TI*MES

TI99/4A USER'S GROUP (U.K.) CONTACTS

Chairman : Trevor Stevens. Tel.0623 793077
249 Southwell Rd. East, Rainworth. Notts. NG21 OBN
Vice Chairman & Programming: Mark Wills.
12 "Rosehill" , Betton St., Shrewsbury, Shropshire. SY3 7YN
General Secretary: Jim Ballinger. Tel.0332 772612
5, Offerton Ave., Derby. DE3 8DU
Publicity Officer: Philip Trotter. Tel.0642 817356
80, Martonburn Rd., Grovehill, Middlesborough. TS4 2TH
Membership Secretary: 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
TI*MES Editor & Distribution: Alan Bailey. Tel.081 508 1053
14, Shelley Grove, Loughton, Essex. IG10 1BY
Hardware & Projects: Mike Goddard. Tel.0978 843547
"Sarnia", Cemetary Rd., Rhos, Wrexham, Clwd. LL14 2BY
Librarians:Cassette: Nicky Goddard. Tel.etc. as above.
Modules: Edward Shaw. Tel.0538 360382(5pm to 8pm)
Crow Holt Farm, Basford, Leek, Staffs. ST13 7DU
Disk: Stephen Shaw.
10, Alstone Rd., Stockport, Cheshire. SK4 5AH
Publications: Mike Curtis. Tel. 0209 219051
21, Treliske Rd., Roseland Gdns., Redruth, CornwallTR15 1QE

MAGAZINE CONTENTS

ISSUE NO. 35 WINTER 1991/92

EDITORIAL

As you will see from the front cover, we now have a new Membership Secretary, who will of course also handle TI*MES back numbers. Greetings and thanks to Alasdair Bryce on undertaking this considerable task!
Apologies for the late delivery of this issue; it had in fact to wait for the details of the venue for the Maidenhead meeting, and critical post seems to have been completely lost!
I hope you find it worth waiting for, it is of the long winter reading type, and has several new viewpoints to offer. Please remember that we would be happy to add yours to that number , any time you care to let us have them, in whatever form.
Directions are given for getting to the Southern Show, to be held in Maidenhead on 1st. February. This is in place of the Cuffley show when this suddenly lost its organiser. Our special thanks are due to Mike Goddard and the Ross's for taking up the challenge. Your attendance will make it all worthwhile!

DISCLAIMER

Views expressed in this magazine are those of the contributor, and not necessarily supported by the Committee. If you disagree with anything you see we will happily record your objections, or your agreement for that matter!
We would also like to acknowledge here anything we have inadvertantly missed acknowledging in the text.

NEXT COPY DATE

All copy for the next, Spring, issue of the magazine should reach the Editor by 1st.March. Please make the print as black as possible, and keep it within, but also as near as possible to, an area 180mm wide by 262mm high.
Unfortunately computer print-out paper is not either in width or height to any European standard e.g. A4 or A5. Since I have to trim with scissors I would be grateful if the left hand and top margins could be 15mm wide, so that I need only trim the right hand and bottom edges. But therein in fact lies the rub! The length of computer paper is insufficient for A4! Failing a sheet feeder and A4 paper, the optimum would seem to be to ask you to print to the 262mm height, even if this goes on to the next sheet so that I can paste-up to the right A4 length.
Alternatively, how about investing in a box of the "Exact A4 continuous letter" paper, and make my day, or year more exactly.

TIUG

TIUG(UK) MEMBERSHIP NEWS

by T.STEVENS c1991

This last few months has been very hetic in the Membership Secretaries job. Things have happened that you may or may not have noticed. First of all Peter WALKER is no longer carrying out the job, so I have stepped in to stop it going under. Before I go any further I would like to say on behalf of all our members a big thankyou to Peter for his years of good work in filling this very difficult job, from which he is really missed. Peter has not however given us up completely and still remains a member, with a promise of a few of his good articles.

Things do look rosey for the post as Alasdair BRYCE has made it known that he would like to fill the post. So with the postal election being carried out by now will in the new year be your new membership secretary. Watch this space for full address etc.

The Group has just obtained TI BASE V3 to run the groups data base. I have just finished writting a menu driven data base in the TIBASE language for the group. Its taken some debugging and I have learned a lot about TIBASE. I do intend in the near future to pass on a few tips to all you out there with TIBASE, in an Tutor type artical.

Things still look good for the group though we have had a few folks not renew subscription. However we have had some new blood join us, so I welcome you to our group. On a sad note one of our members Victor Vincent died recently. The group sends to his widow our deepest condolances. Anyone knowing Mr Vincent can contact me for further details.

On a more lighter note I have seen some of the comments regarding the improvement of our group. One of the most prevelant is that for Basic and Extended Basic Programs for you to type in. So all you budding programers get going with something. If you hav'nt got a full system do'nt worry. If you wish you can send me your program for listing onto paper and I will return its to you. (S.E.A and stamp please.) So as not to be out done I have submitted a EXB program I wrote some time back. Its called MOON BASE ALPHA I hope you like it.

Just as an Idea which I know Peter has done before, we do have the capablity to put YOU in touch with other members in your area. So if you send me a letter I can give you a print out of everyone wishing contact in your area. Again S.E.A and stamp please. If you want verbal contact do not forget the HOT LINE on 0623 793077

Some folks have had problems with the Monday BBS. To get on set your Modem to 300 300 originate. Use Telco if you have it and set up as ANSI 8N1. This works even for PC's like Amstrad IMB etc.

Have a very Merry Christmas 1991 and happy key tapping.

```
110 REM BY TREVOR STEVENS
120 CALL INTRO :: CALL INST
130 CALL CLEAR
140 CALL CHARSET
150 CALL CHAR(64,"0000000000
000000")
160 RANDOMIZE
170 CALL COLOR(5,2,11):: CAL
L COLOR(6,2,11):: CALL COLOR
(7,2,11):: CALL COLOR(8,2,11
)
180 Y=4000 :: P=50 :: OL0200
 :: FS=200 :: MI=2 :: M=0 ::
 MO=100 :: WS=200
190 CALL SCREEN(4):: CALL CL
EAR
200 A$="IF@YOU@WISH@TO@BUY@M
ORE" :: B$="THEN@ENTER@THE@N
UMBER@OF" :: C$="YOU@WISH@TO
@BUY."
10 D$="IF@NOT@THEN@PRESS@NU
MBER (0)" :: E$="YOUR@CASH@L
EVEL@IS"
220 GOTO 280
230 CALL CLEAR
240 OL=OL-INT(RND*25):: FS=F
S-INT(RND*25):: WS=WS-INT(RN
D*25):: Y=Y+1
250 IF OL<50 THEN P=P-10
260 IF OL>=50 AND OL<=100 TH
EN P=P+INT(ND*5)
270 IF OL>100 THEN P=P+INT(R
ND*10)
280 IF OL<0 THEN 1140 :: IF
FS<0 THEN 1140 :: IF WS<0 TH
EN 1140 :: IF P<0 THEN 1140
290 DISPLAY AT(2,5):"MOON@BA
SE@ALPHA"
300 DISPLAY AT(4,0):"( ) YEA
R,          ";Y :: DISPLAY AT(
6,0):"( ) POPULATION,  ";P :
: DISPLAY AT(8,0):"(1) OXYGE
N@LEVEL,";OL
310 DISPLAY AT(10,0):"(2! FO
OD@SUPPLY, ";FS :: DISPLAY A
T(12,0):"(3) WATER@SUPPLY,";
WS :: DISPLAY AT(14,0):"(4)
MISSILES,     ";MI
320 DISPLAY AT(16,0):"(5) MI
NERALS,     ";M :: DISPLAY AT
(18,0):"(6)MONEY,         ";M
O
330 IF RND>.8 THEN 1010
340 PRINT "PRESS@A@NUMBER" :
:
350 ACCEPT AT(23,17)SIZE(1)B
EEP VALIDATE("123456"):N
360 CALL CLEAR
370 IF N=1 THEN 380 :: IF N=
2 THEN 460 :: IF N=3 THEN 55
0 :: IF N=4 THEN 640 :: IF N
=5 THEN 720 :: IF N=6 THEN 8
50
380 PRINT "OXYGEN@LEVEL": :
390 PRINT "YOU@HAVE";OL; UNI
TS": :
400 L=OL :: GOSUB 940
410 V=INT(RND*2+1):: PRINT E
$;MO;
420 PRINT "OXYGEN@COSTS";V;"
PER@UNIT."
430 PRINT A$;"OXYGEN" :: PRI
NT B$;"UNITS@";C$ :: PRINT D
$
440 INPUT  :: IF N<0 OR N*V
>MO THEN 440
450 OL=OL+N :: MO=MO-(N*V)::
 PRINT :: PRINT "OK" :: CALL
 DELAY :: GOTO 230
460 PRINT "FOOD@SUPPLY" :: P
RINT
470 PRINT "YOU@HAVE";FS;"KIL
OGRAMS" :: PRINT
480 L=FS :: GOSUB 940
490 V=INT(RND*2+1)
500 PRINT E$;MO
510 PRINT "FOOD@COSTS";V;"A@
KILOGRAM" ::
520 PRINT A$;"@FOOD" :: PRIN
T B$;"KILOS@";C$ :: PRINT D$
530 INPUT N :: IF N<0 OR N*V
>MO THEN 530
540 FS=FS+N :: MO=MO-(N*V)::
 PRINT :: PRINT "OK" :: CALL
 DELAY :: GOTO 230
550 PRINT "WAER@SUPPLY"
560 PRINT :: PRINT "YOU@HAVE
";WS;"PINTS." :: PRINT
570 L=WS :: GOSUB 940
580 V=INT(RND*2+1)
590 PRINT E$;MO
600 PRINT "WATER@COSTS";V;"A
@PINT."
610 PRINT A$;"WATER " :: PRI
NT B$;"PINTS@";C$ :: PRINT D
$
```

```
/THEY'RE@GONE,      WATCH@IT@
THEY'LL@BE@BACK" :: MI=MI-N
1110 CALL DELAY :: GOTO 230
1120 DISPLAY AT(6,0):"OK)" :
: FOR Z=1 TO 300 :: NEXT Z :
: DISPLAY AT(6,6):"YOU@MISSE
D/TRY@AGAIN" :: MI=MI-N :: C
ALL DELAY
1130 DISPLAY AT(6,6):"
                " :: GOTO 106
0
1140 CALL CLEAR : FOR ZA=1
TO 100 STEP 2 :: CALL HCHAR(
11,5,32,20)
1150 DISPLAY AT(11,5):"LIFE@
TERMINATED" :: NEXT ZA
1160 IF MI<1 THEN A$="DESTRO
YED@BY@ALIENS"
1170 IF OL<0 THEN A$="OXYGEN
@STARVATION"
1180 IF FS<0 THEN A$="FOOD@S
TARVATION"
1190 IF WS<0 THEN A$="LIQUID
@STARVATION"
1200 IF P<1 THEN A$="POPQLAT
ION@DEAD"
1210 CALL SCREEN(9):: CALL S
OUND(1250,-6,0)
1220 DISPLAY AT(9,5):A$
1230 DISPLAY AT(13,5):"YOU@S
ERVED@FOR@";Y-4000;"YEARS"
1240 CALL DELAY
1250 PRINT "  DO@YOU@WAT@TO
@PLAY@AGAIN?" :: PRINT :: PR
INT "  PRESS@Y@FOR@YES@N@FOR
@NO"
1260 CALL KEY(0,K,S):: IF S=
0 THEN 1260 :: IF K=89 THEN
CALL INTRO :: GOTO 130 :: IF
 K=78 THEN 1270
1270 END
1280 SUB DELAY
1290 FOR Z=1 TO 900 :: NEXT
Z
1300 SUBEND
1310 SUB INTRO
1320 CALL CLEAR :: CALL SCRE
EN(2)
1330 FOR ZZ=5 TO 11 :: CALL
COLOR(ZZ,15,1):: NEXT ZZ
1340 CALL CHAR(97,"000000000
000080F"):: CALL CHAR(98,"00
000000000010F8"):: CALL CHAR
(99,"1F3BEDFFDEFFDEFF"):: CA
LL CHAR(100,"FCF6FFFBFDF7FFF
F")
350 CALL CHAR(117,"00000080
C0F9FBBF"):: CALL CHAR(102,"
0000000000080EFFF"):: CALL CH
AR(103,"00000000CCFFFFFF")
1360 CALL CHAR(104,"00000001
87C7FFFF"):: CALL CHAR(105,"
00000000000000F8"):: CALL CH
AR(106,"000000000000001F")
1370 CALL CHAR(107,"18FFFFFF
999999FF"):: CALL CHAR(116,"
FFFF999999FFFFFF"):: CALL CH
AR(109,"0000FFFF55FF0000")!B
UILDINGS
1380 CALL CHAR(110,"00000018
08080818"):: CALL CHAR(111,"
0000001810101018"):: CALL CH
AR(112,"00007E81817E818")
1390 CALL CHAR(113,"00008"):
: CALL CHAR(114,"FFFFFFFFFFF
FFFFF")
1400 CALL HCHAR(24,1,114,32)
1410 A$="ufufgcdufcdhufitttt
mmttjufgcdugh"
1420 X=23 :: Y=1
1430 GOSUB 1670
1440 A$="     ab  ab    tttt
  tt    ab"
1450 X=22 :: Y=1
1460 GOSUB 1670
1470 A$="                  k"
1480 X=21 :: Y=1
1490 GOPUB 1670
1500 CALL SPRITE(£1,112,5,15
2,129)
1510 FOR TU=1 TO 30
1520 KJ=INT(RND*32)+1
1530 PA=INT(RND*18)+1
1540 IF (PA=24)+(PA=23)THEN
32767
1550 CALL HCHAR(PA,KJ,113)
1560 NEX TU
1570 DISPLAY AT(3,9)SIZE(16)
:"MOON BASE ALPHA"
1580 DISPLAY AT(6,5)SIZE(21)
:"BY T.STEVENS FOR TIUG"
1590 DISPLAY AT(15,5)SIZE(22
):"PRESS ANY KEY TO GO ON"
1595 CALL SPRITE(£2,113,7,10
,50,0,-15)
1600 CALL KEY(0,KK,SS)
```



```
1610 CALL PATTERN(£1,110)::
FOR A=1 TO 6 :: NEXT A
1620 CALL PATTERN(£1,112)::
FOR A=1 TO 6 :: NEXT A
1630 CALL PATTERN(£1,111)::
FOR A=1 TO 6 :: NEXT A
1640 CALL PATTERN(£1,112)
1650 CALL SOUND(50,500,15)
1660 IF SS=1 THEN 1710 ELSE
1600
1670 FOR Z=1TO LEN(A$)
1680 CALL HCHAR(X,Y+Z-1,ASC(
SEG$(A$,Z,1)))
1690 NEXT Z
1700 RETURN
1710 CALL DELSPRITE(ALL):: C
ALL CHARSET
1720 SUBEND
1730 SUB INST
1740 CALL CLEAR
1750 CALL SCREEN(12)
1760 DISPLAY AT(1,7)SIZE(16)
:"MOON BASE ALPHA"
1770 DISPLAY AT(2,7)SIZE(16)
:"****************"
1780 DISPLAY AT(7,1)SIZE(28)
:"YOU ARE THE COMMANDER OF"
1790 DISPLAY AT(8,1)SIZE(28)
:"MOON BASE ALPHA.          "
1800 DISPLAY AT(9,1)SIZE(28)
:"THE OBJECT OF THE GAME IS"
1810 DISPLAY AT(10,1)SIZE(28
):TO SURVIVE ON THE MOON AS
"
1820 DISPLAY AT(11,1)SIZE(28
):"LONG AS YOU CAN BY BUYING
"
1830 DISPLAY AT(12,1)SIZE(28
):"AND SELLING ESSENTIAL
"
1840 DISPLAY AT(13,1)SIZE(28
):"COMMODITIES.
"
1850 DISPLAY AT(14,1)SIZE(28
):"THE YEAR IS A.D.4000 AND
"
1860 DISPLAY AT(15,1)SIZE(28
):"YOU ARE AT WAR WITH ALIEN
S."
1870 DISPLAY AT(18,1)SIZE(28
):"MAY YOU PROSPER AND MULTI
PLY"
1875 DISPLAY AT(24,3):"PRESS
 ANY KEY TO GO ON"
1880 CALL KEY(0,K,S):: IF S=
0THEN 1880
1890 SUBEND
```

MODULES MODULES MODULES MODULES

APPEAL TO ALL DISK DRIVE OWNERS.....

Do you have any modules that you would consider selling or donating to the module library. Reasonable prices paid. For more information please contact me at the address given below.

The latest list of modules available for purchase follows; please note that cheques should be made payable to "E.H.SHAW". Also members are advised to contact me about the modules that they are seeking as the stock is constantly changing.

	ADDITION AND SUBTRACTION 1	2.00		NUMBER MAGIC	4.00
	ADVENTURE and PIRATE TAPE	5.00		HOUSEHOLD BUDGET MAN.	3.00
	BLACKJACK + POKER	2.50		PROTECTOR	4.50
	AMAZING	3.50			
	BEGINNING GRAMMAR	3.00			
	HOME FINANCIAL DECISIONS	2.50		OTHELLO	4.00
	CONNECT 4	3.50		THE ATTACK	3.50
*	DISK MANAGER I	3.00		LOGO I (NO MANUAL)	4.00
	EARLY READING	2.50		SPEECH EDITOR	4.00
	EARLY LEARNING FUN	2.50	*	TI WRITER / MANUAL	12.00
*	EXTENDED BASIC AND MANUAL	22.50	*	TI MULTIPLAN/DISK/MANUAL	25.00
	PERSONAL RECORD KEEPING	3.50	*	EDITOR ASSEM / MANUAL	22.50
	PERSONAL REPORT GENERATOR	3.50		TERMINAL EMULATOR II	5.00
				VIDEO GAMES 1	3.50
				VIDEO CHESS	4.50
				YAHTZEE	4.00

Stocks are lower than usual at present but I am expecting to acquire more modules soon, please give me a ring to see what is in stock.

* MODULES MARKED WITH AN ASTERISK REQUIRE DISKS OR 32K RAM OR BOTH. ALSO PLEASE NOTE THAT EARLY READING NEEDS A SPEECH SYNTH TO RUN.

PURCHASING MODULES FROM THE LIBRARY

You may return any module purchased within four weeks and be refunded the purchase price less postage which will be charged at the rate of 40 pence per module.

Application to loan/purchase modules.

Name:..... Modules required:.....
Address:..... .....
....
....
....

I enclose cheque/PO for £.....(as indicated on the list) & post to

PLEASE MAKE CHEQUES PAYABLE TO E.H.SHAW.
Foreign orders can only be accepted if a BANKERS DRAFT is enclosed drawn in STERLING on a LONDON bank. It also helps if a little extra is added on for postage overseas.

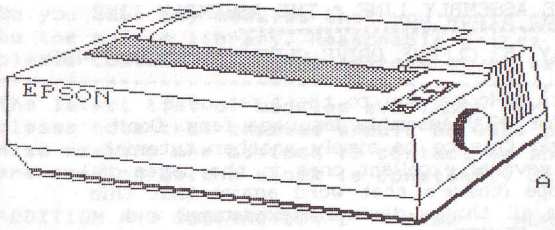
MR. E.H. SHAW
CROW HOLT FARM
BASFORD
LEEK
STAFFS. ST13 7DU

****
* THE ASSEMBLY LINE * THE ASSEMBLY LINE * THE ASSEMBLY LINE *
****

BY N.BARRIE CLARK: SHANTAY, 53 CLEEVE DRIVE CLEEVE AVON BS19 4NP

Greetings coders. I'd like to welcome you to the first of a (hopefully) regular column for 9900 Assembly language fans. Don't get me wrong. I don't intend this to be simply another tutorial page, because there's been several excellent ones in the pages of this tome already. No, I hope (there's that word again) that this will be the place to discuss all things decimal, hexadecimal and binary. In the last issue of TI*MES, as you may or may not recall, there were a couple of my 9900 routines. The response to my article sparked the idea for this column.
Right, lets get the ball rolling. Now I want to hear from everyone who has the capacity to run assembly language programs on their computer. What programs would you like to see? Have you found a new and exciting use for the MINI-MEMORY module? Is there a bug in your program that you can't fix? Write to me at the address above and I'll try to sort it out in this column. We could even form a sort of think thank! We could even write a large assembly program together, with people coding the parts in which they specialise (I/O, GRAPHICS, SPEECH etc.) like the professional software houses do! Write and let me know.
I was tinkering with my TI the other day, trying out some stuff that I was planning to use in a game. I wasn't too sure what sort of values the joystick would produce under a machine code environment. To solve this problem, I wrote the following program which reads the values off the joystick and prints them onto the screen. In doing this, I also created a sub program that could be used in the game itself and hey presto! part of my game was complete! I hope you will find it useful. Bye for now.

```
5       DEF START
        REF KSCAN,VMBW
STATUS  EQU >837C
KEYVAL  EQU >8375
KEYUNT  EQU >8374
YPOS    EQU >8376
XPOS    EQU >8377
JU      BYTE 4
JD      BYTE >FC
JN      BYTE 0
NOWT    BYTE >FF
JKU     BYTE 1
DIS1    DATA '04'
DIS2    DATA '-4'
DIS3    DATA '00'
SAV11   BSS  2
START   MOV  R11,@SAV11

JOYLP   CLR  @KEYUNT
        MOVB @JKU,@KEYUNT
        BLWP @KSCAN
        CB   @YPOS,@JU
        JEQ  YUP
        CB   @YPOS,@JD
        JEQ  YDN
        CLR  @STATUS

        LI   R0,1
        LI   R1,DIS3
        LI   R2,2
        BLWP @VMBW
NUKE    CB   @XPOS,@JU
        JEQ  XUP
        CB   @XPOS,@JD
        JEQ  XDN
        CLR  @STATUS
        LI   R0,9
        LI   R1,DIS3
        LI   R2,2
        BLWP @VMBW
        JMP  JOYLP
YUP     CLR  @STATUS
        LI   R0,1
        LI   R1,DIS1
        LI   R2,2
        BLWP @VMBW
        JMP  NUKE
YDN     CLR  @STATUS
        LI   R0,1
        LI   R1,DIS2
        LI   R2,2
        BLWP @VMBW
        JMP  NUKE

XUP     CLR  @STATUS
        LI   R0,9
        LI   R1,DIS1
        LI   R2,2
        BLWP @VMBW
        JMP  JOYLP
XDN     CLR  @STATUS

        LI   R0,9
        LI   R1,DIS2
        LI   R2,2
        BLWP @VMBW
        JMP  JOYLP
        END
```



NEAR LETTER QUALITY PRINTING for your EPSON

A REVIEW OF THE *DOTS-PERFECT* UPGRADE ROM

You may not know that there is an upgrade kit on the market for those with Epson printers that simply, and relatively cheaply, gives you the facility to print to **Near Letter Quality** standards. (The small manual that comes with the kit is dated 1987, so I suppose it has been for sale since then - but as I hadn't seen it referred to in TI*MES I thought it might be worth a short article).

I found out about Dots-Perfect through a fly-leaf inserted in some printer catalogues I was thumbing through at work. The bumph was from Micro Partners in Wembley, so I gave them a ring, was convinced by the sales talk and two days later received a kit in the post (cost £25 + P & P £1.00 + VAT £4.56 = £30.56 Total). The product consists of three chips to be inserted in your Epson, the manual giving very clear descriptions and labelled photographs to make the process extremely straight forward; it also tells you that Dots-Perfect is made by Dresselhaus Computer Products and was formerly known as Finger Print Letter Writer.

After installing the chips, the printer dip switches have to be reset (again good guidance given) then your modified printer can be tested:

As you can see, a much finer character detail is given and this is achieved by using a two-pass character set which results in four times the resolution of regular draft print. NLQ can be selected from the printer panel buttons or from software, ie. by using the appropriate 'Escape' code in your text. NLQ applies to:

ENLARGED print;
emphasized; *italics;*
proportional and super/sub-scripts.

The clever part of Dots-Perfect is that it allows you to select which features you want for printing out a particular document by selecting them from a printer menu, changing them line by line if so desired, rather than complicated embedding within your text. It does this by re-defining the LF and FF keys and employing the printer's beeper to indicate which feature you have come to in the menu. So, the menu is selected by pressing ON-LINE & FF together, this gives two short beeps; every press of FF then moves the selection to the next function which can be toggled on or off by pressing LF.

Stepping through the functions gives the choice singly, or in combination, of:

1	Condensed	6	Double-strike	11	Fine Print
2	Elite	7	Perf-skip	12	8 lines per inch
3	Proportional	8	1/2" left margin	13	Quiet mode
4	Double-wide	9	Italics	14	Slash zero
5	Emphasized	10	Underline	15	8 1/2" wide paper

I'm very pleased with the upgrade. It gives a more polished look to those important letters (like to the bank manager!) and makes printing style selection very easy. Judge for yourselves.

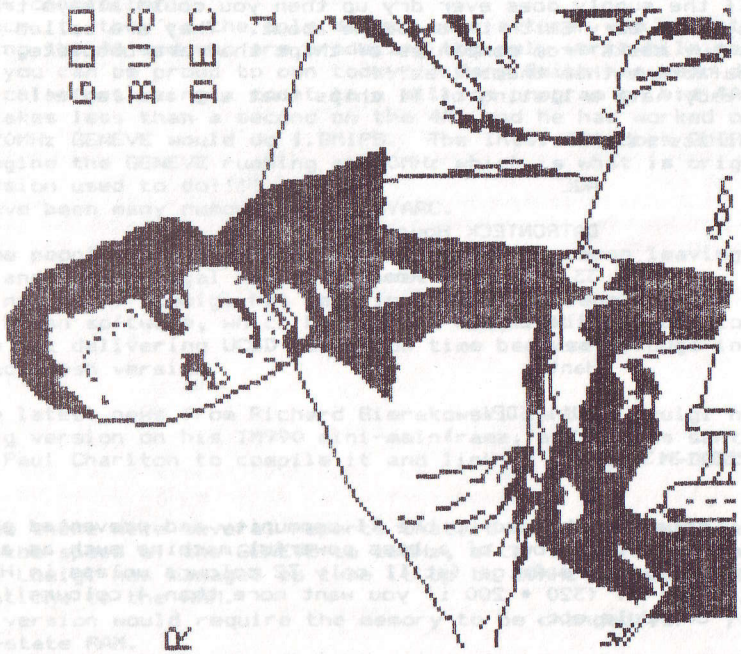
ASHLEY TILLING

PS. I got my kit for the FX/JX at prices stated above from:

Micro Partners
Unit 45
Park Royal Business Centre
9/17 Park Royal Road
LONDON NW10 7LQ (TEL 081-900 2770)

They are also available for the Epson RX at £25 and the MX at £40.

GOOD BUSINESS. DECEMBER 1988



JOHN LEE HOOKER

Dear TI'ers,
This is the very first article I have written for TI*MES as I have only just joined the group after leaving EAR 99'ers.

I was prompted to write this article after reading one or two things in the Autumn issue of TI*MES.

Firstly, I wish people would not keep spreading bad news without researching thier facts first. After reading the article on TI withdrawing certain chips, I contacted Richard Sierakowski who informed me that the TMS9995 would not go out of production since it is used extensively in Real-Time Process Control modules (PLC's etc.)

Also several chips such as the TMS9901 are made under licence by other companies and are available under other numbers.
And who cares about the 9918 when YAMAHA are manufacturing the 9938 and 9958 under licence.

Richard also informed me that the world supply of chips that TI are withdrawing will last for up to ten years yet, so there's no need to rush out to car boot sales looking for bargain prices on second hand consoles!!!

If the supply does ever dry up then you could always try out a company that Gary Smith informed me about. They are called AQL and they buy masks from companies of chips that are obsolete, and then manufacture them themselves!!!
They already have a listing of TI chips that are available!!!

Their address is:

AQL

DATRONTECK House

33 Grovesnor Road

Oldershot

Hants.

GU11 3DP

Tel: (0252) 313155

Hopefully this has cheered up the TI community and prevented people from selling up in favour of a less powerful machine such as an IBM compatible, Commodore Amiga (still only 32 colours unless in HAM mode!!), Atari ST (320 * 200 if you want more than 4 colours!!! Very Poor!!!), or Apple etc.

The other two little moans I must have are related to the

GENEVE. The first moan is about Mark Wills saying that the GENEVE is no more, and that Myarc are dormant. Myarc are not dormant, and GENEVE's are still being shipped out, but there is a 6 month delay on all repairs and there is a delay on hard disk controllers due to the demand from 4A owners in the U.S.A. who upgrade to hard disks before buying a GENEVE (That's the way I did it!!).

Some of you may know that I have been without my GENEVE for over a year since it burned out in August 1990. Part of the problem is that Richard deals with Louigi directly, and Louigi is not the fastest man in the world.

The problem is not helped by the lack of support that Richard suffers from. If more people ordered GENEVE's or other MYARC products from Richard, then he would be able to place more orders with MYARC. Lou Phillips would then be quicker to deal with Richard, and his prices would come down. The problem at the moment is that Richard is not having many orders since Louigi has not sent any equipment lately, because Louigi would sooner deal with a larger dealer such as Texaments.

People would sooner knock MYARC instead of praising them for what they have achieved. Where would we be without the GENEVE and hard disk controller!
Let's face it, they're the only decent manufacturer who are still supporting the 4A and who are producing the only worthwhile new machine you can be proud to own today!!! Gary Smith has been doing some calculations using a count to a million program on his 4A (which takes less than a second on the 4A) and he has worked out that a 20MHz GENEVE would do 1.8MIPS. The Intel 486 does 2MIPS. Just imagine the GENEVE running at 30MHz which is what is original 256K version used to do!!!!
There have been many rumours about MYARC.

Some people said that MYARC's employees have been leaving the company and taking legal action against Louigi.
This is not true. Louigi has been involved in a legal battle against Pecan software, which he has now won. This was due to Pecan Software not delivering UCSD Pascal on time because of bugs in the Apple Macintosh version.

The latest news from Richard Sierakowski is that Louigi has got a working version on his TM990 mini-mainframe, and he has sent a copy to Paul Charlton to compile it and link it in with M-DOS.

Also there were several reports that it would be possible to upgrade the speed of the GENEVE to 16MHz, after researching the problem, Louigi has managed to take it up to 20MHz without making modifications to the RAM.
A 30MHz version would require the memory to be changed for no-wait-state RAM.

The other moan I must have is about the comment that Stephen Shaw made about his Mandelbrot program in the Autumn TI*MES. He said, "the program cries out for some machine code!" What the program really cries out for is XHI running in Extended BASIC on a 20MHz GENEVE, or even C99 M-DOS.

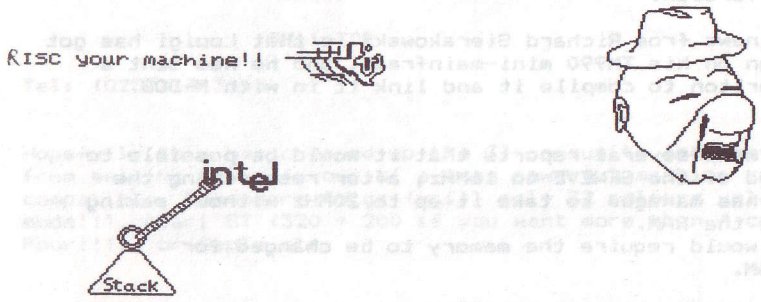
I don't know why, but people seem to treat GENEVE owners as traitors, but the GENEVE is the only worthwhile method of expanding the 99/4A.

Try adding up the cost of a 512K card, 80 column card GRAM emulator, RAVE 99 keyboard adapter.

GENEVE owners are not loyal to the 4A, but to the most advanced computer company in business today, TEXAS INSTRUMENTS. The inventors of the integrated circuit, the manufacturers of the first transistor radio in 1954, the inventors of sprite graphics, speech synthesis, and the first 16-bit microprocessor that provides the most advanced architecture ever introduced.

On my degree course I am doing 68000 programming, and it's just trash at the side of the 9900 and 9995. We recently studied the LINK and UNLINK instructions that allow the 68000 to organize its stack to reserve space for local variables for each subroutine or interrupt. The local variables then have to be pushed and pulled to and from the 68000's registers onto stack!!!

This feature is provided as standard on our stackless RISC architecture.
One Bullwhip!! (BLWP) and we've got a brand new set of General Purpose registers in a different part of memory!!!
At polytechnic, some people say, "doesn't it slow you're machine down having your registers in memory?" "Not at all", I say, "The TMS9995 has 258bytes of on-chip RAM, which provides enough memory for 16 Workspaces of 16 registers each!!" That's 129 sixteen bit registers on-chip!!!



Don't RISK your data!!!

The next part of this article should have been devoted to a new video card that Gary Smith and I were designing using the YAMAHA V9990, but we found at first that we couldn't surface mount the chip, and we found out too late that there was a socket available, and the chip had already gone back to the distributors.
Now we are aiming at the TEXAS INSTRUMENTS TMS34010, or the TMS34020.

Here's some useless information for you!!! The TMS34020 runs at 10MIPS and you would only need 50 of them in parallel to make a Cray II worried!!!!
Just for the processors alone this would cost £3500. If you spent a couple hundred grand on memory and around 50 grand on other things such as storage etc. then you would have a machine of equivalent power to the Cray II, but which would weigh in at alot less than the price of the Cray which is around 17million quid!!!!

The final piece of news is that I am more than half way through writing a windowing environment for 9938 systems which means it will run on the GENEVE, and a 4A with 80 column card.
It will be more suited to the GENEVE however, because of the additional speed, and standard mouse driver.

I thought I had better publicise my ideas after reading about the 4A windowing system that Mark Wills is going to produce.

The very first icon based system I had up and running was in about 1985 using just sprites as icons. I did intend to produce a whole range of programs based on this idea, but other things tended to get in the way like school and college work etc. This wasn't helped much by the fact that I don't have much experience in linking bit mapped graphics from assembly language!!! Therefore, I dropped the idea and I don't think many people saw the program running.

However, alot of people will have seen my text window routines, and my calculator which appeared in a couple of issues of 4Front (Where is Harry Pridmore!!!).
I had intended to incorporate both of these, but never quite got around to it.

After the years went by however, I found myself owning a GENEVE which had the speed, and features to cope with my ideas.
I now had a system that would run Extended BASIC at a decent speed, and would allow me to run a very powerful set of high res.(512 * 212) graphics commands (XHI).

XHI is written by Alexander Hulpke and provides some amazing video routines such as access to the V9938's hardware line, point and block move. This provides speed from an XB program that you would need assembly to do on a 4A. These routines are perfect for creating windowing systems as they allow windows to be moved at high speed from one screen location to another.

The other useful routines that are included are viewports which allow information do be displayed inside a window without overrunning the edges, and also access to sprite mode 2 which allows 8 sprites per line, 16 colours per sprite (stripey sprites!!!), and logical colour filtering on any lines of the sprite.

Unfortunately, without my GENEVE my development of the program slowed down and I could only test the program for the short time that Gary had the use of his GENEVE. Work has now been put on hold, but I have just ordered a new GENEVE from Texaments to keep me going until Richard can get my GENEVE back from Lou Phillips, which should be upgraded to 20MHz!!!

In the meantime here are a few of the features that are already working.

When the program loads there are a set of nine devices at the bottom of the screen which can be opened. These devices can be changed by the user and can be represented as hard disks, floppy disks, or even RAM disks which appears as a little picture of a chip!!!

After a device is selected, it is opened and the root directory is read. If there is not enough room to display all the icons on one screen, the scroll arrows can be used to move left, right, up or down the directory of icons.
Each icon definition specifies the type of file, the colour and pattern of the icon etc.

The program will load XB programs, EA Opt 5 programs, XB Batch files, and TDL routines. TDL is a little language I have written myself called Task Definition Language which is what is used to define the systems Pull Down Menus, but can also define routines that will run from icons.

The system will also recognize subdirectories, and will change into a subdirectory, and then open it if it is selected. If you click on an Artist picture then you current screen will be saved to RAMdisk preferably, and then the Artist picture will then appear on the screen. The same will happen for 256 and 16 colour Myart images!!!

If any of you read an issue of Personal Computer World earlier this year you may have seen the section they devoted to the new Apple Macintosh System 7 operating system. The main feature of this was that it would support aliased files. This means you can have an icon in a subdirectory for a file that does not appear in that directory. You can even have aliases for subdirectories, and other devices. This means that you could have an alias for an hard disk on a floppy disk. You could then jump straight from a floppy disk, to anywhere on the hard disk, or back again if you had the alias for the floppy!!!

The disadvantage with System 7 is that each aliased file takes up 2K. Mine take up 300 bytes!!! Using this aliasing system means that it would also be possible to make more than 127 files APPEAR!

in one subdirectory by keeping them somewhere else, but by keeping the first 127 files, and the aliases for the rest of the files in the same subdirectory!!!

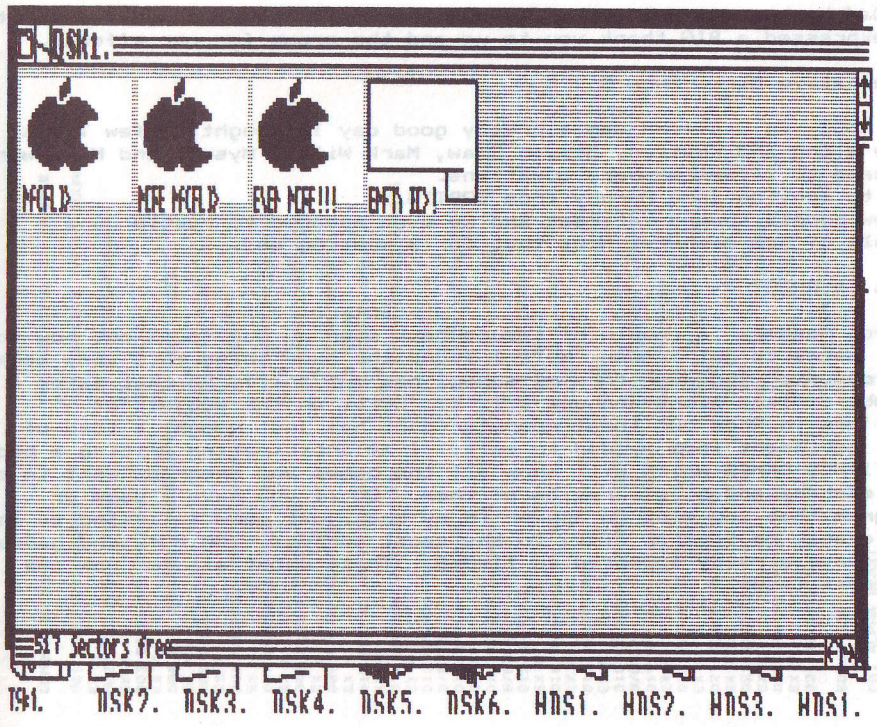
On top of this is a scrolling menu of sprites which contains functions such as calculator, standard disk catalog, trash, and some miscellaneous functions which are too detailed to explain here. You can see an example output of the screen which should give you an idea of what the program will look like when it is completed. It might look a bit bare, and there are one or two faults at that stage but I have tidied them up but now that Gary has been without his GENEVE for a while I have not had chance to redo a printout showing the next stage.

WATCH THIS SPACE!!!!

Well, that's enough waffle for now. I hope there's enough room in the newsletter for an article this long!!!!

All for now from Richard Twyning!!! If this makes it in time for the Christmas newsletter, Merry Christmas to All TI'ers. May all your presents be computer equipment!!!!!

WORKSPACE Version 0.1 (c)1991 Twyning Electronics



CASSETTE LIBRARY REPORT.... ....NICKY GODDARD

At last it is here in full black and white.
Yes the thing you have all been waiting for.... the cassette library list.
The list contains prices and a valid until date because unfortunately as I was searching for my TI*MES disk I came across another 12 disks.
All the programs are tested but if you have problems with your program just send it back and I will either refund your money or give you another tape.
I have no compilation tapes as yet but they will be appearing in the next list which won't be in TI*MES until next year.
Again if you have any original programs i.e. programs that are not in the library if you wish you can send then to me and I will send you a library tape of your choice for only postage costs.
If you order 2 programs before Febuary 1st you will get another one free.

TI-TALK..... ....NICKY GODDARD

The response to the GHANA appeal has been a bit slow with very few members showing any interest. We are very lucky in this country to have so much equipment for use in our schools and education establishments that we don't realise how people in other countries have to struggle So come on folks try and dig out something to donate.
My Dad has received a letter from Dr Van Meir not long ago and in it he expressed a BIG thank you to us and the group for our efforts.

Alternative Micro Show 5

The AMS5 in Stafford was a pretty good day I thought. A few TIers came along and chatted, Steven Shaw, Mark Wills, Myself and Mike were on hand to answer members questions.
Not to mention John Murphy from DORTIG.
A few people showed interest and we even managed to rustle up a couple of new members. Hows that for a meeting eh?.

TI's in RHOS

Do you remember a couple of issues back when my dad found out how the T.I. seems to get around. Well I found the same thing this very week. The computer engineer form my school has got a T.I. So that makes 4 in our village alone who use T.I.'s. Still on the subject of T.I.'ers in Rhos our local computer shop has just bought a load of T.I. equipment from my dad to sell. I am anticipating writing a book of TI*MES GAMES which will show all the games and programs sent in by the members and give a review of them. So if you have a game or program that you want to appear in my book just send a listing to me preferably on tape or disk, to the usual address. DO NOT send games that have already appeared in TI*MES. Thats about all for now see you next issue.

DEMONSTRATIONS
B K
D01 BELL MUSIC(AUTO)
EX K
D02 BELL MUSIC MANUAL
EX K
D03 BICOLOUR SPRITES
EX K
D04 BIG SPRITES DEMONSTRATION
EX K
D05 COLUMBIA
EX K
D06 COMPUTER PHYCHOTHERAPIST
EX K
D07 EXTENDED BASIC DEMO
EX K
D08 FLOATING SPRITES
EX K
D09 MAGIC SQUARES
EX K
D10 POCKET KALIEDOSCOPE
EX K
D11 RANDOM KALIEDOSCOPE
EX K
D12 RANDOM MUSIC
B/EX K
D13 RANDOM NUMBERS
EX K
D14 RANDOM NUMBERS II
B/EX K
D15 SHUTTLE
EX K
D16 SNOWFALL ON GANYMEDE
EX K
D17 SPRITIPED
EX K
D18 SQUASHED RANDOM NUMBERS
B/EX K
D19 BRITISH TELECOM TONES
EX K
D20 TIGERCUB DEMONSTRATION
EX K
D21 TIGERCUB PATTERNS
EX K
D22 TIGERCUB SPRITE DEMO
EX K
D23 WAVING AMERICAN FLAG
EX K
D24 10 SPRITE DEMONSTRATION
EDUCATION
E01 ALPHA BLAST
EX J SP(0)
E02 CHALLENGE BIG BOGGLE
B K
E03 COMPLETE TYPING COURSE
EX K
E04 CONSTELLATIONS
EX K
E05 FACES
B K
E06 HEART QUIZ
EX K
E07 I.Q. TEST
EX K
E08 JAWS
B K
E09 MATHS CALCULATOR
B/EX K
E10 MORSE CODE TUTOR
EX K
E11 MORSE TUTOR
EX K SP(0)
E12 MYSTERY SPELL
EX K
E13 SHAPE GUESSING GAME
B K
E14 TERRA
EX K SP(0)
E15 TOUCH TYPING TUTOR
B/EX K
E16 U.S.A STATES
B K
E17 WORD SPELL
EX K SP(0)
E18 WORD WORLD
EX K
GAMES
G01 AIRCRAFT LANDING
EX J
G02 AIRSTRIKE
EX J
G03 AIR TRAFFIC CONTOLLER
EX K
G04 ALIEN BUSTER
EX K
G05 ALIEN DESTROYER
EX K
G06 ALONE AT SEA
B/EX K
G07 ALPHABET/NUMBER PUZZLES
EX K
G08 ANTI AIRCRAFT GUN
B K
G09 APFELKOBOLD
EX K
G10 APPLE SCRUMPING
EX J
G11 ARCHEODROID
B/EX J/K
G12 ARIES
EX J
G13 ARTILLERIE
EX K
G14 ASTROATTACK
EX K
G15 ASTROMANIA
EX J
G16 ASTROWARS
B K
G17 ATTACK OF THE KILLER CRABS
EX K
G18 ATLAQUE SUR LA VILLE UNSURE
EX K
G19 BACKGAMMON
EX K
G20 BATTLE AT SEA
EX K
G21 BATTLE FRONT
EX K/J
G22 BATTLE STAR
EX K
G23 BEZERK
EX J
G24 BILLY BALL PLAYS CATCH
EX J
G25 BILLY BALL TO THE RESCUE
EX J
G26 BINGO
EX K
G27 BINARY GUESSER
EX K
G28 BIRD KNIGHTS
EX K
G29 BITE THE BLACKFISH
EX K
G30 BLACKBEARDS TREASURE
EX K
G31 BLACK BOX
B/EX K
G32 BLAST IT
B/EX J
G33 BOING
EX K
G34 BONKERS
B K
G35 BOSS'S CHRISTMAS PARTY
EX K
G36 BOWLS
EX K
G37 BREAKOUT
EX K
G38 BUDDY BALLON
EX K
G39 BURGLAR TIME
EX K
G40 CAR RACE
EX K
G41 CAVERN QUEST
EX J
G42 CHARGE
EX K
G43 CHENILLE
B K
G44 CITY ATTACK
EX K
G45 CIRCUS BALLOONS
EX K
G46 CLOSE ENCOUNTERS
EX K
G47 COLOUR MASTER
EX K
G48 COMPUTER CARD
EX K
G49 COMPUTER CRAPS
EX K SP(0)
G50 COME
EX K
G51 COPTAK
B/EX K
G52 COPTER CAPTIVE
EX K
G53 COUNTERFEIT COIN
B/EX K
G54 CRAZY CLIFF THE CLIMBER
EX J
G55 CRAZY CLIMBER
EX J/K
G56 CROSSWORD CHALLENGE
EX K
G57 CYBERNETIC DICE
EX K 32K
G58 DARTS
EX K
G59 DEACTIVATION
B K
G60 DIABLO
EX J/K
G61 DODGER
EX K
G62 DRAUGHTS
B/EX K
G63 DUCK
EX K

```
G64 EGGHUNT
EX J 32K(O)
G65 EVACUPOD
EX K
G66 FIRELADY
EX K
G67 FIRST AND LAST
EX J/K
G68 FOREST FIRE
EX K
G69 FOREST FIRE BASIC VERSION
B K
G70 FOREST RALLY
B K
G71 FORMULA ONE GRAND PRIX
EX K
G72 FOUR IN A LINE
B K
G73 FROG
EX J
G74 FROGGER
EX K
G75 FROGLET
EX J/K
G76 FREEWAY
EX J
G77 GEM GRABBER
B K
G78 GOLF
B K
G79 GRAPHIC PAIRS
B K
G80 HI-LO
B K
G81 HORSE RACING
B/EX K
G82 HUNTED
B/EX K
G83 IN A MAZE
EX K
G84 JACKPOT
EX K
G85 JEBS JOY RIDE
EX J
G86 JOYSKETCH
MM J
G87 JUMPER
EX K
G88 JUMPING JACK
EX K
G89 KALIEDOSCOPES
EX K
G90 KAMIKAZEE
EX J
```



```
G91 KNIGHTS BRIDGE
EX K
G92 KINGDOM
EX K
G93 KNIGHTS AND DRAGONS
EX K
G94 KRAZY KOALA
EX K
G95 KROAKERS
EX K
G96 LEAPER
EX J
G97 LEFT OR RIGHT
EX J/K
G98 L-GAME
EX K
G99 LIMBOPHOBIC LOONEY QUEST
B K
G100 LIGHT RACE
MM K/J
G101 LOST RUINS
EX BOTH JOYSTICKS K
G102 MAGIC SQUARE MAKER
B/EX K
G103 MARKET SIMULATION
EX K
G104 MARS MINE LANDER
B K
G105 METEOR
EX K
G106 MR.D
B K
G107 NAUTILUS
EX K
G108 NAVIGATOR
EX J
G109 NORBERT
EX K/J
G110 NOTENORTHY
EX K
G111 OCTAL
EX K
G112 OTHELLO 1/2 PLAYER OR
EX K      AGAINST COMPUTER
G113 PARANOID PAINTER
EX K
G114 PENGI
B K
G115 PEN THE PIG
EX K
G116 PESTEROIDS
EX J/K
G117 PHOTON ATTACK
B K
```



```
G118 PLANET LANDER
EX K
G119 POGOJUMP
EX J
G120 POKER
EX K
G121 PONGO
EX J
G122 PONTOON
B K
G123 Q*BONO
EX K
G124 RACING
EX K
G125 RAGING RIVER
EX K
G126 RAILWAYS
EX K
G127 RALLY CROSS
B K
G128 ROBIN HOOD
EX K/J
G129 ROLL 5
EX K
G130 RUBICS CUBE
EX K
G131 SHARKS
EX J SP(O)
G132 SKY BUSTER
B/EX K
G133 SLAVE LABOUR
B K
G134 SNAIL-MAN
B K
G135 SNOW TREK
B K
G136 STAR PROBE 9
B K
G137 STRATEGIC COMMAND
EX K
G138 SUB STRIKE
EX K
G139 SUMERIA
B K
G140 TEN PIN BOWLING
EX K
G141 THE MUNCH
B J/K
G142 THE WALL
EX J/K
G143 TI COMBAT
B/EX K
G144 TIME TRAVELLER
B K
```

```
G145 TIRS CROISERS
EX K
G146 TRAFFIC COP
EX J
G147 VICIOUS CIRCLES
EX K
G148 VICTORIAN SEWERS
EX K
G149 WAGON WHEEL
B K
G150 WALLABY
EX J
G151 WEBSTER DINES OUT
EX K
G152 WERMBERMER
B/EX K
G153 WONKAPILLAR
EX K
G154 WYVERN
EX J
G155 YAHTIEEE
B K
G156 3D MAZE
B/EX K

MUSIC

M01 ABIDE WITH ME
EX
M02 AWAY IN A MANGER
EX
M03 BACHS INVENTION IN F
EX
M04 BACHS PRELUDE E IN C
EX
M05 BEER BARREL POLKA
EX
M06 BIG CAT BOOGIE
EX
M07 BLESSED ASSURANCE
EX
M08 CAMPTOWN RACES
EX
M09 CAROLINA MOON
EX
M10 CHURCH IN THE WILDWOOD
EX
M11 DANCING STICKMAN
EX
M12 DIXIE
EX
M13 DOGCATCHER
EX
```



```
M14 ETERNAL FATHER
EX
M15 FAME
EX
M16 GHOSTBUSTERS
EX
M17 GOD REST YE MERRY GENTLEMEN
EX
M18 GRAYMOUSE ROCK
EX
M19 GREENSLEEVES
EX
M20 HAYDYNS SONATA #2
EX
M21 HOLY HOLY HOLY
EX
M22 I LOVE TO TELL THE STORY
EX
M23 I'M FOREVER BLOWING BUBBLES
EX
M24 I NEED THEE EVERY HOUR
EX
M25 JESUS LOVES THE LITTLE
EX          CHILDREN
M26 JESUS SAVES
EX
M27 JUST A CLOSERWALK WITH THEE
EX
M28 LET IT SNOW
EX
M29 LOVE LIFTED ME
EX
M30 MAPLE LEAF RAG
B/EX
M31 MIDNIGHT COWBOY
EX
M32 NOEL
EX
M33 PENNSYLVANIA POLKA
EX
M34 RUDOLPH
EX
M35 SLIEGH RIDE
EX
M36 SOFTLY AND TENDERLY
EX
M37 STAR SPANGLED BANNER
B/EX
M38 TAKE FIVE
EX
M39 THE LORDS PRAYER
EX
M40 THE MASTER PIECE
EX
M41 THE OLD RUGGED CROSS
EX
M42 THE OLD TIME RELIGION
EX          MEDLEY
M43 VARIATIONS ON A THEME
EX
M44 WE HAVE NO BANANAS
EX
M45 WHERE HE LEADS ME
EX
M46 WHY ME
EX
M47 XXX CERTIFICATE(MEN)
EX
M48 YESTERDAY
EX
M49 3 TIME A LADY
EX

UTILITIES

U01 ADDRESS LABEL MAKER
EX P
U02 APHORISM GENERATOR
EX K
U03 BANNER DESIGN
EX P
U04 BARGRAPH PRINTER
EX K P
U05 BASKET BALL SCORE KEEPER
EX K
U06 BIORTHYMNS
B K
U07 CASSETTE FILES UTILITY
EX K
U08 CENTAL HEATING CALCULATOR
EX K
U09 CHARACTER CONSTRUCTOR
B/EX K
U10 CHEQUE BOOK
EX K
U11 CLOCK
EX K
U12 CODE BREAKER
B/EX K
U13 COMPUTERISED CALENDER
B/EX K P
U14 CREDIT UNION LOAN ACC
EX K
U15 DISSASEMBLER
B K
U16 ELECTRONIC TYPEWRITER
EX K
U17 ENCRYPTOR
EX K
U18 FILER 99ER
EX K
U19 GOLF SCORE ANALYSIS
EX K
U20 GOLF SCORES IN 1900
EX K
U21 GRAPH MAKER
EX K
U22 GRAPH PLOTTER
B/EX K
U23 HOME BUDGET
EX K
U24 HOUSE FINANCE SYSTEM
EX K
U25 IDEAL WEIGHT
EX K
U26 INTERRUPT DRIVEN CLOCK
EX K **SEE FOOTNOTE
U27 LABEL MAKER
EX K P
U28 LIFE SPAN
B K
U29 MATH CONVERSION
EX K
U30 MAZE MAKER
EX K P
U31 MINI FORMATTOR.
EX TI-WRITER
U32 MINI WORD PROCESSOR
EX P
U33 N.A.I.A. BASKETBALL STATS
EX K
U34 PERPETUAL CALENDER
EX K P
U35 PLANNING CALENDER
EX K P
U36 PRINT DV80 FILE
EX K P
U37 PROGRAM LISTER
EX K 32K P
U38 SCREEN DUMP
B MM
U39 SCREEN MAP
EX K
U40 SCREEN SCROLL UTILITY
MM
U41 SELF EVALUATION
EX K
U42 SINGLE PIXLE DRAWING
EX K
U43 SPRITE EDITOR
EX K
U44 SYMMETRICAL CHAR GENERATOR
B/EX K
U45 TAX KEEPER
EX K
U46 TELEPHONE COST MONITOR
B K
U47 TEXT WRITER
EX K
U48 THANK YOU NOTE PROGRAM
EX K P
U49 T.I. TEXT
EX K
U50 TITLE PAGE MAKER
EX K P
U51 TI WRITER FONT MAKER
EX K
U52 T.V. TITLES
EX K VIDEO RECORDER
U53 UNIVERSAL LIFE
EX K
U54 VIDEOTAPE DATABASE MAKER
EX K
U55 WILL MAKER
B/EX K
U56 WORD PROCESSOR (V.GOOD)
B P
U57 2 WAY SCREEN PRINTER
EX K
```

** WHEN SETTING INTERUPT DRIVEN CLOCK SET IT 40 SECS FAST BECAUSE THE COMPUTER CHECKS BEFORE STARTS

KEY TO PROGRAM NEEDS

B = BASIC

EX = EXTENDED BASIC

MM = MINIMEM

SP = SPEECH

32K = 32K MEMORY

(O) = OPTIONAL

J = JOYSTICK

K = KEYBOARD

P = PRINTER.

VALID UNTIL 1:6:1992

80P IF YOU SEND A CASSETTE

£1 IF YOU DO NOT

CASSETTE LIBRARY PRICES

TI USER GROUP UK SOUTHERN MEETING 1992. MIKE GODDARD.

A venue for a southern meeting has been arranged for SATURDAY 1st of February 1992 at OLDFIELD PRIMARY SCHOOL in MAIDENHEAD BERKSHIRE, with the generous assistance of Martin Ross and his wife Betty. A detailed map of the location with directions for both rail and car travellers has been provided by Martin. Light refreshments will be provided and the location is central enough to Maidenhead for anything else that may be reqiured.

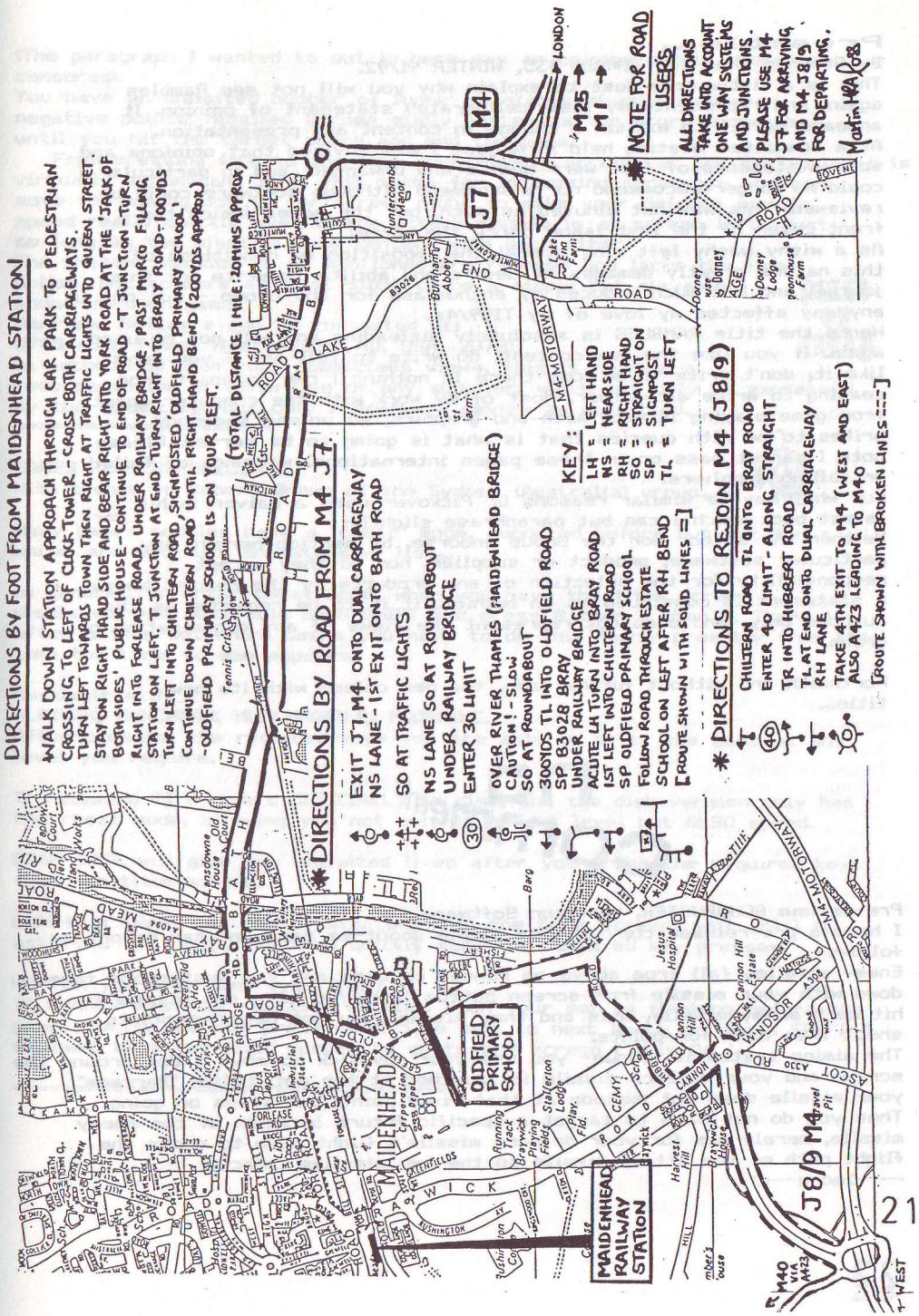
We hope as many members as possible will attend and those of you who are close enough will bring along systems, to save confusion will those who are bringing systems please notify me so that power points can be provided and if any Trade stands are required please let me know, there will not be a charge for these but planning for the room will have to be arranged. As for purchasing equipment from myself please let me know in advance if anything specific is required as because of the distance involved I will not be bringing a lot of stuff with me unless it is required.

If anybody has any other ideas or plans or indeed is willing to demonstrate any particular aspect of the TI-99/4A or derivatives then they will be more than welcome. Thats about all from me I hope to see some old and possibly new friends on the day.

FCTN/QUIT

Mike

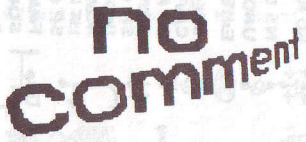
An admission fee of £2 will be charged to participants to help defray costs. Ed.



Preamble...

By Stephen Shaw for TI*MES #35, WINTER 91/92.
This is a little note just to explain why you will not see Rambles again, and represents my last (deliberate!) statement of opinion. It appears merely to explain a change in content and presentation.
At a Committee meeting held in August I was advised that opinions and some statements of fact were unwise and unwanted, and in particular I could no longer recommend nor discommend although software could be reviewed. This was not minuted as such, but the notes inside the front cover of the last issue carry the kernel.
As a wishy washy left wing liberal the imposition of conditions of this nature greatly damages my desire and ability to write for this journal and has much reduced my enthusiasm for the Group - but not in any way affected my love of my TI99/4a.
Hence the title RAMBLES is absolutely misleading and will not be used again. If you like the new content, do write to tell me! If you don't like it, don't write to me 'cos there is nothing I can do apart from ceasing to write altogether. Most of my work with the computer apart from game playing is with math and graphics, so unless somebody writes to me with queries that is what is going to be served. Please note I cannot pass on in these pages international warnings of failed or failing suppliers.
For what may be similar reasons Dr Pickover uses a waiver in his latest book which I can but paraphrase slightly:
Neither the author nor the Group endorse, by mention herein, any particular software, product or supplier, nor do they accept responsibility for the selection of any products by the reader. Where a statement is construed as an opinion, it is to be regarded as the authors only, and does not represent the views of any organisation or group.

Forthwith and without further ado, the new column with its new title...



Pre-review: SCUDBUSTER, Harrison Software.
I have a pre-release copy of this simple shoot-em game and can report as follows-
Enemy missiles fall from above on random courses and you must shoot them down with your missile from screen bottom left. You have one missile to hit each enemy missile, miss and they hit and you get negative points, shoot them down for points.
The aiming system is unique- you use your joystick to move a dot around the screen and your defence missile is launched at the dot. Unlike "Barrage" your missile does not explode at that fixed point, but keeps on going. Thus you do not have to target a specific future location of the enemy missile, merely aim for your defence missile's flight path to cross the flight path of the attack missile so the two missiles coincide.
---->more--->

(The paragraph I wanted to put in here was an opinion and is therefore censored).
You have an unlimited number of lives- missing a missile produces negative points! Missiles become mildly faster as the score increases until you hit the target score and end the game.
From pressing fire to the time your missile arrives at the marked point is virtually a constant time. This makes for interesing strategy- you can move the aiming dot to far screen right to make your defence missile speed really quickly even if the attack missile is at screen left- just so long as the missiles make contact with each other.
The present version is as mentioned pre-release and lacks the final polish, but appears at first sight to be a simple but playable shoot-em-down game.
Harrison Software can be contacted at:
5705 40th Place, Hyattsville, MD, USA, 20781.
In accordance with your committees views, this is neither a recommendation or otherwise of any supplier, which may only be expressed by the committee, but only a software review as permitted.

PEACE ON EARTH

==========

GAMES PLAYERS SPECIAL...
With thanks to Robert Brown of the Sydney (Australia) group...

Many games modules have a "cheat" mode, reported before but not for a while, and not such an extensive list...

In general, to enter cheat mode, when you have the title screen displayed, you must hold down SHIFT and press 8 then 3 then 8. There is often a time-out which means you have to do this fairly promptly from the time the title screen appears.

Available with:
ALPINER, MUNCHMAN, MUNCHMOBILE, MOONMINE.
After pressing the required keys you are able to enter the game at any level you require.

TI released TI Invaders OFFICIALLY on disk, and the disk version only has the cheat mode, allowing you not only to select level but ALSO speed.

STAR TREK will give you unlimited lives after you press the required keys from the title screen.

CHISHOLM TRAIL, instead of starting from the title screen, wait until it asks you for LEVEL 1-9 then quickly make the required key presses.

HOPPER you only need type SHIFT 8.
TI RUNNER was only released on disk. When your man is flashing before the game starts, you can press FCTN 5 to skip to next level; or press SHIFT 3 followed by a two digit number to go to any screen (6 is 06).

==========

May I remind you any program EVER published by Regena, in ANY publication, is available on disk or tape for just US$4 from the author.
Additionally the following books are available from Regena:
Programmers Reference Guide to the TI99/4A. 358 pages. US$15.00
First Book of TI Games. (29 listings). US$13.00
Basic Programs for Small Computers (including TI99/4A) (40 programs). US$13.00
and on disk...

ALL programs by Regena published in Micropendium in 1989 for $5-(One disk):
San Diego Guide, 12 cake recipes with print option, printing random math problems, knit a sweater for 25 to 35" chest using different guages, high school science multiple choice test; Four card solitaire, Magic Boxes math puzzle; Presidents of the USA, learning South American geography, White Christmas greetings "card".

ALL programs by Regena published in Micropendium in 1990 for $10-(Two disks):
Tour of Tucson, Dice game-Yacht; Card patience-Pyramid, Plane Geometry Postulates and Theorums, Learning to read lower case letters; Multiply and Divide math tests; Books of the Bible test; Fairisle pattern design, The 12 days of Christmas; Scripture Quiz.

Disk of 16 game programs (see below) $10
Disk of 17 math programs $10
Disk of 10 music demo programs $10.

For all of the above, please add something extra for overseas postage!!!!!

REVIEW:
REGENA: DISK OF 16 GAME PROGRAMS FOR TI99/4A
Cost US$10 plus overseas postage, say $2:
From: Regena, 918 Cedar Knolls West, Cedar City, Utah, USA, 84720
Sixteen programs on disk for $10 can't bee bad! This disk contains some golden oldies, some brand newies...
Solitaire is the peg jumping game where you need to leave one peg in the middle. The program is a nice simulation for speed of entry, and for the ability to progressively take back moves. You can also play back your moves and print the details.
Yacht is the old dice game which some sharp games company snazzied up, made a minor name change, and made millions of dollars... this is the ORIGINAL which differs a little from the commercial version.
Pyramid is a card patience game, which I have not been able to make work out- my best is to have four cards left stranded!
Fourcard is an easier card patience.
There are three "pick a pair" memory games, differing in the graphics and the number of pairs- housework mixup, concentration, and match-em.
Colour Code is not unlike Mastermind, while FlipFlop bears an uncanny resemblance to that other old game tarted up and sold for lots of money-the original game was called Reversi. In this version the computer plays rather dumbly!
TicTac is ordinary noughts and crosses, boxes is a math puzzler, grid is a "find the bomb" type puzzle (if you need more than 6 guesses, sharpen up!). ---> continued

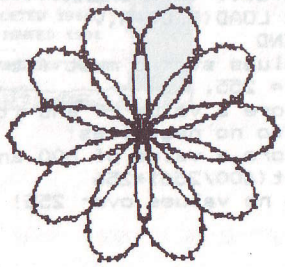
Superchase is a maze in which you collect treasures while a dumb robot bumbles after you. Hidden maze is an invisible maze for you to move out of. And Closeout is not unlike the Stainless Software program "Man and Monsters" in which you cover several levels of floors avoiding monsters which are after you.
And Poker Solitaire is very like the Pewterware program "Challenge Poker" but without the wild card! You arrange cards in a 5x5 grid to try to score poker hands.
Overall a good collection of programs with something for everyone. Everything will run in TIBasic, and only two require VDP or BXB to run in Extended Basic.
Worth looking at.
Note: This is a software review only and is not to be taken as a recommendation to purchase from any particular supplier. This waiver is also not to be read as having any negative connotation. Such views are restricted to your committee.
==========
Last year I reviewed a book by Clifford A Pickover entitled COMPUTERS, PATTERN, CHAOS AND BEAUTY. I recently advised that a new book was coming out...
I do not know if anyone else purchased (or borrowed) the first book, and I do not have a dispensation from the committee to review books, so here is an extract from the publishers announcement:

COMPUTERS AND THE IMAGINATION. Clifford A Pickover.
"...a treasure of breathtaking computer graphics, and startling applications of computer science to art, music, poetry, science technology, and the mystery of creativity" [Martin Gardner, Scientific American].
"...Chaos theory and fractals...have shown the revolutionizing role of the visualisation of complex mathematical data. Computers and the Imagination pushes the adventure one step further. ... includes a range of topics from the how-to construction of artificial spider webs to pain inducing patterns to computer generated poetry...."
casebound, 416 pages, with 296 mono illustrations and 8 in colour.
Publisher: Alan Sutton Publishing.
ISBN: 0 86299 999 5
Price: Around £27.00

==========
EXTENDED BASIC TIP

When programming in extended basic, you do NOT have as much room as you may need for some exotic programs requiring large dimensioned arrays. Just try a program with one line - DIM P(8000) and run it!!! It is however possible to have an effective array of 8000 cells (or less of course) as demonstrated below...

```
1 ! HOW TO HAVE A NUMERIC ARRAY
2 ! OF 8000 VALUES
3 ! WITHOUT GETTING
4 ! OUT OF MEMORY
5 ! (just try DIM A(8000)!
6 !
7 ! S SHAW  OCT 1991
8 ! STOCKPORT   ENGLAND
9 !        -----> continued -->
```



```
11 ! USES 8K LOW MEMORY
12 ! IN 32K RAM EXPANSION
13 ! which is required
14 ! and assumes no machine code is loaded!
15 !
16 ! set up with CALL DIM
17 ! and used with
18 ! CALL S(cell#,value) to store
19 ! CALL R(cell#,value_out) to read value.
20 ! eg instead of A=B(254)
21 ! use CALL R(254,A)
22 !
96 ! call init allows access to 32k ram
97 ! only needed once, then even after RUN "DSK1.P" provided you
98 ! dont use it again, all the stored values are available
99 ! to program P.
100 CALL INIT
110 PRINT "SIZE OF ARRAY?" :: INPUT "FROM 0 TO...?":HIGH
120 CALL DIM(HIGH)
121 ! line 120 stores 0 values in the necessary addresses
122 ! only need to use once or to reset whole array
130 INPUT "ARRAY CELL NUMBER?":ADR :: IF ADR>HIGH THEN 130
140 INPUT "Read OR Save?":A$ :: IF POS("RS",A$,1)<1 THEN 140
150 PRINT
160 IF A$="R" THEN 180
170 INPUT "ARRAY CELL VALUEA?":V :: IF V>250 THEN 170
180 IF A$="R" THEN CALL READ(ADR,V):: PRINT "VALUE IS ":V ELSE
  CALL SAVE(ADR,V)
190 PRINT "------------"
200 GOTO 130
210 SUB DIM(N)
211 ! fill array with 0 values in blocks of 10 cells for speed.
220 IF N>8100 THEN DISPLAY AT(24,1)BEEP:"ARRAY TOO LARGE" :: BREAK
230 START=8200
231 ! we have available addresses 8192 to 16384 actually!
240 FOR T=0 TO N/10+2
250 CALL LOAD(START+T*10,0,0,0,0,0,0,0,0,0,0)
260 NEXT T
270 SUBEND
280 SUB READ(A,V):: A=ABS(A)
290 IF A>8100 THEN BREAK
300 CALL PEEK(8200+A,V)
310 SUBEND
320 SUB SAVE(A,V):: A=ABS(A):: V=ABS(V):: IF V>250 THEN BREAK
330 IF A>8100 THEN BREAK
340 CALL LOAD(8200+A,V)
350 SUBEND
351 ! values stored must fit into one byte hence maximum value
  2^8-1 = 255.
352 ! store a value -9 and it reads as 247 which is
  256-9 so no negatives!
353 ! store a value of 600 and it reads as 88, which is
  600-int(600/256)*256
354 ! so no values over 255!
355 !
360 END
```


While there are restrictions- values stored must be positive and below 256- these limitations are usually not exceeded in most arrays.

The above program uses the 8k low memory block which is not used by Extended Basic, unless you are running some machine code routines. If you can write machine code for ExBas you don't need this tip...

You are not limited to single dimensioned arrays, but will need to formulate an equation to address second and higher dimensions....

For example, a two dimensioned array of 30,30 could be used in a form R(A,B) where memory location = 8200+A*30+B and so on and so on.

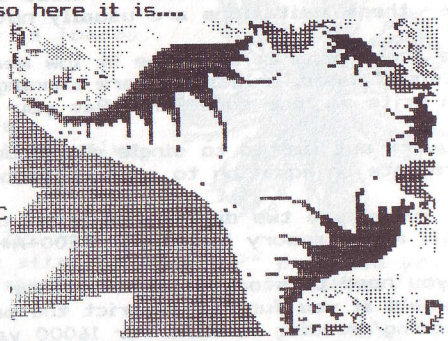
If you need to store values for more than 8000 cells, then this is possible if you further restrict the possible values and construct an encoding/decoding formula. For 16000 values, you could store values from 0 to 15, with two values per memory location (technically speaking, one value stored in the low nibble and one in the high nibble). The ultimate in storage is at bit level, where you can store 8 values per memory location for a total array of 64000 cells, more than enough for anyone, although restricted to 0 and 1 values only.

==========

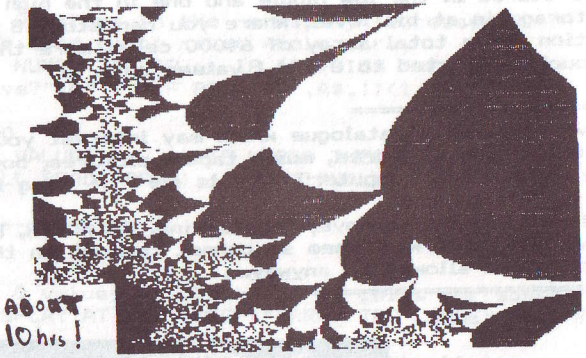
I have received a catalogue which may interest you, containing as it does Fractal badges, T-Shirts, music tapes, pictures, books (and software for some odd none-TI computer!). If this sort of thing interests you send an SAE for a copy to:
Frachaos, Higher Trengove, Constantine, FALMOUTH, Cornwall, TR11 5QR
I have not dealt with them so cannot comment on their service, but then again I'm not allowed to anyway....

==========

In the last issue we gave a number of programs which counted to 99999 but omitted the listing for TI LOGO so here it is....

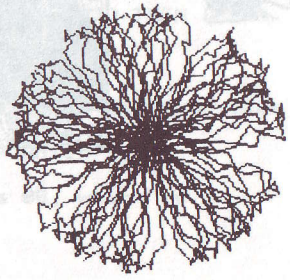


```
PROCEDURES
----------

TO START
; P_99999
; TI - 99 / 4A LOGO PROCEDURE
; WESLEY R. RICHARDSON DEC 1987
; BLUEGRASS 99 COMPUTER SOCIETY, INC
; LOGO RUN TIME 38 MIN 28.8 SEC
CS COUNT
END

TO COUNT
TEST CNT = "Y IFT CS PT 44 15 12 VL COU NT IFF STOP
END
```




```
TO CNT
PT 67 11 16
PT 79 12 16
PT 85 13 16
PT 78 14 16
PT 84 15 16
PT 63 17 16
PT 89 19 16
PT 78 20 16
MAKE "K RC
OUTPUT :K
END

TO VL
MAKE "V 48
REPEAT 10 [PT :V 13 12 WL MAKE "V :V + 1 ]
END
```

```
TO WL
MAKE "W 48
REPEAT 10 [PT :W 14 12 XL MAKE "W :W + 1 ]
END

TO XL
MAKE "X 48
REPEAT 10 [PT :X 16 12 YL MAKE "X :X + 1 ]
END

TO YL
MAKE "Y 48
REPEAT 10 [PT :Y 17 12 ZL MAKE "Y :Y + 1 ]
END

TO ZL
MAKE "Z 48
REPEAT 10 [PT :Z 18 12 MAKE "Z :Z + 1 ]
END
==============================
```



As requested in the last issue, something on Pilot...

PILOT 99 DEMO
by WESLEY R. RICHARDSON
BLUEGRASS 99 COMPUTER SOCIETY, INC.

Additions by S Shaw.
PILOT 99 was written by Thomas P. Weithofer and is being distributed as Fairware for $10 - donations to your national Cystic Fibrosis charity. PILOT 99 is an adaptation of PILOT for the TI-99/4A. Additional features have been included for using sprites, bit-map graphics, sound and files. PILOT 99 was written in FORTH, although the user does not need to know FORTH to be able to use it.

PILOT99 is on two disks, and is available from your user group disk library - send five pounds or two disks plus three pounds to S Shaw. The demo listed below is a demonstration of some of the commands in PILOT 99. Many of the ideas were adapted from the examples in the PILOT 99 documentation and have been combined into one program.
To key in the program, use the Editor/ Assembler editor and save the file as a DIS/VAR 80 file. To run the program, use the Editor/ Assembler and put the PILOT 99 program disk in drive 1.
Use option 3 to Load and Run, and give DSK1.PILOT as the file name.
At the program name prompt, put the disk containing the P:PILOTDMO, that you keyed in, in drive 1, and type DSK1.P:PILOTDMO and then press enter.
At the printer prompt, just press enter. CAUTION: the program P:PILOTDMO will create a file TEST1 on disk 1. If you have another file by that name, be sure to change the filename in the demo.

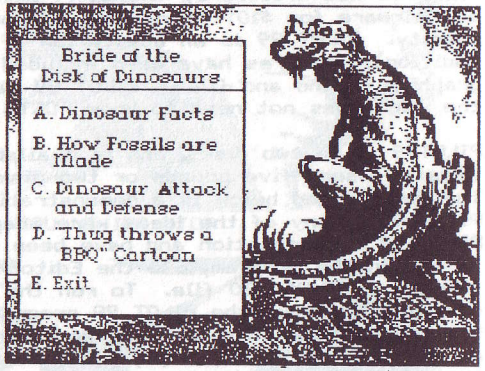
In my [WR] opinion PILOT 99 shows that a great deal of effort and knowledge was used in developing it, however, there are some problems. Differences between the documentation and the way in which program lines are interpreted while running often create routines which do not execute as planned.
When an error is encountered, the statement number with the error is printed, but there is no way to test variables at that point to determine the cause of the error. The disk drive appears to come on each time a line is read, so it seems like the disk drive runs almost the entire time the program is executing.
The positive aspects of PILOT 99 are that it demonstrates the power of FORTH and shows disk accessing and string manipulations that can be done. To see how it was done you would need the source code. I do not believe that PILOT 99 in its present form will catch on as a popular programming language, but I do feel that PILOT 99 could be an excellent instructional example.

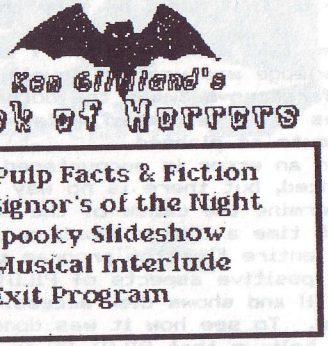
The PILOT 99 author, Thomas P. Weithofer, died in April, 1986 from complications associated with cystic fibrosis.

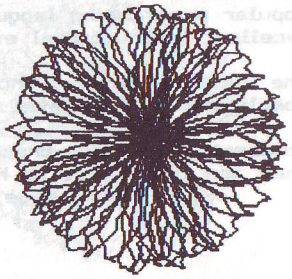
[comments in square brackets like this must be ignored when keying in, they are there just to help you follow the code!]

------> next page ---->>

```
R: P:PILOTDMO PILOT 99 DEMO APRIL, 1986
R: WESLEY R RICHARDSON [ R: = Remark]
R: BLUEGRASS 99 COMPUTER SOCIETY, INC.
IT:
T: PUT PILOT-DEMO IN DRIVE 1
T: TYPE ANY TEXT [ T: = Type = Print]
A: $B [ A: = Answer = Input]
T: $B
OF: DSK1.TEST1 [OF: = Open File]
T: WRITE TO DISK 1
WR: BLUEGRASS 99 COMPUTER SOCIETY
RF: 0
T: READ DISK 1
RE: $C          [RE: = READ file]
CF:             [CF = close file]
TC: 9,2
T: $C
T: MAKE A SPRITE MOVE
GP: 1,1898,FF3D,3C3C,E404
SP: 3,1
SC: 3,14        [SC = Sprite Colour]
SL: 3,100,125 [SL = Sprite Location]
SM: 3,10,15  [SM = Sprite Motion]
SS: 2           [SS = Sprite Speed]
LP: 500
EL:
SG:
C: #N<-5
T: DEMONSTRATE SUBROUTINES
LP: 9           [LP = loop to EL n times]
C: #N<-#N+1
U: *CALC
T:
EL:             [EL= End Loop]
T: CALC FINISHED
LP: 200
EL:
J: *GRAPH       [J = Jump = Goto]
*CALC
C: #A<-SQR(#N) [make A=SQR(N)]
C: #B<-#N^2  [make B=N*N]
T: N IS #N    [use of numeric variables]
T: SQR(N) IS #A
T: N^2 IS #B
E:
*GRAPH
IG:
T: GRAPH MODE
T: DRAW 6 CIRCLES
C: #N<-50       [make N=50]
LP: 6           [loop 6 times]
DC: 55,125,#N [draw circle centre 55,125 radius N]
C: #N<-#N-7
 ----> a bit more on next page ----->
```





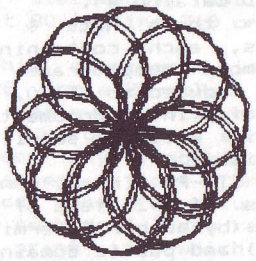



```
EL:
TG: 15,1,DRAW DIAGONAL LINE
DL: 15,15,80,100
LP: 200
EL:
IT:
E:
```

[this bit by s shaw:]
Historically PILOT was a tutorial language reading an input, checking it against a series of answers, and incrementing a counter if the answer matched. The commands were therefore either unconditional (always do...) or conditional upon a match (do if true, do if false). The magic of Pilot lay in the checking of answers against a data list, and the fairly easy conditional reactions to the answers. Very primitive Pilots were able to either Match (Y matches Yellow) and Match Exactly.
Stephen Shaw wrote a simple Pilot interpreter in Extended Basic way back in 1982 and this is also in the Group library with docs on disk SSPD1.

Here is an extract from those docs:
Note that these commands only apply to THIS version of Pilot!

```
T/HELLO WHAT IS YOUR NAME?
A/@NAME
T/HI @NAME
T/LIKE TO DO THAT AGAIN?
ME/Y;YES;YEP;OK;O.K.;YEAH;
RY/          [no outstanding JUMP so return      ]
             [will RUN program again from line 1]
T/YOU ANSWERED IN THE NEGATIVE!
T/LETS USE A SUBROUTINE...
J/*GOODBYE
T/THATS ALL FOLKS
E/
*GOODBYE
T/IT HAS BEEN GOOD TALKING WITH YOU
T/PLEASE PLAY AGAIN
R/
E/
```



==========

With reference to TI*MES #34 page 43, and the super fast program by Dr H B Philips to find the first 100 prime numbers, I have an excellent explanation of it from member Walter Allum, and would be happy to send a copy to anyone interested -it is rather lengthy! - in essence Phillips has used several variations on the Sieve of Eratosthenes.

==========

No. 63

Tigercub Software
156 Collingwood Ave.
Columbus, OH 43213

****

My stock of Tigercub Software catalogs is depleted and it would not pay me to reprint it. Therefore I have released all copyrighted Tigercub programs, except the Nuts & Bolts Disks, for free distribution providing that no price or copying fee is charged. All of my Tigercub programs have been added to my TI-PD library and are cataloged, by category, in TI-PD catalog #4.

My three Nuts & Bolts disks, each containing 100 or more subprograms, have been reduced to $5.00. I am out of printed documentation so it will be supplied on on disk.

My TI-PD library now consists of 492 disks of fairware (by author's permission only) and public domain, all arranged by category and as full as possible, provided with loaders by full program name rather than filename, Basic programs converted to XBasic, etc. The price is just $1.50 per disk(!), post paid if at least eight are ordered. TI-PD catalog #4 with Supplement #1, listing all titles and authors, is available for $1 which is deductible from the first purchase.

Several articles have been published on the subject of using Funlweb as a simple fixed-field data base. Sometimes you might want to rearrange the sequence of fields in such a file. This mini-program will quickly change the position of any field in a D/V80 file.

```
100 DISPLAY AT(3,8)ERASE ALL
:"FIELDSWITCHER":"":"    by
Jim Peterson":"":"  To chang
e sequence of     fields in
a DV80 fixed fieldfile creat
ed by Funlweb or  other mean
s"
110 DISPLAY AT(23,6):"PRESS
ANY KEY" :: DISPLAY AT(23,6)
:"press any key" :: CALL KEY
(0,K,S):: IF S=0 THEN 110 EL
SE CALL CLEAR
120 DISPLAY AT(8,1):"FILENAM
E? DSK" :: ACCEPT AT(8,14):F
$
130 OPEN #1:"DSK"&F$,INPUT
140 DISPLAY AT(12,1):"MOVE F
IELD STARTING AT WHAT POSITI
ON?" :: ACCEPT AT(13,11)VALI
DATE(DIGIT):N
150 DISPLAY AT(15,1):"LENGTH
 OF FIELD?" :: ACCEPT AT(15,
18)VALIDATE(DIGIT)BEEP:L
160 DISPLAY AT(17,1):"TO WHA
T POSITION?" :: ACCEPT AT(17
,19)VALIDATE(DIGIT)BEEP:T
170 IF T>N+L-1 OR T<N THEN 1
90
180 CALL SOUND(400,110,0,-4,
0):: DISPLAY AT(23,1)BEEP:"C
ANNOT MOVE FIELD WITHIN ITSO
WN PARAMETERS!" :: GOTO 140
190 DISPLAY AT(19,1):"OUTPUT
 FILENAME? DSK" :: ACCEPT AT
(19,21)BEEP:OF$
200 OPEN #2:"DSK"&OF$,OUTPUT
210 LINPUT #1:M$ :: M$=M$&RP
T$(" ",80-LEN(M$)):: IF T<N
THEN M$=SEG$(M$,1,T-1)&SEG$(
M$,N,L)&SEG$(M$,T+1,N-T)&SEG
$(M$,N+L+1,255)
220 IF T>N THEN M$=SEG$(M$,1
,N-1)&SEG$(M$,N+L,T-N-L)&SEG
$(M$,N,L)&SEG$(M$,T+1,255)
230 PRINT #2:M$ :: IF EOF(1)
<>1 THEN 210 ELSE CLOSE #1 :
: CLOSE #2
```



```
240 DISPLAY AT(12,1)ERASE AL
L:"ANOTHER? Y/N" :: ACCEPT A
T(12,14)VALIDATE("YN")SIZE(1
)BEEP:Q$ :: IF Q$="Y" THEN 1
20 ELSE CALL CLEAR :: STOP
```

And this one will make it easy to completely rearrange the sequence of any number of fields.

```
100 DISPLAY AT(3,9)ERASE ALL
:"REARRANGER":"":"      by Ji
m Peterson"
110 DISPLAY AT(7,1):"  To re
arrange the sequence of fiel
ds in a DV80 file of fixed f
ields created by     Funlweb
 or otherwise."
120 DISPLAY AT(24,7):"PRESS
ANY KEY" :: DISPLAY AT(24,7)
:"press any key" :: CALL KEY
(0,K,@):: IF @=0 THEN 120
130 DIM L(20),S(20),F$(20)::
 CALL CLEAR
140 DISPLAY AT(8,1):"INPUT F
ILENAME?":"":"DSK" :: ACCEPT
 AT(10,4)BEEP:I$ :: OPEN #1:
"DSK"&I$,INPUT
150 DISPLAY AT(12,1):"OUTPUT
 FILENAME?":"":"DSK" :: ACCE
PT AT(14,4)BEEP:O$:OPEN #1:"
DSK"&O$,OUTPUT
160 DISPLAY AT(16,1):"HOW MA
NY FIELDS?" :: ACCEPT AT(16,
18)VALIDATE(DIGIT)SIZE(2):F
:: CALL CLEAR
170 FOR J=1 TO F :: DISPLAY
AT(12,1):"FIELD #";J;"LENGTH
?" :: ACCEPT AT(12,20)VALIDA
TE(DIGIT)BEEP:L(J):: NEXT J
:: FOR J=1 TO F
180 DISPLAY AT(12,1):"IN FIE
LD #";J:"":"PLACE FIELD #" :
: ACCEPT AT(14,15)VALIDATE(D
IGIT)BEEP:S(J)
190 IF S(J)<1 OR S(J)>F THEN
 CALL SOUND(300,110,0,-4,0):
: GOTO 180
200 IF POS(E$,CHR$(S(J)),1)=
0 THEN E$=E$&CHR$(S(J)):: GO
TO 220
210 CALL SOUND(300,110,0,-4,
0):: DISPLAY AT(16,1):"FIELD
 #";S(J);"HAS ALREADY  BEEN
PLACED!" :: GOTO 180
220 NEXT J
230 LINPUT #1:M$ :: M$=M$&RP
T$(" ",80-LEN(M$)):: P=1 ::
FOR J=1 TO F
240 F$(J)=SEG$(M$,P,L(J)):: 
P=P+L(J):: NEXT J
250 FOR J=1 TO F :: N$=N$&F$
(S(J)):: NEXT J :: PRINT #2:
N$ :: N$=""
260 IF EOF(1)<>1 THEN 230 EL
SE CLOSE #1 :: CLOSE #2 :: S
TOP
```

If you need to use either of those programs on files with a record length other than 80, just add VARIABLE (or FIXED) and the record length to all the file opening statements, and change that 80 in line 210 or 230.

This subprogram, in which X=28 for a 28-column screen or whatever width you want, will reformat a string of almost any length to print on screen without breaking words, and will return in L the number of lines required to print it, which can be used to space DISPLAY AT statements.

```
31993 SUB FORMAT(X,M$,L):: Y
=X
31994 IF LEN(M$)<Y+1 THEN 31
996 ELSE IF LEN(M$)<Y+X+1 AN
D SEG$(M$,Y,1)=" " THEN 3199
6 ELSE IF LEN(M$)<Y+X+1 AND
SEG$(M$,Y+1,1)=" " THEN 3199
6 ELSE P=Y-1
31995 IF P<1 THEN 31996 ELSE
 IF SEG$(M$,P,1)=" " THEN M$
=SEG$(M$,1,P)&RPT$(" ",Y-P)&
SEG$(M$,P+1,255):: Y=Y+X ::
GOTO 31994 ELSE P=P-1 :: GOT
O 31995
31996 L=INT(LEN(M$)/X)-(LEN(
M$)/X<>INT(LEN(M$)/X)):: SUB
END
```

The following little program, plus the magic of Funlweb, should be all the mailing list program that most people would need for home use. Just use Funlweb to create a file with name on the first line, address on the second line, city and state on the third - or use 4 or even 5 lines for the address if you need to, but the 6th line must either be blank or contain selection codes. These codes can be anything you want, such as C for everyone you want to send a Christmas card to, or B11 to send a birthday card in November, or whatever.

You can put as many codes as you want to on that line, separated or strung together but be sure not to use a code that is part of another code - for instance, if you use B11 for those November birthdays, don't use B or 1 or B1 or 11 for something else.

Then continue with the next address in another block of six lines. Just be sure that the line number of the line just above the first address line is always a multiple of six.

```
100 DISPLAY AT(12,1)ERASE AL
L:"Filename? DSK" :: ACCEPT
AT(12,14)BEEP:F$ :: OPEN #1:
"DSK"&F$,INPUT :: OPEN #2:"P
IO"
110 DISPLAY AT(14,1):"Print
addresses with code -":"":"(
to print all addresses,  j
ust press Enter)"
120 ACCEPT AT(15,1)BEEP:X$
130 LINPUT #1:A$ :: LINPUT #
1:B$ :: LINPUT #1:C$ :: LINP
UT #1:D$ :: LINPUT #1:E$ ::
LINPUT #1:F$
140 IF POS(F$,X$,1)<>0 OR X$
="" THEN PRINT #2:A$:B$:C$:D
$:"":""
150 IF EOF(1)<>1 THEN 130 EL
SE CLOSE #1
```

In Tips #62 I reported on the weird behavior of the CALL LOAD(-31961,149), when used to clear all defaults and search for a LOAD file on DSK1. I have since found that if you put this CALL at the beginning of a program, it will not execute until an END or STOP is reached - but if you break the program with FCTN 4, it will not be in memory!

I stated that after this CALL LOAD was executed, any number taken to the power of 0 (which should be a value of 1) acquired a value of 220.5727273. I was led astray by the INT in the the formula in which I first found this puzzle. Actually it is 220.57000101, which prints to the screen in the peculiar format F0.57000101.

If a number between 1 and 9 is added to that, it is printed as 1< followed by the number being added, followed by the decimal part. For a number between 10 and 19, the < is changed to = and between 20 and 29 it becomes > (note the ASCII sequence); from 30 to 35 it becomes ? but from 36 to 99 99 the decimal portion is preceded by 0 to 63 respectively. 100 is 2<0.570001 and the pattern continues.

Although these are not valid representations of numbers, they are treated as such. Run a program to give N the power of 2^0, then break the program and experiment in immediate mode.

PRINT N gives that strange F0.57000101. PRINT N+1, or whatever, gives values represented in the format described above. PRINT N*1 will give the true numeric value 220.57000101 but multiplying by some other values gave me results in the odd format, as did dividing.

Peter Walker pointed out to me that trying to subtract from N within a program resulted in printing a value followed by a crash reporting a SYNTAX ERROR (in the line which had just been executed!) followed by a jump to a non-existent line zero!

N-1 should be 219.57.. of course, but in immediate mode PRINT N-1 results in 63.57000101. In the format in which added values are printed, this would be 319.57000101 but the 63.. is actually a decimal value, as can be proved by PRINT CHR$(INT(N-1))! When I tried to get a zero value by PRINT N-64.57000101, the computer blew its mind.

Does anyone know what is going on here?

An IBM program called DOC-SMASH, which sells for about $35, will read a D/V80 file and output it to a printer in full carriage-width lines of elite condensed subscript thereby getting up to 216 lines per page. Bud Wright wrote a TI version, with assembly links, to let us do the same thing for free. His version wouldn't work on my trusty old Gemini 10X, which does not support condensed elite, so I wrote this mini-program which is not as fast as Bud's, but does the job.

```
100 DISPLAY AT(3,9)ERASE ALL
:"TEXTSMASHER":"":"For the G
emini 10X printer, to print
D/V80 text in linesof 136 ch
aracters closely  spaced, i
n subscript."
110 DISPLAY AT(20,1):"Press
Enter to end input" :: DIM F
$(20)
120 F=F+1 :: DISPLAY AT(12,1
):"FILE #";F:"DSK" :: ACCEPT
 AT(13,4)BEEP:F$(F):: IF F$(
F)<>"" THEN 120
130 OPEN #2:"PIO",VARIABLE 2
55 :: PRINT #2:CHR$(27)&CHR$
(83)&CHR$(1);
140 PRINT #2:CHR$(15)&CHR$(2
7)&CHR$(51)&CHR$(12);:: LN=1
36
150 FOR J=1 TO F-1 :: OPEN #
1:"DSK"&F$(J),INPUT
160 LINPUT #1:M$
170 IF LEN(T$)>0 THEN M$=T$&
" "&M$ :: T$=""
180 IF LEN(M$)<LN+1 AND POS(
M$,CHR$(13),1)<>0 THEN PRINT
 #2:M$ :: GOSUB 260 :: M$=""
 :: GOTO 230
190 IF LEN(M$)=LN THEN PRINT
 #2:M$ :: GOSUB 260 :: M$=""
 :: GOTO 230
200 IF LEN(M$)<LN AND EOF(1)
<>1 THEN LINPUT #1:X$ :: M$=
M$&" "&X$ :: GOTO 170 ELSE I
F LEN(M$)<136 THEN PRINT #2:
M$ :: GOSUB 260 :: GOTO 240
210 P=LN
220 IF SEG$(M$,P,1)=" " THEN
 T$=SEG$(M$,P+1,255):: M$=SE
G$(M$,1,P):: PRINT #2:M$ ::
GOSUB 260 :: M$="" :: GOTO 2
30 ELSE P=P-1 :: GOTO 220
230 IF LEN(T$)<LN+1 AND POS(
T$,CHR$(13),1)<>0 THEN PRINT
 #2:T$ :: GOSUB 260 :: T$=""
240 IF EOF(1)<>1 THEN 160
250 CLOSE #1 :: NEXT J :: ST
OP
260 X=X+1 :: IF X<121 THEN R
ETURN
270 X=0 :: FOR K=1 TO 8 :: P
RINT #2 :: NEXT K :: RETURN
```

For that to work properly, your paragraphs must end in carriage returns, and so must the title line, etc. If

such is not the case, try Bill Wood's method - load the file into Funlweb, enter RS for Replace String, then /. /.X/ but instead of X type CTRL U SHIFT M. At the first prompt, enter A for All. If your text has any paragraphs ending in ? or !, get your cursor back to the beginning, change that first period to ? or !, and do it again. You might also need to manually add carriage returns to titles, etc. Just type CTRL U, then SHIFT M wherever a CR is needed.

Without having printers to test it on, I think the program can be modified for the SG-10 by changing line 140 to

```
140 PRINT #2:CHR$(27)&"B"&CHR$(4)&CHR$(27)&CHR$(51)&CHR$(12);:: LN=160
```

And for old Epson-type printers which don't support elite condensed by

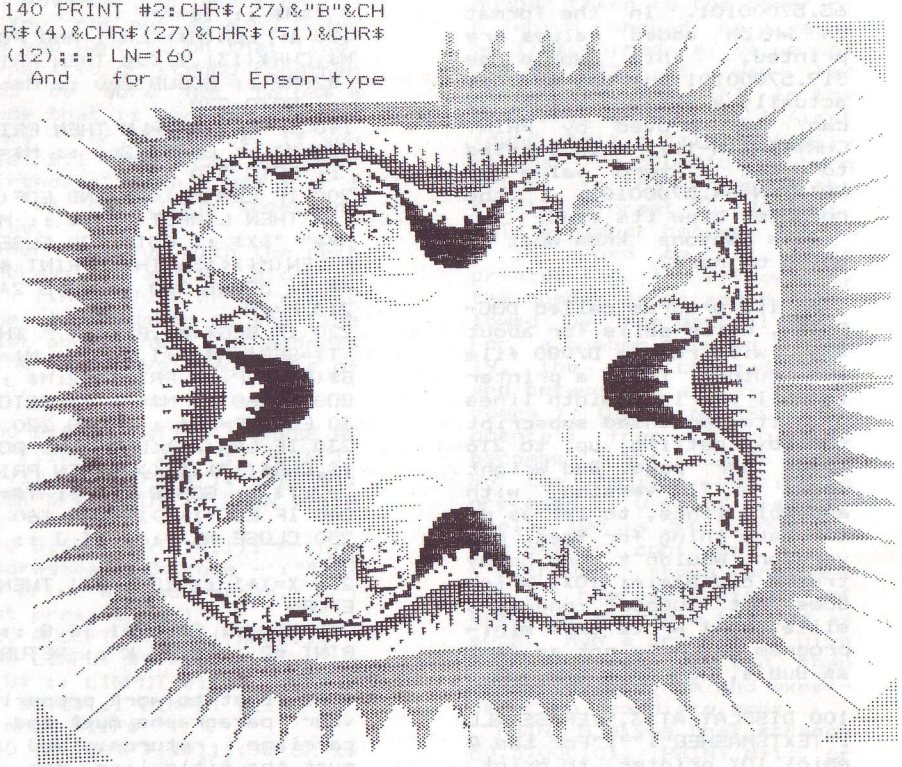
```
140 PRINT #2:CHR$(27)&CHR$(77)&CHR$(27)&CHR$(51)&CHR$(18);:: LN=132
```

And new Epson compatibles by

```
140 PRINT #2:CHR$(27)&CHR$(77)&CHR$(15)&CHR$(27)&CHR$(51)&CHR$(18);:: LN=160
```

You might also have to change that 8 to 12 in line 270 - my old Gemini seems to think that 11*12=128.

COMPLETELY out of memory,
Jim Peterson



HINTS, TIPS & ANSWERS (HTA)
ASSEMBLY LANGUAGE

Have you ever loaded up a DF80 file and didn't know the name to make it run. Well here is a short BASIC method to find you that name:

i) Load up TI BASIC from E/A
ii) CALL INIT :: CALL LOAD("DSKn.filename")
iii) CALL PEEK(16176,A,B,C,D,E,F)
iv) PRINT CHR$(A)&CHR$(B)&CHR$(C)&CHR$(D)&CHR$(E)&CHR$(F)

The name will be displayed and if you want to Run the program right away then use: CALL LINK("name")

HINTS, TIPS & ANSWERS (HTA)
MISCELLANEOUS

1. When using DM1000 if you wish to send printer codes to the printer (eg - to get condensed printing) then when the program displays the main menu with its three options press FCTN 3. You will be first prompted for a Device name (with "PIO" given as a default; then it asks if you wish to send control codes.

At this point if you enter "Y" you will be able to enter the printer codes you wish. The screen gives an example of how the code is to be input. Please note that the last character must be a star "*". If you disk is not write-protected these settings can be saved as your defaults.

2. If your diskette has fractured files (meaning parts here, parts there) this can cause added loading times, worn drives and possible data loss. You can tell this is happening if you hear the drive head "seesawing" back and forth when you load the program or file. Many of the disk managers will indicate fractured files by marking them with a "*".

To get rid of the fractures copy the disk onto a blank diskette using the "file copy" (option 1 of DM1000). Remember - you must copy each file individually, do not copy the whole disk using the "copy disk".

3. Have you ever wanted to put comments on a catalog listing, especially when you are passing the disk to a friend. One way to do it is to write the comments in with pencil but a neater way is to use TI-Writer. Here's how:

   i) When you go into DM1000 to catalog the disk first change the output device (see #3 above). Press FCTN 3 when the main menu screen appears and then change "PIO" default to "DSKn.filename". You don't want printer codes and you don't want to save this change permanently so press "N" for both these questions.

   ii) Go through the normal steps to catalog you disk. (ie - Main Menu: Option 2; Disk Menu: Option 1; once the catalog is displayed then FCTN 7 to get a printing.) However insteadof printing to the printer it will print to the disk giving you a DV80 file under the name you gave it in (i).

   iii) You can now access this file with TI-Writer to add your comments and for printing out. It is suggested that you turnoff Word Wrap (CTRL 0 - which will display a hollow cursor); if comments take more than one line than you will have to insert lines (FCTN 8).

--------> continued--->

By using this method you can make the listing more informative to the person getting it. For example programs that are related can be grouped regardless as to their names.

4. Another way to cut down on disk wear: if your disk has more than a dozen files it will run smoother if you process it with M-COPY. This is a life saver for disks with more than 20 files and absolutely essential if you have 30 files!!!

HINTS, TIPS & ANSWERS (HTA)
MULTI-PLAN

1. To speed up Multiplan there are two ways:
   i) turn the "recalculation" feature off. Go into "Options" and set "Recalc" to "NO". This should be done not only when inputting data but also when assembling the sheet (setting up the formulae, etc.) To calculate the sheet then press "FCTN 8"
   Warning: If the Recalc is turned off then the computer will automatically do a recalculation when you save the sheet; however if asked to print the sheet the computer WILL NOT recalculate first thus the printed sheet will be wrong. A good habit to get into is to always do a recalculation (FCTN 8) before printing.

   ii) keep the spreadsheet as small as possible. Although some sheets must be large most can be divided up into smaller sheets. For example if you have one large sheet showing "Revenue" and "Expenses" then you should consider making two small sheets; one for Revenues and one for Expenses. The extra time spent setting up two sheets will be more than offset by the savings in recalculation time.

2. A spreadsheet can be Printed to disk which can be loaded into the TI-Writer for merging with another document, etc.
   WARNING - be careful to use a different file name than the one used to save the Multiplan file as Multiplan will not warn you if you are about to overwrite an existing file when printing to disk.
   If you set the margins greater than 80 Multiplan will write the DV80 file wherein each record is longer than 80 characters. If you attempt to read this file with a BASIC program your system will produce a strange error code and lock up.
   TI-Writer will read the file but will only input the first 80 characters.

3. If you have Names that you wish to eliminate (un-name) this can be done by going into Name and deleting the reference in the "to refer to" field.

4. If you have "linked" the spreadsheet to another to eliminate this link all you must do is to erase the "Name" field. With a link (or reference) to nothing the link is broken and deleted.

---> more ---->

5. To change the target of an Xternal copy just specify the new cell (or cells) in the "to:" field. Since each cell (or range) on the supporting sheet can have only one target (on the dependent sheet), the old link will be replaced by the new.

6. If your RECALC is OFF, you can recalculate a single cell by setting the pointer to the cell, press "E" for Edit, then press ENTER. Only that cell will be recalculated.
   Note: If that cell relies on another which also had to be calculated then the information used from that other cell may be uncalculated and therefore the cell you just recalculated will be wrong.

7. To increase the response time of MP you should file copy the following files in the following order (one at a time):
   OVERLAY, MPHLP, MPCHAR, MPDATA, MPINTR and then MPBASE

8. When you become familiar with MP you can eliminate the MPHLP (Help) file from your work disk and there by free 158 sectors.

9. If you choose to lock the formulas in a worksheet make an unlocked backup "BEFORE YOU LOCK THEM". Locking the formulas is easy; however unlocking requires you to do it piece by piece.

10. If someone else will be doing data entry on a complex worksheet, it is a good idea to have them working with a locked copy. This avoids problems such as having someone enter data into a cell containing formulas or information you use elsewhere.

11. Use relative references wherever possible. This allows for copying of formulas without editing. Editing of formulas is both time consuming and prone to error.

12. References in formulas should be done by "pointing". This method is simple, creates relative references and is subject to less errors.

13. Once you start scrolling, you can release the FCTN or CTRL key and just keep the ARROW key depressed.

14. On a data disk with more than 18 files you can catalog (display filenames) the additional files (remember the SD command only shows the first 18 files) by placing the cell pointer on the last filename then pressing REDO (FCTN 8). After the screen has been redrawn and displays "TRANSFER LOAD filename" (where filename is the one you have just highlighted) the message line will now display "Enter a filename (arrow for directory)". Now press FCTN down-arrow and the last filename from the previous screen plus the additional files will be displayed.

15. If there is a rectangular area that will be used frequently on your worksheet, consider giving it a Name. You may then refer it by this name and thus will speed up the moving around the worksheet.

16. Do you have one large file and wish to make it into two or more smaller parts? It's easy, first you give a name to the section you wish to move to the second sheet and then save the large file to disk. Then load in the blank template and make use of the "eXternal" copy command. Make sure you tell it "no" to linking.

17. For a REAL speed up, use Art Greens re-write of Multiplan.

MULTIPLAN requires a Multiplan module AND a disk.

DISK LIBRARY REPORT.

Members with disk drives are reminded that we have a disk library with over 1000 disks of very assorted material available at low cost.
The majority of the disks are over 300 sectors, as we cram on as much as we can- several are a very full 358 sectors! (or 360 depending on how you measure usage!).
Almost all disks meet the minimum standard of single sided single density 5.25", and will run happily on all 5.25" drives the TI can commonly drive. Ownership of Extended Basic and the Funlweb disk package, which is available from the disk library, will enable you to run the vast majority of the disks. Joysticks and speech are required for some.
To obtain a copy of the 66 or so pages listing and describing the offerings, please send either four disks and return postage, or five pounds. The listings are only available on disk in DV80 format, and suitable XB programs are included to view the contents.
Disks are available for one pound per side copied, plus one pound per order for post and handling. If you prefer not to send blank disks, there is an extra charge of one pound per disk - we only use quality disks here!
No new disks this quarter, due to a decline in general TI activity, a decline in distribution channels, and a decline in library funds due to low usage.
Write Stephen Shaw for your disks today!

MACHINE CODE TUTORIAL

LET THERE BE SOUND - ASSEMBLER TUTORIAL
by Mack Mc Cormick
As you know from your BASIC programming sounds can be from 110 Hertz to 44,733 Hertz plus 8 noises may be generated. Durations may be from 1 to 4250 milliseconds. (.001 to 4.25 seconds). The volume can be from 0 (loudest) to 30 (quietest). Up to three tones and one noise may be generated simultaneously by the TMS9919 sound generator controller chip.
Three steps must occur to produce a sound using assembly language:

1. Load the Sound Table which begins at VDP address >83CC with the sound data.

2. Set the least significant bit of the byte at CPU address >83FD to indicate to the computer that the sound table is in VDP RAM.

3. Enable interrupts by using the LIMI 2 instruction.

Once each of these conditions are met you can start the sound generator by placing a value of >01 at CPU address >83CE. This address is used by the interrupt routine as a count down timer during sound generation.

The Sound Table
You must produce a sound table which describes the characteristics of the sound you wish to produce. The sound generators are numbered 1, 2, and 3. To produce a sound you must enter the following information:

1. Specify the tone generator
2. Frequency
3. Volume
4. Duration

Noises:

1. White or Periodic
2. Shift Rate
3. Volume
4. Duration

All bytes are specification bytes except duration. It takes three specification bytes to hold the generator, volume, and frequency. The frequency must be entered as a code.

CONTINUED----->

Specification Bytes for Tones

Byte	Bit#	Contains
ONE	0	Always set to 1
	1-2	Specifies the sound generator
	3	Always 0
	4-7	Contains the 4 least significant frequency code bits
TWO	0-1	Always 00
	2-7	Contains the 6 most significant frequency code bits.
THREE	0	Always 1
	1-2	Indicates sound generator used.
	3	Always 1
	4-7	Volume level

Bits 1 and two of all bytes indicate the tone generator:
00 is generator #1. 01 = #2. 02 = #03. 11 = noise.

Frequency vs. Frequency Code
============================
The frequency code is defined as half the period of the specified frequency. Here's the formula:

```
      111860.8
      -------- = Freq Code
      Frequency
```

Example: To find "middle C" which has a frequency of 523.25.
111860.8/523.25=213.8 .
This rounds to 214 or >0D6.
Bits 0-5 are placed in bits 2-7 of the second specification byte. The four least significant bits of the freq code are placed in bits 4-7 of the first specification byte.
Example to enter a tone of 392 Hz in generator 1 this equates to a frequency code of 285 or >11D.

```
   1000 XXXX   00XX XXXX  = >8---
```

Here we have selected generator 1.
Now we take our freq code >11D and place its 4 least significant bits (>D) in bit position 4-7 of the first specification byte:

```
   1000 1101   00XX XXXX  = >8D--
```

Finally we take the most significant 6 bits of the frequency code (>11) and place them in bit positions 2-7 of the second specification byte:

```
1000 1101 0001 0001 = >8D11
```

We have created the first two specification bytes to generate a tone on generator 1.
-----> CONTINUED---->

42

Volume Specification Byte
=========================
Volume is held in bits 4-7 of the third specification byte. It can range from 0 to 30. The 0-3 bits contain the generator number. You must pad the volume on the right with 0 always. For example: A volume of 0 on generator 2 = 1011 0000.

Duration
========
Not a specification byte. How long the tone or noise will last. Measured in 1/50 of a second. Can be from >00 to >FF.

Loading the Sound Table
=======================
You must indicate the number of @specification bytes you are going to feed the sound generator.
For example:
>03,>89,>3F,>91,30
>03 indicates three specification bytes.
Second and third bytes mean generator 1 with a tone of 110 Hz.
The fourth byte sets the volume at 2 on generator 1.
The 30 indicates 30/50 of a second duration.

So go FORTH and make music!!

```
****************************************
* ACCOMPANIES THE SOUND TUTORIAL       *
*    PLAYS "HOME ON THE RANGE"         *
*                                      *
****************************************
       DEF  START
       REF  VMBW
* SET UP VARIABLES FOR PROGRAM
MYREG  BSS  >20
SOUNDT EQU  >1000        SOUND TABLE ADDRESS
ONE    BYTE >01
       EVEN              FORCE LOCATION COUNTER TO WORD BOUNDRY
* FIRST EXECUTABLE INSTRUCTION
START  LWPI MYREG
       LI   R0,SOUNDT
       LI   R1,SDATA     SOUND DATA
       LI   R2,274
       BLWP @VMBW
* GENERATE THE SOUND
LOOP1  LIMI 0            LOAD INTERRUPT MASK IMMEDIATE
       LI   R10,SOUNDT   LOAD R10 WITH SOUND TABLE ADDRESS
       MOV  R10,@>83CC   VDP SOUND TABLE
       SOCB @ONE,@>83FD  SET ONES CORRESPONDING BYTE (SOUND TABLE
                                                    IN VDP RAM)
       MOVB @ONE,@>83CE  START SOUND PROCESSING
       LIMI 2
LOOP2  MOVB @>83CE,@>83CE *WHEN CPU ADDRESS >83CE = 0
       JEQ  LOOP1         *SOUND PROCESSING IS
       JMP  LOOP2         *FINISHED & PROGRAM REPEATS
                 -----> CONTINUED----->
```

```
* SOUND DATA
SDATA  BYTE >03,>8D,>11,>91,40
       BYTE >04,>AD,>11,>9F,>B1,40
       BYTE >03,>A6,>0D,>B1,40
       BYTE >06,>BE,>0B,>AD,>11,>95,>B5,40
       BYTE >09,>8A,>0A,>A6,>0D,>CD,>11,>95,>B5,>D5,60
       BYTE >05,>86,>0D,>91,>BF,>DF,20
       BYTE >03,>82,>0E,>91,40
       BYTE >03,>8E,>0F,>91,40
       BYTE >03,>80,>0A,>91,40
       BYTE >04,>A0,>0A,>9F,>B1,40
       BYTE >09,>80,>0A,>A6,>0D,>CD,>10,>95,>B5,>D5,60
       BYTE >05,>80,>0A,>91,>BF,>DF,20
       BYTE >03,>80,>0A,>91,20
       BYTE >03,>8F,>08,>91,40
       BYTE >09,>8A,>0A,>A6,>0D,>CD,>11,>95,>B5,>D5,40
       BYTE >05,>86,>0D,>91,>BF,>DF,20
       BYTE >04,>A6,>0D,>9F,>B1,40
       BYTE >05,>C6,>0D,>9F,>BF,>D1,40
       BYTE >03,>C2,>0E,>D1,40
       BYTE >03,>C6,>0D,>D1,40
       BYTE >03,>CE,>0B,>D1,80
       BYTE >03,>CD,>11,>D1,40
       BYTE >04,>8D,>11,>91,>DF,40
       BYTE >03,>86,>0D,>91,40
       BYTE >06,>8E,>0B,>AD,>11,>93,>B3,40
       BYTE >09,>8A,>0A,>A6,>0D,>CD,>11,>95,>B5,>D5,60
       BYTE >05,>86,>0D,>91,>BF,>DF,20
       BYTE >03,>82,>0E,>91,40
       BYTE >03,>8E,>0F,>91,40
       BYTE >03,>80,>0A,>91,40
       BYTE >04,>A0,>0A,>9F,>B1,40
       BYTE >06,>80,>0A,>AD,>10,>93,>B3,60
       BYTE >04,>80,>0A,>91,>BF,20
       BYTE >04,>A0,>0A,>9F,>B1,40
       BYTE >09,>8A,>0A,>A6,>0D,>CD,>11,>95,>B5,>D5,50
       BYTE >05,>8E,>0B,>91,>BF,>DF,30
       BYTE >03,>86,>0D,>91,40
       BYTE >09,>82,>0E,>AD,>11,>CD,>17,>95,>B5,>D5,40
       BYTE >05,>86,>0D,>91,>BF,>DF,40
       BYTE >03,>8E,>0B,>91,40
       BYTE >03,>86,>0D,>91,100
       BYTE >01,>FF,0
       END
=========================================
```


DISK DRIVE XB AUTO-LOAD

Back in Issue 32, page 24, I reported to you a "bug" in the system such that if you permitted your system to auto-load a program from disk when selecting XB, you lost the use of randomize.

In Issue 32 I gave a short XB program to cure it.

No response from anyone! I tested the bug out on my console and found that the bug was there and the fix cured it! However I am now using a different (cosmetically older) console, and find the bug is NOT there.

Disk owners... can you do a little test and report the results please?

Type in this program:

```
100 RANDOMIZE
110 FOR T=1 TO 5
120 PRINT INT(RND*10);
130 NEXT T
140 RUN "DSK1.LOAD"
```

and save this little program onto a new disk in drive one as "LOAD". Now reboot XB from the title screen so that your LOAD program is auto-loaded and watch the result.

Do you keep getting a repeating pattern of five numbers or is each group of five numbers different? Please let me know, together with the serial numbers of your console (on the base) and your XB module. These typically are in the form ATA0583 - goods manufactured later in Italy often omit serial numbers, so advise "no serial/Italian" or whatever.

In issue 32 there is a simple XB cure, but Bruce -who is an assembly programmer- found a similar problem occurring with some ramdisk operating systems booting machine code, so he wrote the following two routines, both intended to be used with XB. These are available on disk from the disk library ready assembled on a utility disk.

MACHINE CODE PROGRAMMERS- CIF & CFI

I received an enquiry about using CFI in a machine code program to be used from XB, and was unable to find any source code using it. The second program below uses it and works!

```
* ASSEMBLY SUBROUTINE "SEED"
* FOR USE with EXTENDED BASIC PROGRAMS
* SEEDS RANDOM NUMBERS
* THIS SEEDS RANDOM NUMBER PROCESS AND REPORTS KEY PRESSED
* INTO VARIABLE IN XB BEHAVES LIKE A "CALL KEY" LOOP
* I.E. "SEED" WILL KEEP LOOPING ITSELF UNTIL A KEY IS STRUCK
* CALL LINK("SEED",K) SEEDS THE RANDOMIZE PROCESS
*   use as CALL LINK("SEED",K) :: RANDOMIZE
* AND REPORTS THE KEY STRUCK BY THE USER INTO THE XB VARIABLE K
* AFTER THIS LINK, RANDOMIZE WILL WORK REGARDLESS OF HOW PROGRAM STARTED
* CODE BY BRUCE HARRISON
* RELEASED TO PUBLIC DOMAIN
* 18 AUG 1991
NUMASG EQU  >2008        NUMERIC ASSIGNMENT VECTOR
XMLLNK EQU  >2018        XML LINKAGE VECTOR
KSCAN  EQU  >201C        KEYBOARD SCAN VECTOR
KEYADR EQU  >8374        KEY-UNIT LOCATION
```

-----> more ---->

```
KEYVAL EQU  >8375        KEY VALUE BYTE
FAC    EQU  >834A        FLOATING POINT ACCUMULATOR
CIF    EQU  >20          CONVERT INTEGER TO FLOATING POINT
CFI    EQU  >12B8        CONVERT FLOATING POINT TO INTEGER
NUMREF EQU  >200C        NUMERIC REFERENCE VECTOR
STATUS EQU  >837C        GPL STATUS BYTE
       DEF  SEED         DEFINE ENTRY POINT
SEED
       LWPI WS           LOAD OUR WORKSPACE
       MOV  @>8378,R10   TAKE THE VDP INTERRUPT TIMER INTO R10
       ANDI R10,>0001    MASK OFF ALL BUT THE LOWEST BIT
       CLR  @KEYADR      CLEAR KEY-UNIT
KEYIN  MOVB @>83D7,@>83C1 TAKE THE SCREEN TIMEOUT'S
                             LOW BYTE INTO SEED + 1
       CLR  @STATUS      CLEAR GPL STATUS
       BLWP @KSCAN       SCAN KEYBOARD
       LIMI 2            ALLOW INTERRUPTS
       LIMI 0            DISALLOW INTERRUPTS
       CB   @ANYKEY,@STATUS HAS A KEY BEEN STRUCK?
       JNE  KEYIN        IF NOT, GO BACK
       XOR  @>83C0,R10   NOW XOR SO LOW BIT OF R10 IS
                            LOW BIT TAKEN ABOVE
       MOV  R10,@>83C0   PUT R10 AT SEED
       MOVB @>8379,@>83C0 PUT BYTE FROM VDP INTERRUPT INTO
                             HIGH BYTE OF SEED
       CLR  R0           CLEAR R0 FOR NUMBER ASSIGN
       LI   R1,1         FIRST PARAMETER TO PASS
       MOV  @KEYADR,@FAC PLACE KEY'S ASCII VALUE AT FAC
       BLWP @XMLLNK      USE XML LINKAGE
       DATA CIF          TO CONVERT INTEGER TO FLOATING POINT NUMBER
       BLWP @NUMASG      ASSIGN NUMBER TO PARAMETER
       LWPI >83E0        LOAD GPL WORKSPACE
       CLR  @STATUS      CLEAR GPL STATUS BYTE
       B    @>006A       RETURN TO GPL INTERPRETER
WS     BSS  32           OUR OWN WORKSPACE
ANYKEY BYTE >20          SPACE CHARACTER ASCII
       END
```

The following program uses both CFI and CIF and works. The comments are very useful too. Note that these listings are for use with the XB module, and amendments are required for use with EdAs or MiniMem - notably you do not need some of the EQUates.

The source code below is "standalone" and does not require the above code, they are alternates!

```
* QUICK RANDOM
* MAKES RANDOM NUMBERS QUICKLY
* USE WITH EXTENDED BASIC
* TWO CALL LINKS ARE INCLUDED
*   CALL LINK("SEED",K) :: CALL LINK("RKWIK",1,10,B)
*   randomizes AND sets variable B from 1 to 10.
* CALL LINK("SEED",K) ACTS LIKE A CALL KEY LOOP, BUT
* SETS A RANDOM VALUE IN RANDOM NUMBER SEED
* AND REPORTS THE KEY VALUE INTO A VARIABLE (K)
* AFTER SEED HAS BEEN PERFORMED,
* CALL LINK ("RKWIK",LOW,HIGH,VAR) WILL WORK
```

----> more ---->


```
* GIVE LINK THREE PARAMETERS:
* FIRST THE LOWEST INTEGER IN DESIRED RANGE
* SECOND THE HIGHEST INTEGER IN DESIRED RANGE
* THIRD THE VARIABLE INTO WHICH NUMBER IS TO BE ASSIGNED
* LIMITS FOR LOW AND HIGH ARE (-32768 AND +32767)
* SO LONG AS "SEED" HAS BEEN USED, RANDOMIZE IS UNNECESSARY
* TO GET RANDOM NUMBERS FROM RKWIK
* IF RND IS USED, RANDOMIZE MUST BE DONE AFTER "SEED" AND
* BEFORE RND IS USED
* CODE BY BRUCE HARRISON
* RELEASED TO PUBLIC DOMAIN 30 AUGUST 1991
NUMASG EQU  >2008        NUMERIC ASSIGNMENT VECTOR
XMLLNK EQU  >2018        XML LINKAGE VECTOR
KEYADR EQU  >8374        KEY-UNIT ADDRESS
KEYVAL EQU  >8375        KEY VALUE ADDRESS
KSCAN  EQU  >201C        KEYBOARD SCANNING VECTOR
FAC    EQU  >834A        FLOATING POINT ACCUMULATOR
CIF    EQU  >20          CONVERT INTEGER TO FLOATING POINT
CFI    EQU  >12B8        CONVERT FLOATING POINT TO INTEGER
NUMREF EQU  >200C        NUMERIC VARIABLE REFERENCE
STATUS EQU  >837C        GPL STATUS BYTE
       DEF  SEED,RKWIK
SEED
       LWPI WS           LOAD OUR WORKSPACE
       MOV  @>8378,R10   TAKE THE VDP INTERRUPT TIMER INTO R10
       ANDI R10,>0001    MASK OFF ALL BUT THE LOWEST BIT
       CLR  @KEYADR      CLEAR KEY-UNIT
KEYIN  MOVB @>83D7,@>83C1 TAKE THE SCREEN TIMEOUT'S
                             LOW BYTE INTO SEED + 1
       CLR  @STATUS      CLEAR GPL STATUS
       BLWP @KSCAN       SCAN KEYBOARD
       LIMI 2            ALLOW INTERRUPTS
       LIMI 0            DISALLOW INTERRUPTS
       CB   @ANYKEY,@STATUS HAS A KEY BEEN STRUCK?
       JNE  KEYIN        IF NOT, GO BACK
       XOR  @>83C0,R10   NOW XOR SO LOW BIT OF R10 IS
                            LOW BIT TAKEN ABOVE
       MOV  R10,@>83C0   PUT R10 AT SEED
       MOVB @>8379,@>83C0 PUT BYTE FROM VDP INTERRUPT
                             INTO HIGH BYTE OF SEED
       CLR  R0           CLEAR R0 FOR NUMBER ASSIGN
       LI   R1,1         FIRST PARAMETER TO PASS
       MOV  @KEYADR,@FAC PLACE KEY'S ASCII VALUE AT FAC
       BLWP @XMLLNK      USE XML LINKAGE
       DATA CIF          TO CONVERT INTEGER TO FLOATING POINT NUMBER
       BLWP @NUMASG      ASSIGN NUMBER TO PARAMETER
       LWPI >83E0        LOAD GPL WORKSPACE
       CLR  @STATUS      CLEAR GPL STATUS BYTE
       B    @>006A       RETURN TO GPL INTERPRETER
RKWIK
       LWPI WS           LOAD OUR OWN WORKSPACE
       CLR  R0           CLEAR R0, NOT ARRAY VARIABLE
       LI   R1,1         SET FOR FIRST PARAMETER
       BLWP @NUMREF      GET FIRST PARAMETER (LOW END OF
                            DESIRED RANGE)
       BLWP @XMLLNK      USE XML LINKAGE
      -----> continued ---->
```

```
       DATA CFI          TO CONVERT VARIABLE TO INTEGER
       MOV  @FAC,R12     R12 HAS LOW NUMBER
       INC  R1           POINT TO SECOND PARAMETER
       BLWP @NUMREF      GET SECOND PARAMETER (HIGH END
                              OF DESIRED RANGE)
       BLWP @XMLLNK      USE XML VECTOR
       DATA CFI          TO CONVERT TO INTEGER NUMBER
       MOV  @FAC,R13     R13 HAS HIGH NUMBER
       INC  R13          INCREMENT TO INCLUDE BOTH ENDS
       S    R12,R13      SUBTRACT LOW LIMIT FROM HIGH LIMIT
       LI   R4,28645     PUT A BIG NUMBER IN R4
       MPY  @>83C0,R4    MULTIPLY BY THE RANDOM NUMBER SEED
       AI   R5,31417     ADD A BIG NUMBER TO RESULT IN R5
       MOV  R5,@>83C0    PLACE THAT BACK AT SEED LOCATION
       CLR  R4           CLEAR R4 SO NUMBER IS RIGHT
                              JUSTIFIED IN R4-R5 PAIR
       DIV  R13,R4       DIVIDE BY THE RANGE +1
       A    R12,R5       ADD THE LOWER LIMIT TO REMAINDER
                              FROM INTEGER DIVISION
       MOV  R5,@FAC      MOVE THAT NUMBER TO FAC
       BLWP @XMLLNK      USE XML LINKAGE
       DATA CIF          TO CONVERT TO FLOATING POINT FORMAT
       INC  R1           POINT AT THIRD PARAMETER (VARIABLE
                              FOR RANDOM NUMBER)
       BLWP @NUMASG      ASSIGN THE VALUE TO THE VARIABLE
       LWPI >83E0        LOAD UP GPL WORKSPACE
       CLR  @STATUS      CLEAR STATUS BYTE
       B    @>006A       RETURN TO GPL INTERPRETER
WS     BSS  32           OUR OWN WORKSPACE
ANYKEY BYTE >20          THE SPACE CHARACTER VALUE
       END
```

==========

RIEMANN SPHERES....

One of the difficulties with taking an image and plotting it to a sphere is that the image can spread and distort and gaps left between pixels which then need filling.

Another way of tackling this is to take the sphere, and from each point on its surface, calculate if the point is visible from the chosen viewing angle, and then calculate back to the place (flat) surface for an equivalent x,y point and see if that pixel is on or off, and plot the sphere accordingly.

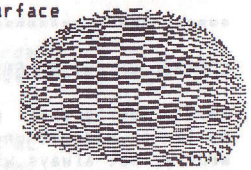
This is a program that does that- although for The Missing Link (a commercial program from Inscebot/Texaments) any pixel addressable graphics program can be used.

Some interesting results can be obtained with this program! Rather than use a specific graphic I have here merely used a "chequer board" pattern test which in theory plots a pattern of black and white squares onto the plane. The results have been very varied. Instead you could perhaps utilise a look up table of some sort - or even use Myarc XB to look up a graphic on the screen.

-----> continued--->

The sphere image is based on 0,0 at the North pole, extending to infinity,infinity at the South pole. An image from -2,-2 to 2,2 will cover the Northern hemisphere. Note that in the program below multipliers are used to control the size of the squares. The offset prevents having to deal with the way INT affects negative numbers, but does not affect the pattern.

While the ability to see through the sphere to the "back" surface has been included, it will usually result in a very untidy plot.



```
100 ! RIEMANN INVERSE V3
110 ! R CASTLE-SMITH
120 ! 2 MAY 91
130 ! Original in Hewlett Packard Rocky Mountain Basic
      for plotter output
140 ! for ti99/4a + extended basic + the missing link
      by stephen shaw September 1991
150 RANDOMIZE
160 CALL LINK("CLEAR")
170 ! X,Y,Z LOCATION OF VIEWER
180 A1=(RND*2-RND*2)*RND :: A2=(RND*2-RND*2)*RND ::
   A3=RND*RND*2-RND*2-.2
190 A1=INT(A1*1000)/1000 :: A2=INT(A2*1000)/1000 ::
   A3=INT(A3*1000)/1000
200 CALL LINK("PRINT",1,1,STR$(A1)&" "&STR$(A2)&" "&STR$(A3))
210 IF A1=0 AND A2=0 AND A3=0 THEN 180
220 MULT=(INT(RND*3)+1)*2+1 ! size of squares. must be odd.
    7=small sq 3=big sq
230 CALL LINK("PRINT",20,200,STR$(MULT))
240 ! PLOT REVERSE? 1=YES
250 PLOBAC=-(RND>.75)
260 CALL LINK("PRINT",180,1,STR$(PLOBAC))
270 !
280 IF A1=0 AND A2=0 THEN B1,C2=1 :: B2,B3,C1,C3=0 :: GOTO 320
290 M=SQR(A1*A1+A2*A2):: B1=-A2/M :: B2=A1/M :: B3=0
300 N=SQR(M*M+A3*A3):: C1=-A1*A3/M*N :: C2=-A2*A3/M*N :: C3=M/N
310 A1=A1/N :: A2=A2/N :: A3=A3/N
320 !
330 FOR YPIX=-60 TO 60
340 YV=YPIX/60
350 FOR XPIX=-60 TO 60
360 XV=XPIX/60
370 IF XV*XV+YV*YV<1 THEN GOSUB 400
380 NEXT XPIX :: NEXT YPIX
390 GOTO 390
400 T(1)=SQR(1-XV*XV-YV*YV):: T(2)=-T(1)
410 FOR I=1 TO PLOBAC+1
420 XR=A1*T(I)+B1*XV+CV*YV
430 YR=A2*T(I)+B2*XV+C2*YV
440 ZR=A3*T(I)+B3*XV+C3*YV
450 XC=2*(XR/(1-ZR))
460 YC=2*(YR/(1-ZR))
470 GOSUB 500
480 NEXT I
490 RETURN
500 ! IS THE PIXEL AT LOCATION XC,YC ON OR OFF?
510 ! plot of a chess board
```

----> continued --->

```
515 IF XC+YC>6 THEN 540 ! reduces chaos pictured near south
    pole due to squares too close to show pattern.
520 PLOT=INT(MULT*XC+99)+INT(MULT*YC+99)! +99 offset makes
    all values +ve. thats all. MULT must be odd (3,5,7...)
530 IF PLOT/2=INT(PLOT/2)THEN CALL LINK("PIXEL",YPIX+90,XPIX+90)
540 RETURN
```

==========

CONTROL OF CS1 BY PROGRAM:

The following small program will turn your cassette recorder or other device on and off under software control. The program has appeared widely, not always with the authors name attached (rectified here!) - below you will also find some hints on the practicalities of its use!

```
10 ! program by Ed Hall
11 ! to control the
12 ! "CS1" remote control
13 ! put in a music tape
14 ! plug in the remote
15 ! and press keys P and S
16 !
17 ! reverse polarity of
18 ! remote if it won't operate your recorder!
19 !
20 !
100 CALL INIT
110 CALL LOAD(16368,79,70,70,32,32,32,36,252)
120 CALL LOAD(16376,79,78,32,32,32,32,36,244)
130 CALL LOAD(8194,37,4,63,240)
140 CALL LOAD(9460,2,12,0,45,29,0,4,91,2,12,0,45,30,0,4,91,203,78)
150 PRINT "PRESS:":" P Play":" S Stop"
160 CALL KEY(3,A,B)
170 IF B<1 THEN 160
180 ON POS("PS",CHR$(A),1)+1 GOTO 160,190,200
190 CALL LINK("ON"):: GOTO 160
200 CALL LINK("OFF"):: GOTO 160
210 END
```

This program requires Extended Basic and 32k ram. It can operate the cassette player to provide audio-visual tuition, with cassette parts triggered at the appropriate stage by the program.

Note 1. The remote control is an electronic switch (for the technically minded, most consoles use a Darlington driver controlled by an optically isolated device). The polarity of its connection to the remote device MATTERS! If it does not work, try reversing the wires to the 2.5mm plug.

Note 2. As an electronic device, it uses up some of the voltage available to the remote device, a minimum of 1 volt. Some devices may be unhappy to have a whole volt removed - use mains supply for your recorders if possible and note that ni-cad rechargeable batteries only start with a lower open circuit voltage than other cells! Not too much lower but if the one volt drop counts...

-----> continued---->

You can use this program together with a clock such as that to be found in Triton ExBas, or Enhanced Display Package (library supply) etc etc to turn the cassette on and off at specific times (who needs a time switch! You can turn this device on and off several times a minute all year if you wish!).

A simple for-next loop delay program can be used to time your on-off periods.

Of course you are not limited to turning just a cassette recorder on and off!!! You can turn anything on and off... provided you observe the correct polarity and do not try to switch too great a load!

In simple terms the ABSOLUTE maximum you can switch is 40V DC at 400mA, but you can exceed that by using a relay - ensuring that you use a diode to protect the computer circuitry (I gather a diode clamp is often used with relays...). For the technically minded, most consoles use a TIL119 isolator, and add to it a TIS92 NPN transistor, with the collecter connected (normally!) to the tip of the plug. Thanks to Ross Mudie of Australia for this detail.

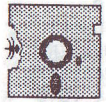
==========

The following program will run with EITHER mini memory OR with Extended Basic and 32k ram.

This is by Art Green of Ottawa, and demonstrates the interrupt routine capability. It can also be used in a LOAD program on disk to convince your best mate that his TI really has got a computer virus....

The first listing is in Basic for either module - it actually tests to see which module you are using!

```
100 REM interrupt demo
110 REM
120 REM MACHINE LANGUAGE
130 REM ROUTINE LOADED AT
140 REM >2600 XB OR E/A WITH 32K
150 REM >7200 MINI MEM NO 32K
160 REM
170 CALL INIT
180 XM=9728
190 MM=29184
200 LAD=XM
210 REM TEST XB OR MM?
220 CALL LOAD(XM,170)
230 CALL PEEK(XM,X)
240 IF X=170 THEN 270
250 REM NO 32K MUST BE MM
260 LAD=MM
270 A=LAD
280 REM LOAD M/C
290 CALL CLEAR
300 FOR D=540 TO 630 STEP 10
310 CHECK=0
320 FOR N=1 TO 10
330 READ X
340 CALL LOAD(A,X)
350 CHECK=CHECK+X
360 A=A+1
370 NEXT N
```



-----> continued--->

```
380 READ X
390 IF CHECK<>X THEN 490
400 NEXT D
410 REM POKE INTERRUPT
420 REM ROUTINE ADDRESS
430 REM INTO >83C4
440 CALL LOAD(-31804,LAD/256)
450 REM JUST IDLE AWAY TIME
460 FOR N=1 TO 9940
470 NEXT N
480 STOP
490 PRINT "ERROR IN DATA STATEMENT ";D
500 STOP
510 REM EACH DATA STATEMENT
520 REM HAS 10 DATA BYTES
530 REM PLUS A CHECK SUM
540 DATA 192,236,000,092,004,194,005,131,002,131,987
550 DATA 000,060,026,003,004,195,006,236,000,094,624
560 DATA 203,003,000,092,060,172,000,090,006,002,628
570 DATA 017,015,019,010,006,002,019,004,002,000,94
580 DATA 002,039,010,083,016,002,002,000,002,086,242
590 DATA 096,003,016,007,002,000,000,119,010,083,336
600 DATA 016,002,002,000,000,072,160,003,002,096,353
610 DATA 064,000,006,192,215,192,006,192,215,192,1274
620 DATA 016,000,216,044,000,094,140,000,004,091,605
630 DATA 000,015,000,000,138,128,000,000,000,000,281
640 END
```

For those of you interested in machine code, this is the SOURCE CODE for the machine code element contained in the above program...

```
* Interrupt demonstration
* GPL WS >83E0
* R11=RETURN ADDRESS
* R12=ADDRESS OF THIS ROUTINE
* R13 >9800 GROM READ DATA
* R14 SYSTEM FLAGS
* R15 >8C02 VDP WRITE ADDRESS
* PROGRAM IS SELF RELOCATABLE
*
USRINT  MOVV  @T(R12),R3   GET TICK COUNTER
        CLR   R22          SET R2,R3 FOR DIVIDE
        INC   R3           COUNT TICKS
        CI    R3,60
        JL    TOCK         JUMP NO
        CLR   R3           WRAP FROM 60 TO 0
        SWPB  @C(R12)      AND SWAP SCREEN CHARS
TOCK    MOV   R3,@T(R12)   SAVE NEW TICK COUNTER
        DIV   @F(R12),R2   DIVIDE INTO 4 QUADRANTS
        DEC   R2           R2 HAS QUADRANT 0 TO 3
        JLT   QZERO        JUMP IF QUAD 0
        JEQ   QONE         JUMP IF QUAD 1
        DEC   R2
        JEQ   QTWO
                           ----> continued--->

QTHREE  LI    R0,551       LEFT SIDE GOING UP
        SLA   R3,5         Q3=551-32*REM
        JMP   QS
QTWO    LI    R0,598       BOTTOM GOING LEFT
QS      S     R3,R0        Q2=598-REM
        JMP   OUT
QONE    LI    R0,119       RIGHT SIDE GOING DOWN
        SLA   R3,5         Q1=119+32*REM
        JMP   QA
QZERO   LI    R0,72        TOP GOING LEFT
QA      A     R3,R0        Q0=72+REM
OUT     ORI   R0,>4000     VDP WRITE BIT
        SWPB  R0
        MOVB  R0,*R15      SET VDP ADDRESS FOR WRITE
        SWPB  R0
        MOVB  R0,*R15
        NOP
        MOVB  @C(R12),@>8C00  * CHAR TO VDP
        B     *R11         RETURN TO ROM ROUTINE
F       DATA  15           # OF CHARS PER QUADRANT
T       DATA  0            TICK COUNTER
C       DATA  >8A80        "*" CHARS FOR DISPLAY
        END
```

And before we leave this...

You do not have to turn your console off or quit...
Enter in command mode: CALL LOAD(-31804,0,0).

==========

TEN YEARS AGO.....

JANUARY 1982

The Peripheral Expansion Box and the P-Code and 32K memory expansion cards to go in the PEB debut at the Winter Consumer Electronics Show in Las Vegas, Nevada on January 7th.
MINI MEMORY, TI INVADERS, CARWARS, TOMBSTONE CITY, BLASTO, THE ATTACK and MUNCHMAN also make their official debut.

Computer and Video Games Magazine publish Extended Basic program by S Shaw. .

FEBRUARY 1982

-UK Group members offered copies of TI Archive Document dealing with extra CALLS from PRK-Stats, CALL A,D,S,L,P,G, and H.

-UK Group publish TI Basic program which allows pseudo-hi resolution drawing.

MARCH 1982

-UK Games Publisher WADDINGTONS, grants first software licence in respect of its games to Stephen Shaw for TI99/4A program BLACK BOX. Inventer Dr Eric Solomon also consents.
-Stephen Shaw approaches several U.S. software firms for U.K. licences to their software. Agreements reached with Norton Software, FFF Software and PRP Computergraphics. .

APRIL 1982

-TIHOME membership 180
-First RAMBLES article by S Shaw
-TI99/4 seen on display in Science Museum, London. .

==========

RIEMANN SPHERE. Graphic program...

This program illustrates how to project a graphic onto a sphere. The projector has a wide angle lens and is at the South pole of the sphere (which of course is translucent). This location leads to some squeezing of the image as it nears the South pole, which could be regarded as being "at infinity".

The graphic must have coordinates within the range +2 to -2 in both directions if the image is to appear entirely within a polar view otherwise the image will go round the sphere! The far pole has a value of infinity so dont worry about going past it... - larger graphics can easily be scaled down!

The program presented here produces two types of graphic- one is a set of nested squares and the other is a spiral starting from the North pole, which has location 0,0.

A further variation shown here is to have the globe slightly transparent, such that you can see the plot on the "other side" as well as the side facing you- to make these less sonfusing, the sample plots are done with lines on the face and with spaced dots on the back.

If only the face is to be viewed then you can easily plot with just dots and forget about lines- and you do not even have to draw the circle either.

This program is for Extended Basic plus The Missing Link but can be easily transferred to any other language that allows use of pixel graphics.

The original program was written 9.3.91 by R Castle-Smith in Rocky Mountain Basic for output to a plotter. The conversion for the TI is by Stephen Shaw.

The original program was published in Fractal Report #16 (8/91) cost Two Pounds, or a 6-issue subscription for Ten Pounds, from Reeves Telecom. Labs. Ltd., West Towan House, Porthtowan, TRURO, Cornwall, TR4 8AX.

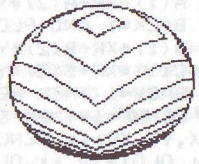
This program allows you to view the globe from any direction. The direction is given by three inputs, which are the angle to the centre of the sphere from each of the three planes (left-right, up-down, towards-away) passing through the centre. The values of these three inputs are relative rather than absolute- an input of 1,1,0 is the same view as 2,2,0.

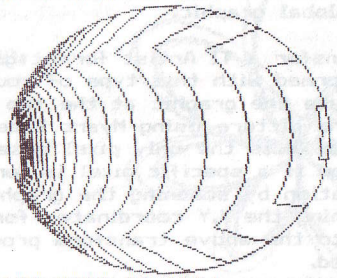
The three values are in the variables A(1), A(2), and A(3). If the first two are kept at zero, and A(3) is varied from +1 to -1 for successive pictures, you will see a view from one pole passing to the equator to the other pole.

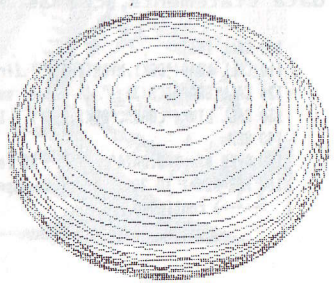
```
100 ! RIEMANN SPHERE R CASTLE-SMITH 3/91 for TI by S
SHAW 8/91
110 ! EX BAS + THE MISSING LINK
120 CALL LINK("CLEAR")
130 RANDOMIZE
140 IF RND<.5 THEN SP$="SP" ELSE SP$="SQ"
150 A(1)=3*RND*RND :: A(2)=3*RND*RND :: A(3)=4*RND-4*RND
:: IF A(1)+A(2)+ABS(A(3 ))<.01 THEN 150
160 IF A(1)>2 OR A(2)>2 THEN 150
170 FOR T=1 TO 3 :: CALL SHORT(A(T)):: NEXT T
     ----> continued
```




```
180 IF RND<.5 THEN PLOBAC=0 ELSE PLOBAC=1
190 REM
200 REM DERIVE TRANSFORM
210 REM
220 IF A(1)=0 AND A(2)=0 THEN B(1),C(2)=1 :: B(2),B(3),C
(1),C(3)=0 :: GOTO 290
230 M=SQR(A(1)*A(1)+A(2)*A(2))
240 B(1)=-A(2)/M :: B(2)=A(1)/M :: B(3)=0
250 N=SQR(M*M+A(3)*A(3))
260 C(1)=-A(1)*A(3)/(M*N)
270 C(2)=-A(2)*A(3)/(M*N)
280 C(3)=M/N
290 ! DRAW CIRCLE!!!
300 CALL LINK("PRINT",1,1,SP$&" "&"X="&STR$(A(1))&"Y="&
STR$(A(2))&" Z="&STR$(A( 3)))
310CALL LINK("PRINT",180,1,"PLOBAC="&STR$(PLOBAC)&"(0=
NO PLOT ON BACK)")
320 !
330 !
340 CALL LINK("CIRCLE",101,101,70)
350 !
360 REM GENERATE PATTERN
370 !
380 ! SPIRAL
390 IF SP$="SQ" THEN 480
400 FOR I=1 TO 5000 STEP 4 :: R=I/1000 :: T=I/10 :: XC=R
*COS(T):: YC=R*SIN(T)
410 GOSUB 630
420 NEXT I
430 !CALL LINK("DUMP")
440 CALL LINK("PRINT",9,1,"!*!")
450 CALL KEY(5,K,S):: IF S<>1 THEN 450
460 RUN
470 RUN
480 ! SQUARE
490 !
500 FOR SQUARE=.1 TO 4 STEP .2
510 ! FOUR SIDES, 1 TO 4
520 YC=-SQUARE :: FOR XC=-SQUARE TO SQUARE STEP SQUARE/
10 :: GOSUB 630 :: NEXT C
530 XC=SQUARE :: FOR YC=-SQUARE TO SQUARE STEP SQUARE/10
:: GO SUB 630 :: NEXT YC 540 YC=SQUARE :: FOR XC=SQUARE
TO -SQUARE STEP -SQUARE/10 :: GOSUB 630 :: NEXT C
550 XC=-SQUARE :: FOR YC=SQUARE TO -SQUARE STEP -SQUARE/
10 :: GOSUB 630 :: NEXT YC
560 PENUP=1
570 NEXT SQUARE
580 CALL LINK("PRINT",9,12,"*!*")
590 CALL KEY(5,Q,W):: IF W<>1 THEN 590
600 ! CALL LINK("DUMP")
610 CALL LINK("CLEAR")
620 RUN
 ----> continued ---->
```





```
630 MD=XC*XC+YC*YC
640 XR=2*XC/(1+MD)
650 YR=2*YC/(1+MD)
660 ZR=(MD-1)/(1+MD)
670 !
680 BACK=0
690 IF A(1)*XR+A(2)*YR+A(3)*ZR<0 THEN BACK=1
700 IF BACK=1 AND PLOBAC=0 THEN PENUP=1 :: GOTO 750
710 XV=B(1)*XR+B(2)*YR+B(3)*ZR
720 YV=C(1)*XR+C(2)*YR+C(3)*ZR
730 IF BACK=1 THEN PENUP=1
740 IF PENUP=1 THEN OLDX=XV :: OLDY=YV :: CALL LINE(OL
DY,OLDX,YV,XV):: PENUP=0 ELSE CALL LINE(OLDY,OLDX,(YV),
(XV)):: OLDX=XV :: OLDY=YV
750 RETURN
760 STOP
770 SUB LINE(OLDX,OLDY,X,Y)
780 IF OLDX=0 AND OLDY=0 THEN SUBEXIT
790 OLX=OLDX*70+101 :: X=X*70+101 :: OLY=OLDY*70+101 ::
Y=Y*70+101 :: CALL LINK( "LINE",OLX,OLY,X,Y):: SUBEND
800 SUB SHORT(X)
810 S$=STR$(X):: S$=SEG$(S$,1,5):: X=VAL(S$)
820 SUBEND
```




You will quickly see that by varying the viewing angles to produce several pictures and then putting them together with COMIC SHOW 4 (available from the disk library) you can have an animated tour around your global graphic.

To transfer a TI Artist format graphic to a form that can be transformed with this type of routine, the easiest route seems to be to locate the graphic at the top left of a TI Artist picture, then load the picture using Myarc Extended Basic.

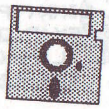
Myarc XB is the only pixel based graphic program I know that will tell you if a specific pixel is "on" or "off". If you know this information by scanning the graphic, you can set up a disk file containing the X,Y coordinates for each ON pixel, then read that as an input to the above transform program, with scaling and positioning as required.

This little program requires Myarc Extended Basic, which in turn requires the Myarc Ram Card. It makes use of a unique feature of Myarc XB, the ability to scan the screen and determine if a pixel is on or off. This is by means of a modified CALL GCHAR.

The program is used to create a DV80 data file to be read by another program, or converted into a mergeable set of DATA lines after adding line numbers and data statements, commas etc, which can of course be printed (after amendment).

```
100 CALL INIT
110 CALL LOAD("DSK1.MYARTIST") ! [from disk library]
120 CALL GRAPHICS(3)           !  set bit map mode
130 CALL LINK("LOAD","DSK1.PIC") ! load pic- no need to add _P
140 REM
150 OPEN #1: "RD.DATA"         ! RD is optional device name for myarc ramdisk
160 FOR ROW=1 TO 20            !  approx size of graphic- overscan
```

-----> continued




```
170 FOR COL=1 TO 160           !
180 CALL GCHAR(ROW,COL,VAR)  ! myarc xb variant form
190 IF VAR=1 THEN PRINT #1: ROW;COL
200 NEXT COL
210 NEXT ROW
220 CLOSE #1
230 END
```

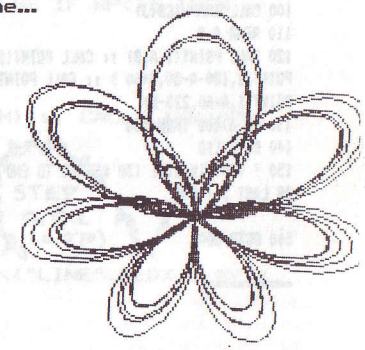
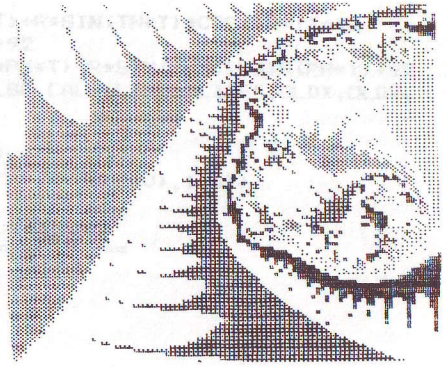
The object code utility MYARTIST is available from the disk library. The data is saved by the above program in a DV80 file which can be edited by TI Writer or read by your graphic drawing program. The data file looks like this in TI Writer- part only-

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

and so on and so on and so on...


Of course, it makes life easier if we prepare a MERGE file directly after scanning the image, and the following program will do this, collecting ten pixel coordinates per program DATA line...

```
100 CALL GRAPHICS(3)   ! MYARC XB ONLY
110 CALL INIT :: CALL LOAD("DSK2.MYARTIST")
120 CALL LINK("LOAD","DSK1.PIC")
130 OPEN #1: "RD.DATA", DISPLAY ,VARIABLE 163 ! will create loadable file
140 L=1 :: CT=0
150 REM
160 FOR ROW=1 TO 20     ! approx size of image
170 FOR COL=1 TO 160
180 CALL GCHAR(ROW,COL,VAR)
190 CALL WRITE(0,20,2,STR$(ROW)&" "&STR$(COL)&" "&STR$(VAR)&"   ")
200 IF VAR<>1 THEN 360
210 CT=CT+1
220 IF CT=1 THEN A$=CHR$(0)&CHR$(L)&CHR$(147)
230 IF CT=11 THEN A$=A$&CHR$(0) :: PRINT #1: A$ :: L=L+1 :: CT=1
:: A$="" :: GOTO 220
240 IF CT>1 THEN A$=A$&CHR$(179)
250 N$=STR$(ROW) :: LN=LEN(N$)
260 A$=A$&CHR$(200)&CHR$(LN)
270 FOR I=1 TO LN
280 A$=A$&CHR$(ASC(SEG$(N$,I,1)))
290 NEXT I
300 A$=A$&CHR$(179)
310 N$=STR$(COL) :: LN=LEN(N$)
320 A$=A$&CHR$(200)&CHR$(LN)
330 FOR I=1 TO LN
340 A$=A$&CHR$(ASC(SEG$(N$,I,1)))
350 NEXT I
360 NEXT COL :: NEXT ROW
370 A$=A$&CHR$(0) :: PRINT #1: A$
380 A$=CHR$(255)&CHR$(255)
390 PRINT #1: A$ :: CLOSE #1
```

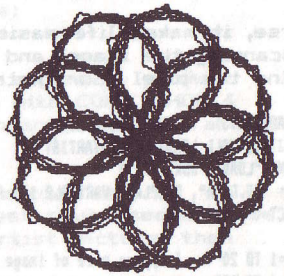



The merge data file will load directly using MERGE RD.DATA and it starts off like this...

```
1 DATA 1,132,1,133,1,134,2,132,2,133,2,134,3,132,3,133,3,134,4,132
2 DATA 4,133,4,134,5,132,5,133,5,134,6,1,6,2,6,3,6,4,6,5
3 DATA 6,6,6,7,6,8,6,9,6,15,6,16,6,17,6,18,6,19,6,20
4 DATA 6,21,6,47,6,48,6,49,6,50,6,51,6,52,6,53,6,59,6,60
5 DATA 6,61,6,62,6,63,6,64,6,65,6,69,6,70,6,71,6,72,6,73
6 DATA 6,74,6,75,6,79,6,80,6,81,6,82,6,83,6,87,6,88,6,89
7 DATA 6,90,6,91,6,92,6,93,6,97,6,98,6,99,6,100,6,101,6,107  ! and so on
```

The data is a record of all pixels which are ON in the format row,column. The program that reads this data can relocate the image by varying the values... this little program will give you some idea. The data can be used by any language and is available for use by the Riemann projection program.

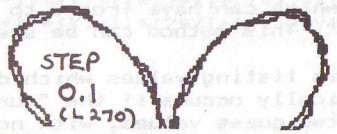
```
 1 REM USE MERGED DATA FILE TO PRINT GRAPHIC EVERY WHICH WAY ALL AT ONCE
 2 ! THIS LISTING FOR MYARC XB BUT ANY PIXEL ADDRESSABLE
   LANGUAGE CAN BE USED
 3 !
 4 ! DATA FILE TO BE MERGED INTO PROGRAM
 5 !
 100 CALL GRAPHICS(3)
 110 READ A,B
 120 CALL POINT(1,A,B) :: CALL POINT(1,180-A,240-B) :: CALL
 POINT(1,180-A-50,B+60 ) :: CALL POINT(1,180-B,A+10) :: CALL
 POINT(1,A+60,220-B)
 130 IF A>400 THEN 500
 140 GOTO 110
 150 ! TEST IN LINE 130 REFERS TO END OF DATA MARKER INSERTED
 AS LAST 2 ITEMS

 500 GOTO 500

 ==========
```



We have had a number of programs producing graphics based on drawing curves from two input variables. This program varies a little by the addition of a second part.
The second part can operate on its own, or with the first part. What the second part does is to retrace the original curve from the first part, but curve around that curve...

In the first part, a "step" of 0.1 is quite adequate on our screens, but in the second part (line 270) it produces rather an averaged curve- quite attractive in its own way. To produce a spikier second curve we must reduce the size of the step, alas at a cost of much increased processing time. You can see a difference by going down to 0.02, but 0.01 or less is better.

This program is from Dr Pickovers latest book and as presented requires the Missing Link, available from Inscebot or Texaments.




```
100 RANDOMIZE
110 CALL LINK("CLEAR")
120 TH=RND*2
130 PH=TH/(RND*8+1)
140 RP=INT(TH/PH)*6+1 :: IF RP>45 THEN RP=45 ELSE IF RP<15 THEN
    RP=15
150 CALL LINK("PRINT",180,1,STR$(RP))
160 R=70
170 R2=INT(RND*4+2)
180 CALL LINK("PRINT",1,1,STR$(TH)&"  "&STR$(PH)):: CALL LINK("PRI
    NT",180,211,STR$(R2))
190 IF RND<.2 THEN 260
200 FOR T=0 TO RP*PI STEP .10
210 X=R*SIN(TH*T)*COS(PH*T)+92
220 Z=R*COS(TH*T)+92
230 Y=R*SIN(TH*T)*SIN(PH*T)+92
240 IF T=0 THEN OLDX=X :: OLDY=Y ELSE CALL LINK("LINE",OLDX,OLDY,X,
    Y):: OLDX=X :: OLDY=Y
250 NEXT T
260 CALL LINK("PRINT",120,210,"PT2")
270 FOR T=0 TO RP*PI STEP .10
280 X=R2*SIN(140*TH*T)*COS(140*PH*T)+R*SIN(TH*T)*COS(PH*T)+92
290 Z=R2*COS(140*TH*T)+R*COS(TH*T)+92
300 Y=R2*SIN(140*TH*T)*SIN(140*140*PH*T)+R*SIN(TH*T)*SIN(PH*T)+92
310 IF T=0 THEN OLDX=X :: OLDY=Y ELSE CALL LINK("LINE",OLDX,OLDY,
    X,Y):: OLDX=X :: OLDY=Y
320 NEXT T
330 CALL SOUND(1000,660,4,664,4,667,8)
340 CALL LINK("SAVEP","RD."&SEG$(STR$(RND*1E10),1,5))
350 RUN
```

==========

SUCCESSIVE INVERSE INTERPOLATION

How does that heading grab you (it didn't huh...).
It refers to a method of solving an equation by making two guesses and then working your way towards the true answer. Many equations have more than one possible solution - for example we can say:

X*X-4=0

is true if X equals EITHER 2 OR -2 that is, two solutions.

If we make the x,y locations of each pixel on the screen the two guesses for this method, each pixel location will tend towards a final answer, and if there is more than one possible solution, we can colour that pixel according to the solution its values lead to. Some guesses may not lead towards a solution so we need to deal with that possibility too...

The program below deals with this, producing some interesting graphics - quite slowly I should warn! What interested me in particular in this listing was the way that colours are represented on a mono screen, by plotting not a single pixel but a block of four pixels, which can have from 0 to 4 pixels "on", giving rise to five "colours". This method can be used with other programs of course!

In this listing values which do not lead to a solution are white- this typically occurs if the "curve" of the equation is horizontal for the two guess values, with no angle to indicate a direction to head to!

This method of solution can lead to "false" solutions, due to wiggles in the equations graph, and these are plotted in two shades of grey. Values which lead to a "true" solution are in black.

The "true" solutions to the equation are held as the values of program variables r1,r2,r3,r4 and the formula is in lines 190 and 200. If you change the formula you will need to find the solutions and change line 100 accordingly - possibly other parts of the program as well.

The on screen plotting is the reverse of "normal" plots, and has positive (or larger) values at bottom left and negative (or smaller) values at top right. X is plotted vertically and Y is plotted horizontally.

```
1 ! METHOD OF DOUBLE POSITION
2 ! Paul Gailiunas in Fractal Report 17
3 ! for ti99/4a with ex bas and The Missing Link by Stephen
   Shaw Sept 1991
4 ! easily amended for any pixel addressable language
5 ! note particularly how the effect of five colours is given
    on a mono screen/printer
6 ! this effect can be used on any other graphic program needing
    up to 5 colours , including white.
7 !
8 ! formula is in lines 190 and 200 . Experiment with other
    polynomials or other formulae.
```

---> CONTINUED--->

60

```
9 ! move centre by setting XO and YO. Range is half the value
    of a side. Reduce range for magnification- well worth while!
10 ! takes a long time to plot!
11 !
80 RANDOMIZE
90 ON WARNING NEXT ! AVOID DIVIDE BY ZERO ERRORS
100 R1=-1 :: R2=-.5 :: R3=.5 :: R4=1
110 XO=2*RND-2*RND :: YO=2*RND-2*RND
120 CALL LINK("PRINT",1,1,STR$(XO)&":"&STR$(YO))
130 RANGE=RND*2 :: CALL LINK("PRINT",170,1,STR$(RANGE))
140 FOR X=30 TO 220 STEP 2 !  only plotting part of screen!
150 FOR Y=8 TO 168 STEP 2
160 XI=XO+(X-120)*RANGE/120 ! scaling
170 YI=YO+(95-Y)*RANGE/95
180 FOR I=1 TO 20 ! give up after 20 as unlikely to converge
190 FX=4*XI^4-5*XI*XI+1 ! EG FX=(XI+1)*(XI-1)*(2*XI+1)*(2*XI-1)
200 FY=4*YI^4-5*YI*YI+1 ! EG FY=(YI+1)*(YI-1)*(2*YI+1)*(2*YI-1)
! FIRST FORM IS FASTER
210 Z=(XI*FY-YI*FX)/(FY-FX)
220 XI=YI :: YI=Z
230 IF ABS(XI-YI)<0.00001 THEN 250
240 NEXT I
250 IF ABS(YI-R1)<.001 THEN GOSUB 320 ! which solution
    is it tending to?
260 IF ABS(YI-R2)<.001 THEN GOSUB 330 ! r1 or r2 or r3 or r4
270 IF ABS(YI-R3)<.001 THEN GOSUB 340
280 IF ABS(YI-R4)<.001 THEN GOSUB 350
290 NEXT Y
300 NEXT X
310 GOTO 310
320 CALL DOT(X,Y):: CALL DOT(X+1,Y):: CALL DOT(X,Y+1):: CALL
    DOT(X+1,Y+1):: RETURN
330 CALL DOT(X,Y):: CALL DOT(X+1,Y+1):: RETURN
340 CALL DOT(X,Y):: RETURN
350 CALL DOT(X,Y):: CALL DOT(X+1,Y):: CALL DOT(X,Y+1):: RETURN
360 END
370 SUB DOT(X,Y)
380 CALL LINK("PIXEL",Y,X)
390 SUBEND
400 ! FRACTAL REPORT is uk pounds ten for 6 issues in uk
401 REM from
402 REM reeves telecom labs ltd   west towan house
porthtowan TRURO cornwall TR4 8AX
```

==========

The mandelbrot set produces an image of what could be an odd bug, and as we zoom in we find more and more- continued magnification merely produces more and more detail, until we hit the numeric limits of our computer!

The program below produces images which include what could be some exotic jellyfish, and as we zoom in ever closer we find more and more of them. For magnification, concentrate on the area just inside the first black boundary.

The total image is within the range -1.4 to +1.4, and is symetrical about horizontal and vertical axes at point 0,0.

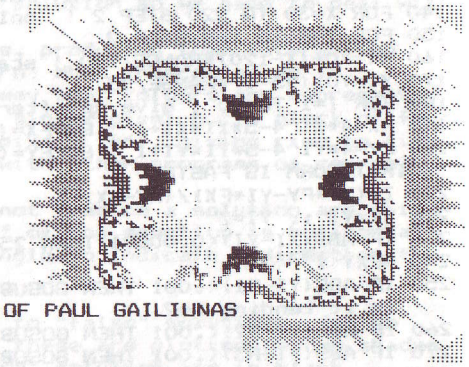
---> continued--->

61

The image can take some hours to complete, and as ever the use of machine code would be very nice - requiring maximum math accuracy, and some means of saving the images, to TI Artist format or to printer.

The program is for TI Extended Basic and the commercial utility The Missing Link (Inscebot/Texaments) but any language that allows pixel plotting will do.

A routine is included to show "colours" on a mono screen or printer dump, with five textures, here representing 16 colours.



```
1 ! BIOMORFOS
2 ! by JOSE E MURCIANO
3 ! APDO 192
4 ! 44080  TERUEL
5 ! SPAIN
6 !
7 ! FROM FRACTAL REPORT # 17
8 ! FOR TI99/4A BY
9 ! STEPHEN SHAW  SEPT 91
10 !
11 ! DRAWS GRAPHIC
12 ! YOU CAN ZOOM IN ON
13 !
14 ! EMPLOYS MONO COLOUR RENDITION OF PAUL GAILIUNAS
15 !
16 ! FOR EX BAS + THE MISSING LINK
17 ! BUT CAN USE ANY PIXEL ADDRESSABLE LANGUAGE
18 !
100 CALL LINK("CLEAR")
101 ! line 110 sets upper,lower,left and right margins
110 XI=-1.4 :: XA=1.4 :: YI=-1.4 :: YA=1.4
120 P=.7 :: Q=.01 :: LX=190 :: LY=190
130 DX=(XA-XI)/LX :: DY=(YA-YI)/LY
140 FOR NX=1 TO LX STEP 2
150 FOR NY=1 TO LY STEP 2
160 X=XI+NX*DX
170 Y=YI+NY*DY
180 FOR K=1 TO 16
190 XN=X^4+Y^4-6*X*X*Y*Y+P
200 Y=4*X^3*Y-4*X*Y^3+Q
210 X=XN
220 IF X*X+Y*Y>100 THEN IF ABS(X)<50 OR ABS(Y)<50 THEN CALL
DOT(NX,NY,K):: K=16 :: GOTO 230 ELSE K=16
230 NEXT K :: NEXT NY :: NEXT NX
239 CALL LINK("SAVEP","RD.PIC")
240 GOTO 240
250 SUB DOT(X,Y,K)
260 ON K GOTO 270,280,290,300,270,280,290,300,270,280,290,
    300,270,280,290,300,270,280
270 CALL LINK("PIXEL",X,Y):: SUBEXIT
280 CALL LINK("PIXEL",X,Y):: CALL LINK("PIXEL",X+1,Y+1):: SUBEXIT
290 CALL LINK("PIXEL",X,Y):: CALL LINK("PIXEL",X+1,Y)::
    CALL LINK("PIXEL",X+1,Y+1):: SUBEXIT
```

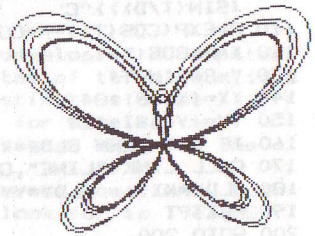
---> continued--->

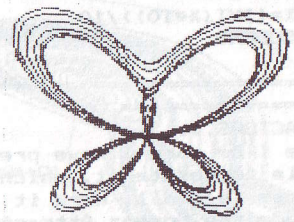

```
300 CALL LINK("PIXEL",X,Y):: CALL LINK("PIXEL",X+1,Y):: CALL
    LINK("PIXEL",X,Y+1) :: CALL LINK("PIXEL",X+1,Y+1)
310 SUBEND
320 ! fractal report costs ten uk pounds in uk for 6 issues from
330 ! reeves telecom labs ltd
340 ! west towan house  Porthtowan TRURO
350 ! Cornwall  TR4 8AX
360 !
370 END
```

NOT a long program, but the results are quite fascinating even if they do take a little while!

==========

In the last issue I mentioned that there was a formula for everything, it would appear even a butterfly...





```
1 ! BUTTERFLY CURVES 1
2 ! EXTENDED BASIC+THE MISSING LINK
3 ! S SHAW OCT 1991
4 ! from Pickover
5 ! computers and the imagination
6 ! isbn 0 86299 999 5
7 ! 1991
8 !
100 FOR T=0 TO 100*PI STEP 0.020
110 R=EXP(COS(T))-2*COS(4*T)+(SIN(T/12))^5
120 X=R*COS(T)   ! changes to polar
130 Y=R*SIN(T)   ! coordinates
140 XX=(X*18)+94    ! scales and
150 YY=(Y*18)+129   ! offsets
160 IF T=0 THEN OLDX=XX :: OLDY=YY
170 CALL LINK("LINE",OLDX,OLDY,XX,YY)
180 OLDX=XX :: OLDY=YY
190 NEXT T
200 GOTO 200
```

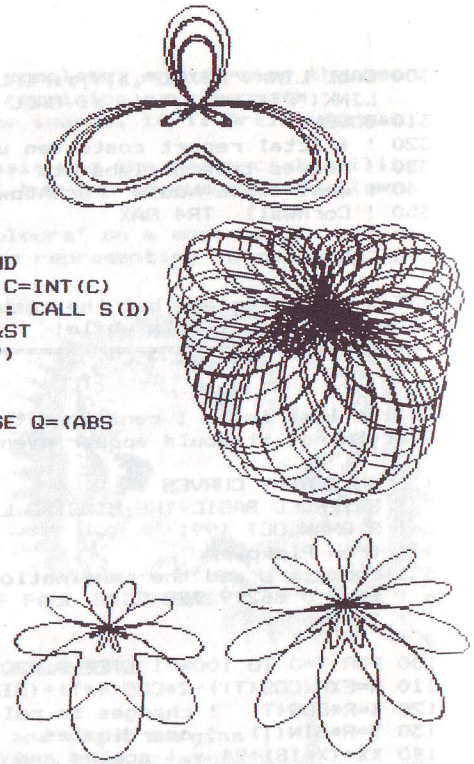
By using random variables as multipliers we can obtain a variety of results, some interesting, some slightly messy. The large number of steps are rarely needed and the picture could be printed or dumped when it looks finished or when you are happy with it...

Notice the sub program S which is used to shorten the random number to just one decimal place - it makes an easier screen display of the variables!

Note also that you cannot take a square root of a negative number- just try keying in: A=SQR(-1) or the same thing expressed a little differently A=(-1)^.5 which is why we need to add an ABS operator if the variable C is a fraction!

-----> more--->

```
1 ! BUTTERFLY CURVES 2
2 ! EXTENDED BASIC+THE MISSING LINK
3 ! S SHAW OCT 1991
4 ! from Pickover
5 ! computers and the imagination
6 ! isbn 0 86299 999 5
7 ! 1991
8 !
70 RANDOMIZE
80 A=1+RND*3 :: B=3+4*RND :: C=3+5*RND
   :: D=10+40*RND :: IF RND<.2 THEN C=INT(C)
85 CALL S(A):: CALL S(B):: CALL S(C):: CALL S(D)
90 CALL LINK("PRINT",1,1,STR$(A)&":"&ST
   R$(B)&":"&STR$(C)&":"&STR$(D)&" ")
95 IF C<>INT(C)THEN FLAG=1
100 FOR T=0 TO 100*PI STEP PI/150
105 IF FLAG=0 THEN Q=(SIN(T/D))^C ELSE Q=(ABS
   (SIN(T/D)))^C
110 R=EXP(COS(T))-A*COS(B*T)+Q
120 X=R*COS(T)
130 Y=R*SIN(T)
140 XX=(X*18)+84
150 YY=(Y*18)+129
160 IF T=0 THEN OLDX=XX :: OLDY=YY
170 CALL LINK("LINE",OLDX,OLDY,XX,YY)
180 OLDX=XX :: OLDY=YY
190 NEXT T
200 GOTO 200
210 SUB S(X)
220 X=(INT(X*10))/10
230 SUBEND
```



==========

ATTRACTORS...

While I am sure I have previously given the Ikeda attractor, I am unable to locate it, which may mean it was a while back, and as the program is small, here it is (again ?) together with a program to produce the Lorenz Attractor.

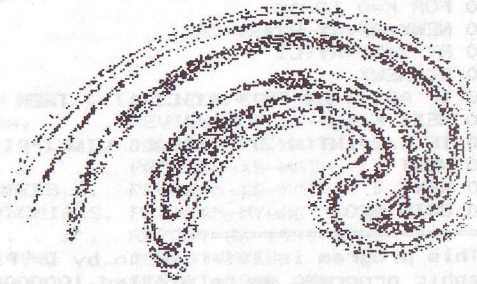
Both routines are taken from Computers and The Imagination by Clifford Pickover (1991: Alan Sutton).

Back in 1979, K Ikeda published a paper entitled "Multiple valued stationary state and its instability of the transmitted light by a ring cavity system" in the Journal Optical Communications (#30). Horrid title for a short program which produces a fascinating plot. The Ikeda Attractor is referred to as a Strange Attractor because it has -within limits - an unpredictable behavior. It is not possible for us to predict where the next pixel will appear.

------> more---->

64

```
100 XOFF=60 :: YOFF=170 :: SCALE=70 ! FILLS SCREEN 192*256
110 X,Y=0.1
120 C1=.4 :: C2=.9 :: C3=6 :: R=1
130 FOR L=0 TO 9940
140 !
150 T=C1-C3/(1+X*X+Y*Y)
160 ST=SIN(T)
170 CT=COS(T)
180 XT=R+C2*(X*CT-Y*ST)
190 Y=C2*(X*ST+Y*CT)
200 X=XT
210 J=X*SCALE+XOFF
220 K=Y*SCALE+YOFF
230 CALL LINK("PIXEL",J,K)
240 NEXT L
250 GOTO 250
```

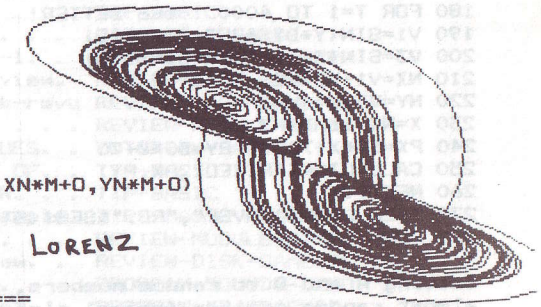


==========

Back in 1962 MIT meteorologist E N Lorenz was developing a computer model of the weather and came to a weather system of three equations which on a computer plot a path which has a distinctive shape regardless of the initial points. The program, for simplicity, ignores the z-plane and just gives a two dimensional image in the x,y planes.

This program is only an approximation to the Lorenz equations, but the shape is fairly represented. The paper to look for is 1963, E Lorenz, "Deterministic Nonperiodic Flow".

```
80 H=0.01 :: M=4.0 :: O=99
90 X,Y,Z=0.6
100 F=8/3
110 FOR T=1 TO 9940
120 XN=X+H*10*(Y-X)
130 YN=Y+H*((-X*Z)+28*X-Y)
140 ZN=Z+H*(X*Y-F*Z)
150 Z=ZN
160 CALL LINK("LINE",X*M+O,Y*M+O,XN*M+O,YN*M+O)
170 X=XN :: Y=YN
180 NEXT T
190 GOTO 190
```



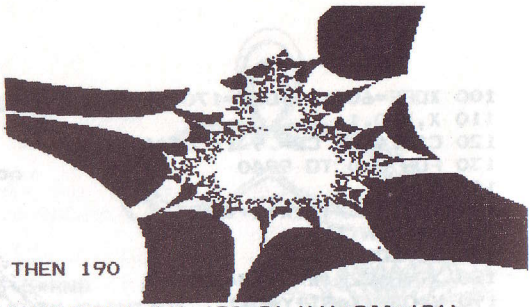
Short enough!

==========

The next two programs also come from Clifford Pickovers latest book, furst a slightly different way of plotting a Mandelbrot set. If you want to magnify a portion, adjust the parameters in lines 100 and 110, adjust the scaling in lines 100,110 and 190, and adjust the offset in 190 - we have done this plenty of times now. Write if you hit difficulties. Although written for The Missing Link, any program/language that addresses single pixels can be used wiith only minor adjustments.

----> continued---->

65

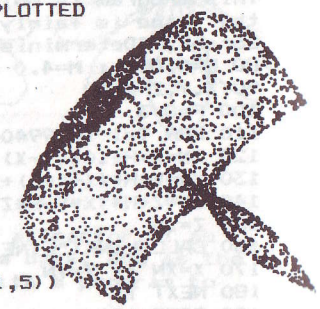


```
100 FOR X=-3 TO 3 STEP 6/190
110 FOR Y=-3 TO 3 STEP 6/240
120 CR=X :: CI=Y :: RX,RY=0
130 FOR K=0 TO 60
140 NEWX=RX*RX-RY*RY+CR
150 RY=2*RX*RY+CI
160 RX=NEWX
170 IF ABS(((RX-CR)+(RY-CI)))>4 THEN 190
180 NEXT K
190 IF K/2=INT(K/2)THEN CALL LINK("PIXEL",X/6*190+96,Y/6*240+121)
200 NEXT Y
210 NEXT X
220 GOTO 220
```

==========

This program is referred to by Dr Pickover as a million point graphic program, as he plotted 1000000 points - using a pretty high resolution plot, probably a laser printer. For speed we use a little less.

```
100 RANDOMIZE
110 CALL LINK("CLEAR")
120 A=2
130 B=1
140 ! try A=2 :: B=2 or even just RND! (A=B=0 or 1 is not so good)
150 X=INT(RND*40):: Y=INT(RND*40) ! FIRST POINT PLOTTED
160 CALL LINK("PRINT",1,1,STR$(X)&"    "&STR$(Y))
170 CALL LINK("PRINT",1,190,STR$(A)&"  "&STR$(B))
180 FOR T=1 TO 4000 ! 1E6 BETTER!
190 V1=SIN(Y*B):: V2=SIN(X*B)
200 V3=SIN(X*A):: V4=SIN(Y*A)
210 NX=V1+V2*V2
220 NY=V3+V4*V4
230 X=NX :: Y=NY
240 PX=50*X+70 :: PY=50*Y+70
250 CALL LINK("PIXEL",PX,PY)
260 NEXT T
270 CALL LINK("SAVEP","RD."&SEG$(STR$(RND*1E10),1,5))
280 RUN
```



Setting A and B to random numbers, the results seem to be either an almost random scatter (chaos) plot or a plot with a strong sinusoidal appearance.

Dr Pickover made some other suggestions for lines 210 and 220, but I have not found any attractive plots with them. Here are the variants for your exploration:

```
210 NX=V1+V2*V2+V2*V2*V2
220 NY=V3+V4*V4+V1*V1*V1
or
210 NX=V1+V2*V2+V2*V2*V2
220 NY=V3+V4*V4+V4*V4*V4
or
210 NX=V1+V2*V2+V2*V2*V2
220 NY=V3+V1*V1+V1*V1*V1
```

==========

INDEX TO TI*MES LAST EIGHT ISSUES - NUMBERS 27 TO 34:

I S S U E	P A G E		
33	43	32K RAM DISCUSSION ON USE .	ARTICLE-32K RAM
29	47	4A FLYER module review. . . .	REVIEW-MODULE-GAME
34	41	ADDING FRACTIONS 1.	PROGRAM-XB-MATH
34	45	ADDING FRACTIONS 2.	PROGRAM-XB-MATH
33	54	ADDITION OF LARGE NUMBERS. .	PROGRAM-XB-MATH
28	58	AFFINE PROGRAM + POLYNOMIALS.	PROGRAM-MYARC XB-GR
33	01	AGM 1991 REPORT.	REPORT ON 1991 AGM
30	01	AGM MINUTES CHESTER 1990. . .	AGM MINUTES
29	30	AGM REPORT 1990.	REPORT-1990 AGM
27	01	AGM REPORT JUNE 89.	REPORT-AGM 89
31	56	ALPHALOCK-UP OR DOWN? PROG. .	PROGRAM-M/C-XB UTIL
28	54	ALPINER	REVIEW-MODULE-GAME
28	51	AMBULANCE module.	REVIEW-MODULE-GAME
27	25	AMNION SOFTWARE appreciation.	OBITUARY
34	52	ANALYTIC COMPUTER ART PROG. .	PROGRAM-XB+TML-GRPH
30	42	ANCIENT COMPUTERS 1982 & 1984	HISTORY-OLD H/W
28	52	ANTEATER	REVIEW-MODULE-GAME
33	46	ARROW DEFINITIONS.	TIP-BASIC-CHAR DEFS
33	41	ARTIST CATALOGUER REVIEW. . .	REVIEW-DISK-GRAPHIC
28	40	ARTIST PRINT SHOP REVIEW. . .	REVIEW-DISK-UTIL
31	64	ASC("LONG STRING").	TIP-BASIC
32	24	ASGARD PROGS JAN 91.	CATALOG-ASGARD
34	36	ATTRACTION BASIC Z^(-3)-1. .	PROGRAM-XB+TML-GRPH
32	37	BACKSTEINE game disk review.	REVIEW-DISK-GAME
30	46	BALLOON WARS-Asgard-disk-revu	REVIEW-DISK-GAME-XB
28	52	BARRAGE module.	REVIEW-MODULE-GAME
31	42	BASIC TIPS-SWOPPING VALUES. .	TUTORIAL-BASIC
32	30	BASIC-MAKING BETTER USE OF. .	TIP-BASIC
32	40	BASIC-PROGRAMMING FASTER. . .	TIP-BASIC
32	30	BETTER BASIC PROGRAMS-sample.	TIPS-BASIC PROGRAMS
28	52	BEYOND PARSEC.	REVIEW-MODULE-GAME
28	45	BEYOND VIDEO CHESS review. .	REVIEW-DISK-GAMEUTL
30	49	BIFURCATED SOUNDS.	PROGRAM-SOUNDS-DEMO
31	67	BIFURCATION PLOT (affine). .	PROGRAM-XB-GRAPHIC
31	72	BIFURCATION PLOT (Becker). .	PROGRAM-XB-GRAPHIC
28	53	BIG FOOT	REVIEW-MODULE-GAME
27	19	BINARY DATA COMPACTION. . . .	TIP-DATA STORAGE
33	51	BINARY NUMBERS.	ARTICLE-NUMBER SYST
32	24	BIRDWELL,JOHN DEATH OF. . . .	OBIT
28	52	BLACK HOLE.	REVIEW-MODULE-GAME
27	09	BLACKBEARDS TREASURE-REVIEW.	REVIEW-TAPE-GAME
28	54	BLACKJACK & POKER module. . .	REVIEW-MODULE-GAME
27	09	BLAST IT- REVIEW.	REVIEW-TAPE-GAME
28	53	BUCK ROGERS PLANET OF DOOM. .	REVIEW-MODULE-GAME
28	53	BURGER BUILDER.	REVIEW-MODULE-GAME
28	52	BURGER TIME.	REVIEW-MODULE-GAME