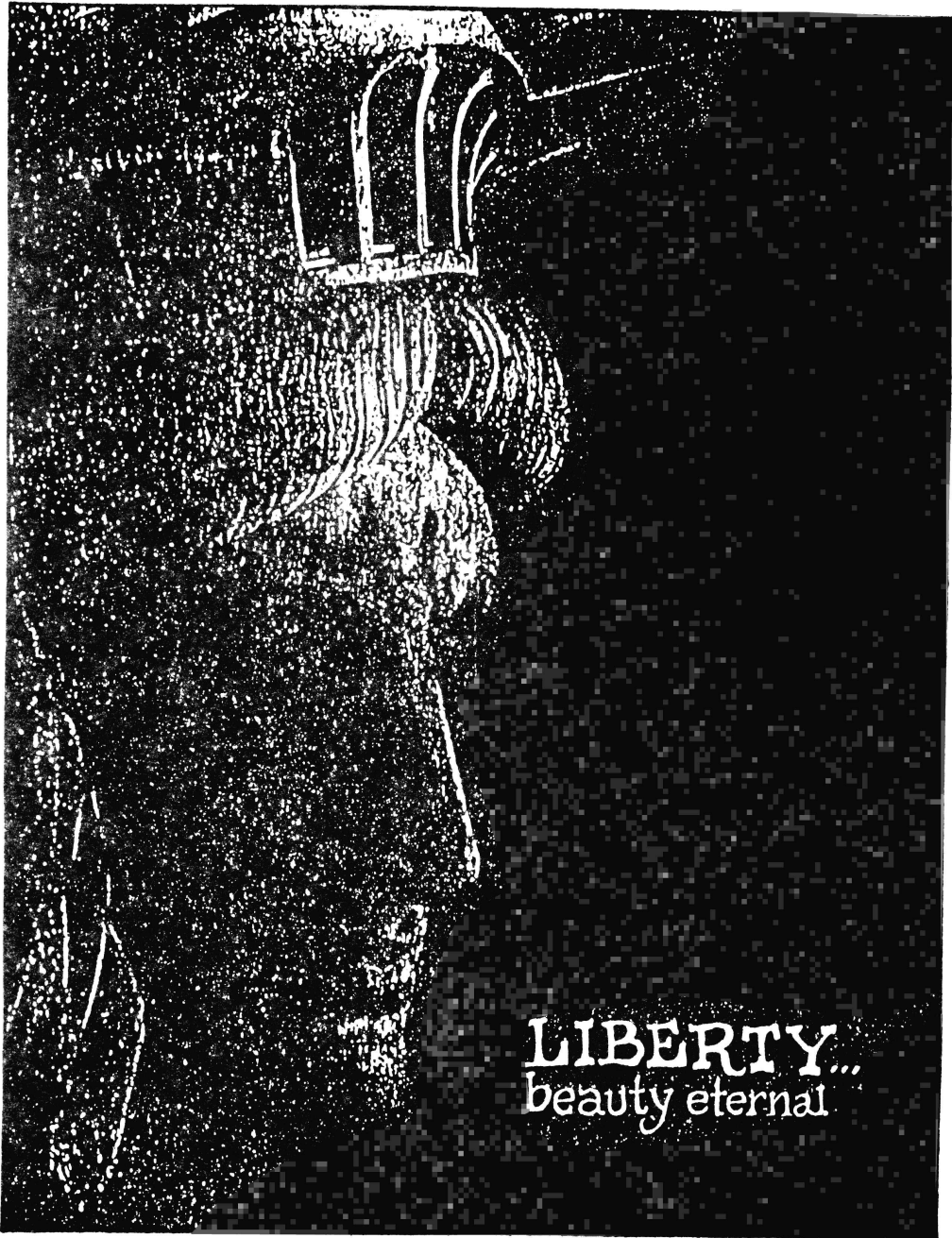


Spirit of 99



THE OFFICIAL NEWSLETTER OF THE CENTRAL OHIO NINETY-NINERS INC.

PUBLISHED MONTHLY IN COLUMBUS OHIO



LIBERTY...
beauty eternal

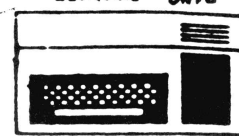
\$1.50

VOL. 5

NO. 7

JULY

1987



Spirit of 99

THE OFFICIAL NEWSLETTER OF CENTRAL OHIO SOCIETY-WOMENS

COMPUTER FLEA MARKET!

Our August Meeting this year will be a **COMPUTER-FLEA-MARKET** - - - held at the Electrical Workers Union Hall, located 23 West Second Avenue, Columbus, Ohio. The Flea Market will take the place of the normal Club meeting for the month of August. We'll be asking a sellers fee to help support the Club. The exact amount of the sellers fee will be discussed at the July Meeting. The Flea Market might be the perfect time to bring that hold-out friend who would never come to a meeting before. Be sure to contact some of our past club members. Maybe they have things to sell too!

Let's make the Flea Market a lot of fun and a success, too!

See you there - - - - Jim Seitz.

Membership dues are (\$20.00) and are payable yearly to C.O.N.N.I. and cover the immediate family. Please address it to:
EVERETT WADE
 179 ERIE RD.
 COLUMBUS, OHIO 43212.

Editorial, advertising and subscription address is!
 181 Heischman Ave.
 Worthington, Ohio 43085.

****INDEX****

- Best of 1985 Series.....P. 3
- Tips From The Tiger Cub..P. 4
- From Out 'N 'About.....P. 6
- From Out 'N About.....P. 7
- Jack Sughrue.....P. 8
- TI Character Codes.....P.10
- President.....P.11
- Best of 1985.....P.12
- Minutes.....P.15
- Letter to the Editor....P.16
- From the Archives.....P.18
-
-
-
- Editor.....Jean Hall
- Assistant.....Carole Parkins
- **OFFICERS****
- PRESIDENT.....Irwin Hott
- VICE PRES.....Jim Seitz
- SECRETARY.....Jere Singleton
- TREASURER.....John Cummings

Best of 1985 Series.

DISKO, a program that enables you to read individual bytes and sectors on a disk, was released into public domain by TI and is in our library. In addition, commercially available programs from Navarone and Miller's Graphics, among many, allow you to read disk contents, byte by byte. As many of our members have these programs, we feel the following article will be useful.

TI-99/4(A) Disk Format
=====

This article was received from Steven J. Royce, of the Western New York 99'ers Users Group. Thanks Steve!!

The following is a complete and, to the best of my knowledge, accurate description of the Disk Directory format and file storage allocation used by the TI-99/4(A) Earl Hall CompuServe ID - 72746,3244

SECTOR 0 - Volume Information Block

ESS	CONTENTS
0000-0009	Disk name - up to 10 characters
000A-000B	Total number sectors on disk (>0168=360, >02D0=720, >05A0=1440)
000C	>09 (# of sectors/trk)
000D-000F	'DSK' (>44534B)
0010	>50 = Disk backup protected, >20 = not protected
0011	# of tracks per side (>28=40, >23=35)
0012-0013	# of sides/density (>0101=SS/SD, >0201=DS/SD, >0202=DS/DD)
003B-end	Sector allocation bit map. See note below

NOTE on >003B-end: This is a sector-by-sector bit map of sector use; 1=sector used, 0=sector available. The first byte is for sectors 0 through 7, the second for sectors 8 through 15, and so on. Within each byte, the bits correspond to the sectors from right to left. For example, if byte >003B contained >CF00 then the first byte equals 1100 1111. This means that sectors 0 through 3 are used, sectors 4 and 5 unused and sectors 6 and 7 used. Information for the 2nd side of a DS/SD disk starts at byte >0065 and ends at byte >0091.

SECTOR 1 - Directory Link

Each 16-bit word lists the sector number of the File Descriptor Record for an allocated file, in alphabetical order of the file names. The list is terminated by a word containing >0000; therefore, the maximum number of files per disk is 127 [(256/2)-1]. If the alphabetical order is corrupted (by a system crash during name change, for instance), the binary search method used to locate files will be effected and files may become unavailable.

SECTOR >2 TO >21 - File Descriptor Records

ADDRESS	CONTENTS
0000-0009	File name - up to 10 characters
000C	File type: >01=Program(memory-image) >00=DIS/FIX >02=INT/FIX >80=DIS/VAR >82=INT/VAR
000D	File deletion protection invoked by Disk Manager 2 will be shown by >08 added to the above.
000E-000F	# of (MAXRECSIZE) records/sector Number of sectors allocated to the file. (Disk Manager 2 will list one more than this number, thereby including this sector in the sector count)
0010	For memory-image program files and variable-length data files, this contains the number of bytes used in the last disk sector. This is used to determine end-of-file.
0011	MAXRECSIZE of data file.
0012-0013	File record count, but with the second byte being the high-order byte of the value.
001C - end	Block Link (see note)

Note on file storage: Files are placed on the disk in first-come / first-served manner. The first file written will start at sector >0022, and each subsequent file will be placed after it. If the first file is deleted, a newer file will be written in the space it occupied.

If this space isn't big enough, the file will be 'fractured', and the remainder will be placed in the next available block of sectors. The block link map keeps track of this fracturing. Each block link is 3 bytes long. The value of the 2nd digit of the second byte followed by the 2 digits of the first byte is the address of the first sector of this extent. The value of the 3rd byte followed by the 1st digit of the 2nd byte is the number of additional sectors within this extent.

Sectors 2 through >21 are reserved for File Descriptor Records and are allocated for file data only if no other available sectors exist. If more than 32 files are stored on a disk, additional File than 32 files are stored on a disk, additional File Descriptor Records will be allocated as needed, one sector at a time, from the general available sector pool.

HAPPY BIRTHDAY

AMERICA

TIPS FROM THE TIGERCUB



#41

Copyright 1986

TIGERCUB SOFTWARE
156 Collingwood Ave.
Columbus, OH 43213

Distributed by Tigercub Software to TI-99/4A Users Groups for promotional purposes and in exchange for their newsletters. May be reprinted by non-profit users groups, with credit to Tigercub Software.

Over 130 original programs in Basic and Extended Basic, available on cassette or disk, only \$3.00 each plus \$1.50 per order for PPM. Entertainment, education, programmer's utilities. Descriptive catalog \$1.00, deductible from your first order.

Tigercub Full Disk Collections, just \$12 postpaid! Each of these contains either 5 or 6 of my regular \$3 catalog programs, and the remaining disk space has been filled with some of the best public domain programs of the same category. I am NOT selling public domain programs - my own programs on these disks are greatly discounted from their usual price, and the public domain is a FREE bonus!

TIGERCUB'S BEST, PROGRAMMING TUTOR, PROGRAMMER'S UTILITIES, BRAIN GAMES, BRAIN TEASERS, BRAIN BUSTERS!, MANEUVERING GAMES, ACTION REFLEX AND CONCENTRATION, TWO-PLAYER GAMES, KID'S GAMES, MORE GAMES, WORD GAMES, ELEMENTARY MATH, MIDDLE/HIGH SCHOOL MATH, VOCABULARY AND READING, MUSICAL EDUCATION, KALEIDOSCOPIES AND DISPLAYS

NUTS & BOLTS (No. 1), a full disk of 100 Extended Basic utility subprograms in merge format, ready to merge into your own programs. Plus the Tigercub Menuloader, a tutorial on using subprograms, and 5 pages of documentation with an example of the use of each subprogram. All for just \$19.95 postpaid.

NUTS & BOLTS NO. 2, another full disk of 108 utility subprograms in merge format, all new and fully compatible with the last, and with 10 pages of documentation and examples. Also \$19.95 postpaid, or both Nuts Bolts disks for \$37 postpaid.

TIPS FROM THE TIGERCUB, a full disk containing the complete contents of this newsletter Nos. 1 through 14, 50 original programs and files, just \$15 postpaid.

TIPS FROM THE TIGERCUB VOL. 2, another diskfull, complete contents of Nos. 15 through 24, over 60 files and programs, also just \$15

\$ TIPS FROM THE TIGERCUB \$
\$ VOL. 3 is now ready. \$
\$ Another 62 programs, \$
\$ routines, tips, tricks \$
\$ from Nos. 25 thru 32. \$
\$ Also \$15 postpaid. \$
\$ *****

\$ TIPS FROM THE TIGERCUB \$
\$ Vol. 4 is now ready. \$
\$ Another 48 programs, \$
\$ routines, tips, tricks \$
\$ from Nos. 33 thru 41. \$
\$ Also \$15 postpaid. Any \$
\$ two Tips disks for \$27, \$
\$ any 3 for \$35, all 4 \$
\$ for \$42, postpaid. \$
\$ *****

This will be the last issue of the Tips from the Tigercub.

I started this newsletter over 3 years ago, as a means of promoting my software business. It has never been a success for that purpose, but I have kept it going because of the many interesting newsletters that I have received in exchange, and the many friends that I have made around the world.

I know, from the editors' comments in many of your newsletters, that many of you are finding it difficult to finance a newsletter for your shrinking membership, and even more difficult to find the time, and the material to print. For a one-man user's group pretending to be a business which is getting very little business, it has become impossible. User group members have never been good customers for anyone's software, for reasons which you all know, and those who are remaining active in the TI world are wanting more sophisticated software than I have to offer.

Some of you have offered to subscribe to my Tips, but I just don't have the time to get involved in anything like that. I have had some other projects on the back burner for too long, and it's time I got to work on them - they can hardly turn out to be less profitable than trying to sell software!

I am NOT going out of business, and I am NOT releasing my programs to the public domain. I will continue to sell them, and will continue some classified advertising.

My heartfelt thanks to the many user group editors and officers who have tried in many ways to encourage and help me. Many thanks to those who have purchased my programs.

I will greatly miss your

newsletters. I do hope to keep in contact with some of you. Perhaps now I can find time to browse in the TI sections of CompuServe or GENIE, and perhaps I will meet you there.

The answer to the challenge in the last Tips? For a clue, try -

DISPLAY AT(24,1):0 in Basic. Still don't get it? In Basic, DISPLAY is the same as PRINT, but AT is not recognized, so the computer thinks you are telling it to print the variable AT(1,1) - which, being undefined, is 0 - and advance to the next line (the :) and print 0.

I have always wanted a pocket calculator with several memories and a window to display the contents of each one. So, since there is plenty of room for windows on a TV screen, I wrote one.

It does not require any use of the Enter key, but each CALL KEY input must be validated and processed, so don't type too fast. It will accept such inputs as M1=7 or M1=7+1 or M2=1-M1 to put a value in a memory, or 6+7 or 6+M2 to calculate and display, or 6+7M1 or M1-M2M3 to calculate and put into memory, and will even do multiple calculations such as 1+2-3/4*5%6, subtotaling after the first two.

```
100 CALL CLEAR :: CALL SCRE
N(5):: DEF S$(X)=SEG$(A$,X,1
)&" = " :: CALL PEEK(8198,A)
:: IF A<>170 THEN CALL INIT
110 CALL LOAD(-31806,16):: D
N WARNING NEXT :: 60TO 140
120 SET,M$( ),K,S,A$,S$( ),R,C
,N,M1,N2,N1F,N2F,M1F,M,MF,DF
,FF,VF,EF,FL,N$,F2,T,M2,MEM(
),ST,NX,ZF
```

```
130 CALL COLOR :: CALL CHAR
:: CALL KEY :: CALL SOUND !E
P-
```

```
140 FOR SET=0 TO 4 :: CALL C
```

```

OLOR(SET,16,1):: NEXT SET ::
FOR SET=5 TO 8 :: CALL COLO
R(SET,5,16):: NEXT SET :: CA
LL CHAR(64,"0")
150 FOR SET=9 TO 12 :: CALL
COLOR(SET,16,1):: NEXT SET
160 DISPLAY AT(1,10):"TIGERC
UB": " MULTIMEMORY@CALCULAT
OR": "MEMORY #1": "MEMORY
#2": "MEMORY #3": "MEMORY
#4": "MEMORY #5"
170 M$(1)="0123456789.+-%/=
CXM" :: M$(2)="0123456789.AS
MDPECM" :: DISPLAY AT(20,1)
:"use ?":(1) symbols":(2)
alpha characters"
180 CALL KEY(0,K,S):: IF S=0
OR K<49 OR K>50 THEN 180 ::
A$=M$(K-48)
190 DISPLAY AT(20,1):S$(12);
"add";TAB(16);S$(16);"percen
t" :: DISPLAY AT(21,1):S$(13
);"subtract";TAB(16);S$(17);
"equals"
200 DISPLAY AT(22,1):S$(14);
"multiply";TAB(16);S$(18);"c
ancel" :: DISPLAY AT(23,1):S
$(15);"divide by";TAB(16);S$
(19);"clear all"
210 DISPLAY AT(24,1):"M1 to
M5 = memories #1 to #5"
220 R=15 :: C=1 :: N,N1,N2,M
1F,N2F,M1F,M,MF,DF,FF,VF,EF,
FL,ZF=0 :: N$="" :: DISPLAY
AT(18,1):""
230 CALL KEY(3,K,S):: IF S<1
THEN 230 :: CALL SOUND(50,5
00,5):: DISPLAY AT(R,C):CHR$
(K):: C=C+1
240 ON POS(A$,CHR$(K),1)+1 6
OTO 260,270,270,270,270,270,
270,270,270,270,280,290,
250,290,290,290,340,410,420,
430
250 IF VF=1 OR MF=1 THEN 290
:: ZF=1 :: N$="-" :: GOTO 2
30
260 DISPLAY AT(R,C-1):"?" ::
C=C-1 :: GOTO 230
270 IF MF=1 THEN 260 :: FL=0
:: VF=1 :: IF DF=0 AND ZF=0
THEN N=N$10+K-48 :: GOTO 23
0 ELSE N$=N$&CHR$(K):: GOTO
230
280 IF DF=1 THEN 260 :: DF=1
:: MF,FL=0 :: IF ZF=1 THEN
N$=N$& "." :: GOTO 230 ELSE N
$=STR$(N)& "." :: GOTO 230
290 IF C=2 OR FL=1 THEN 260
:: FL=1 :: IF FF=0 THEN 320

```

```

300 F2=POS(A$,CHR$(K),1)-11
:: IF VF=1 THEN 60SUB 480
310 60SUB 520 :: N1=T :: DIS
PLAY AT(18,1):"SUBTOTAL";T :
: N2F,N2=0 :: FF=F2 :: 60TO
230
320 IF VF=0 THEN 330 :: VF,M
F=0 :: 60SUB 480
330 MF=0 :: FF=POS(A$,CHR$(K
),1)-11 :: 60TO 230
340 IF C=2 OR(FF=0 AND M1F=0
)OR(C=4 AND M1F=0)OR FL=1 TH
EN 260
350 IF C=4 THEN EF=1 :: M2=M
:: N1F,MF=0 :: 60TO 230
360 IF VF=1 THEN 60SUB 480
370 IF EF=0 THEN 400
380 IF N2F=0 THEN MEM(M2)=N1
:: DISPLAY AT(M2*2+2,11):N1
:: 60TO 220
390 60SUB 520 :: MEM(M2)=T :
: DISPLAY AT(M2*2+2,11):T ::
60TO 220
400 60SUB 520 :: DISPLAY AT(
15,C):T :: 60TO 220
410 DISPLAY AT(R,1):":::""
:: 60TO 220
420 MEM(1),MEM(2),MEM(3),MEM
(4),MEM(5)=0 :: FOR R=4 TO 1
2 STEP 2 :: DISPLAY AT(R,10)
:"" :: NEXT R :: 60TO 410
430 IF EF=1 AND MF=1 THEN 26
0
440 CALL KEY(3,K,ST):: IF ST
<1 OR K<49 OR K>53 THEN 430
ELSE CALL SOUND(50,500,5)::
M=K-48 :: DISPLAY AT(R,C):CH
R$(K):: C=C+1 :: MF=1 :: FL
=0 :: IF VF=1 THEN 60SUB 480
450 IF N1F=0 THEN M1F,N1F=1
:: N1=MEM(M):: IF ZF=1 OR DF
=1 THEN N1=VAL(N$&STR$(N1)):
: DF,ZF=0 :: 60TO 230 ELSE 2
30
460 IF N2F=0 THEN N2F=1 :: N
2=MEM(M):: IF ZF=1 OR DF=1 T
HEN N2=VAL(N$&STR$(N2)): DF
,ZF=0 :: 60TO 230 ELSE 230
470 60SUB 520 :: MEM(M)=T ::
DISPLAY AT(M*2+2,11):T :: 6
0TO 220
480 IF DF=0 AND ZF=0 THEN NX
=N ELSE NX=VAL(N$):: DF,ZF=0
490 IF N1F=0 THEN N1=NX :: N
1F=1 :: 60TO 510
500 N2=NX :: N2F=1
510 VF,M=0 :: N$="" :: RETUR
N
520 IF FF=1 THEN T=N1+N2 ELS
E IF FF=2 THEN T=N1-N2 ELSE

```

```

IF FF=3 THEN T=N1*N2 ELSE IF
FF=4 THEN T=N1/N2 ELSE T=N1
*N2/100
530 RETURN

```

I have always been annoyed by the difficulty of hyphenating with TI-Writer, when I want to avoid the gaping holes that wraparound and Fill and Adjust can cause. Manually filling and adjusting with carets is slow, and leaving a space after the hyphen is unreliable, so I wrote this program.

```

100 DISPLAY AT(2,10)ERASE AL
L:"TIGERCUB": " HYPHENATED F
ILL AND ADJUST"
110 DISPLAY AT(6,1):" Prepar
e text with TI-Writer": "Edit
or. Leave left TAB at 0,": "s
et right TAB at the actual"
:"value of the line length d
e-"
120 DISPLAY AT(10,1):"sired
(i.e., for a 28-char": "lin
e, set it at 28)."
130 DISPLAY AT(12,1):" Inden
t as desired. Center": "hea
dings as desired but be": "
sure to follow them with a
": "line feed (Enter). Hyphen
ate"
140 DISPLAY AT(16,1):"as de
sired and follow the": "hyp
hen immediately with a": "
line feed (Enter)."
150 ON ERROR 160 :: 60TO 170
160 ON ERROR 160 :: RETURN 1
70
170 DISPLAY AT(20,1):"INPUT
FILE? DSK" :: ACCEPT AT(20,1
6)BEEP:F$ :: OPEN #1:"DSK"&F
$,INPUT
180 DISPLAY AT(22,1):"OUTPUT
FILE? DSK" :: ACCEPT AT(22,
17)BEEP:MF$ :: OPEN #2:"DSK"
&NF$,OUTPUT
190 DISPLAY AT(24,1):"LINE L
ENGTH?" :: ACCEPT AT(24,14)V
ALIDATE(DIGIT):L
200 LF$=CHR$(13):: H$="-"&CH
R$(13)
210 ON ERROR 210 :: 60TO 220
220 ON ERROR 210 :: RETURN 3
10
230 LINPUT #1:M$ :: IF M$="
" OR M$=LF$ OR M$="" OR ASC(

```

```

M$)>127 OR(LEN(M$)=L AND POS
(M$,LF$,1)=0)OR POS(M$," ",1
)=0 THEN 310
240 IF POS(M$,LF$,1)<>0 AND
POS(M$,H$,1)=0 THEN 310
250 IF POS(M$,H$,1)<>0 THEN
M$=SEG$(M$,1,LEN(M$)-1)
260 IF LEN(M$)=L THEN 310
270 P=1
280 X=POS(M$," ",P):: IF X=P
THEN P=P+1 :: 60TO 280 ELSE
Y,P=X :: IF POS(M$," ",P)=0
OR P=L THEN 310
290 M$=SEG$(M$,1,X)&" "&SEG$
(M$,X+1,255):: IF LEN(M$)=L
THEN 310 ELSE P=X+2
300 X=POS(M$," ",P):: IF X=0
THEN P=Y :: 60TO 300 ELSE 6
0TO 290
310 PRINT #2:M$ :: IF EOF(1)
<>1 THEN 230 ELSE CLOSE #1 :
: CLOSE #2

```

Here is one for the pre-schoolers -


```

100 CALL CLEAR :: CALL SCREE
N(14):: CALL COLOR(1,11,11,1
2,5,5):: DISPLAY AT(3,10):"S
EE-N-SAY": " : "PRESS ANY KEY
" 'by Jim Peterson based on
a routine by Michael Lyons
110 DIM E$(16),PAT$(16):: CA
LL CHAR(123,RPT$( "F",16))
120 DATA " ", " {", " {
", " {{", " { ", " { ", " {{
", " {{{", " { ", " { ", " {{{
", " {{{", " {{{ ", " {{{ ", " {{{
", " {{{"
130 FOR J=0 TO 15 :: READ PA
T$(J):: NEXT J
140 CALL KEY(0,K,S):: IF S=0
THEN 140
150 CALL CHARPAT(K,CP$):: FO
R X=1 TO 16 :: Y=ASC(SEG$(CP
$,X,1)): E$(X)=PAT$(Y+(Y>57
)&7-48):: NEXT X :: IF K>96
AND K<123 THEN K=K-32
160 CALL CLEAR :: CALL SAY(C
HR$(K)): FOR X=2 TO 16 STEP
2 :: DISPLAY AT(8+(X/2),12)
:E$(X-1);E$(X):: NEXT X
170 CALL SAY(CHR$(K)): 60TO
140

```

And so, one more time

MEMORY FULL

Jim Peterson 

FROM OUT 'N ABOUT
JUST FOR YOU

Rocky Mountain 99ers April, 1987



DM1000

Changelog

Modified by Ralph Romans:

Ver 3.0 Fixes to Ver 2.4

- Incorrect file count when going from 'M' to 'C'
- File copy would give you a bad copy if the file being copied was stored on the master disk as a non continuous file and the size of the first segment was exactly 39 sectors with additional sectors in another segment on the disk.

Ver 3.1 Fixes to Ver 3.0

- File copy would give you a bad copy if the master file was a fracter file of exactly 39 sectors and the same file name was on copy disk.
- When entering a file name in various modes it was possible to mess it up.

Unfixed bugs in Ver 3.1

- unable to display some dis/var 80 files that are full of control characters. Computer hangs up.

Ver 3.3

- Changed defaults on sweep and disk initialization.
- Disk initialization works for Myarc and Corcom.
- read/write errors gets cleared after 1st use on disk copy.
- file 'Mgr1' may now be called any name and all features of DM1000 will work.!! This will only work with TI controller and Corcom controller.
- The loader for Myarc controller is called loadmy.
- During disk initialization menu, you can use the up arrow to go back to previous question.

Ver 3.4

- Able to delete/move/copy 1 sector files.
- Added 'up arrow active' notice when up arrow will take you back to previous question.

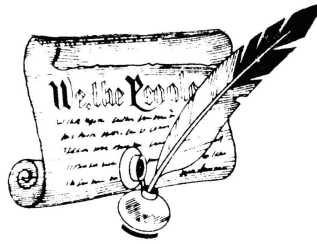
Ver 3.5

- Able to type/print display var 80/fixed 80 files while the file listing is on the screen by pressing a 'T' for type(display) file to screen or 'P' for print to list device with optional control codes sent to printer first.
- The 'P' and 'T' for print or type are only valid in the left most field.
- 'EOF' noticed added in the lower lefthand corner of screen.
- Display var 80/fixed 80 menu removed.

The latest version of Disk Manager 1000 is 3.5 and it is truly remarkable. The above is a listing of the version 3.x history showing the additions, improvements, and the enhancements.

This latest version has two significant enhancements, which involve the displaying and printing of d/v-80 files directly to the printer from the [cmd] prompt as the cursor is sitting there a "t" for type (to display), or a "P" for the printer may be typed instead of the normal copy/move/or delete.

FROM OUT 'N ABOUT



```

100 ! ORACLE
110 ! VERSION XB.2.1
120 ! 08 MAR 85
130 ! BY JIM SWEDLOW
140 !
150 DISPLAY AT(10,4)ERASE ALL BEEP:"** I AM THE ORACLE **" :: CALL DELAY :: RAN
DOMIZE
160 CALL INIT :: CALL PEEK(-28672,I):: IF I=0 THEN DISPLAY AT(20,1):"I cannot op
erate without the Speech Synthesizer!" :: STOP
170 DISPLAY AT(15,1):" I answer all questions": : "Ask questions with YES or
NOanswers -- When you are donepress ENTER."
180 CALL DELAY :: DISPLAY AT(24,1):" PRESS ANY KEY TO BEGIN"
190 CALL KEY(0,I,S):: IF S=0 THEN 190 ELSE CALL CLEAR
200 PRINT : : "WHAT IS YOUR QUESTION?" :: INPUT Q$ :: IF Q$="" THEN 220
210 CALL DELAY :: CALL REPLY(Q$):: CALL DELAY :: GOTO 200
220 DISPLAY AT(10,1)ERASE ALL:"THANK YOU FOR CONSULTING": : : : : " ** THE
ORACLE **" :: CALL DELAY :: STOP
230 !
240 SUB DELAY :: FOR I=1 TO 200 :: NEXT I :: SUBEND
250 !
260 SUB REPLY(A$)
270 A$=SEG$(A$,1,2):: IF A$="WH" OR A$="HO" THEN CALL OTHER ELSE CALL YESNO
280 SUBEND
290 !
300 SUB YESNO
310 ON INT(10*RND)+1 GOTO 320,330,320,340,350,350,360,370,380,390
320 CALL SAY("YES"):: SUBEXIT
330 CALL SAY("I THINK SO"):: SUBEXIT
340 CALL SAY("LOOKS POSITIVE"):: SUBEXIT
350 CALL OTHER :: SUBEXIT
360 CALL SAY("LOOKS NEGATIVE"):: SUBEXIT
370 CALL SAY("I DO NOT THINK SO"):: SUBEXIT
380 CALL SAY("NO WAY"):: SUBEXIT
390 CALL SAY("NO"):: SUBEND
400 !
410 SUB OTHER
420 ON INT(10*RND)+1 GOTO 430,440,450,460,470,480,490,500,510,520
430 CALL SAY("I CAN NOT TELL YOU THAT"):: SUBEXIT
440 CALL SAY("SAY THAT A DIFFERENT WAY"):: SUBEXIT
450 CALL SAY("YOU DO NOT WANT TO KNOW"):: SUBEXIT
460 CALL SAY("I DO NOT KNOW"):: SUBEXIT
470 CALL SAY("I AM NOT SUPPOSED TO SAY"):: SUBEXIT
480 CALL SAY("I WILL NOT TELL YOU"):: SUBEXIT
490 CALL SAY("I CAN ONLY GUESS"):: SUBEXIT
500 CALL SAY("I CAN NOT ANSWER THAT"):: SUBEXIT
510 CALL SAY("I DO NOT REMEMBER"):: SUBEXIT
520 CALL SAY("TRY SOME THING ELSE"):: SUBEND

```

TEXTWARE, SOFTWARE, and ELSEWHERE
 Happenings in the T.I. World Community
 by JACK SUGRUE



WORD HUNTING
 PRINTING
 AND XB SPEED

Puzzles are a word-processing tool we often forget about. Although I'm not much of a word hunt person (word search, find-a-word, whatever), my fifth-grade students are. I prefer crosswords or cryptograms, but wordhunts have their value. They are also a very pleasant way to introduce the names of bones in the human skeleton or the Presidents or geometrical terms. Computers are great teacher helpers in these matters. All one has to do is find a puzzle-creator program from somewhere and type in the words. The sorting and placement and completion of the puzzle is done by our 99. So is the making of hardcopies of puzzle, wordlist, answers. It's so easy. But not always so quick.

Way back in 1982 *WORD SAFARI* was programmed for TI. It's amazing how many people own it but never use it. Most people got it with *OLDIES BUT GOODIES* before they got printers or disk drives. By the time they got these things they'd long since forgotten this puzzle gem.

I have six word hunt programs, PD and commercial, but there was always a few things about WS that appealed to me. (The "thinking numbers" on the lower right screen, for one.) Though it was soundless and colorless and was SLOW BASIC and had very poor printouts, the program had a structure I liked. The way the menu operation worked appealed to me.

I decided to use that original structure to make a more modern version of this puzzle, suited for disk. First it had to be XB for speed and because I don't know assembly. Then it had to have a continuing music pattern while the reading was going on internally. It also had to have color. And some better screen directions. And much better hardcopies (enlarged, double strike, etc.) with a better overall look. It also had to be drastically reduced in size, though XB would take care of some of that automatically.

I printed out all six of my word hunt puzzles to see what features I wanted for this new one and how I could most efficiently program it. It was fun. Besides the menu operation, I found the sizes offered were excellent for the different kinds of things I planned to do in my classroom. Each of the programs presented me with interesting concepts and interesting problems. However, after three days (nights, actually) I had a debugged version which satisfied me. Then, as I do with all programs, I gave it to some adults and some kids to play with. The input from them helped me fine tune the project. I then (as I always do) put it away for a couple weeks.

Later, I took the "worker" disk out of storage, turned on my Gemini, and set out to do my first "official" school puzzle: names of dinosaurs (thus I had to use the large square).

Perfect.

I made 27 copies of the puzzle and word list for the kids and one copy of the answers for me. FAST!

Then I put this right on my *FUNLPLUS!* and was ready to use it whenever I word processed. Very handy. I since have added a few subprograms from Jim Peterson's wonderful *NUTS 'n BOLTS* disk which will not appear in the succinct piece below.

Anyway, I'm not permitted to print the six original hunt programs, but if you have a copy of *WORD SAFARI* and/or some others you might want to try them out if you haven't used them for a while; then print out a copies of the listing.

Next type in a copy of *WORD HUNT*, try it out, and make some comparisons. This will give you a good idea of how I went about the task.

Think about some really good programs you recall from the past, dig them out, look them over, see if you can brighten them up, make them better. Sometimes it'll take drastic changes. Sometimes little. But in either case you've had some fun with some programs you probably would have left forgotten in a box in the closet. Now you can enjoy them all over again.

NOTE: When I first wrote the program I was going to write it in BASIC. After I translated quite a few line from my flowchart notes, I decided to go into XB. I continued with the PRINT and colon lines instead of DISPLAY AT statements until testing. It turned out to be fine and fast as it was, so I left them in. Those of you who'd like to tighten this even more, adding or changing features, send me a copy. If the improvements are really good, I'll print an update in a future column. (Am also looking for original *SHORT* programs in XB for inclusion here with full credit.)


```

WORD HUNT
100 G=1 : CALL CLEAR : DIM
A$(728),A(7),B$(25)
110 C$="0" : FOR B=0 TO 8 :
: CALL COLOR(B,2,13)
120 NEXT B : CALL CHAR(96,"
0") : GOSUB 200 : PRINT "
WORD HUNT PUZZLE " : : : :
: : : : PRINT " ~~~~
trike~~~~any~~~~key~~~~" : GOS
UB 750
130 GOSUB 200 : PRINT " WOR
D HUNT PUZZLE " : : : PRINT
: : " CHOOSE" : PRINT " :
1 SETUP (CHOOSE FIRST) : :
PRINT " 2 INPUT PUZZLE"
140 PRINT " 3 OUTPUT LIS
T"
150 PRINT " 4 OUTPUT ANS
WER" : PRINT " 5 QUIT" :
: : :
160 GOSUB 750 : IF C<49 THE
N 160
170 IF C>53 THEN 160
180 ON C-48 GOSUB 770,230,97
0,870,190 : GOTO 130
190 CALL CLEAR : STOP
200 CALL CLEAR : CALL SCREE
N(1) : CALL VCHAR(0,0,96,48
FOR D=0 TO 8
210 CALL SOUND(-999,VAL(SEG$
("26226229433034939244049452
3587659698784",INT(12*8ND+0)
*3-2,3),0,VAL(SEG$("1311751
96",INT(3*8ND+0)*3-2,3),5)
220 NEXT D : RETURN
230 GOSUB 200 : PRINT " CHO
OSE " : PRINT " 1
10
X10 PUZZLE" : PRINT " 2
20X20 PUZZLE" : PRINT :
" 3 25X25 PUZZLE" : : :
: : : : :
240 GOSUB 750 : IF C<49 THE
N 240
250 IF C>51 THEN 240
260 E=(C-48)*10-5*INT(C/51) :
: GOSUB 200 : F=0 : IF G=0 :
H=0 : IF I=0 THEN 280
270 INPUT " TITLE OF PUZZLE?
" : D$ : INPUT " HOW MANY CO
PIES? " : F
280 INPUT " HOW MANY WORDS?
" : J : IF J<26 THEN 300
290 PRINT "SORRY: 25-WORD MA

```

```

XIMUM." : GOTO 280
300 FOR B=0 TO J-0
310 INPUT " WORD "&STR$(B+0)
&" ? " : B$(B) : IF LEN(B$(B))
>2 THEN 330
320 PRINT "SORRY: 3-LETTER M
INIMUM." : GOTO 310
330 IF LEN(B$(B))<11 THEN 35
0
340 PRINT "SORRY: 10-LETTER
MAXIMUM." : GOTO 310
350 NEXT B
360 INPUT "ANY CHANGES? (Y/N
)" : E$ : IF E$<>"Y" THEN 39
0
370 INPUT "WHICH NUMBER? " : B
380 INPUT "MAKE CORRECTION? "
: B$(B-0) : GOTO 360
390 GOSUB 200 : PRINT " : :
" THINKING..." : A(
0)=0 : A(4)=-0 : A(0)=E+0
: A(5)=-A(0) : A(2)=A(0)+0
: A(6)=-A(2) : A(3)=A(2)+0
: A(7)=-A(3) : B=E+3
400 K=(E+2)^2-0-B : FOR L=B
TO K : A$(L)=""
410 NEXT L : M=E+2 : K=M*M
-M : FOR B=0 TO K STEP M :
A$(B)="*" : A$(B+E+0)="*
M)=""
420 NEXT B : K=0 : FOR N=0
TO J-0 : D=0
430 D=D+0 : E$=STR$(D)&" "
: CALL HCHAR(23,25,ASC(E$))
: CALL HCHAR(23,26,ASC(SEG$
(E$,2,0)) : IF D<76 THEN 49
0
440 J=N : GOSUB 200 : PRIN
T "SORRY: CAN ONLY SQUEEZE I
N" : STR$(J) : WORDS." : : :
: : PRINT " CHOOSE" : PRIN
T " 1 USE " : STR$(J) : P
RESENT LIST." : PRINT " :
2 BEGIN AGAIN" : : : :
: : : : :
450 GOSUB 750 : IF C<>49 TH
EN 470
460 GOSUB 200 : PRINT "
THINKING..." : : : :
: : : : : GOTO 620
470 IF C=50 THEN 740
480 GOTO 450
490 B=INT(((E+2)^2-E)*8ND)+E
: IF A$(B)<>" " THEN 430
500 FOR D=0 TO 7
510 IF K<B THEN 530
520 K=0

```

```

"1"
860 RETURN
870 GOSUB 200 : IF I=0 THEN
890
880 INPUT "HOW MANY COPIES?
" : G : PRINT " : "PRINTING
ANSWERS."
890 FOR M=0 TO G : IF I=0 T
HEN 910
900 PRINT #1 : : " : ; D
$ : : : : :
910 FOR B=0 TO E+0 : PRINT
#1 : " : : : : FOR K=B*(E+2) T
O (B+0)*(E+2)-0 : PRINT #1 :
SEG$(A$(K),2,0) :
920 NEXT K
930 NEXT B
940 NEXT M : IF I<>0 THEN 9
60
950 PRINT " : : : : : PRESS ANY
KEY" : GOSUB 750
960 PRINT #1 : : : : : RETURN
N
970 GOSUB 200 : IF I=0 THEN
990
980 INPUT "HOW MANY COPIES?
" : H : PRINT " : "PRINTING
LIST."
990 FOR M=0 TO H : IF I=0 T
HEN 1010
1000 PRINT #1 : : " : ;
D$ : : : : : : :
1010 FOR B=0 TO J-0 : PRINT
#1 : " : ; B$(B)
1020 NEXT B : PRINT #1 : :
:
1030 NEXT M : IF I<>0 THEN
1050
1040 PRINT " : : : : : PRESS ANY
KEY" : GOSUB 750
1050 RETURN

```

JACK SMITH, Box 459, E. Douglas MA 01516

TI KEY/CHARACTER CODES

This program submitted by Jim Swedlow!

KEY	ASC HEX NAME	KEY	ASC HEX NAME	KEY	ASC HEX NAME	KEY	ASC HEX NAME
	0 >00 EOL marker	@	64 >40	CTRL ,	128 >80 null	FCTN J	192 >C0 >
FCTN 7	1 >01 AID	A	65 >41	CTRL A	129 >81 ELSE	FCTN K	193 >C1 +
FCTN 4	2 >02 CLEAR	B	66 >42	CTRL B	130 >82 ::	FCTN L	194 >C2 -
FCTN 1	3 >03 DElete	C	67 >43	CTRL C	131 >83 !	FCTN M	195 >C3 *
FCTN 2	4 >04 INsErT	D	68 >44	CTRL D	132 >84 IF	FCTN N	196 >C4 /
FCTN =	5 >05 QUIT	E	69 >45	CTRL E	133 >85 G0	FCTN Q	197 >C5 ^
FCTN 8	6 >06 REDD	F	70 >46	CTRL F	134 >86 GOTO	FCTN Y	198 >C6
FCTN 3	7 >07 ERASE	G	71 >47	CTRL G	135 >87 GOSUB	199 >C7 flag quoted	
FCTN 5	8 >08 Left Arrow	H	72 >48	CTRL H	136 >88 RETURN	200 >C8 flag unquoted	
FCTN D	9 >09 Right Arrow	I	73 >49	CTRL I	137 >89 DEF	201 >C9 flag line no	
FCTN X	10 >0A Down Arrow	J	74 >4A	CTRL J	138 >8A DIM	202 >CA EOF	
FCTN E	11 >0B Up Arrow	K	75 >4B	CTRL K	139 >8B END	203 >CB ABS	
FCTN 6	12 >0C PROC'D	L	76 >4C	CTRL L	140 >8C FOR	204 >CC ATN	
ENTER	13 >0D ENTER	M	77 >4D	CTRL M	141 >8D LET	205 >CD COS	
FCTN 5	14 >0E BEGIN	N	78 >4E	CTRL N	142 >8E BREAK	206 >CE EXP	
FCTN 9	15 >0F BACK	O	79 >4F	CTRL O	143 >8F UNBREAK	207 >CF INT	
	16 >10	P	80 >50	CTRL P	144 >90 TRACE	208 >D0 LOG	
	17 >11	Q	81 >51	CTRL Q	145 >91 UNTRACE	209 >D1 SGN	
	18 >12	R	82 >52	CTRL R	146 >92 INPUT	210 >D2 SIN	
	19 >13	S	83 >53	CTRL S	147 >93 DATA	211 >D3 SQR	
	20 >14	T	84 >54	CTRL T	148 >94 RESTORE	212 >D4 TAN	
	21 >15	U	85 >55	CTRL U	149 >95 RANDOMIZE	213 >D5 LEN	
	22 >16	V	86 >56	CTRL V	150 >96 NEXT	214 >D6 CHR\$	
	23 >17	W	87 >57	CTRL W	151 >97 READ	215 >D7 RND	
	24 >18	X	88 >58	CTRL X	152 >98 STOP	216 >D8 SEG\$	
	25 >19	Y	89 >59	CTRL Y	153 >99 DELETE	217 >D9 POS	
	26 >1A	Z	90 >5A	CTRL Z	154 >9A REM	218 >DA VAL	
	27 >1B ESC	[91 >5B	CTRL .	155 >9B ON	219 >DB STR\$	
	28 >1C	\	92 >5C	CTRL ;	156 >9C PRINT	220 >DC ASC	
	29 >1D]	93 >5D	CTRL =	157 >9D CALL	221 >DD PI	
	30 >1E Cursor	^	94 >5E	CTRL B	158 >9E OPTION	222 >DE REC	
	31 >1F Edge	~	95 >5F	CTRL 8	159 >9F OPEN	223 >DF MAX	
Space	32 >20 Space	¯	96 >60	CTRL 9	160 >A0 CLOSE	224 >E0 MIN	
!	33 >21	a	97 >61	161 >A1 SUB	225 >E1 RPT\$		
"	34 >22	b	98 >62	162 >A2 DISPLAY	226 >E2		
#	35 >23	c	99 >63	163 >A3 IMAGE	227 >E3		
\$	36 >24	d	100 >64	164 >A4 ACCEPT	228 >E4		
%	37 >25	e	101 >65	165 >A5 ERROR	229 >E5		
&	38 >26	f	102 >66	166 >A6 WARNING	230 >E6		
'	39 >27	g	103 >67	167 >A7 SUBEXIT	231 >E7		
(40 >28	h	104 >68	168 >A8 SUBEND	232 >E8 NUMERIC		
)	41 >29	i	105 >69	169 >A9 RUN	233 >E9 DIGIT		
*	42 >2A	j	106 >6A	170 >AA LINPUT	234 >EA UALPHA		
+	43 >2B	k	107 >6B	171 >AB	235 >EB SIZE		
,	44 >2C	l	108 >6C	172 >AC	236 >EC ALL		
-	45 >2D	m	109 >6D	173 >AD	237 >ED USING		
.	46 >2E	n	110 >6E	174 >AE	238 >EE BEEP		
/	47 >2F	o	111 >6F	175 >AF	239 >EF ERASE		
0	48 >30	p	112 >70	CTRL 0	176 >B0 THEN	240 >F0 AT	
1	49 >31	q	113 >71	CTRL 1	177 >B1 TO	241 >F1 BASE	
2	50 >32	r	114 >72	CTRL 2	178 >B2 STEP	242 >F2	
3	51 >33	s	115 >73	CTRL 3	179 >B3 ,	243 >F3 VARIABLE	
4	52 >34	t	116 >74	CTRL 4	180 >B4 ;	244 >F4 RELATIVE	
5	53 >35	u	117 >75	CTRL 5	181 >B5 :	245 >F5 INTERNAL	
6	54 >36	v	118 >76	CTRL 6	182 >B6)	246 >F6 SEQUENTIAL	
7	55 >37	w	119 >77	CTRL 7	183 >B7 (247 >F7 OUTPUT	
8	56 >38	x	120 >78	FCTN ,	184 >B8 &	248 >F8 UPDATE	
9	57 >39	y	121 >79	FCTN .	185 >B9	249 >F9 APPEND	
:	58 >3A	z	122 >7A	FCTN /	186 >BA OR	250 >FA FIXED	
;	59 >3B	{	123 >7B	CTRL /	187 >BB AND	251 >FB PERMANENT	
<	60 >3C		124 >7C	188 >BC XOR	252 >FC TAB		
=	61 >3D	~	125 >7D	FCTN ;	189 >BD NOT	253 >FD #	
>	62 >3E	^	126 >7E	FCTN B	190 >BE =	254 >FE VALIDATE	
?	63 >3F	FCTN V	127 >7F del	FCTN H	191 >BF <	255 >FF EOF marker	



FROM THE PRESIDENT'S
COMPUTER DESK
BY IRWIN HOTT

This month I have several unrelated items to write about.

If you were at the last CONNI meeting, you probably heard Jim Peterson offer to bring his public domain library to the July CONNI meeting.

There are some 285 disks and more than 2000 programs in the public domain library. For a copying fee of \$.50 per program, you may select from the library to enhance your own collection. We will be able to copy cassetts or disks. We will have blank disks at the meeting for \$.50 each. Bring your own blank cassettes. We will have DSDD disk copying capability.

All copying and disk fees are going into the CONNI treasury.

There will be a few printed copies of the public domain catalog that Jim has prepared. It is so extensive that you may wish to get an advance look at it. If you have a MODEM,

printer and disk drive you may download the catalog in 5 parts from the Spirit of #99 TIBBS. The number is 614-263-3412.

Jim has noted what hardware and/or software is required and he has rated each program on a scale from 1-10. Bring your change to the next CONNI meeting and have fun adding to your own library.

Now just a couple of personal notes.

I just added a Tripple-tech card to my system. The Tripple-Tech has a 64k printer buffer, clock and provisions for putting the speech synthesizer on the card itself. I have noticed that the number of lockups has been greatly reduced since I moved the speech synthesizer.

If you are having a lock-up problem, and you have your speech synthesizer connected, you have several options: A Remove the speech synthesizer whenever you don't need it.

B Put the speech in the console.

C Buy a Tripple-tech card.

D Buy a PEB board from Rave

#99 to move the speech synthesizer into the PEB box. The board costs \$50. Incidentally I am adding an external speech synthesizer. It works from the RS232 output. It is actually intended for an IBM. I should be able to dramatically increase my speech capability. I am sure that there will be a considerable loss in intelibility however. I am hoping that I will be able to use the printer spooler on FAST-TERM for example to RS232 port 3 and speech. I will keep you posted on developments.

Just a reminder that PattyCakes is open before during and after the CONNI meeting. Whether you need to wake up with coffee, or fill your hollow leg, there will be something for you at PattyCakes. All proceeds go to CONNI. So try to hold out for the extra few minutes it takes to get to the meeting. Thanks to Patty and John Cummings for their work on PattyCakes.

See you next month.





HIDDEN COMMANDS IN PERSONAL RECORDKEEPING

by Newt Armstrong - via LA 99'ers

The TI99/4A is an enigma; many of its capabilities are only alluded to or are hidden. Take the Personal Record Keeping (PRK) module, for example. Did you know that you can call seven PRK subprograms from TI Basic if you have the module installed? Five of these allow you to create and access PRK formatted files, and the other two have the versatility of the ACCEPT AT and DISPLAY AT Extended Basic statements.

Now you say, 'what earthly good does it do to put data from a Basic Program into PRK format? Well, it allows you to massage your data with PRK, Statistics, and the Personal Report Generator modules. Best of all, to save your data on tape in 'program' format. How do you do it? Follow along:

PRK subprograms are names PREP, HEADER, GETPUT, LOAD, SAVE, ACCEPT, and DISPLAY. PREP is used to partition the Video Display Processor (VDP) RAM to provide a dedicated area for working on the PRK formatted file. HEADER is used to define the file structure and to retrieve housekeeping data for working on the file. GETPUT is used to transfer data between the file and the basic program. LOAD and SAVE are used to retrieve and store files in external storage devices. ACCEPT and DISPLAY accept data from the key board and display it on the screen.

PREP: PREP is the subprogram invoked to partition VDP RAM for the work area. Format for the statement is:

CALL P(byte) - where byte is the number of bytes being reserved. The sequence is:

- Main Title Screen
- Press Any Key
- Master Selection List
- Press 1 for TI Basic
- Invoke CALL P(bytes) (enter)
- (Disregard the next command is disk drive is not connected)
- Invoke CALL FILES(number) (enter)
- Invoke NEW

The partition will remain in place until cancelled with BYE or the QUIT command. Size of the work area effects the amount of VDP RAM available for the Basic program, as does buffer space. The CALL FILES(n) command reserves disk buffers; three are reserved automatically if the disk controller is connected to the console with power on. Each disk buffer uses about 520 bytes fo RAM. An interesting exercise is to check memory available both before and after partitioning, and with one or more FILES called. You can use the following routine:

```

1 A=A+B
2 GOSUB 1
>RUN
When response IS
MEMORY FULL IN 1
Invoke PRINT A

```

Notice that preparation is in the Commande Mode and

starts from sonsole power on, essentially. If it is attempted with any Basic commands in VDP RAM an ILLEGAL CALL error will occur.

After the NEW command is invoked, the computer is ready o run a Basic program. A PRK format file can be loaded into the work area from a storage device - or one can be originated or manipulated with data from a Basic program.

HEADER: Header is the subprogram invoked to define the file structure and to transfer housekeeping data between the file and the Basic program. The header is page 0 of the PRK format file. the format statement is:

CALL H(n1,n2,n3,V(\$)) where n1 is the read/write code (1/0 respectively); n2 is the data code (11-114, see following list); n3 is the item number; and V(\$) is the data variable.

CODE	DATA	TYPE
1	file name	0-9 characters
2	day	integer (1-31)
3	Month	integer (1-12)
4	year	integer (0-99)
5	Number of items per page (updated by routine)	
6	Number of pages (maintained by routine)	
7	Header length in bytes (maintained by routine)	
8	Page length in bytes (maintained by routine)	
9	Item name	0-9 characters
10	Item type	1=Characters 2=Integers 3=Decimal 4=Scientific notation
11	Item width: (Data window)	1-15 for Characters 1-10 for Integers 2-11 for Decimals 8-13 for Scientific Notation (maintained by routine)
12	Item decimal places	0 for characters 1 to width-1 for decimal 0 to 5 for Scientific Notation
13	Item storage (bytes) (maintained by routine)	
14	Item position in page (maintained by routine)	

Note that n3, the item number, is ignored for codes 11 thru 8 but must be included in the CALL STATEMENT as a space maintainer. Codes 9 thru 114 are repeated for each defined item.

As you can see there quite a bit of information to be included in the header. I think that it is easier to define file structure within the PRK program (I call it a Key

File), and then enter and manipulate data from a Basic program. Also, with data form codes 6, 7, and 8, you can determine the size of your file, and you will know how large a work area to allocate. PRK files are saved in 256-byte "chunks". So, the actual file length will be rounded to the next 256 multiple. TI, in the PRK manual, suggests a 2% overhead.

ED NOTE: If you exceed this safety margin and get the FULL warning it is too late!!! ALL IS LOST !!! beware of this pitfall.

GETPUT: Getput is the subprogram invoked to transfer data between the file and the Basic program. Formats for the statement are:

CALL G(n1,n2,n3,V(\$))

CALL G(n1,n2,n3,n4,V(\$)) where n1 is the read/write code (0=write 1=read, and 2=no data); n2 is the page number; n3 is the item number; n4 is the return code (used in the read statement only, 0=data found, 1=data missing); and V(\$) is the data variable. Some what ifs, must do's and no-no's about the statement contents follows.

PAGES: results are unpredictable for attempts to read from undefined, zero, or negative numbered pages. Pages should be created sequentially so numbers are not skipped. A page number in a write statement higher than any previously used will be the new highest page number stored in the Header. An error will result from attempts to read a page numbered higher than the highest stored.

ITEMS: Items are defined with the Header write statements and are the same in all pages. Results are unpredictable for attempts to read from zero or negative numbered items. An error will result from attempts to read an item numbered higher than the highest defined.

VARIABLE: The Variable must match data type (v for numeric, v\$ for string) and item definition. when v is an expression, the evaluation will be written, and the evaluation must fit the item definition. e.g. An expression that results in a number with three decimal places will not fit an item defined for two decimal places. Nor will an integer with four numbers of more (1000 up) fit an item defined to have a width of five with two decimal places.

LOAD: Load is the subprogram invoked to load a data file into the work area reserved by the PREP call. Format for the statement is:

CALL L(F\$,n) where F\$ is the file name("CS1", DSK1.____", etc) and n is a return variable. A return of 0 indicates an error occurred. Any other number indicates that the load was successful. Failures will be caused by a Call to a non-existent device or file, by general I/O errors, or by too small or no work space allocated.

SAVE: Save is the subprogram invoked to save a data file from the work area reserved by the PREP call. Format for the statement is the same as CALL L(F\$,n).

ACCEPT: Accept is the subprogram invoked to receive data from the keyboard and to echo that data at a certain screen location. Formats for the statement are:

CALL A(n1,n2,n3,n4,v(\$))

CALL A(n1,n2,n3,n4,n5)

CALL A(n1,n2,n3,n4,n5,n6) where n1 and n2 are row/column respectively. n3 is its item width (data window); n4 is a return code (more about that later); v(\$) is the item number when it is the last numeric in the expression; or n5 is the low value of a low/high range with n6 the high value.

As mentioned before, this statement is similar to ACCEPT AT in extended Basic. Data typed on the keyboard is accepted into the variable v (for numeric) or v\$ (for string) and is echoed on the screen starting at location n1(row), n2(column). Length of the input is governed by the value of n3 (data window) or the end of the row, whichever comes first. The n4 return code allows processing of null entries and also for use of the function keys. Values returned for the various circumstances are listed below:

CODE	MEANING
1	Valid data entered
2	Empty (null) string
3	AID (F7) pressed
4	REDO (F8) pressed
5	PROC'D (F6) pressed
6	BEGIN (F5) pressed
7	BACK (F9) pressed

When n5 is used along, as the item number, the input will be checked for validity against characteristics stored in the Header for that item. Invalid data (wrong type, too many decimal places, etc) will be greeted with the BEEP that we all recognize and will be rejected. Using n5 in conjunction with n6 sets a range of valid data. Inputs outside of that range will be rejected.

DISPLAY: Display is the subprogram invoked to write at a certain screen location. Formats for the statement are:

CALL D(n1,n2,n3,v(\$))

CALL D(n1,n2,n3,n4,n5,etc.) where n1 and n2 are row/column locations, respectively; n3 is item width (data window); and v(\$) is the data variable. Multiple displays can be made with one call listing several screen locations, data windows, and data variables in sequence. Length of this call is limited by the length of a Basic statement. Positive valued data windows causes screen area clearing before data is displayed; Negative valued windows leave area uncleared. As with the accept call, data that extends beyond the end of the row will be chopped.

EXAMPLES: Two sample programs are listed below. Prior to using them, you will have to prepare a Header page (Key file). Just go into the PRK module and define a file structure for six items- Last name, First name, Address, City, State, Zipcode. Save this information under some file name. Next invoke CALL P(2000) as outlined above, and after

the new statent, invoke CALL L to load your key file. Then run either the Read or Write program below, and happy computing. I am preparing a demonstration to place in the library.

READ PROGRAM

```
10 REM READPRK/
20 CALL CLEAR
30 CALL SCREEN(13)
40 CALL D(7,10,2,"ln",9,10,2,"Ln",11,10,2,"ad",13,10,2,"ct",
15,10,2,"st",17,10,2,"zp")
50 CALL H(1,6,0,RE)
60 CALL H(1,5,0,FL)
70 FOR R=1 TO RE
80 FOR F=1 TO FL
90 CALL G(1,R,F,MD,D$)
100 CALL D(5+2*F,MD,D$)
110 NEXT F
120 CA;; KEY(0,K,S)
130 IF S<>1 THEN 120
140 NEXT R
150 STOP
```

WRITE PROGRAM

```
10 REM SMPLPRKW/
20 CALL CLEAR
30 CALL H(1,6,0,R)
40 CALL H(1,5,0,FL)
50 R=R+1
60 CALL CLEAR
70 CALL D(7,10,2,"ln",9,10,2,"Ln",11,10,2,"ad",13,10,2,
"ct",15,10,2,"st",17,10,2,"zp")
80 FOR F=1 TO FL-1
90 CALL H(1,11,F,FW)
100 CALL A(5+2*F,10,FW,FR,D$)
110 CALL G(0,R,F,D4)
120 NEXT F
130 CALL KEY(0,K,S)
140 IF S<>1 THEN 130
150 IF K=13 THEN 50
160 STOP
```

* (Refer to page 76, The Best of 99'er, Copyright 1983, Emerald Valley Publishing Co.)

** (I am indebted to my brother Al of the Southwestern 99'ers in Tuscon and to Jim swedlow of the ROM staff for information about these subprograms, and to David Hough, also of the ROM staff, for some sample programs he 'just' happened to have in his library.)



A REVIEW

by Dick Altman

For those of you using Harry Brashear's USEABLE DISK CATALOGER, and for those of you wanting to sensably catalog your disk libraries, have I got news for you!

Now there is Version 2.3 of the very best program available. It is even better than the earlier versions, and easier to use. Talk about 'user friendly', this one is just that.

It takes awhile (depending on how many disks you have) to run all your disks through it, but once you do, you will be eternally grateful to the Western New York 99ers and Harry. This is a 'fairware' program, but all proceeds go right into the Western NY 99ers treasury, so SUPPORT THE FAIRWARE CONCEPT!

This program does NOT catalog all the files on a disk unless you want it to! After all,

if you have a large program on one disk, there is no need to catalog each of the several files that make up that program, just list something like the 'load', or 'docs', plus the name of the program. There is plenty of room for necessary comments, your opinion of the program, whatever. There are special boxes to indicate whether it requires Extended Basic, Editor/Assembler, etc., merely with one keystroke. There is even a special box to indicate it is a 'game' or 'utility', or any of 5 other classifications.

I have a copy of this beautiful program direct from the writer--I call such a disk a 'virgin' because it is not a copy of a copy of a copy that might have been changed inadvertently or on purpose. Harry Brashear's address, if you wish to obtain your own virgin is: 2753 Main t., Newfane, N.Y. 14108. <<<O>>>

C.O.N.N.I. BUSINESS MEETING
MARTIN JANIS SENIOR CITIZENS CENTER
JUNE 13, 1987

MEETING OPENED 10:04 A.M.
MEETING ADJOURNED 10:42 A.M.

Meeting was called to order by President Irwin Hott, assisted by Vice President Jim Seitz, Irwin introduced three visitors.

Treasurer's report was accepted as read.

President Irwin Hott proceeded to old business on the subject of the August swap meet. A motion by Jim Peterson to use a union hall was followed by discussion and tabled until next month's meeting. A motion by Dick Beery to charge for a table fee carried unanimously.

BBS: Chuck Grimes informed the club of The All New Chuck's BBS, with already over 800 calls in less than two weeks.

Disk of the Month: Librarian Chuck Grimes gave the details on the contents of June disk of the month.

A motion to authorize payment to the Martin Janis Center carried unanimously.

Dick Beery gave the club updates on new products, and reported on the TI system at the Tremont library, encouraging members to help support it.

Jim Peterson offered to donate 285 disks of public domain programs, for the July meeting, with a 50 cents per program copying fee to go to the club.

Dick Heim explained the newsletter library which consist of letters for other user groups, made available to members at all the meetings.

Demonstrations: Ken Marshall demonstrated the use of TI-ARTIST, and Fred Deaner demonstrated his own keyboard modificaton.



LET'S KEEP SHOOTING
FOR A GREAT
NEWSLETTER!

CONTRIBUTE!

Letter to the Editor

Ed Note:

We now know why so many questions came up after printing Jim's XB Articles in April & May. It was brought to our attention that there were no programs to relate to with his articles XB05 & XB06. It was a wrong assumption thinking that they were referring to his SidePrint Program which we have in our library. Our call to Jim helped to set us straight, and now Jim has provided us with those missing programs. In response to your questions, will you please get out your April & May Newsletters and try again! XB05 was in regard to the Disk Menu Program. XB06 dealt with the Disk Cataloger, both are now in this issue! We regret any inconvenience this has caused you. Please accept our appologies!

Jim Swedlow

7301 Kirby Way, Stanton CA 90680

Carol Parkins, Editor
'Spirit of 99'
2215 Bayfield Drive
Columbus, Ohio 43229

May 20, 1987

Dear Carol,

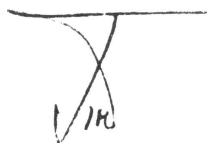
As John requested, I am sending you the programs that were missing from the XB columns. I went thru all of them and found that numbers 5, 6, 7, 8 and 10 were missing their programs. The others (starting with XB11) have the program listing as part of the column text file.

I thought that I had added the program listings to all of the text files. I have been writing these things for three years now and what became standard practice was not always.

On the attached disk are revised copies of XB's 6, 7 and 8 with the programs listings attached. The programs discussed in XB05 and XB10 no longer exist on media. I have attached photocopies of the listings as they originally appeared.

The three programs are also on the disk in a RUNable format in case you want to use them some other way. Also on the disk is ASCHART - the page it prints should be printed when you run XB08. The format you have must be run on an Epson or compatible printer.

Thanks again for running my XB columns. I hope that the CONNI members enjoy them - I certainly had a lot of fun writing them.



This is the Disk Cataloger
 program that was missing
 from Jim's XB06 article for May!
 It will read a disk catalog,
 and make a Loader Program
 that will auto-load for you!

This is the Disk Menu Program
 that was missing from Jim's
 XB05 article for April!

```

100 ! LOADMAKER
110 ! VERSION XB.1.1
120 ! 29 DEC 84
130 ! BY JIM SWEDLOW
140 !
150 DISPLAY AT(10,10)ERASE ALL:"LOADMAKER": : : : : "Initializing . . ." :: @=1
:: DIM A$(26),A(26):: B%=CHR$(182)&CHR$(181)&CHR$(199):: B=100 :: C%=CHR$(179)
160 OPEN #@:"DSK1.",INPUT ,INTERNAL,RELATIVE :: INPUT #@:A$(C),D,E,F :: DISPLAY
AT(16,@):"Disk ";A$(C);" * Free";F :: A(C)=LEN(A$(C))
170 INPUT #@:D$,D,E,F :: IF D$="" THEN 210 ELSE IF ABS(D)<>5 OR D$="LOAD" THEN 1
70
180 C=C+@ :: IF C<27 THEN A$(C)=D$ :: A(C)=LEN(D$):: DISPLAY AT(17,@):"Reading:
";D$ :: GOTO 170
190 DISPLAY AT(16,@)BEEP:"Your disk has more than 26 programs. Do you want to
proceede with the first 26 programs? Press Y or N."
200 CALL KEY(3,F,D):: IF F=78 THEN CLOSE #@ :: STOP ELSE IF F<>89 THEN 200
210 IF C=0 THEN DISPLAY AT(16,@)BEEP:"No programs were found on this disk." ::
CLOSE #@ :: STOP
220 CLOSE #@ :: DISPLAY AT(16,@):"Disk reading completed." : : : : : OPEN #@:"
DSK1.XXX",VARIABLE 163,DISPLAY ,OUTPUT
230 DISPLAY AT(16,@):" Making LOAD line 100" :: D%=CHR$(0)&CHR$(100)&CHR$(131)
&CHR$(32)&CHR$(80)&CHR$(82)&CHR$(79)&CHR$(71)&CHR$(82)&CHR$(65)&CHR$(77)
240 D%=D%&CHR$(32)&CHR$(76)&CHR$(79)&CHR$(65)&CHR$(68)&CHR$(69)&CHR$(82)
250 G,F=@ :: GOSUB 460 :: D%=D%&CHR$(182)&CHR$(239)&CHR$(236)
260 D%=D%&CHR$(181)&CHR$(199)&CHR$(7+A(0))&CHR$(68)&CHR$(105)&CHR$(115)&CHR$(107)
)&CHR$(32)&CHR$(42)&CHR$(32)&A$(0)
270 G=@-7*(C<19):: E=2 :: FOR D=@ TO C :: E=E-(D>C/2)*(6=@)*INT(C/2):: G=6+(D>C/
2)*(6=@)*14 :: IF D=C AND G=15 AND C/2<>INT(C/2)THEN G=8
280 F=E+D :: GOSUB 460 :: D%=D%&B%&CHR$(A(D)+3)&CHR$(64+D)&CHR$(32)&CHR$(32)&A$(
D):: NEXT D
290 F=24 :: G=@ :: GOSUB 460 :: D%=D%&CHR$(182)&CHR$(238)&CHR$(181)&CHR$(199)&CHR
R$(17)
300 D%=D%&CHR$(80)&CHR$(114)&CHR$(101)&CHR$(115)&CHR$(115)&CHR$(32)&CHR$(121)&CHR
R$(111)&CHR$(117)
310 D%=D%&CHR$(114)&CHR$(32)&CHR$(99)&CHR$(104)&CHR$(111)&CHR$(105)&CHR$(99)&CHR
$(101)
320 GOSUB 440 :: D%=D%&CHR$(157)&CHR$(200)&CHR$(3)&CHR$(75)&CHR$(69)&CHR$(89)&CHR
R$(183)
330 D%=D%&CHR$(200)&CHR$(@)&CHR$(51)&C%&CHR$(75)&C%&CHR$(83)&CHR$(182)&CHR$(130)
340 D%=D%&CHR$(132)&CHR$(75)&CHR$(191)&CHR$(200)&CHR$(2)&CHR$(54)&CHR$(53)&CHR$(
186)&CHR$(75)&CHR$(192)
350 F=64+C :: GOSUB 470 :: D%=D%&CHR$(176)&CHR$(201):: GOSUB 450
360 F=24 :: GOSUB 460 :: D%=D%&B%&CHR$(7)&CHR$(76)&CHR$(111)&CHR$(97)&CHR$(100)&
CHR$(105)&CHR$(110)&CHR$(103)
370 D%=D%&CHR$(130)&CHR$(155)&CHR$(75)&CHR$(194)&CHR$(200)&CHR$(2)&CHR$(54)&CHR$(
52)&CHR$(134)
380 D=B :: FOR B=B+10 TO B+10*C STEP 10 :: IF B>D+10 THEN D%=D%&C%
390 D%=D%&CHR$(201):: GOSUB 450 :: NEXT B :: B=D
400 FOR D=@ TO C :: F=24 :: G=9 :: GOSUB 460 :: D%=D%&B%&CHR$(A(D))&A$(D)&CHR$(1
30)&CHR$(169)&CHR$(199)
410 D%=D%&CHR$(A(D)+5)&CHR$(68)&CHR$(83)&CHR$(75)&CHR$(49)&CHR$(46)&A$(D):: NEXT
D
420 PRINT #@:D%&CHR$(0):CHR$(255)&CHR$(255):: CLOSE #@
430 DISPLAY AT(16,@)BEEP:"LOAD program made!": "Enter the following commands":
:" >NEW": " >MERGE DSK1.XXX": " >SAVE DSK1.LOAD": " >RUN" :: STOP
440 B=B+10 :: PRINT #@:D%&CHR$(0):: D$="" :: DISPLAY AT(16,20):B
450 D%=D%&CHR$(INT(B/256))&CHR$(B-256*INT(B/256)):: RETURN
460 GOSUB 440 :: D%=D%&CHR$(162)&CHR$(240)&CHR$(183):: GOSUB 470 :: D%=D%&C% ::
F=6
470 IF F<10 THEN D%=D%&CHR$(200)&CHR$(@)&CHR$(48+F):: RETURN ELSE D%=D%&CHR$(200)
&CHR$(2)&CHR$(48+INT(F/10))&CHR$(48+F-10*INT(F/10)):: RETURN

```

```

100 * DISK MENU PROGRAM
110 * VERSION XB.1.2
120 * 29 DEC 84
130 * FROM THE POMONA 99 UG
140 * MODIFIED BY J SWEDLOW
150 *
160 DIM A$(18):: OPEN #1:"DSK1.",RE
LATIVE,INPUT ,INTERNAL :: INPUT #1:
D$,A,B,C :: DISPLAY AT(1,1)ERASE ALL
:"DISK ";D$;" * FREE";C: "Press Fo
r"
170 INPUT #1:D$,A,B,C :: IF D$="" T
HEN 190 ELSE IF ABS(A)<>5 OR D$="LO
AD" THEN 170
180 S=S+1 :: A$(S)=D$ :: IF S<18 TH
EN DISPLAY AT(S+4,3):CHR$(9+64);"
";D$ :: GOTO 170 ELSE DISPLAY AT(2
2,3):"R To continue"
190 DISPLAY AT(24,1)BEEP:"Press <ER
ASE> to stop"
200 CALL KEY(3,A,B):: IF A=7 THEN C
LOSE #1 :: STOP ELSE IF A<65 OR A>6
4+S THEN 200 ELSE A=A-64
210 IF A=18 AND D$<>"" THEN CALL HC
HAR(3,1,32,32*22):: S=0 :: GOTO 180
ELSE D$="DSK1."A$(A):: DISPLAY AT
(24,1)BEEP:"Loading ";A$(A):: CLOSE
#1
220 CALL INIT :: CALL PEEK(-31952,A
,B):: CALL PEEK(A*256+B-65534,A,B):
: C=A*256+B-65534 :: CALL LOAD(C,LE
N(D$))
230 FOR I=1 TO LEN(D$):: CALL LOAD(
C+I,ASC(SEG$(D$,I,1)):: NEXT I ::
CALL LOAD(C+I,0)
240 RUN "DSKX.1234567890"

```

NEWSLETTER

DEADLINE IS

THE 3rd SATURDAY

EACH MONTH



C.O.N.N.I. IS ONE YEAR OLD

One day in late 1982 Bill Zipf asked me if I was interested in forming a USERS GROUP. At the same time Art Morgan, Paul Powers and Pat Saturn were thinking along the same lines. They had been to some Cin-Day meetings and felt that it was too far to drive. So, Dale Smith's phone number was posted at Sun TV for a free User Group (Ha, Ha). Pat called him, and Paul who called Bill, anyway there were a lot of phone calls. At that time I had had my computer about two months. I do not exactly remember where that first preliminary meeting was held, but we decided to form a club using the constitution Texas Instruments recommended. We subsequently modified it slightly and decided to incorporate the club as a non profit corporation. The first official meeting (following a couple of early morning breakfast meetings at the Village Inn) was in January 1983 at the ELECTRICAL WORKERS BUILDING on 2nd Avenue, courtesy of Joe Hoover. By this time Pat Saturn had worked hard to put together a NEWSLETTER and BIGGIE started to do his thing. The club's name and that of the newsletter were also decided upon in these early days. There was some debate about when we should hold our meetings but a major factor in choosing Saturday morning was the fact that Pat Saturn worked every evening. The one person who has really suffered from this decision is Rod Leverage, our Librarian because he has to work most Saturdays. However this has not prevented Rod from doing an excellent job in helping to run the club and organize the Software Library. Looking at

the early newsletters indicates that most of the people who helped make this club become a roaring success were members in February 1983.

Texas Instruments did not officially recognize the club till February when we had adopted a constitution and elected our officers: President, Vice President, Secretary, Treasurer and Librarian: Roger Wills, Paul Powers, Bill Zipf, Art Morgan and Rod Leverage, respectively. In the first few months the newsletter became better and better due mainly to the efforts of Jim Peterson, Pat Saturn and myself. TI told us that we had one the best club newsletters in the country. They also published one of Jim Peterson's articles in TI's own newsletter.

In the spring we had a few problems. First of all, the newsletter nearly went out of business as a result of growing pains. The second problem was related to the club's rapid growth. We had to find a larger hall for our meetings. At this point we did not know from one month to the next where we would be, and it was very important that the newsletters went out on time so everybody knew where the next meeting would be. There was much discussion about free parking, location etc. Joe Frederick, a relatively new member at that time, suggested that we could probably use THE MARTIN JANIS CENTER. We are really lucky to be able to use this facility. However we must all recognise that we are being allowed to use this building only as a special favor and that we should leave it as we find it—clean and tidy. In the

CONTINUED

summer we spent several months discussing copyrights and a Software Library. This was finally resolved and Rod Leversee, Jim Peterson, Paul Powers and Sam Morabito put in a lot of hard work to get the library off the ground. We have approximately 200 programs. The library is a very important part of the club and I look forward to the next stage of its growth—the interchange of software with other USERS GROUPS. Sam Morabito organized some good visits and speakers during the year—OCLC, ON LINE BANKING WITH THE HUNTINGTON BANK—to name just a couple. I also managed to arrange a visit from Edward Weist, USERS GROUP COORDINATOR FOR TI. Over 100 people attended this special meeting in which Edward and John Phillips demonstrated TI Software, Quadrophonic Sound and answered questions. Ed is now working in the Semiconductor division in Dallas. Thanks for your help, Edward.

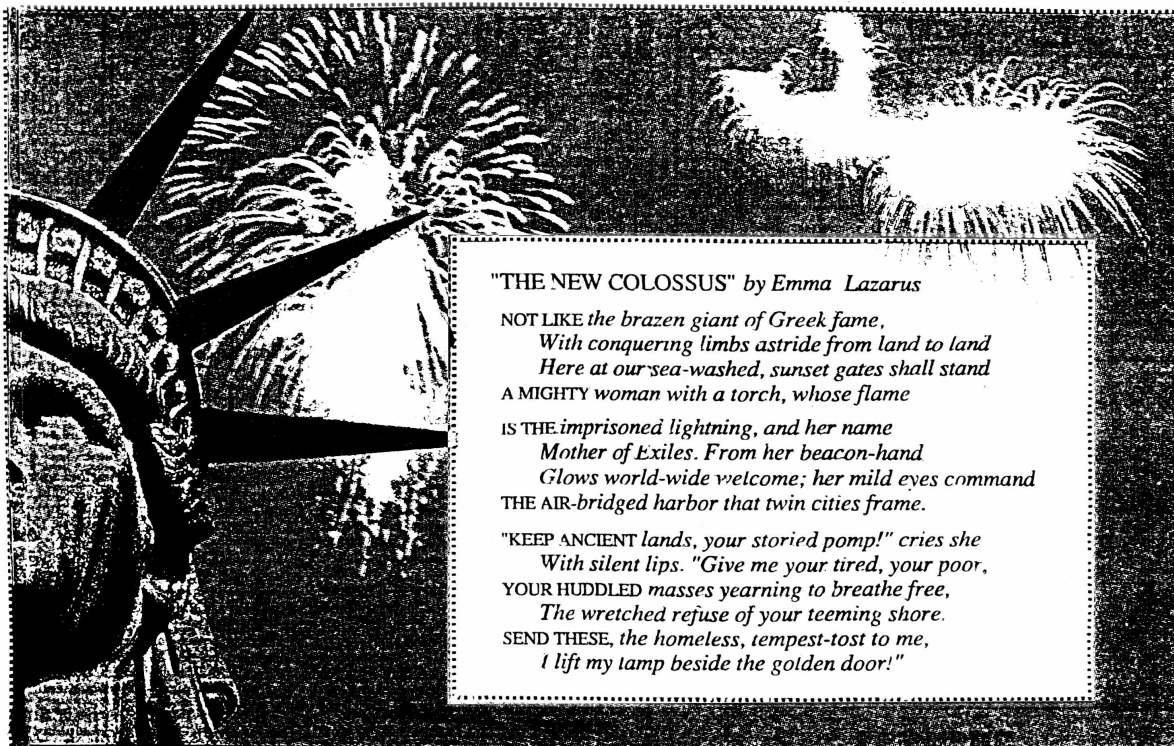
I've tried to give all of

you, but particularly the newer members a bit of the clubs history and tell you about some of the people who helped get the club to its current position. I hope I haven't missed anybody. It hasn't been all smooth riding but the officers and a number of other people have created what many people have said to me is the best club they have ever been in. That is rather gratifying. The main challenge for the incoming officers is to build an organization to cater for the diverse skills and interests of the large and rapidly growing number of members. This will give opportunities to other people to step forward to guide and manage the next stage of the clubs life. I've certainly met

a lot of new people and learned a great deal about my computer as a result of participating in this club.

President, Roger Wills 

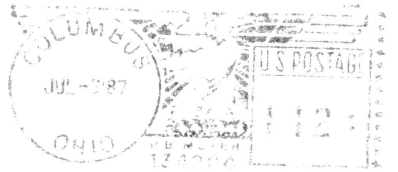
 This series will bring to you some of the finest offerings from past issues of the "SPIRIT OF 99" newsletter. For the next several issues, we plan to present one (or sometimes two) articles of varied content. Enjoy them!



"THE NEW COLOSSUS" by Emma Lazarus

NOT LIKE the brazen giant of Greek fame,
 With conquering limbs astride from land to land
 Here at our sea-washed, sunset gates shall stand
 A MIGHTY woman with a torch, whose flame
 IS THE imprisoned lightning, and her name
 Mother of Exiles. From her beacon-hand
 Glows world-wide welcome; her mild eyes command
 THE AIR-bridged harbor that twin cities frame.

"KEEP ANCIENT lands, your storied pomp!" cries she
 With silent lips. "Give me your tired, your poor,
 YOUR HUDDLED masses yearning to breathe free,
 The wretched refuse of your teeming shore.
 SEND THESE, the homeless, tempest-tost to me,
 I lift my lamp beside the golden door!"



AIR MAIL

**TIME SENSITIVE MATERIAL
POSTMASTER - PLEASE DELIVER PROMPTLY**

U.S. Postage
PAID
COLUMBUS 43221
Permit No. 1945

UK TI 99/4A USERS GROUP EXCH
HTON
ENGLAND,

***** MEMBERSHIP APPLICATION *****

NAME _____ AGE _____
ADDRESS _____
CITY _____ STATE _____ ZIP _____
AREA CODE _____ HOME PHONE _____ BUSINESS PHONE _____ EXT# _____
WHAT IS YOUR PROFESSION/VOCATION _____
HOW LONG HAVE YOU OWNED YOUR COMPUTER _____
DATE OF APPLICATION _____ ACCEPTED BY _____