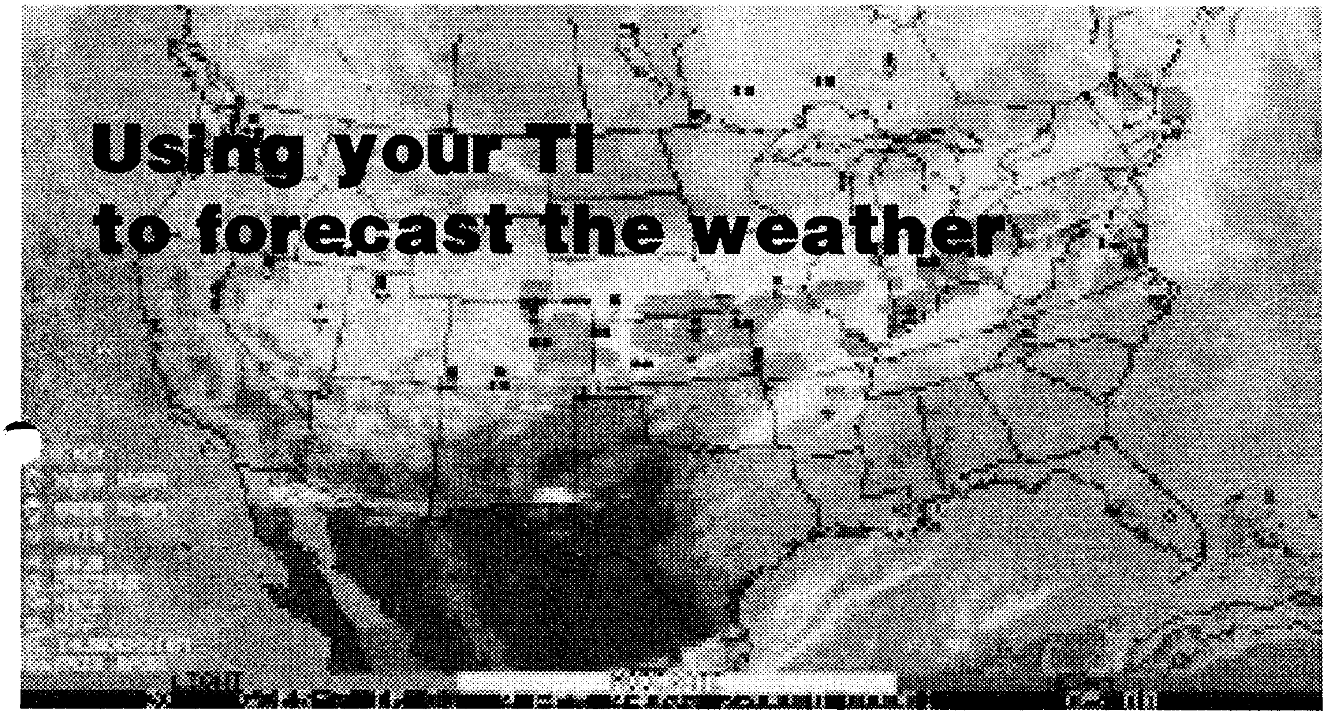
Covering the TI99/4A and the Myarc 9640

MICROpendium

Volume 11 Number 3 April 1994 $3.50



Using your TI to forecast the weather

Programming Funnelweb

The Art of Assembly

Labyrinth Inferno, a maze

Converting assembly into CALL LOADs

Reviews of CFORM, Game Writer's Toolkit, Brukin Golf and Widget

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.

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

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

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

Display advertising deadlines and rates are available upon request.

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

Foreign subscriptions are $40.25 (Mexico); $42.50 (Canada); $40.00, surface mail to other countries; $52 airmail to other countries.

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

Mailing address: P.O. Box 1343, Round Rock, TX 78680.
Telephone: (512) 255-1512
CompuServe: 75156,3270
Delphi TI NET: MICROpendium
GEnie: J.Koloen
Internet E-mail: jkoloen@indial.io.com
John Koloen..............................Publisher
Laura Burns..................................Editor

McCann software

Extended BASIC

The Art of Assembly

Conversions

Funnelweb

Reviews

Newsbytes

User Notes

Departments

*READ THIS

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

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

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

TEX✦COMP Earthquake Clearance Sale!

Our Northridge, California, warehouse is about five blocks from the center of the January 17 earthquake. While our building stayed up, it has taken us three weeks to pick everything up. If we survived 1983, when TI pulled the plug, we can certainly handle a 6.8 earthquake. There is no better way to see what you are overstocked on than when you have to pick up. To celebrate our survival, we have slashed prices.

There may never be a better time to buy! Hurry! Limited quantities.

Modules 99¢ each—NO LIMIT!

Subject to availability. Effective 3-1-94

PHM 3027	ADDITION & SUBTRACTION I
PHM 3028	ADDITION & SUBTRACTION II
PHM 3093	MILLIKEN DIVISION
PHM 3051	NUMERATION II
PHM 3094	MILLIKEN INTEGERS
PHM 3025	MIND CHALLENGER
PHM 3021	WEIGHT & NUTRITION
PHM 3002	EARLY LEARNING FUN
PHM 3003	BEGINNING GRAMMAR
PHM 3112	PARSEC
PHM 3007	HOUSEHOLD BUDGET MANAGEMENT
PHM 3044	PERS. REPORT GENERATOR**
PHM 3083	TI INVADERS
PHM 3194	JAWBREAKERS II
PHM 3016	TAX INVESTMENT RECORD KEEPING***
PHM 3022	PERSONAL REAL ESTATE
PHM 3023	HUNT THE WUMPUS
PHM 3031	THE ATTACK
PHM 3220	MICROSURGEON
PHM 3057	MUNCHMAN
PHM 3110	CHISHOLM TRAIL
PHM 3218	SUPER DEMON ATTACK
PHM 3006	HOME FINANCIAL DECISIONS
PHM 3035	TERMINAL EMULATOR II
PHM 3052	TOMBSTONE CITY
PHM 3109	LOGO II*
PHM 3055	EDITOR ASSEMBLER*
PHM 3010	PHYSICAL FITNESS

Plus $4 additional shipping & handling

Special Purchase

Personal Record Keeping
PHM 3013
$4⁹⁵ (Regular $9⁹⁵)

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

*ADD AN ADDITIONAL $4 .00 EACH ON THESE TITLES DUE TO WEIGHT
**FOR USE WITH PERSONAL RECORD KEEPING ***DISK DRIVE REQUIRED
PERSONAL RECORD KEEPING AVAILABLE AT $4.95

$4.95 Per Disk

SPECIAL Buy one, get one free Any disk or cassette below

EDUCATION

DISKETTE PROGRAMS

PHD 5009	Music Skills Trainer	4.95
PHD 5011	Computer Music Box	4 95
PHD 5018	Market Simulation	4 95
PHD 5030	Speak & Spell II (Ex Basic req.)	9 95
PHD 5031	Speak & Math (TE-II req.)	4 95
PHD 5042	Spell Writer (TE-II req.)	4 95
PHD 5026	Bridge Bidding I	4 95
PHD 5039	Bridge Bidding II	4 95
PHD 5041	Bridge Bidding III	4.95
SPECIAL!!	BRIDGE BIDDING I,II&III	9 95
PHD 5070	Music Maker Demo (use with module!)	4.95
PHD 6010	Mystery Melody (stop the music quiz)	4 95

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

PHT 6009	Music Skills Trainer	4 95
PHT 6010	Mystery Melody	4.95
PHT 6011	Computer Music Box	4 95
PHT 6018	Market Simulation	4.95
PHT 6031	Speak & Math	4.95
PHT 6042	Spell Writer	4 95
PHT 6026	Bridge Bidding I	4 95
PHT 6039	Bridge Bidding II	4 95
PHT 6041	Bridge Bidding III	4 95
SPECIAL!!	BRIDGE BIDDING I,II&III	9.95
PHT 6020	Music Maker Demo (use with module)	4 95

MANAGEMENT AND BUSINESS

DISKETTE PROGRAMS

PHD 5001	Mailing List (upgraded version)	4 95
PHD 5003	Personal Financial Aids	4 95
PHD 5021	Checkbook Manager	4.95
PHD 5022	Finance Manager	4 95
PHD 5024	Inventory Management	4 95
PHD 5027	Invoice Management	4 95
PHD 5029	Cash Management	4 95
PHD 5038	Lease/Purchase Decisions	4 95
PHD 5075	TI/Multiplan upgrade disk	4 95

CASSETTE PROGRAMS

PHT 6003	Personal Financial Aids	4 95
PHT 6038	Lease/Purchase Decisions	4 95

HOME ENTERTAINMENT

DISKETTE PROGRAMS

PHD 5002	TI-TREK (TE-II req. for speech)	4.95
PHD 5010	Mystery Melody	4 95
PHD 5015	Oldies But Goodies I	4 95
PHD 5017	Oldies But Goodies II	4 95
****SPECIAL	Oldies But Goodies I & II	7 95
PHD 5025	Sat Night Bingo (Ex-Basic & Speech)	4 95
PHD 5037	Draw Poker (Ex-Basic)	4.95

CASSETTE PROGRAMS

PHT 6002	TI-TREK (TEII req. for speech)	4 95
PHT 6010	Mystery Melody	4 95
PHT 6015	Oldies But Goodies I	4 95
PHT 6017	Oldies But Goodies II	4 95
****SPECIAL	Oldies But Goodies I & II	7.95
PHT 6026	Sat. Night Bingo (Ex-Basic & Speech)	4.95
PHT 6037	Draw Poker (Ex-Basic)	4.95

COMPUTER PROGRAMMING AIDS

DISKETTE PROGRAMS

PHD 5007	Teach Yourself 99/4A Basic	4.95
PHD 5019	Teach Yourself Extended Basic	4.95
PHD 5004	Programming Aids I	4.95
PHD 5005	Programming Aids II	4.95
PHD 5077	Programming Aids I,II,III	9 95
PHD 5067	Beginning Basic Tutor	4 95
PHD 5076	Text to Speech (Ex-Basic Speech)	4 95
PHD 5098	TI Forth & manual (Ed/Assem req)	19 95
PHD 5078	TI Forth Demo Disk (Ed/Assem)	4.95
PHD 5079	TI Forth Source Code (2 disks)	4.95

CASSETTE PROGRAMS

PHT 6006	Programming Aids I	4 95
PHT 6007	Teach Yourself 99/4A Basic	4 95
PHT 6019	Teach Yourself Extended Basic	4.95
PHT 6067	Beginning Basic Tutor	4.95

MATH AND ENGINEERING

specify disk or tape with order

Math Routine Library	4.95
Electrical Engineering Library	4.95
Graphing Package	4.95
Structural Engineering Library	4.95
AC Circuit Analysis	4.95

***SPECIAL--ALL 5 OF THE ABOVE ON DISK OR TAPE---17.95

99¢ BOOK BLOWOUT!!! NO LIMIT!

PUBLISHER'S CLEARANCE ON
BOOKS FOR THE 99/4A

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

*Add $3 for Games TIs Play disk or cassette
** Add $3 for disk or cassette for disk or cassette for Starter Pack Game Writer Series.

→ **SPECIAL**
Kids and the TI — $2⁹⁵
The TI in Bits and Bytes with disk — $4⁹⁵

SPECIAL PACKAGE AND PRICING ON Adventure

THE ADVENTURE SERIES...

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

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

NEW LOWER PRICE ONLY $4.95 FOR ALL 13

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

NEW LOWER PRICE ONLY $4.95 FOR ALL6

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

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

NEW First time on disk — Return to Pirate's Island with Graphics. XBASIC or Editor/Assembler Load **$4⁹⁵** Includes disk version of Editor/Assembler

ADVENTURE EDITOR

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

SUPER BUY!

$4.95

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

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

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

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

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

****

TERMS: **Add $3 S&H to all orders.** California orders add 8.25% sales tax. Canada, Hawaii and overseas shipping extra.

TEX✦COMP
P.O. Box 33084, Granada Hills, CA 91344

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

Send $2 for our catalog

AUTHORIZED DEALER

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

COMMENTS

Ah, the elusive TI99/5

You'll probably never see one, but what the heck — would'nt it have been nice if TI had come out with the TI99/5, if not the TI99/8. In the February Bugs and Bytes column we published an item that briefly outlined a controversy regarding the 99/5 that had appeared on the Internet and elsewhere. The focus of that controversy was on a so-called marketing plan for the 99/5 that allowed users to trade in their /4As for more money than they'd have to pay for a /5A.

Glen Bernasek, of the Cleveland Area TI99/4A User Group, published an article in the group's newsletter detailing some of the scuttlebutt about and photos of the /5. His information was provided by someone who actually owns one. Here are some exceprts:

The cassette port was replaced with a Hexbus port. The Hexbus was an eight-wire, four-bit wide communication cable. Devices designed for the Hexbus included a floppy disk controller, a serial port, streaming tape drive, 80-column video controller and portable printers, among other things.

The TI99/5 used a 9995 processor.

The TI99/5 had a 32K memory expansion and speech synthesizer built into the console.

The motherboard was imprinted with "99/5 11/2/83." TI was working on it right up to the end.

According to other sources, who posted information on the Internet, the 99/5 was also known as the 99/4B. It had not PEB port because it was meant only for use with hexbus peripherals. He also reported that TI will disavow any knowledge of the 99/5 or the 99/8 or virtually any other hardware or software meant for the home computer market that was never actually shipped. However, he said, when he asked whether he could distribute such software to his user group members TI prohibited it on the basis of holding a copyright to the software it wouldn't acknowledge existed.

Other items that were developed in small quantities for the home computer market include a hard drive controller, a GROM box similar to a GRAM Kracker, an IEEE interface card, a 128K Superram card, a 374K Ultraram card, a 4-channel music card, a PEB interface card for the 99/8 and an RS-232 card with four ports.

AD PRICES CUT IN HALF

Starting in May, and running for an indefinite period, MICROpendium display ads will be sold at half price. A restriction is that the ads must be paid in advance. Only advertisers who do not have an open balance may take advantage of these low prices. We'll be mailing information to most previous advertisers. Write to us if you want additional information.

NO MORE DAYTIME RESPONSE

Laura and I are now working full-time during the day so we won't be around to handle MICROpendium calls during the day. We will return calls in the evening and during the morning on Saturday. We'll be in the office Saturday mornings as well.

CREDIT CARDS

Credit cards are convenient for the buyer but for the seller they are becoming increasingly expensive to accept. We've been hit with several increases in costs charged to us to accept Visa and MasterCard. More than five percent of each credit card sale now goes to the credit card company. To help us out, if you can pay for your purchases from MICROpendium using a check or money order, please do so. Every nickel and dime we save helps.

NEWSLETTER EDITOR DIES

We have learned that Patrick Graham, newsletter editor of the North Bay (Canada) 99ers, has died. Over the years, we received various letters and notes from him, and always enjoyed his cheerful, upbeat attitude. Our condolences to his loved ones.

—JK

BUGS & BYTES

And TIs are forever

We were stirred (not shaken) to learn that Dave Connery of the Chicago TI Users Group is a cousin (OK, fifth cousin) of actor Sean Connery. After all, "You Only Live Twice," is an expression that could apply to the TI99/4A.

CA or CO?

We reprinted an announcement posted on Delphi last month which stated that the 1995 Fest West would be held in San Diego, California, but the latest word is that this has not been settled. Either San Diego or Denver, Colorado, could be the site for the fair.

FEEDBACK

TI group invites D-Day visitors

We are wondering if you could print the following note in the MICROpendium, so that TI99/4A users could contact us, DORTIG (Dorset TI User Group) if they should be in this area.

As 1994 is the 50th anniversary of "D" Day, many Americans, Canadians, etc.: ex-servicemen/ladies may be in our area (Bournemouth, Poole, Southampton) for this event.

Should they like to contact DORTIG during their stay, they could reach us via the following telephone numbers:

Mr. F.T. Taberner (secretary), 0202 880878; Mr J. Murphy, 0202 803043; both Bournemouth area codes.

Also, Mr. Taberner is a Radio Amateur and his call signs are GO(G ZERO) UGS or PACKET RADIO BBS GB7 BNH if any readers would care to contact him.

J. Murphy
Poole, England

For any of our younger readers who may not have been paying attention during history class, D-Day, June 6, 1944, was the day of the invasion of western Europe by Allied forces during World War II — Ed.

Original was right

In response to the letter by P.C. Van Nostrand (Feedback, February 1994), I would like to make note of the following: My original article appeared in the April 1993 issue of the SCCG Newsletter, The Computer Voice. It has since been reprinted in several newsletters. An unknown editor changed "H11" to "M1" and added the note in parenthesis. An updated version of my *original* article has been uploaded to Delphi.

James D. Lanman
San Diego, California

Praise for article

Re Steven L. Richardson's article on literary detection and Harold Hoyt's concerning Earth's population (February 1994), I was quite impressed with their content. Both of them had Biblical references in them that were most welcome, especially Steven's discussion on how the words of Christ, as penned by the four Gospels, were determined to be of a different writing style than of the Gospels themselves! I am pleased that MICROpendium had the "guts" to print these and I pray that you'll allow me to say here that Jesus is alive!

MICROpendium is a fabulous magazine that I enjoy each month. Please try to give us informative pieces like these frequently and keep on publishing MP!

The program I sent you a while back that played the theme from the"Quantum Leap" TV show has been uploaded to the bulletin boards Cactus Patch and Club 99 under the filename "QL" for downloading. Please inform your readers of this.

Brian C. Horner
Woodbury, New Jersey

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

READER TO READER

❑ Phil Van Nostrand, 430 Shadow Creek Dr., Seabrook, TX 77586, writes:

I acquired two orphan Hitachi half height disk drives (at least third hand) that use 3-inch disks (not 3.5-inch). I would like to have more than the one disk that came with them, but have not found anybody that had even heard of such an idem. The disk is labeled AMDISK III 3" Floppy Disk and is an Amdek product made in Japan. They are double-sided and can be flipped over to use the second side, since the drives are only single-sided. The drives are compatible with the TI cable and power supply and work fine at single or double density. Any info would be appreciated.

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

Companies carry all McCann programs

Dee Turner has announced that he is now distributing all programs formerly available from McCann Software. Contact him at 641 S. Laughlin Lane, Merced, CA 95340. Phone is (209) 722-7352 (home); (209) 722-2053 (work).

Also, Ron Markus of RamCharged Computers, adds a correction to Charles Good's column in the February 1994 MICROpendium. RamCharged does not carry just The Printer's Apprentice for MDOS from McCann Software. The company carries the whole McCann line as well as the Notung book, *How to Use the Printer's Apprentice and Not Go Insane*. Contact RamCharged at P.O. Box 81532, Cleveland, OH 44181, (216) 243-1244. or 1-800-669-1214.

The McCann line includes TPA for MDOS and TI, The Geometer's Apprentice, TPA fonts disks 1 and 2, Business Graphs 99 and TPA Toolbox.

voice assistance, call Delphi service representatives at 1-800-695-4005.

Labyrinth Inferno

A challenging Extended BASIC maze game

By LUCIE DORAIS

This article appeared in the newsletter of the Ottawa TI Users Group.

The Labyrinth Inferno is another game, again by Michel Montmigny, of the Sherbrooke, Quebec, SHER-TI group. It is short, but quite interesting. It is also rather hard to play.

When you start, a labyrinth slowly builds up on the screen. Study it carefully because, after it is all built, you only have a few more seconds to look at it before it disappears. A little square (you) then appears at the top of the screen. Your task is to bring it to the bottom row, trying to remember where the passages are. Since the labyrinth is made up entirely of empty squares with one bar at the top or one at the left, it means there are not many ways for you to reach the end. In some cases there is none. But you can always give up by pressing the "Q" key.

You navigate with the arrow keys — ESDX — or with joystick No. 1. If you bump into a wall, you hear a "bad" sound and the wall appears on the screen. Since I am not a joystick fan, I had transposed Michel's all-joystick routine to an all-keyboard one. Jeffrey Brown has designed a very clever routine that uses either joystick or keyboard at any time (lines 280-320).

When you reach the bottom row, or if you give up, the screen briefly flashes the complete labyrinth before telling you how many points you earned. I did not write the game so I am not sure how the points work, but they seem to work like this: you start with 630 points (line 270) and lose one each time you do a "bad" move (subroutine BAD). When you reach the bottom row, the remaining points are multiplied by 32, but this is not done if you pressed "Q" to give up (line 510). Your best REcord for that session is kept in memory.

```
100 ! LABYRINTH INFERNO / M.
 Montmigny, Sherbrooke UG, 1
985 / Adapted to XB by L. DO
RAIS, Ottawa UG, Dec. 1993 !
214
110 CALL CLEAR :: CALL SCREE
N(2):: CALL COLOR(10,15,1)::
 RE=0 :: RANDOMIZE !096
120 GOTO 140 :: K,KAR,KX,KY,
L$,M$,PT,S,T$,X,X1,XX,Y$,Y,Y
1,YY,Z,Z1 !136
130 CALL CHAR :: CALL HCHAR
:: CALL VCHAR :: CALL GCHAR
:: CALL CHARPAT :: CALL KEY
:: CALL JOYST :: !@P- !149
140 FOR X=1 TO 8 :: CALL COL
OR(X,16,1):: NEXT X :: DISPL
AY AT(1,4):"THE LABYRINTH IN
FERNO":TAB(7);"MICHEL MONTMI
GNY" !087
150 FOR X=11 TO 14 :: CALL C
OLOR(X,11,1):: NEXT X !218
160 CALL CHAR(104,"FFFFFFFFF
FFFFFFF"):: T$="FF" :: L$="8
080808080808080" :: M$="0000
18183C5A2424" :: CALL CHARPA
T(121,Y$)!209
170 DISPLAY AT(6,1):"WATCH T
HE LABYRINTH BUILD,": :"THEN
 DISAPPEAR..." !246
180 CALL CHAR(112,T$,113,L$,
120,M$,121,M$,128,T$,129,L$,
136,"FF0018183C5A2424",137,"
80809898BCDAA4A4")!001
190 DISPLAY AT(11,1):"THEN B
RING THE MAN ("&CHR$(121)&")
 TO": :"THE BOTTOM OF THE SC
REEN." !027
200 DISPLAY AT(16,2):"MOVE:
(ESDX) / JOYSTICK 1":" QUIT:
  (Q)   / FIRE BTN" !165
210 L$="*** ALPHA LOCK UP!!!
 ***" :: DISPLAY AT(22,3):L$
: :TAB(5);"ANY KEY TO START.
.." !238
220 DISPLAY AT(22,3):"" :: C
ALL KEY(0,K,S):: IF S=0 THEN
 DISPLAY AT(22,3):L$ :: GOTO
 220 !003
230 ! =====labyrinth===== !2
10
240 CALL CLEAR :: CALL HCHAR
(1,2,104,30):: CALL HCHAR(24
,2,104,30):: CALL VCHAR(1,2,
104,24):: CALL VCHAR(1,31,10
4,24)!196
250 FOR X=3 TO 23 :: FOR Y=3
 TO 30 :: CALL HCHAR(X,Y,111
+INT(2*RND)+1):: NEXT Y :: N
EXT X !223
260 FOR X=1 TO 500 :: NEXT X
 :: CALL COLOR(11,1,1)! look
, then black out !099
270 PT=630 :: X,X1=2 :: Y,Y1
=16 :: Z=32 :: CALL HCHAR(X,
Y,120)!153
280 CALL KEY(5,K,S):: KX=4*(
(K=120)-(K=101)):: KY=4*((K=
115)-(K=100))! check keyboar
d, generate JOYST like value
s !141
290 CALL JOYST(1,YY,XX):: IF
 KY<>0 OR KX<>0 THEN YY=KY :
: XX=KX ! compare if keyboar
d used (or joystick) !029
300 CALL KEY(1,K,S):: IF K=1
8 THEN 490 ! fire (Q)=give u
p !090
310 IF XX=0 AND YY=0 THEN 28
0 ! no move !248
320 X1=X :: Y1=Y :: XX=-XX/4
 :: YY=YY/4 :: CALL HCHAR(X,
Y,Z)!122
330 X=X+XX :: Y=Y+YY :: CALL
 GCHAR(X1,Y1,Z1)!006
340 IF XX=-1 THEN IF Z1=112
THEN 410 ELSE IF Z1=128 THEN
 420 !112
350 IF YY=-1 THEN IF Z1=113
THEN 410 ELSE IF Z1=129 THEN
 420 !116
360 CALL GCHAR(X,Y,Z):: IF Z
1=32 THEN IF Z=112 THEN 450
ELSE IF Z=128 THEN 460 !219
370 IF Z=32 THEN KAR=120 ::
GOTO 470 ELSE IF Z=104 THEN
460 ! space/border !060
380 IF XX=1 THEN IF Z=112 TH
EN 430 ELSE IF Z=128 THEN 44
0 !116
390 IF YY=1 THEN IF Z=113 TH
EN 430 ELSE IF Z=129 THEN 44
0 !120
400 CALL HCHAR(X,Y,Z+8):: IF
 X=23 THEN 490 ELSE 280 ! 1
st row or not !195
410 Z=Z+16 :: CALL BAD(PT)!0
```

(See Page 7)

LABYRINTH—

(Continued from Page 7)

```
38
420 X=X1 :: Y=Y1 :: GOTO 400
 !036
430 CALL HCHAR(X,Y,Z+16):: C
ALL BAD(PT)!098
440 X=X1 :: Y=Y1 :: Z=Z1 ::
GOTO 400 !073
450 CALL BAD(PT):: CALL HCHA
R(X,Y,128)!124
460 Z=Z1 :: X=X1 :: Y=Y1 ::
IF Z=128 OR Z=129 THEN KAR=Z
+8 ELSE KAR=120 !135
470 CALL HCHAR(X,Y,KAR):: GO
TO 280 !028
480 ! ===== give up / did it
 ===== !146
490 FOR X=1 TO 15 :: CALL CO
LOR(11,1,1):: CALL SOUND(-15
0,-7,0):: CALL COLOR(11,11,1
):: NEXT X !001
500 FOR X=1 TO 800 :: NEXT X
 :: CALL CLEAR :: CALL CHAR(
121,Y$)!120
510 IF K<>18 THEN PT=PT*32 !
041
520 IF PT>RE THEN RE=PT !040
530 DISPLAY AT(5,2):"POINTS:
 ";STR$(PT): :" RECORD: ";ST
R$(RE): : : :"ANOTHER LABYRI
NTH? (Y/N) y" !190
540 ACCEPT AT(11,27)SIZE(-1)
VALIDATE("YyNn")BEEP:T$ !006
550 IF T$="y" OR T$="Y" THEN
 CALL CHAR(121,M$):: GOTO 24
0 ELSE END !000
560 !@P+ !062
570 SUB BAD(PT):: CALL SOUND
(100,-2,0):: PT=PT-1 :: SUBE
ND !012
```

The instructions appear while the characters are redefined (lines 140-210) and the Alpha Lock reminder is flashed while Tex waits for a key (lines 220). A grey border is then drawn (line 240), followed by the labyrinth itself, which will be made up of characters 112 (T$, top bar) or 113 (L$, left bar) chosen at random. Line 260 is quite important: you can modify the FOR-NEXT delay (before the labyrinth blacks out) to whatever length you want. More importantly, while debugging you can use another COLOR than "1" (transparent) so that you can see how the program works. Try to use any other color than 11, the yellow used by the program itself for the walls revealed when you bump into them.

The starting row and column of the man are punt into X/X1 and Y/Y1 and the starting default character Z is initialized to a space. Z always holds the character that will be put back on screen after a move.

The main CALL KEYS for keyboard and joystick are then read, with a third one to read the "Q" key from either keyboard. Thus the CALL KEY(1...) or joystick (lines 280-300). Lines 310 to 470 deal with the move itself. First put back the character that was there before (Z, line 320), then find out where the man is going and check the character there (GCHAR, line 320). There are many cases here, and I leave you the fun of understanding them, remembering that the "blacked out" squares are characters 112 and 113, while their "uncovered" counterparts (they have already been bumped into, thus revealed) are 128 and 129. The man can be characters 120 and 121 because he is called by a "Z+8" statement (lines 400 and 460). Characters 136 and 137 are used temporarily when the man is occupying a square with a top bar or a left one.

The give-up or ending routine starts at line 490. It flashes the labyrinth 15 times, then lets you look a bit at the complete thing before erasing the screen (line 500) and showing you your point and record (lines 510-530). In line 500, the CALL CHAR(121,Y$) reverts character 121 to an alpha "y" so you can say "yes I want to play another game!" (line 540; was CALL CHARPATed in line 160). We have to use a lowercase letter because the Alpha Lock key has to be up to let you use the joystick. Don't worry, in line 280 the CALL KEY looks for the lowercase values of "ESDX."

THE ART OF ASSEMBLY — PART 34

Time to cram

By BRUCE HARRISON

It's getting near "final exam" time for our beginner students of Assembly Language, so we're going to "cram" in this month's installment. We'll try to cover as much remaining ground as possible in one session. We'll start by trying to cover a few more important instructions.

IMMEDIATES GALORE

Back in the first part of this mini-series for beginners, we showed examples of two "immediate" instructions, namely Load mediate (LI) and Load Worspace Pointer Immediate (LWPI). The word "immediate" has nothing in particular to do with suddenness of the operation, but simply means that the word in memory that immediately follows the instruction is to be used as data in performing the instruction. Thus LI R2,27 means load register 2 with the next word in memory, which in this case contains the number 27. As we showed in that column, the immediate value itself can be represented in source code by an expression, which the Assembler computes into a single word value.

There are some other important "immediate" instructions, all of which involve a register as the first operand, with an immediate value as the second. In most of these instructions, the usual order of operands for TI Assembly is reversed, in that the first operand (register) is the Destination, and the second (Immediate Value) is the Source. We'll cover the exception in a couple of moments, but first let's simply list all the "Immediate" instructions.

LWPI Load Workspace Pointer Immediate

(See Page 8)

THE ART OF ASSEMBLY—

(Continued from Page 7)

LI Load a register with Immediate value
AI Add an Immediate value to a register
ANDI AND the register with an Immediate value ORI OR the register with an Immediate value
CI Compare the register to an Immediate Value

Add Immediate does exactly what its name implies. If register 2 held a value of 45, and we perform the instruction AI R2,15, then R2 will contain 60 after the operation. Note that there is no such thing as "subtract immediate", but you can make a subtraction happen by placing a minus sign before the immediate value. Thus performing AI R2,-15 in the above case would make R2 become 30.

ANDing an immediate value is a logical operation which is usually used to isolate particular bits in the register for examination. An easy example is hard to come by, but just suppose that it's important to know whether the number in R2 is an odd or an even number. Odd numbers in Binary always have a 1 as their least significant bit, while even numbers always have their LSB set at 0. (Trust us that this is so.) The easy way to tell about the number in Register 2, then, would be simply to ANDI R2,1. The immediate value 1 has only its least significant bit set to 1. (In binary, that's 00000000 00000001.) Thus when an AND operation is performed, R2 will contain 1 if and only if its LSB was 1 to start with. If it was an even number, the result of ANDI R2,1 will be zero. The test could look like this in source code:

```
     ANDI R2,1
     JEQ  EVEN if result is zero, jump
     (else it was an odd number)
EVEN  (do whatever follows for an even number)
```

If the value in R2 were important to begin with as such, then one should move that value to someplace else before the ANDI instruction, because only the LSB of R2's content will survive this operation.

The OR Immediate does effectively the opposite of ANDI, in that it will force a 1 into each bit of the destination register wherever there's a 1 in the Immediate value. Let's take a similar situation involving R2, but this time we want to make sure that the number in R2 is forced to be odd. We can do this by the simple instruction ORI R2,1. If R2 were already odd, this would have no effect, and its value would be the same as before the operation. If, however, it were even, its value would be incremented by one, making it an odd number.

These are not the most common uses for ANDI and ORI, but we chose to use these illustrations because they are simple enough to give the beginner a feel for the instructions.

The final Immediate instruction, Compare Immediate, is the one exception to the unusual Destination,Source relationship for Immediate instructions. If for example, we perform an operation like this:

```
          CI  R8,78
          JGT BIG
```

We will jump to BIG if the value in Register 8 is larger than the immediate value 78. As with all Compare operations, neither Operand is changed by the comparison. (The microprocessor performs a subtraction internally, but does not "output" the results except into the Status Register.)

A SHIFTY BUSINESS

Way back when, those special memory words that we call Registers were called by the name Shift Registers. One of the operations that is exclusive to the registers in the TI is the series of instructions called Shift Instructions. These are hard to visualize, even in Hex numbers, and virtually impossible to understand in decimal numbers, since they are operations that happen to the bits in the registers. Perhaps then you'll forgive us if we move back for a minute or two into that mysterious world of binary numbers.

Let's make this as easy as possible by starting with a register that contains the value 1. In binary, the content of the register would be 00000000 00000001. Now we can perform some shift operations on this register (let's say it's Register 2). For openers, let's shift it left by one bit. That's done by the source statement SLA R2,1. The result will be that the single one bit in the register will move over one position to the left, and its original place will become a zero. It would then look like this in binary: 00000000 00000010. The effect in this case is that the register that contained 1 now contains 2. Within limits, the SLA instruction can

(See Page 9)

THE ART OF ASSEMBLY—

(Continued from Page 8)

be used as a multiply operation, multiplying by two for each bit position that the register is shifted to the left. If the situation above were SLA R2,4, then the register that contained 1 would contain 16.

Before getting too involved, let's try an opposite case, where we shift to the right instead of to the left. If we take the register as starting with 16, then SRL R2,4, we wind up with the value 1 in the register.

The menu for shift instructions goes like this, with left shifts consisting of just one item, while right shifts have three choices:

SLA Shift Left Arithmetic
SRL Shift Right Logical
SRA Shift Right Arithmetic SRC Shift Right Circular

The difference between Logical and Arithmetic shifts involves what happens to the Most Significant Bit in the register, known also as the Sign Bit.

In integer arithmetic operations, any quantity in which the leftmost bit is a one is treated as a negative number. Let's say that a register contains 10000000 00000000 in binary. That's a negative number (8000 in Hex, -32768 in decimal) to the computer. Similarly, the number expressed in binary as 11111111 11111111 is also negative (-1 in decimal, or FFFF in hex). If we simply change that leftmost bit to a zero, as 01111111 11111111 it becomes the positive number 32767. If we do a SRL operation, the leftmost bit will be replaced by zero regardless of what its previous value was. If, on the other hand, we SRA, the leftmost bit will remain whatever it was, so that a negative number will stay negative after the shift is performed. In other words, for an Arithmetic shift, the sign of the number is preserved, while for a Logical shift, it can be changed.

That leaves only SRC to discuss. Shift Right Circular means that when a bit is shifted out on the right side, it "circles" back to the left side. Let's take our old friend R2, load it with just 1, and then do an SRC R2,1 operation on it. Before, it has 00000000 00000001, and after this operation it will have 10000000 00000000. In decimal values, that one instruction changed 1 to -32768. Quite a shift!

We'll leave this subject here, because there's lots more ground to cover, and there are excellent descriptions of what happens in these shift instructions in the E/A Manual.

OTHER INSTRUCTIONS

As a quick check in the Assembly Manual will reveal, there are a lot of instructions we haven't touched. Some are very simple, like the instruction CLR, which has only one operand. CLR makes its one operand equal zero, which can come in very handy indeed. The operand can be any memory location or a register. Other handy one-operand instructions are INC and INCT, DEC and DECT. These add or subtract one or two from their destination operands.

Once you've started playing around with your own Assembly programs, and gotten some things working, you'll find that the explanations of the instructions as given in the Manual will start to make sense, and you'll be able to apply them for yourself.

ADDRESSING MODES

In most of the lessons in this mini-series, we have stuck to the simplest modes of addressing memory. The TI has a number of very interesting and useful ways to address memory. Here's a brief summary and a couple of examples.

The addressing modes are these:

Direct addressing
Register indirect addressing
Register indirect auto-increment
Indexed

Direct addressing simply means that the operand named in the instruction is the address for the source or destination. This can be either a register or any location in memory. For example, we can write MOVB R1,@PABDT+1. This will move the high order byte from R1 into the memory location one byte past the label PABDT.

Register indirect addressing means that the register's contents are to be used as the address. For example, if we write MOV *R9,*R10, the asterisk tells the Assembler that whatever numbers are in R9 and R10 are addresses for the source and destination operands, and the registers themselves will not be changed by this operation.

In addition to the asterisk in front of the register numbers, we can add a plus (+) sign after the register to cause the number in the

(See Page 10)

THE ART OF ASSEMBLY—

(Continued from Page 9)

register to auto-increment after the operation. (e.g. MOV *R9+,*R10+) In the case just cited the plus would mean that after moving the word from the location pointed to by R9 into the word pointed to by R10, both registers would be incremented by two, so they'd point at the next word. For MOVB operations, the auto-increment will advance the values in the registers by only one, so they'll point at the next byte in memory.

Finally, there's indexed addressing. This uses a combination of direct addresses and registers. Suppose there's a lookup table, at label LUT, and R1 contains the member of the table we want to access. We can write MOVB @LUT(R1),R2. This will add the value in R1 to the address LUT, then move the byte at that location into the high byte of R2. This mode can be a very powerful means of accessing arrays of data in memory.

All these addressing modes have been used in our own Assembly work, and they've all had their advantages depending on what we were trying to do. We trust that you too will come to have great respect for the designers who built all this power into that tiny little computer chip.

FINAL EXAMS

Sorry, but there will be no final exams. This headline was just to get your attention. If you're reading this, it worked, so we can try to start wrapping up our beginners' mini-series. As you all knew at the start, this introductory material would not create "instant assembly programmers" from neophytes. 5Its purpose is to give you all a "feel" for the language, and in some degree to get you ready to tackle the "big book", which can be very confusing at times. Also, those of you who've kept all your back issues of MICROpendium can go through some of the earlier parts of this series and perhaps understand them this time. If you have MICROpendium on disk, you'll find many useful "snippets" oı source code in the Sidebars to these articles. Using those, either "as is" or just as examples should move you well along toward becoming an Assembly programmer.

PARTING THOUGHTS

During this mini-series, we have made some very general comparisons between various operations as represented in Assembly and "similar" operations in Basic. The similarities are mostly superficial. When we perform an operation like A = A + B in Basic, what really happens is that two eight-byte floating point numeric quantities are added, and the resulting eight byte floating point number is stored in memory by Basic at the location reserved for the variable A. In our Assembly "equivalent", A @B,@A the quantities being added are one-word integers, with a range of values limited to -32768 through +32767. In the Basics, the statements placed in the program need to be interpreted by the computer, and even a very small statement like that above results in the execution of hundreds of machine instructions to first interpret and then execute the desired operation. This interpretation that takes place when Basic is running is both a strength and a weakness. It's a strength because of the simplicity for the programmer, but the need to interpret and then perform all those steps makes the operations slower by many fold when compared to similar operations coded in Assembly.

We hope that at least some of you will "stay the course" learning Assembly. There's an enormous effort involved to become proficient in this language, and even your author, who's been doing Assembly programming for some years on both TI and PC computers, still has things to learn. The work is difficult at times, but the reward, in having programs that do "impossible" things at incredible speed, makes it all worth the pain.

Extended BASIC

Forecaster helps predict weather

Predicting the weather is more an art than a science. Just ask any weatherman. The following program, by Gary Cox puts a little science into the process for TI users. The program, which is very large, requires Extended BASIC and a memory expansion.

Users input information based on prompts and the program then calculates the type of weather to expect.

```
100 ! WEATHER FORCASTER (C)
!048
110 ! (C) 1985 !046
120 ! BY Gary Cox !059
130 !    3174 MELBOURNE !155
140 !    MEMPHIS, TN 38127 !
041
150 !!131
160 !    901-358-0667 !106
170 !!131
180 ! ====FREEWARE==== !220
190 ! Read lines 5560-5780 !
075
200 ! for details or run the
 !199
210 ! program. thank you....
 !052
290 ! CURSOR CHANGE !071
300 CALL INIT !157
310 CALL LOAD(8196,63,248)!1
56
320 CALL LOAD(16376,67,85,82
,83,79,82,48,8)!222
330 CALL LOAD(12288,48,48,63
,255,254,124,24,12)!139
340 CALL LOAD(12296,2,0,3,24
0,2,1,48,0,2,2,0,8,4,32,32,3
6,4,91)!129
350 CALL LINK("CURSOR")!175
360 GOSUB 5150 !130
370 CALL COLOR(0,2,1)!170
380 CALL CLEAR :: DISPLAY AT
(5,7):"Weather Forecaster" !
143
390 DISPLAY AT(10,3):"Do you
 need instructions":"  Y/N"
:: ACCEPT AT(11,7)VALIDATE("
```

(See Page 11)

FORECASTER—

(Continued from Page 10)

```
YN")BEEP SIZE(1):A$ !220
400 RESTORE :: IF A$="Y" THE
N GOSUB 5560 !171
410 DIM A(8,2),B(8,2),C(8,2)
,D(8,2),E(8,2),F(8,2)!070
420 CALL CLEAR !209
430 DISPLAY AT(2,7):"LOCATIO
N MENU" !228
440 DISPLAY AT(4,2):"1 SOUTH
EAST STATES" !061
450 DISPLAY AT(5,4):"(FLA,GA
,ALA,N.C.,S.C." !030
460 DISPLAY AT(6,4):"MISS,&T
ENN.)" !080
470 DISPLAY AT(8,2):"2 SOUTH
 CENTRAL STATES" !066
480 DISPLAY AT(9,4):"(TEXAS,
 OKLA, RK,LA." !049
490 DISPLAY AT(10,4):"AND EA
STERN N.MEX.)" !075
500 DISPLAY AT(12,2):"3 SOUT
HWEST STATES" !133
510 DISPLAY AT(13,4):"(CAL.,
NEV.,UTAH,ARIZ." !197
520 DISPLAY AT(14,4):"WESTER
N PARTS COLO.&N.M.)" !006
530 DISPLAY AT(16,2):"4 NORT
H CENTRAL STATES" !108
540 DISPLAY AT(17,4):"(MINN.
,IOWA,MO.,N.D.,S.D." !137
550 DISPLAY AT(18,4):"NEB.,K
ANS.,EASTERN PARTS" !232
560 DISPLAY AT(19,4):"OF MON
T.,WYO,&COLO.)" !149
570 DISPLAY AT(21,2):"5 NORT
HWEST STATES" !127
580 DISPLAY AT(22,4):"WASH.,
OREG.,IDAHO," !016
590 DISPLAY AT(23,4):"WESTER
N PARTS MONT.&WYO.)" !077
600 CALL KEY(0,G,H)!172
610 DISPLAY AT(24,8):"(press
 any key)" !189
620 DISPLAY AT(24,2):"
                   " !212
630 IF H=0 THEN 600 !086
640 CALL CLEAR !209
650 DISPLAY AT(3,2):"6 NORTH
EAST STATES" !057
660 DISPLAY AT(4,4):"(NEW EN
GLAND,N.J.DEL.,MD." !122
670 DISPLAY AT(5,4):"VA.,EAS
TERN PARTS OF N.Y." !199
680 DISPLAY AT(6,4):"& PA.)"
 !094
690 DISPLAY AT(8,2):"7 GR. L
AKES-MIDWEST" !053
700 DISPLAY AT(9,4):"(ILL.,I
ND.,OHIO.,WISC." !200
710 DISPLAY AT(10,4):"MICH.,
KY.,W.VA.,WESTERN" !132
720 DISPLAY AT(11,4):"PARTS
OF N.Y. & PA.)" !027
730 DISPLAY AT(19,2):"(selec
t 1-7 or press any" !136
740 DISPLAY AT(20,2):"any ot
her key for previous" !102
750 DISPLAY AT(21,2):"menu.)
" !104
760 CALL KEY(0,G,H)!172
770 IF H=0 THEN 760 !246
780 CALL CLEAR !209
790 IF G<49 OR G>55 THEN 420
 !212
800 IF G=50 THEN RESTORE 226
0 !161
810 IF G=51 THEN RESTORE 239
0 !037
820 IF G=52 THEN RESTORE 252
0 !168
830 IF G=53 THEN RESTORE 264
0 !034
840 IF G=54 THEN RESTORE 277
0 !165
850 IF G=55 THEN RESTORE 289
0 !031
860 ! THICKENING CIRRUS !127
870 DISPLAY AT(10,2):"PLEASE
 WAIT WHILE I READ" !185
880 DISPLAY AT(11,2):"DATA..
.." !047
890 FOR I=1 TO 8 !063
900 READ J,K !223
910 A(I,1)=J !172
920 A(I,2)=K !174
930 NEXT I !223
940 ! ATTOCUM. W/CIRROCUM. !
038
950 FOR I=1 TO 8 !063
960 READ J,K !223
970 B(I,1)=J !173
980 B(I,2)=K !175
990 NEXT I !223
1000 ! LOWERING ALTOSTRATUS
!144
1010 FOR I=1 TO 8 !063
1020 READ J,K !223
1030 C(I,1)=J !174
1040 C(I,2)=K !176
1050 NEXT I !223
1060 ! DEVELOPING CUMULUS !2
22
1070 FOR I=1 TO 8 !063
1080 READ J,K !223
1090 D(I,1)=J !175
1100 D(I,2)=K !177
1110 NEXT I !223
1120 ! STRATOCUMULUS !174
1130 FOR I=1 TO 8 !063
1140 READ J,K !223
1150 E(I,1)=J !176
1160 E(I,2)=K !178
1170 NEXT I !223
1180 ! NIMBOSTRATUS !078
1190 FOR I=1 TO 8 !063
1200 READ J,K !223
1210 F(I,1)=J !177
1220 F(I,2)=K !179
1230 NEXT I !223
1240 ! COMPUTATIONS !073
1250 CALL CLEAR !209
1260 DISPLAY AT(3,10):"TYPE
OF CLOUDS" !068
1270 DISPLAY AT(5,2):"1 CIRR
US" !088
1280 DISPLAY AT(6,2):"2 ALTO
CUMULUS" !229
1290 DISPLAY AT(7,2):"3 ALTR
OSTRATUS" !066
1300 DISPLAY AT(8,2):"4 CUMU
LUS" !181
1310 DISPLAY AT(9,2):"5 STRA
TOCUMULUS" !154
1320 DISPLAY AT(10,2):"6 NIM
BOSTRATUS" !099
1330 DISPLAY AT(11,2):"7 LES
S THAN 30% CLOUDS" !214
1340 DISPLAY AT(12,2):"8 FOR
 HELP" !232
1350 ACCEPT AT(23,1)VALIDATE
("012345678")BEEP SIZE(1):L
!045
1360 IF L=8 THEN GOTO 5310 !
096
1370 IF L=0 THEN 420 !165
1380 CALL CLEAR !209
1390 DISPLAY AT(3,10):"MONTH
" !224
1400 DISPLAY AT(9,2):"1 MAY
THROUGH OCTOBER":" 2 NOVEMBE
R THROUGH APRIL" !037
1410 ACCEPT AT(23,1)VALIDATE
("012")BEEP SIZE(1):M !231
```

(See Page 12)

FORECASTER—

(Continued from Page 11)

```
1420 IF M=0 THEN 1250 !231
1430 CALL CLEAR !209
1440 DISPLAY AT(3,8):"WIND D
IRECTION" !044
1450 DISPLAY AT(4,3):"(the d
irection the wind)" !229
1460 DISPLAY AT(5,6):"(is co
mming from.)" !118
1470 DISPLAY AT(8,2):"1 NORT
H" !013
1480 DISPLAY AT(9,2):"2 NORT
H EAST" !097
1490 DISPLAY AT(10,2):"3 EAS
T" !218
1500 DISPLAY AT(11,2):"4 SOU
TH EAST" !149
1510 DISPLAY AT(12,2):"5 SOU
TH" !069
1520 DISPLAY AT(13,2):"6 SOU
TH WEST" !175
1530 DISPLAY AT(14,2):"7 WES
T" !248
1540 DISPLAY AT(15,2):"8 NOR
TH WEST" !171
1550 ACCEPT AT(23,1)VALIDATE
("012345678")BEEP SIZE(1):N
!047
1560 IF N=0 THEN 1380 !107
1570 CALL CLEAR :: DISPLAY A
T(3,9):"WIND SPEED" :: PRINT
 "WIND SPEED IN MPH" :: ACCE
PT AT(23,19)VALIDATE(DIGIT)B
EEP:O !160
1580 CALL CLEAR !209
1590 IF L=1 THEN P=A(N,M)!06
6
1600 IF L=2 THEN P=B(N,M)!06
8
1610 IF L=3 THEN P=C(N,M)!07
0
1620 IF L=4 THEN P=D(N,M)!07
2
1630 IF L=5 THEN P=E(N,M)!07
4
1640 IF L=6 THEN P=F(N,M)!07
6
1650 CALL CLEAR :: DISPLAY A
T(15,1):"READING SOME MORE D
ATA":"PLEASE WAIT..." !191
1660 GOTO 3030 !048
1670 ! TEMP !217
1680 CALL CLEAR !209
1690 DISPLAY AT(3,10):"TEMPE
RATURE" !174
1700 PRINT "TEMPERATURE IN D
EGREES" :: ACCEPT AT(23,24)B
EEP:Q !038
1710 ! BARAMETRIC PRESSURE !
022
1720 CALL CLEAR !209
1730 DISPLAY AT(10,1):"DO YO
U HAVE ANY BARAMETRIC":"PRES
SURE READINGS FROM":"YESTERD
AY AND TODAY? " !109
1740 ACCEPT AT(12,22)VALIDAT
E("YN0")BEEP SIZE(1):B$ :: C
ALL CLEAR !201
1750 IF B$="0" THEN 1430 !18
0
1760 IF B$="N" THEN 1840 !11
0
1770 CALL CLEAR :: DISPLAY A
T(2,2):"BARAMETRIC PRESSURE"
 :: DISPLAY AT(19,2):"A PRES
SURE READING FROM ":" YESTER
DAY" !188
1780 ACCEPT AT(20,13)BEEP:R
!003
1790 IF R=0 THEN 1670 !146
1800 DISPLAY AT(22,2):"TODAY
S READING" :: ACCEPT AT(22,1
7)BEEP:S :: CALL CLEAR !050
1810 IF S=0 THEN 1670 !147
1820 T=ABS(R-S):: IF T>1.00
THEN U=2 :: IF T>1.00 THEN 1
840 :: IF T>.50 THEN U=3 ::
IF T>.50 THEN 1840 !200
1830 IF T>.20 THEN U=4 ELSE
U=5 :: IF R>S THEN V=2 ELSE
V=1 !254
1840 ! WIND CHILL !097
1850 W=.0817*((3.71*SQR(O))+
5.81-(.25*O))*(Q-91.4)+91.4
:: IF O<4 THEN W=X :: W=INT(
W+.5)!216
1870 IF R<S THEN V=1 ELSE V=
2 !054
1880 IF V=1 THEN C$="RISING"
 ELSE C$="FALLING" !113
1890 IF (V=1)*(U=2)THEN D$="
CLEARING & COOLER" !090
1900 IF (V=1)*(U=3)THEN D$="
SKYS BECOMING CLEAR" !035
1910 IF (V=1)*(U=4)THEN D$="
BETTER WEATHER COMING" !196
1920 IF (V=2)*(U=2)THEN D$="
THUNDERSTORMS LIKELY!" !253
1930 IF (V=2)*(U=3)THEN D$="
RAIN LIKELY" !251
1940 IF (V=2)*(U=4)THEN D$="
SHOWERS POSSIBLE" !153
1950 IF U=5 OR U=0 THEN D$="
LITTLE CHANGE" !153
1960 IF R=S THEN C$="STEADY"
 !083
1970 CALL CLEAR :: DISPLAY A
T(9,2):"Do you know the HUMI
DITY?" :: ACCEPT AT(9,28)VAL
IDATE("YN0")BEEP SIZE(1):E$
!169
1980 IF E$="N" THEN 2030 !04
7
1990 IF E$="0" THEN 1720 !21
8
2000 CALL CLEAR :: DISPLAY A
T(9,2):"HUMIDITY=" :: ACCEPT
 AT(9,12)BEEP:Y !146
2010 IF Y=0 THEN 1720 !203
2020 CALL HUMID(Q,Y,DPT,HUMI
TURE,CLB)!240
2030 GOTO 4520 !008
2040 DATA 2,72,64,71 !064
2050 DATA 63,70,63,70 !116
2060 DATA 63,76,63,76 !128
2070 DATA 1,4,2,3 !154
2080 DATA 15,20,63,68,63,66
!066
2090 DATA 61,66,61,66,62,77
!074
2100 DATA 78,18,10,20 !111
2110 ! ALTOSTRATUS !009
2120 DATA 10,20,60,68 !107
2130 DATA 63,67,63,66 !127
2140 DATA 63,66,60,18 !120
2150 DATA 60,18,10,20 !102
2160 ! CUMULUS !209
2170 DATA 11,19,14,52 !108
2180 DATA 13,52,80,53 !111
2190 DATA 80,49,40,47 !120
2200 DATA 40,49,11,49 !116
2210 ! STRATOCUMULUS !174
2220 DATA 11,18,12,65 !109
2230 DATA 12,45,21,44,12,44,
11,17,11,17 !184
2240 ! NIMBOSTRATUS !078
2250 DATA 10,139,140,141,140
,141,31,142,140,141,140,141,
147,85,15,20 !049
2260 ! SOUTH CENTRAL ST. !08
4
2270 ! CIRRUS !123
2280 DATA 23,23,23,23,23,69,
69,69,73,69,23,28,23,23,23,2
```

(See Page 13)

FORECASTER—

(Continued from Page 12)

```
3 !050
2290 ! ALTOCUMULUS !001
2300 DATA 89,20,61,18,89,89,
61,53,61,73,23,25,23,26,23,2
0 !053
2310 ! ALTOSTRATUS !009
2320 DATA 89,18,61,89,61,51,
61,45,12,45,23,27,23,23,23,1
0 !041
2330 ! CUMULUS !209
2340 DATA 12,20,61,39,82,42,
81,42,61,42,47,92,47,26,10,2
0 !032
2350 ! STRATOCUMULUS !174
2360 DATA 89,66.1,89,93,71,4
2,93,104,93,61,43,25,12,26,2
2,20 !205
2370 ! NIMBOSTRATUS !078
2380 DATA 30,18,51,132,33,12
5,33,132,35,133,51,47,51,26,
30,20 !217
2390 ! SOUTHWEST STATES !109
2400 ! CIRRUS !123
2410 DATA 10,27,64,27,29,91,
9,91,64,63,89,23,23,23,3,23
 !001
2420 ! ALTOCUMULUS !001
2430 DATA 10,20,86,59,29,59,
29,112,111,112,110,106,89,24
,10,20 !022
2440 ! ALTOSTRATUS !009
2450 DATA 23,77,29,77,29,47,
29,70,76,31,90,90,23,90,23,9
0 !067
2460 ! CUMULUS !209
2470 DATA 23,20,23,20,100,11
3,101,102,101,61,100,61,77,2
7,22,20 !029
2480 ! STRATOCUMULUS !174
2490 DATA 23,10,23,18,23,61,
77,61,89,61,89,61,12,96,15,9
6 !060
2500 ! NIMBOSTRATUS !078
2510 DATA 10,126,85,33,84,35
,54,127,54,127,54,127,84,47,
10,47 !001
2520 ! NORTH CENTRAL STATES
!075
2530 DATA 23,59,23,59,23,135
79,53,79,95,23,95,23,95,23,
3 !072
2540 ! ALTOCUMULUS !001
2550 DATA 90,94,90,136,90,13
6,43,61,43,61,12,71,12,26,23
```

```
,20 !135
2560 ! ALTROSTRATUS !091
2570 DATA 12,94,89,130,63,13
0,63,120,63,120,12,120,12,12
0,23,20 !051
2580 ! CUMULUS !209
2590 DATA 2,20,40,120,40,120
,81,120,81,53,89,12,89,17,2,
20 !065
2600 ! STRATOCUMULUS !174
2610 DATA 89,46,42,89,42,89,
53,125,42,53,63,63,12,63,89,
18 !128
2620 ! NIMBOSTRATUS !078
2630 DATA 144,105,51,130,51,
130,33,130,51,130,134,18,134
,20,134,20 !197
2640 ! NORTHWEST STATES !101
2650 ! CIRRUS !123
2660 DATA 2,3,59,10,59,70,92
,70,97,70,97,69,23,59,10,23
!216
2670 ! ALTOCUMULUS !001
2680 DATA 15,18,96,89,90,89,
111,121,111,121,116,120,90,2
4,10,20 !074
2690 ! ALTOSTRATUS !009
2700 DATA 15,18,96,89,90,89,
53,121,53,121,116,120,12,21,
10,20 !233
2710 ! CUMULUS !209
2720 DATA 115,87,114,87,48,1
14,53,51,53,51,114,114,114,1
2,15,20 !076
2730 ! STRATOCUMULUS !174
2740 DATA 10,87,23,87,23,89,
89,122,89,122,89,124,12,89,1
0,87 !229
2750 ! NIMBOSTRATUS !078
2760 DATA 10,85,84,125,84,12
5,51,125,51,125,51,125,85,85
,10,85 !045
2770 ! NORTHEAST STATES !079
2780 DATA 2,3,59,59,59,59,59
,59,79,54,79,54,1,6,2,3 !033
2790 !ALTOCUMULUS !225
2800 DATA 106,59,63,145,63,1
46,48,68,54,54,54,54,1,17,10
,3 !104
2810 ! ALTOSTRATUS !009
2820 DATA 2,54,63,129,63,132
,63,132,77,77,48,54,48,12,2,
10 !098
2830 ! CUMULUS !209
2840 DATA 55,18,55,138,48,63
```

```
,13,63,13,77,7,18,1,10,147,1
6 !050
2850 ! STRATOCUMULUS !174
2860 DATA 15,19,48,138,89,10
7,45,107,53,54,54,15,12,10,1
0,19 !196
2870 ! NUMBOSTRATUS !090
2880 DATA 85,19,36,125,9,125
,9,107,12,137,85,18,85,10,85
,19 !165
2890 ! GR. LAKES-MIDWEST !06
8
2900 ! CIRRUS !123
2910 DATA 2,5,59,90,63,69,63
,69,63,69,63,69,1,6,2,5 !022
2920 ! ALTOCUMULUS !001
2930 DATA 10,17,63,123,63,12
3,78,123,59,63,59,63,64,64,1
0,17 !202
2940 ! ALTOSTRATUS !009
2950 DATA 10,18,63,123,63,12
3,78,128,63,122,63,122,47,12
6,10,17 !081
2960 ! CUMULUS !209
2970 DATA 10,19,63,63,63,63,
81,53,81,53,63,63,63,41,10,1
9 !044
2980 ! STRATOCUMULUS !174
2990 DATA 10,19,63,75,63,75,
63,75,63,75,63,75,22,75,17,1
9 !070
3000 ! NUMBOSTRATUS !090
3010 DATA 85,87,98,130,10,13
0,10,130,144,131,85,131,85,8
5,87,87 !105
3020 ! CALCULATE FORCAST !09
9
3030 ! FORECAST DATA !052
3040 RESTORE 3100 !133
3050 IF L=7 THEN F$="LITTLE
CHANGE." !253
3060 IF L=7 THEN 1670 !147
3070 FOR AA=1 TO P !199
3080 READ F$ !001
3090 NEXT AA !024
3100 DATA FAIR,FAIR AND COOL
,FAIR AND COLD,FAIR AND OFTE
N COLD,FAIR AND OFTEN VERY C
OLD,FAIR AND MILDER,FAIR AND
 WARMER !192
3110 DATA 8 !148
3120 DATA 9 !149
3130 DATA CLEARING AND COOLE
R !138
```

(See Page 14)

FORECASTER—

(Continued from Page 13)

```
3140 DATA CLEARING AND BECOM
ING FAIR !083
3150 DATA SLOW CLEARING !018
3160 DATA SHOWERS LIKELY WIT
H SLOW CLEARING !183
3170 DATA SHOWERS LIKELY BUT
 CLEARING SKYS !106
3180 DATA SLOW CLEARING AND
COOLDER !057
3190 DATA SLOW CLEARIN AND T
URNING    COLDER AND WINDY !
251
3200 DATA CLEARING AND COLDE
R !127
3210 DATA SLOW CLEARING AND
COLDER !233
3220 DATA CLEARING AND TURNI
NG COLDER WITH GUSTY WINDS !
156
3230 DATA CLEARING AND TURNI
NG COLDER AND WINDY !118
3240 DATA SLOW CLEARING AND
WARMER !254
3250 DATA CLEARING SKYS !023
3260 DATA CLEARING TO FAIR S
KYS !036
3270 DATA SLOW CLEARING AND
MILDER !237
3280 DATA SLOW CLEARING AND
MILDER AND WINDY !149
3290 DATA SLOWLY CLEARING SK
Y AND     WINDY !000
3300 DATA CLEARING SKYS AND
WINDY !191
3310 DATA CLEARING SKYS AND
WINDY !191
3320 DATA 29 !200
3330 DATA RAIN ENDING AND SK
YS        CLEARING !032
3340 DATA RAIN AND WINDY BUT
 BECOMING SHOWERY !006
3350 DATA CONTINUED RAIN AND
 SLEET !246
3360 DATA MORE RAIN AND OFTE
N WINDY !043
3370 DATA SLOWLY DECREASING
RAIN !154
3380 DATA RAIN BUT BECOMING
SHOWERY !094
3390 DATA MORE RAIN AND GUST
Y WINDS !069
3400 DATA 37 !199
3410 DATA 38 !200
3420 DATA SHOWERS AND TURNIN
G COLDER  AND WINDY !124
3430 DATA PROBABLE SHOWERS !
253
3440 DATA PROBABLE SHOWERS B
UT        CLEARING SKYS !208
3450 DATA INCREASING CLOUDS
WITH      SHOWERS LIKELY !13
1
3460 DATA CLOUDY WITH SHOWER
S LIKELY !214
3470 DATA OCCASIONAL SHOWERS
 AND      OFTEN COLDER !178
3480 DATA OCCASIONAL SHOWERS
 !148
3490 DATA PROBABLE SHOWERS B
UT SKYS   CLEARING AND TURNI
NG COLDER !081
3500 DATA POSSIBLE SHOWERS A
ND        SKYS CLEARING !210
3510 DATA POSSIBLE SHOWERS !
023
3520 DATA POSSIBLE SHOWERS B
UT        CLEARING AND MILDE
R !086
3530 DATA POSSIBLE LIGHT SHO
WERS WITH SKYS CLEARING !243
3540 DATA INTERMITTENT SHOWE
RS AND    WARMER !176
3550 DATA FREQUENT SHOWERS !
032
3560 DATA POSSIBLE SHOWERS A
ND MILDER !242
3570 DATA SHOWERS AND WARMER
 !121
3580 DATA SHOWERS AND COOL !
214
3590 DATA 56 !200
3600 DATA 57 !201
3610 DATA 58 !202
3620 DATA INCREASING CLOUDS
!057
3630 DATA DECREASING CLOUDS
AND OFTEN COLDER !164
3640 DATA CLOUDY WITH SHOWER
S POSSIBLE !111
3650 DATA CLOUDY WITH POSSIB
LE        THUNDERSHOWERS !11
9
3660 DATA INCREASING CLOUDS
AND       POSSIBLE SHOWERS !
211
3670 DATA HIGH CLOUDS !112
3680 DATA INCREASING CLOUDS
WITH RAIN AND WIND !055
3690 DATA INCREASING CLOUDS
WITH RAIN BUT MILDER !220
3700 DATA INCREASING CLOUDS
AND RAIN  OR SLEET !007
3710 DATA INCREASING CLOUDS
AND OFTEN RAIN !033
3720 DATA INCREASING CLOUDS
AND MILDER !020
3730 DATA INCREASING CLOUDS
AND       OCCASIONAL RAIN !0
76
3740 DATA CLOUDY AND OFTEN M
ILDER !174
3750 DATA INCREASING HIGH CO
LUDINESS !177
3760 DATA CLOUDY MILDER AND
WINDY !189
3770 DATA CONTINUED CLOUDY A
ND COLDER !219
3780 DATA INCREASING CLOUDS
AND RAIN OR SNOW !175
3790 DATA INCREASING HIGH CL
OUDS AND  MILDER !122
3800 DATA DECREASING CLOUDS
!043
3810 DATA INCREASING CLOUDS
WITH      SHOWERS AND GUSTY
WINDS !246
3820 DATA INCREASING CLOUDS
AND WARMER !037
3830 DATA FREQUENT THUNDERSH
OWERS !065
3840 DATA THUNDERSTORMS WITH
 POSSIBLE HAIL! !186
3850 DATA PROBABLE THUNDERSH
OWERS !030
3860 DATA POSSIBLE SCATTERED
 THUNDER  SHOWERS !067
3870 DATA DECREASING SHOWERS
 WITH SKYSSLOWLY CLEARING !1
87
3880 DATA DECREASING SHOWERS
 WITH SKYSCLEARING !170
3890 DATA DECREASING CLOUDS
AND OFTEN HOT !211
3900 DATA DECREASING CLOUDS
AND COOLER !013
3910 DATA DECREASING CLOUDS
AND MILDER !006
3920 DATA CONTINUED CLOUDY !
004
3930 DATA CONTINUED HIGH CLO
UDS !067
3940 DATA HIGH CLOUDS AND CL
EARING    SKYS !167
```

(See Page 15)

FORECASTER—

(Continued from Page 14)

```
3950 DATA SHOWERS BUT SKYS C
LEARING !121
3960 DATA SHOWERERS AND WIND
Y !206
3970 DATA CONTINUED CLOUDY A
ND COLD !066
3980 DATA CLOUDS AND WINDY !
211
3990 DATA DECREASING CLOUDS
!043
4000 DATA CLOUDY AND WARMER
!029
4010 DATA RAIN SLOWLY DECREA
SING !154
4020 DATA RAIN WITH POSSIBLE
 THUNDER  STORMS !230
4030 DATA POSSIBLE SCATTERED
 SHOWERS !224
4040 DATA POSSIBLE SCATTERED
 THUNDER  SHOWERS !067
4050 DATA RAIN OFTEN HEAVY A
T TIMES !046
4060 DATA RAIN AND WINDY !04
9
4070 DATA CLOUDY WITH DIZZLE
 OFTEN !045
4080 DATA RAIN OR SNOW COLD
AND WINDY !168
4090 DATA LITTLE CHANGE !252
4100 DATA STORMY AND MILDER
!042
4110 DATA 108 !247
4120 DATA 109 !248
4130 DATA CLOUDY WITH A FEW
SCATTERED SHOWERS !023
4140 DATA POSSIBLE LIGHT RAI
N AND     WARMER !033
4150 DATA PROBABLE RAIN AND
GUSTY WIND !009
4160 DATA PROBABLE SHOWERS A
ND FASTLY  CLEARING SKYS !23
6
4170 DATA SCATTERED SHOWERS
!086
4180 DATA SCATTERED SHOWERS
AND COOLER !056
4190 DATA CLOUDY WITH A FEW
ACATTERED SHOWERS !005
4200 DATA 117 !247
4210 DATA 118 !248
4220 DATA 119 !249
4230 DATA PROBABLE RAIN OR S
NOW         SHOWERS !066
4240 DATA PROBABLE RAIN OR S
NOW AND   WINDY !019
4250 DATA RAIN OR SNOW AND M
ILD !251
4260 DATA RAIN OR SNOW WITH
GUSTY     WINDS !011
4270 DATA OCCASIONAL LIGHT R
AIN OR    SNOW !193
4280 DATA RAIN OR SNOW AND W
INDY !097
4290 DATA RAIN WITH SLOWLY C
LEARIN    SKYS !243
4300 DATA RAIN OR SNOW WITH
SKYS       CLEARING AND MILD
!222
4310 DATA RAIN OR SNOW !185
4320 DATA CLOUDY WITH RAIN O
R SNOW !017
4330 DATA HEAVY RAIN OR SNOW
 !092
4340 DATA STORMY BUT SLOWLY
CLEARING !221
4350 DATA RAIN OR SNOW AND C
OLD !247
4360 DATA HEAVY RAIN !044
4370 DATA SHOWERS ENDING AND
 SKYS     CLEARING !193
4380 DATA INCREASING CLOUDS
AND OFTEN FOGGY !116
4390 DATA INCREASING CLOUDS
AND RAIN  OR SNOW !208
4400 DATA POSSIBLE SHOWERS A
ND SLOW   IMPROVEMENT !060
4410 DATA STORMY AND OFTEN S
NOW !084
4420 DATA DECREASING STORMS
WINDY AND COLD !056
4430 DATA SLOW DECREASING RA
IN !243
4440 DATA DECREASING STORMS
AND SLOWLYCLEARING SKYS !013
4450 DATA CONTINUES RAIN OR
SLEET !210
4460 DATA RAIN ENDING AND SK
YS        CLEARING !032
4470 DATA RAIN ENDING AND SK
YS        CLEARING !032
4480 DATA CLOUDY AND OFTEN R
AIN OR    SNOW !172
4490 DATA UNSETTLED AND MILD
 !094
4500 DATA OCCASIONAL SHOWERS
 !148
4510 GOTO 1680 !229
4520 ! FINAL CALCULATION & !
162
4530 ! FORECAST DISPLAYED !1
85
4540 IF O=0 THEN G$="CALM, S
MOKE RISES VERTICALLY." !034
4550 IF O=1 OR 2 OR 3 THEN G
$="LIGHT AIR, WEATHER VANE
   INACTIVE; SMOKE DRIFTS."
!202
4560 IF (O>=4)*(O<=7)THEN G$
="LIGHT BREEZE, WEATHER VANE
S ACTIVE; WIND FELT ON FACE
  LEAVES RUSTLE" !169
4570 IF (O>=8)*(O<=12)THEN G
$="LEAVES AND SMALL TWIGS MO
VE LIGHT FLAGS EXTEND" !020
4580 IF (O>=13)*(O<=18)THEN
G$="FRESH BREEZE, SMALL TREE
S   SWAY DUST AND LOOSE PAPE
R   BLOW ABOUT." !073
4590 IF (O>=19)*(O<=24)THEN
G$="FRESH BREEZE, SMALL TREE
S   SWAY WAVES BREAK ON INLA
ND  WATERS." !096
4600 IF (O>=25)*(O<=31)THEN
G$="STRONG BREEZE, LARGE
    BRANCHES SWAY UNBRELLAS
    DIFFICULT TO USE." !029
4610 IF (O>=32)*(O<=38)THEN
G$="WHOLE TREES SWAY; DIFFIC
ULT TO WALK AGAINST WIND." !
035
4620 IF (O>=39)*(O<=46)THEN
G$="FRESH GAIL, TWIGS BROKEN
 OFF TREES; WALKING AGAINST
WIND VERY DIFFICULT." !073
4630 IF (O>=47)*(O<=54)THEN
G$="STRONG GAIL, SLIGHT DAMA
GE  TO BUILDINGS; SHINGLES B
LOWN OFF ROOF." !113
4640 IF (O>=55)*(O<=63)THEN
G$="WHOLE GAIL, TREES UPROOT
ED  CONSIDERAL DAMAGE TO
     BUILDINGS." !182
4650 IF (O>=64)*(O<=73)THEN
G$="STORM! WIDE SPREAD DAMAG
E,  TAKE COVER!" !215
4660 IF O>74 THEN G$="HURRIC
ANE!!!! VIOLENT        DESCRU
CTIONS, TAKE SHELTER!" !068
4670 CALL CLEAR !209
4671 CALL CHAR(96,"FFFFFFFF
FFFFFFF")::CALL VCHAR(1,1,9
6,25)::CALL VCHAR(1,32,96,2
5)!212
```

(See Page 16)

FORECASTER—

(Continued from Page 15)

```
4680 DISPLAY AT(2,2):"TEMP=
";Q !194
4690 DISPLAY AT(3,2):"WIND S
PEED= ";O !084
4700 IF E$="N" THEN 4720 !18
8
4710 IF Q>85 THEN DISPLAY AT
(4,2):"HUMITURE= ";HUMITURE
!163
4715 DISPLAY AT(7,2):"CLOUD
BASE AT ";CLB;" FEET" !007
4720 IF O<4 OR Q>35 THEN 474
0 !155
4730 DISPLAY AT(4,2):"WINDCH
ILL= ";W !055
4740 IF S=0 THEN 4760 !177
4750 DISPLAY AT(5,2):"PRESSU
RE ";S;" ";C$ !135
4760 IF E$="N" THEN 4790 !00
2
4770 DISPLAY AT(6,2):"DEWPOI
NT=";DPT !116
4780 IF E$="N" THEN 4800 !01
2
4790 DISPLAY AT(9,9):"WIND A
DVISORY" !002
4800 DISPLAY AT(10,1):G$ !24
7
4810 DISPLAY AT(15,12):"FORE
CAST" !234
4820 DISPLAY AT(17,1):F$ !25
3
4830 IF B$="N" THEN D$="INSU
FFICIENT DATA...." !233
4840 DISPLAY AT(21,2):"SECON
DARY FORECAST":" ";D$ !064
4850 PRINT "PRINTER Y/N" ::
ACCEPT AT(23,13)VALIDATE("YN
")BEEP SIZE(1):H$ !087
4860 IF H$="N" THEN 5120 ELS
E 4870 !180
4870 INPUT "PRINTER NAME? ":
I$ !077
4880 IF I$="0" THEN 4850 !03
6
4890 OPEN #1:I$ !184
4900 PRINT #1:"-------------
----------------------------
----------------------------
-----------" !111
4910 PRINT #1:"TEMP= ";Q !17
3
4930 PRINT #1:"WIND SPEED= "
;O !062
4950 IF E$="N" THEN 4980 !19
3
4970 IF Q>85 THEN PRINT #1:"
HUMITURE= ∫";HUMITURE !071
4975 PRINT #1:"CLOUD BASE AT
 ";CLB;" FEET" !237
4980 IF O<4 OR Q>35 THEN 500
1 !161
4990 PRINT #1:"WIND CHILL= "
;W !065
5001 IF E$="N" THEN 5010 !22
3
5002 PRINT #1:"DEWPOINT= ";D
PT !124
5010 IF S=0 THEN 5030 !192
5020 PRINT #1:"PRESSURE ";S;
" ";C$ !111
5030 PRINT #1:"":"":"
             WIND ADVISORY":
"" !231
5040 PRINT #1:G$ !179
5070 PRINT #1:"":"FORECAST=
";F$ !103
5100 PRINT #1:"SECONDARY FOR
ECAST= ";D$ !187
5110 PRINT #1:"":"----------
----------------------------
----------------------------
--------------" !235
5111 CLOSE #1 !151
5120 PRINT "ANOTHER FORECAST
?" :: ACCEPT AT(23,19)VALIDA
TE("YN")BEEP SIZE(1):J$ !018
5125 DPT=0 :: S=0 !043
5130 IF J$="Y" THEN 380 !199
5140 CALL CLEAR :: STOP !235
5150 ! TITLE SCREEN !005
5160 CALL CLEAR :: CALL SCRE
EN(2)!230
5170 CALL SAY("HELLO")!002
5180 FOR I=2 TO 14 :: CALL C
OLOR(I,16,2):: NEXT I !130
5190 RESTORE 5260 !253
5200 FOR BA=2 TO 28 !173
5210 READ CA !027
5220 CALL SOUND(-30,-5,2)!11
2
5230 CALL HCHAR(7,BA,CA)!170
5240 FOR DA=1 TO 50 :: NEXT
DA !070
5250 NEXT BA !025
5260 DATA 87,69,65,84,72,69,
82,32,70,79,82,69,67,65,83,8
4,69,82,32,40,67,41,32,49,57
,56,53 !079
5270 FOR I=1 TO 12 :: DISPLA
Y AT(9,I-1):" " :: DISPLAY A
T(9,I):"BY" :: CALL SOUND(-1
0,I+110,2):: NEXT I !211
5280 FOR I=20 TO 11 STEP -1
:: DISPLAY AT(I+1,9):" " ::
DISPLAY AT(I,9):"GARY COX" :
: CALL SOUND(-10,I+110,2)::
NEXT I !025
5285 DISPLAY AT(16,9):"Versi
on 2" !162
5290 FOR EA=1 TO 1000 :: NEX
T EA !166
5300 CALL CLEAR :: FOR I=2 T
O 14 :: CALL COLOR(I,16,5)::
 NEXT I :: CALL SCREEN(5)::
RETURN !250
5310 CALL CLEAR !209
5320 DISPLAY AT(2,10):"HELP
SCREEN":"":"":" 1. Cirrus":"
 2. Altocumulus":" 3. Altost
ratus":" 4. Cumulus":" 5. St
ratocumulus" !106
5330 DISPLAY AT(10,2):"6. Ni
mbostratus":" 7. Return to c
loud menu" !186
5340 ACCEPT AT(23,1)VALIDATE
("1234567")BEEP SIZE(1):FA !
254
5350 CALL CLEAR !209
5360 IF FA=1 THEN 5380 :: IF
 FA=2 THEN 5430 :: IF FA=3 T
HEN 5460 :: IF FA=4 THEN 550
0 :: IF FA=5 THEN 5520 !065
5370 IF FA=6 THEN 5540 :: IF
 FA=7 THEN 1240 !154
5380 ! CIRRUS EXPLINATION !2
30
5390 CALL CLEAR :: PRINT "Ci
rrus clouds are high":"cloud
s (above 20,00 feet)":"and a
re usually white, in":"varie
d forms such as tufts" !157
5400 PRINT "of plumes, often
 fusing":"together. They are
 very":"light and you can us
ually":"see some blue skys t
hrough" !172
5410 PRINT "them. They seldo
m produce":"rain." !033
5420 PRINT : : : :: INPUT "P
RESS ENTER ":K$ :: GOTO 5310
 !180
5430 PRINT "Altocumulus clou
```

(See Page 17)

FORECASTER—

(Continued from Page 16)

```
ds have a base above 8.00 fe
et with":"patches of detache
d fleecy":"clouds with globu
lar masses" !015
5440 PRINT "which are often
arranged in lines or waves.
When cover  is dark occasion
al showers  are possible." !
211
5450 PRINT : : : :: INPUT "P
RESS ENTER ":K$ :: GOTO 5310
 !180
5460 PRINT "Altostratus clou
ds are like":"thick Cirrus b
ut lower.":"Possible of prod
ucing":"moderate precipitati
on." !106
5490 PRINT : : : :: INPUT "P
RESS ENTER ":K$ :: GOTO 5310
 !180
5500 PRINT "cumulus clouds h
ave":"vertical developement
and":"are puffey. The tops a
re":"often dome-shaped with":
"horizontal bottoms." !049
5505 PRINT "They are often c
alled":"thunder heads." !002
5510 PRINT : : : :: INPUT "P
RESS ENTER ":K$ :: GOTO 5310
 !180
5520 PRINT "Stratocumulus ar
e roll":"clouds with flat ba
sses":"ranging from 3 to 5,0
00 feet":"Generally light gr
ay with":"darker parts." !17
2
5530 PRINT : : : :: INPUT "P
RESS ENTER ":K$ :: GOTO 5310
 !180
5540 PRINT "Nimbostratus are
 low, dark":"thick clouds an
d are":"sometimes accompanie
d by":"low flying black patc
hes." !020
5550 PRINT : : : :: INPUT "P
RESS ENTER ":K$ :: GOTO 5310
 !180
5560 ! INSTRUCTIONS !088
5570 CALL CLEAR !209
5580 PRINT "        INSTRUC
TIONS":"":"" !069
5590 PRINT "A Limited licens
e is granted":"to all users
of this program":"to make co
pies of this":"program and d
istribute it" !150
5600 PRINT "only if the prog
ram is not":"in any way alte
red." !130
5610 PRINT "No fee for profi
t is to be":"charged for cop
ying or":"distributing the p
rogram" !145
5620 PRINT "without express
writtent":"agreement with Ga
ry Cox":"This program is not
":"guaranteed to be free fro
m" !096
5630 PRINT "errors nor accur
ate....":"The users accept t
his":"program on an 'as is'
basis.":"" !168
5640 INPUT "       PRESS ENT
ER ":L$ :: CALL CLEAR !015
5650 PRINT "  ====== FREEWAR
E ======":"":"User Supported
 Software":"If you have used
 this":"program and found it
 of" !093
5660 PRINT "value any contri
bution":"    ($5.00 suggeste
d)" !114
5670 PRINT "will be apprecia
ted.":"Regardless of whether
 you":"make a contribution,
you":"are encouraged to copy
 and":"share this program."
!168
5680 PRINT "":"Thank you":""
:"Gary Cox":"3174 Melbourne"
:"Memphis, Tn 38127":"":"901
-358-0667":"" :: INPUT "
PRESS ENTER ":L$ !175
5690 CALL CLEAR :: PRINT "
 INSTRUCTIONS":"":"In order
for the program":"to forecas
t the weather":"with some ac
curacy you must" !019
5700 PRINT "enter accurate i
nformation,":"in particular
the wind":"direction must be
 correct." !080
5710 PRINT "At most prompts
you may ":"enter a 0 to retu
rn to a":"previous question
in case":"you need to change
 the" !008
5720 PRINT "answer, except f
or the wind speed and temper
ature":" prompts....":"" ::
INPUT "    PRESS ENTER ":M$
:: CALL CLEAR !005
5730 PRINT "When the compute
r displays":"the forecast th
e wind chill":"factor will b
e given which":"is how cold
it feels..." !203
5740 PRINT "or the humiture
will be":"given which is how
 hot it":"feels." !128
5750 PRINT "Also a secondary
 forecast":"will be displaye
d to give":"you a comparison
 of 2 ":"forecasts." !124
5760 PRINT "The rest is abou
t self":"explanitory. So hav
e fun!":"" :: INPUT "     PR
ESS ENTER ":M$ !164
5770 RETURN !136
5780 END !139
8000 SUB HUMID(F2,R,DPT,HUMI
TURE,CLB)!020
8001 K2=273.16+5/9*(F2-32)::
 E2=6.108*EXP((17.27*(K2-273
.16))/(K2-35.86))!191
8002 E=R*E2/100 :: F1=999 ::
 THI=F2-(.55-.0055*R)*(F2-58
):: THI=INT(THI+.5)!124
8003 X=LOG(E/6.108)/17.27 ::
 D=(273.16-(35.86*X))/(1-X):
: D1=D-273.16 !192
8004 D2=(9/5)*D1+32 :: D2=IN
T(D2+.5):: H=225*(F2-D2):: H
T=E-10 !189
8005 IF HT<0 THEN HT=0 !219
8006 HUMITURE=INT(F2+HT+.5):
: DPT=D2 :: CLB=H
8007 SUBEND
```

Converting assembly object code into Extended BASIC CALL LOADs

The following program, ACE, was written by Italian programmer Paolo Bagnaresi. It requires Extended BASIC and a memory expansion.

The program is used to convert an uncompressed assembly language D/F80 object file into CALL LOAD statements for use in Extended BASIC programs.

The program prompts the user for inputs.

```
1 ! ACE : Assembly Object to
 Extended Basic CALL LOADs C
onverter        8/3/1984 !
153
2 !           By
     Paolo Bagnaresi
    Tel.(02)-514.202
Address: !116
3 !  Via J.F. Kennedy 17
  20097 San Donato Milanese
       (Milan)- Italy !206
10 GOTO 40 :: DIM STDEF$(100
):: D$,N$,F$,DEF$,PB$,SC$,RI
$,CT$,L$,HEX$,H$,DSC$,DECC$,
PROG$ !018
20 CALL LOAD :: CALL INIT ::
 CALL LINK :: CALL PEEK :: C
ALL CHAR :: CALL HCHAR :: CA
LL KEY !255
30 AUT,N,A,B,C,D,E,F,G,H,I,L
,M,N,CT,MS,LS,DBM,DBL,FINELO
C,LOC,INIZLOC,INDEF,NDEF,NLI
NK,NL,NLINE,NST,KY,ST,DEC,PO
,Z !113
40 CALL CLEAR :: CALL SCREEN
(16):: FOR T=0 TO 14 :: CALL
 COLOR(T,13,16):: NEXT T ::
!@P- !052
50 CALL CHAR(128,"00282828",
131,"000000FF"):: L$=RPT$(CH
R$(131),28):: H$="0123456789
ABCDEF" :: CALL CLEAR !187
60 PB$="BY Paolo Bagnaresi
            Via J.F. Kennedy
17          20097 San Donato
Milanese   (Milan)- ITALY" !
237
70 DISPLAY AT(1,1):L$:L$: :T
AB(13);"ACE": :TAB(14);"by":
TAB(7);"Paolo Bagnaresi":TAB
(7);"Tel(02)-514.202":"San D
onato Milanese-ITALY":L$ !07
3
80 DISPLAY AT(11,1):TAB(10);
"Assembly":TAB(10);"Converte
r to":TAB(10);"Extended":L$
!145
90 DISPLAY AT(14,1):L$:"ACE
converts the Object":"of an
Assembly Program into":"an E
xtended Basic Program.":"the
 Assembly Program MUST be" !
180
100 DISPLAY AT(19,1):"suitab
le for Extended Basic":"envi
ronment and MUST NOT":"conta
in any AORG.":L$ !029
110 DISPLAY AT(24,6)BEEP:"Pr
ess any key" !054
120 CALL KEY(0,KY,ST):: IF S
T=0 THEN 120 ELSE CALL CLEAR
 !015
130 DISPLAY AT(1,1):L$:TAB(5
);"Are the Assembly":"Object
s already loaded ?": :"  Ans
wer: (Y/N) N":L$ !153
140 ACCEPT AT(5,17)SIZE(-1)V
ALIDATE("YN")BEEP:RI$ :: IF
RI$="" THEN 50 ELSE IF RI$="
Y" THEN CALL CLEAR :: GOTO 3
70 ELSE N$="1" :: CT$="Y" ::
 GOSUB 900 !177
150 DISPLAY AT(6,1):L$:"Inse
rt the diskette with the":"a
ssembly object and enter":"t
he object progr. name": :"Na
me :  ";PROG$:L$ !065
160 DISPLAY AT(22,1):L$:"era
se and press ENTER if you":T
AB(9);"are through" !080170
ACCEPT AT(11,9)SIZE(-10)BEEP
:PROG$ :: IF PROG$="" THEN C
ALL CLEAR :: GOTO 320 !160
180 DISPLAY AT(13,1):"Disk D
rive ? (1-3) ";N$:L$ :: ACCE
PT AT(13,20)SIZE(-1)VALIDATE
("123")BEEP:N$ :: IF N$="" T
HEN 130 !086
190 ON ERROR 360 :: CALL LOA
D("DSK"&N$&"."&PROG$):: ON E
RROR STOP !220
200 DISPLAY AT(15,1):"Do you
 want to check the":"loaded
program ? (Y/N) ";CT$:L$ ::
ACCEPT AT(16,24)SIZE(-1)VALI
DATE("YN")BEEP:CT$ :: IF CT$
="N" THEN 150 !242
210 DISPLAY AT(18,1):"Does t
he program come back":"to Ex
tended Basic ? (Y/N) Y":L$ :
: ACCEPT AT(19,27)SIZE(-1)VA
LIDATE("YN")BEEP:SC$ !232
220 IF SC$="Y" THEN 240 ELSE
 FOR T=1 TO 10 :: DISPLAY AT
(21,1)BEEP:"In this case no
check":"is possible ":L$:L$
!010
230 FOR I=1 TO 100 :: NEXT I
 :: CALL HCHAR(21,1,32,96)::
 NEXT T :: GOTO 150 !053240
CALL PEEK(8196,A,B):: NST=0
:: INDEF=A*256+B :: FOR T=16
376 TO INDEF STEP -8 :: NL=T
 :: GOSUB 870 :: NST=NST+1 :
: STDEF$(NST)=DEF$ :: NEXT T
 !052
250 CALL CLEAR :: A=0 :: D$=
"1" !236
260 DISPLAY AT(1,1):L$:"List
 of DEFS to choose from":"fo
r checking purposes":L$ !167
270 FOR T=5 TO 20 STEP 2 ::
FOR Z=1 TO 19 STEP 9 :: A=A+
1 :: DISPLAY AT(T,Z)BEEP:A;S
TDEF$(A):: IF A>=NST THEN 28
0 ELSE NEXT Z :: NEXT T !071
280 DISPLAY AT(T+1,1):L$ ::
DISPLAY AT(20,1):L$:"DEF No.
 ? (1 -";NST;") ":L$:"Press
ENTER when finished":L$:"Pre
ss ENTER when finished":L$ !
190
285 ACCEPT AT(21,21)VALIDATE
(DIGIT)SIZE(-2)BEEP:D$ :: IF
 D$<>"" THEN 340 !004
290 CALL CLEAR :: DISPLAY AT
(1,1):L$:"Are the programs l
oaded": :"so far OK. ? (Y/N)
 Y":L$ :: ACCEPT AT(4,20)VAL
IDATE("YN")SIZE(-1)BEEP:SC$
:: IF SC$="Y" THEN 320 !238
300 DISPLAY AT(6,1):L$:"Unfo
rtunately in this case": :"i
t's not possible to": :"elim
inate just one program": :"b
```

(See Page 19)

CALL LOAD CONVERTER—

(Continued from Page 18)

```
ut it's necessary to load" !
098
310 DISPLAY AT(15,1):"all th
e program(s) all ": :"over a
gain.": :"OK? (Y) Y":L$ :: A
CCEPT AT(19,9)VALIDATE("Y")S
IZE(-1)BEEP:SC$ :: CALL INIT
 :: CALL CLEAR :: GO TO 150
!017
320 DISPLAY AT(6,1):L$:"Are
all the programs": :"loaded
already ? (Y/N) Y":L$ :: ACC
EPT AT(9,24)VALIDATE("YN")SI
ZE(-1)BEEP:SC$ !105
330 IF SC$="N" OR SC$="" THE
N CALL CLEAR :: GOTO 150 ELS
E 370 !176
340 A=VAL(D$):: IF A>NST THE
N 280 ELSE CALL LINK(STDEF$(
A)):: GOTO 250 !212
350 ! Error handling !018
360 FOR T=1 TO 8 :: DISPLAY
AT(20,1)BEEP:L$:TAB(6);"Driv
e error or":TAB(6);"Program
name error":L$:L$ :: FOR I=1
 TO 100 :: NEXT I !246
365 CALL HCHAR(20,1,32,128):
: NEXT T :: RETURN 150 !230
370 CALL CLEAR !209
380 ON ERROR 400 :: CALL PEE
K(8194,A,B,C,D):: FINELOC=A*
256+B :: NL,INDEF=C*256+D ::
 GOSUB 870 !223
385 IF ASC(DEF$)=255 THEN 40
0 ELSE INIZLOC=DBM*256+DBL !
215
390 ON ERROR STOP :: GOTO 43
0 !088
400 CALL CLEAR :: FOR I=1 TO
 10 :: DISPLAY AT(10,1)BEEP:
L$:L$:"  The Assembly Progra
ms": :"   have not been load
ed": :TAB(10);"LOAD THEM!":L
$:L$ !104
410 FOR T=1 TO 100 :: NEXT T
 :: CALL HCHAR(12,1,32,160):
: NEXT I :: GOSUB 900 :: GOT
O 150 !194
420 !Disk-printing routine !
204
430 CALL CLEAR :: GOSUB 790
:: IF F$="" OR N$="" THEN 32
767 :: ON ERROR 840 :: GOSUB
 920 :: OPEN #2:"DSK"&N$&"."
&F$,VARIABLE 163 :: ON ERROR
 STOP :: N=0 !252
440 !Address of the programm
er !251
450 PRINT #2:CHR$(0)&CHR$(N)
&CHR$(131)&CHR$(199)&CHR$(LE
N(PB$))&PB$&CHR$(0):: N=1 ::
 GOSUB 940 !240
460 !Insert CALL INIT !136
470 PRINT #2:CHR$(0)&CHR$(N)
&CHR$(157)&CHR$(200)&CHR$(4)
&"INIT"&CHR$(0):: N=2 :: LOC
=INDEF :: GOSUB 940 !186
480 ! DEFs name printing !04
9
490 FOR NDEF=INDEF TO 16376
STEP 8 !005
500 PRINT #2:CHR$(0)&CHR$(N)
&CHR$(157)&CHR$(200)&CHR$(4)
&"LOAD"&CHR$(183)&CHR$(200)&
CHR$(LEN(STR$(NDEF)))&STR$(N
DEF);!081
510 FOR LOC=NDEF TO NDEF+6 S
TEP 2 !128
520 CALL PEEK(LOC,MS,LS):: P
RINT #2:CHR$(179)&CHR$(200)&
CHR$(LEN(STR$(MS)))&STR$(MS)
&CHR$(179)&CHR$(200)&CHR$(LE
N(STR$(LS)))&STR$(LS);!075
530 NEXT LOC !116
540 PRINT #2:CHR$(182)&CHR$(
0):: GOSUB 940 :: N=N+1 :: N
EXT NDEF !051
550 !Print DEF pointer and F
FALM !121
560 PRINT #2:CHR$(0)&CHR$(N)
&CHR$(157)&CHR$(200)&CHR$(4)
&"LOAD"&CHR$(183)&CHR$(200)&
CHR$(LEN(STR$(8194)))&STR$(8
194);!091
570 FOR LOC=8194 TO 8196 STE
P 2 !204
580 CALL PEEK(LOC,MS,LS):: P
RINT #2:CHR$(179)&CHR$(200)&
CHR$(LEN(STR$(MS)))&STR$(MS)
&CHR$(179)&CHR$(200)&CHR$(LE
N(STR$(LS)))&STR$(LS);!075
590 NEXT LOC !116
600 PRINT #2:CHR$(182)&CHR$(
0):: GOSUB 940 :: N=N+1 :: L
OC=9460 !187
610 ! Main program printing
!203
620 PRINT #2:CHR$(0)&CHR$(N)
&CHR$(157)&CHR$(200)&CHR$(4)
&"LOAD"&CHR$(183)&CHR$(200)&
CHR$(LEN(STR$(LOC)))&STR$(LO
C);!211
630 FOR LOC=LOC TO LOC+20 ST
EP 2 !047
640 IF LOC>FINELOC THEN 670
:: CALL PEEK(LOC,MS,LS):: PR
INT #2:CHR$(179)&CHR$(200)&C
HR$(LEN(STR$(MS)))&STR$(MS)&
CHR$(179)&CHR$(200)&CHR$(LEN
(STR$(LS)))&STR$(LS);!008
650 NEXT LOC !116
660 PRINT #2:CHR$(182)&CHR$(
0):: GOSUB 940 :: N=N+1 :: I
F LOC<=FINELOC THEN 620 ELSE
 680 !058
670 PRINT #2:CHR$(182)&CHR$(
0):: GOSUB 940 !103
680 N=N+1 !021
690 !CALL LINK printing !120
700 FOR NLINK=INDEF TO 16376
 STEP 8 :: NL=NLINK :: GOSUB
 870 !245
710 PRINT #2:CHR$(0)&CHR$(N)
&CHR$(157)&CHR$(200)&CHR$(4)
&"LINK"&CHR$(183)&CHR$(199)&
CHR$(LEN(DEF$))&DEF$&CHR$(18
2)&CHR$(0):: GOSUB 940 !174
720 N=N+1 :: NEXT NLINK !169
730 PRINT #2:CHR$(255)&CHR$(
255):: CLOSE #2 !111
740 CALL CLEAR !209
745 DISPLAY AT(5,1)BEEP:L$:"
The assembly program ";DEF$:
 :"has been recorded as a":
:"DIS/VAR 163 file. The name
": :"of this file is ";F$:L$
 !226
750 DISPLAY AT(14,1):"You ca
n MERGE this file": :"and ob
tain an Ext.B.Program":L$:"E
xecute now in command mode:"
: :">NEW":">MERGE DSK";N$;".
";F$ !113
760 FOR T=1 TO 70 :: DISPLAY
 AT(23,1)BEEP:">SAVE DSK";N$
;".";SEG$(F$,1,LEN(F$)-3)&"E
XT":L$ :: CALL KEY(0,KY,ST):
: IF ST<>0 THEN STOP !165
770 NEXT T :: END !247
780 !Open file: disk drive
& name selection !033
790 DISPLAY AT(1,1):L$:"Name
 of the last DEF":"of the as
```

(See Page 20)

CALL LOAD CONVERTER—

(Continued from Page 19)

```
sembly programs": :"loaded i
n memory : ";DEF$:L$ !109
800 F$=DEF$&"MRG" :: DISPLAY
 AT(8,1):L$:"proposed name f
or the file": :"Max 10 chara
cters ";F$: :L$ :: ACCEPT AT
(11,19)SIZE(-10)BEEP:F$ !076
810 IF F$="" THEN RETURN ELS
E IF POS(F$," ",1)>0 OR POS(
F$,".",1)>0 THEN 800 !118
820 DISPLAY AT(14,1):L$:"Dis
k Drive? (1-3) ";N$:L$ :: AC
CEPT AT(15,19)VALIDATE("123"
)SIZE(-1)BEEP:N$ :: RETURN !
119
830 !Sub file error !183
840 ON ERROR 850 :: CLOSE #2
 !120
850 RETURN 430 !000
860 !call peek DEF names !00
7
870 CALL PEEK(NL,E,F,G,H,I,L
,DBM,DBL):: DEF$=CHR$(E)&CHR
$(F)&CHR$(G)&CHR$(H)&CHR$(I)
&CHR$(L)!141
880 PO=POS(DEF$," ",1):: IF
PO>0 THEN DEF$=SEG$(DEF$,1,P
O-1):: RETURN ELSE RETURN !1
63
890 !Sub CALL INIT once only
 !228
900 IF CT=1 THEN RETURN ELSE
 CALL INIT :: CT=1 :: RETURN
 !004
910 ! Sub # of necessary pri
ntings !240
920 NLINE=ABS(INT(-((FINELOC
-9460)/22+(16384-INDEF)/4+3,
))!062
925 DISPLAY AT(17,1)BEEP:"Th
e necessary Printing":"opera
tions with Disk Drive": :"(m
ax 172) will be";NLINE:L$ !2
43
930 IF NLINE>172 THEN FOR T=
1 TO 10 :: FOR I=1 TO 90 ::
NEXT I :: CALL HCHAR(23,1,32
,32):: DISPLAY AT(22,1)BEEP:
L$:"OBJECT SIZE IS TOO LARGE
":L$ :: NEXT T :: STOP ELSE
RETURN !024
940 NLINE=NLINE-1 :: DISPLAY
 AT(21,1):L$:"# of printings
 yet to be":"executed will b
e";NLINE:L$ :: RETURN !231
```

Living with spiders

Programming with Funnelweb

By TONY MCGOVERN

The following is the first of a series of three articles about programming with Funnelweb. Funnelweb is a multi-purpose operating environment for the TI.

What does it take to coexist with the infamous Atrax Robustus, the funnelweb spider? Mainly it needs a certain amount of discretion about walking out in the yard at night in bare feet, care in gardening, and remember always to shake out your rubber boots before putting them on.

Less seriously, though, what we want to do here is give a workthrough on how to write assembly programs to live in spiderland.

The Funnelweb system, I suppose, can be categorized as something between an operating system for the TI-99 and a shell for what already exists there in various forms. TI was trying to sell lots of cartridges and either didn't care to tie it all together at the user level, or even took definite steps to prevent it. The expanded TI-99 system has lasted so well since Orphan Day because it was thoroughly engineered behind the scenes, and outside programmers have been able to draw on this underlying strength. The Funnelweb system is an attempt to gather those strengths in a form convenient for the general user Perhaps the heading on the XB LOAD program of "Utility Environment" sums it up best.

Whatever you call it, though, it can provide improved user convenience for programs written to the normal rules. If you further respect its own internal rules, just as you would do with the DOS on any other system, then better programs still can be written. The Funnelweb package contains several utility programs which do just this, some usable only with Funnelweb, and some which can run free-standing. This article is intended to give guidance on writing dual-mode programs which can function with or without FWB. Two examples are the DM-1000 files MG/MG and Linehunter LH which were used as a detailed example in the V4.12 update.

Whatever you call it, though, it can provide improved user convenience for programs written to the normal rules.

The requirements for dual-mode programs are that they

(1) Observe FWB style
(2) Know the load path
(3) Don't tread on Atrax R.
(4) Make a graceful exit

Let's look at these in more detail. The first item is not essential but is nice to have. A good example to follow is the way FWB has in its turn followed the TI-Writer Editor in making CTRL-C an alternative to FCTN-9 for BACK or Escape. Once using CTRL-C has become familiar, FCTN-9 seems like the imposition of an awkward stretch. A FWB convention to save more awkward stretching is to use CTRL-A as an alternative to FCTN-6 for Proceed. Think of "A for Action" as mnemonic.

The next hurdle is for the program to

(See Page 21)

PROGRAMMING IN FUNNELWEB—

(Continued from Page 20)

know whether it has been loaded from FWB so that it can call on details or routines from FWB. What sort of details? One is to know what the screen colors were so that these can be retained and a jarring transition to a standard color set avoided. Remember that FWB gives great flexibility for you to choose your favorite colors of the moment.

Another is to know what drive the program was loaded from and, if complete reload of FWB is needed, to know where FWB was lurking also. The nature of a dual-mode application means that the special FWB information will be used only at entry and exit, the main exception here being the availability of QD from FCTN-7 Aid as a "hot key" consistent with normal FWB usage as per item No. 1.

Next we want to avoid conflict with FWB or, if this is unavoidable, to know how to restore FWB before exit. The primary requirement is for FWB itself to remain unmolested. This means that the program must leave the top 5K or so of himem alone — more precisely, from >EBC8 to >FFD7. The UL file resides at >E9B0 when it is in memory and is allowed a length of >218 bytes. FWB does not define XOP 1 but the Editor/Assembler manual cautions that this is not necessarily available on all consoles, in any event.

If QD is to be called, then this extends down to >CE00 to cover QD and its RAM buffers. QD also keeps various information buffers in VDP on a temporary basis during use which might affect your program. VDP is used from >17FE through >19FF and from >2B70 through >3359. QD's memory usage may seem strange but remember that it has to live with XB LOAD, FWB itself, and the Assembler and Formatter, too.

The final area of memory that needs to be preserved is the FWB "mailbox" from >A000 t0 >A04F which is used to preserve and transfer the D/V80 workfile name. For program files this is most easily handled by AORGing your code suitably. XML addresses at >2000, >2002, and >8300 may be used with various modules and may need to be preserved depending on the details of your exit code. FWB programs always follows TI-Writer practice and preserve the contentd of >8300.

A dual-mode program will most likely be an E/A Option 5 Run Program File or its FWB equivalent, Loaders Option 2 GPL environment, which does not assume E/A utilities are present. As an alternative it could be a D/F80 Object File for Load and Run, but we'll discuss the program file format here.

The code examples will be drawn from Linehunter so that they reflect an actual working program.

The extracts, with nonessential minor changes, from the LH source which follow cover a number of complexities which may not always be found. The chief of these is that FWB, E/A, and QD work in Graphics mode while LH is in Text mode. Some of the details are of course specific to LH.

```
*  Funnelweb system equates
QDLEN      EQU     >A40
MLBOX      EQU     >A000
QDPOS      EQU     >CE00
QDSAV      EQU     >FF20
QDLOAD     EQU     >FF3C
QDXFL      EQU     >FF52
CMSRET     EQU     >FF5C
FWREGS     EQU     >FF7C
LDR11      EQU     >FF9C
*  Loader entry point
           AORG    >A050
           B       @START
*  Data block
MAINWS     BSS     >20
SAVRET     DATA    0
QDFL       DATA    0
FWEB       DATA    0
VDPR7      DATA    >F487
DSID       TEXT    'DS'
K          TEXT    'K.'
B20        BYTE    >20

AID        BYTE    >01
           EVEN
*
*  Real entry
START      LIMI    0
           MOV     R11,@SAVRET
           C       R11,@LDR11
           JNE     NOFWEB
           SETO    @FWEB
           LWPI    MAINWS
           MOV     @QDSAV,@QDFL
           JNE     STARU
           MOV     @QDLOAD,R1
           BL      *R1
           DATA    'QD'
           DATA    QDPOS,QDLEN
           JMP     STARU
           SETO    @QDFL
STARU      MOV     @QDFL,@QDXFL
           LI      R9,>380
           BL      @VAD
           MOVB    @VDPRD,@VDPR7
           BL      @LOADNB
*  Continue all paths
NOFWEB     EQU     $
```

First thing to note is that the code is AORGed above the mailbox area, and the first word is an executable instruction as required by generic program file loaders. Here it branches to the real start of the program. Just to be sure we turn off interrupts, and the interrupt hook should also be cleared as you can't guarantee that all loader programs clear this properly, though if you have to clear the hook, it is probably too late. This isn't necessary for E/A or FWB.

The next thing to do is to save the return address in R11. This is not strictly necessary if your final return doesn't use it, but here we do in order to cover as many exit paths as possible.

Now we determine if the program was loaded from FWB. The FWB program file loaders always hand over with R11 loaded with the re-entry address also stored as a data item at LDR11 (>FF9C). This is good enough to tell FWB from anything else, such as an E/A load, while the FWB code is still all or partly in memory. If the comparison fails the FWEB flag is left at its load time initialized reset state, and the code jumps to the common continuation at NOFWEB. If it is FWB, the flag is set, and the program's own register set loaded.

At this stage no registers have been written to, so even if the program had been loaded as FWB Option 3 with workspace at >20BA and had loaded over that, it would still function correctly.

When FWB loads a program it has no way of knowing if the QD code has been overwritten, so it always resets its internal QD flag at QDXFL (>FF52), but preserves it at QDSAV (>FF20), just in case the program loaded can use the information. The flag is saved and tested. If not null then QD was already loaded and we can jump to the next item. If null then we try to load it using the special utility file loader in FWB (used for things like QD, EA, UL, LL etc). A pointer to this routine is stored at QDLOAD (>FF3C) and it requires asinline data the 2-letter filename, the CPU load address, and the file length. This

(See Page 22)

PROGRAMMING IN FUNNELWEB—

(Continued from Page 21)

routine looks on both the primary and secondary FWB drives before giving up. The 6 bytes worth of file header details are ignored and the code is not executed as in a normal loader. The following JMP is the error return, and here it just steps over setting the QDFL flag. At label STARU the flag is returned to FWB so that if QD was already loaded or has been reloaded, then FWB knows it immediately on return and doesn't bother to reload QD if it is summoned.

Now the FWB color is read. This particular program happens not to have an explicit VSBR utility, so the first byte of the color table in normal E/A position is read and stored for use in setting text mode later, and for rewriting the E/A color table on final exit. Non-FWB loads will use the default value. This little bit of code helps give that seamless effect which we have been trying to achieve in the FWB system software.

The final subroutine call to LOADNB checks the mailbox and loads a valid name into various file name buffers. The actual code is further on. It checks that "DSK" and "." are there and if so transfers a block containing the name into various buffers. The drive number itself is not checked and the form RD. for Myarc RAMdisks has not been explicitly catered for, but could easily be added. Length byte derivation is done elsewhere in this particular program. The exact details are up to you.

```
*  Key returned in R0
*  AID call for QD
          CB    R0,@AID
          JNE   NXKEY
          ABS   @QDFL
          JEQ   NXKEY

          BL    @DPREP
          STWP  R0
          BL    @QDPOS
          BL    @RESUME
          JMP   WARM
NXKEY     CB    R0,....
```

This next piece of code is the call sequence for QD disk directory. The key routine has returned the key value in R0 with Keyscan 5 assumed. If QDFL is not set then the QD code is ignored.

The next routine, DPREP (code listed later), is necessary only to set up graphics mode for QD because LH is in text mode. Such a routine would not be necessary in a graphics mode program but, if sprites are defined, it would be necessary to shut them off and restore them on return. QD saves and restores the graphics screen >0->2FF itself but more is needed for text screens. Sprites are shut off, graphics mode set, and the E/A color table rewritten.

The next instruction, STWP R0, is a special one for QD, to render harmless an action set up for some special load paths. Then BL to the QD code at QDPOS (>CE00). QD does not interfere with the caller's workspace. The RESUME routine restores the full text mode screen, resets text mode, and in this application writes the marked filename into the buffers. After return it branches where needed.

```
*  Exits
EXIT   ABS  @FWEB
               JNE  FWEX
       BL     @CLRSCR
       BL     @EXITEA
       LWPI   GPLWS
       MOV    @SAVRET,R11
       B      @>6A
*      FWB    exit path
FWEX   LI     R1,>50
FWEX05 MOVB   @B20,@MLBOX-1(R1)
       DEC    R1
       JGT    FWEX05
       LI     R1,25
FWEX10 MOVB   @FIL2-1(R1),@MLBOX-1(R1)
              DEC   R1
       JGT    FWEX10

       LWPI   FWREGS
       MOV    @CMSRET,R11
       SETO   R13
       SETO   R4
       RT
```

This code excerpt shows the exits. If the FWEB flag is not set then it executes a normal return. There are enough different loaders around now which do not handle standard exits gracefully that the simplest thing might just be a return to the title screen.

The first instructions set graphics mode again to make the return to E/A look smoother. The FWB return first sets the FWB workspace, and to leave fetches the central menu screen return, and does some housekeeping in the funnelweb's burrow. SETO of R13 is necessary and reminds FWB that it was a utility load and not a return from the Editor for example. SETO of R4 flags the E/A side of the Central Menu screens and CLR of R4 would re-enter on the TI-Wr side.

The other code before the exit writes spaces in the mailbox out to >A050, and then returns a filename from a buffer. In this case the filename is already padded out with spaces, otherwise the length byte would be used. Again, this is specific to this particular program, and you would write code to suit your own.

```
 *  VDP address set
VAD      EQU     $
         SWPB    R9
VAE      MOVB    R9,@VDPWA
         SWPB    R9
         MOVB    R9,@VDPWA
         RT
*  Directory prepare
DPREP    EQU     $
         BLWP    @VMBRD
         DATA    >300,HIBUF,>1C0
EXITEA   MOV     R11,R10
         LI      R0,>D000
         BLWP    @VFILL
         DATA    >300,1
         LI      R9,>E081
         MOVB    R9,@>83D4
         BL      @VAE
         MOVB    @VDPR7,R0
         BLWP    @VFILL
         DATA    >380,>20

         B       *R10
*  Resume after QD
RESUME   EQU     $
         MOV     R11,R10
         BLWP    @VMBWD
         DATA    >300,HIBUF,>1C0
         LI      R9,>F081
         MOVB    R9,@>83D4
         BL      @VAE
         BL      @LOADNB
         B       *R10
*  Load filename buffers
LOADNB   EQU     $
         C       @MLBOX,@DSID
         JNE     NOFN
         CB      @MLBOX+2,@K
         JNE     NOFN
         CB      @MLBOX+4,@K+1
         JNE     NOFN
         LI      R1,25
BOXNM    MOVB    @MLBOX-1(R1),@FIL1-1(R1)
         MOVB    @MLBOX-1(R1),@FIL2-1(R1)
         DEC     R1
         JNE     BOXNM

         NOFN    RT
```

This has been an example where FWB remains in memory. The next article will look at code for reloading FWB when you know your code will destroy FWB in memory.

MICRO-REVIEWS

Game Writer's Toolkit, Brukin Golf, Widgit

By CHARLES GOOD

GAME WRITERS TOOLKIT by Mark Wills of Abbots Software

Last month I described software that may cause some of you to dust off a long unused bit of TI hardware, the cassette cable. This month I am reviewing software that uses another little used piece of TI hardware, the Mini Memory cartridge. Many of us have one of these cartridges gathering dust since it was purchased in 1983 or 1984. You may be suprised to learn that your Mini Memory's battery may still work. I have two Mini Memory cartridges, both with working 1983 batteries.

Game Writers Toolkit is assembly software that you load into the Mini-Memory, either from disk using the Load and Run option or from cassette tape using Easy Bug's Load option. Once loaded you can put your master disk or cassette tape away because the software stays in the Mini Memory's battery backed memory. The software provides numerous functions that can be CALL LINKed into any TI BASIC program. That's right, I said "TI BASIC". The Toolkit adds many of the important screen display capabilities of TI Extended BASIC to the TI BASIC environment and allows you to do some things that are not normally possible in either of these BASIC versions.

Because many of these enhanced TI BASIC capabilities relate to sprites, the product is called a "Game Writers" Toolkit because sprites are most often found in games. However many of the Toolkit's features will find uses in non-game TI BASIC software. You can play with 32 sprites at a time. Extended BASIC only allows 28 sprites to be simultaneously displayed at one time. Anything that can be done with sprites in XB can also be done in TI BASIC with the Toolkit.

The demo programs that come with the Toolkit are really incredible! Over the last decade I have seen many graphic display demo programs designed to run on our computers, so I have a pretty good idea of what the old 99/4A is supposed to be capable of graphically. I was astounded by the Toolkit graphic demos. I have rarely before seen on our computer screen objects change color and shape so rapidly. I had no idea this sort of blinding graphic speed was possible from a BASIC environment on the 99/4A. And the code for all of this is in easily understandable single line TI BASIC statements.

> **The demo programs that come with the Toolkit are really incredible! I have rarely before seen on our computer screen objects change color and shape so rapidly. I had no idea this sort of blinding graphic speed was possible from a BASIC environment on the 99/4A.**

In addition to sprites, here are some of the other things you can do with the Toolkit using the appropriate CALL LINKs. You can define a portion of the screen as a window and then scroll everything in the wondow up/down/left/right very rapidly independent of the rest of the screen. You can increment the color set number by one in all the 16 color sets sumultaneously, defining how many times you want to increment the color set and how long a delay between color set changes, using any number between 1 and 65535! This produces absolutely blinding kaleidoscope effects. Color sets can be made to change so rapidly that either the scan rate or the phospors on the inside of my Commodore 1702 monitor's screen can't always keep up, resulting in whiteish flikering screen objects that never stay one color long enough to actually show that color.

You can capture part of the screen (or the whole screen) and store it in the Mini Memory. Later you can retrieve this image and display it anywhere on the screen, not just where it was originally positioned. You can display the retrieved image normally or distort the image in various ways. A true lower case character set is available as is a larger than normal set of digits and upper case letters. XB's POS statement is emulated. So is DISPLAY AT, but the Toolkit's version of this is better because you can display a string of up to 255 characters. You can draw any rectangular shape positioned anywhere on the screen outlined by any ASCII character. You can play chimes and you can exit to the title screen from a running program. (BYE normally only works from command mode.)

I hope this product isn't too late on the market to be significant. A lot of effort obviously went into developing the Toolkit and the author deserves some recognition in the form of purchases. Unfortunately, most routine user programming these days is in XB and many 99/4A owners who acquired their systems after 1983 do no have a Mini Memory. The Game Writers Toolkit is commercial software distributed by Mike Goddard Computer Support, "Sarnia", Cemetary Road, Rhos, Wrexham, Clwyd LL14-2BY, United Kingdom. It costs $10 (cash, or international money order) which includes airmail delivery.

BRUKIN GOLF by Brukin Software

I have seen several public domain Extended BASIC "aim and shoot" golf

(See Page 24)

MICRO-REVIEWS—

(Continued from Page 23)

games for the TI. With these games you see a generic golf course on the screen along with the location of your golf ball. You then point golf club in the desired direction, hit the ball, and hope for the best. Brukin Golf is much more than just "aim and shoot". Many options are available and you play golf holes mapped from actual golf courses. One to four players can play. You need a SSSD disk system, a joystick, and Extended BASIC.

At the beginning of each player's turn you see a map of the course along with the location of the player's ball. Using the joystick you aim your shot for a particular point on the course by moving a cursor to that point. If your aim point is off the screen, the screen scrolls up/down and left/right to accommodate you. Nine holes the course map are in memory at one time for viewing and aiming. You play the front nine and then load the back nine for additional play. Screen graphics are sort of blocky, but everything is proportioned properly on the maps and it is easy to figure out what all the screen objects mean. When aiming you have to realize that the ball may not land at your aim point. First of all you are likely to hook or slice on long shots. Also the wind, whose direction and approximate strength are shown, will affect where the ball lands. You also have to take into consideration trees. Balls may go over or (rarely) through trees (depending on tree size, club used, and distance from tree), or hit a tree and come to rest under the tree. Water hazards, roughs, and sand traps also exist, each with specal characteristics that may limit your next shot in some way. These all have to be considered whem aiming shots.

Once a shot is aimed you select a club. There are 16 choices of woods, irons, and wedges. Your choice is mainly determined by how far and high you want to hit the ball. At various times you have the option to take a "chip" shot instead of a regular shot. With long puts you have the option of aiming directly for the hole and risking greatly overshooting the hole, or aiming to be just near the hole for an easy second putt. If you want to pretend to be a super talented golfer you can optionally give yourself enhanced capabilities in any or all of these areas; power, putting, chipping, and accuracy. These enhanced capabilities can also be used to balance play between individuals of different skill levels.

My only complaint about game play is that you have to keep turning the Alpha Lock on and off. You need it off to aim the ball with the joystick, but you need it on to properly select a golf club since club designations use upper case letters.

Brukin Golf is shareware. The regular registration fee is $5. If you send Brukin $10 they will send you by return mail the latest version of the game and notify you of future updates. Multiple copyright dates on Brukin Golf and on Widget (described below) suggest that these games have in fact been upgraded several times in the past. Send your registration to Brukin Software, 7919 Mitchell Farm Lane, Cincinnati OH 45242. If you just want to try the game before registering, I will send you both Brukin Golf and Wiget on one SSSD disk if you send me $1. This pays for the disk, postage, mailer, and my Florida vacation fund. A sample 18 hole course is included on the Brukin Golf disk. If you are a registered owner you can purchase directly from Brukin maps of additional golf courses that can be loaded into Brukin Golf. These include Augusta, Merion, Oakmont, and "The Toughest 18 Holes in Golf". You get all four of these for $8.

WIDGET
by Brukin Software

This is an all text free market business simulation game written in TI BASIC. It resembles a somewhat similar public domain game for the TI called Hamurabi. In Widget from 2 to 6 players act as corporate executives, competing against each other in an open market simulation purchasing the same raw materials and trying to sell the same products.

The computer starts the game by giving each player the same amount of starting capital and listing quantities of raw materials available for sale. Each player in each game cycle first writes on a paper secret bid prices for raw materials. These are then simultaneously revealed and entered into the computer. The computer adjusts the price of the available raw materials based on the bids, and sales are made only to those who bid above the computer's calculated minimum price. Each player them assembles the materials into products and secretly writes down an offer to sell these products at a specified price and quantity. These offers are simultaneously revealed by all players and entered into the computer. The computer then decides on a maximim price it will pay and all offers at or under this price are accepted. The game lasts for 5 years (60 cycles) and the object is to accumulate the most money.

A number of adjustments to game speed and parameters are possible. These are spelled out in detail in the on disk game documentation. You can specify the amount of time (in minutes) allowed for each player to decide bids for raw materials and products. The minimum amount of product and and raw material that results in stable prices can also be altered prior to the game.

If you enjoy the world of business and markets then you may like this game. It is shareware, and the requested registration fee is $5. If you are interested in trying before buying I will send it to you for $1 along with Brukin Golf, as detailed above.

I look forward to recieving your software for reviewing in this column. I especially look forward to meeting many of you personally Friday evening May 13 and all day May 14 at the all TI/Geneve Lima MUG conference. My address is P.O. Box 647, Venedocia OH 45894. You may send internet e-mail to me at cgood@magnus.acs.ohio-state.edu and you can phone me most evenings at 419-667-3131.

CFORM

A must-have program for Geneve hard disk users

By GREG KNIGHTES

This article originally appeared in the newsletter of the B.C. 99ers user group.

REVIEW

CFORM is a new hard drive formatting program written by Mike Maksimik and distributed by Cecure Electronics Inc. It was written in C using TIC version 1.67 and executes from the MDOS prompt on the Geneve using MDOS 1.23H or higher. It cannot be used with the TI99/4A. It is priced at $15, plus $3 shipping.

What makes this product so exciting is the ability to format your hard drive from MDOS and gain extra storage space. This is accomplished by formatting each hard drive cylinder to 34 sectors instead of 32, which is what Myarc Disk Manager 5 uses. This is an effective increase of a little more than six percent. I have a Miniscribe 3650 40-megabyte hard drive. When I format with MDM5, I get approximately 144,000 sectors. Using CFORM, I get approximately 165,000 sectors.

CFORM comes on a write-protected disk and is individually prepared for each user. The user's guide states that this software is for your use alone and is not to be used on any system or equipment other than your own, unless prior arrangement is made with Cecure Electronics. This is to protect the author and distributor from piracy, and it states their legal rights. I strongly agree with these statements and encourage you to buy the software. You won't be disappointed. It's like buying another hard drive for a small fee.

The CFORM floppy also includes Beery Miller's Win-Drive standalone, single-task windows driver and a subdirectory containing some files of formatting specs for many hard drives. Unfortunately, my original copy crashed after I made the backup and I wasn't aware they were in a subdirectory. An options file for your setup is created after the program is run.

To run the program, you have to have your Myarc Hard and Floppy Disk Controller upgraded to the 62256 static RAM chip. If you have the 6264 static RAM chip, contact Cecure Electronics about upgrading. You also need a minimum of 128K of RAM available. If you don't have a Memex card, keep TI Mode off and run only a minimal RAMdisk. You will also need at least one floppy drive and, of course, a hard drive. A color monitor is recommended as CFORM has a beautiful graphical interface. Monochrome may be difficult to read. Optionally, you can use a Myarc mouse or Logitech bus/serial mouse with Bruce Hellstrom's mouse driver. I have never been able to get my Asgard mouse to work in the MDOS environment. Also, a printer for recording your formatting specifications, along with bad sector information, is handy but not necessary.

CFORM looks similar to Al Beard's Genbench Shell and Backup Miser programs. CFORM may be run from Shell, but will run slower than if run from the MDOS prompt. I don't see any reason to have this program installed on the hard drive as, once it is run, it will be lost.

Before you run CFORM, make sure you are using a backup copy and make sure you have a bootable floppy of SYSTEM/SYS. I made the mistake of making a backup copy of SYSTEM/SYS but I found out *after* I reformatted that it was a bad copy. Take the extra time to reboot from that floppy, with your hard drive turned off, to make sure it works. Also, make sure you have a way of restoring your files to the hard drive once it is reformatted. I used Backup Miser to backup my hard drive (120 floppies worth). You might also want to have available on floppy disk Clint Pulley's Directory Manager and Sector One, just in case. Oh yeah,, you have Genbench Shell and Backup Miser on a floppy also, right?

Okay, now you're ready to reformat. Make sure you run Win-Drive first. I made the mistake of running CFORM without first installing Win-Drive. The drop-down menus didn't drop and the Crystal Software logo overlapped the menu selections. Also, change the MDOS prompt to the drive or directory that you will be running CFORM from. If you normally boot from your hard drive and exit your AUTOEXEC file leaving you at the D> prompt and you want to run CFORM from floppy drive No. 1, change to the A> prompt. MDOS has to be in the directory that you want to run CFORM from. Type in A:CF from the D> prompt won't work.

When CFORM has finished loading, you will be presented with a screen with three windows. Across the top are your option menus. In the bottom righthand side is your program registration information. This stays on-screen all the time while the program is running. Various other menus appear and disappear during various stages of program execution.

The first option from the left menu is Format. Next are Verify, Setup, Info, Help and Quit. Each of these options has a pull-down menu that is activated using a mouse or arrow keys. The first time you run the program, you will probably want to run Setup. The choices for this option are Colors and System. Almost all the color choices can be changed, except for the red highlight bar. The changes take effect after you exit Setup.

Setup is what you should do first. You need to give CFORM some information about your system. First and most crucial is the CRU base of your HFDC. If you don't tell CFORM the correct address, any access of the hard drive will lock up your system. I've done it more than once. You can also enable or disable the mouse.

CFORM has three Verify methods that you can choose: Turbo, Fastcomp and Slowcomp. After the format is complete, Turbo will verify but not update the sector bitmap if bad sectors are found. Choose this if you want to see if a drive is functional or the interlace setting is good. Fastcomp will update the bitmap but will not display the head and sector progress.

(See Page 26)

CFORM—

(Continued from Page 25)

Slowcomp will display all the information: cylinder, head and sector as each changes. Although the information is updated very quickly, displaying all that information is time-consuming. I recommend Fastcomp initially. If you get a lot of bad sectors, try changing the step-rate and/or interlace and reformat with Turbo. If you find a better combination, write it down and reformat with Fastcomp to update the bitmap. If nothing seems satisfactory, format with Slowcomp and make sure you get a printout. Slowcomp is slow. It took 6.5 hours to format my drive with Slowcomp! I did not time Turbo or Fastcomp.

You also need to tell CFORM what your printer's name is. Any legal MDOS or device name can be used. This means that you can send the output to a parallel or serial printer, or a disk file for later editing. You also have the option of disabling the printout. An illegal device name will cause an error and disable the report. You can also specify the initialization string for your printer.

After all this work setting up CFORM, another window pops up and you are given the option of saving these settings. I would suggest using this option the first time.

When you are done with the setup, you can go through the other options. "Info" is an online help system that answers many questions without having to refer to the manual.

Verify will check a hard drive that you have connected. The verify mode used is whatever you have specified in your Setup file. This is useful for checking the condition of a drive that you have formatted. If you experienced power failure or movement of your drive while writing data (from an earthquake maybe), you can check to see if any physical damage was done. Slowcomp will print error codes for each sector, if they are encountered. Verify will not proceed if there is an error in the first head, first cylinder.

Finally, the main part of the program. Formatting. This is why you bought the program, isn't it? You will be asked to fill in the blanks for a number of parameters for your drive. You will be asked for the number of cylinders, the number of heads, write pre-compensation, reduced write current, step rate, number of sectors per track, sector interlace, and the name you want to give your drive. There are a few things that you can alter, but most are determined by your drive. Unlike MDM5, CFORM can write 34 sectors per track, instead of only 32. This is where the extra storage space comes from. You can also change the interlace to get better performance. The recommended interlace for MDOS 1.50H is seven. If the drive is to be used in a TI99/4A, use an interlace of 22.

Mike Maksimik has done an outstanding job writing this utility. This has to be one of the most useful MDOS programs around. In conjunction with Gen-Bench Shell, Backup Miser, Disk Manager and Sector One, CFORM rounds out the stable of must-have software for the Geneve.

When you press Enter after entering the name of the drive, there is one last check. You must type the word "FORMAT" to start the format process. Pressing ESCape will abort the process at this point and let you start over if you need to change something. If you do not have the correct CRU address in the setup file, your computer will freeze and you will have to reboot and try again.

The only thing that I wish was included in the package but was not is an explanation of the error codes upon verifying. I input some horrendously stupid numbers by mistake and started getting error codes printed out. Well, after a couple of pages I said 'enough of this.' I powered-down by using CTRL-C to abort the format. Some errors that I got were "Sync error," "ECC correction attempted and CRC/ECC error detected," and "CRC/ECC error detected." It wouldn't have made any difference if I knew what these meant because there is nothing that I could have done about them. The only thing to do is to reformat properly.

Mike Maksimik has done an outstanding job writing this utility. This has to be one of the most useful MDOS programs around. In conjunction with Gen-Bench Shell, Backup Miser, Disk Manager and Sector One, CFORM rounds out the stable of must-have software for the Geneve. Mike Maksimik can be reached at: 635 Mackinaw, Calumet City, IL 60409-4014; 708-891-2513.

CFORM can be ordered from Cecure Electronics, P.O. Box 132, Muskego, WI 53150; 414-679-4343 (voice); 414-679-3736 (fax); 414-422-9669 (BBS). The price is $15, plus $3 shipping.

1994 TI FAIRS

FEBRUARY

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

MAY

Lima Multi User Group Conference, May 13-14, Ohio State University Lima Campus, Lima, Ohio. Contact Lima Ohio Users Group, P.O. Box 647, Venedocia, OH 45894.

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

USER NOTES

Horizon RAMdisk mods improve operation

Readers who undertake any of the following modifications do so at their own risk—Ed.

The following item has appeared in a number of TI user group newsletters.

Here are a couple of modifications that you can make to your Horizon RAMdisk to improve its performance.

The first 600 serial numbers have two red LEDs. The top LED should be green or yellow to protect the U11 chip during power-up. The first 1200 serial numers may have too high a resistance value in series in the battery circuit. It should be approximately 66 ohms total, or two 33-ohm resistors — one at each end of the battery string, to ensure a proper charge rate for the NICADs.

The first modification replaces the power-up circuit on the card by wiring it into the console reset line. The modification consists of removing one diode, one resistor, and one capacitor and wiring pin 6 on the card edge to the positive side of the capacitor connection. Edge pin 6 is the third connector from the right on the component side of the card (with the edge connector toward you).

On Horizons with serial numbers below 100, remove C8, CR2 and R2 and connect the wire from pin 6 to the front (or left) hole of the R8 location.

On Horizons with serial numbers above 100, remove C1, CR3 and R5 and connect the wire from pin 6 to the positive side of the C1 location.

The second modification is a card-disable switch to hide the card from the rest of the system. The switch allows you to turn off the RAMdisk in the event of a system crash, or whenever the computer locks up. It preserves the contents of the card and eliminates the need to remove the batteries to regain control of the card.

Thel modification is simple. One cards with serial numbers 1999 and below, remove the voltage from pin 6 of U20 (U20A on HRD+ serial numbers above 1999) and reconnect it via a resistor (1-10K) through a SPST (single-pole single-throw) switch to ground. Closing the switch pulls the pin low and shuts off the CRU access at U20. With Horizons that include a 32K memory modification, use

(See Page 28)

NEWSBYTES

'Constitution Reader' available for TI

Constitution Reader V.1.1 has been produced by Machicolation Systems, formerly Software And More. The 100 percent assembly program requires 32K and an Editor/Assembler loader and contains the entire text of the United States Constitution, allowing the user to read any given part of the text or search for a string. The disk-based program sells for $14.95 plus $1 shipping and handling., payable by check

Machicolation Systems, which recently changed its name, continues to offer the Extended BASIC titles formerly offered by Software and More. Titles include Grafiks and Music V2.2, Musical Christmas Tree, List of Labels Labeler, Valentine Card, Telephone Dialer and Flags and Map.

For further information or to order, write Machicolation Systems, c/o Sam Carey, 5820 S.E. Westfork St., Portland, OR 97206-0742.

Lima conference attracts vendors

A number of vendors have made requests for tables and seminars for the 1994 Lima Multi Users Group Conference May 13 and 14 at The Ohio State University's Lima campus.

Seminars are scheduled from Asgard Peripherals, Bruce Harrison, Beery Miller (9640 News), Bud Mills, MS Express, Barry Traver and Mike Wright (PC99).

Vendors not presenting seminars, but requesting exhibit tables ,include the Cin-Day Users Group, Competition Computer, L.L. Conner Enterprises, the Mid-South Users Group and RamCharged Computers.

The conference is free to vendors and attendees alike. For additional information, write the Lima Users Group, P.O. Box 647, Venedocia, OH 45894; phone Dave Szipple evenings at (513) 498-9713; Phone Charles Good evenings at (419) 667-3131; or send Internet messages to cgood@magnus.acs.ohio-state.edu.

Delphi eliminates storage charges

Delphi Internet Services Corporation has eliminated additional storage charges of electronic mail messages and data files sent and received online.

Russell Williams, Delphi vice president and general manager, Delphi has invested in a new technology to handle file management tasks automatically without restrictive size limits on mail messages, per-message fees, mailbox size limitations or monthly allocations.

Rather, with the new system, messages will be deleted automatically after 30 days if they have not been read or after seven days if they have. Files retrieved by a Delphi member from an Internet site will be held in the member's "Workspace" for up to 48 hours to permit the member to transfer them to his own computer.

Home computer users may join Delphi by dialing by modem 1-800-365-4636. Once connected, press "Return" several times. At the "Username" prompt, enter JOINDELPHI and at the "Password" prompt enter GALERIA to register for the service. For voice assistance, call Delphi service representatives at 1-800-695-4005.

Send your information about products and services for the TI/Geneve community to MICROpendium Newsbytes, P.O. Box 1343, Round Rock, TX 78680.

USER NOTES

(Continued from Page 27)

U20C.

Bend pin 6 of the chip out and attach enough wire to reach the switch and connect the resistor from pin 6 to pin 16 of the same chip. Run the write to the switch that you have mounted at the top edge of the card. Run a lead from the other pole to a nearby ground and you are finished.

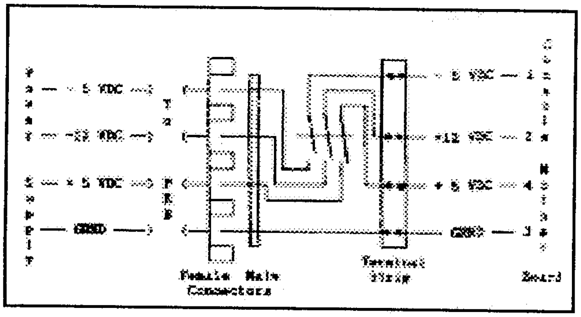
HRD+ circuit boards with serial numbers below 1999 required the stacking of chips at U20. Attach the wire and resistor to pin 6 of the top chip and cut off the bottom end.

Another solution to hot consoles

Readers who undertake any of the following modifications do so at their own risk—Ed.

The following was written by Matt Matthews of the Southwest 99ers user group. It appeared in the group's newsletter.

Okay, now that you have your PEB modified with a high-wattage switching power supply (IBM-style), it's time you got rid of that coffee warmer that resides inside your TI console. You know, the one that is always causing your computer to lock up due to power fluctuations and heat dissipation.



Here is the way to do it, with recommendations from Mike Doane, the Southwest 99ers Cool Card and Console Guru.

Remove the bottom of your console. If you don't know how, you shouldn't be trying this without help.

- Remove the two screws that secure the old power supply to the case.
- Disconnect the molex junction which connects the power supply to the motherboard.
- Life out the receptacle at the rear (where the cord from your wall transformer plugs into your console)(and set the power supply aside.

The schematic shows the voltages supplied to the molex plug are as follows. It is recommended that you double-check with a voltmeter. Remember, one of the lines is a negative 5 volts.

Pin #1	-5 volts
Pin #2	+12 volts
Pin #3	Ground
Pin #4	+5 volts

You will need the following items:

- A length of 4-conductor wire of sufficient length and size to carry the current which is required. I believe a 4-wire telephone cable will do it.
- A 4-pin male pl;ug and matching female receptacle for connecting your power from the PEB.
- A TPST (triple-throw single-throw) switch must be used if you wish to retain your console's on/off switch. If you do not want this option, your console with turn on and off with the PEB switch.

I used a short piece of junction block glued to the inside top of the console to make my connections. It is not a necessity, just a neat way to make the desired connection.

I got my hardware at an Electronic City, but I am sure any electronics store would have at least the first two items. The on/off switch is a hard one to find. I have the switch, Mike does not; and he has no trouble using the PEB switch. My console is completely cool and, as an added bonus, I don't have the wall transformer or its plug to contend with.

Envelope printer

The following program prints envelopes using a dot-matrix printer. It was written by Harry Allston.

Line 140 includes a default return address. Modify the line to include the return address you want to us. The return address may also be changed when the program is running. The program prompts you not only for changes to the return address but for the address of the recipient. It then outputs the information to a printer.

```
1 !SAVE DSK1.MICRO-E-M !121
2 ! My Own Program 3-1993 !0
11
3 ! Files of Harry Allston
   209-638-7523 !213
4 ! RELEASED TO PUBLIC
   DOMAIN ON 01-06-1994 !08
7
100 CALL CLEAR :: CALL BLUE
:: CALL PRINTON !252
110 OPEN #1:"PIO",VARIABLE 1
36 :: PRINT #1:CHR$(27);"x";
CHR$(1);CHR$(27);"8" !011
120 LM$=CHR$(27)&"1" !221
130 DISPLAY AT(2,5)ERASE ALL
:"ENVELOPE-PRINTER" !193
140 DISPLAY AT(5,1):"MICROpe
ndium":"PO Box 1343":"Round
Rock":"TX":"78680" !237
150 DISPLAY AT(12,1):"Name":
"Address":"City":"ST":"Zip #
" !088
160 ACCEPT AT(5,1)SIZE(-25)B
EEP:N$ :: IF N$="" THEN 160
!034
170 ACCEPT AT(6,1)SIZE(-28)B
EEP:A$ :: IF A$="" THEN 170
!022
180 ACCEPT AT(7,1)SIZE(-25)B
EEP:C$ :: IF C$="" THEN 180
!034
190 ACCEPT AT(8,1)SIZE(-2)BE
EP:S$ :: IF S$="" THEN 180 !
013
200 ACCEPT AT(9,1)SIZE(-10)V
ALIDATE(DIGIT,"-")BEEP:Z$ ::
 IF Z$="" THEN 180 !072
210 ACCEPT AT(12,1)SIZE(-25)
BEEP:N2$ :: IF N2$="" THEN 2
10 !231
220 ACCEPT AT(13,1)SIZE(-28)
BEEP:A2$ !100
230 ACCEPT AT(14,1)SIZE(-25)
BEEP:C2$ :: IF C2$="" THEN 2
30 !231
240 ACCEPT AT(15,1)SIZE(-2)B
EEP:S2$ :: IF S2$="" THEN 24
0 !220
250 ACCEPT AT(16,1)SIZE(-10)
```

(See Page 29)

USER NOTES

(Continued from Page 28)

```
VALIDATE(DIGIT,"-_")BEEP:Z2$
 :: IF Z2$="" THEN 250 !129
260 DISPLAY AT(20,3):"TAB FO
R RETURN----->" :: ACCEPT AT
(20,23)SIZE(2)VALIDATE(DIGIT
)BEEP:R$ :: IF R$="" THEN 26
0 ELSE R=VAL(R$)!133
270 DISPLAY AT(20,3):"TAB FO
R ADDRESS---->" :: ACCEPT AT
(20,23)SIZE(2)VALIDATE(DIGIT
)BEEP:AD$ :: IF AD$="" THEN
270 ELSE AD=VAL(AD$)!084
280 DISPLAY AT(20,3):"SPACES
 UNDER RETURN:" :: ACCEPT AT
(20,23)SIZE(1)VALIDATE(DIGIT
)BEEP:SP$ :: IF SP$="" THEN
280 ELSE SP=VAL(SP$)!118
290 PRINT #1:LM$&CHR$(R)!082
300 PRINT #1:N$:A$:C$;" - ";
S$ :: PRINT #1:TAB(15);CHR$(
27);"G";CHR$(27);"W1";Z$;CHR
$(27);"W0";CHR$(27);"H" !211
310 FOR X=1 TO SP :: PRINT #
1 :: NEXT X !117
320 PRINT #1:LM$&CHR$(AD)!13
3
330 PRINT #1:N2$:A2$:C2$;" -
 ";S2$ :: PRINT #1:TAB(15);C.
HR$(27);"G";CHR$(27);"W1";Z2
$;CHR$(27);"W0";CHR$(27);"H"
 !205
340 DISPLAY AT(23,1):"AGAIN?
 Y-N" :: ACCEPT AT(23,18)SIZ
E(-1)VALIDATE("YyNn")BEEP:AN
$ !186
350 IF AN$="Y" OR AN$="y" TH
EN 130 ELSE CALL CLEAR :: PR
INT #1:CHR$(27);"9" :: CLOSE
 #1 :: STOP !247
360 SUB PRINTON !154
370 DISPLAY AT(12,6)ERASE AL
L:"Turn on Printer!" :: OPEN
 #1:"PIO" :: PRINT #1:CHR$(2
7);"E";CHR$(7):: CLOSE #1 ::
 CALL CLEAR !243
380 SUBEND !168
390 SUB BLUE !149
400 CALL SCREEN(5):: FOR C=0
TO 14 :: CALL COLOR(C,16,5)::
NEXT C :: SUBEND
```

A puzzle in Extended BASIC

The following program, written by the late Jim Peterson, is a simple by challenging game called Old-Timer Puzzle. It requires Extended BASIC.

```
1 DATA 4,2,1,3,5,4,2,3 !088
2 DIM S(19)!126
10 GOTO 100 !179
11 N,D$,J,M1$,M2$,M$,TB,FL,V
$,C,P,N$,S(),B,K,ST,W@,M,D,D
IS$,W,T,X,Y,CH,TX !139
30 CALL CLEAR :: CALL SCREEN
 :: CALL CHAR :: CALL COLOR
:: CALL HCHAR :: CALL VCHAR
:: CALL SOUND :: CALL KEY !2
36
40 !@P- !064
100 CALL CLEAR :: CALL SCREE
N(16):: CALL CHAR(94,"3C4299
A1A199423C")!211
110 ! programmed by Jim Pete
rson 12/83, XBasic·version 7
/85 !056
```

(See Page 30)

USER NOTES

(Continued from Page 29)

```
120 ! COPYRIGHT 1983 TIGERCU
B SOFTWARE, 156 COLLINGWOOD
AVE., COLUMBUS OHIO 43213 !2
30
130 DISPLAY AT(2,6):"OLD-TIM
ER PUZZLE": :"TCX-1127 ^ Tig
ercub Software": :"  Make th
e red squares and" !125
140 DISPLAY AT(7,1):" white
squares change":" places. Yo
u can move one":" space or j
ump over one" !069
150 DISPLAY AT(10,1):" squar
e, but red can only":" move
right, white can only":" mov
e left.":"  Can be done in
  moves.": :!181
160 DISPLAY AT(15,1):"  Type
 FCTN 8 to start over":" or
FCTN 7 to see how it's":" do
ne." :: N=2 :: D$="FF8181818
18181FF" :: CALL CHAR(128,D$
):: CALL CHAR(136,D$)!188
170 CALL CHAR(137,"FFFFFFFFF
FFFFFFF"):: CALL CHAR(129,"1
8243C5A5A3C2418"):: CALL COL
OR(13,2,10):: CALL HCHAR(1,2
,129,30):: CALL HCHAR(24,2,1
29,30)!171
180 CALL VCHAR(2,2,129,22)::
 CALL VCHAR(2,31,129,22):: C
ALL COLOR(14,2,16)!044
190 FOR J=1 TO N :: M1$=M1$&
CHR$(128):: M2$=M2$&CHR$(136
)!088
200 NEXT J :: M$=M1$&CHR$(13
7)&M2$ :: TB=16-N !006
210 FOR J=1 TO LEN(M$):: CAL
L HCHAR(20,TB+J,ASC(SEG$(M$,
J,1)))!141
220 NEXT J :: IF FL=1 THEN 2
70 !184
230 V$=SEG$("123456789ABCDEF
GHIJ",1,N*2+1)!018
240 FOR C=TB+1 TO TB+N*2+1 :
: P=P+1 :: CALL HCHAR(18,C,A
SC(SEG$(V$,P,1)))!155
250 NEXT C :: P=0 :: P=0 ::
N$=STR$(N*(N+2)):: FOR J=1 T
O LEN(N$):: CALL HCHAR(13,19
+J,ASC(SEG$(N$,J,1)))!030
260 NEXT J !224
270 FOR J=1 TO N :: S(J)=1 !
209
280 NEXT J :: S(J)=2 :: B=N+
1 :: FOR J=N+2 TO N*2+1 :: S
(J)=0 !139
290 NEXT J :: IF FL=1 THEN 5
30 !189
300 CALL SOUND(10,600,0)!075
310 CALL HCHAR(22,22,63):: C
=22 !227
320 DISPLAY AT(22,2):"Which
number to move?" :: CALL KEY
(3,K,ST):: IF ST<1 THEN 320
ELSE DISPLAY AT(22,2):" " !2
06
330 IF K<>1 THEN 350 !032
340 FL=1 :: W@=0 :: GOTO 360
 !084
350 IF K<>6 THEN 370 !057
360 M=0 :: W@=0 :: GOTO 210
!119
370 IF POS(V$,CHR$(K),1)=0 T
HEN 320 !163
380 IF K>57 THEN 400 !207
390 M=K-48 :: GOTO 410 !186
400 M=K-55 !076
410 CALL SOUND(10,800,0):: W
@=W@+1 :: DISPLAY AT(21,25)S
IZE(3):W@ :: IF M<=N*2+1 THE
N 430 !103
420 CALL SOUND(200,110,0,-4,
0):: M=0 :: CALL HCHAR(22,22
,32,9):: GOTO 310 !138
430 IF ((S(M)=1)*(B<M)+(B-M>
2))THEN 420 !249
440 IF ((S(M)=0)*(B>M)+(M-B>
2))THEN 420 !249
450 IF M=B THEN 420 !239
460 GOSUB 470 :: GOTO 480 !2
17
470 CALL HCHAR(20,TB+M,137):
: CALL HCHAR(20,TB+B,136-S(M
)*8):: S(B)=S(M):: S(M)=2 ::
 B=M :: M=0 :: CALL HCHAR(22
,22,32,9):: RETURN !022
480 FOR J=1 TO N :: IF S(J)<
>0 THEN 310 !197
490 NEXT J :: IF B<>N+1 THEN
 310 !096
500 CALL SOUND(400,392,0,330
,0,262,0):: CALL SOUND(400,4
40,0,349,0,262,0):: CALL SOU
ND(400,494,0,392,0,294,0)!05
7
510 CALL SOUND(800,523,0,392
,0,330,0):: N=N+1 :: W@=0 ::
 IF N=10 THEN 800 !079
520 M1$="" :: M2$="" :: M$="
" :: GOTO 190 !216
530 ON N-1 GOSUB 570,580,590
,600,610,620,630,640 :: FOR
J=1 TO N*(N+2):: READ M :: G
OSUB 470 :: FOR D=1 TO 100 !
192
540 NEXT D !218
550 NEXT J :: FOR D=1 TO 800
 !254
560 NEXT D :: FL=0 :: GOTO 2
10 !072
570 RESTORE 650 :: RETURN !2
43
580 RESTORE 660 :: RETURN !2
53
590 RESTORE 670 :: RETURN !0
07
600 RESTORE 680 :: RETURN !0
17
610 RESTORE 690 :: RETURN !0
27
620 RESTORE 710 :: RETURN !0
47
630 RESTORE 730 :: RETURN !0
67
640 RESTORE 760 :: RETURN !0
97
650 !!131
660 DATA 5,3,2,4,6,7,5,3,1,2
,4,6,5,3,4 !048
670 DATA 6,4,3,5,7,8,6,4,2,1
,3,5,7,9,8,6,4,2,3,5,7,6,4,5
 !120
680 DATA 7,5,4,6,8,9,7,5,3,2
,4,6,8,10,11,9,7,5,3,1,2,4,6
,8,10,9,7,5,3,4,6,8,7,5,6 !1
74
690 DATA 8,6,5,7,9,10,8,6,4,
3,5,7,9,11,12,10,8,6,4,2,1,3
,5,7,9,11,13,12,10,8,6,4,2 !
051
700 DATA 3,5,7,9,11,10,8,6,4
,5,7,9,8,6,7 !173
710 DATA 9,7,6,8,10,11,9,7,5
,4,6,8,10,12,13,11,9,7,5,3,2
,4,6,8,10,12,14,15,13,11,9 !
156
720 DATA 7,5,3,1,2,4,6,8,10,
12,14,13,11,9,7,5,3,4,6,8,10
,12,11,9,7,5,6,8,10,9,7,8 !
84
730 DATA 10,8,7,9,11,12,10,8
,6,5,7,9,11,13,14,12,10,8,6,
```

(See Page 31)

USER NOTES CLASSIFIEDS

(Continued from Page 30)

```
4,3,5,7,9,11,13,15,16,14,12
!125
740 DATA 10,8,6,4,2,1,3,5,7,
9,11,13,15,17,16,14,12,10,8,
6,4,2,3,5,7,9,11,13,15,14,12
 !244
750 DATA 10,8,6,4,5,7,9,11,1
3,12,10,8,6,7,9,11,10,8,9 !0
95
760 DATA 11,9,8,10,12,13,11,
9,7,6,8,10,12,14,15,13,11,9,
7,5,4,6,8,10,12,14,16,17,15
!050
770 DATA 13,11,9,7,5,3,2,4,6
,8,10,12,14,16,18,19,17,15,1
3,11,9,7,5,3,1,2,4,6,8,10,12
 !252
780 DATA 14,16,18,17,15,13,1
1,9,7,5,3,4,6,8,10,12,14,16,
15,13,11,9,7,5,6,8,10,12,14
!056
790 DATA 13,11,9,7,8,10,12,1
1,9,10 !236
800 DIS$="18245AC342667E5A66
 FFDB66429981A5008124665A7E
C3DB7EA5E781249981245A001899
7EA5A5243CC31866BDDBA5C318"
:: CALL CLEAR :: FOR W=1 TO
5 :: FOR T=2 TO 14 :: RANDOM
IZE !079
810 X=INT(15*RND+2)!216
820 Y=INT(15*RND+2):: IF Y=X
 THEN 820 !254
830 CALL COLOR(T,X,Y):: CALL
 CHAR(24+T*8,SEG$(DIS$,INT(4
3*RND+1)*2-1,16))!015
840 NEXT T :: CH=40 :: TX=0
:: FOR X=1 TO 12 :: CALL HCH
AR(X,1+X,CH,29-X-TX):: CALL
HCHAR(25-X,1+X,CH,29-X-TX)::
 CALL VCHAR(X,1+X,CH,25-X-TX
)!098
850 CALL VCHAR(X,31-X,CH,25-
X-TX)!072
860 CH=CH+8 :: TX=TX+1 !233
870 NEXT X !238
880 NEXT W :: END !250
```

MICROpendium pays $10 for items from readers that are used in the User Notes column. Send your User Notes to MICROpendium User Notes, P.O. Box 1343, Round Rock, TX 78680.

Policy

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

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

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

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

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

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

FOR SALE FOR SALE

Try a MICROpendium classified to sell or buy TI/Geneve products

Buy/Sell Used Hardware and Software

National Used Software/Hardware Club has buyers looking for TI products, as well as sellers. More than 170 TI items are listed for sale. The membership fee is only $15/year and entitles you to buy or sell computer items via the NUS/HC database. You will also receive FREE our newsletter (mailed every 8-12 weeks). If you don't buy or sell anything during the term of your membership we will refund your annual fee. **For a free, no-obligation information packet, write to** NUS/HC, P.O. Box 1343, Round Rock, TX 78680; or call 512-255-1512. **Here are some of the items our members want to buy:**

Myarc HFDC	Offer	Triple Tech Card	75.00
GRAMKracker (80K)	95.00	Rave 105 Keyboard Interface	85.00
Super Space II	25.00	80-column card	Offer

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

The ONLY monthly devoted to the TI99/4A

Subscription Fees

❑ 12 issues, USA, $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.

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; Vol. 4, No. 9

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

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

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

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

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

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

GENEVE DISKS (SSSD unless specified)

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

❑ **GPL 1.5** $4.00

❑ **Myarc Disk Manager 1.50**...$4.00

❑ **Myarc BASIC 3.0**...$4.00

❑ **MY-Word V1.21** $4.00

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

GENEVE PUBLIC DOMAIN DISKS

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

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

SECOND CLASS