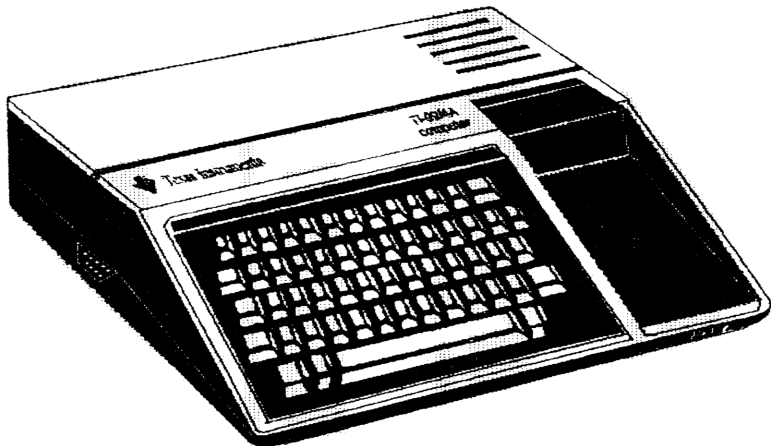


TI*MES

Issue 65

Summer 1999



Committee Members

Position	Name	Address	Telephone
Chairman and Sysop of group BBS	Trevor Stevens	249 Southwell Road East, Rainworth, Notts. NG21 0BN	☎01623 406133 FAX:01623 406134 BBS☐01623 406135
Vice Chairman	Mark Wills	41 Broxtons wood, Westbury, Near Shrewsbury, Shropshire. SY5 9QR	☎01743 885049
General Secretary & TI*MES Editor	Richard Twynning	24 Peel Road, Mansfield, Notts. NG19 6HB	☎Phone: 0467 445658 ☐FAX:0467 449009 ☎/☐ 01623 453934
Interim Newsletter production, web page design, disk li- brary custodian.	Richard Speed	213 Comptons Lane, Horsham, West Sussex. RH13 6BZ	☎01403 242853
Also web Page Design	Ian Pare	10 Sotheby Avenue, Sutton-In-Ashfield, Notts. NG17 5JX	☎01623 552549 ☐FAX: 01623 452729
Treasurer and group membership	A l a n Rutherford	13 The Circuit, Wilmslow, Cheshire. SK9 6DA	☎01625 524642
Module and Cassette Librarian	Francesco L. Lama	48 Mayfair Road, Cowley, Oxford. OX4 3SR	☎01865 435856
Hardware and TI*MES publication	Ross & Christine Bennett	20 Oak Avenue, Romiley, Stockport. SK6 4DN	☎0161 4307298 ☐FAX: 0161 4834516

CONTENTS

1. Front cover!
2. Committe members
3. This page!
4. Module Library - Francesco Lama
6. TI Stuff for Sale
8. More stuff for sale
9. Magazine Deadlines
9. An apology to John Murphy
10. Graphics modes of the V9938 part 3 - by Alan Bray

17. RAMBLES - A re-submission by Stephen Shaw

- 33 Richard T.
- 36 Information on the Stuttgart Tref from Francesco Lama.

- 40 Apology Corner
- 42 The Loss of a TI User Group
- 43 GPL Development package

DISCLAIMER

The views expressed in this magazine are those of the individual authors, and not necessarily those of the editor or the group.

Members on the Internet

Alan Bray:	alan@bric	o.uk
Francesco L. Lama:	mu	rl.ac.uk
Ian Pare:	ian@info:	uk
Alan Rutherford:	a_rut	ot.com
Gary Smith:	gary.sm	.uk
Mark Wills:	markwills2	com
Richard Speed:	richa	om
Stephen Shaw:	stephen.	et.com
Trevor Stevens:	steven	.com
Richard Twyning:	007@r.	co.uk

MODULE LIBRARY

TITLE	PRICE
EDITOR ASSEMBLER Grom Chip	3.00
ADDITION AND SUBTRACTION 1	3.00
ADVENTURE COMPLETE WITH PIRATE TAPE	5.00
ADVENTURE MODULE	3.50
ALPINE	8.00
A-MAZING	3.00
BEGINNING GRAMMAR	3.00
BEYOND PARSEC	18.00
BIGFOOT	3.50
BLACKJACK AND POKER	3.00
BLASTO	5.00
CAR WARS	4.00
CHISHOLM TRAIL	3.50
D-STATION I	18.00
D-STATION II	18.00
DEFENDER	4.50
DISK MANAGER*	2.00
DISK MANAGER 2*	4.50
EDITOR ASSEMBLER + MANUAL AND DISKS	25.00
EXTENDED BASIC II PLUS (MECHATRONIC)	35.00
EXTENDED BASIC INCL. MANUAL	22.50
EXTENDED BASIC MODULE	15.00
HOPPER	3.50
HOUSEHOLD BUDGET MANAGEMENT	3.50
HUNT THE WUMPUS	4.00
INDOOR SOCCER	4.00
INTEGERS	4.50
JAWBREAKER II	4.00
MAGIC MEMORY	40.00
MINI MEMORY + LINE ASSEMBLER + MANUAL	15.00
MINI MEMORY AS ABOVE + MINI WRITER	18.00
MOON MINE	8.00
MOON PATROL	4.00
MULTIPLAN + SOFTWARE + MANUALS	30.00
MUNCH MAN	3.50
MUNCH MOBILE	6.00
PAC MAN	3.50
PARSEC	4.00
PERSONAL RECORD KEEPING	3.50
PERSONAL REPORT GENERATOR	5.50

PHYSICAL FITNESS	4.00
PICNIC PARANOIA	3.50
PROJECTOR	5.00
PROXYPER	20.00
RED BARON (FLIGHT SIMULATOR)	30.00
SHAMUS	3.50
SPACE PATROL	18.00
SPEECH EDITOR	3.50
SPY'S DEMISE	4.50
STATISTICS	4.00
SUPER DEMON ATTACK	4.50
SUPER EXTENDED BASIC + MANUALS	30.00
TAX INVESTMENT RECORD KEEPING	5.00
TERMINAL EMULATOR II	5.00
THE ATTACK	4.00
TIE FIGHTERS	4.00
TIE FIGHTER + ORIGINAL FOLDER WITH MANUAL	15.00
TOMBSTONE CITY	4.00
VIDEO GAMES 1	3.50
YAHOO	3.00
ZERO ZAP	3.50

Magazine scanned 2023 by Stephen Shaw

NOTE This issue was not well printed and many pages showed numerous columns of light or invisible print- see above for example around columns 4 and 5.

Where there is so little image increasing contrast has had no effect.

**TAKE A
LOOK
AT THIS...**



Victor Reiner has the following for sale...

Cartridges

TI-Writer with manual etc., Speech Editor,
Personal Record Keeping
Household Budget Management
Terminal Emulator II, Disk Manager II
TI Extended BASIC with manual
PARSEC, TI Invaders,
Hangman,
Alpiner,
Video Chess,
Indoor Soccer,
Othello.
Video Games 1,
Mini Memory,
The Attack,
Music Maker,
Hunt The Wumpus,
Yahtzee.

Other Hardware

Navarone Widget (3 slot cartridge exander)
PE Box with 4 cards and manuals

TI-99/4A console with Zeno board fitted
Spare 4A console with cables (cassette lead etc.)
EPSON dot matrix printer with 3 spare ribbons and
a spare tabulator motor.
Seikosha 100A printer.
Two Disk Drives
Original TI tape recorder
Original TI joysticks
Cheetah single joystick

Books

Two reference guides for the beginners.
Programming in TI BASIC
Assembly Language for the TI
Making the most of your TI
The best of 99'er magazine
and many more...

Mr. Reiner is selling this lot for £120, but he is willing to separate certain items for the right price.

Victor's address is...
No.1 Amadeus House,
Somerset Road,
East Preston,
West Sussex.
BN16 1BB

His phone number is: 01903 776888, but only call after 6pm please.

FOR SALE

Mr. D. Perryman has the following items for sale...

TI-99/4A console with power supply and a TV modulator,
Speech Synthesizer,
TI Joysticks

Cartridges

Video Chess, Indoor Soccer, TI Invaders,
Beginners BASIC, PARSEC

I think Mr. Perryman is asking £30 for the lot, but postage and packaging are extra.

His address is...

Mr. D. Perryman,
50 Bellingham Crescent,
Plympton,
Devon.
PL7 2QP

His phone number is: 01752 346158, but please only call after 5pm.

Magazine Article Deadline Dates

Spring Issue: 1st March

Summer Issue: 1st June

Autumn Issue: 1st September

Winter Issue: 20th November

Mr. General Secretary / Editor, would like to take this opportunity to apologize to John Murphy for the absence yet again of his articles.

These days I have very little chance to power up my GENEVE system, which is made worse by the state that it is currently in, therefore John has kindly supplied his articles on PC disk.

Unfortunately I didn't have time to meet up with Trevor and collect the disk before he left for Italy, and his son couldn't find the disk when I called around.

Trevor is now back, and I will be able to collect the disk, so expect a John Murphy special in the next magazine!

Richard Twyning.

The Graphics Modes of the V9938

By Alan Bray

Part 3

There are 12 commands on the V9938 that are used for graphics, four of these are 'logical' transfer commands. You can also use logical commands with the 'Line' and 'Pset' drawing functions.

There is a lot of information to look at to understand 'logical' operations and as this would take up far too much magazine space I will have to gradually bring in some of the concepts in future articles. For now, we will look at what can be done and a short 'from the manual' description. Of course, so that you can experiment, I will give you an example of writing the routine to set up the 'LMMC' command, which is a logical move from CPU to VRAM. Also you will need to use a very easy method for defining graphics.

The first thing to do is define your graphics. We really need about five 'frames' for our animation, but, to save space (and typing), I will just define two, the rest will be up to you.

We will also need a delay routine to slow down the speed of animation.

Here is how to set up your graphics (you could draw these on graph paper using coloured pens):-

```
PIC1  BYTE >00,>00,>00,>60,>00,>00,>00,>00
      BYTE >00,>00,>00,>66,>00,>00,>00,>00
      BYTE >00,>00,>06,>00,>00,>00,>00,>00
      BYTE >00,>00,>08,>88,>80,>00,>00,>00
      BYTE >00,>00,>8B,>BB,>B8,>00,>00,>00
      BYTE >00,>08,>BB,>BB,>BB,>B8,>00,>00
      BYTE >00,>8B,>BB,>BB,>BB,>BB,>80,>00
      BYTE >08,>BB,>BB,>BB,>BB,>BB,>B8,>00
      BYTE >8B,>BB,>BB,>BB,>8B,>BB,>BB,>80
      BYTE >08,>BB,>BB,>BC,>CB,>BC,>BB,>80
      BYTE >00,>8B,>BB,>CC,>CB,>CB,>BC,>C8
      BYTE >00,>08,>BC,>CC,>BB,>BB,>BB,>80
      BYTE >00,>00,>4C,>CC,>CC,>CC,>B4,>00
      BYTE >00,>00,>04,>CC,>CC,>C0,>00,>00
      BYTE >00,>00,>00,>48,>84,>00,>00,>00
PIC2  BYTE >00,>00,>00,>A6,>00,>00,>00,>00
      BYTE >00,>00,>06,>B0,>00,>00,>00,>00
      BYTE >00,>00,>0B,>BC,>00,>00,>00,>00
```

```

BYTE >00,>00,>0B,>CC,>00,>00,>00,>00
BYTE >00,>00,>8B,>B0,>C8,>00,>00,>00
BYTE >09,>00,>B0,>0B,>B0,>80,>00,>00
BYTE >00,>0B,>BB,>BB,>80,>00,>00,>00
BYTE >00,>BB,>8B,>B8,>B8,>00,>00,>00
BYTE >00,>CB,>BB,>B9,>00,>00,>00,>00
BYTE >00,>0C,>BC,>CC,>CB,>80,>00,>00
BYTE >00,>00,>CC,>CC,>BC,>C8,>00,>00
BYTE >00,>0C,>CC,>CB,>CC,>CC,>80,>00
BYTE >09,>00,>CC,>BC,>BB,>CB,>C9,>00
BYTE >00,>0C,>CC,>CC,>BC,>8C,>89,>50
BYTE >00,>00,>0C,>CC,>C0,>00,>00,>00
BYTE >00,>00,>00,>CC,>00,>00,>00,>00

```

The above two 'frames' are made up of colour codes to represent a candle flame. You can make up three or four more 'frames' in the same way. Remember that the colour codes are all one less in assembler (colour No 16 (white) will be 'F' which is 15 in hex, therefore >0 is transparent).

To draw more frames, try to use reds, blues and yellows and try to get them to not overlap too much so that you can see what the 'logical' operation does. It would be good practise to also draw a candle beneath your flame. What I do when drawing large graphics is to number the lines in the comment field:-

To draw your graphic 12 pixels wide

```

BYTE >xx,>xx,>xx,>xx,>xx,>xx,>xx,>xx
      >xx,>xx,>xx,>xx
                                     *line 1

```

This way you can draw them any width without moving into the comment field and getting an error message when assembling. Between showing each frame you will need a delay routine. This one will let you adjust the amount of delay between frames so that you can adjust it easily.

```

DEL      DATA DELWS,DEL1
DELWS    BSS      >20
DEL1     MOV      *R14+,R8      We put our delay as DATA >xxxx
after the BLWP @DEL
        MOV      R8,R0
DEL2     DEC      R0
        JNE     DEL2

```

RTWP

And now for the 10 logical operation. In the manual they are set out like:-

Name	Operation	Logical Number
Imp	Dc=sc	>0
And	Dc=sc*dc	>1
Or	Dc=sc+dc	>2
Xor	Dc=sc*dc+sc*dc	>3
Not	Dc=sc	>4
Timp	If sc=0 then dc=dc else dc=sc	>8
Tand	If sc=0 then dc=dc else dc=sc*dc	>9
Tor	If sc=0 then dc=dc else dc=sc+dc	>A
Txor	If sc=0 then dc=dc else dc=sc*dc+sc*dc	>B
Tnot	If sc=0 then dc=dc else dc=sc	>C

*sc= source colour code

*dc= destination colour code

You will have to experiment, but if sc=10 and dc=5 then:-

Imp **dc=sc** so sc=5

And **dc=sc*dc** 1010 b

And 0101 b

= 0000

Or **dc=sc+dc** 1010

Or 0101

= 1111

Read about logical operations in the E/A manual. Imp, And, Or, Xor and Not write the result of each operation to the destination. In the operations that start with 'T' dots that are the same as sc=0 are not the objects operations and stay as dc. So using these operations enables only the colour portions your graphics to be overlapped which makes them ideal for animation (The only way to see what happens really is to try it)! Just remember that if dc=dc then no colour change happens.

In a later article I will have to explain how to use the colour palette so that you can take full advantage of these routines. You can choose colours from a palette of 512, unless of course, you have the SNUG EVPC which lets you choose from 16 million colours!

So now you need the LMMC command and a pointer as to how to use it.

```

LMMC DATA LMMCWS,LMMC1
LMMCWS BSS >20
LMMC1 MOV R13,R9
      MOV *R9+,R1
      MOV *R9+,R2
      MOV *R9+,R3
      MOV *R9+,R4
      MOV *R9+,R5
      MOV *R9+,R6
      LI R0,>2D00

```

get mxd 0 writes to vram, 1 writes to

expansion ram

```
BLWP @VWTR
```

the register >2D can be manipulated

outside of the routine.

```
SWPB R1
LI R0,>0024
MOVB R1,R0
SWPB R0

```

get the destination x

r0 now >24xx where xx is the des-

tination low

```
BLWP @VWTR
SWPB R1
LI R0,>0025
MOVB R1,R0
SWPB R0
BLWP @VWTR

```

get the destination x high

*

```
SWPB R2
LI R0,>0026
MOVB R2,R0
SWPB R0

```

get y

r0 has register 38 (>26) in lsb

r0 has >xx26. where xx is the y byte

now r0 is set up properly for the

bullwhip..>26xx

```
BLWP @VWTR
SWPB R2
LI R0,>0027
MOVB R2,R0
SWPB R0
BLWP @VWTR

```

*

```
SWPB R3
```

```

direction      LI      R0,>0028      the number of dots to transfer in x
               MOVB   R3,R0
               SWPB   R0
               BLWP   @VWTR
               SWPB   R3
               LI     R0,>0029
               MOVB   R3,R0
               SWPB   R0
               BLWP   @VWTR
*
               SWPB   R4
y direction    LI     R0,>002A      number of dots to transfer in the
               MOVB   R4,R0
               SWPB   R0
               BLWP   @VWTR
               SWPB   R4
               LI     R0,>002B
               MOVB   R4,R0
               SWPB   R0
               BLWP   @VWTR

```

Here you could manipulate the argument register for up, down, left, right as in the 'Line' routine

But I will **not** do this to save typing and it is better to do this outside the routine and pass the info back.

```

LI      R0,>2D00      direction is to the right
               BLWP   @VWTR
               LI     R0,>2D00      direction is down
               BLWP   @VWTR
*
               LI     R0,>002C      first byte of your data
               MOVB   *R5+,R0
               SWPB   R0
               BLWP   @VWTR
*
               LI     R0,>2EB8      2E is the command register, B is the
code for the LMMC command

```

```

*          BLWP  @VWTR      and '8' is the logical op (Timp)

          LI      R0,>11AC    r17 is loaded with the the CLR
register (2C) and we set bit 0 to 1
          BLWP  @VWTR      to stop it auto-incrementing the
register. Otherwise it would send
TRY1      MOVB   *R5+,@PORT3  t0 2c, then 2d, then 2e etc.,
          JNE    TRY1

```

Now we have to check status register 2. We check the TR and CE bits of status2 while sending the second and following bytes to the CLR register

```

          LI      R0,>0F02
          BLWP  @VWTR
          CLR    R7
TRWAIT    MOVB   @VDPSTA,R7
          ANDI   R7,>0100
          COC    @ZERO,R7
          JEQ    DONE
TRWAITU   MOVB   @VDPSTA,R7
          ANDI   R7,>8000
          JMP    TRWAIT
DONE      LI      R0,>0F00
          BLWP  @VWTR
          RTWP

```

*
somewhere in you data section you will need:-
ZERO DATA >0000

Before showing how to use the command I will try to explain what the status register part is doing. (If you see the odd reference to 'strange' registers or addresses, it is because I have extracted these routines from various programs that I have written)

This is what status register 2 looks like

TR	VR	HR	BD	1	1	E0	CE
----	----	----	----	---	---	----	----

TR: Transfer ready flag. When the CPU sends commands to the VRAM and other devices, the CPU checks this flag while transferring data.

When this flag is 1 data can be transferred

VR: This flag is set to 1 during vertical scanning

HR: This flag is set to 1 during horizontal scanning

BD: Boundary detect flag. Detects if a boundary colour is found when

using the 'search' command

E0: Display field flag. 0 indicates first field. 1 indicates second field

CE: Command execution flag. This is set to 1 while a command is being executed and reset when the command is finished.

So the status first checks the CE bit (ANDI >0100). If this bit is 1 then it checks the TR bit. If this bit is 1 then transfer can be done. It again checks the CE bit, if it is still 1 then all the data has not been transferred.

If the CE bit is 0 then all the data has been transferred and status register 0 (>0F00) can be called. Status 0 is the normal TI status.

Routine To Use the LMMC Command

```
LI      R7,>10      number of times to run the seque
RUNIT  LI      R0,>100  x co-ordinate
        LI      R1,>4A   y co-ordinate
        LI      R2,>0008  number of dots x dir
        LI      R3,>0010  number of dots y dir
        LI      R4,PIC1
        BLWP @LMMC
        BLWP @DEL
        DATA >3000      time to wait in the delay routine, change to suit
        LI      R4,PIC2
        BLWP @LMMC
        BLWP @DEL
        DATA >3000
Repeat this for however many graphics you have then:-
        DEC      R7
        JNE     RUNIT    loop back to start if R7 is not zero
```

This assumes that you are in G6 mode (R1 sets the x co-ordinate at 256 which is half-way across the screen in G6). You will be able to modify this command very easily to be the HMMC command which is a high-speed command for transferring data from CPU to VRAM, but doesn't allow logical operations.

This should give you lots to play with.

Next time I will tell you how to modify this command into the HMMC command and look at other some other functions of the V9938. Bear in mind that there are more efficient ways to use these commands and that I am using the most simple to follow methods. If you want to see an advanced way to use them, have a look on the Group BBS for a file called MYSS.ARC, this shows a very good way to use commands and makes a great screen saver for the TI.

2023 introduction: This article was printed in TI*MES 65, without its source or history. The item printed in Issue 65 was extracted verbatim from my website, webpage: <http://www.btinternet.com/~shawweb/stephen/artic1.htm>.

The webpage had been taken from a PC disk created by Mike Wright around 1998/9 from the original 90k TI SSSD disk with the original text, in TI Writer DV80 format, from Issues 28 and 30.

The text has been augmented for this digital copy of Issue 65 with two programs from John Seager (which appeared in issue 29), a program by John Murphy, and two programs submitted in 2011.

In 2023 Mike Wright typed in the TurboPas 99 listing and a screen grab of his run of the program has been added. Here we go then..... sjs

R A M B L E S

by Stephen Shaw

Written for TI*MES April 1990. Issue 28

Welcome to another Rambles, which includes bits and pieces for all owners. Even a bit that at first sight you may feel is not appropriate to you is still worth a quick scan...

We are starting this issue with a test which is valid in any language and that includes CONSOLE ONLY OWNERS. I have an answer to this test in Basic

and know it can be done. It is an interesting test for programmers in other languages, and is certainly suitable for c99 and Pascal.

Here we go...

If I write 7^2 , that means SEVEN SQUARED, or put another way, SEVEN TO THE POWER OF TWO, also $7 * 7$.

Equally 7^3 is SEVEN TO THE POWER OF THREE or $7 * 7 * 7$.

Get the idea?

OK. Your test is to write a program, in any language, which will determine the first power of seven which has a result containing six sevens in succession-eg "777777". Clue: The result has MORE than 15 digits!

To make checking easy, your program should print the result:

"SEVEN TO THE POWER OF N IS" where of course N is the first power that gives a result matching our requirements, and also of course giving the actual result.

Programmers in the faster languages may wish to determine if there is a second power of seven with six consecutive sevens in it- and will need to adjust their algorithm slightly to avoid an error message...
Acknowledgement to ACE magazine for this excellent programming test.

Mark Wills is interested in hearing from all our members to find out WHY they still use such a vintage computer.

Me first: [This was written in 1990]

Way way back, when I first thought of buying a computer, I had a very good think about what I wanted one for, what I might need one for, and so on, prioritising everything. There was only one choice then available- the TI99/4, which at that time had not yet been released in a UK version!

I waited for the UK version, and bought a fully expanded system- one of my prime uses was for data, and I saw no future in trying to use a cassette for data!

Many years later, looking back to my original requirements, I still find the TI99/4 (now with an A) to be the best suited to my requirements- A friendly operating system.

Good educational software.

Very rugged construction.

Good choice of languages to learn and use.

Capable of everything I want! in games and utilities.

The competition is actually very thin on the ground in 1990- Atari ST, Amiga and PC seems to be about it. The PC in many ways would be a very retrograde step, and the ST and Amiga require a dedicated monitor for worthwhile results, and are by no means so friendly or easy to program as the TI.

I really dont fancy trying to write a 500k program, or prepare 256 colour graphics! And a slow machine with limited ram makes you learn to program effectively, to do in 16k what other people might take 200k for!

I have a stack of things still to do with this vintage computer, books to read, programs to convert or write, languages to learn and experiment with, many commercial utilities to come to grips with.

About another twenty years worth I guess...

Why move on to fresh pastures when this one still has so much to offer!

SOLUTION TO PUZZLE RE POWER OF 7

```
100 P$="" :: N$="1" :: P=0
110 CARRY=0 :: FOR F=LEN(N$) TO 1 STEP -1
120 V=VAL(SEG$(N$,F,1))*7+CARRY
130 CARRY=INT(V/10) :: V=V-CARRY*10
140 P$=STR$(V)&P$ :: NEXT F
150 IF CARRY<>0 THEN P$=STR$(CARRY)&P$
160 P=P+1 :: N$=P$ :: P$="" :: PRINT P;"
";N$
170 Z=POS(N$,"777777",1)
180 IF Z<1 THEN 110
```

2023 insert: two programs by John Seager from Issue 29 and one previously unpublished in TI*MES from John Murphy:

John Seager submissions 5
5a AND 5b from TI*MES
ISSUE 29- 90q2 -
"Rambles for Times
July 1990"

```
99 REM John Seager
100 !ANSWER TO TEST5/A
110 CALL CLEAR :: NUM$="1176
49" :: DISPLAY AT(1,1):"7 TO
THE POWER OF"
120 FOR POWER=7 TO 300 :: DI
S$=NUM$ :: NUM$="" :: CARRY,
COUNT,SEVENS=0
130 FOR J=LEN(DIS$) TO 1 STE
P -1 :: NEWNUM=(VAL(SEG$(DIS
$,J,1))*7)+CARRY :: CARRY= I
NT(NEWNUM/10)
140 IF NEWNUM-CARRY*10<>7 TH
EN COUNT=0 ELSE COUNT=COUNT+
1
150 SEVENS=MAX(COUNT,SEVENS)
:: NUM$=STR$(NEWNUM-CARRY*1
0)&NUM$
160 NEXT J
170 IF CARRY>0 THEN NUM$=STR
$(CARRY)&NUM$ :: IF CARRY=7
THEN COUNT=COUNT+1
180 SEVENS=MAX(COUNT,SEVENS)
:: DISPLAY AT(1,19):STR$(PO
WER);"="; :: NUM$
190 IF SEVENS<>6 THEN 210 EL
SE DISPLAY AT(24,1):"ANY KEY
TO CONTINUE"
200 CALL KEY(0,K,S) :: IF S=
0 THEN 200 :: DISPLAY AT(24,
1)
210 NEXT POWER
220 REM WHY DOES THIS CRASH
WHEN THE NUMBER OF DIGITS EX
CEEDS 254...
```

```
99 REM FASTER WAY by John Se
ager
100 ! ANSWER TEST5/B
110 CALL CLEAR :: DIM ELEM(2
6) :: ELEM(0)=7 :: POWER,SS=
0 :: DISPLAY AT(1,1):"7 TO T
HE POWER OF"
120 ELM=SS :: SS,CARRY=0 ::
POWER=POWER+1
130 DIS$=STR$(ELEM(ELM)) ::
FOR I=ELM-1 TO 0 STEP -1 ::
DIS$=DIS$&RPT$("0",10-LEN(ST
R$(ELEM(I))))&STR$(ELEM(I))
:: NEXT I
140 DISPLAY AT(1,19):STR$(PO
WER);"="; :: DIS$
150 FOR I=6 TO LEN(DIS$)STEP
6 :: IF SEG$(DIS$,I,1)<>7 T
HEN 190
160 FOR J=I-5 TO I :: IF SEG
$(DIS$,J,6),."777777" THEN 1
80 ELSE DISPLAY AT(24,1):"AN
Y KEY TO CONTINUE"
170 CALL KEY(0,K,S) :: IF S=
0 THEN 170 :: DISPLAY AT(24,
1) :: J=1
180 NEXT J
190 NEXT I
200 ELEM(SS)=ELEM(SS)*7+CARR
Y :: IF ELEM(SS+1)=0 AND ELE
M(SS)&lt;1.E+10 THEN 120
210 CARRY=INT(ELEM(SS)/1.E+1
0) :: ELEM(SS)=ELEM(SS)-CARR
Y*1.E+10
220 SS=SS+1 :: GOTO 200
```

John MURPHY PROGRAM from
DORTIG programmed May 1990:

This program was not
submitted and did not appear
in TI*MES but was found on a
disk after John died in 2009:

```
90 REM SAVE AS 'SS/TEST7'
100 REM ***S.SHAH TEST TI*M
ES ISS:28 P.19. Find the fir
st occurrence of "777777" whe
n 7 is raised to the POWER o
f N??***
110 REM **This Program by J
. MURPHY for DOR-TIG 14 MAY
1990**
120 REM **INPUT NUMBER TO B
E RAISED TO POWER REQUIRED**
130 REM I Have started at N
UMBER^10 to stop running int
o "E"numbers
135 REM MAXIMUM LENGTH OF A
NSWER 254 NUMBERS
140 REM TO SEE NUMBER^ PRES
S SPACE BAR UNTIL PRINTED
150 CALL CLEAR :: NT=0
155 CALL SCREEN(5) :: FOR CO
=1 TO 9 :: CALL COLOR(CO,16,
5) :: NEXT CO
160 INPUT "NUMBER YOU REQUIR
E POWER OF(2 TO 9)":MT
162 IF MT>9 THEN 160
165 DISPLAY AT(12,3):"DO YOU
WISH TO FIND A SERIES OF NU
MBERS (EX:"777777"? ) (Y/N)
" :: ACCEPT AT(15,7)VALIDAT
E(DIGIT,"YNyn")BEEP:FSS$
170 IF FSS$="N" OR FSS$="n" TH
EN 185
180 INPUT "SERIES OF NUMBERS
YOU WISH TO FIND":TE$ :: GO
TO 190
185 NT=1
190 AD=0 :: X$="" :: P=10
210 DISPLAY AT(5,3):" WORK
ING HARD"
215 PRINT "PRESS ANY KEY AND
HOLD DOWN TO SEE PRESENT NU
MBER AND POWER"
220 TOT$=STR$(MT^10)
230 A$=TOT$ :: L=LEN(A$)
240 L1=INT(L/9)*9
250 AA$=SEG$(A$, (L-L1)+1, L1)
260 AAA$=SEG$(A$, 1, L-L1)
270 FOR LE=L1-8 TO 1 STEP-9
280 C$=STR$(VAL(SEG$(AA$, LE,
9))*MT+AD)
```

```
290 IF LEN(C$)>8 THEN 400
300 ON 9-LEN(C$)GOTO 310,320
,330,340,350,360,370,380
310 OA$=""0" :: GOTO 390
320 OA$=""00" :: GOTO 390
330 OA$=""000" :: GOTO 390
340 OA$=""0000" :: GOTO 390
350 OA$=""00000" :: GOTO 390
360 OA$=""000000" :: GOTO 390
370 OA$=""0000000" :: GOTO 39
0
380 OA$=""00000000"
390 C$=OA$&C$
400 IF LEN(C$)=9 THEN 410 EL
SE 420
410 X$=C$&X$ :: AD=0 :: GOTO
440
420 AD=VAL(SEG$(C$,1,1))
430 X$=SEG$(C$,2,9)&X$
440 NEXT LE
450 LEE=LEN(AAA$)
460 IF LEE=0 THEN 530
470 FOR K=LEE TO 1 STEP-1
480 CC$=STR$(VAL(SEG$(AAA$,K
,1))*MT+AD) :: LCC=LEN(CC$)
:: IF LCC=1 OR K=1 THEN 490
ELSE 500
490 X$=CC$&X$ :: AD=0 :: GOT
O 520
500 AD=VAL(SEG$(CC$,1,1))
510 X$=SEG$(CC$,2,LCC-1)&X$
520 NEXT K :: GOTO 560
530 XX$=STR$(AD)
540 X$=XX$&X$
550 AD=0
560 TOT$=X$
570 CALL KEY(0,K,S) :: IF S=
0 THEN 585 ELSE 580
580 PRINT "NUMBER IS":TOT$:
THIS IS ";MT;" TO POWER ";P+
1
585 IF NT>0 THEN 600
590 IF POS(TOT$,TE$,1)=0 THE
N 600 ELSE 610
600 X$="" :: P=P+1 :: GOTO 2
30
610 PRINT "FOUND IT!!!: NUMB
ER IS ";TOT$: "THIS IS";MT;
";^";P+1: "SERIES REQUIRED
IS ";TE$
620 INPUT "ANOTHER NUMBER? (
Y/N)":AN$ :: IF AN$="Y" THEN
150
630 PRINT: "BYE BYE FOR NOW
" :: FOR D=1 TO 200 :: NEXT
D
640 END
```



Did you ever want to make an Extended Basic program unlistable and unalterable but still playable and saveable? This is a bit of fun programming to see how to exploit the way our computer stores its programs!

Original programs from Tidewater Newsletter, Ken Woodcock- I took them from WORDPLAY June 1990:

Your computer stores all the basic program lines in one block, and then in a separate block, a line number table, which identifies where in memory each line number can be found- the line number table is in numeric order, the actual program lines are stored in the order you enter them, which may not be in numeric order! The first byte of the Line Number Table tells the computer how long the line is in bytes- this is only used to list the program or to handle editing functions, when running a zero value byte terminates each program line and the LNT is only used to locate the start of the program line.

To make a program unlistable all you need do is change all the length bytes in the LNT! The following program will set all length bytes to zero! and requires XB and 32k.

First load your program. If it uses lines 1 and 2, resequence it! Then MERGE in (or type in) the following code:

```
1 CALL INIT :: CALL PEEK(-31
952,A,B,C,D):: SL=C*256+D-65
539 :: EL=A*256+B-65536 :: F
OR X=SL TO EL STEP -4 :: CAL
L PEEK(X,E,F,G,H):: ADD=G*25
6+H-65536 :: PRINT E*256+F
```

```
2 CALL LOAD(ADD-1,0):: NEXT
X :: STOP :: !@P-
```

Now RUN the amended program. Now in command mode type in 1 [enter], 2 [enter] to remove the extra lines. Now save using a different file name if to disk or a different tape if to cassette!!!

And you can now RUN the program, no trouble- try it.

BUT... what happens if you try typing EDIT 100 or LIST?

What... you saved the amended program over your original?
Not to worry you can get it back.

We could alter all the length bytes to the maximum possible, which would certainly allow the program to be LISTed again (change the value 0 in line 2 above to 255), but editing could still be problematical, so lets reset the length bytes to what they should be!

Looking for a zero byte is a start, but not the end, as a zero byte may also occur in certain cases in the program line. The simplest thing to do is to look for a zero byte, then look at the line number table to see if the value obtained is an actual start of a program line. So lets do it..

First load the corrupted program- you cannot list it to see if it uses lines 1 to 6, so RESequence it for safety! then MERGE in the following lines:

```
1 CALL INIT :: CALL PEEK(-31
952,A,B,C,D):: SL=C*256+D-65
539 :: EL=A*256+B-65536 :: F
OR X=SL TO EL STEP -4 :: CAL
L PEEK(X,E,F,G,H):: ADD=G*25
6+H-65536 :: PRINT E*256+F
2 I=1 :: CALL PEEK(ADD-1,V
):: IF V THEN 6
3 CALL PEEK(ADD+I,V,W):: I
F V THEN I=I+1 :: GOTO 3
4 FOR Y=SL TO EL STEP -4 ::
CALL PEEK(Y,E,E,E,F):: IF E
*256+F-65536=ADD+I+2 OR W=0
OR ADD+I>-3 THEN CALL L
OAD(ADD-1,I+1):: GOTO 6
5 NEXT Y :: I=I+1 :: GOTO 3
6 NEXT X :: STOP :: !@P-
```

Some of those lines are a little long by the way- when your console honks at you to say it wont take any more characters THANK YOU, you press ENTER, then bring the line back with EDIT N or N (FCI'N X), move the cursor to the end of the line and carry on typing.

Run this amended program- it takes a little longer this time!!!
and when it has finished your program is uncorrupted, lines 1 to 6
can be removed and the program saved, listable and editable.

Too slow? From an anonymous author, a machine code fix!
 This comes from the Sydney News Digest of July 1990 and requires
 assembly WITHOUT the C option:

```
* PROGRAM TO REENTER LINE
* LENGTH VALUES TO A BASIC PROG
* author wishes to be anon
* FROM TISHUG JULY 90
*
* RUNS ONLY FROM XB+32K
* load corrupted program into xb
* CALL INIT
* CALL LOAD("DSKn.FILENAME")
* CALL LINK("UNFIX")
*
```

```
DEF UNFIX
AORG >2500
UNFIX LWPI USRWS
MOV @>8330,R1
MOV @>8332,R2
C R1,R2
JHE FIN
*
INC R2
S R1,R2
SRL R2,1
CI R2,BUFMAX
JGT FIN
*
SRL R2,1
MOV R2,@BUFLEN
INCT R1
LI R4,BUFF
*
LOOP MOVB *R1+,*R4+
MOVB *R1+,*R4
DEC R4
DEC *R4
INCT R4
INCT R1
DEC R2
JNE LOOP
LI R5,>FFE8
AGAIN MOV @BUFLEN,R3
LI R2,ZERO
LI R1,BUFF
DECT R1
*
```

```
*
NEXT INCT R1
C *R1,*R2
JL SKIP
MOV R1,R2
SKIP DEC R3
JNE NEXT
*
MOV *R2,R6
MOV *R1,*R2
S R6,R5
DEC R5
SWPB R5
MOVB R5,*R6
MOV R6,R5
DEC @BUFLEN
JNE AGAIN
*
FIN LWPI >83E0
CLR R0
MOVB R0,@>837C
RT
*
USRWS BSS 32
BUFLEN BSS 2
ZERO DATA >0000
BUFMAX EQU >1800
BUFF BSS BUFMAX
*
END
```

A great deal faster! This machine code is for loading with ExBas ONLY! Why do we need to have the length right for editing? 'Cos if the length byte is 255 bytes, when you delete (or edit) the lines, the computer will delete 255 bytes, and if the line is shorter, you are going to lose other data, probably partial lines, which will be very sad!
 Oh yes... if the corrupted program comes from elsewhere, after making it listable, take a look for any lines that look like our first two lines above, if they are there you need to remove them or they will make the program unlistable once more!

The SEVENS test in issue 28 (page 19) finally provoked some responses, including one from the USA! In TI*MES 29 we printed results for Basic and ExBas, which have not yet been bettered, below you will find coding in c99, Turbo Pascal, and Assembler. Compare these with the Basic versions and try to follow what they are doing. This way you may pick up something of the languages used! If you have access to the languages key them in and see just how fast they are!

First we have c99 source code from John Seager...

c99 was a shareware language based upon c

```
/* S S TEST. 'SEVENS' */
/* VERSION 2 */
/* requires CSUP & PRINTF */
/* to be loaded at run time */
extern printf();
main()
{
/* declare variables */
int power,carry,count,sevens,i,len,temp;
int num[2500];

/* initialise variables */
power=1;
len=0;
num[0]=7;
/* calculate number */
while(1)
{
count=sevens=carry=0; power++;
for(i=0;i<=len;i++)
{
temp=num[i]*7+carry;
num[i]=temp%10;
```



```

carry=temp/10;
    if(num[i]!=7)
        count=0;
    else
        count++;
    if(count>sevens)
        sevens=count;
}
if(carry>0)
{
    len++; num[len]=carry;
    if(num[len]!=7)
        count=0;
    else
        count++;
    if(count>sevens)
sevens=count;
}

locate(1,1);
printf("checking 7 to the power of %d",power);

/* check for 6 sevens & display if found */
if(sevens==6)
{
    locate(1,1);
    printf("7 to the power of %d=",power);
    puts(" ");
    locate(3,1);count=0;

    for(i=len;i>=0;i--)
    {
        printf("%d",num[i]);count++;
        if(count==800)
        {
            locate(24,1);puts("any key for next section of number");
            getchar();locate(24,1);puts(" ");
            locate(3,1);

            for(count=0;count<=800;count++)
                {puts(" ");}
        }
    }
}

```

```

        locate(3,1);count=0;
    }
}

/* continue */
locate(24,1);puts("any key to proceed");
getchar();
locate(1,1);

for(count=0;count<=960;count++)
    {puts(" ");}
}
if(len==2500)
{exit(0);}
}
}

```

<pre> 7 TO THE POWER OF 175= 7801120791220815810240464127911180777777 1881820069326361118396985716038858440268 7177991560647169989331265664440734763224 8554716494939953912586437943 </pre>

Return to top of page

Now for ordinary 9900 machine code, this time from John Stocks...

```

DEF SEVAC

SEVAC LI 1,1  [NO. OF DIGITS]
    LI 7,7
    LI 10,10
    MOV 1,@DIG
    CLR 2  [POWER]
    CLR 3  [CARRY]
A   BL @MULT
    CI 1,6
    JNE A
B   BL @MULT
    BL @TEST
    JMP B

MULT INC 2
    CLR 4

```

```

    LI 0,DIG [DIGIT STORE]
C   INC 4
    MOV *0,5
    MPY 7,5
    A 3,6
    DIV 10,5
    MOV 5,3
    MOV 6,*0+
    C 4,1
    JNE C
    CI 3,0
    JEQ D
    MOV 3,*0
    CLR 3
    INC 1
D   B *11

TEST CLR 4
    LI 0,DIG-2
E   CLR 8
F   AI 4,6
    C 4,1
    JGT RTN
    AI 0,12
    MOV *0,6
    C 6,7
    JNE F
G   DEC 4
    CI 4,0
    JEQ H
    DECT 0
    MOV *0,6
    C 6,7
    JEQ G
H   INC 4
    C 4,1
    JGT RTN
    INCT 0
    MOV *0,6
    CI 6,7

```

```

    JNE E
    INC 8
    CI 8,6
    JNE H
    B @PRINT
RTN B *11

PRINT MOV 1,14
    MOV 2,6
    LI 0,>182
    LI 2,>E
    LI 3,TXT1
    BL @VMBW
    LI 0,>195
    LI 2,2
    LI 3,TXT2
    BL @VMBW
    LI 0,>193
I   CLR 5
    DIV 10,5
    MOV 6,1
    AI 1,>30
    BL @VSBW
    DEC 0
    MOV 5,6
    CI 6,0
    JNE I

    MOV 14,0
    AI 0,>1BF
    MOV 14,4
    INC 4
    LI 5,DIG
J   DEC 0
    DEC 4
    MOV *5+,1
    AI 1,>30
    BL @VSBW
    CI 4,1
    JNE J

```

```

LI 0,>FFFF
LI 1,>B
K DEC 1
L DEC 0
CI 0,0
JNE L
CI 1,0
JNE K
CLR 0
MOVB 0,@>837C
LWPI >83E0
B @>70

```

```

VSBW SWPB 0
MOVB 0,@>8C02
SWPB 0
MOVB 0,@>8C02
SWPB 1
MOVB 1,@>8C04
B *11

```

```

VMBW SWPB 0
MOVB 0,@>8C02
SWPB 0
MOVB 0,@>8C02
CLR 4
M MOV *3+,1
MOVB 1,@>8C04
INC 4
C 15,2
JEQ N
SWPB 1
MOVB 1,@>8C04
INC 4
C 4,2
JNE M

```

```
N B *11
```

```

TXT1 TEXT 'SEVEN TO POWER'
TXT2 TEXT 'IS'

```

```

DIG BSS 400
END

```

That one will only print one answer, but how many powers of seven will produce the desired result? This program will print only the POWERS not the results, up to a power of ten thousand-thats a lot of digits...

```

DEF SEVHI
SEVHI LI 1,1 [NO. OF DIGITS]
LI 7,7
LI 10,10
LI 9,-2
MOV 1,@DIG
CLR 2 [POWER]
CLR 3 [CARRY]
A BL @MULT
CI 1,6
JNE A
B BL @MULT
BL @TEST
CI 2,10000
JNE B
MOV 9,0
AI 0,3
LI 2,10
LI 3,TXT-2
BL @VMBW
DATA >10FF

```

```

MULT INC 2
CLR 4
LI 0,DIG [DIGIT STORE]
C INC 4
MOV *0,5

```

MPY 7,5
A 3,6
DIV 10,5
MOV 5,3
MOV 6,*0+
C 4,1
JNE C
CI 3,0
JEQ D
MOV 3,*0
CLR 3
INC 1
D B *11

TEST CLR 4
LI 0,DIG-2
E CLR 8
F AI 4,6
C 4,1
JGT RTN
AI 0,12
MOV *0,6
C 6,7
JNE F
G DEC 4
CI 4,0
JEQ H
DECT 0
MOV *0,6
C 6,7
JEQ G
H INC 4
C 4,1
JGT RTN
INCT 0
MOV *0,6
CI 6,7
JNE E
INC 8
CI 8,6
JNE H

MOV 11,8
BL @PRINT
MOV 8,11
JMP E
RTN B *11

PRINT MOV 11,13
MOV 1,14
MOV 2,12
AI 9,8
MOV 9,0
MOV 12,6
I CLR 5
DIV 10,5
MOV 6,1
AI 1,>30
BL @VSBW
DEC 0
MOV 5,6
CI 6,0
JNE I
MOV 12,2
MOV 14,1
MOV 13,11
B *11

VSBW SWPB 0
MOVB 0,@>8C02
SWPB 0
MOVB 0,@>8C02
SWPB 1
MOVB 1,@>8C04
B *11

VMBW SWPB 0
MOVB 0,@>8C02
SWPB 0
MOVB 0,@>8C02
CLR 4
M MOV *3+,1
MOVB 1,@>8C04

```
SWPB 1
MOVB 1,@>8C04
INC 4
C 4,2
JNE M
N B *11
```

```
TXT TEXT 'END OF RUN'
```

```
DIG BSS 400
```

```
END
```

Turbo Pascal was a commercial language.

And also from John Stocks is this program for TURBO PASC 99 which is pretty fast for a compiled language...

```
TXT2 TEXT 'IS'
```

```
PROGRAM SEVPAS;
```

```
VAR X,POW,MAXDIG,CARRY : INTEGER;
    DIG : ARRAY[500] OF INTEGER;
```

```
PROCEDURE PRINT;
```

```
VAR Z:INTEGER;
```

```
BEGIN
```

```
    WRITELN("SEVEN TO THE POWER OF",POW," IS :");
```

```
    WRITELN;
```

```
    FOR Z:=MAXDIG DOWNTO 1 DO
```

```
        WRITE(SEG(CIS(DIG[Z]),2,1));
```

```
    WRITELN;
```

```
    WRITELN
```

```
END;
```

```
PROCEDURE SCAN;
```

```
VAR Y,SEV:INTEGER;
```

```

BEGIN
  Y:=X;
  WHILE DIG[Y]=7 DO
    Y:=Y-1;
  SEV:=0;
  Y:=Y+1;
  WHILE DIG[Y]=7 DO
    BEGIN
      SEV:=SEV+1;
      IF SEV>=6 THEN PRINT;
      Y:=Y+1
    END;
  END;
END;

```

```

PROCEDURE TEST;
BEGIN
  X:=6;
  WHILE X<=MAXDIG DO
    BEGIN
      IF DIG[X]=7 THEN SCAN;
      X:=X+6
    END;
  END;
END;

```

```

PROCEDURE MULTIPLY;
VAR X : INTEGER;
BEGIN
  FOR X:=1 TO MAXDIG DO
    BEGIN
      DIG[X]:=DIG[X]*7+CARRY;
      IF DIG[X]>9 THEN
        BEGIN
          CARRY:=DIG[X] DIV 10;
          DIG[X]:=DIG[X]-10*CARRY
        END
      ELSE CARRY =0
    END;
  IF CARRY<>0 THEN
    BEGIN
      DIG[MAXDIG+1]:=CARRY;
      MAXDIG:=MAXDIG+1;
    END;
  END;
END;

```

```

CARRY:=0
END;
POW:=POW+1
END;

```

```

BEGIN
MAXDIG:=1;
DIG[1]:=1;
POW:=0;
CARRY:=0;
REPEAT
MULTIPLY;
UNTIL POW=11;
REPEAT
MULTIPLY;
TEST;
UNTIL POW=500
END.

```

SEVEN TO THE POWER OF 175 IS :

```

78011207912209815910240464127911180777777
1881820069326361118396985716038858440266
7177991560647169999331265664440734763224
8554716494939953912586437943

```

Many thanks to everyone who has submitted code. Can any of this be made faster? Any more languages?

After only a few years here are some additional Basic solutions to the Sevens Problem

```

10 DIM A(256)
30 PRINT "7's Problem"
31 A(1)=7
32 WIN=0
33 POWER=1
41 NUMLEN=1
45 POWER = POWER + 1
46 PRINT "7 ^"; POWER; "IS:" : :
48 CARRY = 0
49 INAROW = 0
50 FOR I=1 TO NUMLEN
60 A(I) = A(I)*7 + CARRY
70 CARRY = INT(A(I) / 10)
80 A(I) = A(I) - CARRY *10
82 IF A(I) <> 7 THEN 89
83 INAROW = INAROW + 1
84 IF INAROW <> 6 THEN 90
85 WIN = 1
86 GOTO 90
89 INAROW=0
90 NEXT I
100 A(I) = CARRY
101 IF CARRY = 0 THEN 109
102 NUMLEN = NUMLEN + 1
109 H=3
110 FOR I=NUMLEN TO 1 STEP -1
120 CALL HCHAR(23,H,48+A(I))
121 H=H+1
122 IF H<32 THEN 130
123 H=2
124 PRINT :
130 NEXT I
131 PRINT : :
140 IF WIN <> 1 THEN 45
150 PRINT "WINNER IS 7 ^";POWER
: From Dragonstomper in Virginia:

```


and from the same source, a faster version:

```
10 DIM A(256)
30 PRINT "7's Problem"
31 A(1)=7
32 WIN=0
33 POWER=1
41 NC=1
45 POWER = POWER +1
46 PRINT "7 ^"; POWER; "IS:" : :
48 CARRY = 0
49 INAROW = 0
50 FOR I=1 TO NC
60 A(I) = A(I)*7 + CARRY
70 CARRY = INT(A(I) / 1000000000)
80 A(I) = A(I) - CARRY *1000000000
90 NEXT I
100 A(I) = CARRY
101 IF CARRY = 0 THEN 103
102 NC = NC +1
103 H=3
104 FOR I=NC TO 1 STEP -1
105 NUM$ = STR$(A(I))
106 L = LEN(NUM$)
107 IF I=NC THEN 119
108 NP=48
115 FOR J=L TO 8
116 GOSUB 200
118 NEXT J
119 FOR J=1 TO L
120 NP = ASC(SEG$(NUM$,J,1))
121 GOSUB 200
122 NEXT J
130 NEXT I
131 PRINT : :
140 IF WIN <> 1 THEN 45
150 PRINT "WINNER IS 7 ^";POWER
160 END
200 CALL HCHAR(23,H,NP)
221 H=H+1
222 IF H<32 THEN 225
223 H=2
224 PRINT :
225 IF NP = 55 THEN 228
226 INAROW=0
227 GOTO 250
228 INAROW = INAROW + 1
229 IF INAROW <> 6 THEN 250
230 WIN=1
250 RETURN
```

Richard Twynning Writes:

Dear Reader,

Well, the AGM was VERY thin on the ground! That meant that there wasn't much of a meeting, and there wasn't much of a discussion, and there were no elections, so all the committee members from last year remain in the same positions.

The sparse turnout to the AGM (which is the worst I've ever seen) I suppose can be attributed to Trevor organizing it a lot earlier than normal, and it's caught many people by surprise! Organizing it on the same day as the Grand National also didn't help it I don't think!

Well I suppose I'd better get the subject of the AGM over and done with and start the minutes of what little meeting we had!

Minutes of the TI-99/4A User Group U.K. 1999 Annual General Meeting

Trevor opened the meeting at 14:18 and welcomed those who had attended and then read some apologies for non-attendance from...

Derek Hayward	Jon James	Francesco L. Lama
Alan Rutherford	Stephen Shaw.	Richard Speed
Trevor Taberner		

Mr. Chairman's report

Trevor said it's been a slow year, but one achievement this year has been to get the Bulletin Board charges reduced by changing from British Telecom to Diamond Cable. This has altered the quarterly line rental charges from £26.50 to £15!!!

The BBS is going very well. Currently there are around 90 users logged onto the BBS and using it on a regular basis. A couple of months ago there was a lot of interest from PSION users due to Mr. General Secretary putting a copy of the PSION emulator on the BBS!

The good news doesn't stop there, because use of the BBS is still growing.

General Secretary.

It's been a fairly quiet year again apart from the 1998 Tref. I've almost managed to get my SCSI card working! Almost, because without having tried it, I thought it was working until Francesco said that there was an extra modification devised by Michael Becker, and I've still got to arrange to get my SCSI card to Francesco for him to modify.

There were no other committee members reports, as no others had attended.

We moved straight onto the group discussion.

We decided to keep the membership fees the same until the next major renewal, and then, based on the number of people who re-subscribe possibly reduce the subscription fee.

We discussed how to keep the group going again, and soon, if not already, the most logical solution could be to join forces with other user groups, such as Sinclair and BBC micro user groups etc.

The success of last years Tref now seems strange with an attendance at the AGM of less than 10 people. Despite the success of the UK Tref, I don't think we've got any hope of repeating the success. As we discussed, I think the only chance of success for another show is to organize it as an "All Formats" show, which would possibly raise a LOT of money for the group.

Additional points of discussed were the clock on the BBS PC. Wildcat Bulletin board stores the year as four digits, so it's OK, but the PC clock chip only stores it as 2, so it's going to screw up Wildcat in a big way on 01-01-2000!

I think the final outcome of this conversation was to let Trevor appropriate whatever funds were necessary to get a Y2K compliant motherboard for it.

Also, thanks go to Berry Harmsen who said he enjoyed every minute of the West Bridgford Tref'98.

Well, no real article again from me this quarter, I will hopefully find time to include an English version of the information about the Stuttgart Tref that Francesco has kindly sent me, here also, for anyone hoping to attend are the phone numbers of Stuttgart Tourist Information....

Phone:0049 7141 278180

FAX: 0049 7141 278137

Dear Richard,

This is what I have been able to put together from my dad's translation of the information sheets I got from Wolfgang Bertsch quite some time ago. I hope it is of some use to inform people about the meeting next September.

Club ERRORFREE

Invites You to

the 14th international meeting of TI99 users

at

Freiberg / Neckar

from the 1st to the 3rd of October 1999

(Continued on page 41)

(Continued from page 40)

Meeting Location: Kleintierzuchtvereinsheim (Cattle Breeders Club),
in Geisingen, Austrasse 3, 71691 Freiberg/Neckar.

Times: Friday from 10 am (opening),
Saturday and Sunday from 8 am.

The club building is fully staffed and managed.

FOR BOOKING AND INFORMATION CONTACTS:

Wolfgang Bertsch
Implerstrasse 8
74321 Bietigheim-Biss.
Tel. +49/7142/221642.

Oliver Arnold Heleneburgweg 61
81731 Meunchen.
Tel. +49/89/7258502.

YOUR PARTICIPATION WOULD BE VERY WELCOME !!!

**ITINERARY FOR THE 1999 INTERNATIONAL T199 USERS
MEETING**

Car Journeys

From the North: take the A81 towards Heilbronn down to the exit for Pleidelsheim. From this point onwards through Pleidelsheim towards Freiberg. At Freiberg turn right and immediately right again into Talstrasse. Follow this road for about 1.5 km (1 ml) and then turn right into Austrasse.

By Rail

From Stuttgart railway station by the S-Bahn line S4 to Freiberg (Neckar). The journey lasts about 20 minutes. From the station take the bus to Geisingen Dorfplatz. From there on foot along Wernerstrasse down to Austrasse (300 m).

Alternatively from Stuttgart station by U-bahn to Ludwigsburg station. From there by bus to Freiberg (Geisingen, Dorfplatz stop). When you contact us to book we will send you local bus and rail maps as well.

By Air

From Stuttgart airport by rail (S-Bahn) line S2 or S3 to the railway station. Then follow instructions as by rail. About one and a half hours should be allowed for the journey from the airport.

(Continued)

ROMOX Cartridges for sale.

These cartridges are wrapped in their original containers and have NEVER yet been opened.

**£3.50 each, including postage and packaging. Contact
Richard Twyning: phone: 01623 627670,
e-mail: 007@r-twyning.demon.co.uk**

These cartridges are ideal for making SUPER-CARTs, or you could actually find someone with an EPROM programmer and load games into them. They will be on sale at the AGM.

(Continued)

Accomodation

Every participant will have to book his/her own accomodation. A list of Freiberg hotels and pensions (B&B), including of prices, follows:

Hotel Schober, Bahnhofstrasse 63-65, Tel. +49/7141/27670

Fax. +49/7141/2767444 Single 55,--bis 128 DM, Double 98,--bis 178 DM.

Gaestehaus Hotel Baumann, Ruitstrasse 67, Tel. +49/7141/73057 Single 68,--bis 78 DM, Double 98,--bis 118 DM, with three beds 148,--bis 168

(Continued)

(Continued

DM, with four beds 178,--bis 198 DM.

Roessle Hotel, Benningerstrasse 11, Tel. +49/7141/27490,
Fax. +49/7141/270739. Single 68,--bis 79 DM, Double 118,--bis 145 DM.

Sammet Gasthof, Gaestezimmer, Bietigheimerstrasse 10, Tel.
+49/7141/72389. Fax. +49/7141/270440. Single 65 DM, Double 85 DM.

Hotel Am Wasen, Wasenstrasse 7, Tel. +49/7141/27470, Fax.
+49/7141/274767. Single 98,--bis 104 DM, Double 140,--bis 170 DM.

Alternative Programme

In case anyone were not interested in computers, or if they wanted to bring the family with them, there would be no problem. The meeting takes place in the outskirts of Freiberg. At no distance from it there is open park-land on the banks of the river Neckar

(Wiesental). Nearby there is also a playground and a riding school. The nearest point for boarding a cruise boat over the Neckar is only one km away. For those interested in history there are many castles within the urban area. Culturally Freiberg has also got a lot to offer. A detailed list of events could be supplied closer to the time.

Moreover Ludwigsburg (rich Baroque town) is only 8 km away, and Marbach, Schiller's birthplace only 10 km. Stuttgart is 25 km away

WELCOME TO APOLOGY CORNER!

From: Richard Speed [mailto:RichardSpeed@rsq1.com]
Sent: 25 March 1999 21:18
To: Richard Twyning (E-mail)
Subject: AGM

<Richard - can you forward this on to Trev; I've lost his email address>

Guys,

I'm not going to be able to attend the AGM again this year due to work and family constraints (I'm doing 5am to midnight at work at the mo, and the missus will not appreciate me disappearing to Derby for an entire day as well!)

A couple of updates for you - since I sent off the last interim newsletter I've heard nothing, nor seen a TI*MES since before the Tref. I did pay my subscription last year, so what's happened? In the meantime I've begun putting the updates on the site (which is currently moving - Tripod have begun being a bit nasty - should still be free at the new location.)

The disk library is stacked up on the shelf on my left. I'm transferring disks at the rate of 10 a day or so (although I missed a few months due to work.) There've been a couple of requests for disks which I've fulfilled. Its worth pointing out the I DO NOT CHARGE FOR COPYING DISKS. Members need

only send disks and postage. If the disks are faulty then I'll supply new ones and if the cost rises above a tenner or so then I might ask the group to provide some blank disks for this purpose. Can't see it happening though.

I'm hoping to get the entire library done before the Stuttgart Tref (which I hope to attend, but funds are looking dicey at the moment.)

So there's my update.

Is anyone there?

R

Date: Sat, 10 Apr 1999 21:05:45 +0100
To: Richard Twyning <007@r-twyning.demon.co.uk>
From: Shaw Family <shawweb@mail.btinternet.com>
Subject: AGM

Work has been particularly hectic these last two weeks (we finally gave up hope of converting our last remaining 300 DOS customers to y2k

Windows software and turned them off (after 2 1/2 years notice and twice extending the deadline).

Today I went to the railtrack site to find out the times of trains to Derby, and it asks you for the date, so I check TI*MES..... AAAAGH.

I really did mean to come down. Many apologies for missing it.

Hope it went well. Sorry.

Stephen Shaw

From: A B Rutherford <a_rutherford@bigfoot.com>
To: Twyning, Richard <007@r-twyning.demon.co.uk>
Cc: <stevenstrevor@hotmail.com>
Date: 08 April 1999 19:09
Subject: Re: AGM

Gents.

I'm afraid I've screwed up my diary this year. I'm not used to having the AGM so early, and have been under the impression that it was in late May. I settled down to read the Spring mag last night and discovered that the meeting is this weekend, and there is no way I can make it. Also I cannot even attach a report of the finances to this email as they still need a bit of preparation.

So, please accept my apologies for being absent both in respect of my person and my report. No doubt you had a pile of bills? So please reply via email - note new address - with details of cheques required.

cheers
Alan

The loss of another valuable group in the TI community...

From: Stephen Shaw <stephen.shaw@btinternet.com>

Subject: Re: C.O.N.N.I.

In-Reply-To: <324922fb.244201d8@aol.com>

Mime-Version: 1.0

Content-Type: text/plain; charset="us-ascii"

Message-Id: <E10WQNb-0000fQ-00@carbon

Sorry to hear of that, CONNI played an important role in the TI community.

>From the activity on the web it seems there are still quite a few TI users out there, just rather widespread.

best wishes

Stephen

A Quick Guide to The TI Filing System by Andy Frueh

In the TI disk filing system, each file's directory information consumes a sector. Sector 0 points to each of these directory sectors, which is why there's a limit to the # of files a disk can hold (12??). If I had a file called "PICTURE" on a disk, it might start at sector 2. Sector 1 is the directory info (name, type, and the numbers of the sectors that actually hold the file). And sector 0 points to sector 1 as having a directory entry. The sector information is needed because a file can be fragmented over many sectors - a file occupies some sectors, then another file occupies a few, then the next few sectors are part of the first file...the thing that hard drive defrag programs fix.

ANYWAY, the end of each directory sector usually contains several free bytes. I believe every directory sector has the same number of free bytes at the end of the sector. DiskU and other "add comments to files" programs place their data in that normally unused area of the directory sectors.

If you use DiskU to add a comment and then edit the first few sectors to see them in ASCII, you'll see the comments at the end of each directory sector. Hope that helps. Maybe someone can clarify this rather confusing description. :-)

Andy

GPL DEVELOPMENT PACKAGE AVAILABLE.

Rich Gilbertson, best known as the author of Rich Extended Basic (RXB), has made available a GPL software development package, complete with source code to two demonstration GPL programs. This package consists of three disks.

The first disk contains the two demo disks in archive format. Disk two contains the complete assembler with documentation and the first half of the GPL programmer's guide. Disk three contains the second half of the GPL programmer's guide.

The handout shows how to compile the demonstration programs and link them into a stand alone Editor Assembler, Option 5, program image. Yes, no GRAM needed! The handout also contains other useful information for the novice GPL programmer.

If you would like a copy, send seven dollars (if shipping to a US address) to:

Dan H. Eicher (Email: Eicher@delphi.com)
4509 Northeastern Avenue
Indianapolis, IN 46239

L.D.O.M. 12.27.1998

GPL Software Development Kit By Dan Eicher

As some of you may remember, Rich Gilbertson was putting together a package for me. Sort of a teach yourself GPL - with everything you needed to get started.

I was going to release this in Chicago after I had demoed it. In fact, that was one of the main reasons I went! I was thankful that Jeff accompanied me up!

Anyway, the reason I did not demo this package was that I did not have a double density controller at the fair to work with. I had spoke with Hal 3 weeks before the fair and he had asked me if I needed anything special to do these demo's. I told him, just a Corcomp, Myarc or Myarc HFDC disk controller.

He said that wouldn't be a problem, when I got up there, nothing!

I was very disappointed to say the least. This should not detract from

Hal since he tried his best to put on a show, despite many set backs.

Truth be told, I should have listen to my intuition (which is right 95% of the time). When it said garb a controller from one of your two Pboxes.

Anyway after I got back, my wife and I packed, then moved, then unpacked, spent alot of time for work and getting ready for the holidays. So, my time for computer fun time has been on the none side.

I did get sometime today and called up Rich to tell him of the debacle.

He was very understanding. I told him my current plan was to distribute the package, give a quick blurb in Micropendium, the list server, Hugger newsletter and BBS and to send a copy to Charlie Good.

I have already had a number of people request the package.

When I spoke to Rich he was playing with a GPL/Assembler compiler I received a few years back and is very excited about the capabilities.

If enough people are interested, in gpl, he is thinking about writing and article in Micropendium about moving/modifying the GPL interpreter to run from AMS memory and then giving people the ability to run dumped modules from AMS memory. You would need special hardware to do this, an AMS card (for additional memory) and a super cart (to handle assembly in modules that executes in place).

He says he has already done this with a couple of modules.

Anyway at the begining of the new year I will make this package available, about 5 ds/sd disks and a 20-30 page handout that I cobbled together.

While things arn't happening as fast as they used to, I know of at least 20-30 people who still VERY ACTIVELY use their TI's and are still developing for them.

Dano