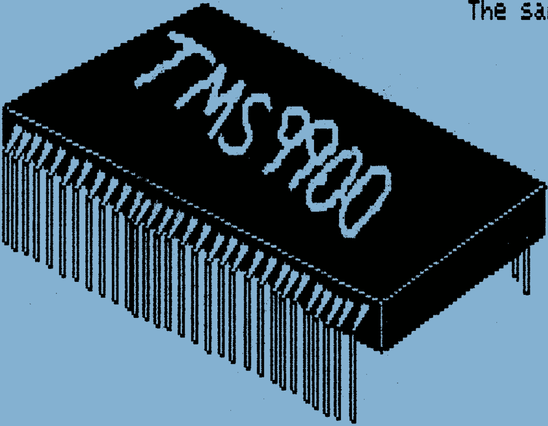
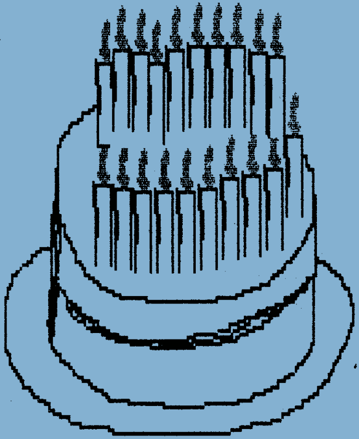
TI*MES

In 2001 a black monolith marked a new dawn in man's intelligence.

The same was true in 1975.





Stackless RISC computing celebrates its 20th birthday this year

Spring 1995

Issue No. 48

TI-99/4A User's Group (U.K.) Head Honcho's!!!

CHAIRMAN:
Trevor Stevens Tel: 0623 793077
249 Southwell Road East, Rainworth, Notts. NG21 0BN

VICE CHAIRMAN, CASSETTE LIBRARIAN, PROGRAMMING:
Mark Wills Tel: 0743 350588
20 Cokayne Green, Harlescott Grange, Shrewsbury, SY1 3QS

GENERAL SECRETARY, AND ALSO PROGRAMMING!
Richard Twyning. Tel: 0623 27670
24 Peel Road, Mansfield, Notts. NG19 6HB

MEMBERSHIP SECRETARY, BACK ISSUES:
Alasdair Bryce. Tel: 0389 65903
51 Dumbuie Ave., Silverton, Dumbarton, Scotland. G82 2JH

TREASURER:
Alan Rutherford. Tel: 0625 524642
13 The Circuit, Wilmslow, Cheshire. SK9 6DA

ACTING TI*MES EDITOR, and the odd bit of hardware!
Gary Smith. Tel: 0636 706767
55 Boundary Road, Newark, Notts. NG24 4AJ

DISK LIBRARIAN, JOURNAL EXCHANGE:
Stephen Shaw.
10 Alstone Road, Stockport, Cheshire. SK4 5AH

PUBLICATIONS:
Mike Curtis. Tel: 0209 219051
21 Treliske Road, Roseland Gardens, Redruth, Cornwall. TR15 1QE

MODULE LIBRARIAN:
Francesco Lama. Tel: 0865 721582
14 Granville Court, Cheney Lane, Oxford. OX3 0HJ

Disclaimer

All views by contributors to this magazine are strictly their own, and do not represent those of the committee. Contrary opinions are very welcome. Errors will be corrected upon request.

Next Copy Date

Instead of sticking to the official copy date for the Summer issue, since the AGM is this year on the 3rd of June, if you wish to submit anything to the newsletter, then bring it along in person to Derby.

Other standard copy dates are normally as follows:

1st of March for the Spring issue
1st of June for the Summer issue
1st of September for the Autumn issue
1st of December for the Winter issue

CONTENTS PAGE

RAMBLES by Stephen Shaw

Hi folks. Here we are, February 24th, and the latest issue of TI*MES comes wafting through the letter box with a next submission date of 1st March. And my father is back in hospital with frequent toings and froings required... aaagh.

Firstly, my absence has been due to sheer lack of time to print things out. My mother died after a long illness, leaving us with my father to care for, and he is a blind insulin dependent diabetic. I thought I would share that with you... I do have lots of text for TI*MES but time is profoundly short in the Shaw household! Then this year we have advance notice of redundancies at work and I have had to apply for my job (results out around early June). The change in corporate culture is sufficiently intense that to stand any chance of maintaining employment TIME has had to be put in....

OK, that's out of the way. Anyone left grumbling can volunteer to print my stuff out - editing it and putting it together take long enough!

The last issue of TI*MES contained a mention of TI EMULATE to enable you to run TI programs on a PC. NB- although I have used a PC at work for some years, I do not at present own one!!!

Why at present? My son George has been brought up with the TI, is a good programmer, keen on computers, and next year will be moving to a secondary school with a computer almost in every clasroom, and five IT rooms absolutely full of them. And as the majority of pupils have access to a PC at home, it follows that George will need access to keep up!

The availability of TI Emulate means that I can continue to program in my favorite computing languages - just take a close look at the programming languages available for the PC! They can be very expensive, very inefficient, and produce scads of code to do simple things!

TI Emulate allows you to utilise TI Basic, TI Extended Basic, and TI Logo- and you can even write and assemble 9900 code to run on your new 586-chipped PC! The speech by the way is digitised (.WAV files) from a TI speech synth, requires a PC Sound Card, and is fairly slow to use- the data occupies over 1.5 meg of hard disk! There are well over 100 modules ported over to Emulate format.

To my mind a PC emulating a TI is still a TI!!! I can still use TI Writer or Funlweb, still write and run XBas programs....

It may be appropriate to build up a disk library of Emulate format files- feel free to drop me a line on the subject (SAE appreciated!). Until I buy a PC I can't copy PC files, but by the time you read this there is a passing chance I will have one, and be programming in TI XB on it! I already have access to over 120 TI modules in the format.

Enough of the PC, and I assure you that not only am I still programming in TI XB but also still using my TI consoles....

In past issues I have brought you some puzzles, and have received one or two letters in response. Here is a nice one...

PUZZLE.

Once more, from the pages of NEW SCIENTIST a simple math puzzle crying out for computer aided solution!

This one comes from the issue dated 14th May 1994 and is Enigma No 770. Yes, there is a similar puzzle to solve EVERY WEEK! If you can't afford to buy the magazine every week, check to see if your local library stock it! They may also have back issues.

Notice that date- that is how far behind I am in bringing this to you, I actually wrote this piece when the issue was current!

New Scientist also frequently has articles and news relevant to computing- the issue of 14th May has articles on flat screen displays; artificial intelligence; Compuserve; virtually real actors; and a lively letters page on an earlier issue on netwroking.

Here is the puzzle exactly as printed:

Long Distance Calls by Josie Hewson.

My telephone has a push-button dialling system in which the digits 1 to 9 are set out as:

```
1  2  3
4  5  6
7  8  9
```

While using those buttons to telephone my friend Chris recently, I notices that her six digit telephone number had some interesting properties.

It was an odd perfect square which used six different digits and no two digits adjacent in her phone number were adjacent (horizontally, vertically, or diagonally) on the telephone.

What is Chris's phone number?

This puzzle offers many many ways to approach it from a programming viewpoint, and lots of ways to try to speed up the response time. Why not have a go?

My attempt in Extended Basic took 65 seconds to find the solution and 142 seconds to exhaust all other possibles and prove there was only the one solution possible.

I shall give my answer to that one later on.

Meanwhile here is a little something to key in - be very careful how you key it in, the spaces are very important!

```
100 REM PROGRAMMED FOR TI/994A
110 REM EXTENDED BASIC
120 REM SEPTEMBER 1994
130 REM
140 CALL CLEAR
150 PRINT "Your task is to map a route around Shaw Mansion so that each room is
visited once & once only"
160 PRINT "You start at > and exit at <"
170 PRINT "Use keys ESDX to move aroundand key Q to start the flooragain."
180 PRINT "There are two floors"
190 PRINT " ":"From Games and Puzzles Mag  May 1994":"subs:":"8 Arbor Court Lond
on N16 0QU"
200 PRINT " ":"space to continue"
210 CALL KEY(5,A,B):: IF B<1 THEN 210
220 CALL CLEAR.
230 REM
240 CALL CHAR(122,"EF818100818181EF")
250 REM z is a room!
260 REM START AT ROOM ONE
270 FLOOR=1
280 CALL CLEAR :: ON FLOOR GOSUB 480,550
290 IF FLOOR=1 THEN R=20 :: C=5 ELSE R=21 :: C=5
300 IF FLOOR=1 THEN ROOMS=105 ELSE ROOMS=71
310 CALL HCHAR(R,C,122):: FOR T=1 TO 7 :: NEXT T
320 CALL KEY(5,A,B)
330 IF A=81 THEN 280
340 CALL HCHAR(R,C,42):: IF B<1 THEN 310
350 ON POS("ESDX",CHR$(A),1)+1 GOTO 310,360,370,380,390
360 CALL GCHAR(R-1,C,P):: GOTO 400
370 CALL GCHAR(R,C-1,P):: GOTO 400
380 CALL GCHAR(R,C+1,P):: GOTO 400
390 CALL GCHAR(R+1,C,P)
400 IF P=32 THEN GOTO 310 ELSE IF P=62 THEN 310
410 IF ROOMS>1 AND P=60 THEN 310 ELSE IF P=60 THEN FLOOR=FLOOR+1 :: IF FLOOR=2 T
HEN 280 ELSE 460
420 CALL HCHAR(R,C,32)
430 IF A=69 THEN R=R-1 ELSE IF A=83 THEN C=C-1 ELSE IF A=68 THEN C=C+1 ELSE R=R+
1
440 ROOMS=ROOMS-1
450 GOTO 310
460 FOR T=1 TO 30 :: CALL CLEAR :: PRINT "WELL DONE!!!"
470 NEXT T :: STOP
480 REM SCREEN ONE
490 PRINT "  zzz zzzzz":"  zzzzzzzzzz":"   zzzzz zzz":"  zzz    zz"
500 PRINT "  zzz    zzz":"  zz     zzz":"   zz zzzzz":"   zzzzzzzz":"   zzzzz zz
"
510 PRINT "  zzz    zzz":"  zzz    zzz":"   zz     zz"
520 PRINT " >zzzzzzzzzz":" <zzzzzzzzzz":"   zzz  zzz"
530 PRINT
540 RETURN
550 REM SCREEN TWO
560 PRINT "  zzz   zzzz":"  zzzzzzzzzz":"  zzzz zzzzz":"  zzzz   zz"
570 PRINT "   zz    zz":"   zz    zz":"   zzz  zzzz"
580 PRINT " <zzzzzzzzzz":" >zzzzzz zzz":"   zzz   zzz"
590 PRINT
600 RETURN
```

I have mentioned Games and Puzzles magazine a long time back- it is back in publication again, and most issues have something worth programming - especially some keen solitaire card games.

++

THE BUSH OF AUSTRALIA

This lengthy music program is included as an illustration of how easy it can be to enter music!

The first task was to deal with an on screen display of the words. Given a 28 column PRINT line, some verses would take up more lines than others, so there had to be a flag to identify the end of a verse. Why not use a lower case "v"?

DATA statements are a handy way to deal with text of this type, where we are just reading and printing, and doing a little bit of something else between verses (eg play some music).

To key in lots of words... use a word processor. TI Writer/Funlweb is fine.
The verses were keyed in with a right margin setting of 29, allowing use of word wrap, and a lower case v put on a line to itself between verses. Then a quick glance through to see if any lines needed adjusting for any reason (artistic purposes here).

Next to add the word DATA at the start of the line, and of course add line numbers:
To add the word DATA, move right margin back to 80, TURN OFF word wrap (CTRL 0) and add a carriage return to every line (RS : 40 40 //CTL U-SHIFT M-CTL U/ : All.

Move the cursor to the top left of the screen:
  S(how) 0 - CTL L
And RS - 0 0 //DATA/
will add the word DATA to the beginning of every line. To add the line numbers we need to Print File so:
PF - C L DSK2.FILENAME
The C will strip out the carriage returns, the L will add line numbers and the PF will make sure there are no tab settings in the last record.

Now we have a program of DATA statements. No we don't- it's a DV80 file.... use TEXTLOADER to OLD the text file in. Now it's in memory, you can RESequence to any set of line numbers you want, and save it out to disk as a program. But do it in merge format, as we still need to add the music! So:
  SAVE DSK2.FILENAME,MERGE

The music to our piece uses exactly 9 notes, which can be coded A to I. There are four different note types, occupying times 1,2,4 and 6. Then there is volume from 0 to 30.
Using TI Writer each note is entered onto a separate line as a three character string, indicating duration, note, and volume. Like this:
2D0
4G0
S
2G3
and so on.

----> CONTINUED

The S?.... When playing two notes of the same pitch one after the other, it helps to insert a LITTLE period of silence, to accentuate the fact that we have two notes! In addition, it helps to insert a slightly longer silence between each line of the song, so we use an L in that position.
Variations in volume are a part of musical interpretation, although there are some rules- try this program first, then make all the notes the same volume and play it again... is it better or worse?

To turn these three characters into a usable CALL SOUND we use Replace String, (WORD WRAP OFF!!!!!) starting with the closing bracket), which we can place in column 4 with: 4 4 //)/

The volumes are as we want them so now to replace the note descriptors with actual frequencies, which we can look up from our Basic manual.
The bottom line of the score is note E, which in OUR music we have marked as a G - with 9 notes in the music there are two note As so we cant use the proper notation in one character!

Lets set all our G's to play a multiple of 659Hz using Replace String- remember to go to the top left of the text first.
/G/,659*M,/ will do the trick. And so on for all the notes -only nine remember!

We now only need to replace the duration number with something we can use and also insert the CALL SOUND, but first let's change those S and L notations to something that compaers with the notes so far:
RS- /S/.25,20000,30)/ and
RS- /L/.50,20000,30)/
and now insert the CALL SOUNDs with:
RS- 0 0 //CALL SOUND(T*/
The two zeroes tell it to insert the new text at the start of the line.
Now add line numbers with PF as before, change to program format and resequence to different line numbers to the DATA statements!

All that is needed now is to write a small section of control code to display the text and play the music. Here is the finished result, entered more quickly and accurately than you could expect, directly from sheet music-to word processor!

```
100 T=90
110 M=1/2
120 CALL CLEAR
130 PRINT "THE BUSH OF AUSTRALIA"
140 GOSUB 1250
150 FOR I=1 TO 600 :: NEXT I
160 PRINT
170 RESTORE
180 READ A$
190 IF A$="v" THEN 220
200 PRINT A$
210 GOTO 180
220 VERSE=VERSE+1
240 ON VERSE GOSUB 260,270,280,290,300,310
245 IF VERSE=6 THEN 2310
250 PRINT : : : :: GOTO 180
260 M=1/3 :: GOSUB 1250 :: RETURN
```

```
270 M=1/4 :: GOSUB 1250 :: RETURN
280 M=1/2 :: GOSUB 1250 :: RETURN
290 M=1/3 :: GOSUB 1250 :: RETURN
300 M=1/2 :: GOSUB 1250 :: RETURN
310 M=1/3 :: GOSUB 1250 :: RETURN
320 DATA As I walked out by those
330 DATA Oxberry banks
340 DATA Where the maids of
350 DATA Australia they play their
360 DATA wild pranks
370 DATA By a shady green bower I
380 DATA sat myself down
390 DATA Where the birds sing so
400 DATA gaily and chanting all
410 DATA round
420 DATA In the forests of happy
430 DATA Australia
440 DATA In the forests of happy
450 DATA Australia
460 DATA Where the maidens are
470 DATA handsome and gay
480 DATA v
490 DATA I had not been long in that
500 DATA beautiful scene
510 DATA Where the fields are
520 DATA delightful  the trees ever
530 DATA green
540 DATA When a lovely young damsel
550 DATA to me did appear
560 DATA From the banks of the river
570 DATA she quickly drew near
580 DATA She's a native of happy
590 DATA Australia
600 DATA She's a native of happy
610 DATA Australia
620 DATA Where the maidens are
630 DATA handsome and gay.
640 DATA v
650 DATA She took off her clothes
660 DATA and before me she stood
670 DATA As naked as Venus just come
680 DATA from the flood
690 DATA She looked me in the face
700 DATA and smiling said she:
710 DATA This is the robe that Dame
720 DATA Nature gave me
730 DATA On the day I was born in
740 DATA Australia
750 DATA On the day I was born in
760 DATA Australia
770 DATA Where the maidens are
780 DATA handsome and gay
790 DATA v
800 DATA She leapt in the water
810 DATA without fear or dread
820 DATA Her beautiful limbs she
830 DATA quickly outspread
840 DATA Her hair hung in ringlets
850 DATA her colour was black
```

```
860 DATA She said 'You can see how I
870 DATA swim on my back
880 DATA In the streams of my native
890 DATA Australia
900 DATA In the streams of my native
910 DATA Australia
920 DATA Where the maidens are
930 DATA handsome and gay'
940 DATA v
950 DATA Being tired of swimming she
960 DATA came to the brink
970 DATA 'Assistance' said she  or
980 DATA surely I'll sink'
990 DATA Like lightning I flew- took
1000 DATA her out by the hand
1010 DATA I put out my foot  she fell
1020 DATA down on the sand
1030 DATA And I entered the bush of
1040 DATA Australia
1050 DATA Then I entered the bush of
1060 DATA Australia
1070 DATA Where the maidens are
1080 DATA handsome and gay.
1090 DATA v
1100 DATA We sported together in the
1110 DATA highest of glee
1120 DATA In the fairest Australia
1130 DATA that ever could be
1140 DATA My head on her beautiful
1150 DATA breast was inclined
1160 DATA Til the sun in the West all
1170 DATA its glories resigned.
1180 DATA Then I left this fair maid
1190 DATA of Australia
1200 DATA Then I left this fair maid
1210 DATA of Australia
1220 DATA Where the maidens are
1230 DATA handsome and gay.
1240 DATA v
1250 REM
1260 REM
1270 CALL SOUND(T*2,587*M,0)
1280 CALL SOUND(T*4,784*M,0)
1290 CALL SOUND(T*.25,22000,30)
1300 CALL SOUND(T*2,784*M,3)
1310 CALL SOUND(T*.25,22000,30)
1320 CALL SOUND(T*2,784*M,3)
1330 CALL SOUND(T*2,740*M,4)
1340 CALL SOUND(T*2,784*M,4)
1350 CALL SOUND(T*2,880*M,0)
1360 CALL SOUND(T*2,659*M,3)
1370 CALL SOUND(T*2,740*M,3)
1380 CALL SOUND(T*4,784*M,3)
1390 CALL SOUND(T*1,22000,30)
1400 CALL SOUND(T*1,988*M,3)
1410 CALL SOUND(T*.25,22000,30)
1420 CALL SOUND(T*1,988*M,4)
1430 CALL SOUND(T*.25,22000,30)
```

```
1440 CALL SOUND(T*2,988*M,0)
1450 CALL SOUND(T*2,880*M,3)
1460 CALL SOUND(T*2,784*M,3)
1470 CALL SOUND(T*.25,22000,30)
1480 CALL SOUND(T*2,784*M,3)
1490 CALL SOUND(T*2,740*M,4)
1500 CALL SOUND(T*2,659*M,4)
1510 CALL SOUND(T*2,587*M,0)
1520 CALL SOUND(T*2,659*M,3)
1530 CALL SOUND(T*2,523*M,3)
1540 CALL SOUND(T*4,494*M,3)
1550 CALL SOUND(T*1,22000,30)
1560 CALL SOUND(T*1,587*M,3)
1570 CALL SOUND(T*.25,22000,30)
1580 CALL SOUND(T*1,587*M,4)
1590 CALL SOUND(T*2,784*M,0)
1600 CALL SOUND(T*.25,22000,30)
1610 CALL SOUND(T*2,784*M,3)
1620 CALL SOUND(T*.25,22000,30)
1630 CALL SOUND(T*2,784*M,3)
1640 CALL SOUND(T*.25,22000,30)
1650 CALL SOUND(T*2,784*M,3)
1660 CALL SOUND(T*2,740*M,4)
1670 CALL SOUND(T*2,784*M,3)
1680 CALL SOUND(T*2,880*M,0)
1690 CALL SOUND(T*2,659*M,3)
1700 CALL SOUND(T*2,740*M,3)
1710 CALL SOUND(T*4,784*M,3)
1720 CALL SOUND(T*1,22000,30)
1730 CALL SOUND(T*1,988*M,3)
1740 CALL SOUND(T*.25,22000,30)
1750 CALL SOUND(T*1,988*M,4)
1760 CALL SOUND(T*.25,22000,30)
1770 CALL SOUND(T*2,988*M,0)
1780 CALL SOUND(T*2,880*M,3)
1790 CALL SOUND(T*2,784*M,3)
1800 CALL SOUND(T*.25,22000,30)
1810 CALL SOUND(T*2,784*M,3)
1820 CALL SOUND(T*2,740*M,3)
1830 CALL SOUND(T*2,659*M,3)
1840 CALL SOUND(T*2,587*M,0)
1850 CALL SOUND(T*2,659*M,3)
1860 CALL SOUND(T*2,523*M,3)
1870 CALL SOUND(T*4,494*M,3)
1880 CALL SOUND(T*1,22000,30)
1890 CALL SOUND(T*1,587*M,3)
1900 CALL SOUND(T*.25,22000,30)
1910 CALL SOUND(T*1,587*M,4)
1920 CALL SOUND(T*.25,22000,30)
1930 CALL SOUND(T*2,587*M,0)
1940 CALL SOUND(T*2,523*M,3)
1950 CALL SOUND(T*2,494*M,3)
1960 CALL SOUND(T*2,M*440,3)
1970 CALL SOUND(T*2,587*M,4)
1980 CALL SOUND(T*2,740*M,4)
1990 CALL SOUND(T*2,880*M,0)
2000 CALL SOUND(T*4,587*M,3)
2010 CALL SOUND(T*4,587*M,3)
```

```
2020 CALL SOUND(T*1,22000,30)
2030 CALL SOUND(T*1,494*M,3)
2040 CALL SOUND(T*.25,22000,30)
2050 CALL SOUND(T*1,494*M,4)
2060 CALL SOUND(T*2,523*M,0)
2070 CALL SOUND(T*2,494*M,3)
2080 CALL SOUND(T*2,523*M,3)
2090 CALL SOUND(T*2,587*M,3)
2100 CALL SOUND(T*2,659*M,4)
2110 CALL SOUND(T*2,740*M,4)
2120 CALL SOUND(T*2,988*M,0)
2130 CALL SOUND(T*.25,22000,30)
2140 CALL SOUND(T*4,988*M,3)
2150 CALL SOUND(T*4,988*M,3)
2160 CALL SOUND(T*1,22000,30)
2170 CALL SOUND(T*1,880*M,3)
2180 CALL SOUND(T*.25,22000,30)
2190 CALL SOUND(T*1,880*M,4)
2200 CALL SOUND(T*2,784*M,0)
2210 CALL SOUND(T*2,740*M,3)
2220 CALL SOUND(T*2,659*M,3)
2230 CALL SOUND(T*2,587*M,3)
2240 CALL SOUND(T*2,988*M,4)
2250 CALL SOUND(T*2,880*M,4)
2260 CALL SOUND(T*6,784*M,0)
2270 CALL SOUND(T*4,784*M,3)
2280 CALL SOUND(T*1,22000,30)
2290 REM
2300 RETURN
2310 M=1/4 :: T=100 :: GOSUB
1250 :: M=1/2 :: T=80 :: GOS
UB 1250 :: END
```

LONG DISTANCE CALLS PUZZLE.....
Anyone try it?

My approach was firstly to consider the limiting rules:

1. The six digits are all different.
   Min 123456 max 987654
2. You noticed of course that zero is not used!
3. The number is a square- so instead of testing some 800,000 numbers we can test less than 700 by just taking the square roots!
4. The number is an ODD square so we can reject any even numbers!
5. The adjacency rules mean that the digit 5 is not used- think about it!
6. Using the adjacency rules we can quickly deduce a minimum number of 134672 and a maximum number of 976438 so we need only test the range of numbers of which these are the squares.

Here is my imperfect program to solve the program- I am sure you can make it work faster!!!

```
100 FOR N=366 TO 990  ! min and max possible
110 N2=N*N
120 IF N2/2=INT(N2/2)THEN 1000 ! must be odd
130 N$=STR$(N2)
132 IF POS(N$,"0",1)>0 THEN 1000
133 IF POS(N$,"5",1)>0 THEN 1000 ! no zero and no 5
140 S=0
150 FOR T=1 TO 6
160 S=S+POS("1234567890",SEG$(N$,T,1),1):: NEXT T
170 IF S<21 THEN 1000
180 IF S>39 THEN 1000  ! QUICK test for 6 diff digits
190 FOR T=1 TO 9 :: R(T)=0 :: NEXT T
200 FOR T=1 TO 6
210 V=POS("123456789",SEG$(N$,T,1),1):: R(V)=R(V)+1 :: NEXT T
220 FOR T=1 TO 9 :: IF R(T)>1 THEN 1000
230 NEXT T        ! more accurate test for 6 dif digits
240 PRINT N;N2;" to 2nd stage check"
250 FOR T=1 TO 9 :: R(T)=0 :: NEXT T
260 FOR T=1 TO 6 :: R(T)=VAL(SEG$(N$,T,1)):: NEXT T
265 FLAG=0
270 FOR T=1 TO 6  ! slow test for adjacency rules
280 ON R(T)GOSUB 2100,2200,2300,2400,2500,2600,2700,2800,2900
288 NEXT T
290 IF FLAG=1 THEN 1000
300 PRINT N;N2;"<---"
301 CALL SOUND(200,200,0)
1000 NEXT N
1001 CALL SOUND(300,600,0):: STOP
2100 IF R(T+1)=2 OR R(T+1)=4 THEN FLAG=1
2110 RETURN
2200 IF R(T+1)=1 OR R(T+1)=3 OR R(T+1)=4 OR R(T+1)=6 THEN FLAG=1
2210 RETURN
2300 IF R(T+1)=2 OR R(T+1)=6 THEN FLAG=1
2310 RETURN
2400 IF R(T+1)=1 OR R(T+1)=2 OR R(T+1)=7 OR R(T+1)=7 THEN FLAG=1
2410 RETURN
2500 STOP
2600 IF R(T+1)=2 OR R(T+1)=3 OR R(T+1)=8 OR R(T+1)=9 THEN FLAG=1
         ! ----> MORE --->
```

```
2610 RETURN
2700 IF R(T+1)=4 OR R(T+1)=8 THEN FLAG=1
2710 RETURN
2800 IF R(T+1)=4 OR R(T+1)=6 OR R(T+1)=7 OR R(T+1)=9 THEN FLAG=1
2810 RETURN
2900 IF R(T+1)=6 OR R(T+1)=8 THEN FLAG=1
2910 RETURN
3000 END
```

Anyone find the answer in less than 65 secs or prove no other possible in less than 142 secs? How did you do it!

==

OK - from the first title to this point (assuming publication in order submitted, a very dodgy assumption indeed!) how much disk space did that text use? Answer = 147 sectors! with just 1400 bytes of free text memory to spare!

Oh yes- please - if you don't like the format of these pages or you don't like the quality of the print, no sarcy comments - unless you are prepared to print it out yourself for next issue. My time IS severely limited. Would you like all this lot like this or just a couple of pages just slightly better looking?

Here is a simple piece for anyone still trying to get to grips with TI BASIC. One of the really nice things about the TI is that it is relatively easy/peasy to make up your own program. My ten year old son produces some quite interesting things...

```
[][][][][][][][][][][][][][][][][][]
[]                                []
[]  P R O G R A M I N G   T I P S []
[]                                []
[]    BY: Paul E. Scheidemantle   []
[]                                []
[]    P  &  A   S O F T W A R E   []
[]                                []
[]         Copyright 1987         []
[]                                []
[]    May be reprinted only in    []
[]         it's entirety.         []
[]                                []
[][][][][][][][][][][][][][][][][][]
```

I'm certain that most of you may have mastered the CALL HCHAR statement with all it's niceities. If so STOP READING NOW... IF NOT READ ON!! lets look at this very versatile statement. With it you can put anything anywhere on the screen that you want; when coupled with other commands.

The normal format of the CALL HCHAR statement is CALL HCHAR(A ,B,C,D). Where A = the row, B = the column, C = the character number, and D = the number of times you wish to put it on the screen.

Well when coupling it with a FOR NEXT LOOP and the ASC, LEN, and SEG$ commands we can write a subroutine that will display at anywhere on the screen without scrolling. A full 32 columns and 24 rows!

Shown below is such a routine. We will learn what makes it work. First the routine starting at line 5000, & then the small portion at lines 1000 -1060 (which is used to demonstrate the routine) .

```
1000 REM TEST DISPLAY AT ROU
TINE
1010 CALL CLEAR
1020 A=1
1030 B=1
1040 MSG$="Hello from: P & A
  Software withthis  test  o
f  the  display  atroutine..
IT FINALLY WORKED**$%#!!"
1050 GOSUB 5010
1060 GOTO 1060
5000 REM DISPLAY AT ROUTINE
5010 FOR LOOP=0 TO LEN(MSG$)
-1
5020 ROW=A+INT((LOOP+B-1)/32
)
5030 IF ROW>24 THEN 5070
5040 COL=B+LOOP-(32*INT((LOO
P+B-1)/32))
5050 CALL HCHAR(ROW,COL,ASC(
SEG$(MSG$,LOOP+1,1)))
5060 NEXT LOOP
5070 RETURN
```

Starting at line 5000. We have a REM statement; all this does for us is identify the routine. In case we want to find it easily later on in a large program.

Now the loop at line 5010 is set so that you can step through the message one character at a time. The string function LEN will tell the computer how many characters are in the string MSG$.

Normally you would want the loop to go from 1 to 90 (or however long the string is) but to keep all the equations working it is necessary to go from 0 to the length of the string. Thus giving us the need to subtract 1.

```
5020 ROW=A+INT((LOOP+B-1)/32)
```

Line 5020 sets up the row that the character from the string is to be printed at. In the base program we sent a value of 1 for A that we wanted to start on row 1. But that's great providing you never print any- thing longer than 32 characters.This line will reset the row to the next line when loop divided by 32 (the number of characters that fit on a line) is reached.

ROW [row character is to be placed at] = A [row that we want to start at] + INT [integer value/those numbers on the left side of the decimal point only] ((LOOP [for next loop to step the row and column and determine which character to print] + B [starting point/column location] -1)/32 [number of characters on a line].

So lets put some numbers in the equation to test it.
A = 1; B = 1; LOOP = 0.
Now figure it out
1+INT((0 +1-1)/32)=1,
Thus ROW = 1. Now lets use a larger value for B (starting in the middle of the screen)
A = 1; B = 16; LOOP = 16.
1+INT((16+16-1)/3 2=1 OR ROW = 1.
However had the loop value been 17 the ROW would have equaled 2; and so on.

Line 5030 is a trap for the row number, should it be greater than the exceptable number of 24.It works like this.... If the value of ROW is greater than 24 then it sends you to line 5070 (return to main program with no action). Otherwise the statement is ignored, and control passed to the next line.

```
5040 COL=B+LOOP-(32*INT((LOOP+B-1 )/32))
```

Line 5040 sets up the column that the character from the string is to be printed in. In the base program we sent a value of 1 for B (meaning that we wanted to start in col 1). This line will reset the col to the next col, and back to 1 when the loop plus the B value -1 divided by 32 (the number of characters that fit on a line) is reached.

COL [column character is to be displayed at] = B [col that we want to start at] + LOOP [next character location] - (32 * INT [integer value /those numbers on the left side of the decimal point only] ((LOOP [for next loop to step the row and column and determine which character to print] + B [starting point column location] - 1)/32 [number of charac- ters on a line])).

So lets put some numbers in the equation to test it.
A = 1; B = 1; LOOP = 0.
Now figure it out
1+0-(32* INT((0+1-1)/32))=1.
Thus COL=1.
Now lets use a larger value for B (starting in the middle of the screen)
A = 1; B = 16; LOOP = 14.
16+14-(32*INT((14+16-1)/32)) = 30 OR COL = 30. However had the loop value been 17 the COL would have equaled 1; and so on.

```
5050 CALL HCHAR(ROW,COL,ASC(SEG$( MSG$,LOOP+1,1)))
```

Line 5050. We see the CALL HCHAR statement. The first two parameters of ROW and COL have been discused as to how we get these values. Now we need to understand how to get the ASC II value required for the third parameter.

First you must understand what ASC and SEG$ do. When a letter is placed in quotes inside parentheses behind the ASC command; will return the ASC II value of the character. For example ASC("A") would return a value of 65. The SEG$ command has the purpose of getting segments of strings out of strings, and works likes this. Its format is SEG$(A$,S, H).

The A$ would be the string that you want to use; the "S" would equal the starting location; and the "H" would be how many characters to get. This could be 1 character or all of the string (depending on the length)

In this case we trying to obtain the ASC II value of a segment of the MSG$ string at the starting point of the LOOP+1 (since we start at 0) and 1 character long to display on the screen.

The values obtained from lines 5010, 5020, and 5040 now give us enough information to use the CALL HCHAR statement. The 1st time thru the loop We would have a state- ment like this. CALL HCHAR(1,1,72) next like this. CALL HCHAR(1,2,69).

Line 5060 is the NEXT statement. This causes the loop to continue un- till the top half of the parameter is reached. Then control is returned to the program and will pass through to the next line.

Line 5070 returns us from the the subroutine to the main program.

Now the first part of the program starting with Line 1010. 1010 clears the screen. 1020 set the value of A to 1 (the starting location that we want for the row).1030 set the value of B to 1 (the starting location that we want for the column).1040 is the message that we want to display on the screen. 1050 branches to the subroutine and then back to 1060 to keep the message on the screen from scrolling. Well I've run out of room so I'll wish you HAPPY COMPUTING!!!!

==========

MULTIPLAN

[from ROM newsletter 8/89 Orange Co Ca USA]

MULTIPLAN DEMONSTRATION [for BEGINNERS] By: Kathryn D Coonley and Erwin H Metz

1. PRELIMINARY START UP:

NOTE: Required: Multiplan module and disk; 32k ram; disk system.

a. Insert Microsoft Multiplan TI Solid State Cartridge into the TI99/4A computer console.

b. Turn on the peripheral equipment (i.e. Disk Drives, Printer, etc.). Turn on the Monitor and Console.

c. Press any key on console to make the master selection list appear and press "2" to select Multiplan from the menu.

d. Select the screen color schemes (12 selections) by depressing the "space bar" on the console until your selection occurs.

e. Insert the Multiplan diskette into Disk Drive #1.

f. Press "enter key" on the console and the Multiplan Overlay will appear on the monitor screen.

2. DATA TYPES:

a. Cells - On a spreadsheet information is entered into "cells". A typical cell is 10 to 30 characters long, so several cells fit on one printed line. A whole column of cells have the same width, and that width can be changed.

b. Once a number has been typed into a cell, it can be added, subtracted, multiplied, etc. to dther numbers in other cells. If the number is too big for the width of cell, the number will not show (rather a # symbol or an exponential value will be displayed) until the column width is increased (see step #8 below).

c. Alpha characters act differently. If more characters are typed than the width of the cell, the number of characters equal to the width of the cell are displayed. All the characters can be displayed in the succeeding cells of the row by using special format commands (see step #9 below).

3. SPREADSHEET COMMAND MENU:

a. The menu at the bottom of the screen, (including submenus obtained after the initial menu selection) permits access to all Multiplan features, to include:

1) set up the initial spreadsheet

2) change entries on the existing spreadsheet.

3) print out the spreadsheet (total or partial)

4) save the spreadsheet to the data diskette in a selected drive

5) delete an existing spreadsheet Page 10

b. The commands can be selected by typing the first letter of the command, or pressing the space bar to move the highlight over the command and pressing enter.

c. Press "Ctrl" + "C" (at the same time) to return to the main menu from any submenu.

4. SET UP FOR 2ND DISK DRIVE (Data Disk):

a. Select "Transfer" by pressing "T".

b. Select "O" for "Options" from the submenu that appears.

c. Press "Ctrl" + "A" (at the same time) to move to the next field.

d. Type "DSK2" next to "SETUP:"

e. Press "enter key" to process the completed command.

5. LOAD FILE:

a. Insert data disk into drive 2.

NOTE: Systems with only one disk drives should switch the data diskette with the Multiplan diskette located in drive 1.

b. Press "T" for "Transfer".

c. Press "L" to select "Load" from the submenu that appears.

d. To show all the files available to load on disk 2: press an "arrow key" ("Function" + "E,S,D,or X") OR press "Ctrl" + "1".

e. Use the "arrow keys" to move the highlight down to the file you want to load.

f. Press "enter" to load the highlighted file.

6. TO SPEED PROCESSING:

a. Press "O" for "Options".

b. Press space bar to move lighlight over "No".

c. Press enter to process the command.

d. Formulas will no longer recalculate after every change (a very slow, frustrating mode to be in). Instead they automatically calculate just before saving. Also you can manually start a recalculation by pressing "FCTN" + 8". Do this before printing to be sure you have the latest numbers.

7. MOVING AROUND THE SPREADSHEET:

a. Special keys ...

1) "Ctrl" + "1" goes to upper left corner.

2) "Function" + "1" goes to the lower right corner.

b. Use arrow keys ...

1) "Function" + "E,S,D or X" will move the highlight around the spreadsheet.

2) "Control"+ "E,S,D or X" moves up, down, or over a screen at a time.

c. The command menu ...

1) Press "G" for "Go to".
2) Press "R" for "Row-column".
3) Type the row number to go to.
4) Press "Ctrl" + "A" to go to next field.
5) Type the number of the column to go to.
6) Press "enter" to process the command.

8. SETTING COLUMN WIDTH:

a. Move the cursor to the column which will be changed.
b. Press "F" for "Format".
c. Press "W" for "Width".
d. Enter the number for the new width of this column.
e. Press "enter" to process the command.

9. FORMAT A CELL (displays Alpha info in
adjacent cells)

a. Move cursor highlight over the cell.
b. Press "F" for "Format".
c. Press "C" for "Cell".
d. Press ":->->->" (arrow over & down).
e. Press "CTRL" + "A" (TWICE).
f. Press "C" for "Continuous".
g. Press "enter" to process the command.

10. EDITING A CELL:

a. Move cursor highlight over the cell.
b. Press "E" for "Edit".
c. Press "Function" + "4" to move the cursor right on the displayed text without deleting characters.

d. Press "Ctrl" + "4" to move the cursor left on the displayed text without deleting characters.
e. Press "Function" + "9" to delete characters.

11. UPDATING AND SAVING:

a. When you change any number in one of the cells included in formulas, all dependent cells are automatically updated to reflect the change.

b. After making desired changes to the spreadsheet, it is important to SAVE the spreadsheet to the data diskette.

1) Press "T" for "Transfer".
2) Press "S" for "Save".
3) Press "enter" to process command.

12. WINDOWING

a. Place cursor in the first cell with data. (To the RIGHT of the verticle title cells & BELOW the horizontal title cells.

b. Press "W" for "Windows".
c. Press "S" for "Split".
d. Press "T" for "Titles".
e. Press "enter" to process the command.
f. To cancel the WINDOWING:
1) Press "W" for "Window".
2) Press "C" for "Close".

- ENJOY, HAVE FUN -

====================================

Peeks and POKES

Compiled by Scott Darling
GEnie ID TIKSOFT

24K OF DATA STORAGE

If you need to work with quite a bit of data or would like to change programs, but save the data after you press CALL QUIT then you can set up the 24K of High-Memory in the PEB as a single data file called "EXPMEM2", you open this file just as you would a disk file with one exception - you must PRECEED th OPEN statement with a CALL LOAD to the location -24574 as follows:

For INT/VAR files - 24
For DIS/VAR files - 16
For INIT/FIX files - 8
For DIS/FIX files - 0

Heres and example:
If you want to open up the Expansion Memory for Display,Variable 80 files this is what you'd do:

```
100 CALL INIT
110 CALL LOAD(-24574,16)
120 OPEN #1:"EXPMEM2",RELATIVE,UPDATE,DISPLAY,VARIABLE 80
```

Then continue on as you normally would.

If you want to store both data and assembly language routines at the same time do this:

```
100 CALL INIT
110 CALL LOAD(-24574,-16)
120 OPEN #1:"EXPMEM2"
130 CALL LOAD ("DSK1.ASSM1")
140 CALL LOAD ("DSK2.ASSM2")
150 CALL LINK ("START")
160 REM CONTINUE REST OF PROGRAM
```

In the above example the 24 K of high-memory was saved for use as a DATA file (DIS/VAR 80 format) then the assembly routines were loaded. The computer wll look for the best place to put the routines and will adjust the pointer accordingly. After the routines are loaded, a LINK statement starts the first rutine and off we go.

If that's not enough for you, you can also use the MINI-MEMORY for 4K more of storage of assembly routines! Now that's 16K of program space, 12K of assembly routine space!

****

NEWS AND REVIEWS From the man with THE BLUES.

By Richard Twyning

Dear TI'ers,

Greetings again to all in the TI community. I'm sorry I didn't have chance to do a proper article in the Winter issue, but I did promise ages ago that I wouldn't miss an issue, and I hope that what I wrote wasn't too disheartening for you.

This quarter I don't have much more time to write my article, so I've tried to start early. I've probably started too early! Today is New Year's Day!!! So, if it's not too late, I'll wish you a happy new year!

The Winter newsletter is running late, but if you think you can do any better, then you had better let Gary know, because he's said he's giving up the position of Editor at the AGM, after all the unfounded complaints he received after previous issues.

Well, as I had time to report at the end of the last issue, we did receive the S&T Bulletin Board from Tim Tesch.

I was going to report that it had problems, but if Gary managed to include a Post Script at the end of the Winter issue, then you will know now that the problem is rectified. Unfortunately it did have problems, and there is a problem with the upload and download section, but it's not VAPOURWARE!!! It's being used on many Bulletin Boards in the States, and included with the board is a list of all registered users of the system, including quite a few who are on 9600 baud, and even 14400 baud.

Gary said it was the Geneve causing problems, but Gary and Trevor visited me recently and we had a demo of the program running on my Geneve, and it was 100% perfect. The graphics are too impressive to describe, and look like a very advanced 4A multicolour mode.

And YES, the group's Multi-Tech MODEM is working PERFECTLY with the program. I have the same MODEM, and dialled in to Trev's with TELCO set to 4800bps. Since we were only across town on the same exchange, the line noise was reasonably low, and the MODEMs decided they would connect at 16800bps!!!

I didn't get one breakup of signal at all, and it's this session when I printed the BBS logs that you can see included with this article.

As I have already said in previous issues, we are planning to make MODEMs available to members (and also other hardware if necessary for "console only" users), so no one should feel left out. Technology has moved on since the 4A was designed, but no machine has represented such an advance in technology as the TI-99/4A and the TMS9900 microprocessor have. As you can see from the front cover of this issue, the Stackless RISC architecture has come of age, and celebrates its 20 th birthday this year.

I didn't do a report on the Sandbach workshop did I???

I think I mentioned something about it in my previous "non-article", and was disappointed at the lack of turnout. I pictured something more resembling the AGM when Trevor first considered organizing another workshop, but there were very few of us there. Mark sent his apologies as he was working in Singapore for a week. The rest of us were familiar faces, and we were hoping for more unfamiliar faces to get involved.

I was hoping that enough of us would be there to allow us to have a discussion on the state of the group, but even people living on the doorstep failed to put in an appearance, and it was to attract these people that prompted us to organize the workshop in their area. ***

You can please some of the people all of the time, but some people can never be pleased!

By saying "the state of the group", I mean the things that were going off at the end of last year. This seemed to arise from certain members thinking that other members had not had anything printed because of deliberate "muting", but if articles arrive in time for an issue, and are acceptably produced, then they will be easy to include, and will be included. Things that are submitted on paper (with correct margins etc.) are more likely to be included, simply because we don't have as much to do to assemble it into the newsletter. To my knowledge, I don 't know of any article actually being rejected, and we don't exclude any small article, just because someone writes big articles! It seemed like people thought that things were left out, just to make room for my article! (@@@)

I like to write lot, when I can, but I don't want that to discourage other members from writing. If an issue appears to be getting too big, then who cares. Let's have a big issue once in a while. We've got plenty of money in the bank to cover the costs.

Supporters of the 99/4A can't be very dedicated if they seem intent on spreading pessimism and ill feeling among members.

This might seem a bit strong, but I'm a bit sick myself of the things that were happening and being said during 1994. I can't remember if I said this before, but it doesn't hurt to say it again. There aren't alot of us left, and we are only here because we know that our machine is best, and stands for something that shouldn't be lost.

By saying it's the best, I know it's not faster than a Pentium, but it IS a more advanced architecture. The idea of having multiple sets of registers anywhere in memory is unique, and gives us alot of

@@@
2022: In the beginning members were invited to submit items any way they wanted including handwriting, which would be transcribed. There was no notification of "required format" at any time! sjs

*** I was probably "one of these people" as folks with cars would consider Sandbach on the doorstep of Stockport. BUT I was now carer for a very poorly father - and a day out was too risky! sjs (2022) power and flexibility.

Our software is a lot better written than PC software, and if they had our memory capacity to work with, then they wouldn't have anything as good as we've got. It's like comparing steel and spider's web. If you have steel that's the same thickness as spider's web, then the spider's web is much stronger.

I said once to a fellow student that it's crazy to write a word-processor that eeds 4 or 6 Megs of RAM before it will run!
He said "there's alot of difference between a word-processor and a text editor!" He's right. One is well known, and one is well written, and I would choose TI-Writer every time. PC word processors have cacky formats that don't store text as text!, and when they print documents they prepare each page as a bit-map and output it as printer graphics!

I will bet any PC owner that if the same document is printed on a 4A and a Pentium PC, both using the same printer, and the PC word processor is using the same size font that the printer on the 4A uses, then the 4A copy will have finished printing first!

You will see the latest version of my Screen Writer program that has been debugged and greatly improved by John Murphy & Co. at DORTIG. John broke his hip last Autumn, but he didn't winge and moan.

My grattitude goes to him and all at DORTIG.
(You will see John's efforts somewhere in my article, or at the end.)

(ADDITIONAL: Thanks to John for cheering me up with a late phone call last night [8-3-95] . "hopefully" John will find a copy of both versions of Screenwriter included with this article, I say hopefully both versions, because, being able to include one of them is dependant on the completion of another little program! Read on for more information!!!)

We should be worrying about finding new members and showing them what a good group they've still got, rather than having internal arguments. Now I hope you are pleased with yourselves, those who may have upset Alasdair Bryce, our valued Membership secretary who has not only decided to leave his post, but is also selling up and leaving the group.

As far as I'm concerned, I don't care what you say to me. You can't upset me by calling me names, but if you bad-mouth the 4A, then that WILL upset me. Then I'll be really mad!!!

If you're thinking about complaining about the big DELAY in receiving the Winter issue, then don't!! I was worried about it too, but Gary has already said he wants to give up the position, and we haven't got alot of spare time to devote to it. I'm in the middle of my final year of my degree, and I'm behind with my final year project. My articles have diminished considerably lately, as I have already explained.

This state of affairs should change in the Autumn issue though.

O.K. that's enough of the moaning. It's time for a bit of 80-Column news, and also 16 million colours! I was greatly cheered up recently by a phone call from Derek Hayward who has received his 80-column card from Germany, and was just trying to sort out the instructions. I think it uses either a 9938 or 9958, but I don't know really how the 16 million colour bit works. I know it uses an extra palette and Digital to Analog Convertor, but that's all I know.

I think Francesco had also either ordered one, or already received one, I can't remember exactly, but I know he said he would bring it to the AGM in Derby on the 3rd of June.

Derek has promised to do an article on the 80-column card which he will hopefully have completed in time for the summer issue. Since the AGM has been set for June 3rd, we will postpone the deadline date for the Summer newsletter until after the AGM. This will give me time (hopefully) to write a report on the AGM, and include the minutes of the meeting.

Right, now what? Well, what's all this I was on about earlier? Well, I was hoping to include two versions of John's modifications to screenwriter. The one I have included I have had to modify slightly to take out the references to the assembly calls.

What I have just tried to work on, was some way of encoding text so that we can do silly things like actually print archived files as text in TI*MES!!!! Yes, I know it sounds silly, but I have had the idea from a utility that is available for the PC and AMIGA etc. This is called UUENCODING which takes a byte from the original file (which is a single character) and encodes it in printable text. Since a byte can be from 0 to 255, the character might be unprintable, and so would be no good for inclusion in TI*MES.

However, if we divide the character into two, and represent it as two nybbles (4-bits, or half bytes) then we only need sixteen different values for each half, and these could be easily represented in TI*MES as printable characters.

You would then just need to type the UUENCODED file into the Funnelweb text editor for example, and then run it through the decoder, which would produce the original archived file.

This could then be unarchived to reveal the original program or programs, which could include embedded assembly language.

I have spend all day today (10th March) to try and get it working, but with no success! I have created an encoded file, which you can see included after the listing of my encoder and decoder as they currently stand.

It sounds like something that DORTIG could make themselves busy with!!!! Then we can have archived files included in future issues of TI*MES.

I must cut my article short now. I'm completely out of TI*ME!!! See you all in Derby on the 3rd of June.

```
50 ! 'SCREENWRITER' by Richa
rd Twning from TI*MES ISS. 4
4 SPRING 1994.
55 ! MODIFIED by J.Murphy so
 that if you put in wrong
parameters (I DO!!!) you do
not lose your drawing, OR
LOAD instead of SAVE ect:.
70 CALL CLEAR :: CALL SCREEN
(13)
75 GOSUB 905
99 CALL KEY(0,K,S):: IF S=0
THEN 99
100 DIM SCREEN(24,32),COLORS
(14,2),H$(15):: GOSUB 800 ::
 CALL CLEAR :: CH=65 :: SC=8
 :: CALL SPRITE(#1,143,7,96,
128,#2,65,16,96,128)
110 DISPLAY AT(15,1):" L-LOA
D PREVIOUS SCREEN      S-SAV
E CURRENT SCREEN       9-GEN
ERATE 9900 ROUTINE"
120 DISPLAY AT(19,1):" B-CAL
L SCREEN(A)            C-CAL
L  COLOR(A,B,C)        D-CAL
L CHAR(A,A$)           N-NEW
 CHARACTER             E-EXI
T"
122 DISPLAY AT(18,1):" T-INP
UT TEXT"
125 DISPLAY AT(24,1):" Z-TO
CANCEL LOAD/SAVE FILE"
130 CALL JOYST(1,X,Y):: CALL
 MOTION(#1,-Y,X):: GOSUB 170
 :: CALL KEY(1,K,S):: IF K=1
8 THEN GOSUB 180 ELSE CALL K
EY(3,K,S):: IF S=0 THEN 130
140 IF K=76 THEN GOSUB 210 E
LSE IF K=83 THEN GOSUB 250 E
LSE IF K=57 THEN GOSUB 310 E
LSE IF K=66 THEN GOSUB 500
150 IF K=67 THEN GOSUB 540 E
LSE IF K=68 THEN GOSUB 600 E
LSE IF K=78 THEN GOSUB 640 E
LSE IF K=69 THEN GOTO 680
155 IF K=84 THEN GOSUB 840
160 GOTO 130
170 CALL POSITION(#1,R,C)::
CALL LOCATE(#2,MIN(MAX(1,8*I
NT(R/8)+1),192),MIN(MAX(1,8*
INT(C/8)+1),256)):: RETURN
180 CALL POSITION(#1,R,C)::
R=INT(R/8)+1 :: C=INT(C/8)+1
 :: IF R<1 OR R>24 OR C<1 OR
 C>32 THEN CALL SAY("OFF+OF+
SCREEN"):: RETURN
190 CALL HCHAR(R,C,CH):: SCR
EEN(R,C)=CH :: RETURN
200 ! TI*MES ISS:44 SPRING 9
4 PRGROM BY R Twyning copied
 by J.M. for DORTIG. SAVED A
S SCR/WRTMOD . ERROR TRAPS P
UT IN BY J.M.
210 ! L-Load Screen
```

```
220 INPUT "LOAD DEVICE.FILEN
AME:":F$ :: IF F$="Z" THEN C
ALL CLEAR :: GOSUB 730 :: GO
TO 130
225 OPEN #1:F$,DISPLAY ,VARI
ABLE 20 :: INPUT #1:COLORS(0
,1):: INPUT #1:COLORS(0,2)
226 FOR D=1 TO 13 STEP 2 ::
INPUT #1:COLORS(D,1):: INPUT
 #1:COLORS(D,2):: INPUT #1:C
OLORS(D+1,1):: INPUT #1:COLO
RS(D+1,2)
230 NEXT D :: INPUT #1:SC ::
 FOR R=1 TO 24 :: FOR C=1 TO
 31 STEP 2 :: INPUT #1:A ::
IF A=0 THEN SCREEN(R,C)=32 E
LSE SCREEN(R,C)=A
235 INPUT #1:A :: IF A=0 THE
N SCREEN(R,C+1)=32 ELSE SCRE
EN(R,C+1)=A
240 NEXT C :: NEXT R :: FOR
D=32 TO 143 STEP 2 :: LINPUT
 #1:A$ :: CALL CHAR(D,A$)
245 LINPUT #1:A$ :: CALL CHA
R(D+1,A$):: NEXT D :: CLOSE
#1 :: GOSUB 730 :: RETURN
250 ! S-SAVE SCREEN
260 INPUT "SAVE DEVICE.FILEN
AME:":F$ :: IF F$="Z" THEN C
ALL CLEAR :: GOSUB 730 :: GO
TO 130
265 OPEN #1:F$,DISPLAY ,VARI
ABLE 20
270 IF COLORS(0,1)=0 THEN PR
INT #1:2 :: PRINT #1:1 ELSE
PRINT #1:COLORS(0,1):: PRINT
 #1:COLORS(0,2)
274 FOR D=1 TO 13 STEP 2
275 IF COLORS(D,1)=0 THEN PR
INT #1:2 :: PRINT #1:1 ELSE
PRINT #1:COLORS(D,1):: PRINT
 #1:COLORS(D,2)
276 IF COLORS(D+1,1)=0 THEN
PRINT #1:2 :: PRINT #1:1 ELS
E PRINT #1:COLORS(D+1,1):: P
RINT #1:COLORS(D+1,2)
280 NEXT D :: PRINT #1:SC ::
 FOR R=1 TO 24 :: FOR C=1 TO
 31 STEP 2 :: IF SCREEN(R,C)
=0 THEN PRINT #1:32 ELSE PRI
NT #1:SCREEN(R,C)
281 IF SCREEN(R,C+1)=0 THEN
PRINT #1:32 ELSE PRINT #1:SC
REEN(R,C+1)
290 NEXT C :: NEXT R :: FOR
D=32 TO 143 STEP 2 :: CALL C
HARPAT(D,A$):: PRINT #1:A$
300 CALL CHARPAT(D+1,A$):: P
RINT #1:A$ :: NEXT D :: CLOS
E #1 :: GOSUB 730 :: RETURN
310 !9-GENERAATE 9900 ROUTIN
E
320 PRINT "GENERATE SOURCE C
ODE TO     PRINTER, DISK, OR
 BOTH?":"":"ENTER:" :: INPUT
 "(P)RINTER D(ISK) B(OTH)":C
$ :: IF C$="P" THEN S=2 :: E
=2
330 IF C$="D" THEN S=1 :: E=
1 ELSE IF C$="B" THEN S=1 ::
 E=2
340 IF C$="D" OR C$="B" THEN
 PRINT "":"" :: INPUT "PLEAS
E ENTER DEVICE.FILENAMEFOR S
OURCE CODE:             ":D$
350 IF C$="P" THEN OPEN #2:"
PIO" ELSE IF C$="D" THEN OPE
N #1:D$ ELSE IF C$="B" THEN
OPEN #1:D$ :: OPEN #2:"PIO"
360 F$="" :: FOR D=LEN(D$)TO
 1 STEP -1 :: K$=SEG$(D$,D,1
):: IF K$="." THEN F$=F$&RPT
$(" ",7-LEN(F$)):: GOTO 380
ELSE F$=K$&F$
370 NEXT D
380 FOR COPY=E TO S STEP -1
:: PRINT #COPY:"      DEF  "
&F$:"VMBW   EQU  >2024":"VWT
R   EQU  >2030"
390 PRINT #COPY:F$&"LI   R0,
>070"&H$(SC-1):"        BLWP
@VWTR":"       LI   R0,2063"
:"       LI   R1,CDAT":"
   LI   R2,15"
400 PRINT #COPY:"        BLWP
 @VMBW":"       CLR  R0":"
     LI   R1,SDAT":"        L
I   R2,770":"        BLWP @VM
BW":"       LI   R0,1024"
410 PRINT #COPY:"         LI
 R0,1024":"       LI   R1,PD
AT":"       LI   R2,872":"
     BLWP @VMBW":"         RT"
420 PRINT #COPY:"SDAT   BYTE
 ";SCREEN(1,1)+96 :: FOR C=2
 TO 32 :: PRINT #COPY:"
  BYTE ";SCREEN(1,C)+96 :: N
EXT C
430 FOR R=2 TO 24 :: FOR C=1
 TO 32 :: PRINT #COPY:"
  BYTE ";SCREEN(R,C)+96 :: N
EXT C :: NEXT R
440 PRINT #COPY:"CDAT   BYTE
 >"&H$(COLORS(0,1)-1)&H$(COL
ORS(0,2)-1):: FOR D=1 TO 14
:: PRINT #COPY:"        BYTE
>"&H$(COLORS(D,1)-1)&H$(COLO
RS(D,2)-1):: NEXT D
450 CALL CHARPAT(32,A$):: PR
INT #COPY:" ":"PDAT   DATA >
"&SEG$(A$,1,4)&",>"&SEG$(A$,
5,4)&",>"&SEG$(A$,9,4)&",>"&
SEG$(A$,13,4)
460 FOR D=33 TO 143 :: CALL
CHARPAT(D,A$)
470 PRINT #COPY:"        DATA
 >"&SEG$(A$,1,4)&",>"&SEG$(A
$,5,4)&",>"&SEG$(A$,9,4)&",>
"&SEG$(A$,13,4):: NEXT D ::
PRINT #COPY:"        END"
```

```
480 NEXT COPY :: GOSUB 730 :
: IF C$="B" THEN CLOSE #1 ::
 CLOSE #2 ELSE IF C$="P" THE
N CLOSE #2 ELSE IF C$="D" TH
EN CLOSE #1
490 RETURN
500 !S-Screen color
510 DISPLAY AT(1,1):"SCREEN
COLOR:" :: ACCEPT AT(1,14)VA
LIDATE(DIGIT):SC
520 IF SC<1 OR SC>16 THEN 51
0
530 CALL SCREEN(SC):: GOSUB
770 :: RETURN
540 ! C-CALL COLOR(A,B,C)
550 DISPLAY AT(1,1):"CALL CO
LOR(" :: ACCEPT AT(1,12)VALI
DATE(DIGIT):A :: DISPLAY AT(
1,14):"," :: ACCEPT AT(1,15)
VALIDATE(DIGIT):B :: DISPLAY
 AT(1,17):"," :: ACCEPT AT(1
,18)VALIDATE(DIGIT):C
560 DISPLAY AT(1,20):")"
570 IF S>14 THEN 540 ELSE IF
 B<1 OR B>16 THEN 540
580 IF C<1 OR C>16 THEN 540
590 CALL COLOR(A,B,C):: COLO
RS(A,1)=B :: COLORS(A,2)=C :
: GOSUB 770 :: RETURN
600 ! D--CALL CHAR(A,A$)
610 DISPLAY AT(1,1):"CHAR("
:: ACCEPT AT(1,6)VALIDATE(DI
GIT):A :: DISPLAY AT(1,9):",
"&CHR$(34):: ACCEPT AT(1,11)
VALIDATE(DIGIT,"ABCDEF"):A$
:: DISPLAY AT(1,20):CHR$(34)
&")"
620 IF A<31 OR A>143 THEN 61
0
630 CALL CHAR(A,A$):: GOSUB
770 :: RETURN
640 ! N-NEW CHARACTER
650 DISPLAY AT(1,1):"ENTER A
SCII CODE:" :: ACCEPT AT(1,1
8):CH
660 IF CH<32 OR CH>143 THEN
650
670 CALL PATTERN(#2,CH):: GO
SUB 770 :: RETURN
680 !E-EXIT
690 CALL SCREEN(8):: PRINT "
":"* READY *":"" :: CALL HCH
AR(24,2,62):: FOR D=1 TO 10
:: CALL HCHAR(24,3,32):: FOR
 C=1 TO 90 :: NEXT C
700 CALL HCHAR(24,3,30):: FO
R C=1 TO 90 :: NEXT C :: NEX
T D
710 PRINT "":"":"JUST KIDDIN
G!":"":"":"IF YOU'VE QUIT AC
CIDENTALLY,YOU CAN STILL SAV
E YOUR WORK":"":"DO YOU WANT
 TO SAVE IT (Y/N)"
720 CALL KEY(0,K,S):: IF S=0
 THEN 720 ELSE IF K=78 OR K=
110 THEN PRINT "O.K. BYE!" :
: END ELSE GOSUB 250 :: END
730 CALL CLEAR :: ! REDRAW S
CREEN
740 FOR R=1 TO 24 :: FOR C=1
 TO 31 STEP 2 :: IF SCREEN(R
,C)<32 THEN 745 ELSE CALL HC
HAR(R,C,SCREEN(R,C))
745 IF SCREEN(R,C+1)<32 THEN
 750 ELSE CALL HCHAR(R,C+1,S
CREEN(R,C+1))
750 NEXT C :: NEXT R :: IF C
OLORS(0,1)=0 OR COLORS(0,2)=
0 THEN 755 ELSE CALL COLOR(0
,COLORS(0,1),COLORS(0,2))
755 FOR C=1 TO 13 STEP 2 ::
IF COLORS(C,1)=0 OR COLORS(C
,2)=0 THEN 756 ELSE CALL COL
OR(C,COLORS(C,1),COLORS(C,2)
)
756 IF COLORS(C+1,1)=0 OR CO
LORS(C+1,2)=0 THEN 760 ELSE
CALL COLOR(C+1,COLORS(C+1,1)
,COLORS(C+1,2))
760 NEXT C :: CALL SCREEN(SC
):: RETURN
770 ! RE-DRAW FIRST LINE
780 FOR C=1 TO 32 :: IF SCRE
EN(1,C)=0 THEN CALL HCHAR(1,
C,32)ELSE CALL HCHAR(1,C,SCR
EEN(1,C))
790 NEXT C :: RETURN
800 CALL CLEAR :: DISPLAY AT
(12,2):"     SETTING UP DATA
": :" PLEASE WAIT A MOMENT T
HANKS"
810 FOR D=0 TO 9 :: H$(D)=ST
R$(D):: NEXT D :: FOR D=10 T
O 15 :: H$(D)=CHR$(D+55):: N
EXT D
820 FOR R=1 TO 24 :: FOR C=1
 TO 32 :: SCREEN(R,C)=32 ::
NEXT C :: NEXT R :: FOR C=0
TO 14 :: COLORS(C,1)=2 :: CO
LORS(C,2)=8
830 NEXT C :: CALL MAGNIFY(1
):: CALL CHAR(143,"814224182
44281"):: RETURN
839 ! INPUT TEXT
840 DISPLAY AT(1,1):"ROW" ::
 ACCEPT AT(1,5)VALIDATE(DIGI
T):R :: IF R>24 OR R<2 THEN
840
842 DISPLAY AT(1,8):"COL" ::
 ACCEPT AT(1,11)VALIDATE(DIG
IT):C :: IF C>32 OR C<2 THEN
 842
850 ACCEPT AT(1,1):T$ :: DIS
PLAY AT(R,C):T$
860 FOR D=C TO C+LEN(T$)+1 :
: CALL GCHAR(R,D,TC):: SCREE
N(R,D)=TC :: NEXT D :: GOSUB
 770 :: RETURN
900 !HGD
905 CALL CHAR(91,"0F0F0F0F0F
0F0F0F")
910 DATA 1,1,"**************
**************"
912 DATA 2,1,"* R.TWYNING'S
SCREENWRITER *"
914 DATA 3,1,"* MODIFIED SO
YOU CAN INPUT*"
916 DATA 4,1,"*  TEXT WITHOU
T CHANGING   *"
918 DATA 5,1,"*SPRITE CHARAC
TER EACH TIME*"
920 DATA 6,1,"**************
**************"
922 DATA 7,1,"[ [ [ [ [ [ [
[ [ [ [ [ [ ["
923 DATA 8,1," [ [ [ [ [ [ [
 [ [ [ [ [ [ ["
925 DATA 9,1,"++++++++++++++
++++++++++++++"
927 DATA 10,1,"+   J.M DORTI
G DORSET UK   +"
928 DATA 11,1,"+ WE HOPE YOU
 LIKE THE MOD +"
930 DATA 12,1,"+++++++++++++
+++++++++++++++"
932 DATA 13,1,"
               "
934 DATA 14,1,"
               "
936 DATA 15,1,"
               "
938 DATA 16,1,"
               "
940 DATA 18,1,"
               "
942 DATA 19,1,"*PRESS ANY KE
Y TO CONTINUE *"
944 !DATA 20,1,"*
                *"
946 DATA 21,1,"*************
***************"
950 RESTORE 910
952 FOR T=1 TO 19 :: ! DISPL
AY AT(R+1,C,W$)
953 READ R,C,W$ :: !PRINT R;
C;W$
954 PRINT W$ !DISPLAY AT(R+1
,C,W$)
955 NEXT T
956 RETURN
```

UNFINISHED EXPERIMENTAL ENCODING AND DECODING FILES USING UUENCODE

```
1 PRINT "":"":"":"" :: FOR D
=0 TO 14 :: CALL COLOR(D,3,1
):: NEXT D :: CALL SCREEN(2)
:: CALL CHAR(140,RPT$("F",16
))
2 INPUT "Path.filename of fi
le to    encode:":SF$ :: INP
UT "Path.filename of text fi
le  :":TF$
3 OPEN #5:SEG$(SF$,1,5),INPU
T ,RELATIVE,INTERNAL
4 INPUT #5:A$,A,B,C :: PRINT
 "":"selected disk:"&SEG$(SF
$,1,4):"Sectors free:";B-C:"
Sectors available:";C:"":"**
* SEARCHING ***":""
5 INPUT #5:A$,A,B,C :: IF A$
=SEG$(SF$,6,255)THEN PRINT A
$ :: GOTO 7
6 IF A=0 THEN CALL SOUND(200
,217,0):: PRINT "":"":"*** F
ILE NOT FOUND ***" :: RUN EL
SE 5
7 A=ABS(A):: IF A=5 THEN PRI
NT "":"":"":"Memory image (P
ROGRAM) filescannot be direc
tly encoded! Please archive
the file.":"" :: CALL SOUND(
4250,217,0):: STOP
8 PRINT "":"File matched O.K
.             Encoding Started
"

9 OPEN #2:TF$,DISPLAY ,VARIA
BLE 80 :: PRINT #2:"*TI UUEN
CODE*" :: IF A=1 THEN OPEN #
1:SF$,INPUT ,DISPLAY ,FIXED
C :: PRINT #2:"DIS/FIX"
10 IF A=2 THEN OPEN #1:SF$,I
NPUT ,DISPLAY ,VARIABLE C ::
 PRINT #2:"DIS/VAR"
11 IF A=3 THEN OPEN #1:SF$,I
NPUT ,INTERNAL,FIXED C :: PR
INT #2:"INT/FIX"
12 IF A=4 THEN OPEN #1:SF$,I
NPUT ,INTERNAL,VARIABLE C ::
 PRINT #2:"INT/VAR"
13 PRINT #2:C:SEG$(SF$,6,255
)
14 CALL GETCH(C):: IF C=3276
7 THEN PRINT "":"":" *** DON
E ***" :: CLOSE #1 :: CLOSE
#2 :: CLOSE #5 :: END
15 IF C=3000 THEN PRINT #2:"
a" :: GOTO 14
16 PRINT #2:CHR$(64+((A AND
240)/16))&CHR$(64+(C AND 15)
);:: GOTO 14
30000 SUB GETCH(C):: IF F$<>
"" THEN 30002 ELSE IF EOF(1)
THEN C=32767 :: SUBEXIT ELSE
 INPUT #1:F$ :: C=3000 :: SU
BEXIT
30002 C=ASC(F$):: F$=SEG$(F$

,2,255):: SUBEND
```

UUENCODE

```
1 PRINT "":"":"":"" :: FOR D
=0 TO 14 :: CALL COLOR(D,3,1
):: NEXT D :: CALL SCREEN(2)
:: CALL CHAR(140,RPT$("F",16
))
2 INPUT "Path.filename of fi
le to    decode:":SF$
3 OPEN #1:SF$ :: INPUT #1:A$
 :: IF A$="*TI UUENCODE*" TH
EN 4
4 INPUT #1:A$ :: INPUT #1:RL
 :: INPUT #1:N$ :: PRINT "":
"Use original filename (Y/N)
?"
5 CALL KEY(3,K,S):: IF S=0 T
HEN 5
6 IF K=ASC("Y")THEN INPUT "O
.K., what pathname (inc .) :
":P$ :: N$=P$&N$ :: GOTO 8
7 INPUT "Enter path.filename
      :":N$
8 IF A$="DIS/FIX" THEN OPEN
#2:N$,DISPLAY ,FIXED RL
9 IF A$="DIS/VAR" THEN OPEN
#2:N$,DISPLAY ,VARIABLE RL
10 IF A$="INT/FIX" THEN OPEN
 #2:N$,INTERNAL,FIXED RL
11 IF A$="INT/VAR" THEN OPEN
 #2:N$,INTERNAL,VARIABLE RL
12 PRINT "":"* Decoding *"
13 CALL GETCH(C):: IF C=3276
7 THEN 50 ELSE IF C=97 THEN
PRINT #2:"" :: PRINT :: GOTO
 13 ELSE C2=C :: CALL GETCH(
C)
14 PRINT #2:CHR$((C2*16)+C);
:: GOTO 13
50 PRINT "":"":" *** DONE **
*" :: CLOSE #1 :: CLOSE #2
30000 SUB GETCH(C):: IF F$<>
"" THEN 30002 ELSE IF EOF(1)
THEN C=32767 :: SUBEXIT ELSE
 INPUT #1:F$
30002 C=ASC(F$):: F$=SEG$(F$
,2,255):: SUBEND
```

UUDECODE

```
*TI UUENCODE*
INT/FIX
 128
ARK
a
@A@I@D@B@I@@@J@@@D@H@@@@@@@@@@E@M@L@H@D@B@A@H@D@F@M@G@L@F@C@A@H@D@F@M@O@@@H@D@A@H
@A@@@B@@@E@L@A@C@B@@@M@N@E@I@M@G@G@A@@@A@F@@@H@L@A@@@@@H@@@@@@@@G@M@J@@@@@H@A@H@@
@F@M@K@@@@@H@E@L@@@E@M@E@@@@@H@A@L@@@D@M@K@@@@@H@H@@@C@L@N@@@A@K@H@@@D@K@L@@@@
@A@I@@@@@B@K@B@@a
a
@H@A@K@E@C@E@E@N@A@A@H@B@L@C@C@I@E@O@O@C@L@@@E@J@@@J@D@N@L@J@@@F@H@A@M@L@A@C@G@I
@N@N@G@E@@@G@@@F@G@I@D@@@M@@@D@G@A@E@A@L@H@M@D@B@I@D@@@M@@@L@B@M@C@@@M@C@J@B@@@E
@H@K@D@A@L@N@A@O@B@H@@@J@C@H@M@H@L@C@E@H@E@@@D@B@E@A@E@A@H@B@J@M@K@B@L@@@I@K@C@E
@@@@@H@C@I@O@@@@@@@@A@@@B@L@E@F@@@@@@@C@@@H@D@C@K@B@N@A@B@K@@@A@@@I@@@@@@@@@@@B@@
@K@@@B@@@@@@A@@@@@E@@@M@@@E@@@@@@@@@@@@H@@@A@@@L@@@@@@@D@@@M@@@K@@@@@@@@@@@@H@@a
@B@H@@@G@@@H@N@A@M@H@B@L@A@I@G@@@D@E@N@H@H@F@E@A@E@H@B@I@N@K@A@L@E@N@K@F@A@D@G@L
@B@H@@@D@I@H@@@E@E@K@B@L@B@L@O@F@@@H@L@N@I@L@G@C@B@A@@a
@C@@@@@B@N@L@I@I@J@D@M@G@J@B@O@A@D@A@H@D@K@E@A@E@C@M@D@G@@@A@L@K@J@B@B@O@I@J@N@J
a
@D@E@I@N@L@I@G@D@D@A@D@M@K@G@A@D@@@J@O@A@N@E@J@E@A@H@O@A@E@A@D@M@J@B@A@J@C@I@G@A
@L@A@K@F@A@L@D@H@F@K@G@J@L@F@A@L@D@K@B@A@G@G@B@O@@@D@@@D@O@A@D@C@M@A@K@H@I@M@O@A
@B@K@M@C@@@I@G@@@H@H@I@C@E@M@C@@@D@M@C@G@M@@@G@N@G@H@H@@@E@K@K@@@M@N@B@H@L@A@L@@
@@@A@D@C@M@I@A@@@@@A@@@B@L@E@F@@@@@@@F@@@H@D@C@K@B@N@A@B@K@@@C@@@@@@@@@@@B@@@N@@
@L@@@I@@@I@@@J@@@D@@@M@@@G@@@J@@@B@@@O@@@A@@@D@@@A@@a
@D@A@L@B@M@D@G@@@C@H@D@B@H@F@A@G@@@H@C@@@E@A@E@B@C@C@H@M@@@A@A@K@M@H@L@H@B@L@A@M
@@@A@I@O@G@I@I@L@H@F@J@F@M@M@K@K@J@B@E@C@E@K@C@G@Oa
@G@J@B@M@@@@@F@H
```

DATES FOR THE TEST OF THE TIUG

"THE MOBB BBS"

The board will be open to all users on the following dates.

Monday 10th April 1995
Tuesday 18th April 1995
Sunday 30th April 1995
Monday 8th May 1995
Monday 15th May 1995
Wednesday 24th May 1995
Saturday 3rd June 1995 (Remote only at the AGM)
Friday 9th June 1995
Monday 19th June 1995

All dates will be on line for the period of 7.30pm to 10.30pm
The only thing that will not be available until it is fully sorted out will be the UPload and Download area. ALL else will.

How to get on.

You will need a modem that will send and recieve at 300bps minimum.
You will require a Terminal program such as TEII or TELCO.
The set up should then be as follows.
CR and LF flags OFF
Speed (yours) 8N1 using ADM3 or similar.
Ensure you have AT access and then ATD01623793077
Wait for the connect and you will find that all will reveal its self to you. ONE THING, Read the Board and it will be simple.

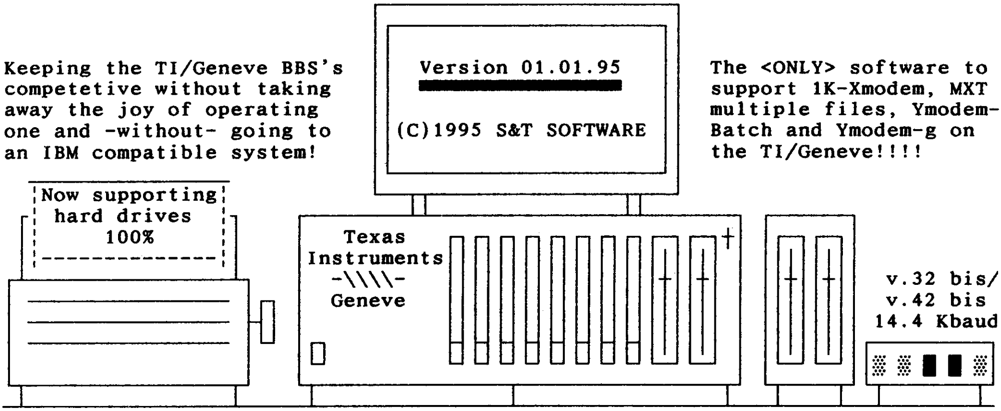
If you do have a go at seeing whats on I wish you a good time.
If in doubt there is a help section for you to look at.
If you want a chat and I am there I will respond to your calls.

```
Booting...THE MOBB 1995 TIUG

* Please Wait *


Hello Richard Twyning

Your password: ........

* Please Wait *
  DDDDDDDDDDD
            =ResumSu (cX)=Abort
```

Welcome to Another S&T BBS
Dedicated to the Free Exchange of Fairware & Public Domain Programs & Ideas

Keeping the TI/Geneve BBS's competetive without taking away the joy of operating one and -without- going to an IBM compatible system!

The <ONLY> software to support 1K-Xmodem, MXT multiple files, Ymodem-Batch and Ymodem-g on the TI/Geneve!!!!



<Press any key>

```
TRANSFERS
(U)ploads
(D)ownloads

OTHER
(C)hat/page SysOp
(G)oodbye
($) Quick Logoff
(-) Help!

INFORMATIONAL
(A)rticles of interest
(B)ulletin
(L)ast few callers
(I)nfo about this BBS
(O)ther BBS numbers
(V)endor Listing
(W)ant to purchase/
   register S&T BBS?

AREAS
(M)essage base
(R) Game Doorway
(E)dit your account
(S)ub-boards
(F)eedback to SysOp

*Registered v01.01.95*
```

```
FEEDBACK:


Enter Message (75 cols/30 lines Max!)
Press (Enter) on blank line for menu.

   (---------------------------------------------------------------------------)
01: I decided to use the GENEVE after all.   I was going to use the Amiga and
02: PROCOMM on the KCS, but I'm taping Quantum Leap for Paul abddthbbughttbhe
03: Amiga might interfere.
04: I'm glad I've used TELCO, I've discovered that I'm getting full ANSI
05: graphics out on my printer!!!!
06: Richard.
07:

<A>bort <C>ontinue <E>dit <L>ist <S>ave <N>ew <V>iew <D>elete <->Help: s

* Message saved *

<Press any key>


                         SUB BOARD MENU

          (1) BBS Advertisement      (2) General Stories

          (3) General Humor          (4) Doctor Who Fans

               [Press ENTER to return to Main Menu]


Select [c/r= exit (1 - 4)] :



FEEDBACK:


Enter Message (75 cols/30 lines Max!)
Press (Enter) on blank line for menu.

   (---------------------------------------------------------------------------)
01: Yo Dude.
02:     I don't know what occurred at the beginning of my session, but at first
03: it didn't seem to want to let me one   The modems are talking at 16800!!!
04: My terminal is set to 4800!!!!!
05: I might try an upload!
06:

<A>bort <C>ontinue <E>dit <L>ist <S>ave <N>ew <V>iew <D>elete <->Help: s

* Message saved *

<Press any key>
```

```
                    (cQ)=Resumes (cS)=Pause (cX)=Abort


 Written by SYSOP on  pe leav
/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-
HELLO!! All you Ti friends and citizens. This is the For sale Section
In this section you can leave your advert for sales and wants.
** REMEMBER ** To leave your name and address or telephone number so that
you can be contacted. Also that big question HOW MUCH.
The adverts will remain on line for as long as there is room or they are
sold. If sold leave SYSOP a message. TA.......

Written by SYSOP on  ae
>v
/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-
For SALE
Epson 80 dot matrix printer.
Will plug into the Rs232 PIO port with ease...
Cost 60.00 English pounds
Contact
T. Stevens
01623793077


<Press any key>


Add text (Y/n)? n




01: Greetings from your ever-TI-faithful General Secretary, Richard Twyning.
02: I am pleased to write this message to you, loyal supporters of this
03: excellent machine.  Today (9th Feb 1995) I have been contacted by the
04: Science Museum in London with regards to finally making our donation of
05: an old console to their establishment!
06:     Thank you for supporting your user group, and accessing this BBS.
07: Your subscriptions have provided it, and your loyalty is appreciated.
08: Long live TI-99/4A User Groups, and long live The MOBB and all other TI
09: BBSs around the world.  Thanks to Tim Tesch for this excellent software.
10: Richard Twyning.


<A>bort <C>ontinue <E>dit <L>ist <S>ave <N>ew <V>iew <D>elete <->Help: s

Saving message #2

* Message saved *

          Sorry..

          This section is closed for now.  If/when this area
          becomes available you will be notified!


<Press any key>
```

```
TRANSFERS                 INFORMATIONAL                       AREAS

(U)ploads                 (A)rticles of interest              (M)essage base
(D)ownloads               (B)ulletin                          (R) Game Doorway
                          (L)ast few callers                  (E)dit your account
OTHER                     (I)nfo about this BBS               (S)ub-boards
                          (O)ther BBS numbers                 (F)eedback to SysOp
(C)hat/page SysOp         (V)endor Listing
(G)oodbye                 (W)ant to purchase/
($) Quick Logoff             register S&T BBS?
(-) Help!                                                     *Registered v01.01.95*
```

```
S&T Software copyright (c)1995 by Tim Tesch / Scott Stasiowski

**  Thank you for calling!  Now giving Ma Bell the boot...  **


Thanks for calling THE MOBB 1995 TIUG

Logging you off at: *  Thank am

(c)1995 S&T Soft.

Disconnecting...

OK
```

CaDD ELECTRONICS release Stage 3 of PC99

Review by Trevor STEVENS

CaDD Electronics have released another stage of the IBM & Pc Compatable program PC99. The program will only run on a machine which is 286 or higher but is only effective on a 386/33 or above. The speed that the program runs below this machine is Slooowwww. However with the 486/33 the speed compares with that of your original TI.

Cpu	99/4a	486/33	486/50	486/100	Pentium
Norton SI	n/a	72	108	180	286
MG clock	30.00	30.84	21.04	12.57	8.11
1-10000	34.4	39.50	26.70	15.20	10.7
1-500 With Print	26.3	35.80	23.10	13.49	8.64

The new program in fact comes in the full packeage and what is called a light package. The full costs $95 and the light $47. The difference between the two is the light version is the cut down program to the full program and does not offer ALL the extras.

The program now emulates the TI99/4a which would equate to a TI99/4a with full expansion containing 32k of memory, TI disk controller, RS232card, TI p-Code Card, Three Disk drives (Dssd), with modules of TI Extended Basic, TI Editor Assembler and Tombstone City. If you have your TI it has with the package a five and quarter TI disk to transfer files from your TI to the Pc. The files are passed across as disk image files, (A whole disk) or as a gramcracker format for modules. Also the package gives a bonus of 16 grom slots which enable you to load up 16 cartridges and access them from the unused Reveiew Module library that is in the TI. What this does is bring up a list of modules on screen 2 after the main startup screen. Included in the top program is mini screen debugger which gives you total control over the TI memory map.

The only draw back that has been found is that some programs such as Telco, that mess around the memory map and access various compares do screw up the system but these programs are few and far between. As far as value for money this is an A1 product. If any of you out there are thinking of turning to a IBM Pc and don't want to lose your faithfull friend or your I Group then this is a good answer. The only thing which is a draw back is that it is a TI99/4a so it has the same screen defaults. Prehaps the next step will be to make the emulator run 80col software, I hope so because that will be a real world beater.

PC99 Stage three, can be obtained from
CaDD Electronics
45 Centersville Drive,
Salem
NH 03079
Tele (603)8931450 or (603)8950119

Prices are $95 for the full version $47 for the light. Updates 2a to 3 $7 (new disks). Return originals $5. If ordering you will find sending a International Money order a real help.

2022 note: PC99 (dos) last release was Stage 6

DISK LIBRARY....

Here is a long list of additions to the disk library over the past couple of years- not too well reported in TI*MES recently, although as always all you need to obtain a copy of the NEW/DISKS list is to send a disk and return postage, nothing else!

For a COMPLETE library listing send four disks and return postage!

The disk library charges one pound per disk copied plus one pound per order - eg 4 disks = 4 + 1 = FIVE pounds. That is- if you send in the blank disks!!!

Note: ONLY DOUBLE (or quad) DENSITY DISKS!!!

I WILL NOT PUT HIGH DENSITY DISKS INTO MY DRIVES!!!

High Density disks have a very different coating and utilise different signal levels - both for recording and playback!

If you prefer, I can use some surplus (eg used) library disks, supplied for 50p each in addition to copying costs.

NEW DISKS JANUARY 1993

THE DRAWING MASTER is now updated to version 1.3.

I have a new version of Danny Michaels DUMP program modified by Bruce Harrison- ask for the Harrison version if interested.

>UTIL26 has added FANCYLIST which reads a DV163 merge file and prints various types of token in differing fonts.

>UTIL32 is now full following the addition of Jim Peterson's PRINTALL Version 1.8, a multi-column printing utility.

>MICROPENDIUM- programs from the magazine- CHEMICAL Elements-Regena-10/92; ; SKI/UTAH -Regena-11/92-Tourist guide! ; SPRITES -W Shepard, 10/92- demo ; XBCOMPARE- Barry Traver-11/92- compare XB progs for modificatios ; plus game FORE/AFT and utility HELPWRITER- Romstadt and Harrison- insert HELP screens into XB programs to call up at the touch of a button, and return to calling screen.

>TI GAMES VIDEO. About 5 hours of play showing various modules in use, including some "unreleased" modules. No commentary. Two pounds per week rental plus fifteen pounds returnable deposit.

>PHILLIPS 1. This series of disks is based upon the many programs of John Phillips. This first disk contains STARGAZER 1, 2, and 3, runnable from XB. These are educational programs based around the patterns of the Constellations. Also on the disk is to be found: MUNCHMOBILE, and a version of SCRABBLE, which with the loader on this disk has the LARGE character set.

>PHILLIPS 2. MUNCHMAN II-the two screen version with transporter pads. REQUIRES A LOADER eg FUNLWEB. D STATION- loader included is inadequate, use funlweb. 4A FLYER with manual on disk, MR PACMAN (by W Becherer), and STRIKE 3, a baseball simulation, which like all baseball simulations I have seen I personally find quite impossible.

>PHILLIPS 3. TWO DISKS. THE COMMENTED SOURCE CODE for 4A FLYER. Includes assembled versions to run from EA3 and EA5 and the manual, on disk.

>PHILLIPS 4. TWO DISKS. The commented source code for MUNCHMAN II. Includes versions to run from EA3 and EA5.

>PHILLIPS 5. Commented source code for STARGAZER 1.

>PHILLIPS 6. Commented source code for STARGAZER 2.

>SOKOBAN. A new machine code game from our group Treasurer, a block moving puzzle said to have 50 solvable screens. I can't get past level 5 however! Versions for Ed As Opt 2 and 3, the latter of which should load with most XB loaders such as that to be found with Funlweb.

>THE ULTIMATE ACCEPT AT by Bruce Harrison, a machine code CALL LINK for Extended Basic programmers. The title says it all!!!
-documentation in TI*MES Issue 47-

NOVEMBER 1993....

>BODENMILLER DEMO DISK. DOUBLE SIDED DISK ONLY. A very unusual demo disk of two programs from Bodenmiller Computers. (dated 1993).

>GOBLINS+ STAR MISSION. DOUBLE SIDED DISK ONLY. Both games are simple shoot outs, but Goblins has some very interesting use of sampled sound-speech without a synthesiser (the speech is Italian, as this came from Rome!). Games instructions are very brief and in English! No apparent scoring for either game, just keep going as long as you can!

>ODDMOD3 now has both the UK and USA versions of Household Money Management/Household Budget Management. Take your pick!

>XBMOD4 now has an ancient unreleased module added, DISK DUPLICATOR, a cut down version of disk manager BUT this one has the facility to copy USCD Pascal Disks, disk manager cannot do that!

>MINESWEEPER. An old program with varieties from TI Basic to PC+Windows, this one is in c99, and will run from XB. Several options, and quite a challenge!

new series of util disks-

UTIL A:
DEFRAGMENTER uses just ONE disk drive, and puts fragmented files back into one piece quite quickly. Up to 32 files maximum. plus DIGITISER which reads sound into the cassette port and "digitises" it- note that this is a "one byte sampler" using square waves, so the quality can be difficult!

UTIL B:
KWIKDUMP- Harrison- Fctn 7 dumps screen TEXT-no graphics- fast. ; REMINDER-scans a DV80 address list and prints labels date flagged; ; GROM Master-experimental program to drive a self build gram type device; ; SAY-barry boone- experimental speech program ; SEARCH- adds to XB: CALL LINK("SEARCH",A$(),"LOOK",FOUND) to quickly locate a particular string in an array ; READER.

>UTIL C is a new disk and at present JUST has a program to print labels for those rather solid 3.5" disks plus a useful XB utility to allow you easy access to VDP ram for fast character definitions, screen displays etc.

>MAZE MANIA is a nice disk with 13 mazes for you to solve on screen (some of them are bigger than the screen and scroll up and down!) or you can make it easy and print them out to do!

>DIGISYNT from Italy. Feed sound into the cassette port and store it on disk! Then from an XB program, load it in and PLAY the sound! Limited due to the one level limited frequency input, but can be quite understandable. Works best with sounds which are not too complex, and not too great a variation in volume. Turn volume as high as needed to avoid "noise" but not so high as to create distortion. Computer needs 1V sound input.

THE CASTLE by Vern Jensen. DOUBLE SIDED DISK FOR DSK2. Can supply on two SS disks but you'll have to do a little work amending many XB program segments! SUPERB game program for all the family. Explore the Castle room by room, picking up treasure and avoiding triangular things.... XB but with m/code support for fast action. THE US CONSTITUTION. Not as text files but as odd data files with its own reader program which includes a search facility.

SPACE STATION PHETA DOCUMENTATION DISK to go with the game we have had for quite some time now.

Jeff's GENEVE disk- programs indicated as being for the GENEVE: BREAKUP - a file which will break a large file into smaller "chunks" SHOWALLI - a file that will show all chosen ti-artist instances (using wildcards SHOWGR - a file to show the "/gr" files that were popular a few years back note: files must be renamed "_gr" SHOWTIAI - a file to display a given ti- artist instance

MORE FROM BRUCE HARRISON:

HARRISON REFORMATTER. A utility which takes as input any DV80 file, and outputs a separate DV80 file reformatted to any width, with options for changing the left margin or adding right adjustment. VERY FAST. It is also possible to reformat text with no carriage returns, using a line terminating as a full stop instead.

HARRISON BACKGROUND MUSIC. A disk of machine code utilities to allow you for example to play music while awaiting an ACCEPT input, and to "time out" an ACCEPT AT input. Users will find it handy to also ask for disk SSPD4 which contains a utility that provides the sound data table values.

>VIDEO TITLER and drawer. Main program loads two TI Artist pictures into memory and offers several ways of changing from one to the other (scroll from left, right, top, bottom, curtain open, curtain close) and the Drawer program lets you create screens and can also use TI Artist files. TWO DISKS.

>COMPILER. TWO DISKS. Takes an XB program as input, compiles FOR NEXT, ON GOTO, COLOR, KEY, GCHAR, HCHAR, VCHAR, and PRINT, and leaves the rest alone. Final result is a file which looks and runs like an XB program to the system, and can be BROKEN etc but is really machine code! Programs using the compiled parts of the language most heavily will see the greatest speed increase (up to 4 or 5 times) while some programs may run slower.

>COMPILER SOURCE CODE for anyone who wants to improve it- Bruce has had enough! TWO DISKS.

>CALL FILES UTILITY allows you to do a CALL FILES in a running program!

>TURNFONT allows you to rotate the characters in a CHARA1 disk file by 90 degrees (or 180 or 270 of course!).

ALL DISKS ARE AVAILABLE AT A COPYING FEE OF ONE POUND PER DISK SIDE PLUS ONE POUND PER ORDER. If you do not wish to send blank disks please add 50p per disk requested. Disks are are single density, and unless otherwise specified are single sided. The vast majority of disks assume ownership of Extended Basic and sometime also Funlweb or an equivalent machine code loader.

FULL DISK LIBRARY details available on DISK ONLY in the form of DV80 files -just send four disks and return postage.

>FUNLWEB version 5.0 80 column only. EDITOR ONLY- add on files to main disks. ON TWO DISKS. Includes lots of extras- UPDATED TO JULY 1993.

>FUNLWEB verson 5.01 80 column FOR 192K VRAM ONLY. SPECIFY VERSION!!! DS DISK. May 94 (could supply on 3 SS disks-please specify).

>EURO-FORMATTER is for an option of the above, and includes text files on the new Editor. TWO DISKS.

>FUNLWEB VN 5.01 40 COLUMN EDITOR- updated to May 1994- includes multi language and 8-bit modes and lots of extras! TWO DISKS. EDITOR ONLY! Intended as replacement files for Version 4.40 Funlweb only- if you have an earlier version you need Vn 4.40!

>FUNLWEB STANDALONE, by special request, the simplified version, only TWO DISKS please, for a working copy of the enhanced VERSION 5.01 EDITOR plus the original formatter plus Disk Review and the loaders for machine code programs, together with docs for ONLY the editor and disk review. Nothing else to confuse you. But DONT complain about missing utilities or docs! You get only what is described here! 40 column version. Probably all that most owners will use! TWO DISKS.

Received August 1994:

>TEXT SORT by Bruce Harrison, modified by Bob Carmany. Sorts DV80 records.

>NEWSLETTER EDITORS AND FORMATTER by Bill Gaskell- allows printing in two columns, with up to 114 lines per page (2 x 57) and rows of up to 36 characters. Enlarged headings, page headings and numbers possible.

>CARD FILE vn 3.0 by Bill Gaskell- a sort of computerised 3x5" record card system on computer. No docs.

>CARTRIDGE DATABASE FOR CARD FILE. TWO DISKS. Data on modules for the above utility. Two disks.

>GRAM FILES. In the format: FATOM1 plus FATOM11. If your GRAM DEVICE uses files in this format try these! FOUR DISKS but could split! Cannot use so don't know what they are but titles are: EURODM FATOM FUNK GAME GPLASSM HENHOUS MANCALA DEMOMOD(said to be German) READADVEN READCHEER DATABASE PAINT/PRI SUPERDUP TI-IBM-CC TYPOMAN

>RISK. Machine code war game for 2 to 6 players. Move armies around world map and do battle. Simple play. By O Arnold. Vn 1.02 1994 WRITTEN TO USE TWO JOYSTICKS!

Added to >5 EDAS-T3 are three new adventured for the TUNNELS OF DOOM module which is required (on module or disk!)- Dune, Garfield, Ninja

Added to >GAMES 23 is Vn 3.1 of the Orphaned Instruments Title Screen by Chris Lang- a sort of impolite alternative to TI's. The FCTN NUMBER keys produce interesting results in this version.

>SCOTT ADAMS ADVENTURES 5. Requires ADVENTURE module on module or disk. Titles are Alcatraz, Bigfoot*, Cannibal, Cap'Kidd, Great White, Moon, Nessy, and Travelling.

>ADVENTURES OF TOM SAWYER Text 20 chapters on 4 disks. DV80 format.

>SHERLOCK HOLMES. In ARCHIVED format - requires Archiver (availabler from library). Also has a utility ro READ the dv80 text files (speech synth required). Utility available separately if required. 28 DISKS! Unarchives to a huge number of disks! More details available on request- could unarchive for you if required, and could supply just single stories - higher charges for specialist work like this though!

NEW DECEMBER 1994:

>BRUCE HARRISONS SLIDESHOW... this program produces a slideshow of TI Artist pictures, either timed or keyed changes. Timed changes are up to every·5 minutes, and accuracy is good (program is written for EUROPEAN consoles). Pictures on a disk can be sequenced as you wish with each file shown up to three times in the sequence. The usual screen time out remains active should you leave a picture on screen long enough! A chosen sequence may be run continuously or once only. (Up to 64k times!).

>BRUCE HARRISONS EUROPETIME... this is a tool for XB programmers as they can CALL LOAD a short machine code routine and then CALL LINK to obtain a time elapsed value. Useful when maximising the efficiency of an XB program could also be used to display elapsed times in a program. Depends upon interrupts, so very accurate - to 0.3% - written for European consoles. Note however that interrupts halt for file access, screen scroll and speech, which can result in inaccurate reports, but constant for the particular program element.

>BRUCE HARRISONS ULTIMATE DELAY... another timing program from Bruce, again written for Europe so very accurate subject to particular program elements! This program can halt execution of a program for a fixed period, or halt execution for a fixed period or prior keypress, AND can report which key was pressed. For you to insert into your XB programs. Can be employed usefully when displaying instruction screens, to wipe the screen after a keypress or after an excessive waiting period! The delay is passed to the routine in seconds; decimals are permitted, maximum... 64k! say 5 mins.

Please send your orders to:
Stephen Shaw 10 Alstone Road STOCKPORT Cheshire SK4 5AH

Cheques payable to Stephen Shaw please.

Have you counted the number of disks correctly? Please recheck!

FROM THE CHAIRMANS CHAIR

Winter of 1995

Hello to you out there in TI land. I have been very busy over the past few months with several projects. Two of which are not TI related and the others very much so. Due to this I did not write last quarter. So I must say sorry if any of you missed my input, or perhaps not whichever the case .

The first of these TI projects was to get up and running the TIUG BBs. We now have the very good program S&T bbs software. This is a multi width BBS so that you can log on as 40 or 80 coloumns, depending upon your system. Most TI'ers use 40. However if you have a PC, Amiga, Mac, Geneve or 80 col TI/994a then you can log on at 80. The board has full ANSI graphics which come up like the Dos type windows on the Pc. There is a on line game with a colour graphics as a startup screen. You will however need the correct terminal program to access the extra modes, but they are there!!!
I have had everything running at 19200bps connected on the modem. Richard Twyning came on and managed to get round most of it, in its test form. There is one problem that I have at the present time and thats the upload and download section. Try as I might I can't get the darn thing to save a file or send one out. So I will be writing to Tim Tech for som BIG BIG help on that one. I know the program works as there are several around that are being used at this time in the USA. I am now hoping in spite of this problem to do some test dates for you so look at the end of this article for those special dates.

The second project was helping Gary get the Magazines out to you. Issue 47 for some reason was a real blighter. The mag was late in being put together due to various problems which were out of our control. When the thing did get back to us, the printers did a right good one.
They printed one artical (Tips 35) twice. By this time it was getting late and we had to go ahead with it. So SORRY for that. I do intended to reprint this missing article as soon as possible and get some compensation from the printers.

Gary SMITH and Alistair BRYCE have made it known that they will be giving up the posts of EDITOR and MEMBERSHIP SECRETARY with effect from the AGM. So if any of you out there fancy a real challange, and wish to give up a little of your spare time then write to me.

I have received some letters of late from Ashley Tilling, Ken Hughes and Walter Allum. I am sorry that I have not replied to you but I will now in public so that everyone can see that replies are made eventually. Ken started with a letter saying he had been on to Mark Wills and used his small talk BBS. This BBs is a test version run by Mark. The full time one will be on line in about three months on its own private line, 24hrs a day. The group are also looking at hiring out at a small charge Modems for those with full sytems or RS232's. The reason why members were not informed about the Small Talk BBs was that it was done when Mark was contacted and was not on line at any specified dates.

Ashley Tilling wrote to me as he felt that the TI was now subject to vapourware. He mentioned the 80 col card. There are avalable now (80 col cards for sale NEW) from GERMANY. These cards have 16 million colour ranges approx and fit onto the side port. Francesesco LAMA (See front cover) can provide you with all the information and I believe other products. He I am told has only just purchased one, as has one other member Dereck Hayward. Our card is still in the production stage. What held it up was a faulty transitor which stopped us reading the address lines. Gary SMITH is the one developing the package which will include a speech card throughport. Write to him for details. As regard to the other vapour ware stuff well, they said the BBS was such an animal but IT IS here. Ashley stated also in his letter that due to the group getting smaller this ment the die hards and TI core were exposed and left the small user out of it. He stated he was seriously thinking about going over to the IMB PC. I have a IMB Pc a Amiga and a TI Power Pc. The best one of the lot for the serious user is the TI99/4a. YOU CAN DO MORE ON THE TI. That is a fact. I had a call not so long ago and I got the following remarks. I used to own a TI and I now own a PC. I loved doing things on my TI like programing. It was so easy. I have found a program called Pc99, on the IBM. (On sale by CaDD Electronics ($95) and now out as revision 3 which includes 16 grom banks and P-System), Can I come up and see you as I want the fun back in my life. This came from a very expert Pc programmer who owns his own softwarehouse SIMONSOFT. I will say no more. However if you feel left out write to your Editor put in a query to say the consoltation zone. Every one will learn and find out more.
Walter was also of the same vain about the group. He has stated that he has heard of personality clashes. This of course happens in all walks of life. However I try to be the man in the middle and do not try to take sides. However there will always be someone who is not as satisfied as others. What the group is trying to do is to bring everone just a little pleasure into their life. If you feel that you could write a small artical or whatever we will print it as long as it falls within the Group rules.* Walter produced some "Codes of Pratice". These Walter I will take to the AGM and will let every one discuss the proposals.

THE AGM

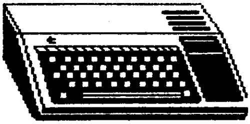
[Never specified or printed anywhere...]*

The AGM will be held this year at our usual venue of Derby. The full map is on the rear of this issue. The AGM starts at 10am and finishes at 5pm.
There will demo's and talks this year. At the show will be a TI Power Pc, Amiga 500+ with data link to the TI. for you to use. Progams also will be available.

DATE OF THE AGM IS: 10am Sat 3rd June 1995
St. John Ambulance Training Centre,
Trinity Street,
Derby.
WE HOPE TO SEE YOU THERE. WE LIKE TO TALK TO YOU
SO WE CAN FIND OUT HOW TO MAKE YOUR GROUP BETTER.

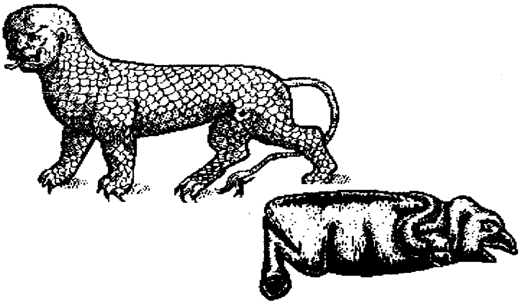
DORTIG

{DORSET T.I GROUP}



As I was using PRO PAGE I thought, why not do a letter using this. Somewhere in me chaotic system I have some stuff I've dne for various occassions should I manage to find a decent one then I will slot a page in to make the space up
Still wet and windy but so far NO SNOW not like you unfortunate people u north,and no floods near us yet.
I am told I can drive again now so I will call on Trevor{the MICRO},my pet name for him, I must get his Radioham call sign s that anyone wishing to contact him over the air that will Keep him happy.

Trevor had trouble with his HAMSOFT program so he has the system built by David Hewitt,which we purchased off David, and I must see it working with the Hamsoft program that David built into the system.
The program we are submiting this time is one based on another sprite article by T. Stevens, I hope it Keeps fingers busy,or are we all too busy decorating during the winter months!!. I will be glad to get out on the garden again and do some cleaning up.
Trevor and I are paying a call on Graham to see his TI Emulator running on a PC,will have to make a report on how it works and is it worth getting a PC!!!.
Thought you might like to see some of our PETS?.



It' best to Knock

at the front door!!

Do not walk around

to the back door. JM

TS/SPRITE1 P.Y.

TI Sprite 1 by JM based on T Stevens TI*MES 41 p26

NOTE: A £ symbol should be replaced with a #

```
40 ! THIS PROGRAM BASED ON:-
   TI*MES ISS: 41 P.26 .
SPRITE PROGRAM by T. Stevens
 SAVED AS 'TS/SPRITE1'.
 BY J.M. DORTIG U.K.
42 NC=1
45 DIM Y(28),X(28)
50 RANDOMIZE :: CALL CLEAR :
: C=3
100 FOR N=1 TO 28 :: CALL SP
RITE(£N,42,C,N*6,N*2,0,N*4):
: C=C+1
102 IF C=16 THEN C=3
105 NEXT N
106 ON NC GOTO 107,5000,7390
,5320
107 FOR T=1 TO 2 :: PRINT "H
ELLO" :: CALL DELAY(500):: C
ALL CLEAR
140 PRINT "HELLO THERE" :: C
ALL DELAY(500):: CALL CLEAR
:: NEXT T
163 DISPLAY AT(23,2):"ALL TO
GETHER NOW CHANGE" :: CALL D
ELAY(100):: CALL SCREEN(2)
170 CALL PATTERN(£1,71,£2,82
,£3,69,£4,69,£5,84,£6,73,£7,
78,£8,71,£9,83)
180 CALL PATTERN(£11,70,£12,
82,£13,79,£14,77,£16,68,£17,
79,£18,82,£19,84,£20,73,£21,
71)
185 CALL PATTERN(£23,68,£24,
79,£25,82,£26,83,£27,69,£28,
84)
190 CALL DELAY(380)
200 FOR CC=5 TO 8 :: CALL CO
LOR(CC,16,2):: NEXT CC
210 DISPLAY AT(23,2):"NOW SO
RT YOURSELFS OUT" :: CALL DE
LAY(860)
220 FOR N=1 TO 14 :: CALL MO
TION(£N,0,0):: NEXT N :: CAL
L DELAY(150)
230 FOR N=15 TO 21 :: CALL M
OTION(£N,0,0):: NEXT N :: CA
LL DELAY(160)
240 FOR N=22 TO 28 :: CALL M
OTION(£N,0,0):: NEXT N
250 DISPLAY AT(23,2):"THAT I
S BETTER JUMP TO IT" :: CALL
 DELAY(100):: CALL MAGNIFY(2
)
260 FOR N=1 TO 18 :: CALL MO
TION(£N,-15,0,£N+10,-25,0)::
 CALL DELAY(10):: CALL MOTIO
N(£N,15,0,£N+10,25,0):: CALL
 DELAY(10)
265 CALL MOTION(£N,0,0,£N+10
,0,0):: NEXT N
270 CALL DELAY(200):: DISPLA
Y AT(23,2):" AND THE REST YO
U DOZY LOT" :: CALL DELAY(15
0)
272 CALL MOTION(£2,-2,0,£3,-
3,0,£4,-4,0,£5,-4,0,£6,-5,0,
£7,-6,0,£8,-7,0,£9,-7,0)
275 DISPLAY AT(23,2):"SPELL
THIS YOU LOT":"   TO TI*MES"
 :: CALL DELAY(150)
276 CALL MOTION(£2,0,0,£3,0,
0,£4,0,0,£5,0,0,£6,0,0,£7,0,
0,£8,0,0,£9,0,0)
280 CALL MOTION(£5,4,-28,£13
,-10,-5,£19,-18,-4,£6,10,-20
,£22,-18,3,£14,-4,9,£4,19,11
,£9,16,-16):: CALL DELAY(173
)
290 CALL MOTION(£5,0,0,£13,0
,0,£19,0,0,£6,0,0,£22,0,0,£1
4,0,0,£4,0,0,£9,0,0)
300 DISPLAY AT(23,2):"NOT BA
D FOR LEARNERS":" NOW BACK T
O NORMAL" :: CALL MAGNIFY(1)
:: CALL DELAY(300)
310 NC=2 :: GOTO 50
4990 ! WIRLY-BIRDS
5000 CALL CLEAR :: CALL CHAR
(96,"9F447C7A3F1E0A1F",97,"2
444DC7A3F1E0A1F")
5010 CALL SCREEN(2):: CALL M
AGNIFY(2)
5020 DISPLAY AT(22,2):"  HOL
D ONTO YOUR HATS":"   WIRLY
BIRDS COMING"
5040 FOR N=1 TO 27 STEP 2 ::
 CALL PATTERN(£N,96,£N+1,97)
:: NEXT N :: CALL DELAY(200)
:: DISPLAY AT(22,2):" ":" "
5045 FOR H=1 TO 5
5046 FOR N=1 TO 21 STEP 6
5050 CALL PATTERN(£N.97,£N+1
,96,£N+2,97,£N+3,96,£N+4,97,
£N+5,96,£N+6,97)
5052 CALL PATTERN(£N,96,£N+1
,97,£N+2,96,£N+3,97,£N+4,96,
£N+5,97,£N+6,96):: CALL DELA
Y(2):: NEXT N
5060 NEXT H
5070 FOR N=1 TO 28 :: CALL M
OTION(£N,0,0):: NEXT N
5080 FOR PN=1 TO 28 :: CALL
POSITION(£PN,Y(PN),X(PN))::
NEXT PN
5085 FOR N=1 TO 27 STEP 2
5090 CALL MOTION(£N,5,0,£N+1
,5,0)
5095 CALL PATTERN(£N,96,£N+1
,96):: CALL DELAY(10):: CALL
 PATTERN(£N,97,£N+1,97):: CA
LL POSITION(£N,YY,XX)
5100 IF YY>170 THEN CALL MOT
ION(£N,0,0,£N+1,0,0)ELSE 509
5
5120 NEXT N
5130 DISPLAY AT(2,3):"RIGHT
BACK TO YOUR PLACES" :: CALL
 DELAY(300):: DISPLAY AT(2,3
):" "
5133 FOR N=1 TO 27 STEP 2 ::
 CALL MOTION(£N,-5,0,£N+1,-5
,0)
5134 CALL PATTERN(£N,97,£N+1
,97):: CALL DELAY(5):: CALL
PATTERN(£N,96,£N+1,96)
5135 CALL DISTANCE(£N,Y(N),X
(N),D)
5136 IF D<10 THEN 5140 ELSE
5134
5140 CALL MOTION(£N,0,0,£N+1
,0,0):: NEXT N :: CALL DELAY
(200)
5300 NC=3 :: GOTO 50
5320 CALL SCREEN(2):: CALL M
AGNIFY(2)
5330 FOR N=1 TO 10 :: CALL M
OTION(£N,0,0):: NEXT N :: CA
LL DELAY(140)
5340 FOR N=11 TO 16 :: CALL
MOTION(£N,0,0):: NEXT N :: C
ALL DELAY(40)
5350 FOR N=16 TO 28 :: CALL
MOTION(£N,0,0):: NEXT N
5355 A$="THANK**YOU*FOR***WA
TCHING*"
5357 L=1
5360 FOR N=1 TO 26
5370 B$=SEG$(A$,L,1):: CALL
PATTERN(£N,ASC(B$)):: L=L+1
:: NEXT N
5380 CALL DELAY(800):: END
7000 ! TI*MES ISS: 41 P.26 .
  SPRITE PROGRAM by T. Steve
ns SAVED AS 'ST/MUS9'
```

£ = #

```
7010 RANDOMIZE :: CALL CLEAR
7390 CALL CHAR(139,"3828BA92
FE38382828",141,"3828BA523C7
CFE28",140,"382838163838181 4
",142,"382879523C7CFE14")
7410 CALL MAGNIFY(2):: DISPL
AY AT(23,2):" NOW GET CHANGE
D AGAIN" :: CALL DELAY(100)
7420 FOR N=1 TO 27 STEP 2 ::
 CALL PATTERN(£N,139,£N+1,14
1):: DISPLAY AT(23,2):"HURRY
 UP YOU LOT" :: CALL DELAY(1
0):: NEXT N
7430 FOR N=1 TO 28 :: CALL M
OTION(£N,0,0):: NEXT N
7440 GOTO 7780
7765 ! ** SOUND AND GRAPHICS
,PAGE 131,ENGLISH GARDEN **
7770 REM **Copied by J. Murp
hy for DOR-TIG**
7780 CALL CLEAR :: CALL DELA
Y(200)
7810 S1=4 :: S2=-4 :: P=139
:: P1=140 :: P2=141 :: P3=14
2 :: T=650
7820 LC=131 :: LD=147 :: LE=
165 :: LF=175 :: LG=196 :: L
AF=208 :: LA=220 :: LBF=233
:: LB=247
7830 C=262 :: D=294 :: E=330
 :: F=349 :: G=392 :: A=440
:: B=494 :: HC=523
7835 DISPLAY AT(24,2):"   DO
RTIG DEMON DANCERS!!"
7840 FOR TT=1 TO 5
7850 CALL SOUND(T,HC,0,LC,0)
7860 CALL MOTION(£1,0,S1,£2,
0,S2):: CALL SOUND(T/2,HC,0,
LE,0):: GOSUB 8280
7890 CALL SOUND(T/2,B,0,LE,0
):: CALL MOTION(£27,0,S1,£28
,0,S2):: CALL SOUND(T,A,0,LF
,0):: GOSUB 8290
7930 CALL SOUND(T,A,0,LF,0):
: CALL MOTION(£3,0,S1+1,£4,0
,S2-1):: CALL SOUND(T,G,0,LG
,0):: CALL MOTION(£25,0,S1+1
,£26,0,S2-1):: GOSUB 8300
7970 CALL SOUND(T/2,G,0,LB,0
):: CALL MOTION(£5,0,S1,£6,0
,S2):: GOSUB 8310 :: CALL SO
UND(T/2,F,0,LB,0):: CALL MOT
ION(£23,0,S1,£24,0,S2):: GOS
UB 8320
8010 CALL SOUND(T,E,0,C,0)::
```

```
 CALL MOTION(£7,0,S1+1,£8,0,
S2-1):: GOSUB 8330 :: CALL S
OUND(T/2,E,0,C,0):: CALL MOT
ION(£21,0,S1+1,£22,0,S2-1)::
 GOSUB 8280
8050 CALL SOUND(T/2,F,0,C,0)
:: CALL MOTION(£9,0,S1,£10,0
,S2):: GOSUB 8290 :: CALL SO
UND(T,G,0):: CALL MOTION(£19
,0,S1,£20,0,S2):: GOSUB 8300
8090 CALL SOUND(T,C,0,LBF,0)
:: CALL MOTION(£11,0,S1+1,£1
2,0,S2-1):: GOSUB 8310 :: CA
LL SOUND(T,D,0,LA,0):: CALL
MOTION(£17,0,S1+1,£18,0,S2-2
):: GOSUB 8330
8140 CALL SOUND(T,F,0,LAF,0)
:: CALL MOTION(£13,0,S1,£14,
0,S2):: GOSUB 8280 :: CALL S
OUND(T,E,0,LG,0):: CALL MOTI
ON(£15,0,S1+2,£16,0,S2-2)::
GOSUB 8290
8170 CALL SOUND(T/2,E,0,LF,0
):: GOSUB 8300 :: CALL SOUND
(T/2,D,0,LF,0):: GOSUB 8310
8210 CALL SOUND(T,C,0,LE,0):
: GOSUB 8320
8230 S1=S1+3 :: S2=S2-3 :: T
=T-10
8250 NEXT TT
8260 FOR N=1 TO 28 :: CALL M
OTION(£N,0,0):: NEXT N
8270 CALL DELAY(300):: NC=4
:: GOTO 50
8280 FOR CP=TO 9 STEP 2 ::
 CALL PATTERN(£CP,P1,£CP+1,P
3):: NEXT CP :: RETURN
8290 FOR CP=11 TO 19 STEP 2
:: CALL PATTERN(£CP,P1,£CP+1
,P3):: NEXT CP :: RETURN
8300 FOR CP=21 TO 27 STEP 2
:: CALL PATTERN(£CP,P1,£CP+1
,P3):: NEXT CP :: RETURN
8310 FOR CP=1 TO 9 STEP 2 ::
 CALL PATTERN(£CP,P,£CP+1,P2
):: NEXT CP :: RETURN
8320 FOR CP=11 TO 19 STEP 2
:: CALL PATTERN(£CP,P,£CP+1,
P2):: NEXT CP :: RETURN
8330 FOR CP=19 TO 27 STEP 2
:: CALL PATTERN(£CP,P,£CP+1,
P2):: NEXT CP :: RETURN
8340 SUB DELAY(D)
8350 FOR A=1 TO D :: NEXT A
:: SUBEND
```

**PLEASE NOTE:
Replace ALL £
symbols with
a #**

TS/SPRITE 2 by JM based on T Stevens TI*MES 41 p26.
NOTE: Replace all £ symbols with a #

```
40 ! TI*MES ISS: 41 P.26 .
SPRITE PROGRAM by T. Stevens
   SAVED AS 'TS/SPRITE2'
   by J.M. for DORTIG
45 NS=1
50 RANDOMIZE :: CALL CLEAR
60 C=3
100 FOR N=1 TO 28 :: CALL SP
RITE(£N,42,C,N*6,N*2,0,N*4):
: C=C+1
102 IF C=16 THEN C=3
105 NEXT N
106 ON NS GOTO 107,270,362
107 FOR T=1 TO 2
110 PRINT "HELLO"
112 CALL SAY("HELLO")
120 CALL DELAY(500)
130 CALL CLEAR
140 PRINT "HELLO THERE"
142 CALL SAY("HELLO+THERE")
150 CALL DELAY(500)
160 CALL CLEAR :: NEXT T
161 DISPLAY AT(23,2):"LET US
 SEE SOME COLORS?"
162 CALL SAY("I LIKE TO SEE
COLOR")
163 CALL DELAY(100):: CALL M
AGNIFY(2)
164 FOR CC=1 TO 16 :: CALL S
CREEN(CC):: CALL DELAY(40)::
 NEXT CC
165 CALL COLOR(2,2,2,3,5,16)
166 CALL CHAR(48,"26D93B843A
843A81",50,"101814327FD07E3E
",132,"001C2AFFFE42",133,"7C
FC787CFD7038",134,"060AA1238
7C261C08",135,"01060B1D3E7CF
0E0")
167 CALL CHAR(136,"0C4646FF4
6460C",137,"3844EEEE7C281038
")
168 CALL HCHAR(12,1,48,224):
: CALL HCHAR(19,1,40,160)::
CALL DELAY(200):: CALL SCREE
N(5)
170 FOR N=1 TO 28 :: CALL MO
TION(£N,0,0):: NEXT N :: CAL
L MOTION(£20,-7,0,£21,-8,0,£
22,-6,0,£23,-7,0):: CALL DEL
AY(50)
171 FOR N=20 TO 23 :: CALL M
OTION(£N,0,0):: NEXT N :: CA
LL DELAY(100)
190 FOR N=24 TO 28 :: CALL P
ATTERN(£N,132):: CALL MOTION
(£N,0,-3):: NEXT N
195 FOR N=24 TO 28 :: CALL M
OTION(£N,0,-(INT(RND*15)+9))
:: CALL MN :: NEXT N
200 FOR N=12 TO 23 :: CALL P
ATTERN(£N,50):: CALL MOTION(
£N,0,-(INT(RND*6)+1)):: NEXT
 N
204 CL=133
210 FOR N=1 TO 3 :: CALL PAT
TERN(£N,CL):: CALL COLOR(£N,
16):: CALL MOTION(£N,0,N)::
CL=CL+1 :: NEXT N
220 CALL PATTERN(£4,136):: C
ALL MOTION(£4,0,10)
230 FOR N=5 TO 11 :: CALL PA
TTERN(£N,137):: CALL MOTION(
£N,0,N-3):: NEXT N
231 DC=25
235 M$="PRESS A KEY TO END "
236 FOR LM=1 TO LEN(M$)
237 M1$=SEG$(M$,LM,1)
238 M2$=M2$&M1$
240 DISPLAY AT(24,DC):M2$
242 DC=DC-1
243 NEXT LM :: CALL DELAY(99
)
245 CALL KEY(0,K,S):: IF K>0
 THEN 260
246 DISPLAY AT(24,1):" "
248 M2$=" " :: GOTO 231
260 M2$=" " :: CALL DELSPRIT
E(ALL):: CALL CLEAR :: NS=NS
+1 :: CALL CHARSET :: GOTO 6
0
270 DISPLAY AT(8,2):"    PRE
SS ANY KEY TO STOP   SPRITES
":"FLY THROUGH ANTI-AIRCRAFT
    SHELLS": :" AND YOUR OTH
ER PLANES"
271 DISPLAY AT(15,2):"SEE HO
W MANY SURVIVIORS YOU CAN GE
T"
273 CALL KEY(0,K,S):: IF K<1
 THEN 273
280 FOR N=1 TO 28 :: CALL MO
TION(£N,0,0):: CALL DELAY(IN
T(RND*20)):: NEXT N
290 CALL TEST(1,28)
292 CALL SCREEN(5)
295 R,C=20 :: TOT,NN=0
300 FOR N=1 TO 28 :: CALL PO
SITION(£N,SR,SC)
310 IF SR=0 AND SC=0 THEN 33
0
320 CALL LOCATE(£N,R,C):: R=
R+10 :: C=C+10 :: PRINT "NUM
BER  ";N;" SURVIVED" :: NN=1
 :: TOT=TOT+1
330 NEXT N
340 IF NN=1 THEN PRINT : :"
WELL DONE ";TOT;"SURVIVIORS"
 ELSE PRINT "OH DEAR NO SURV
IVIORS"
345 IF TOTM<TOT THEN TOTM=TO
T
346 DISPLAY AT(24,3):"PLAY A
GAIN? Y/N" :: ACCEPT AT(24,2
0)VALIDATE("YN")BEEP:PA$
347 IF PA$="Y" THEN 350 ELSE
 360
350 CALL DELAY(100):: CALL C
LEAR :: GOTO 60
360 CALL CLEAR :: CALL DELSP
RITE(ALL):: NS=3 :: GOTO 60
362 CALL SCREEN(12):: CALL M
AGNIFY(2):: FOR N=3 TO 8 ::
CALL COLOR(N,16,2):: NEXT N
:: CALL CHAR(140,"0024001842
2410")
367 PRINT "     GOOD BYE FOR
 NOW": :"   NICE TO HAVE MET
 YOU": : :"YOUR MAXIMUM SURV
IVIORS WERE ";TOTM;"PLANES":
 : : :
370 CALL DELAY(200):: FOR N=
1 TO 28 :: CALL PATTERN(£N,1
40):: NEXT N
390 DISPLAY AT(3,2):"RIGHT Y
OU LOT OFF TO BED":"      SA
Y GOODBYE"
395 FOR N=1 TO 28 :: CALL SA
Y("GOODBYE"):: CALL DELSPRIT
E(£N):: CALL DELAY(30):: NEX
T N
400 CALL DELAY(200):: END
10000 SUB TEST(SC1,SC2)
10010 CALL SCREEN(2):: CALL
MAGNIFY(2):: CALL CHAR(128,"
044644FF444604",129,"A242A30
A087828")
10011 A2=12
10012 FOR SC=SC1 TO SC2 STEP
 4
10013 CALL PATTERN(£SC,128,£
SC+1,128,£SC+2,128,£SC+3,128
)
10030 FOR A=1 TO A2
10040 CALL MOTION(£SC,2,2,£S
C+1,3,3,£SC+2,4,4,£SC+3,5,5)
10045 CALL COINC(ALL,CS)
10047 IF CS=-1 THEN CALL DON
G(SC)ELSE 10050
10048 CALL DELSPRITE(£SC,£SC
+1,£SC+2,£SC+3):: GOTO 10060
10050 NEXT A :: CALL MOTION(
£SC,0,0,£SC+1,0,0,£SC+2,0,0,
£SC+3,0,0)
10060 IF SC>20 THEN A2=60
10065 NEXT SC
10070 SUBEND
10200 SUB DELAY(D)
10201 FOR DL=1 TO D :: NEXT
DL :: SUBEND
10204 SUB MN
10205 FOR F=-5 TO -7 STEP -1
 :: CALL SOUND(-99,F,0):: NE
XT F
10206 SUBEND
10210 SUB DONG(SC)
10211 FOR L=0 TO 16 :: CALL
SOUND(-99,-7,L,120,L,131,L)
10212 CALL PATTERN(£SC,129,£
SC+1,129,£SC+2,129,£SC+3,129
):: NEXT L :: SUBEND
```

£ = #

Back again this time using F/Web and trying not to make too many spelling mistakes.Must get a Spell Check program one day.
We sorted Trevors problem out, bad connection on the aerial plug at rear of the telly!!. So he's back in business on his second TI setup.
Once again, so far we haven't had the snow down this area gone colder and still raining, so we must not complain. The other program we are submitting is Terry Leaches Music 'CAVATINA' and that will get your fingers busy(we hope).
Some of the programs have emmbedded Call Load and Call Links in them so if anyone wants a copy of these drop me a line with a disk and return postage, or a pound coin, to cover cost of disk and postage and I will pass them on. I have PUTALPHA compiled by Bruce Harrision if anyone wants a copy. I had problems compiling it so Bruce sorted it out for me, A very helfull gentleman and we owe a lot to him , and others in the U.S.A. for so many good assembley programs.
Better print this and see how many mistakes I've made then get the programs printed. By the way I spotted a nice little timer program in TI*MES Issue.17 P.47 that uses a sprite and I fitted this into 'Putalpha" and it gives the time taken in seconds to complete putting the letters in place, modified by me to give time in Mins & seconds.I must dig it out and add it onto the end of this letter.

```
4000 !
4001 REM *******************
        *    TI-TIMER     *
        * BY IAN HAKANSON *
        * EXTENDED BASIC  *
        *******************
4002 REM
    THE PROGRAM DEMONSTRATES
 THE USE OF A SPRITE TO TIME
 THE PROGRAM EXECUTION AND
  PROMPT RESPONSES.
4003 REM THE TIMER IS SLOWED
 DOWN IF THERE ARE MANY
 CALCULATIONS, SQUARE ROOTS
 OR TRIG FUNCTIONS 1N THE
 PROGRAM.
4004 ! THE TIMER IS ALLWAYS
ACCURATE FOR PROMPT REPONSES
.
4005 REM TIMES UP TO A
     MAXIMUM OF 10 MINUTES
     56 SECS CAN BE READ.
4011 !CALL READ_TIMER(TIME)
4012 !DISPLAY AT(5,1):"EXCUT
ION TIME=";TIME;"SECONDS"
4017 !CALL READ_TIMER(TIME)
4020 !***** ROUTINE TO START
 TIMER.***********
4021 SUB START_TIMER
4022 CALL SPRITE(£4,143,1,25
6,256,1,1):: SUBEND
```

Replace £ with a #

In listing below,
- Replace £ with a # -

```
4023 ! *********************
*      ROUTINE TO READ TIMER
       TO NEAREST SECOND.
       *********************
*
4024 SUB READ_TIMER(T)
4025 CALL POSITION(£4,Y,X)::
 D=X-(Y+(Y>194)*32):: IF D>0
 THEN P=(224-D)/32 ELSE P=AB
S(D/32)
4026 T=INT((P*256+X)/3.12+.5
)
4027 SUBEND
```

In "PUTALPHA"
Add L 365 CALL START TIMER

```
L 440 CALL READ_TIMER(T)::
  DISPLAY AT(4,10):N$: :
  " YOU SCORED ";SC;"   POINTS"
  :      IN";INT(T/60);"mins ";
  T-(INT(T/60)*60);"secs"
*****************************
```

Next line restores colors after scoring 26 points (I did at long last)

```
Add L 935 FOR CS=1 TO 9::CALL
       COLOR(CS,16,13)::NEXT CS
       ::CALL SCREEN(5)
```

This is a repeat of L 160

PAGE 1 CAVATINA

MUSIC - CAVATINA - PAGE 1

```
10 CALL TITLE
20 CALL INIT :: T=250 :: X,Y
,Z=1 :: U=5 :: V=10 :: W=6 :
: R=40000 :: RANDOMIZE
30 LLA=110 :: LLBF=117 :: LL
B=123 :: LC=131 :: LDF=139 :
: LD=147 :: LEF=156 :: LE=16
5 :: LF=175 :: LGF=185
40 LG=196 :: LAF=208 :: LA=2
20 :: LBF=233 :: LB=247 :: C
=262 :: DF=277 :: D=294 :: E
F=311 :: F=349 :: GF=370 ::
G=392 :: AF=415 :: A=440 ::
BF=466 :: B=494
50 HC=523 :: HDF=554 :: HD=5
87 :: HEF=622
60 IF Z>1 THEN 860
70 CALL SCENE
80 CALL DEER
90 CALL MAGNIFY(3)
100 CALL SNOW
110 CALL BIRDS
120 FOR REP=1 TO 4
130 CALL SOUND(T,R,20,R,V,LE
F,W):: CALL SOUND(T,R,20,LBF
,V,LEF,W):: CALL SOUND(T,R,2
0,EF,V,LEF,W)
140 CALL SOUND(T,R,20,G,V,LE
F,W):: CALL SOUND(T,R,20,EF,
V,LEF,W):: CALL SOUND(T,R,20
,LBF,V,LEF,W)
150 NEXT REP
155 !@P-
160 GOSUB 180
170 D=D*2 :: EF=EF*2 :: F=F*
2 :: GF=GF*2 :: G=G*2 :: AF=
AF*2 :: A=A*2 :: BF=BF*2 ::
B=B*2 :: HC=HC*2 :: HD=HD*2
:: HEF=HEF*2
180 FOR REP=1 TO 2 :: CALL S
OUND(T,BF,3,LEF,W)
190 CALL SOUND(T,BF,2.5,LBF,
V,LEF,W):: CALL SOUND(T,BF,2
,EF,V,LEF,W):: CALL SOUND(T,
BF,4,G,V,LEF,W)
200 CALL SOUND(T,BF,6,EF,V,L
EF,W):: CALL SOUND(T,BF,8,LB
F,V,LEF,W):: NEXT REP
210 CALL SOUND(T/2,AF,4,LC,W
):: CALL SOUND(T/2,BF,2,LC,W
):: CALL SOUND(T,HC,0,LAF,V,
LC,W):: CALL SOUND(T,HC,0,C,
V,LC,W)
220 CALL SOUND(T,HC,0,AF,V,L
C,W):: CALL SOUND(T,HC,0,EF,
V,LC,W):: CALL SOUND(T,HC,0,
C,V,LC,W)
230 CALL SOUND(T,HC,0,LF,W):
: CALL SOUND(T,HC,0,LC,V,LF,
W):: CALL SOUND(T,HC,0,LAF,V
,LF,W):: CALL SOUND(T,HC,0,A
F,V,LF,W)
240 CALL SOUND(T,HC,0,EF,V,L
F,W):: CALL SOUND(T,HC,0,C,V
,LF,W):: CALL SOUND(T/2,AF,4
,LLBF,W):: CALL SOUND(T/2,BF
,2,LLBF,W)
250 CALL SOUND(T,HC,0,LF,V,L
LBF,W):: CALL SOUND(T,HC,0,L
AF,V,LLBF,W):: CALL SOUND(T,
HC,0,AF,V,LLBF,W)
260 CALL SOUND(T,HC,0,EF,V,L
LBF,W):: CALL SOUND(T,HC,0,C
,V,LLBF,W):: CALL SOUND(T,HC
,0,LLBF,W):: CALL SOUND(T,HC
,0,LF,V,LLBF,W)
270 CALL SOUND(T,HC,0,LAF,V,
LLBF,W):: CALL SOUND(T,HC,0,
F,V,LLBF,W):: CALL SOUND(T,H
C,0,D,V,LLBF,W)
280 CALL SOUND(T/2,BF,2,LAF,
V,LLBF,W):: CALL SOUND(T/2,A
F,4,LLBF,W):: CALL SOUND(T,B
F,0,LEF,W):: CALL SOUND(T,BF
,0,LLBF,V,LEF,W)
290 CALL SOUND(T,BF,0,LG,V,L
EF,W):: CALL SOUND(T,BF,0,G,
V,LEF,W):: CALL SOUND(T,BF,1
,D,V,LEF,W):: CALL SOUND(T,B
F,3,LBF,V,LEF,W)
300 CALL SOUND(T,BF,4,LEF,W)
:: CALL SOUND(T,BF,4,LLBF,6,
LEF,W):: CALL SOUND(T,BF,4,L
G,6,LEF,W):: CALL SOUND(T,BF
,4,G,6,LEF,W)
310 CALL SOUND(T,BF,4,DF,6,L
EF,W):: CALL SOUND(T/2,AF,3,
LBF,6,LEF,W):: CALL SOUND(T/
2,G,2,LBF,6,LEF,W)
320 CALL SOUND(T,G,1,LAF,W):
: CALL SOUND(T,G,1.5,LEF,V,L
AF,W):: CALL SOUND(T,G,2,LAF
,V,LAF,W):: CALL SOUND(T,G,4
,C,V,LAF,W)
330 CALL SOUND(T,G,6,LAF,V,L
AF,W):: CALL SOUND(T,G,8,LEF
,V,LAF,W)
340 CALL SOUND(T,F,3,LDF,W):
: CALL SOUND(T,F,2.5,LAF,V,L
DF,W):: CALL SOUND(T,F,2,LF,
V,LDF,W):: CALL SOUND(T,F,2,
C,V,LDF,W)
350 CALL SOUND(T/2,F,2,LAF,V
,LDF,W):: CALL SOUND(T/2,F,2
,LAF,V,LDF,W):: CALL SOUND(T
/2,F,2,LF,V,LDF,W):: CALL SO
UND(T/2,EF,2,LF,V,LDF,W)
360 CALL SOUND(T,F,3,LGF,W):
: CALL SOUND(T,F,2.5,LDF,V,L
GF,W):: CALL SOUND(T,F,2,LGF
,V,LGF,W):: CALL SOUND(T,F,2
,LBF,V,LGF,W)
370 CALL SOUND(T,F,2,LGF,V,L
GF,W):: CALL SOUND(T,F,2,LDF
,V,LGF,W):: CALL SOUND(T,EF,
3,LLB,W):: CALL SOUND(T,EF,2
.5,LGF,V,LLB,W)
380 CALL SOUND(T,EF,2,LEF,V,
LLB,W):: CALL SOUND(T,EF,2,L
B,V,LLB,W):: CALL SOUND(T,EF
,3,LGF,V,LLB,W):: CALL SOUND
(T,EF,4,LEF,V,LLB,W)
390 Y=Y+1
400 IF Y=5 THEN 1260
410 CALL SOUND(T,EF,3,LLBF,W
):: CALL SOUND(T,EF,2.5,LF,V
,LLBF,W):: CALL SOUND(T,EF,2
,LAF,V,LLBF,W):: CALL SOUND(
T,EF,2,LBF,V,LLBF,W)
420 CALL SOUND(T,EF,3,LAF,V,
LLBF,W):: CALL SOUND(T,EF,4,
LF,V,LLBF,W):: CALL SOUND(T,
EF,2,LLBF,W):: CALL SOUND(T,
EF,2,LF,V,LLBF,W)
430 CALL SOUND(T,F,2,LAF,V,L
LBF,W):: CALL SOUND(T,F,2,LB
F,V,LLBF,W):: CALL SOUND(T,D
,2,LAF,V,LLBF,W)
440 CALL SOUND(T,D,2,LF,V,LL
BF,W)
450 IF D=588 THEN 510
460 CALL SOUND(T,EF,3,R,V,LE
F,W):: CALL SOUND(T,EF,2.5,L
BF,V,LEF,W):: CALL SOUND(T,E
F,2,EF,V,LEF,W)
470 CALL SOUND(T,EF,4,G,V,LE
F,W):: CALL SOUND(T,EF,6,EF,
V,LEF,W):: CALL SOUND(T,EF,8
,LBF,V,LEF,W)
480 CALL SOUND(T,R,8,LEF,V,L
EF,W):: CALL SOUND(T,R,8,LBF
,V,LEF,W):: CALL SOUND(T,R,8
,EF,V,LEF,W)
490 CALL SOUND(T,R,20,G,V,LE
F,W):: CALL SOUND(T,R,20,EF,
V,LEF,W):: CALL SOUND(T,R,20
,LBF,V,LEF,W)
500 RETURN
510 CALL SOUND(T,EF,3,LAF,V,
LEF,W):: CALL SOUND(T,EF,2.5
,LF,V,LEF,W):: CALL SOUND(T,
EF,2,LG,V,LEF,W)
520 CALL SOUND(T,EF,4,LEF,V,
LEF,W):: CALL SOUND(T,EF,U,L
G,V,LEF,W):: CALL SOUND(T,EF
,6,LBF,8,LEF,W)
530 CALL SOUND(T,EF,9):: CAL
L SOUND(T,LBF,9):: CALL SOUN
D(T,EF,W):: CALL SOUND(T,G,5
):: CALL SOUND(T,BF,3):: CAL
L SOUND(T,D,2)
540 CALL SOUND(T,HEF,2,LC,W)
:: CALL SOUND(T,HEF,1.5,LBF,
V,LC,W):: CALL SOUND(T,HEF,1
,EF/2,V,LC,W):: CALL SOUND(T
,HEF,1,G/2,V,LC,W)
550 CALL SOUND(T,HEF,2,EF/2,
V,LC,W):: CALL SOUND(T,HEF,4
,LBF,V,LC,W)
560 CALL SOUND(T,HEF,2,LF,W)
:: CALL SOUND(T,HEF,1.5,LBF,
V,LF,W):: CALL SOUND(T,HEF,1
,EF/2,V,LF,W):: CALL SOUND(T
,HEF,1,G/2,V,LF,W)
570 CALL SOUND(T,HEF,1,EF/2,
V,LF,W):: CALL SOUND(T/2,HD,
2,LBF,V,LF,W):: CALL SOUND(T
/2,HC,2,LBF,V,LF,W)
580 CALL SOUND(T,HC,2,LLBF,W
):: CALL SOUND(T/2,HD,1.5,LF
,V,LLBF,W):: CALL SOUND(T/2,
HEF,1,LF,V,LLBF,W):: CALL SO
UND(T,HD,0,LBF,V,LLBF,W)
590 CALL SOUND(T,HD,0,F/2,V,
LLBF,W):: CALL SOUND(T,HD,0,
D/2,V,LLBF,W):: CALL SOUND(T
,HD,0,LBF,V,LLBF,W)
600 CALL SOUND(T,HD,0,LLBF,W
):: CALL SOUND(T,HD,0,LF,V,L
LBF,W):: CALL SOUND(T,HD,0,L
A,V,LLBF,W):: CALL SOUND(T,H
D,0,F/2,V,LLBF,W)
610 CALL SOUND(T,HC,1,D/2,V,
LLBF,W):: CALL SOUND(T,BF,1.
5,LBF,V,LLBF,W):: CALL SOUND
(T,AF,0,LF,W):: CALL SOUND(T
,AF,0,LC,V,LF,W)
620 CALL SOUND(T,AF,0,LBF,V,
LF,W):: CALL SOUND(T,AF,1,EF
/2,V,LF,W):: CALL SOUND(T,AF
,1.5,C/2,V,LF,W)
630 CALL SOUND(T,AF,2,LBF,V,
LF,W)
640 CALL SOUND(T,AF,1,LLBF,W
):: CALL SOUND(T,AF,.5,LF,V,
LLBF,W):: CALL SOUND(T,BF,0,
LBF,V,LLBF,W)
650 CALL SOUND(T,BF,0,D/2,V,
LLBF,W)
660 CALL SOUND(T,F,.5,LBF,V,
LLBF,W):: CALL SOUND(T,F,1,L
F,V,LLBF,W)
670 CALL SOUND(T,G,0,LEF,W):
: CALL SOUND(T,G,0,LLBF,V,LE
F,W):: CALL SOUND(T,G,0,LG,V
,LEF,W):: CALL SOUND(T,G,2,D
/2,V,LEF,W)
680 CALL SOUND(T,G,3,LBF,V,L
EF,W):: CALL SOUND(T,G,4,LG,
V,LEF,W):: CALL SOUND(T,G,4,
LEF,W):: CALL SOUND(T,G,4,LL
BF,V,LEF,W)
690 CALL SOUND(T,G,4,LG,V,LE
F,W):: CALL SOUND(T,G,3,DF,V
,LEF,W):: CALL SOUND(T,EF,2,
LBF,V,LEF,W):: CALL SOUND(T,
F,1,LG,V,LEF,W)
700 REM TO CODA
710 X=X+1
720 IF X=3 THEN 320 ELSE 730
730 CALL SOUND(T,G,2,LG*2,V,
LAF,W):: CALL SOUND(T,G,1.5,
LEF,V,LAF,W):: CALL SOUND(T,
G,1,LG*2,V,LAF,W)
740 CALL SOUND(T,G,1,C,9,LAF
,W):: CALL SOUND(T,G,1,LG*2,
8,LAF,W):: CALL SOUND(T,G,1,
LEF,8,LAF,W)
750 CALL SOUND(T,AF,2,LAF*2,
W,LDF,W):: CALL SOUND(T,AF,1
.5,LG,W,LDF,W):: CALL SOUND(
T,AF,1,LF,W,LDF,W)
760 CALL SOUND(T,AF,1,C,6.5,
LDF,W):: CALL SOUND(T,AF,1,L
F,6.5,LDF,W):: CALL SOUND(T,
AF,1,LG,6.5,LDF,W)
770 CALL SOUND(T,A,2,LA*2,6,
LF,W):: CALL SOUND(T,A,1.5,L
C,6,LF,W):: CALL SOUND(T,A,1
,LA,6,LF,W)
780 CALL SOUND(T,A,1,F/2,5.5
,LF,W):: CALL SOUND(T,A,1,C,
5.5,LF,W):: CALL SOUND(T,A,1
,LA,5.5,LF,W)
790 CALL SOUND(T,BF,2,LBF*2,
U,LD,W):: CALL SOUND(T,BF,1.
5,LLBF,U,LD,W):: CALL SOUND(
T,BF,1,LF,U,LD,W)
800 CALL SOUND(T,BF,1,F/2,5,
LD,W):: CALL SOUND(T,BF,1,LB
F,U,LD,W):: CALL SOUND(T,BF,
1,LF,U,LD,W)
810 CALL SOUND(T,BF,2,LBF*2,
4.5,LDF,W):: CALL SOUND(T,BF
,1.5,LLBF,4.5,LDF,W):: CALL
SOUND(T,BF,1,LF,4.5,LDF,W)
820 CALL SOUND(T,BF,1,F/2,4,
LDF,W):: CALL SOUND(T,BF,1,L
BF,4,LDF,W):: CALL SOUND(T,B
F,1,LF,4,LDF,W)
830 CALL SOUND(T,HEF,2,EF,3.
5,LLB,W):: CALL SOUND(T,HEF,
1.5,LGF,3.5,LLB,W):: CALL SO
UND(T,HEF,1,LB,3.5,LLB,W)
840 CALL SOUND(T,HEF,1,F/2,3
,LLB,W):: CALL SOUND(T,HEF,1
,LB,3,LLB,W):: CALL SOUND(T,
HEF,1,LGF,3,LLB,W)
850 Z=Z+1 :: GOTO 30
860 CALL SOUND(T,HEF,1,AF,V,
LLBF,W):: CALL SOUND(T,B,1,A
F,V,LLBF,W):: CALL SOUND(T,B
,1,AF,V,LLBF,W):: CALL SOUND
(T,HDF,1,AF,V,LLBF,W)
870 CALL SOUND(T,HDF,1,AF,V,
LLBF,W):: CALL SOUND(T,BF,1,
AF,V,LLBF,W)
880 CALL SOUND(T,BF,1,AF,V,L
LBF,W):: CALL SOUND(T,B,2,AF
,V,LLBF,W):: CALL SOUND(T,B,
2):: CALL SOUND(T,AF,2):: CA
LL SOUND(T,BF,2)
890 CALL SOUND(T,B,2)
900 CALL SOUND(T,EF,2):: CAL
L SOUND(T,LB,2):: CALL SOUND
(T,LB,2):: CALL SOUND(T,DF,2
):: CALL SOUND(T,DF,2):: CAL
L SOUND(T,LBF,2)
910 CALL SOUND(T,LBF,2):: CA
LL SOUND(T,LB,2):: CALL SOUN
D(T,LAF,2):: CALL SOUND(T,LF
,2):: CALL SOUND(T,LEF,2)::
CALL SOUND(T,LLB,2)
920 CALL SOUND(T,LLBF,2):: C
ALL SOUND(T,LAF,3):: CALL SO
UND(T,LGF,4):: CALL SOUND(T,
LF,U):: CALL SOUND(T,LGF,6):
: CALL SOUND(T,LAF,W)
930 CALL SOUND(T,R,20,R,20,L
LA,U):: CALL SOUND(T,R,20,LE
F,V,LLA,U):: CALL SOUND(T,R,
20,C,V,LLA,U):: CALL SOUND(T
,R,20,F,V,LLA,U)
940 CALL SOUND(T,R,20,C,V,LL
A,U):: CALL SOUND(T,R,20,LEF
,V,LLA,U)
950 CALL SOUND(T,R,20,LLBF,U
):: CALL SOUND(T,LF,3,LLBF,U
):: CALL SOUND(T,LBF,3,LLBF,
U):: CALL SOUND(T,D,3,LF,U)
960 CALL SOUND(T,LBF,3,LLBF,
U):: CALL SOUND(T,LF,3,LC,U)
970 CALL SOUND(T,R,20,LDF,V,
LG,U):: CALL SOUND(T,LE,3,LD
F,V,LG,U):: CALL SOUND(T,LA,
3,LBF,V,LG,U):: CALL SOUND(T
,DF,3,LDF,V,LG,U)
980 CALL SOUND(T,LA,3,LDF,V,
LG,U):: CALL SOUND(T,LE,3,LD
F,V,LG,U)
990 CALL SOUND(T,R,20,LE,V,L
F,U):: CALL SOUND(T,LA,3,LE,
V,LF,U):: CALL SOUND(T,D,3,L
D,V,LF,U):: CALL SOUND(T,F,3
,LAF,V,LF,U)
1000 CALL SOUND(T,D,3,LD,V,L
F,U):: CALL SOUND(T,LA,3,LE,
V,LF,U)
1010 CALL SOUND(T,R,20,LGF,V
,LC,U):: CALL SOUND(T,LA,3,L
GF,V,LC,U):: CALL SOUND(T,D,
3,LGF,V,LC,U):: CALL SOUND(T
,GF,3,LGF,V,LC,U)
1020 CALL SOUND(T,D,3,LGF,V,
LC,U):: CALL SOUND(T,LA,3,LG
F,V,LC,U)
1030 CALL SOUND(T,R,20,LA,V,
LLBF,U):: CALL SOUND(T,D,3,L
A,V,LLBF,U):: CALL SOUND(T,G
,3,LG,V,LLBF,U):: CALL SOUND
(T,BF,3,LD,V,LLBF,U)
1040 CALL SOUND(T,G,3,LG,V,L
LBF,U):: CALL SOUND(T,D,3,LA
,V,LLBF,U)
1050 CALL SOUND(T,LBF,U,LG,V
):: CALL SOUND(T,LBF,U,LEF,V
):: CALL SOUND(T,LBF,U,LBF,V
):: CALL SOUND(T,LBF,U,HEF,V
)
1060 CALL SOUND(T,LBF,U,BF,V
):: CALL SOUND(T,LBF,U,EF,V)
1070 CALL SOUND(T,LBF,U,LG,V
):: CALL SOUND(T,LBF,U,LEF,V
):: CALL SOUND(T,LBF,U,LBF,V
```

```
):: CALL SOUND(T,LBF,U,HEF,V
)
1080 CALL SOUND(T/2,LBF,U,BF
,V):: CALL SOUND(T/2,LBF,U,B
F,V):: CALL SOUND(T/2,LBF,U,
EF,V):: CALL SOUND(T/2,C,U,E
F,V)
1090 CALL SOUND(T,LBF,U,LF,V
):: CALL SOUND(T,LBF,U,LD,V)
:: CALL SOUND(T,LBF,U,LBF,V)
:: CALL SOUND(T,LBF,U,HD,V)
1100 CALL SOUND(T,LBF,U,AF,V
):: CALL SOUND(T,LBF,U,D,V)
1110 CALL SOUND(T,LBF,U,LF,V
):: CALL SOUND(T,LBF,U,LD,V)
:: CALL SOUND(T,LBF,U,LBF,V)
:: CALL SOUND(T,LBF,U,HD,V)
1120 CALL SOUND(T/2,LBF,U,AF
,V):: CALL SOUND(T/2,LBF,U,A
F,V):: CALL SOUND(T/2,LBF,U,
D,V):: CALL SOUND(T/2,C,U,D,
V)
1130 CALL SOUND(T,LBF,U,LEF,
V):: CALL SOUND(T,LBF,U,LLBF
,V):: CALL SOUND(T,LBF,U,LG,
V):: CALL SOUND(T,LBF,U,HDF,
U)
1140 CALL SOUND(T,LBF,U,G,4)
:: CALL SOUND(T,LBF,U,EF,3)
1150 CALL SOUND(T,EF,3,LBF,V
,LAF,U):: CALL SOUND(T,EF,3,
C,V,LAF,U):: CALL SOUND(T,EF
,3,LAF,V,LAF,U):: CALL SOUND
(T,EF,3,LG,V,LAF,U)
1160 CALL SOUND(T,EF,3,LF,V,
LAF,U):: CALL SOUND(T,EF,3,L
EF,V,LAF,U)
1170 CALL SOUND(T,R,20,LEF,V
,LF,U):: CALL SOUND(T,LAF,3,
LEF,V,LF,U):: CALL SOUND(T,L
BF,3,LEF,V,LF,U):: CALL SOUN
D(T,EF,3,LEF,V,LF,U)
1180 CALL SOUND(T,LBF,3,LEF,
V,LF,U):: CALL SOUND(T,LAF,3
,LEF,V,LF,U)
1190 CALL SOUND(T,R,20,LEF,V
,LLBF,U):: CALL SOUND(T,LAF,
3,LEF,V,LLBF,U):: CALL SOUND
(T,LBF,3,LF,V,LLBF,U):: CALL
 SOUND(T,D,3,LF,V,LLBF,U)
1200 CALL SOUND(T,LBF,3,LD,V
,LLBF,U):: CALL SOUND(T,LAF,
3,LD,V,LLBF,U)
1210 CALL SOUND(T,LAF,3,LEF,
V,LEF,U):: CALL SOUND(T,LF,3
```

```
,LEF,V,LEF,U):: CALL SOUND(T
,LAF,3,LEF,V,LEF,U):: CALL S
OUND(T,LD,3,LEF,V,LEF,U)
1220 CALL SOUND(T,LEF,3,LEF,
V,LEF,U):: CALL SOUND(T,LAF,
3,LEF,V,LEF,U)
1230 CALL SOUND(T,LBF*2,7)::
 CALL SOUND(T,LG*2,7):: CALL
 SOUND(T,LBF*2,6):: CALL SOU
ND(T,EF*2,U):: CALL SOUND(T,
G*2,4)
1240 CALL SOUND(T,AF*2,3)
1250 GOTO 170
1260 CALL SOUND(T,EF,3,LF,W)
:: CALL SOUND(T,EF,3,LC,V,LF
,W):: CALL SOUND(T,EF,3,LAF,
V,LF,W):: CALL SOUND(T,EF,3,
EF/2,V,LF,W)
1270 CALL SOUND(T,EF,3,C,V,L
F,W):: CALL SOUND(T,EF,3,LAF
,V,LF,W)
1280 CALL SOUND(T,EF,3,R,20,
LLBF,W):: CALL SOUND(T,EF,3,
LF,V,LLBF,W):: CALL SOUND(T,
F,3,LAF,V,LLBF,W)
1290 CALL SOUND(T,F,3,LBF,V,
LLBF,W):: CALL SOUND(T,D,3,L
AF,V,LLBF,W):: CALL SOUND(T,
D,3,LF,V,LLBF,W)
1300 CALL SOUND(T,EF,3,LAF,V
,LEF,W):: CALL SOUND(T,EF,3,
LF,V,LEF,W):: CALL SOUND(T,E
F,3,LG,V,LEF,W)
1310 CALL SOUND(T,EF,3,LD,V,
LEF,W):: CALL SOUND(T,EF,3,L
EF,V,LEF,W):: CALL SOUND(T,E
F,3,LG,V,LEF,W)
1320 CALL SOUND(T,EF,3,LBF,V
,LEF,W):: CALL SOUND(T,R,20,
LG,8):: CALL SOUND(T,R,20,LB
F,W):: CALL SOUND(T,R,20,EF,
6)
1330 CALL SOUND(T,R,20,G,U):
: CALL SOUND(T,R,20,BF,4)
1340 CALL SOUND(T*2,HD,3,LBF
,V,LC,W):: CALL SOUND(T,HEF,
2.5,LBF,V,LC,W)
1350 CALL SOUND(T,HEF,2,LBF,
V,LC,W):: CALL SOUND(T,HEF,3
,LBF,V,LC,W):: CALL SOUND(T,
HEF,3,LBF,V,LC,W)
1360 CALL SOUND(T,D,2.5,LAF,
V,LF,W):: CALL SOUND(T,D,2,L
AF,V,LF,W):: CALL SOUND(T,EF
,2,LAF,V,LF,W):: CALL SOUND(
T,EF,2.5,LAF,V,LF,W)
1370 CALL SOUND(T,EF,3,LAF,V
,LF,W):: CALL SOUND(T,EF,3,L
AF,V,LF,W)
1380 T=400 :: CALL SOUND(T,E
F,2,C,V,LLBF,W):: CALL SOUND
(T/2,F,3,C,V,LLBF,W):: CALL
SOUND(T/2,G,3,C,V,LLBF,W)
1390 CALL SOUND(T,F,3,C,V,LL
BF,W):: CALL SOUND(T,EF,3,C,
V,LLBF,W):: CALL SOUND(T*2,E
F,3,LAF,V,LEF,W):: CALL SOUN
D(T*4,EF,3,LBF,V,LEF,W)
1395 !@P+
1400 CALL DELSPRITE(ALL):: C
ALL CLEAR :: CALL CHARSET ::
 CALL SCREEN(12):: FOR R=1 T
O 14 :: CALL COLOR(R,13,12):
: NEXT R
1410 DISPLAY AT(8,10):"CAVAT
INA" :: DISPLAY AT(12,1):"HE
AR AGAIN ??? <PRESS ENTER>"
:: DISPLAY AT(24,1):"PRESS A
NY OTHER KEY TO QUIT!"
1420 CALL KEY(0,KEY,S)
1430 IF S=0 THEN 1420 :: IF
KEY=13 THEN 20 ELSE CALL SAY
("GOODBYE"):: CALL CLEAR ::
END
1435 1430
1438 !@P+
1440 SUB TITLE
1450 CALL CLEAR :: CALL CHAR
SET :: CALL COLOR(2,2,1):: C
ALL SCREEN(12)
1460 RESTORE 1490 :: FOR L=4
0 TO 47 :: READ B$ :: CALL C
HAR(L,B$):: NEXT L
1470 CALL HCHAR(1,1,40):: CA
LL HCHAR(1,2,41,30):: CALL H
CHAR(1,32,42):: CALL VCHAR(2
,32,43,22)
1480 CALL HCHAR(24,32,44)::
CALL VCHAR(2,1,45,22):: CALL
 HCHAR(24,1,46):: CALL HCHAR
(24,2,47,30)
1490 DATA "FFFFC0DFD0D7D7D7"
1500 DATA "FFFF00FF00FFFFFF"
1510 DATA "FFFF03FB0BEBEBEB"
1520 DATA "1B1B1B1B1B1B1B1B"
1530 DATA "EBEBEB0BFB03FFFF"
1540 DATA "D7D7D7D7D7D7D7D7"
1550 DATA "D7D7D7D0DFC0FFFF"
1560 DATA "FFFFFF00FF00FFFF"
1570 FOR J=3 TO 14 :: CALL C
OLOR(J,13,1):: NEXT J :: CAL
L SCREEN(12)
1580 FOR J=1 TO 26 :: CALL C
HARPAT(64+J,A$):: A$=SEG$(A$
,3,14)&"FF" :: CALL CHAR(96+
J,A$):: NEXT J
1590 RESTORE 1630 :: FOR I=4
 TO 19 STEP 3 :: READ A$ ::
FOR H=2 TO LEN(A$):: CALL SP
RITE(£1,ASC(SEG$(A$,H,1)),16
,1,H*8-7,I*5,0)
1600 IF ASC(SEG$(A$,H,1))=32
 THEN CALL SOUND(1,-1,7)
1610 CALL COINC(£1,I*8-7,H*8
-7,I*4,C):: IF C=-1 THEN CAL
L HCHAR(I,H,ASC(SEG$(A$,H,1)
))ELSE 1610
1620 NEXT H :: NEXT I :: CAL
L DELSPRITE(ALL)
1630 DATA "        C A V A T
 I N A "
1640 DATA "   THEME FROM THE
 DEER HUNTER "
1650 DATA "            FROM T
HE "
1660 DATA "   JOHN WILLIAMS
ARRANGEMENT "
1670 DATA "   TRANSPOSED FOR
 THE TEXAS "
1680 DATA "          BY T J
LEACH "
1690 SUBEND
1694 !@P-
1698 !@P+
1700 SUB SCENE
1710 CALL SCREEN(2):: CALL C
LEAR :: CALL CHARSET :: CALL
 COLOR(2,14,1,3,5,1,4,6,14,5
,13,14,7,16,5,8,9,5,10,13,5)
1720 A$="00070F1F1F3F3F3F3F3
F3F1F1F0F070000E0F0F8F8FCFCF
CFCFCFCF8F8F0E0"
1730 B$=RPT$("FF",8):: C$="9
1B9387E387DFC7E" :: D$="FFFF
FE7C3D19181C" :: E$="0000000
0FFFFFFFF" :: F$="0103070F1F
37F7FFF" :: G$="80C0E0F0F8FC
FEFF"
1740 L$="90CAD66D5AF7FFFF" :
: M$="22924B66F7FEFFFF" :: N
$=RPT$("55AA",4):: P$="5B37A
E59AA27C965" :: Q$="B1E1C5D4
B4B5D1A1"
1750 H$="0010000000000000" :
: CALL CHAR(40,B$,56,B$,64,C
$,65,D$,57,E$,104,F$,105,G$,
80,H$,48,B$,88,A$,67,L$,68,M
$,107,N$,66,Q$,106,P$)
1760 N=5
1770 FOR R=16 TO 24 :: CALL
HCHAR(R,1,40,32):: NEXT R ::
 FOR R=16 TO 22 :: CALL HCHA
R(R,R-12,56,N):: N=N+3 :: NE
XT R
1780 FOR R=12 TO 13 :: CALL
HCHAR(R,1,64,32):: NEXT R ::
 CALL HCHAR(15,1,57,32):: CA
LL HCHAR(14,1,65,32):: FOR R
=1 TO 11 :: CALL HCHAR(R,1,4
8,32)
1790 NEXT R :: N=11 :: FOR R
=7 TO 11 :: CALL HCHAR(R,15-
R,64,N):: N=N+3 :: NEXT R ::
 FOR R=7 TO 11 :: CALL HCHAR
(R,14-R,104,1):: NEXT R
1800 N=1 :: FOR R=7 TO 11 ::
 CALL HCHAR(R,11+R+N,105,1):
: N=N+1 :: NEXT R :: FOR L=1
 TO 25 :: J=1+INT(RND*26)::
K=1+INT(RND*6):: CALL HCHAR(
K,3+J,80,1):: NEXT L
1810 FOR C=1 TO 2 :: FOR R=1
 TO 2 :: CALL HCHAR(R+3,C+4,
85+(C*2)+R,1):: NEXT R :: NE
XT C
1820 CALL HCHAR(24,1,1+INT(R
ND*2)+66,32)
1830 N=1 :: FOR R=1 TO 2 ::
CALL HCHAR(R,20+R+N,107,6)::
 N=N+1 :: NEXT R :: FOR C=1
TO 2 :: CALL HCHAR(4,23+C,10
6,1):: NEXT C
1840 FOR R=1 TO 2 :: CALL HC
HAR(3+R,27,106,1):: NEXT R :
: FOR R=1 TO 3 :: CALL HCHAR
(3+R,31,107,2):: NEXT R :: C
ALL HCHAR(7,31,106,2)
1850 CALL VCHAR(8,32,66,17):
: CALL VCHAR(4,1,66,20)
1860 N=1 :: CALL HCHAR(N,N,1
06,N*3):: CALL HCHAR(N,7,106
,N*3):: CALL VCHAR(N*2,N,106
,N*2)
1870 CALL HCHAR(2*N,3*N,106,
N):: CALL HCHAR(N,4,106,N)::
 FOR R=1 TO 3 :: CALL HCHAR(
R,18+R+N,106,1):: N=N+1 :: N
EXT R
1880 CALL VCHAR(4,29,106,1):
: CALL VCHAR(4,30,106,3):: C
```

NOTE: Instead of £ use #

Cavatina - continued

```
ALL HCHAR(3,25,106,4):: CALL
 HCHAR(3,29,107,3)
1890 CALL HCHAR(1,21,106,1):
: CALL HCHAR(2,23,106,1):: N
=1 :: FOR R=1 TO 3 :: CALL H
CHAR(R,26+N+R,106,7-N-R):: N
=N+1 :: NEXT R
1900 SUBEND
1902 !@P-
1908 !@P+
1910 SUB DEER
1920 A$="171F1E0C0C0C1C183
A3A1E0E070707000000000000000
00000000000000B0F0"
1930 B$="0002060E0F070301000
00000000010210000004850F0E0F
8F0E0C8D4E8D0F0E0"
1940 C$="0100000000000000000
000000000000E0F17BDFFFFFFFF
FFF7F3F1F1F1F1F1F"
1950 D$="41C3C78F9FBFFFFFFFF
FFEFCFCFEFEFCC080E0E0C080B08
00000000000000F1F"
1960 E$="0000000000000000000
000000000FCFF000000000000000
0000000000000000000"
1970 F$="0000000000000000000
000000000000000F0F0F1F1F1F1F1
F1F1F0F0F07070303"
1980 G$="FFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFF"
1990 H$="FFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFC0E0F0F8F8F8F8F
8F8F8F0F0E0E0C080"
2000 J$="FFFF7F7F7F3F3D3D3D1
D1D1D1D1D0D0DFFF0C0C0C0C0C0C
0C0C0C0C0C0808080"
2010 K$="F7F77B3938383830303
03030606060C080C0C0E0E0E0E0C
0C0C0C0C0C0C0C0C0"
2020 CALL CHAR(96,A$,100,B$,
112,C$,116,D$)
2030 CALL CHAR(120,E$,124,F$
,128,G$,132,H$)
2040 CALL CHAR(136,J$,140,K$
)
2050 CALL SPRITE(£1,96,2,121
,169,£2,100,2,121,185,£3,112
,2,137,169,£4,116,2,137,185)
2060 CALL SPRITE(£5,120,2,13
7,201,£6,124,2,153,169,£7,12
8,2,153,185,£8,132,2,153,201
)
2070 CALL SPRITE(£9,136,2,16
9,185,£10,140,2,169,201)
2080 SUBEND
2082 !@P-
2088 !@P+
2090 SUB SNOW
2100 A$="1"&RPT$("0",48)
2110 CALL CHAR(32,A$)
2120 FOR R=13 TO 28 :: RANDO
MIZE :: CALL PEEK(-31808,A,B
):: RANDOMIZE :: CALL PEEK(-
31808,C,D):: C=3+INT(C/20)::
 D=INT(D/50):: A=A+1 :: B=B+
1 :: IF A+B<256 THEN D=-D
2130 CALL SPRITE(£R,32,16,A,
B,C,D):: NEXT R
2140 SUBEND
2142 !@P-
2148 !@P+
2150 SUB BIRDS
2160 A$="0041221408004200885
1200040110A04000542000088502
00400004122140800"
2170 B$="0000007708002100000D
8200020001B0400002700000082
00280000000770800"
2180 CALL CHAR(72,A$,76,B$)
2190 CALL SPRITE(£11,72,2,25
,240,0,-3):: CALL SPRITE(£12
,72,2,29,224,0,-3):: CALL LO
AD(-31878,28)
2200 FOR R=1 TO 60 :: CALL P
ATTERN(£11,76):: CALL PATTER
N(£12,76):: CALL PATTERN(£11
,72):: CALL PATTERN(£12,72):
: FOR D=1 TO 50 :: NEXT D
2210 NEXT R
2220 SUBEND
2222 !@P-
```

END

£ = #

Copied from Sydney News Digest of January 1991:

THE GOLDEN RATIO

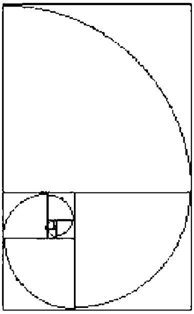
```
1 ! THE GOLDEN RATIO   -   1: 1.618...
2 ! removing a square from a rectangle with sides in
  this ratio leaves a smaller rectangle which also has
  sides in the Golden Ratio
3 ! -points dividing sides lie on a logarithmic spiral
  which can be found in shells, and in art works by
  da Vinci, Dali- even in
4 ! the Parthenon.
5 ! Fractal in nature- it keeps getting smaller or
  larger depending on which way you go...
6 ! program written by Ashley Tilling for JBM103
  converted by S Shaw for TML.
7 !
8 ! Due to our not having square pixels the rectangle
  on screen is NOT in the golden ratio as viewed.
  Ah well...
9 !
99 ! remove ! from line 100 for use with JBM103.
100 ! call load(-31890,56,0) :: CALL LOAD(-31964,56,0)
110 AX=12 :: AY=8
120 CL=200 ! side length
130 CALL LINK("CLEAR")!for jbm103 CALL LINK("SCR2") also.
140 RL=INT(CL/1.618)
150 FOR I=0 TO 1 :: FOR J=0 TO 1
160 CALL LINK("LINE",AX+RL*I,AY+CL*I,AX+RL*J,AY-CL*(J=0))
162 ! CALL LINK("LIGNE",16,AX+RL*I,AY+CL*I,AX+RL*J,AY-
    CL*(J=0))
170 NEXT J :: NEXT I
180 FOR K=1 TO 3
190 M=CL-RL :: N=INT(CL*.236):: P=INT(CL*.146)
195 CALL LINE(AX,AY,AX,AY)
200 CALL LINE(RL,M,1,M)
210 CALL LINE(N,1,N,M)
220 CALL LINE(1,N,N,N)
230 CALL LINE(P,M,P,N)
240 AX=AX+P :: AY=AY+N :: CL=P :: RL=INT(CL/1.618)
250 NEXT K
260 GOTO 260
270 SUB LINE(A,B,C,D)
280 REM ! LINES for JBM103 INSTEAD of CALL LINK(
    "LINE".. and CALL LINK("PIXEL"...
290 IF (A=C)*(B=D)THEN AX=A :: AY=B :: SUBEXIT
300 CALL LINK("LINE",AX+A,AY+B,AX+C,AY+D)
310 ! CALL LINK("LIGNE",16,AX+A,AY+B,AX+X,AY+D)
320 IF B=D THEN 350
330 L=ABS(B-D)-1
340 ST=-PI*(B>D):: FI=PI/2+PI*(D>B):: GOTO 370
350 L=ABS(A-C)
360 ST=-PI/2-PI*(A>C):: FI=PI*(C>A)
370 IF L<3 THEN SUBEXIT
380 FOR J=ST TO FI STEP -1/L
390 Y=AY+D+INT(.5+L*SIN(J))
400 X=AX+C+INT(.5+L*COS(J))
410 CALL LINK("PIXEL",X,Y)
420 ! CALL LINK("POINT",16,X,Y)
430 NEXT J
440 SUBEND
450 END
```

NOTE: REQUIRES " THE MISSING LINK" & 32K RAM

Also for jbm103

(Move the ! from 162 to 160 for jbm103)

(Move the ! from 310 to 300 and from 420 to 410 for jbm103)



THE TI99/4A AND ADVANCED MATHEMATICAL CALCULATIONS

The title sounds like a contradiction in terms. Nobody has ever tried to claim the TI is some kind of superfast computer. to my knowledge, and I am not about to make such a claim here. The truth is that there are a number of mathematical algorithms which do not require a great deal of speed, but which can still be classed among the ones used for advanced mathematical calculations.

Most of us are familiar with X-Y graphs for representing functions on paper or on a computer screen. For instance. **Y=sin(X)** gives the well known periodic curve called a **sinusoid.** and **Y=exp(-X)** gives the quickly decaying curve for X>0 and fast rising curve for X<0 called an **exponential.**
There is, however, another very important class of equations which relate the rates of change with X of various (X dependent) quantities to X and Y: these are known as **differential equations.** Typical examples are:

$$\frac{dY}{dX} + aY = K \tag{1}$$

$$\frac{d^2Y}{dX^2} + w^2Y = 0 \tag{2}$$

$$\frac{d^2Y}{dX^2} + w^2\mathrm{SIN}(Y) = 0 \tag{3}$$

where equation 1 represents the voltage curve for a charging capacitor, equation 2 a body oscillating with Simple Harmonic Motion (such as a weight hanging on a spring). and equation 3 represents a Simple Pendulum (such as the ones in grandfather clocks). In all these examples the X **variable**, known as the independent variable, represents **time**.
Finding a solution of the above means obtaining a function **Y(X)** (this means Y which is a function of X. **not Y*X**) for each of the three equations such that, on substitution of this function into each of them, the left-hand-side = the right-hand-side for any value of X.
The first two equations have simple solutions:

$$Y = (Y_0 - \frac{K}{a}) * \mathrm{EXP}(-aX) + \frac{K}{a} \tag{4}$$

$$Y = \frac{Y_0}{\mathrm{SIN}[\mathrm{TAN}^{-1}(\frac{Y_0 w}{Y_0'})]} * \mathrm{SIN}[wX+\mathrm{TAN}^{-1}(\frac{Y_0 w}{Y_0'})] \tag{5}$$

where Y_0 and Y'_0 represent the values of Y(X) and Y'(X) respectively for X=0, and are known as the initial conditions. If the order of the differential equation is 2 (as in the above). Y_0 and Y'_0 must be specified as initial conditions. if 3 then Y_0, Y'_0,and Y''_0 are required etc..

Equation 3 has no simple solution as the above and only a numerical evaluation is possible. The program I am presenting here can do exactly this. In fact, it can do more: it can solve any linear differential equation of up to tenth order, despite being written in TI BASIC. Even console only users can run it effectively, though saving the data file to tape can be somewhat tedious. Such a data file can subsequently be plotted by any plotting routine, either written especially or commercially available.

```
1 REM Kutta-Simpson algorithm
2 REM    by Francesco Lama
9 CALL CLEAR
10 DIM YD(10).YD0(10).K1(10).
K2(10).K3(10).K4(10).Y(10.110)
20 INPUT "ORDER OF DIFFERENTIAL
EQUATION?":NEQ
30 INPUT "INITIAL VALUE OF
INDEPENDENT VARIABLE?":XD
40 X0=XD
45 PRINT
60 PRINT "INITIAL CONDITIONS"
70 FOR I=0 TO NEQ-1
80 PRINT "INITIAL VALUE OF
DERIVATIVE OF ORDER";I
90 INPUT YD(I)
100 NEXT I
110 PRINT
120 INPUT "FINAL VALUE OF THE
INDEPENDENT VARIABLE?":XF
130 INPUT "NUMBER OF INTERVALS OF
INTEGRATION?":NINT
135 REM it is important to choose
NS carefully in order not
overflow the Y array.
136 REM in practice NINT/NS must
be less than 110.
140 INPUT "EVERY HOW MANY
INTERVALS    SHOULD A POINT BE
STORED?":NS
150 INPUT "EVERY HOW MANY
INTERVALS    SHOULD A POINT BE
DISPLYED?":ND
160 H=(XF-X0)/NINT
170 K=0
180 FOR J=1 TO NEQ-1
190 Y(J,K)=YD(J)
200 NEXT J
201 GOSUB 6000
202 YD(NEQ)=F
203 Y(NEQ,K)=YD(NEQ)
204 REM these print statements
should suit individual require
ments as in lines 281-309
205 PRINT TAB(6):"X":TAB(21):"
Y0"
206 PRINT TAB(5);X0;TAB(20):YD
(0)
207 PRINT TAB(6);"Y1";TAB(21):
"Y2"
208 PRINT TAB(5);YD(1):TAB(20)
;YD(2)
209 PRINT
219 FOR K=1 TO NINT
220 GOSUB 5000
230 IF INT(K/NS)<>K/NS THEN 270
240 FOR J=0 TO NEQ
250 Y(J.K/NS)=YD(J)
260 NEXT J
270 IF INT(K/ND)<>K/ND THEN 310
280 REM these PRINT statements
should suit individual require
ments as in lines 205-218
281 PRINT TAB(6);"X":TAB(21):"Y0"
282 PRINT TAB(1):X0+K*H:TAB(16)
:YD(0)
283 PRINT TAB(6):"Y1":TAB(16):
"Y2"
284 PRINT TAB(1):YD(1):TAB(16):
YD(2)
285 PRINT
310 NEXT K
311 PRINT
312 PRINT "INITIAL X=":X0
313 PRINT "FINAL X=":XF
314 PRINT "INCREMENT=":H
315 PRINT "INCREMENT IN STORED
DATA=";H*NS
316 PRINT "INCREMENT IN DISPLAY
ED      DATA=";H*ND
```

```
319 REM this part of the program
creates a TI BASIC cassette file
. if a disk file is required
modify OPEN statement.
320 INPUT "DO YOU WISH TO SAVE T
HIS     FILE (Y/N)?":S$
330 IF S$<>"Y" THEN 420
340 INPUT "FILE NAME?":NAME$
350 OPEN #1:"CS1".INTERNAL.OUTPU
T,FIXED
360 PRINT #1:NEQ.INT(NINT/NS).X0
.NS*H
370 FOR II=0 TO INT(NINT/NS)
380 FOR JJ=0 TO NEQ
390 PRINT #1:Y(JJ,II)
400 NEXT JJ
410 NEXT II
415 CLOSE #1
420 END
4999 REM the Kutta-Simpson integ
ration procedure starts here.
5000 XD0=XD
5020 FOR I=0 TO NEQ-1
5030 YD0(I)=YD(I)
5040 NEXT I
5050 FOR I=1 TO NEQ-1
5060 K1(I)=H*YD(I)
5070 NEXT I
5080 GOSUB 6000
5090 K1(NEQ)=H*F
5100 XD=XD+H/2
5110 FOR I=1 TO NEQ
5120 YD(I-1)=YD0(I-1)+K1(I)/2
5130 NEXT I
5140 FOR I=1 TO NEQ-1
5150 K2(I)=H*YD(I)
5160 NEXT I
5170 GOSUB 6000
5180 K2(NEQ)=H*F
5190 FOR I=1 TO NEQ
5200 YD(I-1)=YD0(I-1)+K2(I)/2
5210 NEXT I
5220 FOR I=1 TO NEQ-1
5230 K3(I)=H*YD(I)
5240 NEXT I
5250 GOSUB 6000
5260 K3(NEQ)=H*F
5270 XD=XD0+H
5280 FOR I=1 TO NEQ-1
5290 YD(I-1)=YD0(I-1)+K3(I)
5300 NEXT I
5310 FOR I=1 TO NEQ-1
5320 K4(I)=H*YD(I)
5330 NEXT I
5340 GOSUB 6000
5350 K4(NEQ)=H*F
5360 FOR I=1 TO NEQ
5370 YD(I-1)=YD0(I-1)+(K1(I)+2*
(K2(I)+K3(I))+K4(I))/6
5380 NEXT I
5385 GOSUB 6000
5386 YD(NEQ)=F
5390 RETURN
5999 REM this procedure specif
ies f(XD.YD(0).YD(1).....) in
the differential equation.
6000 F=-10*SIN(YD(0))
6010 RETURN
```

```
1 REM this program reads in
the data file created by the
Kutta-Simpson one.
100 CALL CLEAR
110 DIM Y(10.110).YMAX(10).
YMIN(10)
120 FOR I=0 TO 10
130 YMAX(I)=0
140 YMIN(I)=0
150 NEXT I
160 INPUT "FILE NAME?":NAME$
170 OPEN #1:"CS1".INTERNAL.
INPUT.FIXED
180 INPUT #1:NEQ.N.X0.H
190 FOR I=0 TO N
200 FOR J=0 TO NEQ
210 INPUT #1:Y(J.I)
220 IF Y(J.I)<YMIN(J) THEN
222 ELSE 230
222 YMAX(J)=Y(J.I)
230 IF Y(J.I)<YMIN(J) THEN
232 ELSE 240
232 YMIN(J)=Y(J.I)
240 NEXT J
250 NEXT I
260 CLOSE #1
270 FOR I=0 TO N
280 PRINT TAB(6):"X":TAB(21)
:"Y0"
290 PRINT TAB(1):X0+I*H:TAB(
16):Y(0.I)
300 PRINT TAB(6):"Y1":TAB(21
):"Y2"
310 PRINT TAB(1):Y(1.I):TAB(
16):Y(2.I)
315 PRINT
320 NEXT I
330 END
```

Line 10 sets aside space in the computer memory for all the arrays used in the program. **Lines 20-150** are used to input the order of the differential equation (line 20) (i.e. the number corresponding to the highest derivative, bearing in mind that zero order derivative means no derivative), the initial value of the independent variable X (line 30), X0 is set to the initial value of the independent variable (line 40), the initial conditions (lines 70-100), the final value of X (line 120), the number of intervals of integration (line 130, the larger this is the longer the program will take to run, but the more precise the answer will be; this number should be gradually increased until no significant change is detected in the new Y array), the number of steps to be performed between two storing operations into the Y array (line 140), the number of steps between two subsequent displays of the variables on the screen (line 150).

Line 160 calculates the increment in X used by the Kutta-Simpson integration procedure. **Lines 180-203** set the first column of the Y matrix (K=0). **Lines 205-208** print the initial conditions. **Lines 219-310** deal with the periodic storing and printing of the values of X and of all the derivatives of Y. **Lines 312-316** print X0, XF, the increment (H), the increment in the stored data (H*NS), the increment in the displayed data (H*ND). **Lines 320-420** deal with the writing of the data to a cassette file and end the main program. The operation of writing data to cassette is very time consuming and no more than 20-50 data should rasonably be processed in this way.

Lines 5000-5390 carry out the Kutta-Simpson integration procedure. I will not explain it in detail except to say that a clear exposition is to be found in the book entitled **DIFFERENTIAL EQUATIONS** of the Schaum Outline Series (author Frank Ayres Jr).

Lines 6000 and 6010 represent the function to which the highest order derivative is equal (in the case of equation 3 **-5*SIN[2*YD(0)]**).

The other program given above simply reads the cassette file, and can be used as the basis of a plotting routine. In order to assist in this, the maximum and minimum values of the rows of the Y matrix are picked out in the arrays YMAX and YMIN, whilst reading the file in. **Lines 270-320** simply print out the saved results again.

Have fun!?

P.S. If you want to test the program after typing it in follow this procedure:

```
RUN [enter]
2   [enter]
0   [enter]
0   [enter]
3   [enter]
2.5 [enter]
500 [enter]
10  [enter]
10  [enter]
```

and you will observe the output of a single complete oscillation of the simple pendulum. If you wish to store the data to a cassette recorder answer Y to the question appearing at the end of the calculation and subsequently follow instructions.

MODULE LIBRARY

TITLE	PRICE
ADDITION AND SUBTRACTION 1	3.00
ADDITION AND SUBTRACTION 2	3.00
ADVENTURE COMPLETE WITH PIRATE TAPE	5.00
ADVENTURE MODULE	3.50
ALPINER	8.00
A-MAZING	2.50
BEGINNING GRAMMAR	3.00
BLACKJACK AND POKER	3.00
BUCK ROGERS	4.50
COMPUTER WARS	14.00
CONNECT FOUR	3.50
DESK TOP PUBLISHER	9.00
DISK MANAGER*	2.00
DISK MANAGER 2*	3.50
DIVISION 1	3.00
DEFENDER (NOT FOR MARK II CONSOLES)	4.50
DONKEY KONG (NOT FOR MARK II CONSOLES)	4.00
EARLY LEARNING FUN	3.00
EARLY READING	3.00
EDITOR ASSEMBLER + MANUAL & DISKS	25.00
EXTENDED BASIC II PLUS (MECHATRONIC)+MANUAL	50.00
EXTENDED BASIC INC MANUAL	22.50
EXTENDED BASIC MODULE	15.00
FATHOM	4.50
HOUSEHOLD BUDGET MANAGEMENT	3.50
HUSTLE	3.00
INDOOR SOCCER	4.00
INTEGERS	5.50
JUNGLE HUNT (NOT FOR MARK II CONSOLES)	3.50
M*A*S*H	4.50
METEROR BELT	11.00
MICROSURGEON	4.50
MINI MEMORY + LINE ASSEMBLER + MANUAL	15.00
MINI MEMORY AS ABOVE + MINI WRITER	18.00
MINUS MISSION	4.50
MOON MINE	16.00
Ms PAC-MAN	4.00
MULTIPLICATION 1	3.00
MUNCH MAN	3.50
MUNCH MOBILE	11.00

NUMBER MAGIC	3.50
NUMERATION 1	3.00
NUMERATION 2	3.00
PAC MAN (NOT FOR MARK II CONSOLES)	3.50
PARSEC	4.00
PERSONAL RECORD KEEPING	3.50
PROTECTOR (NOT FOR MARK II CONSOLES)	4.50
SHAMUS (NOT FOR MARK II CONSOLES)	3.50
SOUND TRACK TROLLEY	3.50
SPACE JOURNEY	2.50
SPY'S DEMISE	4.50
STAR MAZE +	4.50
STAR TREK	4.50
STATISTICS	4.00
TERMINAL EMULATOR II	5.00
TI INVADERS	4.00
TI LOGO + ORIG. FOLDER WITH MANUAL*	15.00
TI LOGO II + ORIG. FOLDER WITH MANUAL*	25.00
TI WRITER + ORIG. FOLDER WITH MANUAL*	30.00
THE ATTACK	4.00
TOMBSTONE CITY	3.50
TUNNELS OF DOOM	3.50
VIDEO CHESS	5.00
YAHTZEE	3.00

Trials and Tribulations of a TI Emulator! Aficionado.

Thanks must go to Peter Hutchison whose article about **Edward Swartz' TI-Emulator!** (TI*MES Issue 45, page 56) brought the existence of this excellent PC software to my attention, and for sending me a copy of version 5.01. Even greater thanks must go to Mr. Swartz himself, for ensuring that the TI-99/4A lives on forever!

Now, I am not going to suggest for one minute that you should consign your trusty (or should that be 'rusty') old TI-99/4A console to the rubbish bin, and rush down to Dixon's with your spare pocket money to buy a multimedia PC with all the bells and whistles. But hey, if you've got one already (or maybe just access to one at work), then TI-Emulator! (don't forget the '!') has to be about the best UPGRADE around! And if you don't have a PC, rest assured that one day you WILL have ('cos Mr. Bill Gates says you will), and then you will still be able to use your TI-99/4A Emulator! when your TI console bites the dust. Doesn't that warm the cockles of your heart?

Anyway, having received the shareware version of TI-Emulator! V.5.01, I promptly sent my registration fee to Mr. Swartz, and received in return a speech file that enables the emulator to access all the speech that was programmed into the speech synthesiser (the emulator cannot as yet manipulate speech strings). Also a suite of programs to transfer TI files via a null-modem cable to the emulator on the PC (most programs work fine, and in particular there is no problem with TI-Basic/Extended Basic files).

Details about the shareware version of TI-Emulator!, and how to download it from the internet, are in Peter Hutchison's article in TI*MES Issue 45. It is well worth registering your copy, not only for the extra programs, but to give Mr. Swartz encouragement to further refine his excellent emulator. By the way, he is no longer able to include a selection of transferred modules with the registered version due to copyright restrictions, but then you can transfer your own modules anyway, so no problem.

The following is an extract from my correspondence with Mr. Swartz, which I hope will be of interest to TI*MES readers:

MIKE: I must congratulate you for your excellent TI-99/4a Emulator! v5.01.

EDWARD: Thank you for your kind words. I always felt it was a worthy successor to v4.0. In maybe a month or sooner (around March-April 1995), v6.0 should be out, and it will prove a worthy successor to v5.01, even though it's still incomplete.

MIKE: I have a few queries/comments regarding the emulator and documents. It hasn't locked up on me yet, other than when I load too many modules into the modules list, in which case it locks when I type MODULES/ALL at the dos prompt.

EDWARD: Yes, I'm sorry that the MODULES.EXE program had a rather silly bug which didn't allow more than 32 modules to be in the MODULES.LST file. Anyway, the newer versions of MODULES should easily handle around 500 items in the list. Also, v6.0 will have a much more intelligent format for the modules list, as well as a program to convert the MODULES.LST file for use with v6.0.

MIKE: If transferring modules to the PC, if you have a Navarone (or similar) Widget, it is possible to transfer without physically altering the module, simply by switching between modules without first pressing the on-board re-set button. However, it often took me as many as 20 attempts before the data downloaded onto my PC! Successful transfers by this method included Super Extended Basic, Editor/Assembler and TI-Writer.

EDWARD: I knew those devices existed but failed to mention that they could be used just as easily as the module-switching scheme I advised. I assumed that people resorting to that 'destructive' method of altering the cartridge, etc., didn't have a Widget. Hmm.

MIKE: An easier method of transfer is of course by downloading Gramcracker disk files. I actually have the P-Gram card from Bud Mills, and I can confirm that module dumps for this card can be transferred to PC with your GRAM2TI Program. However, your docs on this subject are rather brief, and being easily confused, took some working out! Can I suggest (for the ignorant amongst us) that you point out that Gramcracker disk files should still be transferred to PC via TRANSA0, and THEN converted to xxxx.HEX files by running GRAM2TI? I know it should be obvious but it had me stumped for a while.

EDWARD: I agree. I'm taking efforts to make comprehensive (and understandable!) documentation for v6.0.

MIKE: On the subject of file transfers, a few more details on how simple it is to transfer TI files to the PC should be included with your docs. My heart sank when I registered and realised what I had to do in order to transfer files. What is communications software? How do I know which one will work? Will I need a custom made lead? A few examples would give encouragement to those of us new to comms. Here's what I did --- it worked first time and I didn't even refer to the comms. software manual! I ordered a ready-made null-modem RS232 cable with a female PC 9-pin connector and a male 25-pin connector (no details of what it was for were given to the supplier) ....and it worked without any alterations!!!
The comms. software I used was PROCOMM PLUS for DOS, which was simplicity itself. All I had to do was start the software, press Alt-F1 to open a log file, press PageDown, then R to open a file for a raw ASCII download, and away it went! I have successfully downloaded masses of TI files to PC without any problems. However, although transfers are easy, they take ages ---- about 3+ hours for a DS/SD disk. Can this be speeded up?

EDWARD: The file transfer procedure is also annoying, as well as being slow. I'm working on programs that will run on both the TI and the PC and let users interactively transfer files back and forth.

MIKE: At the moment TRANSA0 will only allow transfer of the whole disk, or else every file has to be selected or ignored as it is reached on the disk. A useful facility would be to be able to select desired files BEFORE any transfer begins.

EDWARD: Point well taken. I have to admit that it's very frustrating for me to program on actual TI's (having only a SS/SD drive), and I skimped on the user-friendliness of that program a bit. I've now taken to programming on the emulator -- expect better stuff in the future.

MIKE: Will transferred files be compatible with future emulator upgrades (because I haven't been keeping my log files, just the files generated by ASC2BIN).

EDWARD: Yes, module files and disk files should be compatible with future versions of the emulator.

MIKE: If I write a basic program on the emulator, is it possible to transfer it to the TI-99/4a computer?

EDWARD: I don't know. If you have XMODEM on the TI , and also had a 'TI2XMDM' program which I haven't written yet, you could. The future transfer programs should let you do this without major effort, though.

MIKE: The emulator doesn't seem to differentiate between upper and lower case letters. Do you plan to aiter this to make it a more accurate emulation? Also, sometimes the CAPS Lock key doesn't work whilst running the emulator, and this can be frustrating.

EDWARD: The emulator is purposely made to let you enter lowercase filenames and such (like 'dsk1.art'). As for the CAPS LOCK key --- a bug with setting keyboard LEDs forced me to remove that code from the program (which naturally makes it difficult to tell if caps lock is on or not), AND another bug lets the caps lock key repeat! Hopefully all these bugs are solved in v6.0.

MIKE: I have added and deleted a lot of modules to the modules list, and realise now that it would have been best not to edit out removed modules from MODULES.LST, as I now have to manually add them back on the list using EDIT. I should of course have simply deleted the xxxx.HEX files from the directory and they would have remained on the list in a non-highlighted form. However, having done this, do you have a program to reinstate modules on the list without having to use EDIT and type in the details!?

EDWARD: I hope to write a program sometime which will scan a directory and make a list of modules.

MIKE: Super Extended Basic works fine, and it has a disk cataloguing command which is very useful: CALL CAT("DSK1."), especially as (presumably low level) programs like DM1000 and TI's Disk Manager don't seem to work. Can you do anything to overcome the inability of these programs to work?

EDWARD: I've solved the catalogue problem in v6.0. You see, many programs bypass the documented DSK1. file (which supplies a catalogue) and read sectors directly off the disk. Well, with the file-based method of emulating TI disks, there ARE no such sectors. But, I've come up with a way to get around it.

MIKE: Is it possible (or will it be at some stage) to program the emulator using low level languages such as Assembly language?

EDWARD: Any programs that run on the TI (within the constraints of what I've implemented) should run fine. Assemblers, editors, databases, drawing programs.... If Editor/Assembler is transferred to the emulator, that's how to program it in assembly language.

MIKE: I have transferred a lot of DV80 word processor files to the PC. It sure would be nice to be able to read them via a PC word processor. Anything in the pipeline? .....and what about an 80 column TI-Writer for the emulator (please!)

EDWARD: I think I had some utilities someplace which translated those files to *.TXT files, but I may be mistaken. But, as you may know, *.TXT files make for poor word processing documents since they are broken up into 80-character lines while word processors want every paragraph on one line. I'm thinking about coming up with a suite of conversion programs from TI <---> PC.

MIKE: You say that it is unlikely a TI speech synthesiser could be plugged into a PC, so I will have to encourage you to work on full speech emulation for a future upgrade! (No hurry---I can hang on a month or two!)

EDWARD: I've received some excellent articles about how it's done, but there are some details which still elude me.

MIKE: As I have a Soundblaster Pro, would it be possible to overcome the limited vocabulary of the TI speech synth? Could you write a program that would enable me to access my own voice, having digitally recorded it to my soundblaster? Could be interesting.........

EDWARD: It could be done pretty easily, but the v5.0 code is heavily hardcoded to conform to the TI speech ROM. (You'd have to overwrite currently existing words in the TISPEECH.BIN file for it to work). But if I can get full speech emulation to work, such a feature would be unnecessary, because you could use Terminal Emulator II!

MIKE: OK, so it's not the end of the world, but it would be nice to be able to play Donkey Kong, Moon Patrol or even Parker Bros. Popeye on the emulator [for those who don't know, the vast majority of cartridges transfers DO work with the emulator -- mgp].

EDWARD: I don't know why, but I rewrote the opcodes one day and found that those programs now worked!

MIKE: Finally........keep up the good work, it really is appreciated.

It is quite apparent that Edward is dedicated to the task of producing a complete TI-994A Emulator! in the fullness of time. The emulator is already amazing in it's current incarnation, and version 6.0 is just around the corner........

Please send your registration fee of $40 to Edward if you have the shareware version and are using it. A lot of work has gone into producing the TI-Emulator!. If you don't have the shareware version and want to go straight for the registered version, send your $40 bank draft to the following address:

Edward Swartz Esq.
1401 East 18th Street
Georgetown
TX 78626
U.S.A.

Please specify when ordering what disk size and density you require for the DOS programs for your PC. The TI programs will be supplied on 5.25 inch 90k TI disks.

Edward can also be contacted personally by e-mail if you have a modem:

swartze@ralph.txswu.edu

Happy TI-Emulating!,

Mike Poskitt

FCTN-QUIT

Unfortunately, the article on the real time clock will have to wait until the next issue due to lack of time on my part. This article should prove to be interesting reading for anyone wishing to add a real time clock directly to the console or via the Zeno board.

It will highlight some of the issues made by Mr. Zeno and will also cover the problems encountered and solved by members of the group who have given this a go.

Down to business!

I must apologise on behalf of the publishers for the miss-printing of the last issue. We believe that at some point the scripts have been dropped, and reordered, missing some of them out and causing a couple of them to be duplicated.

I have been notified that the price for printing the magazine has increased even though the paper being used is thinner than the paper being used three issues ago.

I would like to thank everyone who has submitted articles for this issue (keep on writing!). Hopefully the two articles which the publishers did not print last time will be in the next issue.

Sorry for the late delivery of the Christmas issue. This was due to more factors than I could list here.

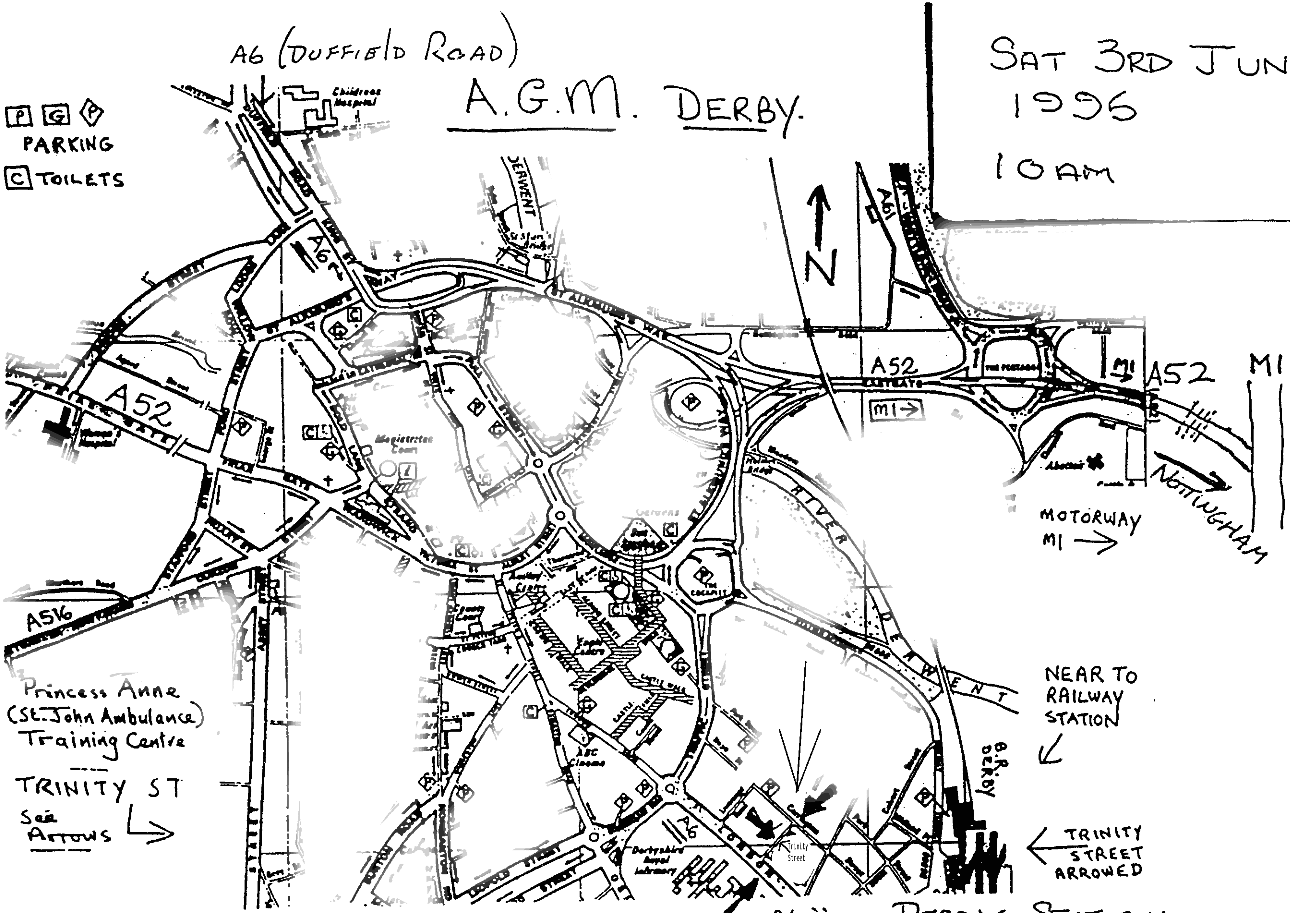
Finally,

Do not forget the A.G.M. this year. If you are interested in looking at new software, maybe the BBS, or talking hardware, or have a problem you need sorting out then come along. There are always people there who are more than willing to try and help out.

The positions of Editor and Membership Secretary are up for grabs so if you want to try your hand at them or wish to vote someone in for these positions then please turn up or let the Chairman know in advance of the meeting.

A.G.M. DERBY.

SAT 3RD JUNE
1996
10AM



NEAR TO RAILWAY STATION ↙

Princess Anne
(St. John Ambulance)
Training Centre

TRINITY ST
See Arrows →

← TRINITY STREET ARROWED

MAIN ROADS : M1
A52
A6

TRINITY STREET,
JUST OFF LONDON

A6

DERBY STATION
EW lines (Birmingham-Lincoln)
via (Nottingham & Newark)