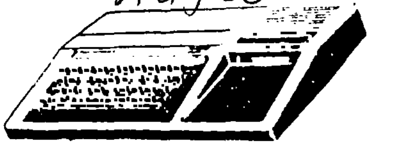
New Hampshire 99'ers

AUGUST 1986

Newsletter

New Hampshire
Aug 86

NH99UG NEWSLETTER - PO BOX 5991 - MANCHESTER, NH 03108-5991 VOL.4 NO. 8



>NEW

NO MEETING IN SEPTEMBER!

Because Labor Day falls on our usual meeting night, we are taking a vacation in September. We will take this opportunity to perfect some demonstrations (bulletin boards, for example) as well as get in some new and exciting equipment. You may still call, write, etc. during this time and I will be happy to respond. Our next meeting will be October 6th. See you there!

RAM DISKS

Concerning the New Horizon RAM disks: they are extremely easy to assemble. Had I possessed all the necessary pieces, I could have built my card in two hours. Believe me, I am not a technician by any stretch of the imagination! As soon as I pick up a few more chips, diodes, etc. I will have another valuable addition to my TI.

WHERE'S THE BEEF? Speaking of additions, has anyone seen or heard of a wonderful new program lately? It seems as if the current emphasis is on hardware. You can purchase any number of RAM disks, GROM emulators, etc. Even disk controllers are still making news - MYARC's controller may be modified to utilize quad density drives (720K on a disk?). My point is this: where are the fantastic programs? Since FUNLWRITER, DM1000, and some Millers Graphics (who else) programs, I haven't seen any earth shaking material. I am not surprised at all. The author of DM1000 sold his system. Tony and Will McGovern (authors of FUNLWRITER) have told me this is the last TI effort for a while. If you are one of the few individuals that pays for SHAREWARE (it's not free!) then ignore my sermon. If you are one of the many who takes and takes and never gives, then beware! It's not too late to repent.

RAFFLE

The club has an extra PEB thanks to Andre' Roy. This box will be raffled off at the December meeting. Tickets are $1.00 and may be purchased at any meeting or through the mail. Send checks and a note to the club P.O. Box. This should make a nice Christmas present for someone.

NEWSLETTERS!

Thank you, Eliot Hardy for taking on the responsibility of managing our newsletter library. We trade newsletters with a number of clubs, as well as subscribe to MICROpendium. If you have never checked out a packet of newsletters, you don't know what you're missing.

ARTICLES!

I am always open to articles from club members. I don't personally deal with FORTH, C, PASCAL, LOGO, PILOT, etc. If any of you have an article or program, please send it to me on a disk. Otherwise, you will be subjected to Assembly and the few BASIC and XBASIC things I did before I became addicted to Assembly.

BULLETIN BOARDS

There are several good bulletin boards in the area; some owned and operated by our own club members! Brian Wallace operates one on his Commodore, and Dave Villeneau operates one on his TI. Hopefully, we will see Dave's in action in September. There is also a good board operating out of Manchester by the name of The Progressive Connection. There are three numbers you may call:

644-3507
434-6225
485-4334

This board was originally run out of Londonderry. However, it has been dramatically upgraded over the last few months. If you were a subscriber, and let your subscription lapse, I urge you to check it out again. If you have never seen it - give it a call. We will publish more BBS numbers in the near future.

I LIED!

I said last month I would publish the program that converts listings into MERGEable files. Sorry, I lied! There are a few bugs in the original (MICROpendium) that have to be stomped first.

NEW ADDITION

The club recently purchased an Epson MX-80 printer with GRAFTRAX. Our system is nearing completion! Anyone have a color monitor they would like to sell to the club (OR DONATE)?

Several months ago, Home Computer Magazine (yes, that long ago) published a program which allows you to play the Tower of Hanoi game. They also challenged readers to solve the program and send in their solutions. Needless to say, the solutions never appeared. However, I did spend a considerable amount of time working on this - so you get it!

To play the game you must move each disk from one peg to the other and end up with the disks in the same order. There's only one catch: you can't put a disk on top of any disk that's smaller. To move disks, press key 1, 2, or 3 to select the pole from which you want to move a disk. Then press the numbers again to select the pole to which the disk will be moved. The graphics are handled by the program.

Finally, I have taken over dishing out the club's hardware. If you want anything, call me before the meeting. Better yet, write me a letter. I can't possibly bring all the club material to each meeting on the chance someone wants it.

See you in October!

LAST MINUTES

The meeting was called to order shortly after 7:30 by our venerable President Curtis Alan Provance with 19 members in attendance. Ellen was a bit under the weather and so she drafted this neophyte to do the honors.

Curtis announced that there would be a raffle later with the winner having the choice of either a box of Nashua Disks or Disk Master I, a disk manager system by DataBiotics, Inc. Tickets were 50 cents apiece. He passed around free brochures from Falcon Safety Products describing a product called "Dust-Off" and other preventative maintenance supplies for the care of electronic office equipment. Copies may be available at the next meeting or it can be ordered from Curtis for 50 cents handling and postage.

Curtis then announced that the brave souls who last month sent in a combined order for six of the Horizon Ramdisks could pick them up. The package included the board, a reference manual and an instruction guide. Dick Bailey made available at the same time a packet of parts that comprised about 50 percent of the parts needed for the board at $10 a packet - a very good deal for which they were thankful. However, they are on their own for the rest of the needed parts.

Demonstrations followed of four new products. "Spy's Demise" as rewritten by Curtis (and listed in the last newsletter) was run up. Disk Master I, v 4.9 from DataBiotics was demoed; it is written in Assembly Language and is a relatively fast disk management system. Also demo'ed was the new, not yet fully functional MYARC Extended Basic-II, v 2.0 that sells for around $79.95. This program includes exceptional graphic capabilities, as well as integer handling and other benefits. Unfortunately, some XBASIC capabilities are not yet supported: user written CALL's and DEF's for example. It should be well received when it is completed. It requires a 128K card to operate. Curtis also demo'ed the new GRAPHX Slide Show available from Asgard Software. The Slideshow program was written by Paul Charlton. The package consisted of two 'flippies' with very elaborate pictures, mostly in black and white. The cost for this package is 16.50 postage paid from Asgard.

Fianlly, Curtis attempted to convert a Foundation 128K card into a fully functinal RAM disk by swapping the 8K DSR chip. He used the MYARC chip from his MYARC card. Both cards utilize the sane bank switching addresses. Curtis hoped that the DSR used the CRU base address stored in workspace register 12 instead of explicitly loading it. Unfortunately, the experiment was less than successful. The RAM chip test (MYARC's CALL RDTEST) was the only subroutine that appeared to work properly. However, with some modification (minor?) the Foundation card's CRU base may be altered to function with the MYARC DSR. This will no doubt be necessary if you wish to purchase the MYARC Extended BASIC and you have the Foundation card.

Oh! Yes! Yours truly lucked out and won the Disk Master I System.

Bye......

Phil Davis

TOWER OF HANOI

Several months ago, Home Computer Magazine (yes, that long ago) published a program which allows you to play the Tower of Hanoi game. They also challenged readers to solve the program and send in their solutions. Needless to say, the solutions never appeared. However, I did spend a considerable amount of time working on this - so you get it!

To play the game you must move each disk from one peg to the other and end up with the disks in the same order. There's only one catch: you can't put a disk on top of any disk that's smaller. To move disks, press key 1, 2, or 3 to select the pole from which you want to move a disk. Then press the numbers again to select the pole to which the disk will be moved. The graphics are handled by the program. The listing appears on page four of this issue.

TIPS FROM THE TIGERCUB

#35

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 casette or disk, only $3.00 each plus $1.50 per order for PPM. Entertainment, education, programmer's utilities.
Descriptive catalog $1.00, deductable from your first order.
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 postpaid. Or, both for $27 postpaid.
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.
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 GAMES
REFLEX AND CONCENTRATION
TWO-PLAYER GAMES
KID'S GAMES
MORE GAMES
WORD GAMES
ELEMENTARY MATH
MIDDLE/HIGH SCHOOL MATH
VOCABULARY AND READING
MUSICAL EDUCATION
KALEIDOSCOPES AND DISPLAYS
For descriptions of these send a dollar for my catalog!

The April Micropendium had a rather slow routine to count the number of words in a D/V text file. I think the following will be much faster. It ignores any lines beginning with a period (TI-Writer formatter commands), otherwise counts each cluster of characters followed by a space, plus the last cluster on the line.

```
10 !WORDCOUNT by Jim Peterso
n
100 DISPLAY AT(12,1)ERASE AL
L:"INPUT FILENAME? DSK" :: A
CCEPT AT(12,20):F$ :: OPEN #
1:"DSK"&F$,INPUT
110 A=1 :: LINPUT #1:M$ :: I
F ASC(M$)=46 THEN 130
120 X=POS(M$," ",A):: IF X=0
 THEN 130 :: IF X=A THEN A=X
+1 :: GOTO 120 ELSE F=1 :: C
=C+1 :: A=X+1 :: GOTO 120
130 C=C+F :: F=0 :: IF EOF(1
)<>1 THEN 110 :: CLOSE #1 ::
 DISPLAY AT(12,1)ERASE ALL:"
APPROXIMATELY "&STR$(C)&" WO
RDS"
```

Have you tried those black write-protect tabs, made of a material similar to electrical tape? They do not become dog-eared from bumping against the drive slot, and do not leave the disk sticky when you remove them.

```
100 !TIGERCUB GRAPHPRINT by
Jim Peterson
110 !Will output to printer
a line graph of 31 items of
data, as for instance the
temperature for each day of
a month
120 !Values must be positive
 integers within a range of
75 from minimum to maximum
130 M$=RPT$("¦_",65):: DIM T
$(31),D$(75):: MN=10000
140 DISPLAY AT(12,1)ERASE AL
L:"Input data - maximum 31":
"items. Enter to finish"
150 FOR X=1 TO 31 :: DISPLAY
 AT(14,1):X;TAB(4);CHR$(1)::
 ACCEPT AT(14,4)VALIDATE(DIG
IT)SIZE(-5)BEEP:T$(X):: IF T
$(X)=CHR$(1)THEN X=X-1 :: GO
TO 170
160 T=VAL(T$(X)):: MX=MAX(MX
,T):: MN=MIN(MN,T):: NEXT X
170 RN=MX-MN :: IF RN>75 THE
N PRINT "EXCEEDS MAXIMUM RAN
GE OF 75" :: STOP
180 IF MX>75 THEN AD=MX-75
190 OPEN #1:"PIO",VARIABLE 1
32 :: PRINT #1:CHR$(15);CHR$
(27);CHR$(51);CHR$(12):: PRI
NT #1:RPT$("_",132)
200 DISPLAY AT(12,1)ERASE AL
L:"Wait, please...": :".....
.this takes time"
210 LM=LEN(STR$(MX)):: FOR J
=1 TO 75 :: J$=STR$(76+AD-J)
220 IF J>66+AD THEN J$=J$&"
"
230 IF J/2=INT(J/2)THEN D$(J
)=RPT$(" ",LM)&SEG$(M$,1,132
-LM)ELSE D$(J)=J$&SEG$(M$,1,
132-LM)
240 NEXT J :: PRINT #1:RPT$(
" ",LM)&SEG$(M$,1,132-LM)
250 J=1 :: T=VAL(T$(J))-AD :
: T=76-T :: D$(T)=SEG$(D$(T)
,1,J*4+4)&CHR$(239)&SEG$(D$(
T),J*4+6,255):: J=J+1
260 T2=T :: T=VAL(T$(J))-AD
:: T=76-T :: FOR N=T2 TO T S
TEP (T2>T)+ABS(T>=T2):: D$(N
)=SEG$(D$(N),1,J*4+2)&CHR$(2
53+(T<T2))&SEG$(D$(N),J*4+4,
255):: NEXT N
270 J=J+1 :: D$(T)=SEG$(D$(T
),1,J*4)&CHR$(239)&SEG$(D$(T
),J*4+2,255):: IF J<=X THEN
260
280 FOR J=1 TO 75 :: PRINT #
1:D$(J):: NEXT J :: PRINT #1
290 T=8 :: FOR J=1 TO 31 ::
PRINT #1:TAB(T);STR$(J);:: T
=T+4 :: NEXT J
```

When you are analyzing an Extended Basic program, or modifying it, it is often easier to work with single-statement lines. This program will break all multi-statement lines into single-statement lines, except when they are followed by IF or ELSE. When you are finished modifying, a Compactor or Smash program can be used to compact it again.

```
100 !DECOMPACTER by Jim Pete
rson
110 DISPLAY AT(3,5)ERASE ALL
:"TIGERCUB DECOMPACTER": :"
Program must first be -": :"
RES 100,100": :"SAVE DSK(fil
ename),MERGE"
120 DISPLAY AT(12,1):"INPUT
FILENAME?":"DSK" :: ACCEPT A
T(13,4):IF$
130 DISPLAY AT(12,1)ERASE AL
L:"OUTPUT FILENAME?":"DSK" :
: ACCEPT AT(13,4):OF$
140 OPEN #1:"DSK"&IF$,INPUT
,VARIABLE 163 :: OPEN #2:"DS
K"&OF$,OUTPUT,VARIABLE 163 :
: LN=100
150 LINPUT #1:M$ :: P=POS(M$
,CHR$(130),3):: IF P=0 THEN
PRINT #2:M$ :: GOTO 270
160 A$=SEG$(M$,1,P-1):: IF P
OS(A$,CHR$(129),1)<>0 OR POS
(A$,CHR$(132),1)<>0 THEN PRI
```

```
NT #2:M$ :: GOTO 270
170 PRINT #2:A$&CHR$(0)
180 AN=LN+1 :: GOSUB 280
190 M$=SEG$(M$,P+1,255)
200 P=POS(M$,CHR$(130),1)
210 IF P=0 THEN PRINT #2:LN$
&M$ :: GOTO 270
220 A$=SEG$(M$,1,P-1)
230 IF POS(A$,CHR$(129),1)<>
0 OR POS(A$,CHR$(132),1)<>0
THEN PRINT #2:LN$&M$ :: GOTO
 270
240 PRINT #2:LN$&A$&CHR$(0)
250 AN=AN+1 :: GOSUB 280
260 GOTO 190
270 LN=LN+100 :: IF EOF(1)<>
1 THEN 150 ELSE CLOSE #1 ::
CLOSE #2 :: END
280 LN$=CHR$(INT(AN/256))&CH
R$(AN-256*INT(AN/256)):: RET
URN
```

I still think of the TI as a HOME computer, and I still think that the home computer is an invaluable educational tool - but I guess not many folks agree with me. I had thought of writing full disks of a progressive series of lessons on one subject, but my present two full disks of math education have sold a combined total of 7 copies in 7 months, so that would obviously be a waste of time.

I had written this next program for that purpose and I guess it's no use wasting it, so -

```
100 CALL CLEAR :: CALL TITLE
(5,"TAKE AWAY")!by Jim Peter
son
110 DISPLAY AT(3,10):"COPYRI
GHT":TAB(10);"TIGERCUB SOFTW
ARE":TAB(10);"FOR FREE":TAB(
12);" DISTRIBUTION":TAB(11);
"SALE PROHIBITED"
120 CALL PEEK(-28672,A@):: I
F A@=0 THEN 150
130 DATA FINE,NO,GOOD,UHOH,R
IGHT,TRY AGAIN,YES,THAT IS N
OT RIGHT
140 FOR J=1 TO 4 :: READ RIG
HT$(J),WRONG$(J):: NEXT J
150 FOR D=1 TO 1000 :: NEXT
D :: CALL DELSPRITE(ALL)
160 CALL CLEAR :: CALL CHAR(
95,"FFFF"):: CALL MAGNIFY(2)
:: RANDOMIZE :: CALL SCREEN(
14):: FOR SET=5 TO 8 :: CALL
 COLOR(SET,16,1):: NEXT SET
170 CALL CHAR(120,"E7004200l
8007E0000E70042009942 3CE7004
20099423C00E70042180003C4200"
)
180 CALL CHAR(124,"0E0004010
00708007000208000E0100")
190 DISPLAY AT(3,10):"TAKE A
WAY" :: CALL CHAMELEON
200 CALL COLOR(14,2,2):: CAL
L HCHAR(4,4,143,2):: CALL HC
HAR(5,4,143,2):: CALL SPRITE
(#25,120,11,25,25)
210 T=T+1 :: N=1-(T>5)-(T>15
):: G=10-(T>5)*80-(T>15)*810
 :: H=0-(T>5)*10-(T>15)*90
220 X=INT(G*RND+H):: Y=INT(G
*RND+H):: IF Y>X THEN TT=X :
: X=Y :: Y=TT
230 IF X=X2 OR Y=Y2 THEN 220
 :: X2=X :: Y2=Y :: Z=X-Y
240 GOSUB 250 :: GOTO 210
250 GOSUB 260 :: GOSUB 280 :
: GOSUB 310 :: FOR D=1 TO 20
0 :: NEXT D :: CALL DELSPRIT
E(ALL):: DISPLAY AT(18,1)::
CALL CHAMELEON :: CALL SPRIT
E(#25,120,11,25,25):: RETURN
260 FOR J=1 TO LEN(STR$(X)):
: :: A(J)=VAL(SEG$(STR$(X),J
,1)):: NEXT J :: FOR J=1 TO
LEN(STR$(Y)):: B(J)=VAL(SEG$
(STR$(Y),J,1)):: NEXT J
270 FOR J=1 TO LEN(STR$(Z)):
: C(J)=VAL(SEG$(STR$(Z),J,1)
):: NEXT J :: W=LEN(STR$(Z))
-LEN(STR$(X)):: RETURN
280 R=96 :: CC=96 :: FOR J=1
 TO N :: CALL SPRITE(#J,48+A
(J),11,R,CC):: CC=CC+16 :: N
EXT J
290 R=116 :: CC=96 :: FOR J=
1 TO N :: CALL SPRITE(#4+J,4
8+B(J),11,R,CC):: CC=CC+16 :
: NEXT J
300 CALL HCHAR(18,12,95,N*3)
:: CC=CC-16 :: RETURN
310 R=140 :: FOR J=LEN(STR$(
Z))TO 1 STEP -1 :: IF LEN(ST
R$(X))=1 THEN M=CC :: GOTO 3
30
320 FOR M=CC TO CC+8 :: CALL
 LOCATE(#J-W,96,M,#J+4-W,116
,M):: NEXT M
330 IF A(J-W)>=B(J-W)THEN 36
0 :: CALL SPRITE(#28,49,16,9
6,M-9)
340 IF F3=1 THEN 360 :: F1=1
 :: A(J-W-1)=A(J-W-1)-1 :: I
F A(J-W-1)<0 THEN A(J-W-1)=9
 :: F2=1 :: A(J-W-2)=A(J-W-2
)-1
350 CALL SPRITE(#22,48+A(J-W
-1),16,80,M-24):: IF F2=1 TH
EN CALL SPRITE(#21,48+A(J-W-
2),16,80,M-40)
360 CALL SPRITE(#27,45,16,11
6,M-12)
370 CALL SPRITE(#20,63,11,R,
M)
380 CALL KEY(3,K,ST):: IF ST
<1 OR K<48 OR K>57 THEN CALL
 PATTERN(#20,32):: CALL PATT
ERN(#20,63):: GOTO 380
390 CALL DELSPRITE(#20,#28):
: CALL SPRITE(#12+J,K,11,R,M
)
400 IF K-48<>C(J)THEN GOSUB
450 :: CALL DELSPRITE(#12+J)
:: F3=1 :: GOTO 330
410 CALL DELSPRITE(#27):: IF
 F1=1 THEN 420 ELSE IF F2=1
THEN 430 ELSE 440
420 F1=0 :: CALL DELSPRITE(#
J-W-1):: FOR P=80 TO 96 :: C
ALL LOCATE(#22,P,M-24):: NEX
T P :: CALL SPRITE(#J-W-1,48
+A(J-W-1),16,96,M-24):: CALL
 DELSPRITE(#22):: GOTO 440
430 F2=0 :: CALL DELSPRITE(#
J-1-W):: FOR P=80 TO 96 :: C
ALL LOCATE(#21,P,M-24):: NEX
T P :: CALL SPRITE(#J-1-W,48
+A(J-1-W),16,96,M-24):: CALL
 DELSPRITE(#21)
440 CC=CC-16 :: NEXT J :: GO
SUB 480 :: F3=0 :: RETURN
450 DATA 123,124,125,123,124
,125,123,120
460 IF A@=0 THEN 470 :: CALL
 SAY(WRONG$(INT(RND*4+1)))
470 RESTORE 450 :: FOR JJ=1
TO 8 :: READ P :: CALL PATTE
RN(#25,P):: XX=2^250 :: NEXT
 JJ :: RETURN
480 DATA 121,122,121,122,121
,122
490 IF A@=0 THEN 500 :: CALL
 SAY(RIGHT$(INT(4*RND+1)))
500 RESTORE 480 :: FOR JJ=1
TO 6 :: READ P :: CALL PATTE
RN(#25,P):: XX=2^250 :: NEXT
 JJ :: RETURN
510 SUB CHAMELEON
520 M$="1800665AC342DB667E18
8100995AC3A5E78142BD24DB6600
81429924007E5AC3A53C241800FF
D85AFF7EFF0099188100660018"
530 RANDOMIZE :: CALL CHAR(1
28,SEG$(M$,INT(43*RND+1)*2-1
,16)):: X=INT(14*RND+3)
540 Y=INT(14*RND+3):: IF Y=X
 THEN 540 :: CALL COLOR(13,X
,Y)
550 CALL HCHAR(1,2,128,30)::
 CALL HCHAR(24,2,128,30):: C
ALL VCHAR(1,31,128,96):: SUB
END
560 SUB TITLE(S,T$)
570 CALL SCREEN(S):: L=LEN(T
$):: CALL MAGNIFY(2)
580 FOR J=1 TO L :: CALL SPR
ITE(#J,ASC(SEG$(T$,J,1)),J+1
-(J+1=S)+(J+1=S+13)+(J>14)*1
3,J*(170/L),10+J*(200/L))::
NEXT J
590 SUBEND
```

When you give your printer instructions, it remembers them until you turn it off. That is why you may find that your letter to Aunt Sally is being printed in double width underlined italics. The solution is found in another gobbledegook paragraph in the Gemini manual - "when (ESC "@") is sent to the printer, the conditions of the printer are initialized."

In plain English, OPEN #1:"PIO" :: PRINT #1:CHR$(27);"@" in your program or CTRL U, FCTN R, CTRL U, SHIFT 2 at the beginning of your TI-Writer text will cancel out any special orders the printer is still remembering and return it to its default conditions.

Here's a bright idea by Scott King in the AVTI UG newsletter. When you load a program in order to modify it, put a reminder of its filename in the first line, such as 1 ! SAVE DSK1.NAME . Then, when you are ready to save it, just list line 1, FCTN 8, use the space bar to erase the 1 !, and Enter.

MEMORY FULL!

Jim Peterson

```
* SOFTWARE DRIVEN CLOCK
* CURTIS ALAN PROVANCE  AUGUST 2Ø, 1986
* NEW HAMSHIRE 99ER'S USER GROUP P.O. BOX 5991 MANCHESTER, NH Ø31Ø8-5991
TEN    EQU  Ø                 'TEN' REFERS TO RØ WITH VALUE OF 1Ø IN UPPER BYTE
SIXTY  EQU  1                 'SIXTY' REFERS TO R1 WHICH HAS A VALUE OF 6Ø
ASCI6Ø EQU  2                 'ASCI6Ø' REFERS TO R2 WHICH HAS VALUES OF '6' AND 'Ø'
VDPWD  EQU  3                 'VDPWD' REFERS TO R3 WHICH CONTAINS >8CØØ
VDPWA  EQU  4                 'VDPWA' REFERS TO R4 WHICH CONTAINS >8CØ2
OFFSET EQU  5                 CHARACTER OFFSET (>9Ø FOR BASIC'S >3Ø OTHERWISE)
HOURS  EQU  6                 HOURS REGISTER
MINUTS EQU  7                 MINUTES REGISTER
SECNDS EQU  8                 SECONDS REGISTER
COUNTR EQU  9                 REGISTER 9 WILL HOLD THE TIME COUNTED DOWN
       AORG >FFE8             FIRST HIGH ADDRESS NOT USED BY XBASIC PROGRAMS
       DEF  CLOCK             WE'LL LINK TO THIS IN BASIC OR XBASIC
ADDRSS DATA START             START ADDRESS TO BE LOADED INTO INTERRUPT HOOK
DIGADD DATA DIGITS            ADDRESS WHERE DIGITS START
CLOCK
       MOV  @ADDRSS,@>83C4 LOAD THE ISR HOOK WITH THE START ADDRESS
       MOV  @DIGADD,@-2       LOAD ADDRESS OF DIGITS INTO HIGH MEMORY
       RT                     ALL SET, DON'T EXECUTE ROUTINE YET
       RORG
START
       LWPI MYWSP             LOAD MY WORKSPACE
       DEC  COUNTR            COUNT DOWN SIXTY INTERRUPTS
       JGT  OUT               NOT YET DOWN TO ZERO, RETURN TO PROGRAM
       MOV  SIXTY,COUNTR      RESTORE TIMER
       INC  SECNDS            INCREMENT SECOND'S "ONE'S" DIGIT
       CB   @R8L,TEN          ARE WE UP TO TEN YET?
       JLT  WRITE             NO, JUMP TO WRITE ROUTINE
       SRA  SECNDS,8          ERASE ALL ONE'S DIGITS
       INC  SECNDS            INCREMENT TEN'S DIGITS
       SLA  SECNDS,8          AND RESTORE TO CORRECT POSITION
       C    SECNDS,ASCI6Ø     ARE WE PAST SIXTY?
       JLT  WRITE             NO, JUMP TO WRITE ROUTINE
       CLR  SECNDS            SIXTY SECONDS ARE UP - RESET COMPLETELY
       INC  MINUTS            INCREMENT MINUTE'S "ONE'S" DIGIT
       CB   @R7L,TEN          ARE WE UP TO TEN YET?
       JLT  WRITE             NO, JUMP TO WRITE ROUTINE
       SRA  MINUTS,8          ERASE ALL ONE'S DIGITS
       INC  MINUTS            INCREMENT TEN'S DIGITS
       SLA  MINUTS,8          AND RESTORE TO CORRECT POSITION
       C    MINUTS,ASCI6Ø     ARE WE PAST SIXTY?
       JLT  WRITE             NO, JUMP TO WRITE ROUTINE
       CLR  MINUTS            SIXTY MINUTES ARE UP - RESET COMPLETELY
       INC  HOURS             INCREMENT HOUR'S "ONE'S" DIGIT
       CI   HOURS,>Ø1Ø3       HAVE WE GONE TO 13-O-CLOCK?
       JLT  OK                NO, OK TO USE THIS HOUR
       LI   HOURS,>FØØ1       RESET TO ONE-O-CLOCK (SPACE CHARACTER IN FRONT)
OK
       CB   @R6L,TEN          ARE WE UP TO TEN YET?
       JLT  WRITE             NO, JUMP TO WRITE ROUTINE
       LI   HOURS,>Ø1ØØ       WE KNOW IT HAS TO BE TEN-O-CLOCK
WRITE
       LI   R1Ø,>164Ø         TWENTY THIRD CHARACTER IN SCREEN, BYTES REVERSED
       MOVB R1Ø,*VDPWA        WRITE BYTE TO VDP ADDRESS REGISTER
       SWPB R1Ø               SWAP BYTES AND KILL TIME
       MOVB R1Ø,*VDPWA       WRITE SECOND BYTE TO REGISTER
       LI   R12,DIGITS       ADDRESS IN WORKSPACE WHERE DIGITS START
       LI   R1Ø,3            THREE PAIRS OF DIGITS TO WRITE
       JMP  WRITE2           SKIP WRITING COLON THE FIRST TIME
WRITE1
       LI   R13,>9AØØ        MOVE ASCII FOR COLON INTO CHARACTER REGISTER
       MOVB R13,*VDPWD       WRITE BYTE TO VDP WRITE DATA REGISTER
WRITE2
       MOV  *R12+,R13        MOVE NEXT SET OF TIME BYTES INTO RØ
       A    OFFSET,R13       ADD OFFSET TO CHARACTERS SO THEY'LL SHOW UP PROPERLY
       MOVB R13,*VDPWD       WRITE BYTE TO VDP WRITE DATA REGISTER
       SWPB R13              SWAP BYTES AND KILL TIME
       MOVB R13,*VDPWD       WRITE BYTE TO VDP WRITE DATA REGISTER
       DEC  R1Ø              COUNT DOWN EACH TIME PAIR
       JGT  WRITE1           LOOP BACK (AND INCLUDE A COLON) FOR NEXT PAIR
OUT
       LWPI >83EØ            LOAD GPL WORKSPACE
       RT                    RETURN TO INTERRUPT ROUTINE
MYWSP  DATA >ØAØØ            RØ = 'TEN' (IN THE HIGH BYTE - USED IN CB OPERATIONS)
       DATA 6Ø               R1 = 'SIXTY'
       DATA >Ø6ØØ            R2 = 'ASCI6Ø'
       DATA >8CØØ            R3 = 'VDPWD'
       DATA >8CØ2            R4 = 'VDPWA'
       DATA >9Ø9Ø            R5 = BASIC OFFSET (>6Ø), PLUS ASCII OFFSET (>3Ø)
DIGITS BYTE >FØ              R6'S HIGH BYTE, HOLDS THE HOUR'S TENS DIGIT OR SPACE
R6L    BYTE Ø                R6'S LOWER BYTE, HOLDS THE HOUR'S UNITS DIGIT
R7H    BYTE Ø                R7'S HIGH BYTE, HOLDS MINUTE'S TENS DIGIT
R7L    BYTE Ø                R7'S LOWER BYTE, HOLDS MINUTE'S UNITS DIGIT
R8H    BYTE Ø                R8'S HIGH BYTE, HOLDS THE SECOND'S TENS DIGIT
R8L    BYTE Ø                R8'S LOWER BYTE, HOLDS SECOND'S UNTIS DIGITS
       BSS  1Ø               THE REMAINING WORKSPACES USED (R1Ø-R13)
LAST   END
```

YOU MAY CREATE THE OBJECT DIRECTLY, BUT IT MUST BE SAVED IN FIXED 8Ø FORMAT AFTER ENTERING THIS WITH THE TI-WRITER EDITOR, USE "PF" THEN "F DSK1.CLOCK"

```
ØØØ92        9FFE8CØØØØCØØ82BC82ØBFFE8B83C4BC82ØBFFEABFFFEBØ45B7F2Ø5F      ØØØ1
AØØØØBØ2EØCØØ76BØ6Ø9B1534BC241BØ588B9Ø2ØCØØ87B111BBØ888BØ5887F2E5F       ØØØ2
AØØ16BØA88B8Ø88B1116BØ4C8BØ587B9Ø2ØCØØ85B1111BØ887BØ587BØA877F2CEF       ØØØ3
AØØ2CB8Ø87B11ØCBØ4C7BØ586BØ286BØ1Ø3B11Ø2BØ2Ø6BFØØ1B9Ø2ØCØØ837F2F2F       ØØØ4
AØØ42B11Ø2BØ2Ø6BØ1ØØBØ2ØAB164ØBD5ØABØ6CABD5ØABØ2ØCCØØ82BØ2ØA7F2DØF       ØØØ5
AØØ58BØØØ3B1ØØ3BØ2ØDB9AØØBD4CDBC37CBA345BD4CDBØ6CDBD4CDBØ6ØA7F22ØF       ØØØ6
AØØ6EB15F6BØ2EØB83EØBØ45BBØAØØBØØ3CBØ6ØØB8CØØB8CØ2B9Ø9ØBFØØØ7F296F       ØØØ7
AØØ84BØØØØBØØØØAØØ887FBA7F                                               ØØØ8
6FFECCLOCK 7FCF3F                                                        ØØØ9
:       99/4 AS                                                          ØØ1Ø
```

```
1ØØ CALL INIT
11Ø CALL LOAD("DSK1.CLOCK")
12Ø CALL LINK("CLOCK")
13Ø CALL PEEK(-2,A,B)
14Ø ADDRESS=A*256+B-1
15Ø INPUT "INPUT TIME: HHMMSS
          ":TIME$
16Ø IF ASC(TIME$)>48 THEN 18Ø
17Ø SPACE=1
18Ø TIME$=SEG$(TIME$&"ØØØØØØØ
Ø",1,8)
19Ø FOR X=1+SPACE TO 8
2ØØ CALL LOAD(ADDRESS+X,VAL(S
EG$(TIME$,X,1)))
21Ø NEXT X
```

This simple program should work in BASIC with either the E/A module or Minimem. It will also work in XBASIC. ADDRESS points to the byte immediately before the DIGITS register. This was done to allow X to start at "1" and to use X as both an address reference and a string reference. You may change the time at any point in your programs. WARNING! Before you run any programs with hidden assembly routines (such as FUNLWRITER) you should kill this routine by: CALL LOAD(-318Ø4,Ø,Ø). You should also disable this routine if you are loading machine code with an absolute origin. If you don't know, try it - the worst that will happen is a lock-up forcing a reboot.

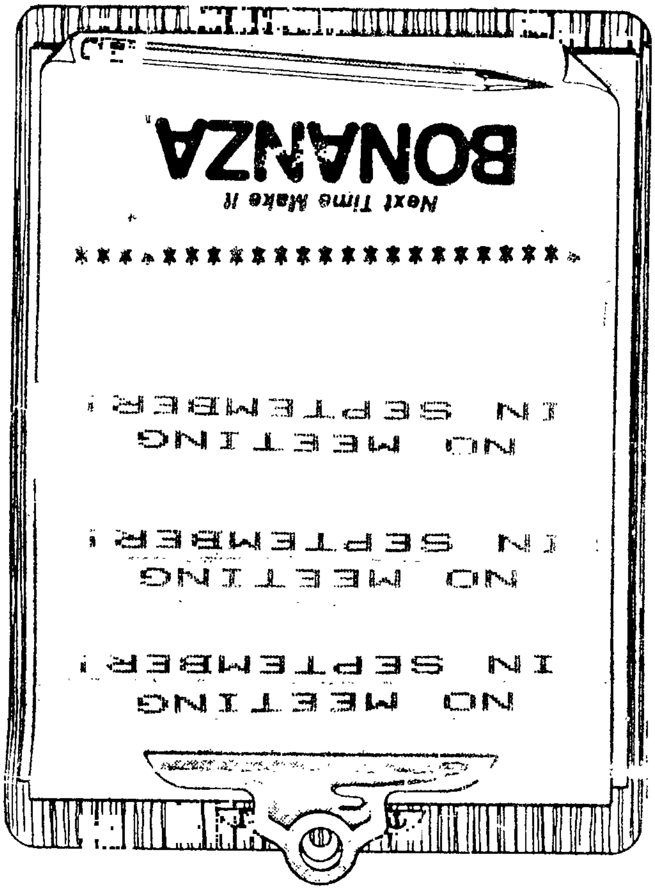
```
100 REM    ******************
110 REM    * TOWER OF HANOI *
120 REM    ******************
130 REM     BY PROF. HOLL AND
            THE HCM STAFF
     HOME COMPUTER MAGAZINE
140 REM        SOLUTION BY:
               CURTIS PROVANCE
     NEW HAMPSHIRE 99ER'S UG

150 REM     VERSION 4.1.1
            TI BASIC
160 CALL SCREEN(16)
170 CALL CLEAR
180 DIM PEG(3),TOP(3),PC(7)
190 FOR COL=1 TO 8
200 CALL COLOR(COL,1,1)
210 NEXT COL
220 PRINT "PRESS 'S'":"TO SE
E A":"SOLUTION.";
230 READ PEG(1),PEG(2),PEG(3
),TOP(1),TOP(2),TOP(3)
240 FOR COL=9 TO 12
250 READ A,B
260 CALL COLOR(COL,A,B)
270 NEXT COL
280 READ PC(1),PC(2),PC(3),P
C(4),PC(5),PC(6),PC(7)
290 FOR X=96 TO 120 STEP 8
300 CALL CHAR(X,"00")
310 CALL CHAR(X+1,"FFFFFFFFF
FFFFFFF")
320 NEXT X
330 CALL HCHAR(8,3,45,13)
340 CALL HCHAR(16,10,45,13)
350 CALL HCHAR(24,17,45,13)
360 FOR X=1 TO 8
370 CALL COLOR(X,2,1)
380 NEXT X
390 CALL HCHAR(8,9,49)
400 CALL HCHAR(16,16,50)
410 CALL HCHAR(24,23,51)
420 FOR X=1 TO 7
430 CALL HCHAR(X,10-X,PC(X),
X*2-1)
440 NEXT X
450 CALL KEY(3,FROM,STATUS)
460 IF STATUS=0 THEN 450
470 IF (SOLUTION=0)*(FROM=83
)THEN 790
480 CALL KEY(3,DUMMY,STATUS)
490 IF STATUS=-1 THEN 480
500 FROM=FROM-48
510 CALL SOUND(100,110,3)
520 IF SOLUTION THEN 870
530 CALL KEY(3,TOO,STATUS)
540 IF STATUS=0 THEN 530
550 CALL KEY(3,DUMMY,STATUS)
560 IF STATUS=-1 THEN 550
570 TOO=TOO-48
580 CALL SOUND(100,262,2)
590 IF (FROM<1)+(FROM>3)+(TO
O>3)+(TOO<1)THEN 450
600 IF (PEG(FROM)=0)+((PEG(T
OO)<>0)*(PEG(FROM)>PEG(TOO))
)THEN 450
610 SIZE=INT(PEG(FROM))
620 TOP(TOO)=TOP(TOO)-1
630 CALL HCHAR(TOP(FROM),((F
ROM-1)*7+10)-SIZE,32,SIZE*2-
1)
640 TOP(FROM)=TOP(FROM)+1
650 CALL HCHAR(TOP(TOO),((TO
O-1)*7+10)-SIZE,PC(SIZE),SIZ
E*2-1)
660 PEG(FROM)=10*(PEG(FROM)-
SIZE)
670 PEG(TOO)=.1*PEG(TOO)+SIZ
E
680 IF (PEG(2)=1.234567)+(PE
G(3)=1.234567)THEN 700
690 GOTO 450
700 A$="YOU WIN!!"
710 FOR X=1 TO 9
720 CALL HCHAR(20,3+X,ASC(SE
G$(A$,X,1)))
730 NEXT X
740 CALL KEY(0,K,S)
750 IF S=0 THEN 740
760 GOTO 780
770 DATA 1.234567,0,0,1,16,2
4,6,4,12,10,8,7,1,14,96,97,1
04,105,112,113,120
780 END
790 SOLUTION=-1
800 CALL CLEAR
810 FOR COL=1 TO 8
820 CALL COLOR(COL,1,1)
830 NEXT COL
840 PRINT TAB(18);"FROM:"::T
AB(18);"TO:":::::::::::::::
::::"PRESS ANY KEY";
850 RESTORE 770
860 GOTO 230
870 MOVE=MOVE+1
880 DUMMY=MOVE
890 FOR SIZE=1 TO 7
900 DUMMY=DUMMY/2
910 IF DUMMY<>INT(DUMMY)THEN
 930
920 NEXT SIZE
930 FROM=-(INT(PEG(1))=SIZE)
-2*(INT(PEG(2))=SIZE)-3*(INT
(FEG(3))=SIZE)
940 TOO=FROM+(-1)^SIZE
950 TOO=TOO-3*(TOO=0)+3*(TOO
=4)
960 CALL HCHAR(1,26,FROM+48)
970 CALL HCHAR(3,24,TOO+48)
980 GOTO 580
```

NEW HAMPSHIRE 99'ERS USER GROUP, INC.
PO BOX 5991
MANCHESTER, NH 03108-5991

NEW ADDRESS

EDMONTON USER'S GROUP
PO BOX 11983
EDMONTON ALBERTA
CANADA T5J-3L1



NO MEETING
IN SEPTEMBER!

NO MEETING
IN SEPTEMBER!

NO MEETING
IN SEPTEMBER!

****

Next Time Make it
BONANZA