTE OF A STRING
      SRL  R5,8         SHIFT TO RIGHT JUSTIFY IN R5
      JEQ  STSCAN       IF ZERO, WE'RE AT END OF ARRAY
      INC  @COUNT       ELSE INCREMENT COUNT
      A    R5,R4        ADD LENGTH TO ADDRESS POINTER
      JMP  COUNTM       JUMP BACK FOR NEXT STRING
STSCAN MOV  @LOWEND,R9   POINT R9 AT LOW END OF UNSORTED STRINGS
NEXT   MOV  R9,R10      MOVE THAT ADDRESS TO R10
      MOVB *R9,R7       GET THE LENGTH OF FIRST STRING IN R7
      SRL  R7,8         RIGHT JUSTIFY
      A    R7,R10       ADD TO R10
      INC  R10          INCREMENT SO R10 POINTS TO LENGTH OF NEXT STRING
MOV9   MOV  R9,R14      SAVE PRESENT R9 IN R14
      MOV  R10,R15      SAVE PRESENT R10 IN R15
      MOVB *R9+,R4      GET LENGTH FIRST STRING IN R4
      MOVB *R10+,R5     LENGTH OF SECOND IN R5
      SRL  R4,8         RIGHT JUSTIFY
      JEQ  MOVIT        IF ZERO, WE'RE THROUGH SCANNING
      SRL  R5,8         RIGHT JUSTIFY
      JEQ  MOVIT        IF ZERO, WE'RE THROUGH SCANNING
CMP910 CB   *R9+,*R10+  COMPARE THE BYTES POINTED BY R9 AND R10
      JGT  BIG          IF R9'S BIGGER, JUMP
      JLT  LESS         IF R9'S LESS, JUMP
      DEC  R4           DECREMENT COUNT
      JNE  DEC5         IF NOT ZERO, DECREMENT R5
      CI   R5,1         ELSE SEE IF R5=1
      JEQ  BIG          IF SO, STRINGS ARE EQUAL
      JMP  LESS         ELSE STRING POINTED BY R9 IS LESS
DEC5   DEC  R5          DECREMENT OTHER COUNT
      JNE  CMP910       IF NOT ZERO, COMPARE NEXT BYTE
BIG    MOV  R15,@LEAST  STASH OLD ADDRESS AT LEAST TABLE LOCATION
      MOV  R15,R9       RESET R9 TO POINT AT CURRENT LEAST STRING
      JMP  NEXT         THEN JUMP TO CONTINUE SCAN
LESS   MOV  R14,@LEAST  STASH OLD ADDRESS AT LEAST TABLE LOCATION
      MOV  R14,R9       RESET R9 TO PRESENT LEAST STRING
      MOV  R15,R10      GET OLD R10 BACK
      MOVB *R10+,R7     GET STRING LENGTH INTO R7
      SRL  R7,8         RIGHT JUSTIFY
      A    R7,R10       ADD LENGTH TO POINTER
      JMP  MOV9         THEN JUMP BACK TO TEST NEXT STRING
* THE SECTION BEGINNING AT MOVIT TAKES THE LEAST STRING FROM THIS SCAN, MOVES IT
* UP TO THE BEGINNING OF THE ARRAY
MOVIT
      MOV  @LEAST,R9    GET ADDRESS OF LEAST STRING
      LI   R10,TEMSTR   POINT AT TEMPORARY STORAGE LOCATION
      MOVB *R9,R4       GET LENGTH OF LEAST INTO R4
      SRL  R4,8         RIGHT JUSTIFY
      INC  R4           INCREMENT TO INCLUDE LENGTH BYTE ITSELF
      MOV  R4,R8        STASH IN R8
      MOV  R4,R5        AND IN R5
      C    R9,@LOWEND   IS THIS ALREADY IN THE RIGHT PLACE?
      JEQ  NOMOVE       IF SO, SKIP MOVING IT
MOVOUT MOVB *R9+,*R10+  ELSE MOVE ONE BYTE TO TEMPORARY STORAGE
      DEC  R4           DECREMENT COUNT OF BYTES
      JNE  MOVOUT       IF NOT ZERO, MOVE ANOTHER
*
* THIS NEXT SECTION MOVES ALL STRINGS ABOVE THE LEAST IN THE UNSORTED ARRAY DOWN
* TO MAKE ROOM FOR THE LEAST TO GO TO THE TOP OF THE UNSORTED PART OF THE ARRAY
*
      MOV  R9,R10       PUT ADDRESS BEYOND END OF LEAST STRING INTO R10
      MOV  @LEAST,R9    PUT LEAST ADDRESS BACK IN R9
      MOV  R9,R4        STASH THAT IN R4
      DEC  R9           POINT AT BYTE JUST BEFORE LEAST'S LENGTH
```

ART OF ASSEMBLY—

MAKE YOUR OWN

For strings, there is no equivalent to the FCOMP we used for floating point numbers, so one must compare byte by byte, walking through the string by incrementing pointers. Our convention is to use R9 and R10 as pointers, and to use the auto-increment option in the compare instruction. Thus to compare one byte of each string and move the pointers to the next byte, we simply:

```
CB   *R9+,*R10+        Compare one byte
```

In most cases involving strings, we don't have characters beyond 127 to worry about, so a simple JGT or JLT operation will be used as the decision making process. In today's sidebar is a complete program that shows a string comparison.

WHAT TO DO ABOUT IT

Once we know whether a particular string or number is bigger than another, we must decide what to do next, or how to proceed with the sorting. One could, for example, proceed with a "bubble" sort, in which each successive pair of the array is compared, and pairs are switched with one another if required. This takes many passes through the array to get it all sorted. Our preferred method, which was used in the MULTISORT routine for our Easy Data disk, is to scan through the entire array once, find either the biggest or the smallest, then re-arrange things to put that member where it belongs, and proceed to repeat this process until all members have been put where they belong. Today's sidebar shows a complete program example which sorts an array of 75 strings in just about one and a half seconds. That's fast. Our MULTISORT routine, which interfaces with XB DATA statements, sorts 55 records of six fields each by two criteria in about 3 1/2 seconds. Try sorting that in XB. Takes forever.

In the MULTISORT routine, we were able to unload our variables into the care of XB by the NUMASG and STRASG links. If we're working in pure Assembly, that luxury is not available to us, so we need another way of handling the "where to put it" problem. Were we working in the PC, for example, we could assign a complete 64K byte segment of memory to put our sorted array in, and simply leave the original data alone in its original segment of memory.

On the TI, we don't have that luxury, so we must do something different. If we're dealing with Floating Point numbers, re-arranging them in an array is made simpler because they all have the same length (8 bytes each). With Strings, the length byte gives us the length of each string in the array, but of course each may have a different length. In the sidebar is one "suggested" way of handling this particular problem, while making efficient use of memory.

This sort algorithm scans through all the strings in the "array," and finds the address of the least of the strings. It then stores that string in a temporary buffer (TEMSTR). Next it takes all the strings that are physically before that one in memory, and moves that entire block down by the length of the string found plus one. It's plus one to account for the length byte. Now the pointer for the start of the "unsorted" array is incremented by that same amount, and the "least" string is moved in at the head of the list. This process continues until there's only one string left, at which

(See Page 15)

```
       DEC  R10          AND AT LAST BYTE OF LEAST STRING IN ARRAY
       S    @LOWEND,R4   SUBTRACT BOTTOM OF UNSORTED PORTION
MOVREV MOVB *R9,*R10     MOVE ONE BYTE UPWARDS
       DEC  R9           DECREMENT R9 POINTER
       DEC  R10          DECREMENT R10 POINTER
       DEC  R4           DECREMENT COUNT OF BYTES TO MOVE
       JNE  MOVREV       IF NOT ZERO, REPEAT
*
* AT THIS POINT ALL THE STRINGS THAT WERE ABOVE THE LEAST HAVE BEEN MOVED
* DOWNWARD IN THE ARRAY BY THE LENGTH OF THE LEAST STRING AND ITS LENGTH BYTE
*
       LI   R9,TEMSTR    POINT AT TEMPORARY STORAGE
       MOV  @LOWEND,R10  AND AT BOTTOM OF UNSORTED LIST
MOVIN  MOVB *R9+,*R10+   MOVE ONE BYTE OF LEAST STRING INTO PLACE
       DEC  R5           DECREMENT COUNT
       JNE  MOVIN        IF NOT ZERO, MOVE ANOTHER
NOMOVE A    R8,@LOWEND   ADD THE LENGTH OF LEAST TO ADDRESS POINTER
       DEC  @COUNT       DECREMENT NUMBER OF STRINGS LEFT TO SORT
       MOV  @COUNT,R4    MOVE THAT TO R4
       CI   R4,1         COMPARE TO ONE
       JGT  STSCAN       IF MORE THAN ONE LEFT, START ANOTHER SCAN
*
* IF THERE'S ONLY ONE STRING LEFT, WE'RE FINISHED SORTING, BECAUSE THAT ONE IS
* ALREADY IN THE CORRECT PLACE
*
       LI   R1,STSTR     POINT AT START OF SORTED ARRAY
       CLR  R4           CLEAR R4
SHOW   MOVB *R1+,R2      GET LENGTH BYTE INTO R2
       JEQ  EXIT         IF ZERO, ALL HAVE BEEN DISPLAYED
       SRL  R2,8         ELSE RIGHT JUSTIFY
       LI   R0,23*32+2   POINT AT ROW 24, COL 3
       BLWP @VMBW        WRITE THE STRING TO SCREEN
       CLR  @>837C       CLEAR GPL STATUS BYTE
       BLWP @GPLLNK      USE GPLLNK
       DATA >4D00        TO SCROLL SCREEN UP ONE ROW
       INC  R4           INCREMENT COUNT OF STRINGS DISPLAYED
       CI   R4,23        ARE 23 ON SCREEN?
       JLT  SHOWON       IF LESS, CONTINUE DISPLAYING
KEY    BLWP @KSCAN       ELSE SCAN THE KEYBOARD
       CB   @ANYKEY,@>837C HAS A KEY BEEN STRUCK?
       JNE  KEY          IF NOT, SCAN AGAIN
       CLR  R4           ELSE CLEAR COUNT OF DISPLAYED STRINGS
SHOWON A    R2,R1        ADD LENGTH TO POINTER IN R1
       JMP  SHOW         THEN JUMP BACK TO SHOW NEXT STRING
EXIT   BLWP @KSCAN       SCAN THE KEYBOARD
       CB   @ANYKEY,@>837C HAS A KEY BEEN STRUCK?
       JNE  EXIT         IF NOT, RE-SCAN KEYBOARD
       LWPI >83E0        ELSE LOAD GPL WORKSPACE
       B    @>6A         THEN RETURN TO E/A CONTROL
*
* DATA SECTION
*
WS     BSS  32           OUR WORKSPACE
COUNT  DATA 0            NUMBER OF STRINGS IN ARRAY
LEAST  DATA 0            ADDRESS OF LEAST STILL IN UNSORTED LIST
LOWEND DATA 0            BOTTOM OF UNSORTED LIST
TEMSTR BSS  256          TEMPORARY STORAGE FOR STRING TO BE PLACED
ANYKEY BYTE >20          BYTE FOR KEYSTROKE DETECTION
*
* FROM HERE ON ARE THE STRINGS TO BE SORTED
* THERE MUST BE A ZERO BYTE AT END TO INDICATE END OF ARRAY
* NULL STRINGS MUST NOT BE INCLUDED IN THE LIST
*
STSTR  BYTE 6            LENGTH OF FIRST STRING
       TEXT 'ZEBRAS'     CONTENT
       BYTE 9
       TEXT 'AARDVARKS'
       BYTE 7
       TEXT 'ANIMALS'
       BYTE 9
       TEXT 'YESTERDAY'
       BYTE 9
       TEXT 'COMPUTERS'
       BYTE 7
       TEXT 'WINDAGE'
       BYTE 14
       TEXT 'DRIVING SCHOOL'
       BYTE 11
       TEXT 'DAILY PAPER'
       BYTE 12
       TEXT 'COMPUTATIONS'
       BYTE 7
```

ART OF ASSEMBLY—

(Continued from Page 14)

point we know they're all in correct order.

THE ORDERED TABLE

Another approach to the process, which doesn't take as much time, is to first build a table of the addresses of all the strings, then re-arrange this table instead of the strings themselves. Of course this method requires that enough memory be available for both the strings themselves and the table of addresses. That approach was used to an extent in the MULTISORT routine, where a table of addresses for the actual content of the DATA in the XB program was compiled in high memory, in the space left unused by the XB program. In a pinch, the address table can be built in VDP Ram instead of the normal expansion memory.

THE PRESORT CONCEPT

We've so far ignored the idea of where the strings or numbers you're sorting came from in the first place. In the case of MULTISORT, the data comes from the Extended Basic program. In the test program shown in the sidebar with this article, the string data was embedded in the source code itself, so it would be instantly available once the program was finished loading. In many cases, however, the data will either be entered from the keyboard or brought in from a file on disk. In such cases, it's possible to do the sorting "on the fly" as the records are entered or read from the disk. We call this process "presort", because it

(See Page 16)

```
      TEXT 'FRIENDS'                BYTE 9
      BYTE 9                        TEXT 'BEAUTIFUL'
      TEXT 'XYLOPHONE'              BYTE 8
      BYTE 7                        TEXT 'UNICORNS'
      TEXT 'MONITOR'                BYTE 7
      BYTE 10                       TEXT 'JAGUARS'
      TEXT 'TELEVISION'             BYTE 7
      BYTE 6                        TEXT 'JOINTLY'
      TEXT 'FROSTY'                 BYTE 6
      BYTE 10                       TEXT 'MARCEL'
      TEXT 'MICROPHONE'             BYTE 8
      BYTE 6                        TEXT 'JEAN-GUY'
      TEXT 'K-MART'                 BYTE 9
      BYTE 8                        TEXT 'POTHOLDER'
      TEXT 'CIRCULAR'               BYTE 6
      BYTE 8                        TEXT 'VERITY'
      TEXT 'SYMPHONY'               BYTE 9
      BYTE 8                        TEXT 'VERITABLE'
      TEXT 'CONCERTO'               BYTE 8
      BYTE 8                        TEXT 'DREADFUL'
      TEXT 'ASSEMBLY'               BYTE 7
      BYTE 9                        TEXT 'WEAVING'
      TEXT 'ANTIPHONS'              BYTE 8
      BYTE 8                        TEXT 'SCRUPLES'
      TEXT 'CALOMINE'               BYTE 8
      BYTE 12                       TEXT 'OPTIONAL'
      TEXT 'FRIENDLINESS'           BYTE 6
      BYTE 10                       TEXT 'XANADU'
      TEXT 'SATURATION'             BYTE 9
      BYTE 10                       TEXT 'INTENSIVE'
      TEXT 'BRIGHTNESS'             BYTE 9
      BYTE 7                        TEXT 'WATERGATE'
      TEXT 'ANTENNA'                BYTE 6
      BYTE 7                        TEXT 'MOVIES'
      TEXT 'COLUMNS'                BYTE 6
      BYTE 5                        TEXT 'BABIES'
      TEXT 'ROWED'                  BYTE 8
```

FEST WEST—

(Continued from Page 15)

sorts as the data is received, so that as soon as all the data is entered or read, the records are in correct order. We've done this for XB programmers with little subroutines that are on our Utilities Volume 3 disk.

The complete program shown in today's sidebar can be typed in, assembled, and run as an Option 3 program under E/A. The program name is SORTEM. As we said, this will sort its 75 self-contained strings in about 1½ seconds. Of course once it has run, the strings are in their proper order, so re-running will in effect not do any sorting. To perform the sort again, one must re-load the program from Option 3. This program should be easy enough to understand from the annotated source code. We think this will serve as a good example of how to perform a quick and memory-efficient sort.

We hope this month's column will inspire some of our readers to take a new look at the idea of sorting data with Assembly routines. Perhaps some creative thinking on your part will improve on our methods, and make life easier for all the rest of us in the TI community.

```
        TEXT 'CHILDREN'                 TEXT 'RESEARCH'
        BYTE 8                          BYTE 9
        TEXT 'CHILDISH'                 TEXT 'POLYANDRY'
        BYTE 10                         BYTE 9
        TEXT 'PERCENTAGE'               TEXT 'MARVELOUS'
        BYTE 8                          BYTE 11
        TEXT 'RELIABLE'                 TEXT 'GRANDMOTHER'
        BYTE 9                          BYTE 9
        TEXT 'CAPRICORN'                TEXT 'GREATNESS'
        BYTE 12                         BYTE 12
        TEXT 'MOBILIZATION'             TEXT 'FAITHFULNESS'
        BYTE 9                          BYTE 8
        TEXT 'ANTIQUITY'                TEXT 'MIRTHFUL'
        BYTE 10                         BYTE 9
        TEXT 'BARBARIANS'               TEXT 'MERRIMENT'
        BYTE 8                          BYTE 9
        TEXT 'SANDOVAL'                 TEXT 'NATURALLY'
        BYTE 9                          BYTE 11
        TEXT 'SAN DIEGO'                TEXT 'OZONE LAYER'
        BYTE 9                          BYTE 11
        TEXT 'OBLIVIOUS'                TEXT 'MASTERPIECE'
        BYTE 7                          BYTE 10
        TEXT 'ANCIENT'                  TEXT 'DEDICATION'
        BYTE 8                          BYTE 11
        TEXT 'POLYGONS'                 TEXT 'NONETHELESS'
        BYTE 10                         BYTE 9
        TEXT 'MARGARITAS'               TEXT 'SUITCASES'
        BYTE 9                  ENDSTR BYTE 0           ZERO LENGTH
        TEXT 'MINISTERS'        BYTE IS END OF ARRAY
        BYTE 8                          END
```

Cliff Climber is Extended BASIC update of original

By LUCIE DORAIS

This article originally appeared in the newsletter of the Ontario TI99/4A User Group.—Ed.

A true user group collaboration: the original game Cliff Climber was written several years ago by Mike Ward in TI-BASIC. As he said to me, "it just cried out to be made into Extended BASIC." So, here it is.

A man has to climb a rocky mountain while avoiding gaps, drops and a falling rock. The translation to Extended BASIC was made by the youngest member of our Ottawa TI99/4A Users Group, Jeffrey Brown. He also added the difficulty levels. My main contribution is in the aesthetic field, and transforming the falling rock into a sprite. I also found a compromise for a fat random screen display: instead of having rocks and gaps created randomly before each new game, random rows are built only when you change levels. After that, the random function is used to display the rows. The loop is thus played only 23 times instead of 644 (23x28) times.

In the original game, there was one falling rock added for each level, and they fell all together: at level five, for example, you had to avoid a row of five rocks stuck together (five columns wide). Transforming the rock into a sprite shows the limits of Extended BASIC, at least if you want your program to stay within reasonable length limits and play reasonably fast. Controlling more than one sprite, especially their coincidence (you don't get killed because two rocks collided), was to complicated to implement. So, in a way, the game is easier because there is only one rock falling, though it changes location and speed. But try the "Professional" and "Impossible" levels — you'll see that it is not so easy anymore.

The game is played with joystick No. 1 and is for one or two players. Hence the DIM SCore, Level and Men in line 110. The other variables dimensioned are the 23 screen rows and the four (0-3) difficulty level names.

```
100 ! ***** CLIFF CLIMBER **
** M. Ward 1987 / rev. J. Br
onw/L. Dorais Feb. 1993 / Ot
tawa UG !226
110 DIM SC(2),L(2),M(2),SCR$
(23),DIFF$(3):: CALL COLOR(1
3,2,15)!162
120 GOTO 150 :: A$,AL,C,CDIF
,DIFF,DIFL,G,HS,J,K,NP,P,Q,R
,RC,RM,RR,S,X,Y,Z :: CALL CL
EAR :: CALL SCREEN :: CALL K
EY :: CALL JOYST !124
130 CALL CHAR :: CALL HCHAR
:: CALL GCHAR :: CALL COLOR
:: CALL SPRITE !089
140 CALL LOCATE :: CALL POSI
TION :: CALL COINC :: CALL P
ATTERN :: CALL MOTION :: CAL
L DELSPRITE :: CALL SOUND ::
 !@P- !129
150 DIFF$(0)="EASY" :: DIFF$
(1)="INTERMEDIATE" :: DIFF$(
2)="PROFESSIONAL" :: DIFF$(3
)="IMPOSSIBLE!" !173
160 CALL CHAR(128,"000814224
2420100",129,"",140,"FFFFFFF
```

(See Page 18)

star micronics Printer Bonus!

Buy A Star Printer And Get . . .

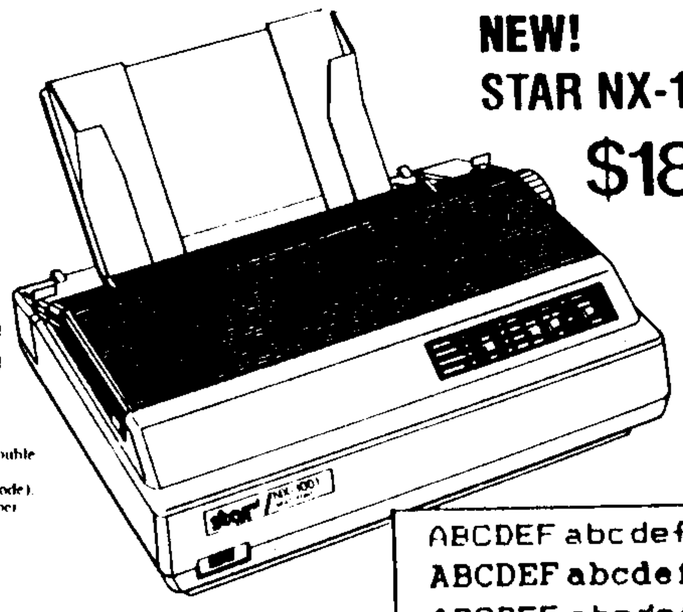
- The Most Advanced Printhead
- Letter Quality Type
- High Speed Operation
- Sophisticated Paper Handling
- Full TI® Compatibility

MULTI-FONT DOT-MATRIX PRINTER

A feature-rich 9-pin printer offering versatile typeface selection and superior ease of use at an unbelievable price!

- 9-pin parallel printer
- Printing styles: Draft, Courier, Sanserif, Orator and Orator-2 in a choice of 10 character sizes with enhancements such as italics, emphasized, double strike, double high, quadruple high and super/subscripts modes
- 180 characters/second (draft mode), 75 characters/second (near letter quality mode)
- Both single and continuous paper handling capability with auto loading and paper parking functions
- IBM and Epson software modes
- Simplified, user-friendly control panel
- 2-year Limited Warranty

(Requires RS-232 interface and parallel cable or ParaPort Interface.)



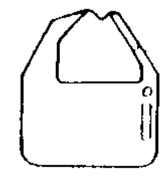
NEW!
STAR NX-1001
$189.95 +S&H

```
ABCDEF abcdef 0123
ABCDEF abcdef 0123
ABCDEF abcdef 0123
ABCDEF ABCDEF 0123
ABCDEF abcdef 0123
```

Plus $100 In FREE

Accessories, Software & Discounts.

TEX-COMP has a great deal in store for you when you buy a STAR printer from us. With your purchase you will receive a free package of printer accessories and discounts on printer related software for the TI-99/4A with a total value of over $100. For starters, you will receive free a $19.95 copy holder, a $9.95 copy of 99 Writer II, the disk version of TI Writer, a $19.95 copy of Better Banners and the $9.95 Better Banners Companion, a $4.95 copy of the TEX-Comp STAR Demo disk, a $7.95 spare ribbon and a $9.95 Head brand printer care kit. In addition, you also receive a $5 discount on a printer stand (reg $14.95), printer cable (reg.$24.95), and a RS232 card or CorComp PIO printer interface. All discounts must be taken at the time of the printer is ordered.



copy holder

■ Here's the printer with the features you've been waiting for.

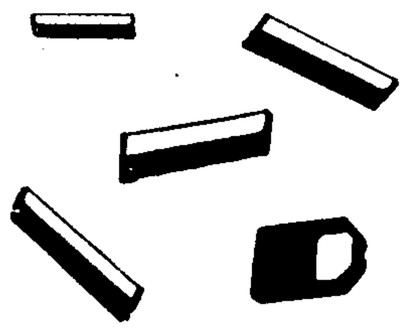
PRINTER CABLES

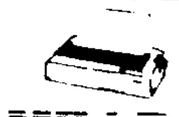
PARALLEL PRINTER CABLE..$24.95
(FROM PIO PORT ON RS232

Printer Ribbons

Be sure to specify your printer make & model with your order.

STAR NX1000.....EACH..$7.95
STAR NX10.....EACH..$7.95
STAR GEMINI, 10, 10X & SG10.....EACH..$3.95
TI IMPACT PRINTER (EPSON MX-80).....EACH..$7.95
AXIOM GP550 NEW LOWER PRICES.....2-PAK$19.95





HEAD BRAND

PRINTER CLEANING KIT

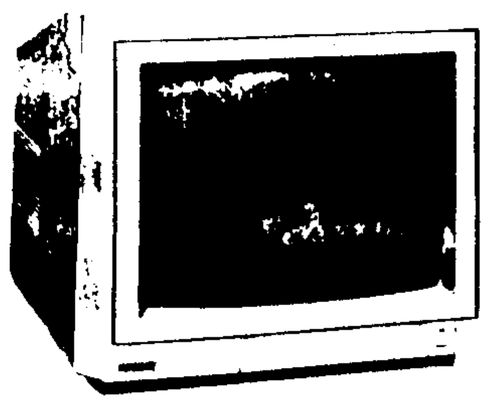
A dot matrix and daisy wheel printer head cleaner from HEAD Computer Products quickly cleans your dot matrix printer heads and daisy wheels to restore clean, crisp printing. Just insert the special cleaning sheet provided in the kit and apply a few drops of the cleaning solution. You actually see the accumulated ink and dirt disappear before your eyes. One kit lasts for dozens of cleanings and will actually extend the life of your printer heads and daisy wheels. Works with all computer printers using dot matrix or daisywheel printer heads. Not intended for thermal or ink jet printers.

NEW LOW PRICE $4.95

****

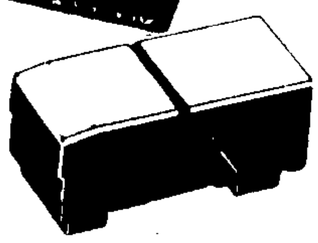
THE NEW MAGNAVOX 14" COMPOSITE/RGB/ANALOG COLOR MONITOR IS THE ONLY MONITOR THAT WORKS WITH EVERY TI SYSTEM INCLUDING THE GENEVE. FEATURES INCLUDE DUAL SPEAKERS AND STEREO AMP SO YOU CAN EVEN HOOK IT UP TO A STEREO VCR OR A LASER DISK PLAYER FOR FANTASTIC PICTURE & SOUND. WE INCLUDE A TI AND A IBM CABLE. WITH A TUNER OR VCR YOU ALSO HAVE A SECOND TV. COMES WITH A 2 YEAR WARRANTY FROM PHILIPS.

MAGNAVOX
RGB/Composite Color Monitor 80



INCLUDES TI-99/4A MONITOR CABLE!! **$279.95** +$10S&H

Two Year Parts and Labor Warranty • solid evidence that this is a reliable product, nationally supported by Magnavox.



COMES WITH 5 FREE DISKS A $25 VALUE!

****

THE ORIGINAL TI EXTENDED BASIC IS AN ABSOLUTE MUST FOR ANY 99\4A SYSTEM. COMPLETE WITH THE ORIGINAL MANUAL. WE WERE ONLY ABLE TO GET A LIMITED QUANTITY OF THESE NEWLY PRODUCED MODULES. WITH EACH MODULE WE INCLUDE YOUR CHOICE OF 5 FREE DISKS FROM OUR FREEWARE COLLECTION, A $25 VALUE.

INCLUDES 5 FREE FREEWARE DISKS .....ONLY $64.95 +s&h

$25 Free Disk Bonus

****

THE SPEECH SYNTHESIZER IS BY FAR, THE TI ACCESSORY THAT IS THE MOST DIFFICULT TO LOCATE. WE WERE ABLE TO GET HOLD OF A A VERY LIMITED QUANTITY. WITH EACH SYNTHESIZER WE INCLUDE 5 FREE DISK FROM OUR FREEWARE COLLECTION. CHOOSE FROM ONE OF THE MANY DISKS WITH SPEECH CAPABILITY.

COMES WITH 5 FREE DISKS .....ONLY $59.95 +s&h

$25 Free Disk Bonus

****

TEX-COMP™

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

VISA and MASTERCARD HOLDERS CALL DIRECT:

ORDER BY PHONE
24 HOURS A DAY
7 Days a Week!

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

AUTHORIZED DEALER

(818) 366-6631

CLIFF CLIMBER—

(Continued from Page 16)

```
FFFFF8080",141,"8080808080808080",142,"F0F0FEFEFEFE8E8E")!mountain, winner flag !097
170 CALL CHAR(64,"005A5A3C18182442",130,"0058583C1A7A4402",131,"001A1A3C585E2240",132,"003C7E7E7E7E3C")! man, rock !202
180 CALL CLEAR :: CALL SCREEN(15):: DISPLAY AT(1,7):"CLIFF CLIMBER" :: IF AL THEN 200 ELSE AL=1 !238
190 DISPLAY AT(5,1):"Use Joystick #1": :"JUMP ROW with FIRE button": :"RELEASE ALPHA-LOCK..." :: ACCEPT AT(9,24):A$ !241
200 DISPLAY AT(5,1):"HIGH SCORE:";HS: :"LAST SCORE PLAYER 1:";SC(1): :"LAST SCORE PLAYER 2:";SC(2)!244
210 DISPLAY AT(15,1):"Press <SPACE> to change":"DIFFICULTY level": : :"Press <1> or <2> for number":"      of PLAYERS": :"Press <Q> to QUIT" !121
220 DISPLAY AT(13,1):"DIFFICULTY: "&DIFF$(DIFL):: CDIF=DIFF !159
230 CALL KEY(0,K,S):: IF S=0 THEN 230 :: IF K=81 OR K=113 THEN END !202
240 IF K<>32 THEN 250 ELSE DIFL=DIFL+1 :: IF DIFL>3 THEN DIFL=0 :: GOTO 220 ELSE 220 !098
250 IF K<49 OR K>50 THEN 230 ELSE DIFF=20+DIFL*20 :: IF DIFF=CDIF THEN 300 !004
260 DISPLAY AT(24,1)ERASE ALL:DIFF$(DIFL)&" mountain data" :: RANDOMIZE !030
270 FOR R=1 TO 23 :: SCR$(R)="" :: FOR C=1 TO 28 :: G=129+(INT(RND*100)>DIFF)!093
280 SCR$(R)=SCR$(R)&CHR$(G):: NEXT C :: DISPLAY AT(24-R,1):SCR$(R):: NEXT R !077
290 ! ===== start ===== !115
300 L(1),L(2)=1 :: NP=K-48 :: SC(1),SC(2)=0 :: M(1),M(2)=5 :: Q=-1 !129
310 FOR P=1 TO NP :: IF M(P)=0 THEN 600 !125
320 CALL CLEAR :: CALL SCREEN(3):: CALL HCHAR(1,3,64,M(P)-1)!175
330 DISPLAY AT(1,9):"LEVEL";L(P):: IF NP=2 THEN DISPLAY AT(1,21):"PLAYER";P !157
340 FOR X=220 TO 880 STEP 110 :: CALL SOUND(50,X,0):: NEXT X :: FOR X=880 TO 220 STEP -110 :: CALL SOUND(-100,X,0):: NEXT X !251
350 RANDOMIZE :: FOR R=24 TO 2 STEP -1 :: C=INT(RND*23)+1 :: DISPLAY AT(R,1):SCR$(C):: NEXT R ! mountain !169
360 C=INT(RND*27)+3 :: CALL GCHAR(24,C,G):: IF G=129 THEN 350 ELSE CALL SPRITE(#1,130,2,185,C*8-7):: R=24 ! man at bottom !185
370 RC=INT(RND*216)+17 :: RM=INT(RND*10)+5*L(P):: CALL SPRITE(#2,132,2,9,RC,RM,0)! rock !035
380 ! ===== climb ===== !076
390 CALL JOYST(1,X,Y):: IF X=0 AND Y=0 THEN 480 !187
400 J=1 :: CALL KEY(1,K,S):: IF K=18 THEN J=2 ! jump one row with fire !016
410 CALL COINC(ALL,Z):: IF Z THEN 550 !061
420 R=R-SGN(Y)*J :: IF R>=25 THEN R=24 !191
430 SC(P)=SC(P)+Y :: IF R<=1 THEN 520 !208
440 C=C+SGN(X)*J :: IF C>2 OR C<31 THEN 460 ELSE C=C-SGN(X)*J !006
450 CALL COINC(ALL,Z):: IF Z THEN 550 !061
460 CALL GCHAR(R,C,G):: CALL SOUND(-10,-6,0):: CALL PATTERN(#1,130+(Q=-1)*-1):: Q=-Q !011
470 CALL LOCATE(#1,R*8-7,C*8-7):: IF G<>128 THEN 550 ! climb !011
480 CALL COINC(ALL,Z):: IF Z THEN 550 ELSE CALL POSITION(#2,RR,RC):: IF RR<193 THEN 500 ELSE CALL DELSPRITE(#2)! check rock row !203
490 RC=INT(RND*216)+17 :: RM=INT(RND*10)+5*L(P):: CALL SPRITE(#2,132,2,9,RC,RM,0)! change rock column/speed !151
500 CALL COINC(ALL,Z):: IF Z THEN 550 ELSE 390 !014
510 ! ===== wins level ===== !062
520 CALL DELSPRITE(ALL)!115
530 FOR X=330 TO 770 STEP 110 :: CALL SOUND(-500,X,0):: NEXT X :: L(P)=L(P)+1 :: IF L(P)>15 THEN 620 ELSE 320 !213
540 ! ===== man falls (hit by rock or reaches hole) ===== !247
550 CALL MOTION(#1,20,0)!048
560 CALL POSITION(#1,R,C):: IF R<193 THEN CALL SOUND(-100,320-R,0):: GOTO 560 ELSE CALL DELSPRITE(ALL)! man falls !201
570 M(P)=M(P)-1 :: IF M(P)<>0 THEN 600 :: CALL HCHAR(8,10,32,12):: CALL HCHAR(9,10,32,12):: CALL HCHAR(10,10,32,12)!052
580 DISPLAY AT(9,9)SIZE(-10):"GAME OVER!" :: IF NP=2 THEN DISPLAY AT(12,9)SIZE(-10):"  PLAYER "&STR$(P)&"  " !014
590 FOR X=1 TO 20 :: CALL SOUND(-100,RND*200+110,0):: NEXT X :: IF HS>SC(P)THEN 600 ELSE HS=SC(P)!017
600 NEXT P :: IF M(1)=0 AND M(2)=0 THEN 180 ELSE IF M(1)=0 AND NP=1 THEN 180 ELSE 310 !218
610 ! ===== total win ===== !215
620 FOR R=1 TO 6 :: DISPLAY AT(R,1):RPT$(CHR$(128),R):: NEXT R :: DISPLAY AT(3,8):"CONGRATULATIONS!" !060
630 CALL HCHAR(6,10,64):: CALL HCHAR(5,11,140):: CALL HCHAR(6,11,141)!024
640 IF NP=1 THEN A$="   YOU WIN!" ELSE A$="PLAYER "&STR$(P)&" WINS" !159
650 Q=-1 :: DISPLAY AT(5,11)SIZE(-13):A$ :: DISPLAY AT(2
```

(See Page 19)

CLIFF CLIMBER—

(Continued from Page 18)

```
4,1):" Press FIRE or ANY KEY
..." !141
660 Q=-Q :: CALL SOUND(-100,
RND*800+110,0):: CALL HCHAR(
5,11,140+Q=-1)*-2)!013
670 CALL KEY(0,K,S):: CALL K
EY(1,X,Y):: IF S=0 AND Y=0 T
HEN 660 !246
680 IF HS>SC(1)THEN 690 ELSE
 HS=SC(1)!224
690 IF HS>SC(2)THEN 180 ELSE
 HS=SC(2):: GOTO 180 !103
```

Excuse the long pre-scan, due to the use of sprites. The characters defined by lines 130-140 are the rocks and gaps on the mountain (128-129), a flag that will be waved if you win (140-142), a standing man and two climbing men (64, 130-131), and the falling rock (132). Then some instructions (shown only once, thanks to the AL variable) and the main menu with high and last scores, difficulty level, and keys to press. The difficulty routine is entirely due to Jeff, except for the percentage values, which I changed a bit to make the EASY screen easier. To change the difficulty level, you press the space bar: the next DIFF$() value is shown (it rolls over) by line 220, referred back from lines 240, and the current DIFF level is kept into CDIF so that you do not have to witness a new random screen building if you keep to the same level. To start the game, you must press 1 or 2 for the number of players.

The DIFFiculty level is calculated each time you change levels (line 250) as 20, 40, 60 and 80. (Jeff's original formula was "40+DIFL*10", for results of 40, 50, 60 and 70.) Lines 260-280 will build your data according to the DIFF level: if it is 20 (EASY), the formula "(INT(RND*100)>DIFF)" will be true (-1) about 80 percent of the time and you will get 80 percent rocks vs 20 percent gaps. If it is 80 (IMPOSSIBLE), you get 20 percent rocks and 80 percent gaps. Good luck on this one! Each new row built is put on screen so you can witness Tex's slow progress.

The new game really starts in line 300 by resetting the Level, Number of Players, Scores, Men (there were originally three, I added two to give myself a chance to win), starting Q value used to change the climber's CHAR pattern later. Then the loop runs according to the Number of Players, P keeping the current Player number. The top row of the screen shows how many men are left (line 320) and the Level; if there are two players, it also gives the current Player number. Each new screen starts with music and the rows are built by lines 260-280 are displayed randomly by line 350. The starting position of the climber is picked by line 360 (always a rock, never a gap). The falling rock is put on screen at random column and random speed (which increases with each level) by line 370.

The JOYSTick is then CALLed. If there is no move, Tex is sent directly to line 480 to check the COINCidence — it is checked often because it is hard to get one if the rock falls too rapidly — and the rock's position. When the rock reaches the bottom of the screen, its column and speed are again changed (line 490; at fast falling rates it sometimes misses that line and the rock falls a few times in the same column. You can try to improve that). If you press the FIRE button (K=18), the value of Jump, originally one, is set to two by line 400. You can jump only one row or column at a time, so don't try it to avoid big gaps. Another COINC check, and then the new climber's Row is calculated, the sign +/- (SGN) of the Y value telling Tex to move him up or down. The Score is incremented each time you move up, decremented if you climb down. I have not fully studied the score routine devised by Mike, I hope I explained it correctly. The new Column is then calculated if you move sideways (you can also jump a column here; line 440). More COINC check, then Tex looks at the character at the new Row or Column. It then changes the man's PATtern (130 or 131 depending on Q), makes some scratching sound, and LOCATES him at the new position.

If the character GCHARed in line 460 is not a rock (gap or drop at the side borders), too bad for you: lines 550-560 will send your climber to the ground with appropriate sounds until he reaches the bottom of the screen. The number of Men is decremented; if zero, "GAME OVER!", Score and High Score reset, time for next Player or next game. The same routine is, of course, used when a COINCidence between ALL sprites (man No. 1 and rock No. 2) is detected.

If you manage to reach the top of the screen alive, line 430 will send you to line 520 for the trumpets of victory and a higher Level. If you have done all 15 levels, you get a special treat: near the mountain's summit, the victorious climber will wave a flag for you (lines 620-670). The Q variable is again used to change the flag chardef (140 or 142), creating the waving motion (line 660) until you press a key or the fire button. The CALL KEY value of zero in line 670 checks the keyboard, the value of 1 checks the joystick. The High Scores are reset before you can start a new game.

'White elephant' console turns into home security system

By BERNARD ZUCKERMAN

The following item appeared in the newsletter of the Cleveland Area TI99/4A User Group. We are publishing it here because it demonstrates what a little ingenuity and a TI console can do.—Ed.

It all started way back in 1990 when I ordered a disk from the user group library. I don't remember the program I wanted but on the disk was a program written by R.E. Lunsden of Winnipeg, Ontario, called BURGLARM. It used the joystick port wired to the perimeter contacts and keyswitch of a house burglar alarm system. The monitor gave all the instructions and the resulting audio (beeps, noise, siren) were appropriate for a burglar alarm. It was fascinat-

(See Page 20)

BURGLAR ALARM—

(Continued from Page 19)

ing to hook up and even more interesting to work out the logic. After a while, the disk was stored away and almost forgotten.

Last spring my wife and I went to a white elephant sale. She usually looks around at the kitchen gadgets and books, while I dig into the camera, electronics, etc. shelves. Lo and behold, there on a back shelf, covered with years of dust, was a TI99/4A console — no cables, no power supply, no modules, no nothing. Just a bar console. How can a true TIer resist? For $1 it was mine.

"What in heaven's name are you going to do with another computer?" was my dear wife's comment.

"Oh, it probably doesn't work so I'll give it to Ken for parts, or I'll make it into a burglar alarm."

Wow! That idea came out of the blue. And the console worked fine.

THE TI IS THE BRAIN

But saying "make it into a burglar alarm" and making it work was another story. In its elementary form, a burglar alarm system consists of perimeter wiring with contacts —NO or NC (Normally Open or Normally Closed) — at each door, window, etc. and a "keyswitch" to arm or disarm the system. The "brains" must be able to distinguish between a legitimate opening and a "break-in" and send or give an alarm when appropriate. In my case, the brain is to be the computer.

Lunsden wrote a program that makes the console into a burglar alarm brain. It checks the perimeter, the keyswitch and provides delays for coming and going. The CALL SOUND command made all the noise, beeps and sirens via the monitor, and the instructions were all displayed on-screen.

Well and good. But I don't need the monitor if I'm not at home, and I do not want to keep my PEB on while I take my Hawaiian vacation for three weeks. And I do not want to activate my automatic dialer to call the police. So I set up the parameters of my project:

- A stand alone console — no PEB, no monitor, no modules.
- It must be able to sound a siren or loud horn, as well as communicate to me, the operator.
- It must provide a NO contact which will close and start my automatic phone dialer.
- It must signal me if the window or door is open or if it is not properly armed.
- No joysticks allowed.

A LITTLE HELP FROM A FRIEND

Well, I can eliminate the monitor by connecting the audio output to a free-standing audio amplifier and speaker and use the CALL SOUND to send beeps, notes, songs and other noises as signals as to what is going on. A cassette will load the program and then be removed. But how do I get an output that will close a contact (energize a relay)?

That's where Ken Gladyszewski came into the picture. At one of our meetings I discussed the project with him and he picked up on it. Inside of a month he designed and built an electronic circuit which, on a CALL JOYST(2,x,y) command energized a relay and closed a set of contacts. *See the companion article for the circuit and how it works.—Ed.*

A audio amplifier integrated circuit from Radio Shack worked fine and the program was modified (see below) to provide the brains.

I broke one part of my original goal. Since I had a spare Speech Synthesizer, I revised the program so that the computer tells me what to do. And it all worked!

But what if the power goes off?

BURGLARM4A

```
100 REM BURGLAR ALARM PROGRA
M, ELECTRONICS BY KEN G., PR
OGRAM BY LUNDSDEN, REV. BY B
.Z. !057
101 REM FILE BURGLARM4A-BASI
C, NO VIDEO, SPEAKING AND SI
REN !012
110 REM Y IS KEYSWITCH; X IS
 PERIMETER.Y=+4 K.S. CLOSED;
 X=-4 PERIMETER CLOSED. !035
120 REM START WITH KEY SWITC
H A NO PERIMETER SWITCH OPEN
. !252
130 ENDEL=3000 !181
140 EXDEL=3000 !191
150 SKIPD=1 !051
160 SW=1 !098
170 CALL JOYST(1,X,Y)!129
180 GOTO 440 !008
190 PRINT #1:"O K THANKS O K
 THANKS" !138
200 CALL JOYST(1,X,Y)!129
210 IF Y=0 THEN 600 !103
220 IF SKIPD>1 THEN 250 !044
230 GOSUB 410 !235
240 SKIPD=SKIPD+1 !111
250 CALL JOYST(1,X,Y)!129
260 IF X=-4 THEN 200 !154
270 REM TIS SITUATION CAN OC
CUR EITHER (1) LEGITIMATE OP
ENING, OR (2) BREAK IN. SO W
E GIVE A FEW SECONDS BEFORE
ALARM IS SOUNDED. !161
280 FOR ENTRDEL=1 TO ENDEL !
107
290 NEXT ENTRDEL !164
300 CALL JOYST(1,X,Y)!129
310 IF Y=-THEN 390 !093
311 PRINT #1 :: "ALARM TRIPP
ED BE ALERT BE ALERT BE ALER
T!" !136
320 CALL JOYST((Y/4+1),X,Y)!
201
329 REM ALARM BROKEN SIREN !
169
330 FOR LOOP=1 TO 5 !045
340 FOR SIREN=700 TO 900 STE
P 10 !031
350 CALL SOUND(-99,SIREN,0)!
062
360 NEXT SIREN !023
370 NEXT LOOP !208
380 GOTO 391 !215
390 PRINT #1:"WELCOME HOME"
!112
391 END !139
400 REM EXIT DELAY INITIATED
 FOLLOWED BY ARMING PROCEDUR
E !028
410 FOR DELAY=1 TO EXDEL !21
4
420 NEXT DELAY !005
430 RETURN !136
440 REM CHECK PERIMETER. IF
O.K. CHECK KEY. IF NOT, RE-C
HECK PERIMETER. !182
450 CALL JOYST(1,X,Y)!129
460 IF X=-4 THEN 190 !144
461 IF SW>1 THEN 481 !067
470 IF SW>1 THEN 481 !067
479 REM PERIMETER OPEN WARNI
NING SOUND !047
480 OPEN #1:"SPEECH",OUTPUT
```

(See Page 21)

BURGLAR ALARM—

(Continued from Page 20)

```
!122
481 PRINT #1:"//40 500" !142
482 PRINT #1:"DE PERIMTER IS
 OPEN SHUT THE WINDOW" !174
490 SW=SW+1 !205
510 GOTO 450 !018
599 REM KEY SWITCH OPEN WARN
ING SOUND !198
600 PRINT #1:"DE KEYSWITCH I
S OPEN THE SYSTEM IS NOT ARM
ED" !009
650 GOTO 200 !023
```

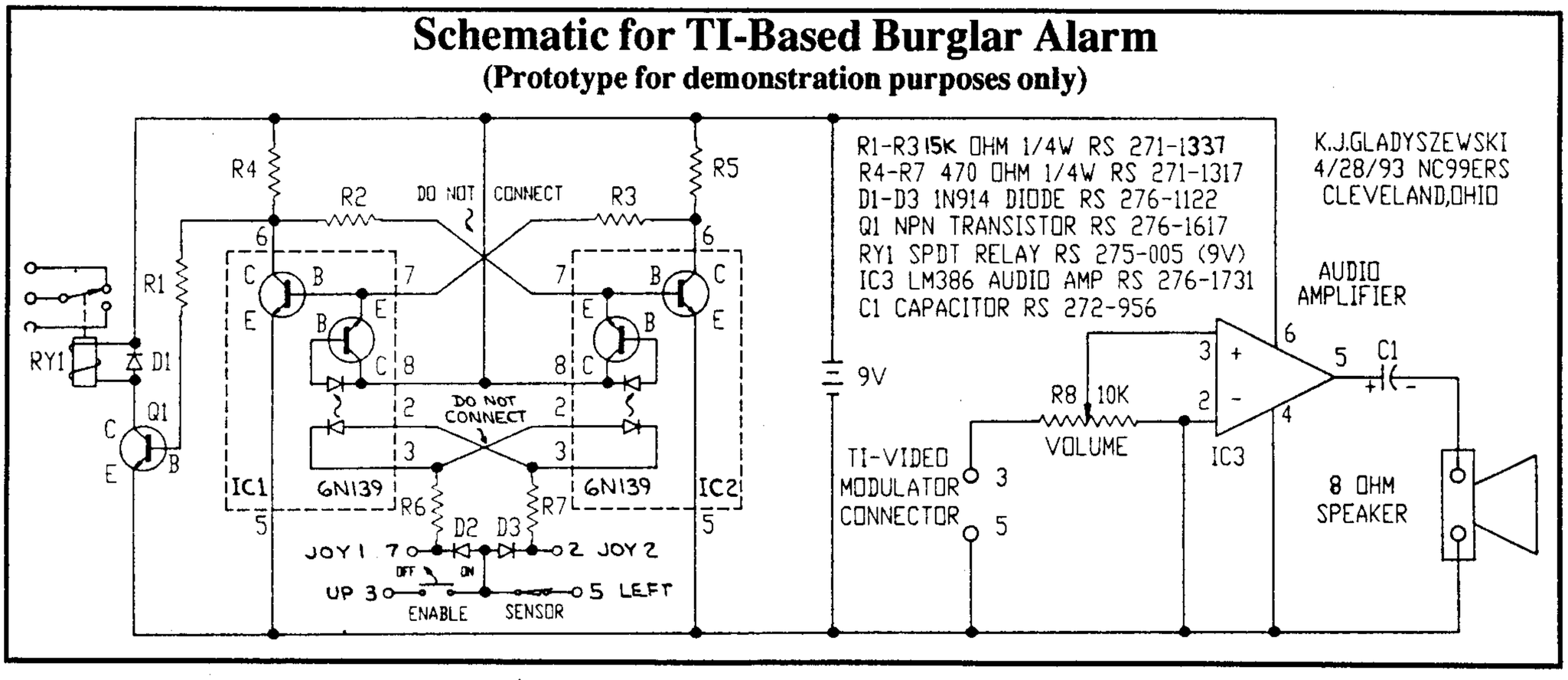
Flip-Flops solve hardware problems for burglar alarm

By KEN GLADYSZEWSKI

When Bernie and I first talked about his desire to control an automatic telephone dialer using a bare console, used as a burglar alarm, he suggested using an electronic circuit to energize a relay when it "hear" the console make a siren sound. I told him I had seen circuits for this somewhere, but thought I knew of a better and simpler way.

I had already used the pulses generated by the joystick port, when a CALL JOYST command in BASIC is executed, to control an analog-to-digital converter chip. And I had thought about using the same pulses to control a flip-flop.

A flip-flop is an elementary building block in digital electronics and is the basis of all memory and counting circuits. It has two stable states and in its simplest form consists of two transistors cross connected so only one transistor is "on" at a time. The circuit is said to "flip" when the on transistor turns "off," and vice-versa. The method normally used to cause the flipping action is to apply a pulse to one of the two transistors.



In review, the joystick port consists of two outputs (pins 2 and 7) which are normally both at +5 volts with respect to the common of the console power supply. When a CALL JOYST command is executed, one of the outputs goes to -5 volts for 150 microseconds. If a connected joystick is pushed in any direction, a switch in the joystick unit is closed and the output is connected to an input. The inputs are scanned and the program returns a value indicating joystick position.

In my early projects I used the 10 volt differential between the two outputs to fire a transistor, which then produced pulses compatible with the integrated circuits I was trying to control. A transistor was used because only 1ma of current is available, which is not enough to light a Light Emitting Diode(LED) in an Optocoupler — or so I thought.

OPTOCOUPLER FORMS FLIP-FLOP

Recently I discovered an Optocoupler (6N139) that requires only 1/2ma, and it just so happens that two of these devices may be interconnected to form a simple flip-flop! By executing a CALL JOYST(2,X,Y) command it will flip in one direction energizing a relay, and a CALL JOYST(1,X,Y) command will make it flip back, de-energizing the relay. At the same time, the status of both the key and perimeter switches are returned in the X and Y values. The circuit shown in the diagram was used to develop the concept and was used in Bernie's demonstration. To be used as an actual burglar alarm, because of the likely distance between the computer and the switches, Optocouplers would need to be used as buffers.

The circuit as shown is for only one zone, but it may easily be expanded to two zones by hooking each zone to a direction input and the keyswitch to the fire input and modifying the program appropriately.

I am working on an expanded version for three zones using two more integrated circuits (eight or more zones are possible using a different scheme) for use in my own home. Three zones would be nice because my house has three levels, with an entrance on each. The computer could detect and announce the level on which the problem occurred.

(See Page 22)

EASY DATA V1.0

Quick and easy databasing

By JOHN KOLOEN

Easy Data is a simple database program designed for use by unskilled Extended BASIC programmers who want to be able to create a custom database but don't want to spend a weekend doing it.

Written in Extended BASIC, with assembly language support, Easy Data provides the user with a skeleton database program that is easily customized by substituting and modifying arrays, labels and DATA statements. No programming is required.

Easy Data comes with two demos, one called BIGDEMO that includes a database with six fields, and another called SMALLDEMO that contains only two fields. Running these gives you a quick idea of what the program can do for you and how it works. When you're ready to create your own database you start by loading the program called SKELETON.

The first thing you do in SKELETON is to decide the variable names for your arrays and the number of arrays to use. Then you modify the menu to reflect the names and input your corresponding DATA into DATA statements. All it takes is modifying a half-dozen or fewer lines, plus whatever additions you want to make to the DATA statements. You can also create a data file by entering your DATA statements as a separate Extended BASIC program and saving it in MERGE format and then MERGEing it into your SKELETON program when needed.

Additionally, the program includes a couple of MERGE programs that can be used to extend the capability of Easy Data. One such program, called FILESERVE, adds "File Services" to the menu which gives you the option of saving or loading your data file to disk as a D/V 80 file. Databases can also be output to a printer.

Another MERGE program, called ADVANDIS, enhances the SKELETON display of sorted lists and provides the capability of paging backward through the data by pressing FCTN-E.

A third program, ADDATA, expands the original SKELETON list of DATA statements to include seven records.

The advantage to these MERGEable programs is that the user can see how easy it is to expand the power of the Easy Data program. You can examine these very short MERGE programs and make up your own.

Also included is a program called PRELOAD which frees up 2248 bytes of high memory if your database gets too large.

DOCUMENTATION: The docs are on disk as two D/V80 files. A driver program is included to output the files to a printer. There are a couple of middling contradictions in the docs, for example the ADVANDIS file is referred to as ADVIS in the docs. But the nine-page manual thoroughly covers the program. It certainly goes beyond what you'll need to know if your goal is to create a simple database of names, addresses and phone numbers.

EASE OF USE: Easy Data, as its name suggests, is easy to use. It took me five minutes to modify the SKELETON program to create a data file capable of holding 100 records of 3 fields each. If you want a larger number of fields, say six, start with the program called BIGSKEL instead of SKELETON. It contains a half-dozen fields.

FINAL GRADE: Easy Data is a simple database which, Bruce Harrison says he wrote because a customer asked for a "poor man's" database to keep track of a collection of video cassettes. And at its most basic, this is exactly what Easy Data is. But by using assembly language routines for sorting and menuing, the program is much faster than one would expect from Extended BASIC. Given its inherent expandability through the use of add-on MERGE programs, this is a program that can satisfy the needs of a host of users who do not need to maintain large databases. Its only limitation in this regard is that all data resides in memory and when you run out of memory, you'll know that you've exceeded the limits of Easy Data. Up until that point, it should do fine.

Note: Bruce Harrison reports that he has made a modification to Easy Data that allows users to input program lines between DATA statements without affecting the operation of the program. Version 1.0 allows only DATA statements after line 1000.

Review

REPORT CARD

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

Cost: $6.00

Publisher: Harrison Software,5705 40th Place, Hyattsville, MD 20781; 301-277-3467.

Requirements: Extended BASIC, expansion memory, disk system, Extended BASIC, printer optional.

BURGLAR ALARM—

(Continued from Page 21)

I plan to remove the motherboard from a spare console, along with the switching power supply, and house them in a wall-mounted metal box, along with some gel cell batteries for program backup. The peripheral circuit shown would be powered from the 12 volts available at the modulator port, thereby eliminating the need for a 9-volt battery.

As further reading, I suggest the following:

• *Home-Security Cookbook*, Parts 1 and 2, by Ray Marston

• *Radio-Electronics Magazine*, May and July 1988

• *Working With Flip-Flops*, by Ray Marston, *Radio-Electronics*, June 1987

• *Optocoupler Devices*, by Ray Marston, *Electronics Now Magazine*, Aug. 1992

MICRO-REVIEWS

Scout Museum Disk and O Say Can U See

By STAN KRAJEWSKI

The generosity of TIers never ceases to amaze me. The author of the first program in review has donated a TI system with P-box, 32K, disk drives and joysticks along with much time in programming to the Central States Scout Museum. Most items were purchases from the Chicago and Phoenix fairs. It looks like our fairs benefit not only our users, but also other organizations.

Ratings for the software reviewed in this column are based on the Star system that follows.

* Leave it alone, back to the drawing board.
** Needs improvements, but workable.
*** A good program, worth trying.
**** Send your money and buy it.

NOTE: If the Geneve 9640 is *not* specifically mentioned in system requirements of any column I write, the program is TI99/4A compatible only.

☆☆☆☆

SCOUT MUSEUM DISK

The TI world always has room for educational software programmers like Don Shorock, and now his 13-year-old son is programming in XB and "c." After all, the TI has always been a good "learning computer" and his other programs that were a real fun learning tool include AIRTAXI, and for me and my kids, SON OF AIRTAXI.

System requirements are Geneve 9640 (Triple Tech Card needed for use of speech) or TI99/4A, 32K RAM, disk drive, Extended BASIC and joysticks.

Items included on the Scout Museum Disk are Scouting Trivia, Square Knot, Finger Spelling, Morse Code, Semaphore Signaling, Kum Ba Yah, 22 Campfire Songs, Capital Ship, Dan Tucker and Star Spangled Banner.

Part of the fun in using the programs included on the Scout Museum disk is being able to sit back and use the joystick. Also incorporated within the program is a timer-counter, returning you to the main menu when the joystick is not used for a period of time. This will occur from any part of the program. Another extra feature is an arcade style screen which constantly changes, prompting you to play and use the program. This feature lets you keep the monitor on and the program running so screen burning does not occur. Speech is also included in both the programs and the waiting mode. A simple press of the joystick fire button will return you to the menu to continue use of the program.

Scouting Trivia is just that. It starts by asking if you live in Kanza Council (located in Larsing, Kansas) or if you are not a local scout. The questions then pertain to your answer. Questions are answered by just using your joystick and moving the BSA cursor to the proper multiple choice box. If your answer is wrong it will display the correct answer. A score is kept with right and wrong answers and you may stop the trivia game any time by just moving the cursor to the box containing "Quit trivia quiz."

Square Knot — draws a square knot while the computer describes in speech what it is doing to create the knot.

Finger Spelling — shows a hand in different finger positions. The alphabet is displayed on the bottom of the screen with an arrow above, allowing you to choose which letter of the alphabet the hand is displaying. If you were incorrect it displays the correct finger symbol. A "Don't Know" option is also included which gives the correct finger symbol.

Morse Code — displays dots and dashes along with a dot and dash tone. Again you just move the arrow to the corresponding letter that you think is displayed. A score is kept and also the "Don't Know" option. The right code is displayed if you were wrong.

Semaphore Signaling — displays a man holding the signal flags. This flag signaling system is used in our Navy to signal from ship to ship. My son loves this one as he is in Junior ROTC in high school. Again the alphabet and arrow are displayed for you to choose the proper letter. The "Tell Me" option is also included if you do not know the answer.

Kum Ba Yah — runs the music for this song, the words of which are also displayed. By moving the cursor with the joystick, you can choose the speed, frequency and how many verses you want to hear.

22 Campfire Songs — lets you choose which one you want to hear, then plays it. These are shortened versions but familiarize you with the songs. You can choose one song or all the songs in a row. Capital Ship and Dan Tucker are other songs with the words displayed. The Star Spangled Banner displays a large flag while playing the anthem. These are true long versions.

This disk is a excellent learning tool for both young and old. The added features I described show the extra effort put into the program earning it the four stars. Half the programs are the work of Don and Tom Shorock. The other half comes from Tigercub Public Domain disks modified for use on this disk. The proceeds from this disk go directly to the Central Scout Museum. A minimum donation of $8 is requested, and I might say is well worth that plus more. Thanks, Don and Tom, for your time in supporting a worthwhile organization.

Scout Museum Disk is available in SS/SD format from Don Shorock, P.O. Box 501 Great Bend, KS 67530-0501. Please enclose an additional donation of at least $2 for cost of disk copying and postage and handling. Make checks payable to Central States Scout Museum. A four-page educational program catalog is also available and can be obtained by request.

☆☆☆

O SAY CAN YOU C

This demo program compares the speed of TI Extended BASIC to Clint Pulley's c99 language. System requirements are Geneve 9640 or TI 99/4A, 32K, disk drive and Extended BASIC.

As you auto-load this diskette, a menu will appear. The four options are: A — Prime Factors/XB, B — Prime Factors/C, C — Ten Spot Keno/XB and D — Ten Spot Keno/C. Prime Factors XB will prompt you to enter a number so the computer can

(See Page 24)

Mini-tower Panda box is loaded with expansion possibilities

Review

By BRYANT C. PEDIGO

This review appeared in the newsletter of the Hoosier User Group.

Last May (1992) at the Lima, Ohio, Multi-User Group Conference, I purchased a Panda Expansion Box, which is a modified IBM mini-tower case marketed by Bill Nelson of Panda Computer Products of Garden Grove, California.

The Panda box has several advantages over the TI Peripheral Expansion Box, the foremost of which is a 200-watt power supply capable of running four floppy drives and a hard drive, all installed internally. There are spaces for 2 half-height 5.25-inch and 2 half-height 3.5-inch floppy drives, as well as a 3.5-inch hard drive. Additionally, the Panda has a reset switch, a power connector at the back for use with a monitor, a very quiet cooling fan and a compact contemporary design. For Geneve users, there is a keyboard port on the front, and a card protector with a support post to help support the protruding part of the Geneve card.

Although it is designed for use with a Geneve 9640 in mind, it can easily be used with the TI99/4A. Although I purchased it to use with my Geneve, I tried it out with a 99/4A. Unlike the Rave Expansion Box, you have to use the flex cable interface card if you want to use a 99/4A console.

The inside of the Panda has been modified to accept TI style cards inserted horizontally. Because the power supply has a regulated +/- 12v output, any cards that use either a 7812 or a 7912 voltage regulator require modification in the form of a jumper to short out the +/- 12 volt regulators. The directions include complete information as to which cards require this modification and how to jumper both types of regulator.

The directions are, for the most part, complete and easy to follow. One item that was not mentioned, and which proved to be a major problem for me, was that the power supply used must have a minimum power draw in order to work. Because of a problem that developed with my hard drive, I tried to set up the Panda with a Geneve, 1 5.25-inch and 2 3.5-inch floppy drives. It would start to power-up and then turn off. With a 99/4A, there was no problem. Upon contacting Bill Nelson, I shipped the box and my Geneve to him. He checked the system out and found that it was not drawing enough power without a hard drive installed. After the box was returned to me, I installed a second 5.25 floppy, thus solving the power draw problem.

I have been using the system for several months without any problem. I recently tested with both internal and external hard drives and had no problems. I have noticed that cards seem to run as hot in the Panda box and in the TI PEB, therefore it would probably be a good idea to block off those parts of the ventilation grill at the rear that do not have cards installed so that air flow inside will be increased.

I have no qualms about recommending the Panda box, especially in view of the excellent support from the vendor.

The Panda Expansion Box is priced at $200. Nelson builds the boxes on request and requires that buyers ship a TI PEB to him to expedite the process. The TI PEB bus and rack is used in the Panda box.—Ed.

MICRO-REVIEWS—

(Continued from Page 23)

factor it for you. This can be a number anywhere from 1 to 32,767. I ran number 5873; this took 10 seconds to factor. When I ran it on Prime Factors in C, it took less than 1 second. In Ten Spot Keno, it asks you for the number of games; I used 1. For demo purposes I used the quick pick instead of the self pick; it took 18 seconds in Extended BASIC versus 1 second in "C." This program proves the speed of "C" over Extended BASIC.

Although this is a demo program, you an also pass some time with it entering umbers to see how many the computer can atch. But the most valuable part of this sk is its tutorial aspect. This is where I rated it as a three-star program. The docs that come with "C" are for the users who already have some familiarity with it. The two D/V files included on this disk give a good insight to someone approaching "C" for the first time. If you are a Extended BASIC programmer, this disk gives you working examples including the c99 source code. As explained in the docs, the advantages of programming in "C" don't optimize program code, but optimize programming time by churning out generic assembly code.

The docs included on this disk include: Getting Started in c99 — Building A Working Copy Of c99, The Language Itself and Getting the Program to Run, Comparing The Programs — Prime Factor Program, Club Keno: Ten Spot.

The most important aspect of this program is the "sharing" of other TIers' knowledge and routines, which with some programmers seems to be a problem. Bruce Harrison has been generous enough to allocate his loader as public domain, which puts Editor/Assembler subroutines into low memory. This enables the Extended BASIC cartridge to be used with c99. It seems that c99 programs use subroutines in the GROM on the Editor/Assembler Cartridge. Tom has incorporated this loader into his c99 programs letting Extended BASIC users take advantage of this fast language.

Let's encourage this talented 13-year-old to continue programming for us by sending your $5 to $10 Fairware donation. O Say Can You C is available on SS/SD disk from Tom Shorock, P.O. Box 501, Great Bend, Kansas 67530. The proceeds from this disk will help him buy much needed TI equipment.

If you would like your software or hardware reviewed in this column, you may send it to: Stan Krajewski, Route 6, Box 568-15, Live Oak, FL 32060. If you would like it returned, please include postage. If you need to call me for any reason, you may reach me at (904)-364-7897 E.S.T.

RXB is turbo-charged version of Extended BASIC

By RICH GILBERTSON

The following article describes RXB, a powerful version of Extended BASIC. It is written by the author of RXB. RXB is available through C.A.D.D. Electronics, 81 Prescott Rd., Raymond, NH 03077; 603-895-0119. The cost is $24.95. We're printing the article because we feel it is an exemplary product that has received little attention It currently works only on a GRAM device or with a Geneve. However, the author is working on developing a module version.—Ed.

Extended BASIC has BUGS?

Yes, bugs! But they don't just glare out at you. If you are a programmer then you will notice them after you've written enough programs.

What are the bugs in XB? Well, CALL INIT moves way too much into the lower 8K for Assembly support. As a matter of fact, about 50 bytes too much. It will overwrite any code there, and that is a problem.

The CALL LOAD command won't work unless you use CALL INIT first, and that would be fine except that includes your only loading address.

When you use the LIST command you can list sections of lines but not the line length to the printer or device.

The PERMANENT command is worthless for disk, and a CALL LOAD can do the exact same thing.

The RESequence command does not allow portions of programs to be resequenced, only the entire program, and this precludes having a hidden loader at the same time. It forces you to MERGE more often then makes sense.

You should also have a COPY command to copy portions of a program, or a MOVE command to move around portions of programs, and the DELete command should allow portions of a program to be deleted.

The TRACE command does not allow output to a printer or disk file. TRACE should have a permanent file access that will output when you access it. Presently it is complicated to list out and TRACE at the same time.

The QUIT key should have commands to turn it off and on., as even I have erased hours of work by hitting it when I should not have.

After selecting Extended BASIC after the title screen you should have a key to bypass the DSK1.LOAD, as waiting all the time to get in command mode is a waste of time.

While editing a program all of us have wished for a full-screen editor instead of just the left and right direction keys. You should be able to move the cursor up and down within a line. And this should also work in INPUT or ACCEPT AT commands, even while running a program.

The TI has three types of memory and they are; VDP, RAM/ROM, and GRAM/GROM. Extended BASIC has CALL LOAD and CALL PEEK for RAM/ROM. But could have CALL POKEV and CALL PEEKV for VDP, and CALL POKEG and CALL PEEKG for GRAM/GROM.

The NEW and BYE commands are not allowed in a program. They should be allowed in a program by a CALL.

XB has always needed a CATALOG routine built into it, as you are always forced to load one or exit to get one done.

Using CLOSE for each file opened is very wasteful and, if your XB program is to done, even more trouble. There should be a command that closes all open files at once.

IT ALL STARTED WITH GKXB

This is not all of the bugs that people have complained about in XB, but a good sized list that has actually been fixed. You see in 1985 a company called Miller Graphics came out with a GRAM device that replaced the cartridges called a GRAM Kracker. In 1986 the company released a product called GKXB or GRAM Kracker Extended BASIC, which was mostly written by Danny Michael. Every one of the above listed bugs was fixed and is actually featured in GKXB.

Later came many other Extended BASICS and everyone of them used the GKXB as a base to build upon. You might have one of them now: Super Extended BASIC, XBII or XB 2 (both German versions of XB), and soon to be released XB3. Not to mention RichGKXB or RXB for short. *(RICHGKXB and RXB are used interchangeably in this article.—Ed.)*

When I started RXB it was to fix other things in XB that have I considered bugs. You know, those times when you said "I wish...." So I got busy converting GKXB into what is now RXB.

TI BASIC will run in RXB with no modifications. RXB allows characters 30-159 to be redefined the same as in TI BASIC. Of course, if characters 144-159 are being used this nulls out the use of sprites at the same time. Characters 30 and 31 will remain redefined when back to command mode. And can be redefined or restored by NEW.

Editor/Assembler has been added to RXB so it is all in one module and Rich Editor/Assembler (REA) has been also updated.

RXB and REA both have true lowercase characters and both use the same definitions. That means whether in graphics or text mode the definitions remain constant.

Auto-repeat commands such as SPRITE allow several sprites to be created in one command. The comma after each set of values allows another sprite to be created. But there are many commands in XB that don't auto-repeat,such as KEY. This has been fixed. Here is an example:

Extended BASIC

```
1 CALL KEY(1,K1,S1) :: CALL KEY(2,K2,S2)
```

RICHGKXB

```
2 CALL KEY(1,K1,S1,2,K2,S2)
```

Compare the two lines above and you will notice just what has been fixed. Here is a list of new autorepeating commands:

CALL KEY
CALL JOYST
CALL COINC
CALL DISTANCE
CALL MAGNIFY

The SIZE command has been fixed so that, from program mode, CALL SIZE will report the amount of stack and memory that is free. Both report now if CALL INIT or CALL LOAD (assembly program) has

(See Page 26)

RXB—

(Continued from Page 25)

been called, or a hidden loader installed in an assembly program, then SIZE will also report assembly language bytes free.

COMBINED COMMANDS

RXB has a new syntax called Combined Commands, and uses the first two characters of the normal Extended BASIC command from two different commands. That means four characters replace two names and only one CALL is needed. An example of this is CALL JOKE, which breaks down to one CALL, JO being short for JOYST and KE being short for KEY.

To understand the new syntax compare the next two lines.

Extended BASIC

```
1 CALL JOYST(1,X,Y) :: CALL KEY(1,K,S)
```

RICHGKXB

```
2 CALL JOKE(1,X,Y)(1,K,S)
```

Currently, 157 commands are available in RXB V. 5.40, and Combined Commands such as COINC and any second command have a new special extra feature: the last variable in the COINC section of the combined command determines if the second half of the command is to be executed or ignored. Here is an example:

Extended BASIC

```
1 CALL COINC(ALL,V) :: IF V THEN CALL DELSPRITE(ALL)
```

RICHGKXB

```
2 CALL CDE(ALL,V)(ALL)
```

In both lines if V=0 then CALL DELSPRITE(ALL) is ignored.

In both lines if V=-1 then CALL DELSPRITE(ALL) is executed.

In COINC the ALL option must always be first if used. Here is an example:

Illegal and incorrect resulting in *SYNTAX ERROR*

```
1 CALL COINC(#1,#3,9,V,ALL,Z)
```

Correct method

```
2 CALL COINC(ALL,Z,#1,#3,9,V)
```

Compare these to normal XB for reference.

MORE NEW COMMANDS

CALL VERSION will report RICHGKXB version numbers. Maximum version number will never be reached as it is above 32000.

CALL BASIC will go to TI BASIC.

CALL XBASIC is the same as RUN "DSK1.LOAD" and can be called from XB.

CALL EAMENU goes directly to the Editor/Assembler menu. And can be called from XB.

CALL EAASS goes directly to Editor/Assembler load Assembler request. And can be called from XB.

CALL EAED goes directly to Editor/Assembler Editor menu. And can be called from XB.

CALL EAPGM takes the string from XB and goes directly to

Editor/Assembler `RUN PROGRAM FILE' menu. It then installs the name on the screen and executes the path name and file name or disk designation number and file name.

CALL EALR takes the string from XB and goes directly to Editor/Assembler 'LOAD AND RUN' menu. It then installs the name on the screen and executes the path name and file name or disk designation number and file name.

CALL DUPCOLOR takes the color of a set and duplicates it into the other set specified.

CALL DUPCHAR takes the character definition of a character and duplicates it into the other character specified.

CALL SWAPCOLOR takes the color of a set and swaps it with the other set specified.

CALL SWAPCHAR takes the character definition of character and swaps it with the other character specified.

CALL INVERSE takes a character specified and inverts all the bytes in the definition. That means it reverses the foreground and background.

CALL HFILL and CALL VFILL does the same thing as CALL HCHAR and CALL VCHAR when the entire screen is to be filled with one character.

CALL DELAY is a smaller replacement of a FOR/NEXT loop which slows a section of a program. The minimum delay is 1/10th of a second. The maximum is over 12 days.

CALL WINDOW takes a string and places it on the screen, but it wraps and breaks up the string according to the height and the width values. So it works with a screen block or strip of characters and wraps the block or strip on screen much like a sprite.

CALL GWINDOW is the opposite of WINDOW, it retrieves into a string variable what is on screen. But like WINDOW breaks up what is in the string variable according to the height and the width values. So it works by making a screen block or strip into a string of characters.

CALL HPUT is a simpler version of XB's DISPLAY AT, but uses columns 1-32 instead of columns 3-28 like XB. Also the command auto-repeats, taking less syntax to do the same thing.

CALL VPUT is a vertical version of CALL HPUT. See CALL HPUT.

CALL HGET is the opposite of HPUT. It gets a string from the screen and puts it into a string variable. Control specified in the length value determines the number of characters read from the screen, and puts it into a string variable.

CALL VGET is a vertical version of CALL HGET. See CALL HGET.

CALL RDIAG is like HCHAR and VCHAR stepping down and to the right. It thereby creates a right diagonal downward. It has a strange wrap around the screen because of this.

CALL LDIAG is a left version of RDIAG. See RDIAG.

(See Page 26)

User Notes

Installing Birdwell utilities on PGRAM+

This comes from Peter deWitte, of Winnipeg, Manitoba. He writes:

The following will allow you to install the late John Birdwell's Disk Utilities files on the PGRAM+, and perhaps also on GRAM Kracker or other GRAM device.

The first step is to copy the files to another disk and rename them at the same time, as in the following examples:

Original File	New File
DSKU1	DSKU

(See Page 27)

User Notes

(Continued from Page 26)

DSKU2	DSKU1
DSKU3	DSKU2
DU	DU
DV	DU1
DW	DU2

The files must be renamed in accordance with the preceding examples to make them compatible with the PGRAM+ loader. This may or may not be required with other GRAM devices.

The second step is to change the first six bytes of each file using the Disk Utilities file editor option. In my copy of V4.12, the first six bytes appear as follows and must be changed as indicated.

File	Original	Change To
DSKU	FFFF 2000 A000	FF04 1FFA 604C
DSKU1	FFFF 2000 BFFA	FF05 1FFA 8046
DSKU2	0000 1BF2 DFF4	0006 1BF2 A040

and in my copy of V4.2, as follows:

File	Original	Change To
DU	FFFF 1FFA A000	FF04 1FFA 604C
DU1	FFFF 1FFA BFFA	FF05 1FFA 8046
DU2	0000 1C0A DFF4	0006 1C0A A040

In my copy of V4.12 the third and fourth bytes contain 2000, which should indicate the number of bytes to be copied into memory In this case it happens to be 1FFA bytes rather than 2000 for the first two files. The PGRAM loader seems to ignore this inconsistency, but I don't know if this is of any consequence to other loaders. The only demand for accuracy here is to determine the load addresses for the altered files.

To demonstrate:

A000+1FFA=BFFA 604C+1FFA=8046

BFFA+1FFA=DFF4 8046+1FFA=A040

Address Data

```
g6000   AA 01 00 00 00 00 60 10 00 00 60 10 00 00 00 00
g6010   00 00 60 1B 03 44 49 52 70 01 1C 39 00 01 01 60
g6020   18 39 00 01 04 60 19 39 00 01 07 60 1A 07 20 35
g6030   00 C0 A3 00 A0 00 BF 4A 09 00 06 00 18 31 5B FE
g6040   8F 1D 00 60 4C BF 00 A0 00 0F F0 0B
```

Fig. 1

The third step is to load these files into the PGRAM page 1, enable the GRAM bank and enter the GPL program data listed in Fig. 1.

The next step is to use the Save P-GRAM function to save these new files. They must be saved as long files rather than as short files. You may now load these files into some other PGRAM page.

You may call up the program from the Review Module Library menu or from the BASIC or Extended BASIC command

(See Page 28)

RXB—

(Continued from Page 26)

CALL GMOTION is a new sprite command. It is the opposite of the CALL MOTION command in XB. GMOTION retrieves the row and column velocities of a sprite and puts them into variables. Many unusual sprite effects can be created with this command.

CALL RMOTION is a new sprite command. It automatically reverses the direction of a sprite. New sprite effects can be created.

CALL XBPGM is a new command that will run an XB program from a string or string variable. Also engages a menu if wanted.

CALL MOVE is a new command that will move any amount or kind of memory to any other kind of memory. Much like a CALL PEEK and a CALL LOAD in one command, but can work on screen also.

CALL MOVES is a new command that will move up to 255 bytes into strings or from strings. It is very fast compared to LOAD or PEEK.

CALL IO is a new command that allows XB to play E/A sound lists from GROM/GRAM or VDP. IO also has direct CRU control for by-passing a DSR of a device and controlling it. IO can also do a direct output or input or verify through cassette. (Bypass of all menus.)

CALL PROTECT is a disk command that protect/unprotects up to 15 different filenames in 15 different drives in one command.

CALL RENAME is a disk command that renames up to 14 different filenames in 14 different drives in one command.

CALL EXECUTE is a new command that will execute a assembly address specified. It is about twice as fast as CALL LINK.

A MENU in RXB now appears in version 5.40 and more features are being added to this feature now, and new Editor/Assembler menu also.

This brings the editor in Extended BASIC to mind. The editor within Extended BASIC only allows a maximum of seven lines before it considers it's memory is full. I saw this as a problem as with a little checking I noticed the crunch buffer in some cases was one-third full. So in RICHGKXB the edit buffer is now over 24 lines tall. That means unless the crunch buffer is full you can keep on adding lines to a single line program. But there is a cost! If you try to edit a 24-line program it will overwrite the color table and the screen will go blank. If this happens, you can get back by not doing anything except hitting Enter several times, then typing in blindly CALL CLEAR. The maximum I've ever gotten to with a one line program was 28 lines. You will discover 90 percent of the time the crunch buffer is full first. The trick to going past 24 lines is to delete all the blank spaces you can before adding anything.

With all these things added to XB I still find from day-to-day that I need to fix something else. So RXB is the only XB that is updated and upgraded several times a year. Yet it always remains 100 percent compatible with previous versions of RXB and TI Extended BASIC and console BASIC. That means that if you ever have any problem with anything, just call me and I'll fix it so that it is compatible with whatever you are problem with. However, I don't think that will happen as we have had only two incompatibility problems turn up in two years of use.

RXB is my only project, and will remain so. I will also produce an 80-column version of RXB when enough 80-column cards or devices are out there.

User Notes

(Continued from Page 27)

line, with the following command: CALL DIR.

Although there is not much sense in it, you may also call up the program from within a BASIC program with the CALL DIR statement. Extended BASIC does not allow this method.

Another challenger comes forward

This comes from Bruce Harrison, MICROpendium's assembly language columnist. He writes:

In the March 1993 User Notes as an answer to last November's challenge from Stephen Shaw. Having once been an Extended BASIC programmer, I've taken up the challenge too.

My solution is the following, which can be saved as a MERGE file and MERGEd with Stephen's original program. This would replace his line 290 and then add lines 300-330 to make an "add fractions with reduction" combination.

My own little "reduction algorithm" is simpler than the one offered by Mr. Mah in the March 1993 edition. Rather than introduce subprograms, it simply adds to the subroutine that was in Stephen's original starting at line 230.

```
290 FOR Y=2 TO MIN(N,L)
300 IF N/Y>INT(N/Y)THEN 330
310 IF L/Y>INT(L/Y)THEN 330
320 N=N/Y :: L=L/Y :: GOTO 290
330 NEXT Y :: RETURN
```

MY-SLEEVE for TI

This comes from Jim Uzzell, of Key West, Florida. He writes:

In response to Robert E. Knight's request for a TI version of MY-SLEEVE (Feedback, August, 1992), do the following:

Delete line 100

Delete line 160

Delete M$=DATE$ in line 170

Add the following line:

```
175 DISPLAY AT(22,1):"DATE" :
: ACCEPT AT(22,6):M$
```

The format of M$ is 00-00-00

Change CLOSE ALL in line 700 and 760 to CLOSE #1.

This program as written will not recognize subdirectories on the TI but, based on limited testing, the program code does not appear to cause a problem

MY-SLEEVE, by Jim Uzzell appeared in the February, 1991 MICROpendium. It was written for the Geneve.—Ed.

Shopping list

This item was written by Jim Ballenger. We picked it up from TI*MES, the newsletter of the TI99/4A User Group of the United Kingdom.

This program is based on one included in Jim Peterson's Tips from the Tigercub column that Jim circulated to club newsletters, and I used it slightly messed about because I'm queer like that. If you read this, you will know that Jim has okayed the idea.

It is a bit cheeky to alter programs by such a respected and energetic programmer, but he has a sense of humor too, it seems.

The program should be run with the printer fired up, and will print headings automatically. It is so designed that you may alter the content of the data lines so that they will print out in the order that you find them in your grocery store.

The goods listed may be altered simply by amending the data list. This list is, or rather was, based on my local store, but a new management has altered the layout.

```
100 CALL CLEAR :: CALL SCREE
N(12)!024
110 FOR C=1 TO 9 :: CALL COL
OR(C,7,16):: NEXT C !071
120 OPEN #2:"PIO" !254
130 PRINT #2:CHR$(27);"W";CH
R$(1);"SHOPPING LIST" :: PRI
NT #2:"**************": : :!1
75
140 DISPLAY AT(6,1):"      S
HOPPING LIST" !190
150 DISPLAY AT(7,1):"      *
************" !029
160 DISPLAY AT(16,1):"  GROU
PS OF GOODS WILL BE     SHOW
N. ENTER A QUANTITY     FOR
EACH ITEM REQUIRED." !203
170 DISPLAY AT(20,1):"  IF N
OT NEEDED PRESS THE
   ENTER KEY              PRES
S ANY KEY TO START" !141
180 CALL KEY(0,K,S):: IF S=0
 THEN 180 !248
190 RESTORE :: FOR F=1 TO 57
 :: READ A$ !005
200 DISPLAY AT(12,1):A$ !243
210 DISPLAY AT(12,LEN(A$)+2)
:"0" !239
220 ACCEPT AT(12,LEN(A$)+2)S
IZE(-4):Q !097
230 IF Q=0 THEN GOTO 250 !13
3
240 PRINT #2:"[ ] ";A$;RPT$(
" ",(17-LEN(A$)));Q !207
250 NEXT F !220
260 FOR DEL=1 TO 250 :: NEXT
 DEL !025
270 DISPLAY AT(12,1):"ADDITI
ONAL ITEMS? Y/N" !026
280 ACCEPT AT(14,13)VALIDATE
("YN")SIZE(-1):Q$ !042
290 IF Q$="N" THEN 370 !185
300 DISPLAY AT(12,1):"ITEM?"
 !200
310 ACCEPT AT(12,7):A$ !251
320 DISPLAY AT(14,1):"QUANTI
TY?" !030
330 ACCEPT AT(14,11):Q !021
340 PRINT #2:"[ ] ";A$;RPT$(
" ",(17-LEN(A$)));Q !207
350 DISPLAY AT(13,1:"
                        "
 !002
360 GOTO 270 !094
370 PRINT #2:CHR$(27);"W";CH
R$(0)!130
380 CLOSE #2 !152
390 CALL CLEAR :: END !222
400 DATA BUTTER,MARGARINE,CH
EESE,EGGS,CREAM,COOKING FATS
,COOKING OILS,TEA,COFFEE,COC
OA !057
410 DATA SUGAR,MILK,TINNED F
OODS,FROZEN FOODS,SOUPS,BREA
KFAST CEREALS,BISCUITS,JAM-M
ARMALADE,FLOUR,CAKE MIXES !0
01
420 DATA CUSTARD,JELLIES ETC
.,RICE/SAGO ETC.,DRIED FRUIT
,SPAGHETTI,PEAS/BEANS,PICKLE
S/SAUCES,SALAD CREAM,SALT/PE
PPER,VINEGAR !168
430 DATA SPICES/MUSTARD,FL
ORINGS,MEAT,POULTRY,SAUSAGES
,BACON,HAM/PIES ETC.,SUET,FI
SH,FRUIT !131
```

(See Page 29)

User Notes

(Continued from Page 28)

```
440 DATA VEGETABLES,SALADS,B
READ,CAKES,PET FOODS,ICE CRE
AM,SWEETS,WINES,SOFT DRINKS
ETC. !104
450 DATA SOAPS,DETERGENTS,PO
LISHES,CLEANERS,SODA/STARCH,
MATCHES,TOILETRIES,STATIONER
Y !098
```

Hints, tips & answers

The following tips were compiled by Bill Sponchia of the Ottawa TI User Group. Individual credits aren't provided because the information is taken from many sources. These items pertain to BASIC or Extended BASIC.

To get true random numbers, install this line into your program:

```
CALL PEEK(-31880,A,B) :: CALL
INIT :: CALL LOAD(-31808,AB)
```

To erase the assembly program from memory without erasing the screen or disturbing any assembly routines in lower memory:

```
CALL INIT :: CALL LOAD(-
31952,255,231,255,231)
```

The manual tells you that there are 16 different character sets that you can redefine and change colors. Actually, there are 17 sets — set No. 0 is never mentioned.

When LISTing a program and you see a line reference to 32767 you should know that it occurs because you resequenced the program while you had a GOTO or GOSUB to a non-existing line number.

To LIST a portion of a program to a printer, enter the following: LIST "filename":line number range. In other words, enter LIST "PIO":130-240 to list lines 130-240 to your printer.

Gemini users get extra 96 characters

The following item was written by Jim Folz of the Charlotte User Group. It appeared in several user group newsletters.

Many Gemini printer users don't realize that they can access another 96 characters from their printers. The TI99/4A uses seven data bits to send characters to the printer. However, the additional characters supported by Gemini and other printers become available when the eighth bit is turned on. Gemini engineers have produced escape sequence to the the eighth bit on and off.

Using .TL60:27,62 in TI-Writer and printing through the formatter will turn the eighth bit on when a < is found in the text. Using .TL62:27,61 will turn off the eighth bit when the > is used. (Check your printer codes to make sure these codes are the same.

Many users will find it easier to use the CTRL-U sequence to access these codes when working from the editor — this avoids having to use the formatter. In this case, the sequence would be: CTRL-U, FCTN-R, CTRL-U, > to turn the eighth bit on; and CTRL-U, FCTN-R, CTRL-U, = to turn it off.

When the eighth bit is turned on, the characters change as shown in the character table.

The eighth bit simply adds 128 to the ASCII value of the character you type. Consult your printer manual for the characters you want to print, subtract 128, and use the ASCII character of the resulting number.

For a complete printout of the ASCII characters and the resulting number, run the following program:

```
100 REM PRINT CHARS 128+ !2
COL
105 OPEN #1:"PIO"
110 PRINT #1:CHR$(15) ! CONDE
NSE
115 FOR X=32 TO 80 ! 2 COL
120 PRINT #1:" ";X;" ";X+128
;" ";CHR$(X+128);"
";X+48;" ";X+48+128;" ";CHR$
(X+48+128)
130 NEXT X
140 CLOSE #1 :: END
```

Analytic computer art

The following was written by Stephen

(See Page 30)

BUGS AND BYTES

Media Ware responds

Mark Wacholtz of Media Ware Software says he has had problems with receipt of mail. (He says he did not, for instance, receive the letter written by MICROpendium inquiring about customers' non-receipt of their software.) Now, he says, he has talked to his postmaster and is having no further problems, and has shipped software to everyone whom he has received mail from. He says that anyone who still is due software from him should write him at 2141 NW 64th Ave. #15, Sunrise, FL 33313-3950.

Don't underestimate TI

According to Don Walden, the ANSI graphics on Tim Tesch's S&T BBS software for the TI99/4A are so good they have caused some individuals to doubt that the board really runs on a TI.

Hewlett-Packard's TI 'service'

The following item is from *A Byte of Info*, the newsletter of the Reading-Berks 99ers in Pennsylvania:

According to our member, John White, Hewlett-Packard is now doing all servicing on Texas Instruments manufactured equipment. John received word of this arrangement by calling the 1-800-TI-CARES hotline about something that he needed to have repaired.

When John called the number supplied by TI, he was greeted by someone who basically told him to junk the TI and replace it with something up-to-date.

If this is the case, then, I guess, the time will be soon upon us that we will be left to our own devices to keep our TI99/4As and their peripherals up and running.

User Notes

(Continued from Page 29)

Shaw of theTI99/4A User Group of the United Kingdom.

Yet another odd magazine to come my way is The Journal of Chaos and Graphics, edited by Clifford A. Pickover. A sample copy is available from: Journal of Chaos and Graphics, IBM T.J. Watson Research Center, Yorktown Heights, NY 10598.

The program below is from the August 1988 issue and produces a different sort of pattern. It is an opportunity for me to show you how to use the MOD function on the TI, instead of A=MOD(B,C) on some other computer. I have used a subprogram and the form CALL MOD(B,C,A).

Also, I have used the DEF function to prepare COS and SIN functions COSD and SIND which expect to find degree measures instead of the usual radians. It is a good way of finding out how efficient DEF is too — just try replacing the SIND and COSD functions and replacing them with the ordinary COS and SIN functions but amend the variable in accordance with the DEF — instead of COS(A) use COS(A/180*PI) in every SIND and COSD line. Any faster?

```
1 REM ANALYTIC COMPUTER ART
!165
2 ! JOE JACOBSON !240
3 ! WARMINSTER PA !096
4 ! fluted scallops !168
5 ! The Journal of Chaos
 Graphics August 1988 !038
6 ! For TI99/4A plus XBASIC
+The Missing Link by
 Stephen Shaw UK !212
8 !!131
100 DEF SIND(X)=SIN(X/180*PI
)!153
110 DEF COSD(X)=COS(X/180*PI
)!143
120 OS=99 :: RANDOMIZE !179
130 L=INT(RND*60)+3 !205
140 CALL LINK("PRINT",180,22
0,STR$(L))!205
150 FOR B=16 TO 76 STEP 2 !0
82
160 FOR A=0 TO 360 STEP 5 !0
73
170 GOSUB 250 !074
180 IF A>0 THEN 210 !201
190 CALL LINK("PIXEL",OS+X,Y
+OS):: OLDX=X :: OLDY=Y !207
200 GOTO 220 !043
210 CALL LINK("LINE",OS+OLDX
,OLDY+OS,OS+X,OS+Y):: OLDX=X
 :: OLDY=Y !015
220 NEXT A !215
230 NEXT B !216
240 GOTO 290 !114
250 R=B*(1+.25*ABS(SIND(L*A)
))! this is function plotted
:R=f(x):=polar coordinate cu
rve !150
260 X=R*COSD(A)!002
270 Y=R*SIND(A)!008
280 RETURN !136
290 CALL LINK("PRINT",180,20
,"+")!249
300 CALL LINK("DUMP")!005
305 CALL LINK("CLEAR")!055
310 RUN !169
```

Keep those print heads clean

This item was written by Chuck Reinhard of the Long Island TI99ers User Group. We picked it up from the group's newsletter.

It takes only three things to get good dark print from your printer:

1. A properly adjusted printer

(See Page 31)

User Notes

(Continued from Page 30)

2. A good ribbon.
3. A clean print head.

The guide for the fine printer wires gradually gets clogged with a mixture of lint, ink, and oils from the ribbon. As this dirt builds up and dries out, the pin wires drag in the guide. The result is you get light, low-contrast print even from a new ribbon. The following is a procedure for cleaning the print head that is quick, simple, and does not require removal of the head.

Obtain an aerosol can of Color TV Cleaner (Radio Shack 164-2320) or equivalent. Make sure the label states that it contains silicone, that it will not harm plastic and that it has a plastic tube to plug into the spray nozzle.

Power-off the printer. Leave paper in the printer, but remove the ribbon. Gently move the print head to the middle of the carriage.

Cut a two-inch square from a lint-free cotton handkerchief. Fold the cut cloth over on top of itself a couple times until it about the width of your printer ribbon and about four layers thick.

Insert the cloth into the print head exactly where the ribbon was, between the pin guide and the ribbon shield. The cloth should not fit too tight.

Insert the tube into the aerosol spray cap. Put the end of the tube in contact with the cloth next to the pin guide of the print head and give a short, quick press to wet the cloth.

Turn on the printer and send a page of print to the printer (self-test can be used). Now, move the cloth a little to one side so that you have a clean spot. It may be necessary to give the cloth another shot of fluid and print out another page.

Remove the cloth from the print head and print a page without the ribbon. If you see any printing on the paper, put the cloth back into the print head and repeat the whole process.

Finally, install the ribbon and enjoy the improved print. It works great.

Send your tips and hints to MICROpendium User Notes, P.O. Box 1343, Round Rock, TX 78680.

Classified

Policy

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

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

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

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

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

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

FOR SALE

Buy/Sell Used Hardware and Software

National Used Software Club has buyers looking for TI products, as well as sellers. More than 100 TI items are listed for sale. The membership fee is only $15/year and entitles you to buy or sell computer items via the NUS/HC database. You will also receive FREE our newsletter (mailed every eight weeks). If you don't buy or sell anything during the term of your membership we will refund your annual fee.

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

Here are some of the items offered for sale:

GRAMKracker (80K)	$90.00
TI-Imager Maker (installed)	$150.00
Mini-Expansion System (Multicom)	$60.00
TI XBASIC	$20.00
MG Explorer	$25.00
Mini-Memory Module	$15.00
Editor/Assembler	$10.00
Hitchhiker's Guide to the Galaxy (XB)	$10.00

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

The ONLY monthly devoted to the TI99/4A

Subscription Fees

❑ 12 issues, USA, **$25**

❑ 12 issues, Mexico, **$30.25**

❑ 12 issues, Canada **$25**

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

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

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

Address Changes

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

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

Check/MO ❑ 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.

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

Name _____

Address _____

City _____

State _____ **ZIP** _____

The set of numbers on the left of your mailing label indicates the cover date of the last issue of your subscription.

Disks, Etc.

❑ **Back Issues**, $2.50 each. List issues: _____

_____

_____

No price breaks on sets of back issues. Free shipping USA. Add 30 cents, single issues to Canada/Mexico. Other foreign shipping 50 cents single issue surface, $1.60 airmail. Write for foreign shipping on multiple copies.

OUT OF STOCK: Vols. 1, No. 1-2; Vol. 2, No. 1; Vol. 4, No. 9

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

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

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

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

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

❑ **Magazine holders** (12/set — add $1 shipping/order)..$3.00
Number of sets _____

GENEVE DISKS (SSSD unless specified)

❑ **MDOS 1.21F** (req. SSDD or larger, for MBASIC).....$4.00

❑ **MDOS 1.21H** (req. for MBASIC).....$4.00

❑ **Myarc BASIC 3.0**.....$4.00

❑ **MY-Word V1.21** $4.00

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

GENEVE PUBLIIC DOMAIN DISKS

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

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

SECOND CLASS