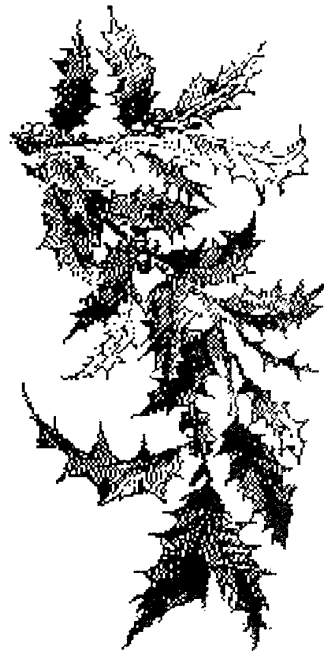
Covering the TI99/4A and the Myarc 9640

MICROpendium

Volume 12 Number 11 December 1995 $3.50

Fest West vendors get extra setup, takedown time ... Extended BASIC 12 O'Clock provides a challenge ... Keeping track of petrol costs ... Decision making ... The Art of Assembly and more random numbers ... Disk drive step motors and the genius of double density ... Funnelweb editor v.5.0 printer codes using the all-CHAR mode to avoid the use of the formatter ...The Home Computer Era part 7 of the series gives a timeline ... Reviews: Term 80 ... User Notes: Search string corrected, a WINDOW:1-6 update, and Q-Bert for the 9640 ....



Happy holidays!

CONTENTS

MICROpendium

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

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

*READ THIS

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

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

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

TEX COMP LTD.

THE TRUCK LOAD SALE IS OVER. WITH YOUR HELP WE MANAGED TO SALE OFF ENOUGH EQUIPMENT TO GET EVERYTHING IN THE STORAGE AREA. THANKS FOR YOUR SUPPORT AND HELP.

SUPER AMS CARD

TAKE YOUR TI TO THE NEXT LEVEL! This memory expansion card breaks the 32K barrier allowing for very large E/A programs or multiple XB programs using the included XB packer. The number of XB programs loadable at one time depends on the memory option ordered. Comes with modified assembler allowing use of all the new features and a multi-page manual on disk. IN THE NEXT COUPLE OF MONTHS YOU WILL BE SEEING NEW PROGRAMS JUST FOR THE AMS CARD FROM TEX-COMP.

There are two versions to chose from
128K version $85
256K version (recommended)$100
128K to 256K upgrade $25
ADD $5 EXTRA FOR SHIP / INSURANCE

OUR 96 CATALOG IS NOW READY !

If you Ordered our new catalog and have not received it as of this add please call or write us. If you have not ordered a catalog then now is the time to order our new 30 page catalog.

TO RECEVE THE NEW 30 PAGE 1996 TEX-COMP CATALOG SEND $2.50 TO THE ADDRESS BELOW. PS REMEMBER TO INCLUDE YOUR MAILING ADDRESS WITH ORDER !

CAN'T FIND DISKETTES FOR YOUR TI ?

By special arrangement with the manufacturer TEX-COMP Ltd will be able to meet and fullfill your diskette needs well into the 21 st century.

**BULK DISKETTES*

BLACK	.19 EA (MIN 50)
COLOR (R,O,Y,G,B,W)	.34 EA (MIN 50 PER COLOR)
"FLIPPY"	.45 EA (MIN 50)
"FLIPPY COLOR (R,O,Y,G,B,W,)	..50 EA (MIN 50 PER COLOR)
PAPER SLEEVES	ADD .02 PER DISK
TYVEC SLEEVES	ADD .04 PER DISK
LABELS	100 PER KIT $2.00

R=RED,O=ORANGE,Y=YELLOW,G=GREEN,B=BLUE,W=WHITE

MAGIC MEMORY

MAGIC MEMORY belongs in every TI99/4A, give your TI the gift of power. Magic Memory brings many of the average users needs right into one handy cartridge. With features like disk manager functions that rival DM1000, and editor assembler loader functions. It also offers the advanced user easy access to many programs that are used regularly like memory manager, advanced sector editor, debugger, and an editor/assembler cartridge that does not need the editor or assembler files off the disk (a real time saver) all in one convenient cartridge. MAGIC MEMORY comes with a 58 page owners manual that will help you get started and explain all the features contained in the cartridge. **$49.95**

4 A FLYER	$24.95	MIDNIGHT MASON	$24.95
BARRAGE	$29.95	MINI WRITER	$24.95
BEYOND PARSEC	$24.95	MUNCHMAN II	$24.95
BEYOND WORD WRITER	$39.95	PARAPRINT	$69.95
BLACK HOLE	$24.95	TI PLANNER	$24.95
BOXER	$24.95	PROTYPER	$24.95
BREAKTHGH/ BREAKOUT	$24.95	QMAZE	$29.95
BURGER BUILDER	$24.95	RED BARON	$39.95
COMPUTER WAR	$29.95	SORGON II	$29.95
CONCOLE CALC	$24.95	SPACE PATROL	$24.95
D-STATION I	$24.95	SPOT SHOT	$29.95
D-STATION II	$24.95	SPY'S DEMISE	$24.95
DRAGON FLYER	$29.95	STARGAZER	$39.95
DESKTOP PUBLISHER	$39.95	STAR RUNNER	$29.95
ESCAPE	$24.95	STAR TRAP	$24.95
FACE CHASE	$24.95	STRIKE 3	$29.95
GREAT WORD RACE	$29.95	TI TOAD	$24.95
JUMPY	$29.95	TI WORKSHOP	$49.95
JUNK MAN JR	$29.95	WORD WRITER	$29.95
MAGIC MEMORY	$49.95	WORD WRITER EXTRA	$29.95
MANCALA	$29.95		
MICRO PINBALL	$29.95		
MICRO TENNIS	$29.95		

DBT Data BioTics

SOLID STATE CARTRIDGES

TEX COMP LTD.
AMERICAS NUMBER ONE TI COMPUTER RETAILER
425 E. ARROW HWY. #732
GLENDORA, CA 91740

VISA & MASTER CARD EXCEPTED

ADD 5% SHIPPING {$5.00 MINIMUM}
FOR ORDERS OUTSIDE USA ADD 10%

ORDER BY PHONE
(800) 846-3474

ALL PRICES ARE FOR CASH/CHECK, ADD 3.5% FOR CREDIT CARD ORDERS

COMMENTS

Shareware authors deserve support

It's unfortunate, but shareware is a concept that sounds workable but in the end isn't. Witness the withdrawal of Edward Schwartz from the field. Schwartz is the developer of TI Emulator, a software package that runs on PCs and emulates a TI. Schwartz developed the program while attending college. He put it on the market as shareware and then improved it. The final version is V6.0. There will likely be no other.

A lot of TIers have made a run at the shareware market, starting out with the hope that users would be honest, or at least recognize the value of their work and compensate them for it, even if in small amounts. The unfortunate reality is that few users support authors of shareware with money.

What makes it doubly difficult for TI shareware authors is that there are limited means for protecting their work. Some programs don't avail themselves of being partially disabled so as to give incentive for users to send money for a fully operational version. Jeffrey Brown is doing this with Term 80. Term 80 is available free from networks but it is not fully functional. The fully functional version requires a payment to the author. It seems that this may be the only way that shareware authors can expect payment for their work. It's a hassle, of course. Usually, shareware authors would rather spend their time programming and developing than taking care of business. But there's little choice if they also want to be paid for their efforts.

Think about this the next time you run a piece of shareware. Better yet, think about sending the shareware author a payment. He'll appreciate it. And who knows, it could lead to the development of another nifty piece of software. And just think, you'll have played a part.

TERM 80

Charlie Good reviews Term 80 this month and, as a Geneve owner, I feel left out. Term 80 does a lot of things that few other terminal programs do. It does some things that no terminal program does. It supports high-speed modems and, of course, it provides a TI99/4A with an 80-column terminal screen. It's a very useful piece of software for TI users who use the Internet, which pretty much requires an 80-column screen.

WINDOWS1:6 DOCUMENTATION

You may recall that in November we ran a program called WINDOWS1:6 by W. Leonard Taffs. It's a complex program and readers may have trouble using it with the limited documentation we provided. That's why we'll be including as much of the docs as we can with this month's MICROpendium disk (it takes up almost 300 sectors). The remainder will be included on the January disk.

—JK

FEEDBACK

Authors comment

A note about your publishing "WINDOW:1-6"; I did not realize that the *README file (from which I believe you extracted the program notes) did not state two important considerations in this program:

1) The necessity of obtaining (and reading) the docs for this program.

2) The need to understand the necessity of doing a "CALL FILES" for more than three files (allowing one file for the use of the disk catalog program within the program) if not using a RAMdisk.

The docs step through the entire program (with screen illustrations), explaining the entire program in detail. More importantly, there are user input options beyond those shown in the "Help" screen. There were not enough program bytes to program all these in on-screen help.

When I used the term "simultaneous" it was not meant *literally*. It referred to being able to switch back and forth between files. I hope this did not mislead your readers!

W. Leonard Taffs
Tucson, Arizona

I realize that as articles are reprinted from one newsletter to another, credit to the author can be lost. Such was the case in an article that appeared in the User Notes section of the February 1995 issue of MICROpendium.

The article in question was titled: "Using Telco to Upload ASCII Text Files." I am the original author of this article as published in the January/February 1994 issue of the Cleveland Area TI99/4A User Group's newsletter.

Thank you for considering the uploading text hints worthy of publishing. I hope these uploading techniques will be helpful.

Glenn Bernasek
TI-Chips
Cleveland, Ohio

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

COMPETITION COMPUTER

800-471-1600 (Nationwide & Canada ORDERS ONLY)

BUY - SELL - TRADE

350 MARCELLA WAY MILLBRAE, CALIFORNIA 94030

HAPPY NEW YEAR! A GOOD WAY TO BRING IN THE NEW YEAR IS WITH OUR NEW 1996 CATALOG! STILL FREE AND CONTAINING MORE FOR SALE THAN ANY OTHER IT IS THE YELLOW PAGES OF THE TI WORLD

COMPETITON COMPUTER BRINGS BACK THE NAVARONE LINE OF CARTRIDGE SOFTWARE , INCLUDING R S SOFTWARE PUBLISHING TITTLES NAVARONE ACQUIRED RIGHTS TO AFTER THE DEMISE OF ROMOX WE ARE CURRENTLY MANUFACTURING THE FOLLOWING CARTRIDGES:

* NEW FOR DECEMBER 95

ANT EATER.....$20
CHICKEN COOP.....$20
CONSOLE WRITER.....$25
DATA BASE MANAGEMENT SYSTEM $30
(FOR DISK BASED SYSTEMS)
DATA BASE MANAGEMENT ENTRY...$20
DATA BASE MANAGEMENT SORT.....$20
(UTILITY FOR DBM ENTRY)
DATA BASE MANAGEMENT REPORT $20*
(REPORT WRITER FOR DBM)
DISK FIXER.....$20
FROG STICKERS.....$20
HEN PECKED.....$20
HOMEWORK HELPER.....$25
HOMEWORK HELPER+.....$30
KING OF THE CASTLE.....$20*
PAINT 'N PRINT A (GP-100,CP-700).....$30
PAINT 'N PRINT B (GP-550, OKIDATA)$30
PAINT 'N PRINT C (IBM,GEMINI).....$30
(FOR OTHER PRINTERS SEE DISK SOFTWARE LISTINGS BELOW)
PRINCESS AND THE FROG.....$20
ROMOX DEMO.....$20
ROTOR RAIDERS.....$20
SPEED READING A (TEENS +ADULTS).$25
SPEED READING B (FOR CHILDREN)....$25
SUPER DUPER (A FAST DISK COPIER)..$20
TOPPER.....$20
TYPO-MAN.....$20
TYPO II.....$20

DISK SOFTWARE:

DATA BASE MANAGEMENT TUTORIAL $3*
EXTENDED GRAPHICS DISK.....$15
EXTENDED GRAPHICS TAPE.....$15
(ADDS COMMANDS AND PRINTERS FOR PAINT 'N PRINT)
MUSIC EDITOR.....$10
SPRITE EDITOR.....$10

SPEED READING DATA DISKS.....$10
(NOT REQUIRED BUT USEFUL WHEN YOU GET TIRED OF READING THE BUILT-IN READING MATTER IN THE CARTRIDGE.)

WE ALSO HAVE THE FOLLOWING FUNWARE CARTRIDGES.

AMBULANCE.....$20
DRIVING DEMON.....$20
HEN HOUSE.....$20
RABBIT TRAIL.....$20
SCHNOZ-OLA.....$20
ST. NICK.....$20
VIDEO VEGAS.....$20

RARE TI SOFTWARE:

PHD 5066 TI PILOT (REQUIRES P-CODE CARD).....$20
(DISK AND 140 PAGE MANUAL)
PHD 5068 COURSE DESIGNER AUTHORING PACKAGE.$15
(TWO DISK SET AND 54 PAGE MANUAL)

IEEE -488 CARD (GPIB) SOFTWARE WRITTEN BY TI FOR IN-HOUSE USE ONLY, USED TO TEST COMPONENTS NOT THE GPIB CARD ITSELF $5

SMU ELECTRICAL ENGINEERING LIBRARY DISKS +MANUAL $20

REQUIRES THE SMU COMMAND MODULE TO RUN

UNRELEASED SMU ELECTRICAL ENGINEERING DISKS+MANUAL $20
THESE DISKS ARE AVAILABLE IN TWO FORMATS:ORIGINAL FORMAT WHICH REQUIRES THE SMU COMMAND MODULE AND A MODIFIED FORMAT WHICH DOES NOT REQUIRE THE SMU COMMAND MODULE TO RUN. PLEASE SPECIFY WHICH FORMAT YOU NEED.

NEW HARDWARE:
DUAL HALF HEIGHT 3.5" DISK DRIVE KIT.....$70

Music Editor (TI only)

Music Editor generates sound definitions data. You can enter notes on a large "music sheet" displayed on the screen. The program generates the data statements required to reproduce sounds in either Basic or Assembler format. The data can be saved on your disk for future editing.

SPEED READING

Improves reading speed and comprehension. Versions for children, teens, and adults. Cartridge Software, no extra equipment required.

Version A for teens and adults: This new program provides extensive practice in taking in more print faster. Frequent tests prove that not only is your rate of reading greatly increased, but you are also improving your level of comprehension. Complete workbook included.

Version B for children: A comprehensive course created especially for children 8 through 13. Practice drills are geared to increasing reading speed and improving comprehension levels. Progress tests and scores add to satisfaction and confidence. Complete workbook included.

DISK FIXER

*INCLUDES HIDDEN POWERS

Unlocks the secrets of the disk and lets you access hidden or "lost" information. Disk Fixer is a must for the serious TI 99/4A user who has a "sick disk" which suffers from a damaged directory. This program lets you recover unscathed information by searching your disk by sector rather than file name. Disk Fixer lets you display/print the actual binary contents with a single command. It allows you to change any byte on any sector, even move data from one sector to another.

WE HAVE ADDED HUNDREDS OF TITLES AND OUR PRICES AVERAGE 50% OF WHAT TED USED TO CHARGE WHEN HE RAN COMPETITION
STILL NO EXTRA CHARGE FOR CREDIT CARD ORDERS
NO "HANDLING" CHARGES
NON-PROFIT SHIPPING CHARGES, YOU ONLY PAY WHAT WE PAY
WE ACCEPT PERSONAL CHECKS(4 BUSINESS DAYS FOR CLEARANCE)
MONEY ORDERS
MASTERCARD AND VISA
WE WILL BE AT TI FEST WEST '96 AND WE WILL BE BUYING AS WELL AS SELLING SO BRING YOUR GOODIES. WE ARE LOOKING FOR ORIGINAL THIRD PARTY CASSETTE SOFTWARE FOR OUR CONSOLE ONLY CUSTOMERS IF YOU HAVE SOME OF THIS STASHED AWAY BRING IT. TOMY TUTOR SOFTWARE, RARE ITEMS WANTED FOR THE TI , AND SOME NOT SO RARE . IF IT IS SOMETHING WHICH I PERSONALLY DON'T OWN YOU WILL GET TOP DOLLAR FROM ME .

Sprite Editor (TI only)

With Sprite Editor you can "draw" characters on the screen in two grid patterns, single (8×8) and magnified (16×16). Once your characters have been drawn, the program generates the data statements needed to reproduce them in either Basic or Assembler format.

Special Editing functions allow your character patterns to be copied and reversed. The data can be saved on your disk for future editing.

PAINT 'N' PRINT

PAINT 'N PRINT IS THE MOST POWERFUL AND VERSATILE GRAPHICS PROGRAM YOU CAN BUY FOR A CONSOLE ONLY SYSTEM. NO MEMORY EXPANSION OR OTHER PERIPHERALS NEEDED TO USE THIS CARTRIDGE BASED GRAPHICS PROGRAM ONLY A JOYSTICK OR TRACKBALL. YOU DON'T EVEN NEED A PRINTER! SELECT FROM 32 DIFFERENT BRUSHES,FULL USE OF BIT MAP COLOR,MAGNIFY SECTIONS OF THE SCREEN, ENTER TEXT INTO YOUR PICTURE FOR GRAPHS,SAVE TO CASSETTE OR DISK

TI-99/4A™ CD ROM

CD ROM UPDATE: WE WILL BE DEMONSTRATING THE TI 99/4A CD ROM AT FESTWEST . NOW SHIPPING THE TI CD ROM FOR SINGLE USER/SUBSCRIBERS . FIRST ISSUE IS 140MB ARCHIVED FOREIGN SUBSCRIBERS SHOULD BE AWARE EXTRA POSTAGE COSTS WILL MEAN FEWER BUT LARGER ISSUES FOR OVERSEAS SUBSCRIBERS. PRICE REMAINS AT $100 FOR A 650MB SUBSCRIPTION(ABOUT 4-5 ISSUES FOR DOMESTIC SUBSCRIPTIONS)

IMAGINE 650 MEGABYTES OF SOFTWARE WRITTEN FOR THE TI99/4A AND 9640 ON ONE CDROM . THAT IS WHAT YOU WILL HAVE AT THE END OF YOUR SUBSCRIPTION TO THE TI 99/4A CDROM. FOR THE PRICE OF 30 -50 PUBLIC DOMAIN DISKS YOU WILL RECEIVE 1900 DISKS INITIALY AND MANY MORE IN PERIODIC UPDATES UNTIL YOU WILL HAVE 650 MEGABYTES OF TI SOFTWARE. THATS A LOT OF SOFTWARE FOR YOUR COST OF ONLY $100 . SEND YOUR ORDER FORM WITH A CHECK OR MONEY ORDER PAYABLE TO MILLBRAE CD ROM,350 MARCELLA WAY, MILLBRAE, CA 94030. VOICE:(415) 697-1108 ,FAX:(415)697-7406

YOUR NAME:_____
ADDRESS:_____
CDROM PLAYER AND PLATFORM INFORMATION:
MY CDROM PLAYER IS MADE BY:_____MODEL#:_____
I HAVE IT CONNECTED TO: IBM/COMPATIBLE___MAC___HORIZON SCSI____
IF YOU CHECKED HORIZON SCSI: IT IS CONNECTED TO:TI99/4A___9640___
I USE THE FOLLOWING OPERATING SYSTEM WITH MY CDROM PLAYER:
WINDOWS___DOS___VERSION:____(3.1 FOR WINDOWS FOR EXAMPLE)
SYSTEM 7.X___MYARC DOS(MDOS)___VERSION_____ TI99/4A____
PLEASE INCLUDE YOUR CDROM PLAYERS SPEC SHEET WITH YOUR ORDER

Fest West gives vendors extra time for setup

Organizers for Fest West '96, sponsored by the SouthWest Ninety Niners, have announced a new format for the event, schedules Feb. 16-18 in Tucson, Arizona.

Vendors will set up Friday evening, and preregistration will be held at that time, with the main event 9 a.m.-6 p.m. Saturday, with door prizes given at the end of the day. Friday and Saturday evenings a hospitality room will be available. Vendors will be able to pack up from 9 a.m. to noon Sunday.

Vendor booths are $20 for the first table, and $15 each for additional tables. Booth reservations postmarked in December 1995 may take $5 off booth costs; reservations for booths must be in by Jan. 20, and an advance deposit of $20 must accompany any reservation request.

The site of the event, the Ramada Inn University at 1601 N. Oracle Rd. in Tucson will provide single or double rooms for $55 per night to Fest West attendees who mention the event when making reservations. To reserve a room, call 1-800-777-2999 or (520) 623-6666. Organizers ask that attendees make reservations by Jan. 15. Airport shuttle service to the hotel is available at $5 per person round trip. The shuttle service requires a minimum 48 hours notice, so organizers advise attendees to reserve van service when making hotel reservations.

For further information, call BJ Mathis, (520) 747-5046; Tom Wills, (520) 886-2460; or the Cactus Patch BBS (520) 290-6277; or write SouthWest Ninety Niners, P.O. Box 17831, Tucson, AZ 85731.

READER TO READER

Kurt Radowisch, Grossbauerstr. 24, A-1210 Vienna, Austria, writes:

I am using a HFDC (hard and floppy disk controller) at >1199, This enables me to run 3.5-inch disks at 1.4 MB. But I was told that PC-Transfer cannot be used with this card (or can it?) Is there any other program around for the Geneve that can convert PC files to TI files and vice versa? I would greatly appreciate any information.

Try the following on a Geneve in ABASIC 3.00:

PRINT VALHEX("8000") returns -32768 (what else?)

PRINT VALHEX("8000")+32768 RETURNS -1 (why?)

```
10 A=VALHEX("8000")        This program returns 0,
20 PRINT A+32768           which one could expect.
10 A=VALHEX("8000")+32768  This program returns -1,
20 PRINT A                 which seems pretty fishy to me
```

Who has an explanation for this phenomenon?

Does anybody know how to switch on and off CapsLock and NumLock by software?

Joe Corrigan, 805 Verna Dr., Vestal, NY 13850, phone (607) 748-4828, writes:

My 1080A Legend nine-pin printer (PIO) has burned out its printhead. No replacement can be found. I have use of an Okidata 390 (24-pin Microline (PIO input) that I would like to use, but my CorComp RS232 card in the peripheral expansion box will not drive the printer. Does anyone have any suggestions?

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

12OCLOCK

A challenging puzzle in Extended BASIC

By WESLEY R. RICHARDSON

The following program originally appeared in user group newsletters. Richardson was a member of the Northcoast 99ers of Cleveland, Ohio, when it was written.—Ed.

12 O'CLOCK is an Extended BASIC program for puzzle lovers. The objective of 12 O'CLOCK is to orient all 18 clocks to the same 12 o'clock vertical position using the rotate buttons 1+5, 2+6, 3+7, 4+8 and the A+E, B+F, C+G, D+H gears. Rotation is clockwise for the clock adjacent to the number. Gears are engaged, "X" for the gear nearest the letter, and disengaged "0"; pressing the gear letter will change it to 0. Number pairs and letter pairs given above work in opposite directions.

Moves are counted for each rotation button used, but not incremented for sequential use of the same button. Although, at first, it will seem impossible to solve this puzzle, it is not impossible.

A green/red indicator is used when the program will/will not accept key presses. You may hold a key down for repeated operation.

The programs uses all 28 possible sprites and all available characters from 32 to 143. The clock positions are maintained in the C(18) array, and the gear positions in the D(4) array. Two functions are defined — E(Z) and F(Z) — since these modular 12 routines are used repeatedly in the calculations.

The program size of 50 sectors causes Extended BASIC to save it in INT/FIX 254 format, rather than program image. This means that 12 O'CLOCK can't be run from an autoloading menu but must be loaded using OLD.DSKx.12OCLOCK.

Editor's note: This program is difficult to type in. It took use nearly a full day to enter and check it for data entry errors. If you have trouble after typing it in, refer to the checksums at the end of each program line. The program is also available on the December MICROpendium disk for $4.

12OCLOCK

```
100 REM 12-O'CLOCK !044
110 REM WESLEY R. RICHARDSON
, SEPTEMBER 1989 !046
120 REM NORTHCOAST 99ERS, CL
EVELAND, OH !248
130 REM TI99/4A EXTENDED BAS
IC !096
140 REM VARIABLES A$,B$,C(18
),D(4),E(Z),F(Z),I,K,M,N,P,Q
,R,S,W,X(18),Y(18),Z !028
150 CALL CLEAR :: CALL MAGNI
FY(3)!051
160 DIM C(18),D(4),X(18),Y(1
8)!037
170 DEF E(Z)=-(Z+1)*(Z<>11)!
204
180 DEF F(Z)=-(Z-1)*(Z<>0)-1
1*(Z=0)!202
190 GOTO 230 :: CALL CHAR ::
 CALL CHARPAT :: CALL CHARSE
T :: CALL COLOR :: CALL DELS
PRITE :: CALL HCHAR !128
200 CALL KEY :: CALL LOCATE
:: CALL PATTERN :: CALL POSI
TION :: CALL SCREEN :: CALL
SPRITE !215
210 A$,B$,I,K,M,N,P,Q,R,S,W,
Z !162
220 !@P- !064
230 CALL CLEAR :: DISPLAY AT
(4,9):"12-O'CLOCK" :: DISPLA
Y AT(6,2):"by WESLEY R. RICH
ARDSON" !062
240 DISPLAY AT(8,1):"THE OBJ
ECTIVE IS TO ORIENT" :: DISP
LAY AT(9,1):"ALL 18 CLOCKS T
O THE SAME" !095
250 DISPLAY AT(10,1):"12 O'C
LOCK VERTICAL POSITION" :: D
ISPLAY AT(11,1):"USING THE R
OTATE BUTTONS" !150
260 DISPLAY AT(12,1):"1+5, 2
+6, 3+7, 4+8 AND THE" :: DIS
PLAY AT(13,1):"A+E, B+F, C+G
, D+H GEARS." !252
270 DISPLAY AT(15,1):"ROTATI
ON IS CLOCKWISE FOR" :: DISP
LAY AT(16,1):"THE CLOCK ADJE
CENT TO THE" !235
280 DISPLAY AT(17,1):"NUMBER
. GEARS ARE ENGAGED," :: DIS
PLAY AT(18,1):"`X' FOR THE G
EAR NEAREST THE" !126
290 DISPLAY AT(19,1):"LETTER
. NUMBER PAIRS AND" :: DISPL
AY AT(20,1):"LETTER PAIRS GI
VEN ABOVE" !112
300 DISPLAY AT(21,1):"WORK I
N OPPOSITE DIRECTIONS" :: DI
SPLAY AT(23,7):"PRESS ANY KE
Y" !155
310 CALL KEY(0,K,S):: IF S=0
 THEN 310 :: CALL CLEAR :: D
ISPLAY AT(12,4):"LOAD:" !007
320 RANDOMIZE :: FOR I=1 TO
18 :: C(I)=INT(12*RND):: NEX
T I :: CALL HCHAR(12,11,58)!
218
330 C(10)=12*(1+(C(1)=0))-C(
1):: C(12)=12*(1+(C(3)=0))-C
(3)!094
340 C(16)=12*(1+(C(7)=0))-C(
7):: C(18)=12*(1+(C(9)=0))-C
(9):: CALL HCHAR(12,12,58)!0
56
350 FOR I=1 TO 4 :: D(I)=1 :
: NEXT I !208
360 FOR I=0 TO 5 :: Y(1+3*I)
,Y(2+3*I),Y(3+3*I)=1+32*I ::
 NEXT I !214
370 FOR I=0 TO 2 :: X(1+I),X
(4+I),X(7+I),X(10+I),X(13+I)
,X(16+I)=57+32*I :: NEXT I !
229
380 CALL HCHAR(12,13,58):: N
=0 :: P=5 :: Q=2 :: R=16 ::
CALL COLOR(7,16,1,8,16,1)!07
1
390 CALL CHARPAT(61,A$,81,B$
):: CALL CHAR(35,A$,59,B$)!2
32
400 CALL CHARPAT(76,A$,78,B$
):: CALL CHAR(33,A$,64,B$)!2
38
410 CALL CHARPAT(79,A$,39,B$
```

(See Page 8)

12OCLOCK—

(Continued from Page 7)

```
):: CALL CHAR(74,A$,73,B$)!2
43
420 CALL HCHAR(12,14,58):: C
ALL CHAR(80,"00E0FCFFFFFFFF
FFFFFFFFFFFFFFFF0000000080C
0E0F0F8F8FCFCFCFEFEFE")! CIR
CLE UR !172
430 CALL CHAR(84,"FFFFFFFFFF
FFFFFFFFFFFFFFFFFCE000FEFEFE
FCFCFCF8F8F0E0C08000000000")
! CIRCLE LR !239
440 CALL HCHAR(12,15,58):: C
ALL CHAR(88,"7F7F7F3F3F3F1F1
F0F07030100000000FFFFFFFFFFF
FFFFFFFFFFFFFFF3F0700")! CIR
CLE LL !251
450 CALL CHAR(92,"0000000001
03070F1F1F3F3F3F7F7F7F00073F
FFFFFFFFFFFFFFFFFFFFFFFFFFF")
! CIRCLE UL !070
460 CALL HCHAR(12,16,58):: C
ALL CHAR(96,"008080808080808
0808080808080808000000000000
0000000000000000000")! 12
O'CLOCK !033
470 CALL CHAR(100,"000000010
302060C081810302060C08000000
0000000000000000000000000"
)! 1 O'CLOCK !055
480 CALL HCHAR(12,17,58):: C
ALL CHAR(104,"00000000000000
00000000030E3860C00000000000
0000001870C08000000000")! 2
O'CLOCK !007
490 CALL CHAR(108,"FF0000000
0000000000000000000000FE000
0000000000000000000000000"
)! 3 O'CLOCK !065
500 CALL HCHAR(12,18,58):: C
ALL CHAR(112,"C060380E030000
0000000000000000000000080
C0701800000000000000")! 4
O'CLOCK !009
510 CALL CHAR(116,"80C060203
01018080C0602030100000000000
0000000000000000000000000"
)! 5 O'CLOCK !066
520 CALL HCHAR(12,19,58):: C
ALL CHAR(120,"00000000000000
0000000000000000000101010101
0101010101010101010100")! 6
O'CLOCK !179
530 CALL CHAR(124,"000000000
0000000000000000000001030
6040C081810306040C080000000"
)! 7 O'CLOCK !071
540 CALL HCHAR(12,20,58):: C
ALL CHAR(128,"0000000001030E
180000000000000000003061C70C0
0000000000000000000000000")! 8
O'CLOCK !048
550 CALL CHAR(132,"000000000
0000000000000000000007F00000
0000000000000000000000000FF"
)! 9 O'CLOCK !054
560 CALL HCHAR(12,21,58):: C
ALL CHAR(136,"00000000000000
00180E0301000000000000000000
000000000000C0701C0603")! 10
O'CLOCK !040
570 CALL CHAR(140,"000000000
0000000000000000000000000000
080C04060301018080C04060301"
)! 11 O'CLOCK !112
580 CALL HCHAR(12,22,58):: C
ALL CHAR(36,"030408080808040
3000000000000000C020101010
1020C000000000000000000")! G
EAR O LOW !153
590 CALL CHAR(40,"0000000000
0000003040808080804030000000
0000000000C0201010101020C0")
! GEAR O HIGH !169
600 CALL HCHAR(12,23,58):: C
ALL CHAR(44,"000000000000000
0080402010102040800000000000
0000010204080804020100")! GEA
R X HIGH !078
610 CALL CHAR(60,"0804020101
0204080000000000000000102040
8080402010000000000000000")
! GEAR X LOW !102
620 FOR I=3 TO 6 :: CALL COL
OR(I,Q,1):: NEXT I :: CALL C
LEAR :: CALL CHAR(76,RPT$("F
",64))!104
630 CALL SCREEN(9):: DISPLAY
 AT(1,3):"1\^PR\^PR\^PR2
\^PR" :: DISPLAY AT(2,4):"]_
QS]_QS]_QS     ]_QS" !134
640 DISPLAY AT(3,4):"XZTVXZT
VXZTV     XZTV" :: DISPLAY A
T(4,3):"AY[UWY[UWY[UW     Y[
UW" !108
650 DISPLAY AT(5,4):"\^PR\^P
R\^PRB 12 JIC!JCK" :: DISPLA
Y AT(6,4):"]_QS]_QS]_QS" !22
5
660 DISPLAY AT(7,4):"XZTVXZT
VXZTV   12345678" :: DISPL
 AT(8,3):"DY[UWY[UWY[UW   AB
CDEFGH" !007
670 DISPLAY AT(9,4):"\^PR\^P
R\^PRC" :: DISPLAY AT(10,4):
"]_QS]_QS]_QS    ;#E@D" !069
680 DISPLAY AT(11,4):"XZTVXZ
TVXZTV" :: DISPLAY AT(12,3):
"4Y[UWY[UWY[UW3" !229
690 DISPLAY AT(13,3):"5\^PR\
^PR\^PR6" :: DISPLAY AT(14,4
):"]_QS]_QS]_QS" !237
700 DISPLAY AT(15,4):"XZTVXZ
TVXZTV  1:5  A:E" :: DISPLA
Y AT(16,3):"EY[UWY[UWY[UW
2:6  B:F" !228
710 DISPLAY AT(17,4):"\^PR\^
PR\^PRF  3:7  C:G" :: DISPLA
Y AT(18,4):"]_QS]_QS]_QS  4
:8  D:H" !217
720 DISPLAY AT(19,4):"XZTVXZ
TVXZTV" :: DISPLAY AT(20,3):
"HY[UWY[UWY[UW" !204
730 DISPLAY AT(21,4):"\^PR\^
PR\^PRG" :: DISPLAY AT(22,4)
:"]_QS]_QS]_QS" !199
740 DISPLAY AT(23,4):"XZTVXZ
TVXZTV" :: DISPLAY AT(24,3):
"8Y[UWY[UWY[UW7" !243
750 FOR I=1 TO 18 :: CALL SP
RITE(#I,96+4*C(I),P,Y(I)+16*
(C(I)>2)*(C(I)<9),X(I)+16*(C
(I)>5)):: NEXT I !206
760 CALL SPRITE(#19,44,R,17,
65,#20,60,R,33,97,#21,60,R,6
5,97,#22,44,R,49,65)!095
770 CALL SPRITE(#23,40,R,113
,65,#24,36,R,129,97,#25,36,R
,161,97,#26,40,R,145,65)!037
780 CALL SPRITE(#27,76,9,89,
165,#28,96,5,1,193)!207
790 REM MAIN LOOP !057
800 W=0 :: FOR I=1 TO 18 ::
W=W+C(I):: NEXT I :: IF W<>0
 THEN 820 !143
810 CALL SCREEN(14):: FOR W=
1 TO 400 :: NEXT W :: CALL S
CREEN(9)!127
820 CALL COLOR(#27,13)!103
830 CALL KEY(0,K,S):: IF S=0
 THEN 830 :: CALL COLOR(#27,
7):: IF K=81 THEN 1950 !160
840 IF (K>64)*(K<73)THEN 860
 !118
```

(See Page 9)

12OCLOCK—

(Continued from Page 8)

```
850 IF (K>48)*(K<57)THEN 910
 ELSE 790 !015
860 REM GEAR ABCDEFGH !029
870 K=K-64 :: IF K<5 THEN D(
K)=0 !187
880 IF K>4 THEN D(K-4)=1 !17
5
890 CALL PATTERN(#19,40+4*D(
1),#20,36+24*D(2),#21,36+24*
D(3),#22,40+4*D(4))!105
900 CALL PATTERN(#23,44-4*D(
1),#24,60-24*D(2),#25,60-24*
D(3),#26,44-4*D(4)):: GOTO 7
90 !096
910 REM ROTATE 12345678 !077
920 K=K-48 :: IF M<>K THEN M
=K :: N=N+1 !006
930 ON K GOTO 940,1060,1180,
1300,1420,1540,1660,1780 !06
4
940 REM ROT 1 !000
950 C(1)=E(C(1)):: C(10)=F(C
(10))!109
960 IF D(1)THEN C(2)=E(C(2))
:: C(4)=E(C(4)):: C(5)=E(C(5
))!063
970 IF D(1)*D(2)+(D(1)=0)*(D
(2)=0)THEN C(3)=E(C(3)):: C(
12)=F(C(12))!230
980 IF D(1)*(D(2)+D(3))THEN
C(6)=E(C(6))!247
990 IF D(1)*D(4)+(D(1)=0)*(D
(4)=0)THEN C(7)=E(C(7)):: C(
16)=F(C(16))!250
1000 IF D(1)*(D(3)+D(4))THEN
 C(8)=E(C(8))!253
1010 IF D(1)*D(3)+(D(1)=0)*(
D(3)=0)THEN C(9)=E(C(9)):: C
(18)=F(C(18))!000
1020 IF D(1)=0 THEN C(11)=F(
C(11)):: C(13)=F(C(13)):: C(
14)=F(C(14))!031
1030 IF (D(1)=0)*((D(2)=0)+(
D(3)=0))THEN C(15)=F(C(15))!
198
1040 IF (D(1)=0)*((D(3)=0)+(
D(4)=0))THEN C(17)=F(C(17))!
204
1050 GOTO 1900 !194
1060 REM ROT 2 !001
1070 C(3)=E(C(3)):: C(12)=F(
C(12))!117
1080 IF D(2)THEN C(2)=E(C(2)
):: C(5)=E(C(5)):: C(6)=E(C(
6))!068
1090 IF D(1)*D(2)+(D(1)=0)*(
D(2)=0)THEN C(1)=E(C(1)):: C
(10)=F(C(10))!222
1100 IF D(2)*(D(1)+D(4))THEN
 C(4)=E(C(4))!244
1110 IF D(2)*D(3)+(D(2)=0)*(
D(3)=0)THEN C(9)=E(C(9)):: C
(18)=F(C(18))!002
1120 IF D(2)*(D(3)+D(4))THEN
 C(8)=E(C(8))!254
1130 IF D(2)*D(4)+(D(2)=0)*(
D(4)=0)THEN C(7)=E(C(7)):: C
(16)=F(C(16))!252
1140 IF D(2)=0 THEN C(11)=F(
C(11)):: C(14)=F(C(14)):: C(
15)=F(C(15))!036
1150 IF (D(2)=0)*((D(1)=0)+(
D(4)=0))THEN C(13)=F(C(13))!
195
1160 IF (D(2)=0)*((D(3)=0)+(
D(4)=0))THEN C(17)=F(C(17))!
205
1170 GOTO 1900 !194
1180 REM ROT 3 !002
1190 C(9)=E(C(9)):: C(18)=F(
C(18))!141
1200 IF D(3)THEN C(8)=E(C(8)
):: C(5)=E(C(5)):: C(6)=E(C(
6))!081
1210 IF D(3)*D(4)+(D(3)=0)*(
D(4)=0)THEN C(7)=E(C(7)):: C
(16)=F(C(16))!254
1220 IF D(3)*(D(1)+D(4))THEN
 C(4)=E(C(4))!245
1230 IF D(2)*D(3)+(D(2)=0)*(
D(3)=0)THEN C(3)=E(C(3)):: C
(12)=F(C(12))!234
1240 IF D(3)*(D(1)+D(2))THEN
 C(2)=E(C(2))!239
1250 IF D(1)*D(3)+(D(1)=0)*(
D(3)=0)THEN C(1)=E(C(1)):: C
(10)=F(C(10))!224
1260 IF D(3)=0 THEN C(17)=F(
C(17)):: C(14)=F(C(14)):: C(
15)=F(C(15))!049
1270 IF (D(3)=0)*((D(1)=0)+(
D(4)=0))THEN C(13)=F(C(13))!
196
1280 IF (D(3)=0)*((D(1)=0)+(
D(2)=0))THEN C(11)=F(C(11))!
190
1290 GOTO 1900 !194
1300 REM ROT 4 !003
1310 C(7)=E(C(7)):: C(16)=F(
C(16))!133
1320 IF D(4)THEN C(8)=E(C(8)
):: C(5)=E(C(5)):: C(4)=E(C(
4))!078
1330 IF D(3)*D(4)+(D(3)=0)*(
D(4)=0)THEN C(9)=E(C(9)):: C
(18)=F(C(18))!006
1340 IF D(4)*(D(2)+D(3))THEN
 C(6)=E(C(6))!250
1350 IF D(1)*D(4)+(D(1)=0)*(
D(4)=0)THEN C(1)=E(C(1)):: C
(10)=F(C(10))!226
1360 IF D(4)*(D(1)+D(2))THEN
 C(2)=E(C(2))!240
1370 IF D(2)*D(4)+(D(2)=0)*(
D(4)=0)THEN C(3)=E(C(3)):: C
(12)=F(C(12))!236
1380 IF D(4)=0 THEN C(17)=F(
C(17)):: C(13)=F(C(13)):: C(
14)=F(C(14))!046
1390 IF (D(4)=0)*((D(2)=0)+(
D(3)=0))THEN C(15)=F(C(15))!
201
1400 IF (D(4)=0)*((D(1)=0)+(
D(2)=0))THEN C(11)=F(C(11))!
191
1410 GOTO 1900 !194
1420 REM ROT 5 !004
1430 C(10)=E(C(10)):: C(1)=F
(C(1))!109
1440 IF D(1)=0 THEN C(11)=E(
C(11)):: C(13)=E(C(13)):: C(
14)=E(C(14))!028
1450 IF D(1)*D(2)+(D(1)=0)*(
D(2)=0)THEN C(12)=E(C(12))::
 C(3)=F(C(3))!230
1460 IF (D(1)=0)*((D(2)=0)+(
D(3)=0))THEN C(15)=E(C(15))!
197
1470 IF D(1)*D(4)+(D(1)=0)*(
D(4)=0)THEN C(16)=E(C(16))::
 C(7)=F(C(7))!250
1480 IF (D(1)=0)*((D(3)=0)+(
D(4)=0))THEN C(17)=E(C(17))!
203
1490 IF D(1)*D(3)+(D(1)=0)*(
D(3)=0)THEN C(18)=E(C(18))::
 C(9)=F(C(9))!000
1500 IF D(1)THEN C(2)=F(C(2)
):: C(4)=F(C(4)):: C(5)=F(C(
5))!066
1510 IF D(1)*(D(2)+D(3))THEN
 C(6)=F(C(6))!248
1520 IF D(1)*(D(3)+D(4))THEN
```

(See Page 10)

12OCLOCK—

(Continued from Page 9)

```
 C(8)=F(C(8))!254
1530 GOTO 1900 !194
1540 REM ROT 6 !005
1550 C(12)=E(C(12)):: C(3)=F
(C(3))!117
1560 IF D(2)=0 THEN C(11)=E(
C(11)):: C(14)=E(C(14)):: C(
15)=(C(15))!220
1570 IF D(1)*D(2)+(D(1)=0)*(
D(2)=0)THEN C(10)=E(C(10)):
 C(1)=F(C(1))!222
1580 IF (D(2)=0)*((D(1)=0)+(
D(4)=0))THEN C(13)=E(C(13))!
194
1590 IF D(2)*D(3)+(D(2)=0)*(
D(3)=0)THEN C(18)=E(C(18)):
 C(9)=F(C(9))!002
1600 IF (D(2)=0)*((D(3)=0)+(
D(4)=0))THEN C(17)=E(C(17))!
204
1610 IF D(2)*D(4)+(D(2)=0)*(
D(4)=0)THEN C(16)=E(C(16)):
 C(7)=F(C(7))!252
1620 IF D(2)THEN C(2)=F(C(2)
):: C(5)=F(C(5)):: C(6)=F(C(
6))!071
1630 IF D(2)*(D(1)+D(4))THEN
 C(4)=F(C(4))!245
1640 IF D(2)*(D(3)+D(4))THEN
 C(8)=F(C(8))!255
1650 GOTO 1900 !194
1660 REM ROT 7 !006
1670 C(18)=E(C(18)):: C(9)=F
(C(9))!141
1680 IF D(3)=0 THEN C(17)=E(
C(17)):: C(14)=E(C(14)):: C(
15)=E(C(15))!046
1690 IF D(3)*D(4)+(D(3)=0)*(
D(4)=0)THEN C(16)=E(C(16)):
 C(7)=F(C(7))!254
1700 IF (D(3)=0)*((D(1)=0)+(
D(4)=0))THEN C(13)=E(C(13))!
195
1710 IF D(2)*D(3)+(D(2)=0)*(
D(3)=0)THEN C(12)=E(C(12)):
 C(3)=F(C(3))!234
1720 IF (D(3)=0)*((D(1)=0)+(
D(2)=0))THEN C(11)=E(C(11))!
189
1730 IF D(1)*D(3)+(D(1)=0)*(
D(3)=0)THEN C(10)=E(C(10)):
 C(1)=F(C(1))!224
1740 IF D(3)THEN C(8)=F(C(8)
):: C(5)=F(C(5)):: C(6)=F(C(
6))!084
1750 IF D(3)*(D(1)+D(4))THEN
 C(4)=F(C(4))!246
1760 IF D(3)*(D(1)+D(2))THEN
 C(2)=F(C(2))!240
1770 GOTO 1900 !194
1780 REM ROT 8 !007
1790 C(16)=E(C(16)):: C(7)=F
(C(7))!133
1800 IF D(4)=0 THEN C(17)=E(
C(17)):: C(13)=E(C(13)):: C(
14)=E(C(14))!043
1810 IF D(3)*D(4)+(D(3)=0)*(
D(4)=0)THEN C(18)=E(C(18)):
 C(9)=F(C(9))!006
1820 IF (D(4)=0)*((D(2)=0)+(
D(3)=0))THEN C(15)=E(C(15))!
200
1830 IF D(1)*D(4)+(D(1)=0)*(
D(4)=0)THEN C(10)=E(C(10)):
 C(1)=F(C(1))!226
1840 IF (D(4)=0)*((D(1)=0)+(
D(2)=0))THEN C(11)=E(C(11))!
190
1850 IF D(2)*D(4)+(D(2)=0)*(
D(4)=0)THEN C(12)=E(C(12)):
 C(3)=F(C(3))!236
1860 IF D(4)THEN C(8)=F(C(8)
):: C(5)=F(C(5)):: C(4)=F(C(
4))!081
1870 IF D(4)*(D(2)+D(3))THEN
 C(6)=F(C(6))!251
1880 IF D(4)*(D(1)+D(2))THEN
 C(2)=F(C(2))!241
1890 GOTO 1900 !194
1900 REM SHOW ROT !016
1910 FOR I=1 TO 18 !113
1920 CALL LOCATE(#I,Y(I)+16*
(C(I)>2)*(C(I)<9),X(I)+16*(C
(I)>5)):: CALL PATTERN(#I,96
+4*C(I))!023
1930 NEXT I :: DISPLAY AT(12
,21):N !112
1940 GOTO 790 !104
1950 CALL CLEAR :: CALL CHAR
SET :: CALL DELSPRITE(ALL)!1
90
1960 DISPLAY AT(4,2):"YOU TO
OK";N;"MOVES." :: DISPLAY AT
(5,2):"IT CAN BE DONE IN 3 M
OVES" !181
1970 DISPLAY AT(7,2):"TRY AG
AIN (Y/N)?" !062
1980 CALL KEY(0,K,S):: IF S=
0 THEN 1980 !007
1990 IF (K=89)+(K=121)THEN 2
30 ELSE CALL CLEAR :: STOP !
150
2000 !@P+ !062
```

THE ART OF ASSEMBLY — PART 54

More Random Numbers

By BRUCE HARRISON

This will make the third column in this series that deals with random numbers. We keep trying new things all the time, though, and sharing what we learn with you, so bear with us; we think this will be worthwhile, especially for those who want to use random numbers in Extended BASIC. As you no doubt know, making random numbers in XB can take a lot of time. Provided that what you want is random integers in an array variable, the source in this month's sidebar will be a great help.

WITHOUT REPLACEMENT

What do we mean by "without replacement"? Just this: Each number will be included once and only once. Let's say you grab a deck of cards, shuffle them, and then pick a card randomly from the deck. Say that's the three of hearts. If you set that aside and pick another card from the deck, you can be certain it won't be the three of hearts. In the case of the first two routines in this month's sidebar, if you ask for 500 numbers, you can be sure that no number will be repeated. Two months ago, we showed a version of this derived from our Titler program. That one dealt only with single-byte numbers, so it wouldn't do as a general-purpose random number routine.

What we set out to do here was to offer a routine that would make up to 500 random numbers without replacement, and would assign them into an XB array variable. It had to do this quickly, and it does. An array of dimension 500 can be filled with random numbers in less than a second. Each number from 1 through 500 will be there, just once. In case you've any doubts, a second entry point in this first routine called CHECK will check each number against all the rest, just to insure that none are repeated. Checking takes longer than making, so you wouldn't want to leave the CHECK in a finished program, but it's there so you can test with it, to be confident in the results. If CHECK finds a duplicated number in the list, it will stop the XB program with a "BAD VALUE IN XXX" report and the "BOOP" sound, just as if the error were in the XB part of the program. (In hundreds of test runs this hasn't happened to us!)

Let's say you grab a deck of cards, shuffle them, and then pick a card randomly from the deck. Say that's the three of hearts. If you set that aside and pick another card from the deck, you can be certain it won't be the three of hearts.

THE CALL LINK

In this case, the CALL LINK may have two, three or four parameters, depending on what you want to do with the numbers, and on whether or not you're using OPTION BASE 1. The two minimum parameters are the number of numbers you want and the variable name for the array. Let's say, for example, that we want 52 numbers, we've DIMed a variable to 51, and that's called A(). To perform the assignment, we do this:

```
CALL LINK("RANDOM",52,A())
```

When that finishes (in very little time) the array members from A(0) through A(51) will contain the numbers 0 through 51 in random order. These could be used as a shuffled deck of cards, for example, with 0 through 3 being aces, 4 through 7 being deuces and so on. That could really speed up many card-playing games in XB. Using the 0th element in the array makes best use of memory, and that's why we only had to DIM A(51) instead of (52).

But suppose you have trouble dealing with the value 0, even if only as a "mental block" because people start counting with one, not zero. You might still want to use the 0th element, but want the numbers in the variable to run from 1 through 52, not 0 through 51. Can do! That's what the third parameter is for in the CALL LINK. You can do this:

```
CALL LINK("RANDOM",52,A(),1)
```

Sure enough, the numbers in your array will range from 1 through 52. If you like, you could put 1001 in that third parameter, and the numbers in A(0) through A(51) will be from 1001 through 1052. You say you're still not happy, and you want to use OPTION BASE 1 like you always do? Yes, Virginia, there is a Santa Claus. You can do this in your XB program:

```
10 OPTION BASE 1
20 DIM A(52)
30 CALL LINK("RANDOM",52,A(),1,1)
```

The fourth parameter is necessary if you've done that OPTION BASE 1 operation. The third parameter may be zero, so the numbers themselves would start at zero, but the fourth parameter has to be 1 for the OPTION BASE 1 case.

THE SECOND ROUTINE

In case you're running short of low memory for assembly routines, there's a shorter version which eliminates the CHECK capability and also eliminates a block of 500 words in memory. This routine, called RAND3, does exactly the same job as the bigger one in less memory, because it doesn't have the CHECK routine and doesn't need to keep the numbers available for checking. The CALL LINK for this works exactly like the previous case.

AND YET ANOTHER

It occurred to us that in some cases you just need a bunch of random numbers in a specific range, and don't care whether numbers get repeated. For that case, we've included RAND4. RAND4 can make any number of random numbers you might want, provided only that the array variable has been DIMed large enough to hold that many numbers. This routine needs more parameters, as the range must be specified by lower and upper limits. (In the "without replacement" cas-

(See Page 12)

THE ART OF ASSEMBLY —

(Continued from Page 11)

es, the range was inherently specified.) With RAND4, you're completely free. The CALL LINK might look like this:

CALL LINK("RANDOM",B,X,Y,A())

Where B is the number of integer random numbers to be created, X is the lower limit of the range, Y is the higher limit of the range, and A() is the array variable into which the numbers will go, starting with A(0) and running through A(B-1). The limits are these: B cannot be larger than the DIM of A() plus 1; X must be less than Y. Either X or Y or both can be negative numbers, as long as X is a smaller (or more negative) number than Y. No checking is available, but you can easily check the numbers in XB, run a few dozen times, then take out the checking part when you're convinced the routine always works as advertised. Like its companion pieces, this routine works fast. In our test program (see sidebar) we've made it assign 1,000 random numbers, and that happens in less than two seconds. That test also includes a checking process, which insures us that the limits given have not been exceeded and shows us where the limiting numbers themselves showed up in the sequence. Having run this test many, many times, we're confident that no number less than X nor more than Y gets generated. Of course now and then neither limit will show up, or else the limits may show up four or five times in any one run, but that's to be expected. The important thing is that the numbers generated are nicely scattered and never exceed the limits.

THE OUTER LIMITS

As for any case involving one-word integers, the limiting numbers on limits are -32768 and +32767. It's unlikely you'd use those, but the routine will still work if you do! For the previous cases, in which numbers are chosen without replacement, the limits on that third parameter are the same, except that the higher positive limit must be 32767 minus the first parameter, else you'll get screwy results, with the numbers "wrapping around" to negative ones.

THAT FIFTH PARAMETER

As with the other routines, this one may take an extra parameter. If OPTION BASE 1 is in effect, there must be a fifth parameter, and its value must be at least 1 This parameter has other uses, as is also true of the fourth parameter on the RAND2 and RAND3 routines. You can use this fifth (or fourth, as appropriate) parameter to partially fill an array. Let's take an example to show what this means. Suppose you have a case where the array is DIMed at 999, but you want to fill only 500 elements, starting at number 300, with numbers in the range of -250 to 250, inclusive. This can be done as follows:

CALL LINK("RANDOM",500,-250,250,A(),300)

After this executes, the A() array will have random numbers in it, starting at A(300) and running through A(799). Other members of the array will be unaffected. The same can be done with the earlier routines, but using the fourth parameter instead of the fifth. To be perfectly honest about this, we're not entirely sure why anybody would want to do this, but since the extra parameter was needed in the case of OPTION BASE 1, we decided to make it do extra duty by offsetting the base to which numbers get assigned.

THE DEMO PROGRAMS

Four demo programs are listed in the sidebar, and these should be available as programs, along with their respective object f on the disk version of MICROpendium. These are called CARDS, CARDS2, BIGRAND2 and BIGRAND4.

THE FINER POINTS

Before using any of these, be sure to have a RANDOMIZE somewhere in the beginning of your Extended BASIC program.

You'll see there's a "mystery" instruction near the beginning of each routine that says A @>8378,@>83C0. This was put in to insure against a phenomenon that can happen without it. We found in testing that sometimes when the number of numbers asked for was even, (52, for example) the random numbers would fall into a pattern in which the first number was always even. It would differ for each run, but always was even, and that meant that our "randomness" was lacking. Adding the number that happens to be at >8378 every time we start the routine avoids that problem, so that on repeated runs, the list will start with odd or even numbers, each about half of the time. The number in >8378 has a limited range, but it keeps changing while XB is running, so it
provides an extra measure of randomness when added to the seed number at >83C0.

For those who care about such things, here's how the memory use stacks up. RAND2/O takes up the most, as it must to include the "CHECK" feature, at 2286 bytes. That, of course, includes 2000 bytes for the two tables of 500 words each. Next in line is RAND3/O, which has no checking, and takes only 1212 bytes, including 1000 for its one table of 500 words. Smallest of the lot is RAND4/O, which doesn't need to keep tables for its numbers. This takes a mere 179 bytes! For most purposes, we'd use RAND3/O or RAND4/O, depending whether we needed the "without replacement" feature. Any of these can be "embedded" into your XB program itself using either Todd Kaplan's ALSAVE or Harry Wilhelm's HML program.

That's all for this month. The source code in the sidebar is well annotated, so you should be able to follow what it's doing without a lengthy explanation. If you don't understand, you can get first-hand help at (301) 277-3467, any time from 9 a.m. through midnight Eastern time. Bye for now.

SIDEBAR 54

```
* SIDEBAR 54
* FIRST, THREE ROUTINES FOR
* USE WITH XB PROGRAMS
*
* PART ONE - AN ELEGANT VERSION
*
* RAND2/S
* ASSIGNS RANDOM NUMBERS
* WITHOUT REPLACEMENT
* INTO AN XB ARRAY VARIABLE
* CAN ASSIGN UP TO 500 NUMBERS
*    CODE BY:  Bruce Harrison
*    PUBLIC DOMAIN
*    25 December 1994
*
```

(See Page 13)

THE ART OF ASSEMBLY —

(Continued from Page 12)

```
      DEF  RANDOM,CHECK DEFINE ENTRY POINTS
*
* REQUIRED EQUATES
*
GPLWS  EQU  >83E0        GPL WORKSPACE
FAC    EQU  >834A        F. P. ACCUMULATOR
NUMREF EQU  >200C        NUMERIC REFERENCE
NUMASG EQU  >2008        NUMERIC ASSIGNMENT
XMLLNK EQU  >2018        XML LINK VECTOR
CIF    EQU  >20          CONVERT INTEGER TO F.P.
CFI    EQU  >12B8        CONVERT F.P. TO INTEGER
ERR    EQU  >2034        ERROR REPORT
ARGNUM EQU  >8312        NUMBER OF ARGUMENTS
*
*
RANDOM LWPI WS           LOAD OUR WORKSPACE
       A    @>8378,@>83C0 ADJUST SEED NUMBER
       CLR  R0          CLEAR FOR NON-ARRAY
       CLR  R14         CLEAR START NUMBER
       CLR  R15         CLEAR BASE OPTION
       LI   R1,1        FIRST PARAMETER
       BLWP @NUMREF      GET PARAMETER
       BLWP @XMLLNK      USE XML LINK
       DATA CFI          CONVERT TO INTEGER
       MOV  @FAC,R13     STASH IN REG 13
       MOV  R13,R12      PUT IN R12 ALSO
       SLA  R12,1        DOUBLE R12
       CLR  R3          CLEAR COUNT
       CB   @ARGNUM,@THREE CHECK ARGUMENTS
       JLT  RAND2        LESS THAN 3, JUMP
       LI   R1,3         PARAMETER 3
       BLWP @NUMREF      GET THAT
       BLWP @XMLLNK      USE XML
       DATA CFI          CONVERT TO INTEGER
       MOV  @FAC,R14     PLACE AS START NUMBER
RAND1  CB   @ARGNUM,@FOUR CHECK FOR FOUR
       JLT  RAND2        IF LESS, SKIP
       LI   R1,4         PARAMETER 4
       BLWP @NUMREF      GET THAT
       BLWP @XMLLNK      USE XML
       DATA CFI          TO INTEGER
       MOV  @FAC,R15     STASH AS BASE NUMBER
RAND2  LI   R9,BUFF2      POINT AT BUFF2
       MOV  R9,R10       PLACE IN R10
       A    R12,R10      ADD DOUBLED NUMBER
BRAN0  MOV  R3,*R9+       PLACE A WORD IN TABLE
       INC  R3           INC COUNT
       C    R9,R10       COMPARE POINTER
       JLT  BRAN0        IF LESS, BACK
       MOV  R3,R6        PUT R3 INTO R6
       SLA  R6,1         DOUBLE THAT
       LI   R10,BUFFER   POINT AT BUFFER
BRAN1  LI   R4,28645      BIG NUMBER IN R4
       MPY  @>83C0,R4    MULT. BY SEED
       AI   R5,31417     ADD BIG NUMBER
       MOV  R5,@>83C0    PUT BACK AT SEED
       CLR  R4          CLEAR HIGH WORD
       DIV  R3,R4        DIVIDE BY R3
       SLA  R5,1         DOUBLE REMAINDER
       MOV  @BUFF2(R5),*R10+ MOVE TABLE TO TABLE
       MOV  @BUFF2-2(R6),@BUFF2(R5) REMOVE USED ONE
       DECT R6          SUBTR. 2 FROM R6
       DEC  R3          SUBTR 1 FROM R3
       JNE  BRAN1        IF NOT ZERO, REPEAT
       LI   R10,BUFFER   POINT BACK AT BUFFER
       CLR  R0          NUMBER 0
       A    R15,R0       ADD BASE OPTION
       LI   R1,2         SECOND PARAMETER
       MOV  R13,R4       NUMBER OF NUMBERS
ASGLP  MOV  *R10+,@FAC   ONE NUMBER TO FAC
       A    R14,@FAC     ADD START NUMBER
       BLWP @XMLLNK      USE XML
       DATA CIF          TO FLOATING POINT
       BLWP @NUMASG      ASSIGN TO PARAMETER
       INC  R0          NEXT ARRAY MEMBER
       DEC  R4          DEC COUNT
       JNE  ASGLP        NOT ZERO, REPEAT
EXIT   LWPI GPLWS        LOAD GPL WS
       B    @>6A         EXIT TO GPL INT.
CHECK  LWPI WS           LOAD OUR WORKSPACE
       CI   R13,3        COMPARE TO 3
       JLT  EXIT         IF LESS, EXIT
       LI   R9,BUFFER    POINT AT BUFFER
       MOV  R13,R4       NUMBER OF NUMBERS IN R4
       MOV  R13,R5       AND R5
       DECT R4           SUBTR 2 FROM R4
       DEC  R5           AND FROM R5
RECMP  MOV  R9,R10       PUT R9 IN R10
       INCT R10          ADD 2
CMP    C    *R9,*R10+    COMPARE
       JEQ  ERROR        IF EQUAL, ERROR
       DEC  R5           DEC COUNT
       JNE  CMP          IF NON-ZERO, REPEAT
       MOV  R4,R5        PUT R4 INTO R5
       INCT R9           ADD 2 TO R9
       DEC  R4           DEC COUNT 2
       JNE  RECMP        IF <>0, NEXT CYCLE
       JMP  EXIT         FINISHED, EXIT
ERROR  LI   R0,>1E00     "BAD VALUE" MESSAGE
       BLWP @ERR         REPORT ERROR
*
* DATA SECTION
*
WS     BSS  32           OUR WORKSPACE
BUFF2  BSS  1000         SELECTION LIST
BUFFER BSS  1000         OUTPUT LIST
THREE  BYTE 3            JUST 3
FOUR   BYTE 4            JUST 4
       END
```

PART TWO — A SIMPLER VERSION

```
* PART TWO - A SIMPLER VERSION
*
*
* RAND3/S
* "COMPACT" VERSION
* ASSIGNS RANDOM NUMBERS
```

(See Page 14)

THE ART OF ASSEMBLY —

(Continued from Page 13)

```
* WITHOUT REPLACEMENT
* INTO AN XB ARRAY VARIABLE
* CAN ASSIGN UP TO 500 NUMBERS
* NO "CHECK" FEATURE
*    CODE BY:  Bruce Harrison
*    PUBLIC DOMAIN
*    26 December 1994
*
      DEF  RANDOM DEFINE ENTRY POINT
*
* REQUIRED EQUATES
*
GPLWS  EQU  >83E0        GPL WORKSPACE
FAC    EQU  >834A        F. P. ACCUMULATOR
NUMREF EQU  >200C        NUMERIC REFERENCE
NUMASG EQU  >2008        NUMERIC ASSIGNMENT
XMLLNK EQU  >2018        XML LINK VECTOR
CIF    EQU  >20          CONVERT INTEGER TO F.P.
CFI    EQU  >12B8        CONVERT F.P. TO INTEGER
ARGNUM EQU  >8312        NUMBER OF ARGUMENTS
*
*
RANDOM LWPI WS           LOAD OUR WORKSPACE
       A    @>8378,@>83C0 ADJUST SEED NUMBER
       CLR  R0           CLEAR FOR NON-ARRAY
       CLR  R14          CLEAR START NUMBER
       CLR  R15          CLEAR BASE OPTION
       LI   R1,1         FIRST PARAMETER
       BLWP @NUMREF      GET PARAMETER
       BLWP @XMLLNK      USE XML LINK
       DATA CFI          CONVERT TO INTEGER
       MOV  @FAC,R12     STASH IN REG 12
       SLA  R12,1        DOUBLE R12
       CLR  R3           CLEAR COUNT
       CB   @ARGNUM,@THREE CHECK ARGUMENTS
       JLT  RAND2        LESS THAN 3, JUMP
       LI   R1,3         PARAMETER 3
       BLWP @NUMREF      GET THAT
       BLWP @XMLLNK      USE XML
       DATA CFI          CONVERT TO INTEGER
       MOV  @FAC,R14     R14 IS START NUMBER
RAND1  CB   @ARGNUM,@FOUR CHECK FOR FOUR
       JLT  RAND2        IF LESS, SKIP
       LI   R1,4         PARAMETER 4
       BLWP @NUMREF      GET THAT
       BLWP @XMLLNK      USE XML
       DATA CFI          TO INTEGER
       A    @FAC,R0      ADD BASE TO R0
RAND2  LI   R9,BUFF2     POINT AT BUFF2
       MOV  R9,R10       PLACE IN R10
       A    R12,R10      ADD DOUBLED NUMBER
BRAN0  MOV  R3,*R9+      PLACE R3 IN TABLE
       INC  R3           INC COUNT
       C    R9,R10       COMPARE POINTER
       JLT  BRAN0        IF LESS, BACK
       MOV  R3,R6        PUT R3 INTO R6
       SLA  R6,1         DOUBLE THAT
       LI   R1,2         PARAMETER 2
BRAN1  LI   R4,28645     BIG NUMBER IN R4
       MPY  @>83C0,R4    MULT. BY SEED
       AI   R5,31417     ADD BIG NUMBER
       MOV  R5,@>83C0    PUT BACK AT SEED
       CLR  R4           CLEAR HIGH WORD
       DIV  R3,R4        DIVIDE BY R3
       SLA  R5,1         DOUBLE REMAINDER
       MOV  @BUFF2(R5),@FAC MOVE WORD TO FAC
       A    R14,@FAC     ADD START NUMBER
       BLWP @XMLLNK      USE XML
       DATA CIF          TO FLOATING POINT
       BLWP @NUMASG      ASSIGN TO PARAMETER
       INC  R0           NEXT ARRAY MEMBER
       MOV  @BUFF2-2(R6),@BUFF2(R5) REMOVE USED ONE
       DECT R6           SUBTR. 2 FROM R6
       DEC  R3           SUBTR 1 FROM R3
       JNE  BRAN1        IF NOT ZERO, REPEAT
EXIT   LWPI GPLWS        LOAD GPL WS
       B    @>6A         EXIT TO GPL INT.
*
* DATA SECTION
*
WS     BSS  32           OUR WORKSPACE
BUFF2  BSS  1000         SELECTION LIST
THREE  BYTE 3            JUST 3
FOUR   BYTE 4            JUST 4
       END
```

PART THREE — A "WITH REPLACEMENT"

```
* PART THREE - A "WITH REPLACEMENT"
* RANDOM NUMBER GENERATOR
*
* RAND4/S
* ASSIGNS RANDOM NUMBERS
* WITH REPLACEMENT
* INTO AN XB ARRAY VARIABLE
* CAN ASSIGN ANY NUMBER OF ELEMENTS
* NO "CHECK" FEATURE
*    CODE BY:  Bruce Harrison
*    PUBLIC DOMAIN
*    26 December 1994
*
      DEF  RANDOM DEFINE ENTRY POINT
*
* REQUIRED EQUATES
*
GPLWS  EQU  >83E0        GPL WORKSPACE
FAC    EQU  >834A        F. P. ACCUMULATOR
NUMREF EQU  >200C        NUMERIC REFERENCE
NUMASG EQU  >2008        NUMERIC ASSIGNMENT
XMLLNK EQU  >2018        XML LINK VECTOR
CIF    EQU  >20          CONVERT INTEGER TO F.P.
CFI    EQU  >12B8        CONVERT F.P. TO INTEGER
ARGNUM EQU  >8312        NUMBER OF ARGUMENTS
*
*
RANDOM LWPI WS           LOAD OUR WORKSPACE
       A    @>8378,@>83C0 ADJUST SEED NUMBER
```

(See Page 15)

THE ART OF ASSEMBLY —

(Continued from Page 14)

```
      CLR  R0          CLEAR FOR NON-ARRAY
      LI   R1,1        FIRST PARAMETER
      BLWP @NUMREF     GET PARAMETER
      BLWP @XMLLNK     USE XML LINK
      DATA CFI         CONVERT TO INTEGER
      MOV  @FAC,R3     NUMBER OF NUMBERS IN R3
      INC  R1          2ND PARAMETER
      BLWP @NUMREF     GET PARAMETER
      BLWP @XMLLNK     USE XML LINK
      DATA CFI         CONVERT TO INTEGER
      MOV  @FAC,R15    LOWER LIMIT IN R15
      INC  R1          3RD PARAMETER
      BLWP @NUMREF     GET PARAMETER
      BLWP @XMLLNK     USE XML LINK
      DATA CFI         CONVERT TO INTEGER
      MOV  @FAC,R13    UPPER LIMIT IN R13
      INC  R13         INCREMENT FOR DIVIDES
      S    R15,R13     ADJUST BY LOWER LIMIT
      CB   @ARGNUM,@FIVE CHECK FOR 5 ARGUMENTS
      JLT  RAND2       LESS THAN 5, JUMP
      INCT R1          FIFTH PARAMETER
      BLWP @NUMREF     GET NUMBER
      BLWP @XMLLNK     USE XML
      DATA CFI         CONVERT TO INTEGER
      A    @FAC,R0     SET BASE NUMBER
RAND2 LI   R1,4        PARAMETER 2
BRAN1 LI   R4,28645    BIG NUMBER IN R4
      MPY  @>83C0,R4   MULT. BY SEED
      AI   R5,31417    ADD BIG NUMBER
      MOV  R5,@>83C0   PUT BACK AT SEED
      CLR  R4          CLEAR HIGH WORD
      DIV  R13,R4      DIVIDE BY R13
      MOV  R5,@FAC     MOVE R5 TO FAC
      A    R15,@FAC    ADD START NUMBER
      BLWP @XMLLNK     USE XML
      DATA CIF         TO FLOATING POINT
      BLWP @NUMASG     ASSIGN TO PARAMETER
      INC  R0          NEXT ARRAY MEMBER
      DEC  R3          SUBTR. 1 FROM R3
      JNE  BRAN1       IF NOT ZERO, REPEAT
EXIT  LWPI GPLWS       LOAD GPL WS
      B    @>6A        EXIT TO GPL INT.
*
* DATA SECTION
*
WS    BSS  32          OUR WORKSPACE
FIVE  BYTE 5           JUST 5
      END
```

PART FOUR

An Extended BASIC program to demonstrate RAND2/O, including the "CHECK" feature program called "CARDS."

```
10 CALL INIT :: CALL LOAD("DSK1.RAND2/O")
20 OPTION BASE 1
30 RANDOMIZE :: CALL CLEAR :: DIM A(52)
40 PRINT "STARTING ASSIGNMENT" :: CALL LINK(
"RANDOM",52,A(),1,1):: PRINT "ASSIGNMENT DO
NE"
50 PRINT "CHECKING" :: CALL LINK("CHECK")::
PRINT "LIST OKAY":"HERE'S THE LIST"
60 FOR I=1 TO 52 :: PRINT A(I);:: NEXT I
```

PART FIVE

An Extended BASIC program to demonstrate the RAND3/O program called "CARDS2."

```
10 CALL INIT :: CALL LOAD("DSK1.RAND3/O")
20 OPTION BASE 1
30 RANDOMIZE :: CALL CLEAR :: DIM A(52)
40 PRINT "STARTING ASSIGNMENT" :: CALL
LINK("RANDOM",52,A(),1,1):: PRINT "ASSIGNME
NT DONE"
50 PRINT "HERE'S THE LIST" :: FOR I=1 TO 52
:: PRINT A(I);:: NEXT I
```

PART SIX

An Extended BASIC program that gets 500 random numbers without replacement. Program called "BIGRAND2."

```
10 CALL INIT :: CALL LOAD("DSK1.RAND3/O")
20 RANDOMIZE :: CALL CLEAR :: DIM A(499)
30 PRINT "STARTING ASSIGNMENT" :: CALL LINK(
"RANDOM",500,A()):: PRINT "ASSIGNMENT DONE"
40 PRINT "HERE'S THE LIST" :: FOR I=0 TO 499
:: PRINT A(I);:: NEXT I
```

PART SEVEN

An Extended BASIC program that gets 1000 "with replacement" random numbers using RAND4/O. Program called "BIGRAND4."

```
10 CALL INIT :: CALL LOAD("DSK1.RAND4/O")
20 RANDOMIZE :: CALL CLEAR :: DIM A(999)
30 N=1000 :: X=-250 :: Y=250 :: PRINT "START
ING ASSIGNMENT"
40 CALL LINK("RANDOM",N,X,Y,A()):: PRINT "AS
SIGNMENT DONE"
50 PRINT "HERE'S THE LIST" :: FOR I=0 TO N-1
:: PRINT A(I);:: NEXT I :: PRINT
60 PRINT "CHECKING LIMITS"
70 FOR I=0 TO N-1 :: DISPLAY AT(24,1):I
80 IF A(I)=X THEN 90 ELSE IF A(I)=Y THEN 100
ELSE IF A(I)<X OR A(I)>Y THEN 110 ELSE 120
90 PRINT X;"FOUND AT";I :: GOTO 120
100 PRINT Y;"FOUND AT";I ::GOTO 120
110 PRINT "ERROR";A(I);"AT";I
120 NEXT I
```

Disk drives

Step motors and the genius of double-density

By JIM NESS

This article first appeared in VAST News.—Ed.

The great things about disk drives is that they can find files buried randomly within a huge field of data, and they do it pretty fast. Actually, they can do it so fast because it's not all random.

The mechanical concept is not all that complicated. A small motor spins at 300 RPM (at least in this country, with its 60 hz power supply), and there is a tiny stepping motor attached to a read/write head. A stepping motor is a common item in indexing applications, where you want a motor to move a precise distance and stop on a dime. The read/write head is just a smaller version of what you have in a cassette recorder.

The stepping motor "steps" the head from track to track on a diskette. The tracks are concentric circles, not a long spiral as you would have on a record album.

All of this is ultimately controlled by the disk software provided with your computer. Usually this is located in ROM within the machine. In most machines, the ROM is only sophisticated enough to load in the official Disk Operating System (DOS), which is located on the disk in the drive when the machine is turned on. The DOS contains all the file handling software, copying software, etc., and, because it is on disk, it can be easily modified or updated as time goes by.

Our friends at TI decided to put the whole thing in ROM, which has a few bad side effects. First, it makes it hard to update and improve the software that is located in the disk controller card. Second, although the TI is a 64K machine, TI set aside so much memory for special purposes that there is only 32K left to play with. They set aside 8K for cartridges, 4K for disk drive, 4K for RS232/PIO cards, 4K for the operating system (can't complain about that one), and 8K for various interfaces (speech, sound, VDP). Okay, those

All of the controlling software for the TI99/4A is located in the ROM card

are all good applications to have, but if you don't use them, you still can't use that memory for other things.

All of the controlling software for the TI99/4A is located in the ROM card, as I said. This software tells the step motor when to step to the next track, when to return to the beginning, etc.

There is no standard for how a computer keeps track of data. In the case of the TI, there is a directory of existing files, and a map of where they are located, at the beginning of each disk. These files are not necessarily all in complete groups. If you delete a 12-sector file from a disk, there is a 12-sector gap recorded in the map. Then, if you add a 20-sector file, the software will put the first 12 sectors in the gap, and put the rest in the first available spot. When you ask for a file that is broken up this way, you can hear the disk head scooting along to read each individual segment.

Because the disk drives themselves are pretty standard, there are a few things that do not change. For instance, there are 48 tracks per inch in most 5 1/4" systems. And most systems use only 35 or 40 of the available 48 tracks. There are either 9 or 18 sectors per track (single- or double-density). Each sector holds 256 bytes of data. And the standard design allows 250,000 bits per second to he written.

Wow, you say, 250K! That is about 25K bytes per second, right? How come I cannot load a 25K program in one second, then?

Two reasons, first, as I said, the transfer of data is controlled by ROM software in the TI. And to be as good as it is, it had to be a little bit slow. Not REAL slow, but not as fast as it could be. The second reason also has to do with software, but it is a universal problem associated with single-density storage.

The major difference between single- and double-density storage is the way in which data is coded. In order for the software to keep track of where the read head is located on a particular track, there are clock or sync bits laid down with the data bits. In the old-fashioned single-density format, a sync bit was laid down ahead of each "0" bit, so there were never two "0" bits in a row. That kept the software from getting lost if there were a lot of "0" bits in the series. Putting all those sync bits on the disk took up a tremendous amount of space that should be used for data.

So, some genius came up with a way of encoding the clock bits in with the data bits so that no unnecessary space was lost. Voila, double-density storage was born! And double-density, as used with the CorComp software, is said to increase transfer speed by at least 80 percent, mostly because the number of bits to transfer is cut away down.

So much for the exciting story of double-density versus single-density. How about double-sided versus single-sided? Well, obviously, it requires two read/write heads in the drive. Did you know that when reading a disk, the software reads first a track from side one, then the opposing track from side two and continues back and forth?

The disk head needs something to keep the disk stationary against it. In a single-sided drive, there is a small arm holding the back side of the disk against the head. In a double-sided drive, that arm would be in the way of the back side read/write head, so the solution was to use the two heads directly across from one another to hold the disk in place. In order to keep them across from one another, they alternate reading or writing.

Printer codes and Funnelweb editor V5.0

All-CHAR mode gives users control without the formatter

By JACQUES GROSLOUIS

The following article appeared in the Bits, Bytes and Pixels newsletter, among others.—Ed.

The use of the all-CHAR mode of Funnelweb editor V5.0 lets users print text without using the formatter. I use an Epson FX-80 printer and will describe these codes. (Readers should check their printer manuals for appropriate codes.—Ed.)

The first line of my document contains the following code:

1. <ESC>"E" — This causes the printer to print in emphasized mode and is entered with the following key presses: CTRL-U, FCTN-R, CTRL-U, Shift-E. If you have converted characters 0 to 31 to reverse video, then CTRL-U, FCTN-R, CTRL-U (which represents <ESC>) will appear on the screen as a reverse bracket ([).

2. <ESC>"1" — This changes line spacing to 7/72 inch and is entered by keying in CTRL-U, FCTN-R, CTRL-U and 1.

3. <ESC>"U1" — This turns continuous unidirectional mode on. No spaces should appear in the printer code. This code will also work correctly on my printer if the number "1" is entered as CTRL-U, Shift-A, CTRL-U. This is used to keep the sides of the border straight.

4. <ESC>"C" <0><11> — This sets the page length to 11 inches and is keyed as CTRL-U, FCTN-R, CTRL-U, Shift-C, CTRL-U, Shift-2, Shift-K, CTRL-U. Shift-2 produces a reverse "at" (@) character and represents character "0," while Shift-K represents "11," being the 11th letter of the alphabet.

5. <ESC>"N"<8> — This produces an eight-line skip over the end of page perforation. The <8> is entered as CTRL-U, Shift-H, CTRL-U because "H" is the eight letter of the alphabet.

The border around the title of my document was created using the all-CHAR mode and uses IBM fonts stored in characters 128 to 254. My FX-80 printer does not come with this feature but permits the downloading of these characters. Additional code is required in line nine of the document as follows:

1. <ESC>"2" — This restores line spacing to 1/6 inch (default).
2. <ESC>"U0" — This turns unidirectional mode off.

My favorite set of printer codes sets the printer to indent eight spaces, using 1/8 line spacing, sets form length to 88 lines, and skips over the perforation by 15 lines. This is very useful when doing program and source code listings, and for printing documents from DM-1000 or any other program that permits D/V80 printing. The code to enter these features in DM-1000 is as follows:

```
27  108  8  27  77  27 48  27  67  88  27  78
15  13  *
```

For some reason, the above will not work with V6.1 of DM-1000. I had to use the following code for V6.1:

```
27  68  08  00  27  77  27  48
27  67  88  27 78  15  13  *
```

The same coding, used as a merge file, will permit you to list an Extended BASIC program in elite mode, indented 8 characters with skip over perforations and with printing at eight lines per inch. This short program also allows you to date your listing, and include your initials. The program that I use is as follows:

```
2 OPEN #2:"PIO" ::
C$=CHR$(27)::
 PRINT
#2:C$&"M"&C$&"0"&C$&"C"&C
HR$(0)&CHR$(11)&C$&"N"&CHR$(8)&C
$&"1"&CHR$(10):: INPUT "DATE? MM
/DD/YY>:D$ :: PRINT #2:TAB(65);
"JJG " &D$ :: CLOSE #2 :: STOP
```

You can use this program provided line 2 and arrays using C$ and D$ are not used in the program you want to list to the printer. Merge this program into any program you wish to list. A prompt to enter a date will appear after you run it. A prompt to enter a date will appear. Press Enter and then delete line 2 by pressing 2 and then pressing Enter. In command mode, enter LIST "PIO". The full program will then print in elite print suitable for filing in a binder.

From Funnelweb, you can also set up the printer as described above, or in any other mode you desire, by creating text files which can be printed from the DiskReview view function, or by using the Print File function of the editor. You can create one file to set up the printer and another to cancel the setup in the first file. To create a file that does the same as the merge file described above, proceed as follows:

One line 1, enter the following: CTRL-U, FCTN-R, CTRL-U, Shift-M, CTRL-U, FCTN-R, CTRL-U, O, CTRL-U, FCTN-R, CTRL-U, Shift-C, CTRL-U, Shift-2, Shift-K, CTRL-U, CTRL-U, FCTN-R, CTRL-U, Shift-N, CTRL-U, Shift-H, CTRL-U, CTRL-U, FCTN-R, CTRL-U, lowercase 1, CTRL-U, Shift-J, CTRL-U.

This coding is long to describe but is short on the screen.

Save this file using a descriptive name, such as XPRLIST.

A file which would cancel the mode set up above should be prepared and saved as, say, UPRLIST. In order to run either file print from editor mode by using PF or from view mode of DiskReview by first pressing "V" to view and then CTRL-P to print.

(See Page 18)

From Funnelweb, you can also set up the printer as described, or in any other mode you desire, by creating text files which can be printed from the DiskReview view function....

FUNNELWEB—

(Continued from Page 17)

A help screen, which I have added to my help files in Funnelweb editor, is printed out below:

Control codes for Epson FX-80	On	Off
Elite mode	#M	#P
Compressed mode	<O>	#
One-line expanded	<N>	<T>
Continuous expanded	#W1	#W0
Emphasized mode	#E	#F
Double-strike mode	#G	#H
Superscript mode	#S0	#T
Subscript mode	#S1	#T
Proportional mode	#p1	#p0
Underline mode	#-1	#-0
Italic character set	#4	#5
Line spacing 1/8	#0	
Line spacing 7/72	#1	
Line spacing 1/6	#2	
Line spacing n/72	#ACHR$(n)	
Line spacing n/216	#3CHR$(n)	
Print characters 128-159	#6	#7
Unidirectional mode	#U1	#U0
Backspace	<H>	
ESC CTRL-U FCTN-R CTRL-U	#	
Select ROM character	#%<@@>	#%<A@>

Since escape cannot be printed, "#" has been used to indicate where CTRL-U, FCTN-R, CTRL-U should be entered. Other places where CTRL-U should be entered are indicated by "<" and ">"

Ms. Pac-Man challenge

Modifications make program compatible with Geneve

By MIKE MAKSIMIK
Crystal Software

The following article is reprinted from Hocus, *the newsletter of the Milwaukee Area 99/4 User Group.*

The challenge is there to make Ms. Pac-Man compatible with the Geneve....I DID IT! If you have the cartridge-saved version of AtariSoft's Ms. Pac-Man, do this change to disable keyboard scan, make the joystick more responsive, and to totally fix it for use with the Geneve.

I wanted to have that game available. My brother Chris said, "Why can't you change Ms. Pac-Man to run on the Geneve, too?" I asked myself the same question, so I used DiskAssembler for MDOS to disassemble the game.

The combined keyboard/joystick scan routine looked like Atari programmers had a nice way to scan for both the keyboard (on the 99/4A) and joystick with the speed of CRU.

I found the combined keyboard/joystick scan routine. It looked like Atari programmers had a nice way to scan for both the keyboard (on the 99/4A) and joystick with the speed of CRU. They also called the standard console SCAN routine (present in the Geneve) to scan for the space bar.

The trick was to "fool" the routine into not scanning the keyboard addresses, which are used for other functions on the Geneve. Also, I noticed that the "up" movement of the joystick continues to move, and can sometimes defeat the "down" movement. This is because the 9901 chip prioritizes its scan, and the Atari programmers tried to overcome this by giving the "boost" to the "UP" movement. The Geneve treats all movements with equal priority, so I reversed the "UP" and "DOWN" scans so that "DOWN" takes priority over "UP."

I know this is not a hardware problem. I had suspected it on the joystick port, but the 9901 is simply playing by its own rules. One or two players can still play, but remember, the arrow keys on the keyboard are not scanned. This will make the game available to Geneve users, and TI users with a P-GRAM or similar device will have a more accurate and more responsive game.

Using Disk Utilities, or another sector editor, search through the Ms. Pac-Man files for the following HEX string:

>06 08 01 10 03 08

Change it to:

>06 08 06 08 00 00

Search for (locate a few words away from the above string in the same sector):

>06 OA 02 10 02 08

Change it to:

>06 OA 06 0A 00 00

Further down, look for:

>06 OE 02 12 02 1A 02 83 00 00 16 03 02 01 00 03

Change this to:

>06 OC 06 OC 00 00 02 83 00 00 16 03 02 01 00 01

Finally, locate:

>06 0C 01 14 OB 06 02 83 00 00 16 02 02 01 00 01

Change it to:

>06 OE 06 OE 00 00 02 83 00 00 16 02 02 01 00 03

This will disable the arrow keys and make the joystick respond better. It does not disable the space bar, so you can still pause the game.

The 1980s Home Computer Era — Part 7

Timeline tells TI99 story

By BILL GASKILL

The year 1995 marks the 12th anniversary of the decision by Texas Instruments to abandon the Home Computer. I have compiled the information in this timeline not in celebration of TI's decision to orphan the 99/4A, but rather to honor the community that remains 12 years after TI's decision. Unlike other installments in this series, this 99/4A info is presented in timeline fashion because I have spent years scratching for every detail I could find pertain to the 4A, and thus have amassed quite a bit of data. I will begin the TI99 story in 1979, though the planning for its introduction began earlier than that.

June 1979: Texas Instruments announces the TI99/4 Home Computer at the Consumer Electronics Show in June at a retail price of $1,150 with a 13-inch color monitor.

January 1979: Double-sided disk drives are announced, but few are available, as manufacturers run into difficulty gearing up for production.

February 1979: Rumors begin to fly about TI's new personal computer, despite the fact that it has not been formally announced. The rumors say the computer will have 40K of ROM, it will generate 20 lines of 40 characters on a standard television, have provisions for accommodating video disk players and videotape recorders and it will have support for sophisticated sound production.

— Atari enters the personal computer market in February by announcing the 400 and 800 models. The 400 is a non-expandable 8K computer with a membrane keyboard, a single cartridge slot and a cassette port. It will sell for $500. The 800 is an 8K computer expandable to 48K. It comes with a cassette recorder, a full keyboard, 8K BASIC built in and high resolution graphics capabilities. It will sell for $1,000. Neither machine appears until August, and then only in limited quantities.

March 1979: The Federal Communications Commission begins regulating microcomputers that employ radio frequency modulators. Their action is spurred by the rash of previous complaints received when Citizen Band radios created havoc for TV viewers.

— Texas Instruments releases the new Speak and Spell learning aid for children. It is based on the TMS 1000 chip and two 128K DRAM chips, each with the ability to store more than 100 seconds of speech.

April 1979: McGraw-Hill buys *Byte* and *onComputing* magazines.

— Tandy begins selling its TRS line of personal computers through its own stores. Several other makers of personal computers withdraw their products from department store shelves after meeting with poor sales and low product acceptance.

— Despite early failure by U.S. department stores to move personal computers, department stores in Europe begin to surface as the major source of sales for Commodore's PET and Radio Shack's TRS-80.

— The CompuServe on-line information service is founded.

May 1979: Dan Bricklin and Bob Frankston release their new Visicalc spreadsheet, written for the Apple II computer.

June 1979: Texas Instruments announces the TI99/4 Home Computer at the Consumer Electronics Show in June at a retail price of $1,150 with a 13-inch color monitor. It will not appear in any quantity until almost a year later, however, and then it will prove to be a flop in the marketplace.

Software titles announced as being available for the new Home Computer include: Beginning Grammar, Demonstration, Diagnostic, Early Learning Fun, Early Reading, Football, Home Financial Decisions, Household Budget Management, Investment Analysis, Number Magic, Personal Record Keeping, Physical Fitness, Speech Construction, Tax/Investment Record Keeping, Video Chess, and Video Graphs.

Peripherals announced as being available are a Speech Synthesizer, an RS232 interface, joysticks, disk storage and a thermal printer. No memory expansion is available. The price for the console/monitor bundle is $1,150 with the Solid State Software command modules listed running anywhere from $19.95 to $69.95 in price.

Actual release dates on several of the announced products would vary from the June 1979 release information.

Beginning Grammar	2q/1979
Demonstration	2q/1979
Diagnostic	2q/1979
Disk Storage	2q/1980
Early Learning Fun	2q/1979
Early Reading	4q/1979
Football	2q/1979
Home Financial Decisions	2q/1980
Household Budget Management	2q/1979
Investment Analysis	never released under this name
Joysticks	2q/1980
Number Magic	2q/1979
Personal Record Keeping	4q/1979
Physical Fitness	2q/1979
RS232 interface	2q/1980
Speech Construction	never released under this

(See Page 20)

HOME COMPUTER ERA —

(Continued from Page 19)

	name
Speech Synthesizer	2q/1980
Tax/Investment Record Keeping	4q/1979
Thermal Printer	2q/1980
Video Chess	2q/1979
Video Graphs	2q/1979

— MicroPro releases WordStar.

— Color monitors for personal computers are expected to drop below the $1,000 mark by late 1979.

July 1979: Milton Bradley Company begins advertising in national trade publications for electronic product engineers, software engineers and microcomputer programmers and electronic technicians.

—Wayne Ratliff develops the Vulcan Data Base at the Jet Propulsion Labs in Pasadena, California. Ashton-Tate later picks up the program and markets it as dBase II.

— Word of a Japanese invasion into the personal computer market hits the media, much like the never-to-appear MSX invasion of the mid-'80s, after Nippon Electric Corporation (NEC) enters the market with its Astra series of 16-bit systems.

August 1979: TI releases a $250 hand-held language translator that features speech, which means translated words are not only displayed, but are also spoken. The unit will have $50 plug-in modules available for English, Spanish, French, German, Russian, Chinese and Japanese. Each module displays 1,000 words in the resident language, 500 of which can be spoken by the Speech Synthesizer.

September 1979: New England Electronics proudly announces that it has been selected to be an authorized distributor of the "revolutionary TI99/4 Personal/Educational Computer!"

— Computerland begins advertising the 99/4 also, calling it "The Remarkable Home Computer." They also carry the Atari 800 and refer to it as the "Timeless" computer.

Several other major distributors are also lined up by TI in the closing months of 1979. They begin advertising the 99/4, but fail to receive it and are forced to placate the few people who are willing to pay $1,150 for the machine. TI has already gotten off on the wrong foot with its retailers.

October 1979: Texas Instruments releases the TMS9927 Video Controller chip.

— Rodnay Zaks, who would write the book *Your First TI99/4A Program* in 1983, releases *6,502 Games* through Sybex Publishing.

November 1979: Moore Business Systems agrees to market the TI99/7, a $5,000 business computer based upon the TMS 9900 microprocessor. The 99/7 is one of three computers to be built on the TMS 9900 chip, but it will eventually die, due to internal squabbling at TI, without any production units being shipped.

—Zenith buys the Heath Company, manufacturers of the H-8 and H-11 computer kits.

—*Computer Shopper* publishes its first issue. A special charter subscription of 12 issues for $10 is offered.

— Milton Bradley releases Big Trak, a programmable toy vehicle. The chip in Big Trak allows the user to program intricate travel paths and fire the truck's weaponry in single burst, short burst, long burst modes. It sells for $43 with trailers that may be purchased separately for $13 each.

— Milton Bradley also releases its Microvision hand-held mini video game machine, which has its own screen. Microvision comes w0ith the game BlockBuster. Six other games, Bowling, Star Trek, Phaser Strike, Connect Four, Vegas Slots and Mindbuster are also available, sold separately.

December 1979: Len Buckwalter reviews the new TI99/4 Home Computer for *Mechanics Illustrated* magazine on page 46. He calls the machine easy to use and delivers a generally positive review, discussing Home Financial Decisions and Milton Bradley's Connect Four, Hangman, Yahtzee and Zero Zap cartridges.

— Image Computer Products of Northbrook, Illinois, announces the TI Six Pack, which consists of six BASIC games on cassette. They are Mind Master, Skill Builder, Strategy Pack, Tournament Brickbat, Wall Street Challenge and Wildcatting.

January 1980: The FCC grants TI a waiver on its RF modulator, which failed to pass FCC muster in early 1979 because Texas Instruments did not send the entire unit in to be tested. The waiver allows the TI99/4 Home Computer to be sold without the expensive monitor that TI was forced to bundle with it because of the RF modulator troubles. In response to the FCC decision TI conducts an extensive advertising blitz to put the 99/4 in the consumer eye. Unfortunately, production problems continue to haunt the Home Computer for the first few months of 1980 and TI is selling fewer than 1,000 units per month.

March 1980: TI has turned out fewer than 30 pieces of software for the 99/4. Although their policy of locking out third-party developers has not yet been articulated, the lack of an Editor/Assembler package for the Home Computer pretty much says it all. This is Mistake No. 3 according to Joseph Nocera. The only way to write programs for the TI99/4 is to buy one of TI's $50,000 minicomputers, which is what TI's in-house programmers use for cartridge program creation. In fact, even after the release of an Editor/Assembler package, programmers at TI continued to use minicomputers. During the entire life of the TI99/4 and 4A, Hopper was the only cartridge program ever developed internally using the TI99 Home Computer.

The existence of the secretive GPL (Graphics Programming Language) has not yet become common knowledge among the computer community, but it will eventually surface and add another nail to the coffin of an already sick reputation the TI99/4 is earning.

— *Creative Computing* magazine calls the TI99/4 "One of the most easy to use systems we've tested...price still beyond the grasp of Middle America."

April 1980: TI releases Disk Drive Controller PHP 1800 and Disk Memory Drive PHP 1850 for $299.95 and $499.95, respectively. This release comes 10 months after the computer was announced! Would you, or did you pay more than $1,000 for a computer that lacked any storage capability? Before you answer, don't forget that there was no cassette recorder offered for the 99/4 yet

(See Page 21)

HOME COMPUTER ERA —

(Continued from Page 20)

either. In fact, TI didn't offer its own cassette recorder until the first quarter of 1983!

May 1980: Bill Hawkins reviews the TI99/4 in *Popular Science* magazine on page 10, along with the Atari 800 and a virtually unknown cartridge-using computer named the APF Imagination Machine from APF of 444 Madison Avenue New York, New York. Although Hawkins is critical of the $1,150 price of the 99/4 (TI has still not dropped the price because there are so many console/monitor bundles in dealer inventories), he raves about the computer's Speech Synthesizer peripheral, and mentions the little-known PHA 2500 Speech Modules designed to increase the vocabulary of the early Speech Synthesizers.

— TI hires William J. Turner away from Digital Equipment and charges him with creating a marketing plan that will move the TI99/4 Home Computer off the shelves of retailers and into the homes of consumers.

June 1980: Radio Shack Executive Vice President John Roach brings down the house at a brokerage house seminar when from the podium he points to a demo model of the 99/4 and quips, "I'm sure glad somebody brought a TI computer here today. It's only the second one I've seen — and the first was when they launched it at the Consumer Electronics Show last June."

— New England Electronics, one of the top 10 personal computer distributors in the U.S., stops carrying the 99/4 after only nine months as a distributor. It tries to help its dealers by shifting 99/4s from the many stores where they are gathering dust to the few stores that have found a way to move the Home Computer.

— The June 16, 1980, issue of *Fortune* magazine says, "...TI has managed to exclude itself from the business and professional market (with the 99/4). It designed the system in a way that makes it difficult (more like impossible) for small, independent companies to write programs or make peripheral gear that can be used on the 99/4."

An unnamed TI engineer states that some of TI's top managers saw the 99/4 as their "private electronic fantasy" in explaining how the 99/4 survived when two other TI projects, the 99/7 business computer and the professional/scientific computer were both canceled. It is reported that the professional/scientific computer died for lack of funds when the Consumer Products Group was allowed to bleed off funds from the professional model's budget in order to speed up the introduction of the 99/4, which had fallen behind schedule.

The same article reports that distributors were so impressed with the TI99/7 business computer that some agreed to take on stocks of the 99/4 just to get at the TI99/7. Despite this, internal competition and squabbling killed the 99/7 when Shepherd and Bucy decided TI could not afford to fund the 99/7, the professional/scientific computer and the 99/4 Home Computer. As a former TI employee put it, "They threw away two pieces of gold and kept the lump of coal" in opting to continue with the development of the 99/4.

— Commodore Business Machines introduces the VIC-20 at a retail price of $299. The VIC-20 is a dismal machine that is easily outperformed by the TI99, but marketing mistakes keep TI from ever taking advantage of the Home Computer's technological advantages. The VIC-20 will become the arch-enemy of the TI99 in the price wars that will begin in the fall of 1982. Getting into a price war with Commodore is Mistake No. 4, according to Joseph Nocera.

July 1980: Texas Instruments hires 150 college students for the summer to convert popular software written for other computers to the 99/4 format.

— The Houston, Texas, TI Users Group (TI-HUG) is formed by Raymond Wells, becoming the first known TI99/4 Users Group.

August 1980: The Author Incentive program is instituted by TI in an effort to get more application programs written for the 99/4. TI also begins publishing the *Home Computer Users* newsletter.

— On Aug. 12, Charles LaFara sends a letter to all known 99/4 users and dealers announcing his intention to start the 99/4 Home Computer Users Group.

September 1980: LaFara incorporates the 99/4 Home Computer Users Group in Oklahoma City, Oklahoma. In its four and one-half year life the organization will move to larger headquarters in Bethany, Oklahoma, it will produce a respected newsletter and a professional 99/4-specific magazine, it will grow to a claimed membership of more than 100,000 users and then ultimately it will file for bankruptcy and breathe its last gasp of life in April 1985.

— The Chicago TI Users Group is formed by Jerry Strauss.

October 1980: Texas Instruments replaces 26 advertising agencies with one agency when it hires New York's McCann-Erickson Inc.

— In a letter to dealers in the United Kingdom, TI announces that an extended GROM is under development for the Home Computer, but that memory expansion for the 99/4 is not planned. The letter points out that no interaction between BASIC and the console GROMs is possible, though that proves not to be true for the Personal Record Keeping and Statistics cartridges. Dutch 99er Paul Karis will uncover this secret almost a year later when he publishes an article on the hidden secrets of the PRK module in the *TIHome TIdings* newsletter.

November 1980: TI finally drops the price of the 99/4 Nov. 28 a full $300, to $650, a move that was first promised for the summer of 1980.

December 1980: *SourceWorld Magazine*, a monthly publication of The Source on-line information service, reports that Texas Instruments has plans to begin a TI sponsored Special Interest Group (SIG) called TexNet on The Source. No date is given for TexNet availability.

— In a products price list sent to retailers, TI announces 38 new hardware and software products for the 99/4 that are scheduled to be available during the first quarter of 1981. This would turn out to be one of the two largest mass releases of Home Computer products that TI would announce during the life of the Home Computer. The other would occur during the second and third quarters of 1983, but by then it was too little, too late.

Keeping track of petrol costs

Program does calculations and creates reports

By BEN V. TAKACH

The following article appeared in the TIsHUG News Digest of Sydney, New South Wales. It uses liters, kilometers and imperial gallons for its calculations, but readers should be able to modify it to work with other systems.—Ed.

Politicians at times will say things on the spur of the moment which happens to be something wise and truthful.

I guess you may well remark that this is a strange way to introduce a computer program. The connection is the famous remark by which one of our otherwise lackluster leaders will be long remembered — "There is no such thing as a free lunch."

Now, as you run this program you will be able to prove conclusively how true this statement is. Did you know that if you live, say, in Sutherland and accept my invitation to a free lunch in Wahroonga, by the time you pack your car in the garage the free lunch will have cost you more than $10? This is exactly what the program will calculate for you. The printout example of a periodical report illustrates the data produced by the program. Naturally, one does not have to print it to paper if it is not needed. The results are displayed on the screen and, if so desired, may be saved to disk as a D/V80 file for future use.

Analyzing past reports and comparing these with current costs yields interesting results. Did you know that the average price of petrol in Sydney has risen 39.6 percent from the beginning of 1989 to July 1992 (from 48 cents/liter to 65 c/l)? The rise has caused an increase in my fuel costs from seven cents to 10 cents per kilometer, an increase of 43 percent. Considering that incomes have not increased during the same period, this is one example of our rapidly decreasing standard of living.

FUELCOST

```
100 ! SAVE DSK1.FUELCOST !15
2
110 !*********************
 !073
120 !* FUEL USED & COST C. *
 !196
130 !*********************
 !073
140 ! BY BEN TAKACH !255
150 ! LAST REV. 2.July.89 !2
13
160 ON WARNING NEXT !215
170 DIM L(25),C(25)!008
180 CALL CLEAR :: GOSUB 890
!032
190 CALL CLEAR :: DISPLAY AT
(8,3):"FUEL COSTS & CONSUMPT
ION." :: DISPLAY AT(14,1):"
Wish to see the instruc-  t
ions (y/n) ?" !017
200 CALL KEY(3,RR,ST):: IF S
T<>1 THEN 200 !210
210 IF RR=78 THEN 220 ELSE I
F RR<>89 THEN 200 ELSE GOSUB
 700 !167
220 CALL CLEAR :: DISPLAY AT
(6,3):"FUEL COST & CONSUMPTI
ON." :: DISPLAY AT(10,1):"CA
LCULATES 1/100 km, M/Gal & F
uel Costs..." :: DISPLAY AT(
23,1):"Printout required ? (
y/n)" !055
230 FL=0 :: IF P$="" THEN P$
="PIO" !220
240 CALL KEY(3,RV,ST):: IF S
T<>1 THEN 240 !254
250 IF RV=89 THEN DISPLAY AT
(23,1):"Print Device? ";P$ :
: ACCEPT AT(23,15)SIZE(-8):P
$ :: FL=1 ELSE IF RV=78 THEN
 270 ELSE 240 !073
260 OPEN #1:P$,OUTPUT !105
270 CALL CLEAR :: DISPLAY AT
(1,1):"Enter start & ending
dates of the period analyzed
 (Max. 21 Char. allowed): ":
DD$ !096
280 DISPLAY AT(6,1):"Initial
 odometer reading at full ta
nk (km) ?";R1 !011
290 DISPLAY AT(8,1):"Final o
dometer reading at  full ta
nk (km) ?";R2 !110
300 DISPLAY AT(11,1):"How ma
ny fillups will be entered ?
";RF !009
310 ACCEPT AT(4,1)SIZE(-21):
DD$ !083
320 ACCEPT AT(7,18)SIZE(-6)V
ALIDATE(DIGIT):R1 !140
330 DISPLAY AT(22,1):"Type 0
 to correct an earlier enter
ed incorrect answer" !037
340 ACCEPT AT(9,18)SIZE(-6)V
ALIDATE(DIGIT):R2 :: IF R2=0
 THEN 270 !088
350 ACCEPT AT(12,18)SIZE(-3)
VALIDATE(DIGIT):RF :: IF RF=
0 THEN 290 !188
360 IF RF>20 THEN DISPLAY AT
(22,1)BEEP:"Only 20 fillups
are allowed per report.  Ple
ase reenter." :: GOTO 350
53
370 RX=R2-R1 :: LX=0 :: CX=0
 :: R0=7 :: CALL CLEAR !247
380 DISPLAY AT(1,3):RPT$("-"
,25):: DISPLAY AT(2,3):"|Fil
lups| Litres|Cost $|" :: DIS
PLAY AT(3,3):"|";RPT$("-",23
);"|" !120
390 FOR I=1 TO RF :: DISPLAY
 AT(I+3,3):"|";I;".";TAB(12)
;"|";TAB(20);"|";TAB(27);"|"
 !119
400 IF I=20 THEN 420 !213
410 NEXT I !223
420 FOR J=1 TO RF !215
430 ACCEPT AT(J+3,14)SIZE(-6
)VALIDATE(NUMERIC):L(J):: IF
 L(J)=0 THEN 430 !246
440 ACCEPT AT(J+3,21)SIZE(-5
)VALIDATE(NUMERIC):C(J):: IF
 C(J)=0 THEN 440 !235
450 LX=LX+L(J):: CX=CX+C(J):
: NEXT J :: GOSUB 900 !182
460 CA=CX/LX :: FCK=(LX/RX)*
CA :: LK=(LX/RX)*100 :: MG=
X/LX*2.82481 :: CALL CLEAR
198
470 DISPLAY AT(5,1):"Total k
```

(See Page 23)

FUELCOST —

(Continued from Page 22)

```
m-s driven  ";RX !027
480 DISPLAY AT(6,1):"Total f
uel used (1)";INT(LX*100)/10
0 !080
490 DISPLAY AT(7,1):"Total c
ost ($)     ";INT(CX*100)/10
0 !023
500 DISPLAY AT(8,1):"Fuel co
st (c/km)   ";INT(FCK*10000)
/100 !065
510 DISPLAY AT(9,1):"Avg. co
st/l (c/l)  ";INT(CA*100)/10
0 !095
520 DISPLAY AT(10,1):"Litres
/100km      ";INT(LK*100)/10
0 !246
530 DISPLAY AT(11,1):"Miles/
imp.gallon  ";INT(MG*100)/10
0 !131
540 DISPLAY AT(22,1):"PUSH A
NY KEY TO CONTINUE" :: CALL
KEY(3,RV,ST):: IF ST<>1 THEN
 540 !181
550 IF FL=0 THEN 980 ELSE DI
SPLAY AT(22,1):"WISH A PRINT
OUT OF THE FULL REPORT OR TH
E SUMMARY (f/s)?" :: CALL KE
Y(3,RV,ST):: IF ST<>1 THEN 5
50 !242
560 IF RV<>70 AND RV<>83 THE
N 550 ELSE IF RV=70 THEN 650
 !053
570 PRINT #1:TAB(10);"FUEL C
OST SUMMARY":TAB(10);RPT$("-
",17):: PRINT #1 :: PRINT #1
:TAB(5);"For period: ";DD$ !
177
580 PRINT 31:TAB(5);"Total k
m-s covered ";RX !150
590 PRINT #1:TAB(5);"Total f
uel used (1)";INT(LX*100)/10
0 !083
600 PRINT #1:TAB(5);"Total c
ost ($)";INT(CX*100)/100 !11
6
610 PRINT #1:TAB(5);"Fuel co
st (c/km)";INT(FCK*10000)/10
0 !223
620 PRINT 31:TAB(5);"Avg. co
st/l (c/l)";INT(CA*10000)/10
0 !182
630 PRINT #1:TAB(5);"Litres/
100km ";INT(LK*100)/100 !040
640 PRINT #1:TAB(5);"Miles/i
mp. gallon  ";INT(MG*100)/10
0 :: PRINT #1 :: PRINT #1:TA
B(10);RPT$("_",5):: PRINT #1
 :: CLOSE #1 :: GOTO 980 !15
7
650 PRINT 31: : :TAB(5);"Fil
l up and cost details:": :!0
70
660 PRINT #1:TAB(5);"ODOMETE
R AT START (km)";R1 :: PRINT
 #1:TAB(5);"ODOMETER AT END
(km)";R2 :: PRINT #1 !227
670 PRINT #1:TAB(5);" Fillup
s litres cost $ " !048
680 FOR J=1 TO RF !215
690 PRINT #1:TAB(7);J;TAB(16
);L(J);TAB(23);C(J):: NEXT J
 :: PRINT #1 :: GOTO 570 !05
0
700 GOSUB 870 !185
710 CALL CLEAR :: PRINT "
 FUEL COST CALCULATOR" :: PR
INT :: PRINT "The program wi
ll calculate, then display a
nd optionally print the foll
owing details:" !193
720 PRINT TAB(5);"Total fuel
 used (1)":TAB(5);"Total cos
t ($)":TAB(5);"Fuel cost (c/
km)":TAB(5);"Av. cost (c/l)"
:TAB(5);"Consumption (l/100k
m)" !159
730 PRINT TAB(5);"Consumptio
n (M/Imp.Gal.)" :: PRINT :"T
he program will accept 20 fi
llups per report. This limit
 will aid statistical evalua
tion of the results." !062
740 PRINT "The results may a
lso be saved (printed) to di
sk in D/V80 format. The prog
ram may be ended after a bat
ch is completed, or it may b
e rerun." !074
750 PRINT "You may proceed t
hen with  the next lot of e
ntries." !020
760 PRINT : : :: GOSUB 880 :
: GOSUB 850 !114
770 CALL CLEAR :: GOSUB 870
!012
780 PRINT :"HOW TO PREPARE T
HE DATA": :"First fill up yo
ur car and note the odometer
 reading. This is the starti
```

```
ODOMETER AT START (km) 13500
ODOMETER AT END (km) 19000

 Fillups litres cost $
    1        50      27.5
    2        75      37
    3        60      31.25
    4        80      41.75
    5        75      40.24
    6        56      30.83
    7        66      33
    8        69      35.12
    9        53      29.2
   10        70      35

      FUEL COST SUMMARY
      -----------------

For period: 1/1/95-5/1/95
Total fuel used (1) 654
Total cost ($) 340.89
Fuel cost (c/km) 6.19
Litres/100km  11.89
Miles/imp. gallon   23.75
```

```
ng km." !218
790 PRINT "Subsequently note
 the date, the odometer read
ing, the litres tanked and t
he amount paid each time the
 tank is filled." !077
800 PRINT "You do not have t
o completely fill the tank i
f you do not wish to do so,
except for the last entry of
 the period. Just enter the
data " !113
810 PRINT "requested by the
program." !240
820 PRINT "The program may b
e modified to display other
data, e.g. the date of each
refill or comments." !072
830 PRINT : : :: GOSUB 880 :
: GOSUB 850 !114
840 PRINT :: RETURN !166
850 DISPLAY AT(23,1):"PUSH A
NY KEY TO CONTINUE" :: CALL
KEY(0,RR,ST):: IF ST<>1 THEN
 850 !230
860 RETURN !136
870 FOR XX=0 TO 14 :: CALL C
OLOR(XX,1,5):: NEXT XX :: RE
```

(See Page 24)

FUELCOST —

(Continued from Page 23)

```
TURN !139
880 FOR XX=0 TO 14 :: CALL C
OLOR(XX,16,1):: NEXT XX :: R
ETURN !190
890 CALL SCREEN(5):: FOR XX=
0 TO 14 :: CALL COLOR(XX,16,
1):: NEXT XX :: RETURN !214
900 DISPLAY AT(24,1):"ANY CO
RRECTIONS (y/n)?" :: CALL KE
Y(3,RV,ST):: IF ST<>1 THEN 9
00 !158
910 IF RV=78 THEN 970 ELSE I
F RV<>89 THEN 900 !208
920 DISPLAY AT(24,1):"ENTER
FILLUP No." :: ACCEPT AT(24,
18)VALIDATE(DIGIT)SIZE(2):R0
 !084
930 LX=LX-L(R0):: CX=CX-C(R0
)!109
940 ACCEPT AT(R0+3,14)SIZE(-
6)VALIDATE(NUMERIC):L(R0)::
IF L(R0)=0 THEN 940 !158
950 ACCEPT AT(R0+3,21)SIZE(-
5)VALIDATE(NUMERIC):C(R0)::
IF C(R0)=0 THEN 950 !147
960 LX=LX+L(R0):: CX=CX+C(R0
):;GOTO 900 !170
970 RETURN !136
980 CALL CLEAR :: DISPLAY AT
(12,1):"Wish to continue wit
h a new lot of entries, prin
t report to a disk file, or
exit the program? (c/d/e)" !
115
990 CALL KEY(3,RR,ST):: IF S
T<>1 THEN 990 !235
1000 IF RR=67 THEN R1=R2 ::
GOTO 220 ELSE IF RR=69 THEN
END ELSE IF RR=68 THEN 1010
ELSE 990 !082
1010 REM FILE PRINT ROUTINE;
 END WITH GOTO 910 COMMAND !
141
1020 CALL CLEAR :: IF DV$=""
 THEN DV$="DSK1" !036
1030 DISPLAY AT(6,1):"DEVICE
 NAME?  ";DV$ !023
1040 DISPLAY AT(7,1):"FILE N
AME?    ";FI$ !189
1050 ACCEPT AT(6,15)SIZE(-12
):DV$ !157
1060 ACCEPT AT(7,15)SIZE(-10
):FI$ !145
1070 DF$=DV$&"."&FI$ :: OPEN
 #2:DF$,DISPLAY ,VARIABLE 80
,APPEND !152
1080 PRINT #2: :  :TAB(5);"Fi
ll up and cost details:": :!
016
1090 PRINT #2:TAB(5);"ODOMET
ER AT START (km)";R1 :: PRIN
T #2:TAB(5);"ODOMETER AT END
 (km)";R2 :: PRINT #2 !230
1100 PRINT #2:TAB(5);"  Fill
ups  Litres Cost $ " !051
1110 FOR J=1 TO RF !215
1120 PRINT #2:TAB(7);J;TAB(1
6);L(J);TAB(23);C(J):: NEXT
J :: PRINT #2 !039
1130 PRINT #2:TAB(10);"FUEL
COST SUMMARY":TAB(10);RPT$("
-",17):: PRINT #2 :: PRINT #
2:TAB(5);"For period: ";DD$
!180
1140 PRINT #2:TAB(5);"TOTAL
Km-s COVERED ";RX !224
1150 PRINT #2:TAB(5);"TOTAL
FUEL USED (l)";INT(LX*34100)
/100 !120
1160 PRINT #2:TAB(5);"TOTAL
COST ($)";INT(CX*100)/100 !1
17
1170 PRINT #2:TAB(5);"FUEL C
OST (c/km)";INT(FCK*3410000)
/100 !105
1180 PRINT #2:TAB(5);"AVER.C
OST/l (c/l)";INT(CA*10000)/1
00 !240
1190 PRINT #2:TAB(5);"litres
/100km      ";INT(LK*34100)/
100 !087
1200 PRINT #2:TAB(5);"miles/
imp. gallon";INT(MG*100)/100
 :: PRINT #2:TAB(10);RPT$("_
",5):: PRINT #2 :: CLOSE #2
:: GOTO 980 !106
```

1995 TI FAIRS

SEPTEMBER

10th International TI-Meeting, Sept. 22-24, Wohlfahrtsgebäude der Wiener E-Werke (Welfare Building of the Vienna Electricity Board), Wachaustr. 28, A-1020 Vienna, Austria. For information write Kurt Radowisch, TI- and Geneve User Group Vienna, Grossbauerstr. 24, A-1210, Vienna, Austria.

TI New England Fall Faire, Sept. 30, Emanuel Lutheran Church, 200 Greenwood St., Worcester, Massachusetts. Contact Jim Cox, 905 Edgebrook Dr., Boylston, MA 01505 or (508) 869-2704.

OCTOBER

Chicago International TI Faire, Oct. 28, Evanston Public Library. Contact Chicago TI Users Group, P.O. Box 7009, Evanston, IL 60204-7009, or Hal Shanafield, (708) 864-8644.

1996 TI FAIRS

FEBRUARY

Fest West '96, Feb.17, Ramada Inn, 1601 Oracle Dr., Tucson, Arizona. Contact SouthWest Ninety-Niners User Group by sending e-mail to twills@primenet.com. Or call the Cactus Patch BBS at (520) 290-6277; BJ Mathis (520) 747-5046; Tom Wills (520) 886-2460; or write Fest West 96 Committee, SouthWest 99ers, P.O. Box 17831, Tucson, AZ 85731-7831.

MAY

Multi Users Group Conference, May 25, Ohio National Guard Armory, Brookpark. Contact Glenn Bernasek, 13246 Harper Rd., Strongsville, OH 44136, or call (after 9 p.m. Eastern time) at (216) 846-0865 or Internet dd314@cleveland.freenet.edu.

Random choices

Decision making in Extended BASIC

By W. LEONARD TAFFS

The following article appeared originally in the newsletter of the Southwest 99ers.—Ed.

Have you ever loaded a program that takes a fairly long time to start up, only to find it won't run or is interrupted or crashes because it requires a file you forgot to have available to the program? Whether the program is large or small, something you can do about it is to insert a line at the beginning that flashes a reminder to you. The following Extended BASIC line is an example:

```
10 CALL CLEAR:: DISPLAY AT(
15,1):"REQUIRES: File ""SCOT
PSTCRD""" :: CALL KEY(0,K,S)
:: FOR DLY=1 TO 100 :: NEXT
DLY :: IF S<1 THEN 10
```

If you have Super-XB, you could add _L HONK or CHIMES, etc., before "IF S<1...."

For TI BASIC use the following:

```
10 CALL CLEAR
20 CALL KEY(0,K,S)
30 FOR DLY=1 TO 5
40 NEXT DLY
50 PRINT "REQUIRES: File ""S
COTPSTCRD"""
60 CALL CLEAR
70 IF S<1 THEN 20
```

This is a simple CALL KEY statement with a CALL CLEAR and a PRINT statement (DISPLAY AT for XB) and a DELAY loop buried in between. The delay (DLY) can be set to your preference as to repetition rate of the flashing reminder. In either above version, pressing Enter will move on to the next command.

It may be the nature of your program is such that even if you include this warning in the early part of the program, it will not appear until it has finished initializing — which defeats tbe purpose of this warning in the first place. If this is the case, then the warning eessage must be a separate program in TI BASIC — the warning can print the name of the program file. In Extended BASIC you simply add a line following line 10 that says: "RUN DSKn YOURPROGRM". Once you are sure the needed files are in place, pressing Enter after the warning will run your program.

This little gimmick might be worth the little trouble it takes, particularly if you run the program 2 years from now, at which time you may have forgotten the program needed certain files.

Fun-type decision makers have been around since the inception of the simplest computer. They make use of random choice, as everyone knows. They are, in essence, the simplest type of gambling.

DECISION MAKERS

Fun-type decision makers have been around since the inception of the simplest computer. They make use of random choice, as everyone knows. They are, in essence, the simplest type of gambling. You truly take a ganble if you let the program make your decision for you. Afterall, the computer has no pros and cons to consider in making a decision.

Somewhere I saw a program that included many considerations for the computer to take into account before giving its answer, but I have long since forgotten where I saw it. It may have been one of those personal analysis type programs. Many games have to rely on similar algorithms. If you any type of such program (not games) done, I would like to hear about it. It would be nice to have such a program to enable you to enter your own data considerations. Then the computer could make an evaluated decision based upon what appears to be your best choice.

At any rate, the following is a simple "YES/NO" random choice program. I say "simple" because the basic algorithm is very simple (lines 90-120). The listing is long, due to the various options and displays. The extra lines are worth it because you can while away some hours with its simple options.

YES/NO#03

```
1 REM [!YES/NO#03] 10-26-95
!135
2 !!131
3 REM TOT1=YES !016
4 REM TOT2=NO !189
5 REM TOT3=TIES !086
6 REM TOT4=TURNS !190
7 REM TOT5=RUNS !107
8 !!131
10 CALL CLEAR :: DISPLAY AT(
8,7):"YES/NO OR TIE?" :: DIS
PLAY AT(13,1):"By W. Leonard
 Taffs, SW99ers" :: DISPLAY
AT(18,10):"10-25-95" !244
20 DISPLAY AT(24,1):"PRESS <
ENTER> to Continue..." !011
30 CALL KEY(0,K,S):: IF S<1
THEN 30 !100
40 CALL CLEAR :: INPUT "SET
ON AUTO? (Y/N) ":AUTO$ :: IF
ON AUTO? (Y/N) ":AUTO$ :: IF
 AUTO$<>"Y" AND AUTO$<>"y" T
HEN AUTO=0 ELSE AUTO=1 !102
50 IF AUTO THEN PRINT :: INP
UT "ENTER # OF RUNS: ":NR EL
SE NR=1 !091
60 RUNN=1 !251
70 CALL CLEAR :: INPUT "# OF
 TRIES: ":N !148
80 CALL CLEAR !209
90 RANDOMIZE !149
100 A=INT(1000*RND)!098
110 IF A<=500 THEN YN$="NO"
:: T2=T2+1 :: GOTO 150 !175
```

(See Page 26)

DECISION MAKING —

(Continued from Page 25)

```
120 IF A>500 THEN YN$="YES"
:: T1=T1+1 !222
130 DISPLAY AT(8,10):"Run #"
;RUNN !206
140 DISPLAY AT(10,10):"Turn"
;TRN !217
150 DISPLAY AT(13,12):YN$ !1
41
160 DISPLAY AT(16,5):"YES=";
T1 :: DISPLAY AT(16,6):"NO="
;T2 !191
170 TRN=TRN+1 :: TOT4=TOT4+1
 :: DISPLAY AT(10,10):"TURN"
;TRN: !097
180 IF TRN=N THEN 190 ELSE 9
0 !095
190 PRINT :" RESULTS
:" !181
200 IF T1>T2 THEN PRINT :"
 THE WINNER IS ""YES""!" ::
TOT1=TOT1+1 !223
210 IF T2>T1 THEN PRINT :"
 THE WINNER IS ""NO""!" :: T
OT2=TOT2+1 !140
220 IF T2=T1 THEN PRINT :"
 IT IS A TIE!!" :: TOT3=TOT3
+1 !210
230 IF AUTO=0 THEN 270 !252
240 IF AUTO THEN CALL KEY(0,
K,S):: IF S<1 THEN 240 !163
250 REM ** ACTIVATE AUTO **
!076
260 IF AUTO THEN 370 ELSE NA
=NA+1 :: IF NA<>N THEN 80 !2
36
270 PRINT : : :: PRINT "RUN
AGAIN? " !202
280 CALL KEY(0,K,S):: IF S<>
1 THEN 280 !031
290 IF K=89 THEN 320 ELSE 30
0 !249
300 IF K=121 THEN 320 ELSE 3
10 !039
310 IF K=81 THEN STOP !008
320 A,T1,T2,TRN=0 :: PRINT :
"SAME # OF TURNS? " !244
330 CALL KEY(0,K,S):: IF S<1
 THEN 330 !145
340 IF K=89 THEN RUNN=RUNN+1
 :: GOTO 80 ELSE 350 !065
350 IF (K=78)+(K=110)THEN 70
 !134
360 IF K=121 THEN RUNN=RUNN+
1 :: GOT0 80 ELSE 380 !131
370 A,T1,T2,TRN=0 :: IF AUTO
 THEN AUT=AUT+1 :: RUNN=RUNN
+1 :: IF AUT=NR THEN 380 ELS
E 80 !171
380 PRINT :"In";RUNN-1;"runs
, Out of";TOT4;"turns": :" W
INNERS were:": : :: PRINT TO
T1;" ""YES"" and";TOT2;" ""N
O"" ": :" and";TOT3;" TIES
." !129
390 IF AUT THEN IF TOT1=TOT2
 THEN CALL SCREEN(11):: CALL
 BEFP :: PRINT :"TOTALS ARE
A TIE!!!" !110
```

There are two modes of using the program. One is to enter "N" for NO to "SET ON AUTO?" or "Y" for "YES."

The AUTO mode allows you to run tbe program for any number of runs you wish. A CALL REY activates at end of each run-display (press Enter to proceed to next run). At the end of the run the total number of runs will be displayed with a tally of YES and NO wins and TIES, if any.

Not choosing AUTO will run the program once. At the end of each run you opt to run again using"'Y." If you run again, you are asked if you with the same number of tries. Again a "Y" will rerun the program with the same number of tries, or if you press "N" to the same number of tries, you are given the chance to input a different number of tries.

To quit the program, enter "Q" at the prompt "SAME # OF TURNS?" The permutations and combinations of winning possibilities of your choice are considerable which makes many games possible (using your imagination). And this is with only two variables you can quickly see why winning the lottery with five or or six numbers is such a gamble!

Note in lines 110 and 120 that (1000*RND) of line 100 is equally divided to give even chances of getting a YES or NO. The number 1000 is arbitrary but, being a large number, expands the random choice. What happens if you change th number?

Another thought that comes to mind is about how gambling casinos often are said to stack the deck against their customers. Perhaps they divide variable A to something like this:

```
110 IF A<=900 THEN YN$="NO"
:: T2=T2+1 :: GOTO 150
120 IF A>900 THEN YN$="YES"
:: T1=T1+1
```

Something to ponder when you are next tempted to visit a casino.

How a TI found a new life

The following was written by Gary Kuehn of the Pittsburgh User Group.

How many of you have an extra 99/4A console and speech synthesizer? Are they sitting around, gathering dust, or in a closet on scme forgotten section of a shelf in a dark part of the house?

How many of you thought that someday you would get the console out, and give it to someone who could use it? Or even use it to teach your children spelling or math or reading.

How many of you thought about using the console for a project that you wanted to work on, but you didn't want to tie up the 486 with a little BASIC programming. Even if you felt like trying to use BASIC on the PC, that is if you knew how to find it?

How many of you thought about these things then decided to forget about it because it was too much work to cable up the TI, then find a TV that wasn't being used, locate a quiet room to work in, and dig out the old BASIC manual? If you find yourself in any of these categories then read on....

I found myself with an extra console and decided to put it to use in our neighborhood daycare center. I mentioned to the dir-

(See Page 28)

MICRO-REVIEWS

80 columns without the cost

By CHARLES GOOD

TERM 80
by Jeffrey Brown

My entire column this month is devoted to a new important piece of software.

In these modern times, with more and more TIers using the Internet, Term 80 is probably going to replace Telco as the premiere terminal emulation software for use on a 99/4A.

Unless you have access to a computer bulletin board system specifically designed for use with the 99/4A you really need an 80-column display to telecommunicate with computers effectively. All Internet access providers and most BBS systems and national information systems produce only an 80-column text display. In most cases these computer telecommunication networks use the VT100 or the ANSI terminal emulation protocol. The ANSI protocol includes eight colors, specifically defined tabs and other standards for formatting text graphics and sound. Both VT100 and ANSI need an 80-column display. If all you have is a basic 99/4A disk system, then the best you can normally get in text mode is 40 columns. Using Telco and a 99/4A to telephone my local library's on-line catalog or dial into my Ohio State University Internet account produces a 40-column display that is hopelessly confusing. These systems require VT100 emulation. Telco has "VT100NET" and "VT100VAX" but these don't work with my Internet account or the library catalog. The confusion is much more than what you get with a word processor, where you can sort of get used to scrolling left/right and up/down to simulate an 80-column screen. Unlike a word processor, when you bring up an 80-column telecommunication screen on a 40-column display the cursor is often anywhere to be found! It is off somewhere . . can't see, and that really makes things difficult.

The new terminal emulation software Term 80 solves these problems. In addition to VT52 emulation, which is the main emulation supported by Telco, you get true VT100 emulation and a nice 80-column display on a regular 99/4A system. Internet access needs VT100 emulation. You need no special hardware (no 80-column card, no analog RGB monitor, no expanded video memory) to get a nicely formatted 80-column display with Term 80. No other software of any type gives you an 80-column text display on a basic 99/4A system. How is this done? Each character is only three pixels wide and there are no spaces between the characters. This uses 240 pixels of the 256 pixels available in the width of a 99/4A display. The character set includes upper and lower case letters as well as a fairly complete set of foreign language accented vowels. Using special key presses any upper or lower case vowel can have an acute (´), grave (`), circumflex (^) or umlaut (¨) over it. You can also put a ( ) over an N and you can make a cedilla (ç). Actually the entire IBM extended character set is displayable, including lines, blocks, mathematical symbols, etc.

All aspects of Term 80 operation are controlled by an initial configuration, which you only have to do once to match the software to your hardware, and by a bunch of 80-column pull-down menus.

"Impossible," you say! Nobody could read three-pixel-wide characters on as TI monitor! Well, I can assure you these characters can be read. You can see them for yourself on the videotapes of the April 1995 Lima Multi User Group Conference. The character set is very cleverly designed. Reading it isn't easy, but having a correctly proportioned 80-column display makes the effort worthwhile. You need a composite color or monochrome monitor to adequately read this tiny character set. A TV isn't good enough. I have tried Term 80 successfully on several composite monitors. You may have to mess around with the focusing and color controls. Term 80 lets you turn off the ANSI color display and view white on black, which often makes it easier to read text. If your eyeglasses just can't handle the 80-column display you can switch to either 64, 54, 40, 36, or 32 columns of text for the input/output display. If you do this you don't lose ANSI colors or VT100 emulation, but your text display may look funny. The help screens and menus are always in 80 columns.

All aspects of Term 80 operation are controlled by an initial configuration, which you only have to do once to match the software to your hardware, and by a bunch of 80-column pull-down menus. Many of the preconfigured options can be altered in these menus. To bring down a menu, press either CTRL or FCTN and a number. If you can't remember how to access a menu you can always press AID. Because of the complexity of Term 80's expanded character set and the many ANSI and other options available it is sometimes necessary to press three TI keys simultaneously to get some things to happen.

Minimum requirements for Term 80 are a 99/4A, one DSSD drive, 32K, RS232 and either a modem or direct serial cable to another computer. If you have Mini-Memory or Supercart or GRAM device (such a PGRAM+) or an Asgard Memory Systems card of any size, any of these can be used to expand the size of the input buffer significantly. The current version allows for up to 64K of buffering space on an AMS card. The author states that he hopes to completely rewrite Term 80 in its next version so that all its modules can re-

(See Page 28)

MICROREVIEWS —

(Continued from Page 27)

side on an AMS card and that all the rest of the memory of the card will be usable as buffering space.

A RAMdisk is nice because Term 80 is kinda slow when run from floppy. Term 80 doesn't like Geneves and it doesn't like hard drive long path names. All configuring requires Term 80 system files to be on a device named "DSK" followed by one digit. To get VT100 emulation on a Geneve you should use PORT, available from Cecure Electronics, P.O. Box 132, Muskego, WI 53150. Phone (414) 679-4343 (voice), (414) 679-3736 (fax).

Although you can dial manually if necessary, you will want to use the autodialer if possible. It is very full featured. You can program in pauses within the dialed number (if you have to wait a second for your long distance service to kick in, for example), and you can tell Term 80 to dial any desired sequence of numbers. Of course, you can program the autodialer to dial the same number several times, with a user-defined pause between dialing attempts. Each number you program into the autodialer can have its own baud rate and autologon script.

Downloads and uploads can be TIBBS Xmodem (uses a TI files header), Xmodem CRC, Xmodem-1K (data segments are in 1K chunks rather than 128-byte chunks), Ymodem batch, and ASCII text. These data transfers can be done rapidly. Downloads work nicely at 9,600 baud and uploads will sometimes work at 19,200 baud! That's right folks, the TI RS232 will support 19,200 baud. This means you can go out to almost any computer store and purchase a really fast external modem to use on the good old TI. I have seen 14,400 external modems for around $100 in catalogs and stores. Term 80 provides the ability to alter RS232 and keyboard scan rates so that you can adjust your system for the best and fastest performance. Jeffrey Brown tells me his fastest measured, upload transfer rate is 14,00 cps from his 99/4A cabled to his Amiga, which is really fast.

A powerful feature of Term 80 is its use of macros. You create macros using any text editor and save them to disk. Later any macro can be called up from a menu of available macros just by pressing one key from the menu display. These macros are normally ASCII text and may be up to 3K in length. There really is no limit, however, because macros can be chained. Having macros call each other gives the potential for unlimited size. In addition to regular ASCII text you can program in the following into macros: ASCII control codes 0-31, half second and two second pauses, any high ASCII number (even those you can't directly type), and an indefinite pause with wait for input (similar to INPUT in BASIC).

Just like other modern software for PCs, Term 80 has a screen saver which you can call manually or have automatically kick in after 10 minutes of inactivity. Incoming data or a key press turns off the screen saver.

What is missing in addition to a couple of file transfer protocols (that I am told are almost impossible to implement on a 99/4A) are a review buffer, a file editor and a disk cataloger. These may be made available in future updates.

In conclusion, Term 80 gives 99/4A owners access to the Internet and it is good for TIers to have this option. Using Term 80 for Internet access is certainly cheaper than the alternatives, which are purchasing a Geneve and PORT or another type of 80-column PC.

Term 80 comes in two forms. There is a demo version which can be freely distributed and put into user group libraries and a complete version. Both include a bunch of useful TI utilities written by various authors as well as a disk defragmenter and a couple of free games written by the Term 80 author. Send me $1 and I will send you the demo version archived on a DSSD disk. The demo does not include the autodialer, or the ability to download and upload. Some sample macros are included. With the demo version you can manually call a BBS and look, but you can't take.

Since the Chicago show Jeff has released a new, less buggy, demo and full version (v3.1.4) of Term 80. If you previously obtained the demo version from me by mail or at the Massachusetts or Chicago fairs you may want to send me $1 and get the newer demo version. Sorry for the inconvenience. Believe me, I don't make a profit on these dollar disks.

The complete version is only available directly from the author. It costs $15 + $4 for a mailer and postage US funds (cash is nice, wrapped in several sheets of paper so you can't tell there is money in the envelope) to United States residents or $20 Canadian money plus $3 for a mailer and postage if you live in Canada. To registered users the author Jeffrey Brown promises support via telephone (not toll free) or the Internet and mail notification of major updates which will only cost media and postage. By the way, the author is 15 years old and has been given permission to take senior level computer courses at his local high school. Like one of my own sons, he probably knows more about computers than any of his high school computer teachers.

ACCESS:

Jeffrey Brown (Term 80). 2111 Montreal Rd. #102, Gloucester, Ontario K1J 8M8, Canada. Phone (613) 746-1013. E-mail bb737@freenet.carleton.ca

Charles Good. P.O. Box 647, Venedocia, OH 45894. Phone 419-667-3131. E-mail cgood@osulima1.lima.ohio-state.edu or good.6@osu.edu

A NEW LIFE—

(Continued from Page 26)

ector that I had an extra computer at home. Since it wasn't being put to use, I offered to set it up in the center with some learning programs to see if the children would use it. She thought it was a good idea.

Being a smart businesswoman, and thinking a computer was an IBM or clone, she naturally thought of all the things she could do with it.

(See Page 29)

USER NOTES

Search string corrected

Reader Stan Ulanoff points out a problem in last month's User Note headlined "24-pin printers and PagePro99." The article includes a search string and changes to make to it. The search string should be 1B 41 08 and the replacement string should be 1B 33 14.

Q*Bert for the 9640

The following was written by Oliver Arnold. We found it in Bits, Bytes & Pixels, the newsletter of the Lima 99/4A User Group.

Until now I didn't get a version of Q*Bert that works correctly with 80-column systems, or the Geneve 9640. So I tried to find the error by myself. I used the Miller Graphics Explorer program to analyze the Q*Bert module and found the error. It is a loop routine that sets the VDP registers. This loop is going to counter eight. On an original TI, this has no effect because the old VDP has only seven registers. But the 9938 used in 80-column systems has an eighth VDP register. This register sets a new mode that blanks the screen.

To correct the program, you need a sector editor. Search for the bytes >02808800. Change them to >02808700.

If you want to start Q*Bert with the Explorer, here are the necessary addresses:

CPU	GROM
WS >83E0	AD >E022
PC >006A	

During the start process you can see how the GPL interpreter works.

Randomizer

The following was written by Woody Wilson and appeared in *The Computer Voice*, the newsletter of the Southern California Computer Group.

This little program is an adaptation of a program by Jim Peterson. Frank Frankenberger asked me for a program that could randomize some numbers and not repeat any of the selections. I recalled Peterson's NO REPEAT program, made a few changes to it and gave it to Frank.

As given here, the program can handle up to 26 numbers. This number can be increased to 62 by changing the A$ in line 120 to use the additional ASCII characters from 91 to 126.

RANDOMIZER

```
100 !RANDOMIZER PROGRAM NEVE
R SELECTS SAME NUMBER TWICE.
  DEVELOPED FROM A PROGRAM B
Y JIM PETERSON. W.A.WILSON,
SCCG 10/24/95
110 INPUT "HOW MANY PRESENT?
":N
120 A$=SEG$("ABCDEFGHIJKLMNO
PQRSTUVWXYZ",1,N)
130 FOR J=1 TO N
140 RANDOMIZE
150 Y=INT(RND*LEN(A$)+1)
160 X=ASC(SEG$(A$,Y,1))-64
170 A$=SEG$(A$,1,Y-1)&SEG$(A
$,Y+1,LEN(A$))
180 PRINT J;"=";X:
190 NEXT J
```

The advantages of cloning

The following was written by Mel Samouri of the TIsHUG user group in Sydney, New South Wales, Australia.

The bit map and sector copy functions of DM-1000 work like this:

Bit Map — only the sectors mapped in the disk directory as being used are copied. This means that fractured files will be un-fractured, if possible, on the copy disk, and only those sectors are copied. This is best as it doesn't copy unused, or blank, sectors. So, it's a little quicker.

Sector — This allows you to "clone" a disk. Some software companies use renumbered sectors as protection or purposely fracture files or part of a disk (blown directory) and, if the defaults are used, will make an exact duplicate of the disk. If you are not familiar with how the disk is set up, use the defaults. If you don't, you could get only part of your programs copied, as it doesn't use the sector map. For double-density, an extended bit map is used. They're not all in sectors zero to 20 if you have a lot of files on the disk.

You should, if possible, perform what is called "optimizing" on file disks that you use frequently or delete or add programs to. You want to do this because as the files become fractured, the length of time it takes to load them increases. It also adds to the wear and tear on the disk drives and could result in data loss. A "fractured" file is a file that is not stored contiguously on a disk but in several pieces in different sectors. You can tell the extent of fracturing

(See Page 30)

A NEW LIFE—

(Continued from Page 28)

When I showed up with a console, a speech synthesizer, a few cartridges, a bunch of cables, and an old monitor, she was a little worried. Would an old computer like this fit the bill at all?

Calmly I hooked up all the wires, plugged in the power leads, set up the monitor, turned it on, plugged in a cartridge and said "try it out." Once the director saw the beautiful colors and saw how easy things were to use, she changed her mind about this old computer. When the the computer started talking to her, she was hooked on the TI.

Children use it regularly for learning to read, count, spell and do math. If we can find another old monitor, I know were there is another console with speech to pass on to the daycare center so more kids can discover the TI and the magic that we all learned about long ago and are still learning about today. Each day the 3-, 4- and 5-year olds have computer class and use the TI. With the speech synthesizer, the TI becomes a tool for very young children to use that is at their level of learning.

If you have an extra console and are looking far someone to use it, think about my experience and perhaps you can pass it along as easily as I did.

USER NOTES

(Continued from Page 29)

when you hear the drive head "seesawing" back and forth while loading a program or file. The worse the fracturing, the more seesawing you'll hear.

Optimizing is performed by using the file copy (option 1 of DM-1000 onto a clean disk. After copying your files to the copy disk, use the sweep disk option on the original disk.) Then copy the files from the copy disk back onto the original disk and presto, no fractured files.

This sounds like a lot of work, but it's all part of file maintenance.

Useful subroutines

Here are some subroutines by Jim Peterson that appeared in his original Nuts & Bolts disk.

CALL SLANT converts all uppercase letters, numerals and punctuation to a slanted form.

SLANT

```
90 CALL SLANT !236
91 FOR LOOP=1 TO 1000 :: NEX
T LOOP !014
100 PRINT "TEST" !167
20003 SUB SLANT !240
20004 DATA 33,00040408080800
1,35,0009127F24FE489,36,001F
24283E0A127C,40,000102040808
0804,41,002010101020408,43,0
00004083E081 !177
20005 DATA 48,001F1122222244
7C,49,0001010202020404,50,00
1F01023E20407C,51,001F01023E
02047C,52,001111223E020404,5
3,001F10203E02047C !250
20006 DATA 53,001F10203E2244
7C,54,001F10203E22447C,55,00
1F010202020404,56,001F11223E
22447C,57,003F21427E02047C,5
8,00000C0C001818 !244
20007 DATA 59,00000C0C001808
1,63,000F11020E08001,65,00
11223E224444,66,001E11213E22
427C,67,001F10202020407C,68,
001E11212122427C !197
20008 DATA 69,001F10203E2040
7C,70,001F10203E204040,71,00
1F10202E22447C,72,001111223E
224444,73,001F04080808107C,7
4,000101020222447C !031
20009 DATA 75,001111223C2844
44,76,001020202040407C,77,00
37294A52428484,78,0011112A2A
2A4444,79,001F11222222447C,8
0,001F11223E204040 !033
20010 DATA 81,001F1122222A44
7A,82,001F11223E284444,83,00
1F20203E02027C,84,001F040808
081010,85,001111222222447C,8
6,0011111222242438 !018
20011 DATA 87,002121424A5294
EC,88,0011110A0C142222,89,00
1111223E081010,90,001F020418
```

(See Page 31)

USER NOTES

(Continued from Page 30)

```
20407C !187
20012 RESTORE 20004 !207
20013 FOR CH=1 TO 45 !179
20014 READ CHN,CH$ !210
20015 CALL CHAR(CHN,CH$)!047
20016 NEXT CH !033
20018 SUBEND !168
```

CALL JIGGLE(M$,R,C) will display text at row R column C and "jiggle" it for attention until a key is pressed.

JIGGLE

```
20523 SUB JIGGLE(M$,R,C)!250
20524 FOR J=1 TO LEN(M$):: C
H=ASC(SEG$(M$,J,1)):: IF CH<
65 OR CH>90 THEN M2$=M2$&CHR
$(CH)ELSE M2$=M2$&CHR$(CH+32
)!088
20525 NEXT J !224
20526 DISPLAY AT(R,C):M$ ::
CALL KEY(0,K,S):: IF S<>0 TH
EN SUBEXIT !210
20527 DISPLAY AT(R,C):M2$ ::
GOTO 20526 :: SUBEND !024
```

CALL SLASHZERO will OPEN #1:"PIO" and initialize an Epson-compatible printer to print slashed zeroes.

SLASHZERO

```
20022 SUB SLASHZERO !045
20023 OPEN #1:"PIO" !253
20024 PRINT #1:CHR$(27);CHR$
(42);CHR$(0);!137
20025 PRINT #1:CHR$(27);CHR$
(42);CHR$(1);CHR$(48);CHR$(0
);CHR$(92);CHR$(34);CHR$(81)
;CHR$(8);CHR$(69);CHR$(2);CH
R$(65);CHR$(34);CHR$(28)!010
20026 PRINT #1:CHR$(27);CHR$
(36);CHR$(1);!141
20027 SUBEND !168
```

CALL PASSWORD lets you replace the asterisks in line 20233 with whatever password you choose, but type it in invisibly with the CTRL key held down. Start your program with CALL PASSWORD(CODE), RUN it. When the password is requested, type it in with the CTRL key down. The value of CODE will print out. Make a note of it, then delete line 20235. Now, in your program, give one or more key variables a value which is its true value multiplied by the value of CODE and divided by CODE. For example, if A=10 and CODE equals 720, write A=7200/CODE.

Now the program can't be run without the password. Users must type it in with CTRL. If they LIST the program to find it, they will find it invisible, and if they delete the password routine the program will not run correctly.

PASSWORD

```
20232 SUB PASSWORD(CODE)!108
20233 INPUT "PASSWORD? ":P$
:: IF P$<>"********" THEN 20
233 !071
20234 FOR J=1 TO LEN(P$):: C
=C+ASC(SEG$(P$,J,1)):: NEXT
J !138
20235 PRINT C :: BREAK !239
20236 SUBEND !168
```

CALL READSCREEN [CALL READSCREEN(P$)], where P$ is the name of the printer, will read the text on the screen (no graphics) and output it to the printer.

CALL READSCREEN

```
20601 SUB READSCREEN(P$):: O
PEN #1:P$ :: FOR R=1 TO 24 :
: FOR C=3 TO 30 :: CALL GCHA
R(R,C,G):: M$=M$&CHR$(G):: N
EXT C :: PRINT #1:M$ :: M$="
" :: NEXT R :: SUBEND !089
```

CALL SCREENGRID in line one will put a temporary grid on the screen to help in programming graphics.

CALL SCREENGRID

```
100 CALL SCREENGRID !085
20019 SUB SCREENGRID !089
20020 CALL CLEAR :: CALL CHA
R(32,"FF80808080808080"):: F
OR J=1 TO 32 :: C=C+1+(C=9)*
10 :: CALL HCHAR(24,J,48+C):
: IF J>24 THEN 20021 :: CALL
 VCHAR(J,14,48+C)!239
20021 NEXT J :: SUBEND !010
```

MICROpendium pays $10 for items used in this column that are sent in by readers. Send User Notes to MICROpendium User Notes, P.O. Box 1343, Round Rock, TX 78680; email jkoloen@io.com.

CLASSIFIED

SELLING USER GROUP SURPLUS Cartridges, utilities, consoles. Inquire about your needs. Merle Vogt, POB 145, Von Ormy, TX 78073 v12n11

HARDWARE FOR SALE

Smart One 2400X 2400baud modem with power supply (works with any TI modem cable), $40; Volksmodem 1200 baud modem (needs TI cable), $20; Signalman Mark XII 1200 baud modem (incl. TI cable, no docs), $20; Signalman III-TI 300 baud modem (incl. TI cable, no docs), $10; TI RS232 interface card (no docs), $35; Commodore 1702 color composite monitor, incl. cables for TI99/4A (measures 13" diagonally), $90; add $5 shipping for each item except monitor. Will split shipping cost of monitor. Call John Koloen, 512-255-1512.

The ONLY monthly devoted to the TI99/4A

Subscription Fees

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

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

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

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

Address Changes

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

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

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

Card No. _____

Expiration Date _____

(Minimum credit card order is $9)

Signature _____

(Required on credit card orders.)

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

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

Name _____

Address _____

City _____

State _____ **ZIP** _____

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

Disks, Etc.

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

_____

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

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

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

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

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

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

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

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

GENEVE DISKS (SSSD unless specified)

❑ **MDOS 2.21** (req. DSSD or larger (for floppy & hard drive systems) $4.00

❑ **GPL 1.5** $4.00

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

❑ **Myarc BASIC 3.0**.....$4.00

❑ **MY-Word V1.21** $4.00

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

GENEVE PUBLIC DOMAIN DISKS

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

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

SECOND CLASS