

NEW HAMPSHIRE 99'ERS
NOVEMBER 1984
NEWSLETTER



NHUG NEWSLETTER - PO BOX 7199 HEIGHTS STATION - CONCORD, NH 03301

<<<<<POTPOURRI>>>>>

We are Pleased to say that our club roster continues to grow as word gets out that NHUG is indeed alive and well. Our current active membership stands at 42. Fifty-four Percent have "full blown" systems, while twenty-four Percent have Extended systems and twenty-two Percent own the "basic" system. In each NHUG NEWSLETTER we endeavor to include articles with respect to these Percentages. You can help by Providing us with ideas, articles, questions, etc. that you would like to see included.

We have heard from the NH Baileys, Rules, Belairs and this month's NHUG NEWS includes a review by one of our newest members, Edward Aho of New Ipswich. We gobble UP (Pun intended) this input... but, we'd love to hear from all of you!

By the way, Edward Aho is currently employed as a Programmer/Analyst using a computer system in which most of the Programming is done in COBOL. Assembly Language on this computer and the TI99/4A use the same instructions and Edward has been studying AL on both systems. He owns a "full blown" 99/4A and is currently dabbling with FORTH. We thank Edward for Providing us with his review of Republic Software's UTILITIES I, and hope that this is only the first of many submissions.

Since we mentioned FORTH, we would like to remind you that we only have one manual left. Because it is a lengthy document to photocopy and collate, PLEASE reserve your copies NOW!

More on FORTH: B. DALTON BOOKSELLERS at the Mall of NH (S. Willow Street-Manchester) currently stocks 4 titles on the FORTH LANGUAGE including Leo Brodie's STARTING FORTH (\$18.95). According to the LEHIGH 99'ers (Allentown, PA), Mr. Brodie has a second manual, THINKING FORTH which they feel "may be the single best book on any computer language" which they have seen. "It features a running commentary about the book, FORTH, and FORTH style by FORTH's creator Moore and others -- making for the fastest and most effective teaching" that they've seen.

THINKING FORTH by Leo Brodie. Prentice-Hall (\$15.95). We haven't found it, as yet, in the bookstores, but they will order it.

Notes on the last meeting... The RESTON book order was received and distributed. We appreciate RESTON's Prompt attention to our order.

A Proposed cartridge (module) loan Program was discussed. Suggestions on how we might run this are welcome!

We voted to Join the 99'ers USER GROUP ASSOCIATION. This organization was founded to Provide an information channel to owners of the 99/4A after TI's departure from the market. They have established a

newsletter as a communication channel to member Users Groups. The staff consists of ten Coordinators and seven Staff Analysts/Writers. They are currently compiling a FORTH Program library which will be made available to the User Groups when completed. We will keep you apprized.

Ellen Rule discussed the SAMS books. These books include a cassette of the Programs which appear in the hardcopy version. You can then use the book for reference or to help make modifications to the Programs.

BASIC TRICKS FOR THE TI99/4A by Wyatt is a collection of some thirty subroutines and Programs which can be used when writing your own Programs. There are routines to set up report formatting, menu screens, converting upper case to lower case letters (and vice versa), four methods to sort array variables, centering headings, etc.

TI99/4A BASIC PROGRAMS by Knight and Labatt contains thirty Programs in categories such as Games, educational, graphics, music and sound effects, and utilities.

These "books" are available locally, usually discounted (K-Mart, Service Merchandise, Treisman's etc.), and make fine Christmas gifts.

Richard Bailey demonstrated his DISKMASTER Program. DISKMASTER is a menu-driven, interactive Program loader in TI Extended BASIC that will allow you to: 1)select and run a Program from disk; 2)delete any Program on disk; 3)print the disk catalog. Instructions plus ample checks and Prompts are included, making the Program easy to use and virtually foolproof. For further information send Richard a SASE (68A Church Street-Gonic 03867).

Roger Roy demonstrated how he used FORTH to modify the FORMA4800A file from the TI WRITER ENHANCEMENTS to change the Printer devicename to PIO.LF:

```
After inserting the EDITOR/ASSEMBLER module select
"3" (LOAD and RUN). When the Prompt "FILENAME?"
appears, [ENTER] DSK1.FORTH. After the FORTH "OK"
Prompt flashes use the FORTH command LIST to find the
screen (disk sector) where the file is located. The
format is "15 LIST" to list screen 15. Once the screen
is found, load the editor by [ENTER]line -EDITOR.
[ENTER] 15 EDIT. Locate the cursor over the "R" in
RS232/1.... Replace this with your Printer devicename
(in his case PIO.LF). Press FCTN BACK to leave the
EDITOR, and [ENTER] FLUSH to write the change to the
disk. Copy the file (remember it has two parts) with
DISK MANAGER using the new filename of EDITA1 and
EDITA2. Now you are ready to use it with TI WRITER.
```

Roger notes, if you need this modification but do not have FORTH and have purchased the ENHANCEMENT disk from the club, that he will do it for you provided that you forward him:

1. an initialized disk
2. a disk mailer
3. one dollar
4. your devicename

Roger Roy - 31 Congress St - Nashua 03062

<<<<<BULLETIN BOARD>>>>>

>CHILD WORLD (Nashua Mall) - last we checked they had five TI DISK CONTROLLERS in stock. \$140.

>CORCOMP 1255 N. Tustin Ave - Anaheim, CA 98207. THE CORCOMP CURSOR arrived in our PO Box last week. It is a newsletter-type brochure describing the history of CORCOMP and their products. It includes an article on the 9900 Disk Controller (which they claim is 100 o/o compatible with all console versions, thermal Printers, and Foundation's 128k card - a FORTH LOADER is now on all 9900 Disk Managers) and TI FORTH.

>K & R Marketing (603-888-1242) - PO Box 1205 - Nashua 03061-1205. Their updated Price list of 99/4A hardware and software currently available for immediate delivery includes: 1) the console @ \$69; 2) 32k card (PHP1260) @ \$129; 3) PHM3026 Extended BASIC @ \$84; and 4) PHM3055 Editor/Assembler @ \$34.

>DATA FLEX SOFTWARE - 4420E 100N - Marion, IN 46952. Announces DFX-Print, a bit image screen Print utility which they claim is the most universal, the most useful, and the easiest to use of all currently available screen Printing utilities for the 99/4A. They offer a 7 day money back guarantee. A "full-blown" system is required + a dot matrix printer. \$24.95 for disk and \$19.95 for cassette.

>MICROPENDIUM - PO Box 1343 - Round Rock, TX 78680. According to E. Butcher TI Riverside (CA) User Group: MICROPENDIUM is a 30+ page paper with lots of the advertising that we used to see in HCM. It has product reviews, third-party news, user suggestions, and columns of interest for the users of the 99/4A. The paper seems to have fresh information on our system. \$12 Per year. We have order blanks.

>EXELTEC, INC. PO Box 54380 - Lubbock, TX 79453 (806/794-9104). Formerly Sunware Ltd., Exeltec has secured the rights to build the TI EXTENDED BASIC module and manuals. They "offer a completely interchangeable EXTENDED BASIC module". \$99.95 for 1 and \$79.95 if 5 or more are ordered.

>TIGERCUB SOFTWARE - 156 Collingwood Ave - Columbus, OH 43213. Tigercub is a kitchen table software company that specializes in low-cost programs in BASIC for those who have only console BASIC + a cassette recorder. (EXTENDED BASIC versions are also available). Each program is thoroughly debugged and user-friendly - all absolutely original - new and different. You can get a feel for Jim's work by typing out the listings which appear in this issue. We recommend that you write for his catalog (\$1.00 which is refundable on your first purchase) by writing to the above address. Most orders will get you extras as Jim seems to fill up the remaining disk/cassette space with freebies. (Ed. note: I mail-ordered from Tigercub on 11/2... on 11/7 the disk was in my mailbox + I received a coupon which entitles me to a free program of my choice with my next order of \$12 or more!). A hearty THANKS to Jim for allowing us to reprint TIPS FROM THE TIGERCUB!

All of the brochures, catalogs, etc. are available for your viewing at the meetings. Please remember to mention that you belong to NAUG when ordering from these companies... It helps us and helps you!

REMINDER: Our meetings are generally held on the last Wednesday of the month. The next meeting will be held in the function room of PABLO'S at 7:30 P.M. on WEDNESDAY, NOVEMBER 28th. We will ask for a show of hands relative to scheduling a December meeting.



A SOFTWARE REVIEW
BY Edward Aho

The software package "Utilities I" by Republic Software comes with a disassembler, point plotting routines, a speech utility and an 28 page instruction manual.

If you order the disk version you will receive the disk with the following files; DIS, PDMLDISK, PLOT, PLOTDUMP, SCREENDUMP and SPEAKOUT1.

The disassembler, in addition to the TI99/4A (or TI99/4) home computer, Monitor (or RF modulator and TV), requires a cassette recorder and cables or disk system (depending on which media you request), and either the Extended Basic, Mini-memory or Editor/Assembler cartridge. This is because the disassembler is written in Console Basic, but uses PEEKs to read the various memory locations. The disassembler does not create compilable source code but only translates what it sees into mnemonics without the parameters, for this reason it can mistake data as machine code instructions. Also because it is written in Console Basic it runs rather slow.

The plotting routines work by modifying character definitions as the user plots points. There are different versions, a TI-Basic, Extended Basic and Assembly Language version of these routines. The file "PDMLDISK" is the Assembly Language version and works much faster than the others. You can use this version in your own programs by using the "CALL" statement to link your program to the various subroutines within the program. The screen dump portion of this program works extremely well. It prints the screen right side up on the printer, while the Basic versions will print a sideways image. The files "PLOT" and "SCREENDUMP" require the Extended basic module and are intended to be used as subroutines that are merged into your own programs. The line numbers for "SCREENDUMP" start at 10000 while the line numbers for "PLOT" start at 10500. These routines, needless to say, work much slower than the Assembly Language version. The file "PLOTDUMP" is the Console Basic version and does not require any of the modules, and only requires the use of a printer if you want to use the screen dump portion.

The one thing I noticed about the plotting routines that I did not like was that as characters are redefined they can be used up quite quickly. When drawing a straight line the computer did not always check to see if a character was already defined that would fit for the portion of the line it was drawing and therefor used up characters unnecessarily. Because the program redefines characters any text that you may have displayed may be redefined on the screen as well. If you plot a point in an existing character, every existance of that character on the screen will in most cases be changed as well.

The file "SPEAKOUT1" requires the Extended Basic module and Speech Synthesizer. Optionally you may wish to use a printer and the Speech Editor module. This routine allows you to key in numbers representing one of 45 phonemes or sounds to have the computer speak a word and then gives you an option that allows you to "smooth" the word out. One the word sounds correct you may have the program show you the data necessary to make the speech synthesizer speak the word with the Extended Basic "CALL SAY" subroutine. Using this program you can teach the computer to say words with your Extended Basic program that are not in the Speech Synthesizers built-in vocabulary.

The instruction manual appears to be understandable and well written.

At the time I ordered my copy of the "Utilities I", I paid \$19.95 including postage. The programs and documentation can be ordered from;

Republic Software, Inc.
P.O. Box 23042 L'Enfant Plaza
Washington, D.C. 20024

CONVERTING TO TI BASIC

PART II

by Ellen Rule

As promised, I am going to tell you all I know about PEEK and POKE and converting them to TI BASIC. Luckily, few generic programs utilize these functions; many authors in fact avoid them in order to ease translation. Like the CALL statements in TI BASIC, however, they are used to expedite or enhance the program or else to perform some system function (i.e.: CALL CLEAR).

The first step in translation is to determine what the function accomplished in the original program. This may be apparent in the context of the program, or the reference manual for the specific computer may clarify the statement. David Brain (et al) have compiled Peeks and Pokes from many reference manuals into THE BASIC CONVERSIONS HANDBOOK FOR APPLE, TRS-80, AND PET USERS. Since TRS-80 BASIC is quite similar to TI BASIC, the manual brings the translation from Apple or PET one step closer to TI BASIC. In correspondence, Paul Paquin, one of the authors of the handbook, advises:

...Be aware that many times [peek and poke] are addressing a binary loaded file which is running at the same time. Or a piece of special hardware like a printer, a modem or graphics card. Whenever you translate you should always try to write a general flow chart as to what the program is doing. Then translate the sections one at a time. At that point if you get stuck with something, you can easily write something which will replace it.

Mr. Paquin also advises that he is willing to discuss problems and can be reached by leaving a message at Computer Mart (882-2439) in Nashua.

Recently I translated a game program that displayed a grid on the screen, with prompting statements printed below. For each move, a prompting statement was displayed, input was keyed in, the screen would clear and a new grid rolled up. Needless to say, this quickly became tedious and the game seemed QUITE dull. Fortunately, I stumbled across a little routine in Howard Budin's SPEED WALKER that converts an alpha-numeric statement into a series of graphics characters and displays them in the graphics mode:

```
100 REM TEXT ON SCREEN
110 M$="This message will be displayed at row R,
    column C."
115 REM assign values to R and C
120 FOR J=1 TO LEN(M$)
130 L$=SEG$(M$,J,1)
140 A=ASC(L$)
150 CALL HCHAR(R,C+J,A)
160 NEXT J
```

If more than one message will be displayed, assign the values of C, R and M\$ in the main program, then utilize lines 120-160 above as a subroutine to be called whenever a message is displayed.

The CALL KEY function can be used to accept single-character input from the screen in a function similar to ACCEPT AT:

```
100 CALL KEY(C,K,S)
110 IF S=0 THEN 100
120 INPUT$=CHR$(K)
```

In the case of the game played on a grid, Up, Down, Left, Right (or North, South, East, West) are entered as commands using the first letter of each word. The "3" (in line 100) returns all alpha input as upper case, alleviating the bugaboo of the alpha-lock being in the wrong position. If longer input is absolutely required, you can use the TI INPUT statement, then clear the screen and continue using the display routine I've described above. (Remember to clear the screen and redisplay after any PRINT statements). If the program uses simple graphics such as X's and O's, use your creativity and redefine the symbols into unique graphics characters. Then use the CALL HCHAR statement to create the display.

Generally speaking, if the program you want to convert is a "number cruncher" with few graphics, the translation should go smoothly. Expect to rewrite a program with heavy use of system-unique graphics. Block out a flow chart before you start; if the program is poorly documented or is disorganized, fix it! While you're at it, utilize TI's special functions such as CALL COLOR, KEY and SOUND to make your program more interesting. You'll be amazed at how much you'll learn!



MAKE YOUR OWN TOUCH-TONE PHONE

To generate the two simultaneous tones needed for touch-tone I have the computer build the tone table from data statements each time the program is run rather than entering 20 definitions of the form $F_1=941$, etc.. This method of building the table becomes more attractive in programs where you might want a large table of numbers.

If your phone can accept the touch-tone codes, just hold the handset to the speaker on your tv/monitor and press the numbers you want converted to tones on your keyboard. You may need to experiment with the volume setting but it will work. By itself this routine isn't too useful but it can be used as part of an address/phone dialing program.

```
100 REM CREATE TONE TABLE
110 REM RICHARD J. BAILEY
120 FOR N=0 TO 9
130 READ F1(N),F2(N)
140 NEXT N
150 DATA 941,1336,697,1447,770,1209,6
97,1336,697,1447,770,1209,77
0,1336,770,1447,852,1209,852
1336,852,1447
```

```
160 REM NUMBERS TO TONES
170 CALL KEY(0,K,S)
180 IF S=0 THEN 170
190 IF (K<48)+(K>57) THEN 170
200 CALL SOUND(200,F1(K-48),0,
F2(K-48),0)
210 FOR I=1 TO 200
220 NEXT I
230 GOTO 170
```



NEW HAMPSHIRE BIVERS
USER GROUP, INC.

TIPS FROM THE TIGERCUB #15

TIGERCUB SOFTWARE
156 Collingwood Ave.
Columbus OH 43213
(614) 235-3545

Copyright 1984 Tigercub Software. May be reprinted by non-profit publications with credit to Tigercub Software. Distributed to TI-99/4A Users Groups for promotional purposes and in exchange for their newsletters.

Tigercub Software has over 130 original programs for the TI-99/4A in Basic and Extended Basic, on cassette or disk, at only \$3.00 each. A descriptive catalog is available for \$1.00, which may be deducted from your first order.

I am presently writing Extended Basic versions of many of my programs, enhancing some programs and adding some new ones, and will soon be publishing a new catalog.

No, the Tips are not available by subscription, and I do not have back issues available. However, the entire contents of Tips #1 through Tips #14 are now available on disk, with more added - a full disk of 50 programs, routines and files for only \$15.00 postpaid.

If your local school has purchased TI-99/4A computers for their classrooms, why not let them know that Tigercub has educational programs at a price they can afford?

Since no one else ever reviews any of my software, I'll have to do it myself. **WHITENATER RUN** is a raft trip down Whitenater Canyon, in 7 levels of difficulty. You must avoid running aground on the green shores, stay away from the black rocks, and do your best to stay out of the whitewater which may conceal other rocks. Every trip is different and none are impossible. The Greenhorn level is a short leisurely ride just for practice. Raftsman is a longer ride which surprises you with a sudden surge of fast water toward the end. Voyageur is a complete change of pace, as your raft leaps through the rapids. The next three levels are much

faster, each with an increasing number of hidden rocks in the whitewater. Long Journey takes you through all six levels. Now also available in an Extended Basic version, very fast.

Most of the speed reading programs on the market are worthless, because they flash a phrase from their data file onto the screen and then require you to retype it exactly. Even if the data file of phrases is large, they will soon be recognized from memory rather than from reading. Also, the purpose of speed reading is to quickly grasp the meaning of a text, not the exact wording.

SPEEDER READER avoids these faults by assembling each sentence at random from files of nouns, verbs, adjectives, adverbs and modifiers. The result is an infinite number of different sentences, always grammatical but usually ridiculous. The sentence is flashed on the screen for whatever interval you select; then you are asked any one of several randomly selected questions about it. Where did the bald ballerina kiss the fat judge? If you can keep from laughing too much, your reading speed can be increased greatly.

JUNIOR SPEEDER READER is the same except that the sentences describe the activities of various animals, in simpler words.

Several people have sent me enhancements to my Menu Loader, and I greatly appreciate them. The trouble is, if I incorporated them all the program would take up about 25 disk sectors! So, I have borrowed some ideas, added a few of my own, and here is the result. It will list and load up to 99 programs, stopping at the end of every screenfull or stopping whenever any key is pressed and then offering you the choice of loading, deleting or quitting. It will ask you to verify a deletion by name before deleting it, and will display the name of the program it is loading. It also contains a feature to warn you if you are getting a bad count of disk sectors used - which I find happening more often than you might realize.

100 !by A. Kludge/M. Gordon/
T. Boisseau/J. Peterson/etc.
110 CALL CLEAR :: CALL INIT
:: CALL LOAD(8196,63,248)::

```
CALL LOAD(16376,67,85,82,63,  
79,82,48,8)  
120 CALL LOAD(12288,129,195,  
126,165,129,153,102,60)  
130 CALL LOAD(12296,2,0,3,24  
0,2,1,48,0,2,2,0,8,4,32,32,3  
6,4,91):: CALL LINK("CURSOR"  
)  
140 CALL CLEAR :: CALL SCREE  
N(5):: FOR S=1 TO 14 :: CALL  
COLOR(S,7,16):: NEXT S :: C  
ALL VCHAR(1,31,1,96):: CALL  
COLOR(0,2,16)  
150 OPTION BASE 1 :: DIM P$(  
199),T$(5)  
160 T$(1)="dis/fix" :: T$(2)  
="dis/var" :: T$(3)="int/fix  
" :: T$(4)="int/var" :: T$(5  
)="program"  
170 IMAGE ##  
180 DISPLAY AT(1,9):"DISKETT  
E MENU"  
190 ! IF YOU HAVE MORE THAN  
ONE DISK DRIVE, DELETE THE !  
IN LINE 200 AND THE FIRST S  
TATEMENT IN 210  
200 ! DISPLAY AT(12,6):"DISK  
? (1-3):" :: ACCEPT AT(12,19  
)SIZE(-1)VALIDATE("123"):D$  
:: D$="DSK"&D$&"."  
210 D$="DSK1." :: OPEN #1:D$  
,INPUT,RELATIVE,INTERNAL ::  
INPUT #1:N$,A,J,K :: DISPLA  
Y AT(1,2):SEG$(D$,1,4)&" - D  
iskname= "&N$;  
220 DISPLAY AT(2,2):"Availab  
le=";K;"Used=";J-K:" Prog Fil  
lename Size Type":"----  
-----" ::  
I,V,T=0 :: TT=J-K  
230 FOR X=1 TO 99 :: IF X/20  
<>INT(X/20)THEN 260  
240 DISPLAY AT(24,1):"Type c  
hoice or 99 for more" :: ACC  
EPT AT(24,27)VALIDATE(DIGIT)  
:K :: IF K=99 THEN 250 :: IF  
K>0 AND K<NN+1 THEN 420 ELS  
E 240  
250 X=1  
260 I=I+1 :: IF I>127 THEN K  
=X :: GOTO 360  
270 INPUT #1:P$,A,J,B :: NN=  
NN+1  
280 IF LEN(P$)=0 THEN 320  
290 DISPLAY AT(X+4,2):USING  
170:NN :: DISPLAY AT(X+4,6):  
P$ :: P6$(NN)=P$ :: DISPLAY  
AT(X+4,18):USING 170:J :: DI
```

```

SPLAY AT(X+4,22):I$(ABS(A)):
: VT=VT+J
300 CALL KEY(0,KK,ST):: IF S
T=0 THEN 310 :: FLAG=1 :: 60
TO 320
310 NEXT X
320 DISPLAY AT(X+4,1):" " ::
  DISPLAY AT(X+4,2):USING 170
:NN :: DISPLAY AT(X+4,6):"Te
rminate" :: DISPLAY AT(X+5,2
):STR$(NN+1)&" Delete"
330 IF VT=TT OR FLAG=1 THEN
350 :: DISPLAY AT(2,25)SIZE(
4):VT
340 FOR @=1 TO 10 :: DISPLAY
AT(2,25)SIZE(1):CHR$(30)::
DISPLAY AT(2,25)SIZE(1):" "
:: CALL SOUND(-99,110,0,-4,0
):: NEXT @
350 DISPLAY AT(X+6,1):" C
hoice?" :: ACCEPT AT(X+6,16)
SIZE(2)VALIDATE(DIGIT):K ::
IF K<>NN AND K<>NN+1 THEN 41
0
360 IF K=NN THEN CALL CLEAR
:: CLOSE #1 :: END
370 DISPLAY AT(X+5,11)SIZE(1
8):" #?" :: ACCEPT AT(X+5,15
)SIZE(2)VALIDATE(DIGIT):KD ::
IF KD<1 OR KD>NN THEN 370
380 DISPLAY AT(X+6,1)SIZE(27
)BEEP:" Verify - Delete ":PG
$(KD);"? " :: DISPLAY AT(X+6,
28)SIZE(1):"Y" :: ACCEPT AT(
X+6,28)SIZE(-1)VALIDATE("YN"
):Q$ :: IF Q$<"Y" THEN 400
390 DELETE D$PG$(KD)
400 CLOSE #1 :: CALL VCHAR(1
,3,32,672):: NN=0 :: X=0 ::
GOTO 180
410 IF K<1 OR K>99 OR LEN(PG
$(K))=0 THEN 320
420 CLOSE #1
430 CALL INIT :: CALL PEEK(-
31952,A,B):: CALL PEEK(A#256
+B-65534,A,B):: C=A#256+B-65
534 :: A=D$PG$(K):: CALL L
OAD(C,LEN(A$))
440 FOR I=1 TO LEN(A$):: CAL
L LOAD(C+I,ASC(SEG$(A$,1,1)
)): NEXT I :: CALL LOAD(C+I,
0)
450 CALL VCHAR(1,3,32,672)::
CALL SCREEN(8):: FOR S=0 TO
14 :: CALL COLOR(S,2,1):: N
EXT S :: DISPLAY AT(12,2):"L
ODDING ":A$
460 RUN "DSKX.1234567890"

```

If you don't like my Tigercub cursor, just delete lines 110 (after the CALL CLEAR), 120 and 130. That routine for redefining the cursor has appeared recently in various newsletters without attribution, and I'd like to know who to credit for it. The secret of it is in line 120, where the numbers after 12288 are the decimal equivalents of the hexadecimal numbers (which are the hex equivalent of the binary numbers represented by the off/on pixels) used to redefine a character.

You may have noticed that all programs published in the Tigercub's Tips are in 28-column format, just the way they will appear on the screen. And they are printed directly from LISTed actual programs, so that they cannot contain typographical errors - don't you wish the computer magazines did that!? The problem is that when a program listing is merged into the TI-Writer buffer and printed in the formatter mode, the @, &, \$ and the exponent sign are treated as control characters, and strange things happen!

The following program will convert a program, which has been listed to disk with LIST "DSK1.FILENAME", into a file in 28-column format which can be loaded into TI-Writer, and will optionally substitute the left and right braces, ASCII 124 and the tilde for the @, &, \$ and the exponent sign, and transliterate them so that they will print correctly in the formatter mode. However, for that very reason this program will not print correctly! When you come to line 280, type DATA shift 2, fctn F, shift 7, fctn G, shift 6, fctn W, shift 8, fctn A.

```

100 DISPLAY AT(2,4)ERASE ALL
:"28-COLUMN CONVERTER" :: DI
SPLAY AT(5,12):"by Jim Peter
son"
110 DISPLAY AT(7,1):" To con
vert a program, saved": "with
LIST ""DSK1.FILENAME",": "i
nto 28-column format which":
"can be merged into the text
"
120 DISPLAY AT(11,1):"buffer
of TI-Writer."
130 DISPLAY AT(13,1):" Optio
nally with transliter-": "ate

```

```

d @, &, $ and ^ for cor-": "r
ect printing from formatter"
:"mode."
140 DISPLAY AT(18,1):" Do yo
u want to print the": "file f
rom the": " (E)ditor?": " (F)
ormatter?"
150 ACCEPT AT(23,1)VALIDATE(
"EF")BEEP:0$
160 DIM A$(1000):: CALL CLEA
R :: INPUT "What is the FILE
NAME? DSK1.":FN$ :: FN
$="DSK1."&FN$ :: PRINT ::
170 INPUT "what is the new F
ILENAME? DSK1.":PN$ :: PN$
="DSK1."&PN$ :: OPEN #1:FN$,
DISPLAY ,VARIABLE 80,INPUT :
: OPEN #2:FN$,DISPLAY ,VARI
ABLE 80,OUTPUT
180 IF Q$="E" THEN 190 :: PR
INT #2:".TL 126:94;" :: PRIM
T #2:".TL 123:64;" :: PRINT
#2:".TL 125:38;" :: PRINT #2
:".TL 124:42;"
190 FOR L=1 TO 1000 :: LINPU
T #1:A$(L):: IF LEN(A$(L-1))
=80 OR LEN(A$(L-1))=160 THEN
A$(L-1)=A$(L-1)&A$(L):: L=L
-1
200 IF EOF(1)THEN L=L+1 :: G
OTO 220
210 NEXT L
220 FOR J=1 TO L-1 :: S=1
230 FOR T=1 TO 10 :: B$(T)=S
EG$(A$(J),S,28):: IF Q$="E"
THEN 240 :: GOSUB 280
240 S=S+28 :: NEXT T
250 FOR N=1 TO 10 :: IF B$(N
)<>" THEN PRINT #2:B$(N)
260 NEXT N
270 NEXT J :: CLOSE #2 :: CL
OSE #1 :: END
280 DATA @, @, &, &, ^, ^, $, $
290 RESTORE 280
300 FOR W=1 TO 4 :: READ CH$
, F$
310 X=POS(B$(T),CH$,1):: IF
X=0 THEN 330
320 B$(T)=SEG$(B$(T),1,X-1)&
R$&SEG$(B$(T),X+1,LEN(B$(T)
)): GOTO 310
330 NEXT W :: RETURN

```

Now, if that's what I give away, isn't it worth a dollar to find out what I'm selling?

Happy hackin'

Jim P.

USING DISKS
AND DISK DRIVES

Richard J. Bailey
68A Church Street
Gonic, N. H. 03867

Most of us take our disks and drives for granted until we have a problem or some error message pops up on the screen. The instruction manuals that come with the disk drive/disk controller must assume you'll never get an error running the disk test and that all sectors will initialize properly because they never tell you what to do if there is an error. I'll try to shed some light on the subject with what information I've been able to find either in literature or by experience. I'll also try to give some information on how programs are put on disks and drive differences.

A disk is divided into concentric tracks and each track is divided into sectors. Most T.I. drives have 40 tracks of 9 sectors with each sector containing 256 bytes of user space. 40 tracks times 9 sectors/tracks gives 360 sectors/disk and 360 sectors/disk times 256 bytes/sector gives 92,160 bytes/disk. On the outside track of 9 sectors (sectors 0-8), sectors 0&1 are used for the directory. If you can't read/write to sectors 0&1 the disk is unusable except as a frisbee. The remaining 358 sectors are for your files and programs. If the disk manager can't initialize one of these sectors it crosses them off the list of usable sectors for that disk. Watch the sector numbers increment on the screen when the disk is being initialized and listen to the drive. You will hear the head step to the next track every 9th sector. When a bad sector is encountered the drive will try to initialize that sector three times and if after the third try it isn't initialized, it crosses the sector off and continues to the next sector. When this happens you will hear the head drive in and out as it tries to initialize that sector and the sector numbers will stop incrementing until it continues to the next sector. When the disk manager finishes initializing a disk you should see the message AVAILABLE=358 USED=0. If 7 sectors were unusable you would see AVAILABLE=351 USED=7 indicating the number of bad sectors.

Running a comprehensive test on a disk with bad sectors will probably show more defective sectors than were found when the disk was initialized. The comprehensive test is more severe and exacting than normal use. Any disk that passes the comprehensive test is really good but if a disk has bad sectors it doesn't mean that the disk is unusable. It will depend on how demanding you are as to what you consider unusable. You obviously can't back-up a single-sided disk that fills 357 sectors on a disk that has 12 (or even 2) bad sectors but you can use all the sectors that were initialized. The T.I. checks all data written to disk so there isn't much chance that a program you write to disk will be unreadable.

You might want to make a note on any new disk you get that has bad sectors. You may find (as I did) there can be a big difference in disk brands. Out of 10 disks I got from one supplier, 3 had bad sectors when initialized with my T.I. drive. The comprehensive tests indicated the sectors were scattered but running the test twice showed the errors were repeatable indicating that the drive was probably ok. The oxide formulation on these disks was apparently different from the ones I had been using and they were harder to write to. I tried having the 3 questionable disks initialized and tested on another system (Atari 800) and they all passed. They would also pass on my second drive (an MPI-52) with no problem. There obviously can be differences amongst the various drives available that can also cause problems. My

conclusion was that all the drives and all the disks were pretty much ok but there were differences. If you think you are having a disk problem, try having a friend test the disk on her or his system to check it out. You may want to stick with one brand that works for you. Also read the article COMPARING FLOPPY DISKS in the September , 1984 issue of BYTE magazine for more detailed information on disks.

Trying to save a program to a disk that doesn't have enough room will give a DISK ERROR 64 and leave part of the program on disk in an unusable format. If this happens you should type (in the immediate mode) DELETE "DSK1.FILENAME" to erase that partial program and free up those sectors it occupies. When you save a program to disk, it won't necessarily be put on consecutive sectors. If you've previously deleted programs freeing up sectors here and there, you may end up with pieces of your program scattered on several different tracks as the T.I. tries to use all sectors available. Normally you don't have to worry about this because the directory on sectors 0&1 does all the housekeeping for you. However you may find your drive seems to "search" to find all the bits and pieces of some programs. To correct this problem simply back up that disk with the disk manager module. All programs on the backup disk are put in alphabetical order on concentric tracks instead of the semi-random way they were entered on the original disk. Backing up a disk can cut down somewhat on drive wear even though the Shugart 400L drive T.I. uses is an excellent long-lived drive.

Some drives have 35 tracks instead of 40 giving them 35 times 9 or 315 sectors. Sectors 0&1 are the directory just as on the 40 track drives but you have only 313 sectors available for your programs and files. All these sectors are useable to a 40 track drive but a 35 track drive cannot access the 5 innermost tracks on a 40 track disk. Any program that occupies sectors 315-359 cannot be read, etc.. Other than this limitation the drives are totally compatible.

Using a double-sided 40 track drive you'll have sectors 0-359 on the front or label side the same as with a single-sided drive and all these sectors can be read, etc. with a single-sided 40 track drive. The backside has sectors 360-719 numbered from the center of the disk out and these sectors are inaccessible to a single-sided drive. The directory will list all programs on both sides but unless the disk has been backed-up to put the programs in alphabetical order on consecutive tracks, there is no way to tell which side any program is on.

Flippy-floppy disks are available for single-sided drives so you can use both sides of the disk. You can also punch mirror-image index holes and write protect notches on a double-sided disk so it can be flipped. Each side is like a separate disk with its own directory. The back side can only be accessed by flipping the disk over, even if you have a double-sided drive.

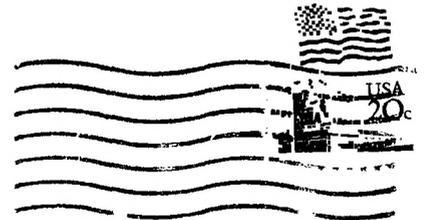
The bottom line is a double-sided 40 track drive will read any single-density disk, whether single or double-sided, 35 or 40 tracks, but the reverse isn't true. If a disk uses more sectors than the drive can access, you'll get an error trying to access what the drive thinks are non-existent sectors.

If there is any interest I'll write an article on the mechanics of adding a second non-T.I. drive and how to check the drive speed with tools everybody has.

HOOKSETT
U.S. ROUTE 3
D.M. HIGHWAY NO.
PABLO'S RESTAURANT
7:00 P.M.
NOVEMBER 28
WEDNESDAY
NEXT MEETING



NEW HAMPSHIRE 99'ERS USER GROUP, INC.
P.O. BOX 7199, HEIGHTS STATION
CONCORD, NEW HAMPSHIRE 03301



FIRST CLASS MAIL

EDMONTON USER'S GROUP
PO BOX 11983
EDMONTON, ALBERTA
CANADA T6J-3L1