

This document contains the collected submissions of Stephen Shaw to the TI99/4A UK User Group Magazine TI*MES over the approximate period 1987 to 1991, with a few outside that period.

Not everything that was published has survived on disk- however an increasing number of issues of TI*MES are being scanned and stored safely online at [http://ftp.whtech.com/user groups/TI-MES-England/Times/](http://ftp.whtech.com/user_groups/TI-MES-England/Times/) which includes submissions from other writers. Some of the text below was retyped from other user group magazines and original sourcing is given where possible.

=====

TEXAS INSTRUMENTS TI-99/4A HOME COMPUTER
U.K.User TI99/4a Group - publication TI*MES

=====

<APOLOGIES FOR ANY DUPLICATED ARTICLES, lots of copying and pasting...>

RAMBLES..

OCT 1987. Stephen Shaw..

My address:.

10 Alstone Road, STOCKPORT, Ches, SK4 5AH..

.
THE SMART PROGRAMMER is still a NO NO with deliveries still not being made. The last issue received by me has a cover date of DECEMBER 1986 and was received on 22nd June 1987. This issue has some ads in it, as I was forewarned by a little rumour, and MAY confirm another rumour that funds are running out, not just time... the publisher has failed to advise his subscribers of what he is up to, and must therefore accept responsibility for any rumours that appear..

.
GOOD SUPPLY/ BAD SUPPLY. This is a personal list! The following have given me GOOD and FAST service:.

TENEX, TEXAMENTS, NOT POLYOPTICS, MICROpendium..

.
THE FOLLOWING ARE DOUBTFUL:.

Pilgrims Pride, Tex Comp, Asgard, Smart Programmer, Chicago User Group, Boston Computer Society, RYTE Data..

.
On NOT POLYOPTICS I understand one member sent for SPAD XIII from them, enclosing US\$34, and received his disk just ten days after posting his (airmail) order. .

.
On Pilgrims Pride, word is the owner came close to losing his life - and that of his family- in bad flooding, but did lose his business documentation. Latest report is that goods remain undelivered!.

.
On Tex*Comp, various grumbles, but they do have a good stock including some new module titles. I may be able to obtain goods from them "retail" if anyone wants anything (via a US friend)..

.
On ASGARD, quite apart from making a mess of my last two orders, they have now REALLY ripped me off, advertising a new copy of FONT WRITER some four or five months before it was written - and sending me the old version. I understand the program author is also a little unhappy. If you want an Asgard program buy from TENEX or a reputable User Group (or possibly through me)..

=====

.
BASIC COMPILER: Ryte Data have not charged my credit card and I have not heard from them. I have heard grumbles elsewhere about unduly long delivery times, and I have two very short reports on the Compiler not being so hot..

"BASIC-COMPILER" from Ryte Data:

No, Ryte Data have not sent me a copy, but your faithful reporter has tracked a copy down in order to report to you on its value..

Firstly, whatever it does, it is NOT a compiler, and anyone selling the program in this country would be a sure loser. My dictionary defines a compiler as a program which produces machine code. .

Here is a quote from the documentation: " Principally the Compiler works in a manner similaire [not my spelling error!] to the Basic-Interpreter that resides in the Console Rom's".

There is a little difference between a compilation and interpretation!.

If you look back to TI*MES No 15, you will see some bench mark programs- not the final measure of a language but at least a rough guide..

TRIGLOG timing- using floating point arithmetic:.

TI Extended Basic= 362 seconds..

Ryte Data "compilation" = 600 seconds!!!.

.
TEXTSCREEN timing- using floating point arithmetic in TI ExBas but using Integer arithmetic for the compilation (supposed to be faster):.

TI Extended Basic = 117 seconds.

Ryte Data "compilation" = 168 seconds!.

(Remember, Myarc XB = 73 sec!).

.
Quote "After the Compiler creates the new version of your program, the Loader can then run the new, faster version." Faster???.

Quote: "With large programs it can be quite a time saver to use the Basic-Compiler". Possibly..

However, the program to be "compiled" must be somewhat shorter than usual! A program which starts as 19 sectors can become 35 sectors after "compilation" while a program starting at 24 sectors can become 37 sectors. It is not possible to indicate a maximum size the "compiler" can handle, you must use trial and error!.

The error message is "PROGRAM TOO BIG" (and no that isn't my spelling error). And this error message is not documented either.

You can of course chain programs together using "RUN DSK1.PROGRAM2"- and you know how long loading can take when you do that....

Faster on long programs? I can't say, as I could not find any programs of 20 odd sectors which would run successfully after "compilation", even after deleting the things the documentation says you can't do:.

NO: DEFs, SUBs, ON ERRORs, LOAD, LINK, ON BREAK..

NOT DOCUMENTED, but you also cannot use ON WARNING..

Maximum string length is declared and subject to an absolute maximum of 64 characters..

I could not find any reason why the programs which failed did so, they were perfectly in accordance with the documentation!.

I did notice that the cursor was not flashing as quickly as it usually does.

The title screen is (c)1984. It is now 1987... and NO major reviews? No reviews full stop... Guess nobody has made it work???

Suppose it did speed things up... it would not help the majority of games programs, as they are dependent on internal timing- it is no good if the alien destroys you before you see it! So rewriting would be necessary to slow some things down!

SPIDER BOP REVISITED or YOUR FIRST c99 GAME:.

.
Using c99! c99 is available on four disks from the Group Disk Library, together with all the other files listed below. c99 Version 2.1 or higher is assumed..

.
To use this listing you will need the following files:.
C99C, C99D, C99E, GRF1RF, GRF1, RANDOM;C, CONC;C, CSUP.

.
First type in the following listing, using the Funlweb Editor set to Ed/As mode or use the Ed/As editor. Save the text as SPIDER;C.

.
Now load C99 (you may use Funlweb loader Option 4, LOAD & RUN).

.
Just press enter for the first two questions, enter source name as DSK1.SPIDER;C and output name as DSK1.SPIDER/S..

.
(If you receive any error messages look at both the line reported and the line above it. I missed out two final semi colons at first! You will need to go back to your Editor to change the C source)..

When the c99 program has finished, you must load the TI Assembler, and assemble the file SPIDER/S into SPIDER. Do NOT use any assembly options, not even R!!!.

.
Now you are ready to play your game..

.
Select Option 4 of Funlweb Loaders (LOAD AND RUN) and load the following files, one after the other:.

SPIDER, CSUP, GRF1.

Then just press enter, then enter the start label: START and off you go..

.
You have just entered a C99 game. Not too bad was it! Just a note: speed 9 is slow, speed 1 is suicide!.

.
Here is the code:..

```
#asm
  REF C$GPLL
HONK BL @C$GPLL
  DATA >36
  B *13
#endasm
#include "dsk1.grf1rf"
#include "dsk1.random;c"
#include "dsk1.conv;c"
int z,temp,a,b,c,e,k,s,j,tt,flg;
int d[32];
char score[8];
main()
{
while (0==0)
{ clear(); grf1(); screen(2);
  chrdef(096,"1818181818181818");
  chrdef(097,"000000FFFF000000");
  chrdef(104,"6699997E5595A5A5");
  chrdef(112,"3C7EFFFFFF7E3C3C");
  chrdef(113,"3C3C7EFFFFFF7E3C3C");
  e=0;
  b=12;
```

```

tt=0;
while (e++<9)
{ color(e+4,16,1);
}
color(13,16,1);
color(14,14,1);
color(15,3,1);
e=0;
while (e++<33)
{ d[e]=3;
}
locate(4,10);
puts("SPIDER BOP 2");
locate(22,5);
puts("USE <ARROW> KEYS TO MOVE");
locate(8,8);
puts("WRITTEN WITH C99");
locate(9,9);
puts("BY JOHN BEHNKE");
locate(24,3);
puts("PRESS /ENTER/ TO BOP SPIDERS");
locate(12,11);
puts("SPEED 0-9?");
k=0;
while ((k<48)|(k>57))
{ k=key(0,&s);
}
j=k-47;
j=j*100;
clear();
locate(1,10);
puts("SCORE:");
hchar(3,1,96,32);
hchar(4,1,104,32);
hchar(24,b,112,1);
randomize();
a=rnd(32)+1;
flg=0;
while (flg==0)
{ a=rnd(32)+1;
++d[a];
k=0;
while (k<j)
{ ++k;
}
vchar(d[a]-1,a,96,1);
vchar(d[a],a,104,1);
k=key(0,&s);
while ((k==68)|(k==83))
{ hchar(24,b,32,1);
if (k==68) { ++b; }
if (k==83) { --b; }
if (b==0) { b=32; }
if (b==33) { b=1; }
hchar(24,b,112,1);
k=0;
}
if (k==13)

```

```

{ k=24;
  temp=5;
  while (--k>temp)
  { hchar(k,b,112,1);
    z=0;
    while (z<50) { ++z; }
    if (k==d[b])
    { honk();
      z=0;
      while (z<300) { ++z; }
      d[b]=d[b]-3;
      if (d[b]<5) { d[b]==5; }
      vchar(d[b],b,32,23-d[b]);
      vchar(d[b],b,104,1);
      locate(1,15);
      temp=24;
      itod(++tt,score,6);
      puts(score);
    }
  }
  while (k<24)
  { hchar(k++,b,32,1);
    z=0;
    while (z<50) { ++z; }
  }
}
if (d[a]==24)
{ hchar(24,a,32,1);
  k=0;
  if (a>b) { k=-1; }
  if (b>a) { k=1; }
  while ((k!=0)&(a!=b))
  { a=a+k;
    hchar(24,a,104,1);
    z=0;
    while (z<j) { ++z; }
    hchar(24,a,32,1);
  }
  hchar(24,a,104,1);
  locate(18,10);
  puts(" GAME OVER ");
  locate(20,5);
  puts("PRESS /ENTER/ TO REPLAY");
  k=0;
  while (k!=13)
  { k=key(0,&s);
  }
  flg=1;
}
}
}
}
/* END OF C SOURCE */
.....

=====

```

USING CHARA1 in EXTENDED BASIC.

.
Edward H Shaw has kindly sent me a routine to enable you to load one of the CHARA1 files into your own XB program, giving you access to a true lower case font- or a font of your own design (see last issue!).. CHARA1 font definition files are supplied by TI on disk eg with TI Writer.

.
Here is the routine as Edward sent it:.

```
.  
*****  
PAB    DATA >0500,>02FA,>0000,>0800,>000B  
        TEXT 'DSK1.CHARA1'  
        EVEN  
SAVR11 BSS 2  
TAB    DATA >0000,>0000,>0000,>0000  
        DEF A  
A      MOV R11,@SAVR11  
        LI R0,>1000  
        LI R1,PAB  
        LI R2,>0015  
        BLWP @>2024 VMBW  
        LI R0,>1009  
        MOV R0,@>8356  
        BLWP @>2532 DSRLNK  
        DATA >0008  
        LI R0,>03F8  
        LI R1,TAB  
        LI R2,8  
        BLWP @>2024 vmbw  
        CLR @>837C  
        MOV @SAVR11,R11  
        B *R11  
        END
```

.
You also need a DSRLNK for Extended Basic - there are several around..

.
Run the following Extended Basic program, which assumes that the object code produced from the above source is called LOWCAS, and that you have DSRLNK and CHARA1 on DSK1 as well:.

```
.  
100 CALL INIT.  
110 CALL LOAD("DSK1.DSRLNK","DSK1.LOWCAS").  
120 CALL LINK("A").  
130 PRINT "ABCDEabcde".  
140 GOTO 140.
```

.
[Ed also inserted CALL DELSPRITE(ALL)].

continued.....

THIS PROGRAM WILL LOAD THE FILE CHARA1 FROM DSK1. Lower case will retain its loaded definition after the program ends..

```
.  
*****
```

continued.....

You dont have a DSRLNK handy? FI! OK here is the same code- adjusted slightly!
- with a DSRLNK inserted, and 'cos the DSRLNK needs it, you get a GPLLNK thrown
in as well..

The DSRLNK and GPLLNK come from The Smart Programmer magazine..

```
.
-----
                DEF  FONT
VMBW   EQU  >2024
*****
* NEXT BIT IS A GPLLNK ROUTINE
* REQUIRED BY FOLLOWING DSRLNK
* Craig Miller & D C Warren
* from THE SMART PROGRAMMER
*****
GPLWS  EQU  >83E0
GR4    EQU  GPLWS+8
GR6    EQU  GPLWS+12
STKPNT EQU  >8373
LDGADD EQU  >60
XTAB27 EQU  >200E
GETSTK EQU  >166C
GPLLNK DATA GLNKWS   *R7
        DATA GLINK1  *R8
RTNAD  DATA XMLRTN  *R9
GXMLAD DATA >176C   *R10
        DATA >50     *R11
GLNKWS EQU  $->18
        BSS  >08      *R12>15
GLINK1 MOV  *R11,@GR4
        MOV  *R14+,@GR6
        MOV  @XTAB27,R12
        MOV  R9,@XTAB27
        LWPI GPLWS
        BL   *R4
        MOV  @GXMLAD,@>8302(R4)
        INCT @STKPNT
        B    @LDGADD
XMLRTN MOV  @GETSTK,R4
        BL   *R4
        LWPI GLNKWS
        MOV  R12,@XTAB27
        RTWP
*****
* NOW THE DSRLNK BIT
*****
PUTSTK EQU  >50
TYPE   EQU  >836D
NAMLEN EQU  >8356
VWA    EQU  >8C02
VRD    EQU  >8800
GR4LB  EQU  >83E9
GSTAT  EQU  >837C
DSRLNK DATA DSRWS,DLINK1
DSRWS  EQU  $
DR3LB  EQU  $+7
DLINK1 MOV  R12,R12
        JNE  DLINK3
```

```

*****
* NEXT BIT ONLY USED ON
* FIRST DSRLNK IN OUR
* PROGRAM - OUR PROGRAM ACTUALLY
* ONLY CALLS DSRLNK ONCE ANYWAY!
*****
        LWPI GPLWS
        MOV  @PUTSTK,R4
        BL   *R4
        LI   R4,>11
        MOVB R4,@>402(R13)
        JMP  DLINK2
        DATA 0
        DATA 0,0,0
DLINK2 MOVB @GR4LB,@>402(R13)
        MOV  @GETSTK,R5
        MOVB *R13,@DSRAD1
        INCT @DSRADD
        BL   *R5
        LWPI DSRWS
        LI   R12,>2000
*****
DLINK3 INC  R14
        MOVB *R14+,@TYPE
        MOV  @NAMLEN,R3
        AI   R3,-8
        BLWP @GPLLNK
DSRADD BYTE >03
DSRAD1 BYTE >00
*****
* ERROR CHECK
*****
        MOVB @DR3LB,@VWA
        MOVB R3,@VWA
        SZCB R12,R15
        MOVB @VRD,R3
        SRL  R3,5
        MOVB R3,*R13
        JNE  SETEQ
        COC  @GSTAT,R12
        JNE  DSREND
SETEQ  SOCB R12,R15
DSREND RTWP
*****
* now to load CHARA1
*****
PAB    DATA >0500,>02FA,>0000,>0800,>000B
        TEXT 'DSK1.CHARA1'
        EVEN
SAVR11 BSS  2
TAB    DATA >0000,>0000,>0000,>0000
FONT   MOV  R11,@SAVR11
        LI   R0,>1000
        LI   R1,PAB
        LI   R2,>0015
        BLWP @VMBW
        LI   R0,>1009
        MOV  R0,@>8356

```



```
BLWP @DSRLNK
DATA >0008
LI R0,>03F8
LI R1,TAB
LI R2,8
BLWP @VMBW
CLR @>837C
MOV @SAVR11,R11
B *R11
END
```

.
Now assemble. This listing produced directly from source code which has been assembled and which works!

.....
=====

CONTENTS ARE CONTINUED ON NEXT PAGE ----->

LOAD MAKER :

This EXTENDED BASIC program requires a disk drive. It will read the disk menu and create a special LOAD program with its own Menu for your disk.. Please refer to the further notes at the end of the listing!.

```
100 ! LOADMAKER
110 ! VERSION XB.1.1
120 ! 29 DEC 84
130 ! by Jim Swedlow
140 ! from SPIRIT OF 99
    July 1987
150 DISPLAY AT(10,10)ERASE ALL:"LOADMAKER": : : : : "Initialising..." : @=1 : DIM A$(26),A(26):: B$=CHR$(182)&CHR$(181)&CHR$(199)
160 B=100 : C$=CHR$(179)
170 OPEN #@"DSK1.",INPUT ,INTERNAL,RELATIVE : INPUT #@:A$(C),D,E,F : DISPLAY AT(16,@):"Disk ";A$(C);" * Free" ;F : A(C)=LEN(A$(C))
180 INPUT #@:D$,D,E,F : IF D$="" THEN 220 ELSE IF ABS(D)<>5 OR D$="LOAD" THEN 180
190 C=C+@ : IF C<27 THEN A$(C)=D$ : A(C)=LEN(D$):: DISPLAY AT(17,@):"Reading:";D$ : GOTO 180
200 DISPLAY AT(16,@)BEEP:"The disk has more than 26 programs. Do you wish to proceed with the first 26 programs? Y or N?"
210 CALL KEY(3,F,D):: IF F=78 THEN CLOSE #@ : STOP ELSE IF F<>89 THEN 210
220 IF C=0 THEN DISPLAY AT(16,@)BEEP:"No programs were found on this disk." : CLOSE #@ : STOP
230 CLOSE #@ : DISPLAY AT(16,@):"Reading completed": : : : : OPEN #@"DSK1.XXX",VARIABLE 163,DISPLAY ,OUTPUT
240 DISPLAY AT(16,@):"--Making LOAD--line 100"
250 D$=CHR$(0)&CHR$(100)&CHR$(131)&CHR$(32)&CHR$(80)&CHR$(79)&CHR$(71)&CHR$(82)&CHR$(65)&CHR$(77)
260 D$=D$&CHR$(32)&CHR$(76)&CHR$(79)&CHR$(65)&CHR$(68)&CHR$(69)&CHR$(82)
270 G,F=@ : GOSUB 490 : D$=D$&CHR$(182)&CHR$(239)&CHR$(236)
----to next column...
280 D$=D$&CHR$(181)&CHR$(199)&CHR$(7+A(0))&CHR$(68)&CHR$(105)&CHR$(115)&CHR$(107)&CHR$(32)&CHR$(42)&CHR$(32)&A$(0).
290 G=@-7*(C<19):: E=2 : FOR D=@ TO C : E=E-(D>C/2)*(G=@)*INT(C/2):: G=G+(D>C/2)*(G=@)*14 : IF D=C AND G=15 AND C/2<>INT(C/2)THEN G=8.
300 F=E+D : GOSUB 490 : D$=D$&B$&CHR$(A(D)+3)&CHR$(64+D)&CHR$(32)&CHR$(32)&A$(D):: NEXT D.
310 F=24 : G=@ : GOSUB 490 : D$=D$&CHR$(182)&CHR$(238)&CHR$(181)&CHR$(199)&CHR$(17).
320 D$=D$&CHR$(80)&CHR$(114)&CHR$(101)&CHR$(115)&CHR$(115)&CHR$(32)&CHR$(121)&CHR$(11)&CHR$(117).
330 D$=D$&CHR$(114)&CHR$(32)&CHR$(99)&CHR$(104)&CHR$(111)&CHR$(105)&CHR$(99)&CHR$(101).
340 GOSUB 470 : D$=D$&CHR$(157)&CHR$(200)&CHR$(3)&CHR$(75)&CHR$(69)&CHR$(89)&CHR$(183).
350 D$=D$&CHR$(200)&CHR$(@)&CHR$(51)&C$&CHR$(75)&C$&CHR$(83)&CHR$(182)&CHR$(130).
360 D$=D$&CHR$(132)&CHR$(75)&CHR$(191)&CHR$(200)&CHR$(2)&CHR$(54)&CHR$(53)&CHR$(186)&CHR$(75)&CHR$(192).
370 F=64+C : GOSUB 500 : D$=D$&CHR$(176)&CHR$(201):: GOSUB 480.
380 F=24 : GOSUB 490 : D$=D$&B$&CHR$(7)&CHR$(76)&CHR$(111)&CHR$(97)&CHR$(100)&CHR$(105)&CHR$(110)&CHR$(103).
390 D$=D$&CHR$(130)&CHR$(155)&CHR$(75)&CHR$(194)&CHR$(200)&CHR$(2)&CHR$(54)&CHR$(52)&CHR$(134).
400 D=B : FOR B=B+10 TO B+10*C STEP 10 : IF B>D+10 THEN D$=D$&C$.
410 D$=D$&CHR$(201):: GOSUB 480 : NEXT B : B=D.
```

Loadmaker continued:

```
420 FOR D=@ TO C :: F=24 ::          480 D$=D$&CHR$(INT(B/256))&C .
G=9 :: GOSUB 490 :: D$=D$&B$      HR$(B-256*INT(B/256)):: RETU .
&CHR$(A(D))&A$(D)&CHR$(130)&      RN.
CHR$(169)&CHR$(199)              490 GOSUB 470 :: D$=D$&CHR$( .
430 D$=D$&CHR$(A(D)+5)&CHR$(      162)&CHR$(240)&CHR$(183):: G .
68)&CHR$(83)&CHR$(75)&CHR$(4      OSUB 500 :: D$=D$&C$ :: F=G.
9)&CHR$(46)&A$(D):: NEXT D       500 IF F<10 THEN D$=D$&CHR$( .
440 PRINT #@:D$&CHR$(0):CHR$      200)&CHR$(@)&CHR$(48+F):: RE .
(255)&CHR$(255):: CLOSE #@      TURN ELSE D$=D$&CHR$(200)&CH .
450 DISPLAY AT(16,@)BEEP:"Pr     R$(2)&CHR$(48+INT(F/10))&CHR .
ogram now on disk": "Now type    $(48+F-10*INT(F/10)).
in:" " NEW" " MERGE DSK1.       510 RETURN.
XXX" " SAVE DSK1.LOAD"         520 END.
460 DISPLAY AT(22,@): "And th    .
en you can try it by typing    .
RUN " :: STOP                  .
470 B=B+10 :: PRINT #@:D$&CH    .
R$(0):: D$="" :: DISPLAY AT(    .
16,20):B                       .
```

Listing produced with Tony McGovern's CO-LIST program, available on disk from the disk library. .

The above program reads the disk catalogue and writes a program which you can call LOAD, which allows you to select a program on the disk from a menu, and then the chosen file will load and run..

There are some limitations however!.

- a. In its present form it can only handle a disk which has 26 or fewer PROGRAM files..
- b. It treats all PROGRAM files as though they were XB/TIB - that is, it will include on the menu any memory image machine code programs, graphic images, character sets and so on. This will not trouble you if you keep all your Basic/Extended Basic programs together on disks with no other form of memory image files!.
- c. However, it will NOT include any long Extended Basic programs which are stored on disk as I-V 254 files. To catch these as well you will need to do a little alteration to LINE 180.

The listing has been produced from a proven working program! If you experience difficulties, check back every line most carefully!.

This program is remarkably quick in operation, and is a good example of a "program that writes programs"..

=====

RAMBLES by Stephen Shaw.
January 1988.

10 Alstone Road, STOCKPORT, Cheshire, SK4 5AH.

Two of our members have supplied some nice disks- from David Vincent a remarkable graphic adventure CARFAX ABBEY in which you explore a 4x5x5 maze to find a crucifix, vanquish Dracula, and escape with treasures! ExBas with machine code utilities. Also a LOT of code, so regret not possible to transfer to tape. From John Seager, a surprise. There are plenty of (unused) character

design utilities around. John has produced (almost) the ultimate- in XB with machine code links for speed. Superb- and also from John is a set of four utilities to help you to manipulate merge format program files. .

From a Boston Computer Society publication, **some tips** from David Taub **for ExBas** programmers:.

1. Is N odd or even?.

Normal:.

```
100 X=(N=2*INT(N/2))+1.
```

will set X=0 if N is even, and X=1 if N is odd..

Faster:.

```
100 X=N AND 1 does the same for values of N up to 32767..
```

2. Stop on space and continue on space:.

```
100 FOR P=1 TO 1000.
```

```
110 PRINT P.
```

```
120 CALL CHECK.
```

```
130 NEXT P.
```

```
140 GOTO 100.
```

```
.....
```

```
900 SUB CHECK.
```

```
910 ST=Y :: CALL KEY(0,Y,Z).
```

```
920 IF (Y<>32 OR Y=ST) XOR F=0 THEN F=1 :: GOTO 910.
```

```
930 F=0 .
```

```
940 SUBEND.
```

TRY IT!.

=====

Some short c routines to get you used to using c99 and maybe show how some things are done / some things are used.

These routines are by Donald L Mahler and come from the BOSTON COMPUTER SOCIETY. They have been printed from tested source code.

Remember:

*s means "pointer to s" while

&s means "the address of s"

File prf is as follows:

```
/* file dsk1.prf */
```

```
/* PRINTF REFS */
```

```
#asm
```

```
REF PRINTF
```

```
#endasm
```

Save it to disk!

```
/* 1;C */
```

```
#include dsk1.prf
```

```
int table[]={3,5,2,9,6};
```

```
/* sets up an array */
```

```
/* see next page for more */
```

```

main()
{
    int i; i=0;
    /* first term of array is "0th" */
    while (i<5)
    {
        printf("The address of the %dth \n",i);
        printf("element of table is %u.\n",&table[i]);
        /*          */
        /* "&table[i]" = */
        /* "address of ith term of array" */
        /*          */
        /* addresses are unsigned integers */
        /* that is why we use 'u' */
        /*          */
        printf(".. and the value stored there\n");
        printf("is %d\n",*(&table[i]));
        /*          */
        ++i;
        /*          */
        /* increment i */
        /*          */
        putchar('\n');
    }
}

```

Type this in using the Assembler editor, save it and then compile it,. assemble the result, to say 1/OB. Do NOT select any assembly OPTIONS!. To run using LOAD AND RUN load:.

```

DSK1.1/OB
DSK1.PRINTF
DSK1.CSUP

```

then start with program name START.

CSUP and PRINTF are supplied with the c99 package.

This second routine uses strings- and also requires the file prf defined above!

```

/* 2;c          */
#include dsk1.prf
main()
{
    char *ptr1, *ptr2 ;
    /* two character pointers */
    ptr1="Boston/Computer/Society";
    /*          */
    /* the address of a string is */
    /* the address of first letter */
    /*          */
    ptr2=ptr1 ;
    while (*ptr2)
    /* continued ->          */

```

```

/*  *ptr2 !=0" */
/*      */
{
    putchar(*ptr2++) ;
}
/*      */
/* spell out the string letter */
/* by letter      */
/*      */
puts("\n \n Now let's reverse it! \n\n");
/*      */
/* ptr2 is now address  of last */
/* letter of string!!!          */
/*      */
while (--ptr2 >= ptr1)
/*      */
/* decrease address until back */
/* at original starting address */
/*      */
{
    putchar (*ptr2);
}
putchar('\n');
}

```

And here is another short example of c99 in action. Try it out now!

```

/* 3;c      */
#include dsk1.prf
main()
{
    char x;
    puts("Enter any letter : \n\n");
    x=getchar();
    putchar('\n');
    printf("The upper case form of %c is",x);
    caps(&x);
    putchar(x);
    putchar('\n');
}
caps(ptr)
char *ptr;
{
    if (*ptr <= 'z' & *ptr >= 'a')
        *ptr = *ptr + 'A' - 'a' ;
}
/*      */
/* if letter is lower case then */
/* decrease ascii value by the */
/* difference between 'A' (65) */
/* and 'a' (97)      */

```

---->

Now compile,
assemble (remember, NO options!)
and load and run:
DSK1.3/0
DSK1.PRINTF
DSK1.CSUP
and program start name is START.

.
If however you wish to transform your program.
to memory image format, to use with .
RUN PROGRAM FILE, then load these files,
using LOAD AND RUN:.

DSK1.C99PFI.
DSK1.2;0.
DSK1.PRINTF.
DSK1.CSUP.
DSK1.C99PFF.
DSK1.FWSAVE.
and now choose the program named SAVE..

.
Now you will have a single "PROGRAM".
file which you can load in one piece,.
instead of having to load lots of other.
files.

.....
=====

.
MAGAZINE REVIEWS....

.
Chicago Times Sept issue:.

News of a considerable upgrade to **SPAD XIII**, the machine code flight simulator
by Not Polyoptics, with a RED BARON option, and lots more function keys,
including one to let you see your own plane in flight! And an article which
talked you through a simple flight on SPAD XIII

.
Chicago Times Oct issue (74 pages!):

A further exploration of SPAD XIII- which, unlike Microsoft's flight
simulator, has trees!

AUTO LOAD CALL FILES(X)

PROBLEM: You have a utility program which you want to autoload on selection of
XB, BUT the program insists on a CALL FILES other than the default of 3. You do
not want to have to remember to do a CALL FILES(X)::NEW before you load the
program. What to do?.

Answer- this is XB but requires 32k ram and *may* not work on all set-ups!.
First obtain values of M1 and M2 as follows:.

Select XB, and with no program resident, type in:.

CALL FILES(X) - insert YOUR required value in place of X!.

NEW.

CALL PEEK(-31888,M1,M2).

PRINT M1;M2.

.
continued.....

----> **Chicago TImes Oct issue (74 pages!):** CONTINUED

Note these down. Now write a LOAD program for your utility disk like this..

```
100 DISPLAY AT(12,3)ERASE ALL:"ONE MOMENT..."
```

```
110 CALL INIT
```

```
120 CALL LOAD(-31888,M1,M2) ! INSERT THE NOTED VALUES HERE INSTEAD OF M1,M2
```

```
130 ON ERROR 150
```

```
140 OPEN #1:"NOSUCH.DEVICE" ! JUST like this!
```

```
150 ON ERROR 170
```

```
160 CLOSE #1
```

```
170 RUN "DSK1.MYPROGRAM" ! YOUR file name here!
```

Thanks to Rich Klein and Mike Maksimik.

A letter to The Editor from D Hathaway, suggesting that the life of the module port can be extended by switching the console OFF before inserting or removing a module - the theory being that TINY arcing between the contacts on insertion or removal takes place if the power is on, and gently removes micron by micron the protective plating on the contacts..

Reviews of new public domain, fairware and commercial items-

DM1000 Vn 3.8 is for the Geneve!

A review of Ryte Data's BASIC COMPILER- I shall go back and have another look at it (nope, it still won't do anything for me. Remains a no no!).

And a demo of how to make music in assembly language with very helpful comments..

Part two of an article on the various graphics programs..

The Chicago User Group has around 800+ members and welcomes overseas subscriptions. They would like you to send US\$24 for 12 months sub to:
Chicago TI Users Group, P O Box 578341 Chicago IL USA 60657.
Allow 6 to 10 weeks for a response! .

..
.MICROpendium August 87 issue:.

Article on Basic strings by Regena, the start of a new column on C99- starting at the beginning for beginners!, a long 'animated video' written in LOG02, with music!, and news, views, reviews, programs and ads..

MICROpendium, TX USA

..
.TopIcs - LA99er Computer Group. October issue.

Review of Texloader freeware program. News that MICROpendium advertiser Innovative Programming are leaving trails of tears (beware). .

TI-LINES Vol 4 Issue 5 Oct 87:.published by Peter Brooks, Oxford:

TI WRITER FORMATTER BUG:

An enquiry from John Seager on how to make the **TI Writer formatter CENTRE lines CORRECTLY** which Pete Brooks was unable to answer... to confound those TI Writer users for whom .CE has always worked perfectly....

The answer is easy: If you wish to use .CE you MUST first use .LM and set it to a value of 1 or higher. .

This is probably a bug in the Formatter program, but .CE does not like it at all if you rely on the default left margin setting, and even gets confused if you try to use .LM 0..

EAST ANGLIA REGION 99ERS..

Scott and JoAnn Copeland.

October 1987:.

Neat one-liner to right adjust money amounts, so that a numeric input of 4 or 4.5 or 4.67 neatly line up:.

If the amount is X:.

```
X$=STR$(X+0.001) :: X$=SEG$(X$,1,LEN(X$)-1).
```

then to print out, with say the pence in column 12:.

```
C=12-LEN(X$)+1 :: DISPLAY AT(ROW,C):X$.
```

No matter what the amount, the one liner will ensure that when printed as a string there will always be two decimal places. Neat..

Thanks to Joe Quigley..

Their October issue also carried a map of ZORK II and some hints.

.

.....

=====

submitted april 1988

RAMBLES.. April 1988. by Stephen Shaw.

. 10 Alstone Road, STOCKPORT, Cheshire, SK4 5AH.

GARBAGE COLLECTION and how to avoid awkward pauses.

First an item for Extended Basic users with 32k ram... one of the things Myarc intended to include in their XB was a forced garbage collection, but they seem to have run out of room. It is really quite easy, but what am I talking about?.

Type in this program and run it:.

```
100 CALL CLEAR.
110 A$="1234567890".
120 FOR T=1 TO 12.
130 CALL SPRITE(#T,42,2,30+T*3,30+T*3,0,34+T).
140 NEXT T.
150 CALL PEEK(-31890,I,M).
160 CALL PEEK(-31974,H,L).
170 A$=A$&A$&A$&A$.
180 DISPLAY AT(20,15):"FREE:".
190 F=(H*256+L)-(I*256+M).
200 DISPLAY AT(21,15):F.
210 GOTO 140.
220 END.
```

.

The number you see counting down is the free stack space- it is being filled with redundant definitions of A\$ (line 170). When the number reaches/passes zero, you will see that it starts again from a high value- but at the moment of changeover, all your sprites will halt briefly. During this brief halt, the computer is checking the stack to see what information there is redundant (no longer required) and getting rid of anything it doesnt need- garbage collection..

.

Garbage collection not only causes sprites to halt but can also interfere with music, causing the odd note to sound for longer. Instead of having garbage collection occur when the stack is full - which can be quite often if the program is long and stack space short- it can be useful to force a garbage collection at a time we choose, when the effect may not be too obvious. Also, if we do a garbage collection before the stack is full, the pause may be quite a bit shorter..

.

Here's how to do it... insert the following lines into the above program:
95 CALL INIT.

....

```
205 CALL LOAD(-31885,144,"",-31858,81,169,152,0).
```

Now run the program again. Notice that the stack space remains constant, as we are forcing a garbage collection after each definition of A\$. How is sprite speed affected?

Try a garbage collection at an intermediate stage:

```
204 IF F>10000 THEN 210.
```

Any difference? Have fun. Thanks to the Sydney User Group (TIsHUG) in Australia for this one, taken from their newsletters for March and July 1987..

LOGO.. at last a LOGO query, and a very interesting one too... the LOGO manual contains an ANIMAL program. If you key it in and build up the KNOWLEDGE a little, say by adding three or four extra choices, as the program runs, then save it to disk... you will find that RECALLing the procedure will lock your console up. Hmmm..

On Page 7 of the Logo manual, you can find the information that input lines are limited to 127 characters..

Here is the clue! Basic program lines are also limited to 127- 127 bytes after tokenisation has occurred, which is why you can sometimes key in less than the 5 screen lines XB allows you and still get a LINE TOO LONG error message..

Back to Logo... as you play ANIMAL, the KNOWLEDGE LIST gets longer and longer and longer... and once it exceeds 127 bytes, it becomes incapable of loading back into the console. The file is just too long for the routine which reads the disk and places the definitions where they belong.

Once you have been playing ANIMAL for a while, if you wish to save it, you must first reduce the LIST attached to KNOWLEDGE to under 127 bytes. You could do this in immediate mode or write a routine to do it- perhaps call it "RESET"..

DISK DRIVES: SECTORS/TRACKS/DENSITIES/SIDES...aaaagh!

To allay a little confusion:.

TI Standalone disk controller:.

Single Sided 40 track Single Density 360 sector 90k 9 sec/track.

TI PEB Card: above OR.

Double Sided 40 track single density 720 sector 180k 9 sec/track.

Myarc/Corcomp PEB cards:.

above OR:.

Double Sided 40 track double density 1440 sec 360k 18 sec/track OR.

1280 sec 320k 16 sec/track..

Myarc OPTIONAL extension to Myarc PEB Card:.

Double sided 80 track QUAD density (no other data to hand)..

>>>>>continued

DISK COMPATABILITY: .

A disk recorded on a system higher in the list can be read on a system lower in the list (except 80 track)..

A disk recorded on a system lower in the list may not be read on a system higher in the list..

Except that a single sided system may be able to read SOME files on a double sided disk..

The Corcomp and Myarc controllers can both read and write in single OR double density, so you can still record SSSD on these!.

.

DISK DRIVES: All systems should be capable of working with any 40 track "standard" disk drive. An 80 track drive is of value only if you have the extended Myarc controller card..

You may need to set small switches in the drives to match the controllers- if disk access time is variable, the TI Controller does not take advantage of faster access times whereas the CorComp and Myarc controllers can- Myarc can use an access time as fast as 6ms. There is no incompatibility unless the disk drive is TOO SLOW, which should be unlikely these days! ALL controllers can work with the same maximum access time as the original TI controller, which is much slower than any modern drive. The TI disk controller has a "hardware" fixed speed. The other controllers can be made to function more efficiently with the faster modern drives..

=====
.

A letter from a member dated December 1987 advises of the partial delivery of an order sent to PILGRIMS PRIDE in October 1986. Speedy!.

.....

MYARC RAM CARD:

Why can you only use 400k of the Myarc ram card for disk emulation? Because the disk usage is BIT MAPPED into an area of the disk which can only map 400k. There is no room to map more- if you want a bigger disk you must use a different mapping system- either increase the area in which the bit map is held, or allow a "floating" bit map on an "as required" basis. That would mean you would need a new disk controller- your ram card would not be strictly compatible with your old software! Myarc were right to stick to 400k. The remainder is available for print spooling, and for your operating system to use as part of the normal computer memory map area..

.....

FORTRAN from Tenex. George Michel reports as follows:" I have the LGMA FORTRAN discs and I have succeeded in producing working object programs.... I concluded that the LGMA discs apply to a very powerful extract of the ANSI 77 FULL language, and is not to be confused with the ANSI 77 SUBSET. .

"The beauty and speed of the TI99/4A is brought out by the fortran program. There are of course limitations such as: having to write a lot of subroutines because the compiler cannot handle large programs; requiring a deep pocket to stock up with printer paper; having sufficient interest to find out how the IBM mathematicians brains work, etc. It is perhaps worth mentioning that assembly language subroutines may be called on as well as Fortran subroutines, thus widening the applications field considerably." Thanks George..

.....

EXTENDED BASIC PROGRAMMING TIP: .

Courtesy John Seager:.

Using *ACCEPT AT* for a numeric variable, how do you idiot proof it so the program will not bomb? You can insert a default input value, and use a negative size, and also use *VALIDATE*. .

However, the user CAN blank the default variable with *CLEAR*, and if the input variable is a numeric variable, trying to input a blank will cause an error condtion. .

I have frequently seen authors input all numbers into a string variable. Inputting a blank does not cause an error, and you can test for a nul input and go back if required..

You dont have to do it that way. John points out that using *ON WARNING NEXT* will test for the nul input and go back for you. Try it..

```
eg. 100 ON WARNING NEXT.
    110 ACCEPT AT(4,5)ERASE ALL VALIDATE(DIGIT):A.
    120 GOTO 110.
```

John also mentions that if you program:.

```
100 ON ERROR 600.
110 RUN "DSK1.NOFILE".
```

```
.....
600 ON ERROR 600 :: RETURN.
```

the RETURN will fail as the failed RUN seems to remove the internal pointers. .

This is a deliberate ploy by TI to avoid the "accidental" removal of the List Protection flag- which happens with Version 100 of ExBas..

You need to use the format *RETURN XXXX* where *XXXX* is a line number to go to which will RUN your original program again- you could use *RUN 110* or something if required. The second RUN will of course reset all required pointers.

TI Cassette software was seen on retail sale in DERBY on February 27th 1988- Virgin Software cassettes at 50p each. There could be other odds and sods around.....

```
-----
|
| We regret the sad and sudden loss of      |.
| Ian Martin, who died in February 1988,  |.
| following a car accident.                 |.
| Ian was the proprietor of TIMELESS SOFTWARE |.
| and was responsible for the UK retail sale |.
| of DIABLO and other classic XB games.    |.
| Ian was an avid adventurer and spent much |.
| time with the Infocom series.             |.
|
|-----
```

.....

"BASIC-COMPILER" from Ryte Data..

No, Ryte Data have not sent me a copy, but your faithful reporter has tracked a copy down in order to report to you on its value..

Firstly, whatever it does, it is NOT a compiler, and anyone selling the program in this country would be a sure loser. My dictionary defines a compiler as a program which produces machine code. .

Here is a quote from the documentation: " Principally the Compiler works in a manner similar [not my spelling error!] to the Basic-Interpreter that resides in the Console Rom's".

There is a little difference between a compilation and interpretation!.

If you look back to TI*MES No 15, you will see some bench mark programs- not the final measure of a language but at least a rough guide..

TRIGLOG- using floating point arithmetic:.

TI Extended Basic= 362 seconds..

Ryte Data "compilation" = 600 seconds!!!.

TEXTSCREEN- using floating point arithmetic in TI ExBas but using Integer arithmetic for the compilation (supposed to be faster):.

TI Extended Basic = 117 seconds.

Ryte Data "compilation" = 168 seconds!.

(Remember, Myarc XB = 73 sec!).

Quote "After the Compiler creates the new version of your program, the Loader can then run the new, faster version." Faster???

Quote: "With large programs it can be quite a time saver to use the Basic-Compiler". Possibly..

However, the program to be "compiled" must be somewhat shorter than usual! A program which starts as 19 sectors can become 35 sectors after "compilation" while a program starting at 24 sectors can become 37 sectors. It is not possible to indicate a maximum size the "compiler" can handle, you must use trial and error!.

The error message is "PROGRAM TO BIG" (and no that isn't my spelling error). And this error message is not documented either.

You can of course chain programs together using "RUN DSK1.PROGRAM2"- and you know how long loading can take when you do that....

Faster on long programs? I can't say, as I could not find any programs of 20 odd sectors which would run successfully after "compilation", even after deleting the things the documentation says you can't do:.

NO: DEFs, SUBs, ON ERRORs, LOAD, LINK, ON BREAK..

NOT DOCUMENTED, but you also cannot use ON WARNING..

Maximum string length is declared and subject to an absolute maximum of 64 characters..

I could not find any reason why the programs which failed did so, they were perfectly in accordance with the documentation!.

I did notice that the cursor was not flashing as quickly as it usually does.

The title screen is (c)1984. It is now 1987... and NO major reviews? No reviews full stop... Guess nobody has made it work???

Suppose it did speed things up... it would not help the majority of games programs, as they are dependent on internal timing- it is no good if the alien destroys you before you see it! So rewriting would be necessary to slow some things down!

What does it do? It tags onto the memory image a table of variable addresses and jump addresses. That is all. Thus jumps and references to variables ONLY are liable to be speeded up, possibly. I cannot confirm any such theoretical speed up. In order to function there must also be an interrupt driven routine to bypass the normal system. The slower flashing of the cursor may indicate that the interrupt is taking too long..

Ryte Data this time:"Otherwise (apart from the commands listed above) the Basic Compiler fully supports all Basic and Extended Basic statements AND it will compile multi-statement lines in Extended Basic. You do not need to rewrite your programs or use single statement lines". Something wrong somewhere!.

This leaves us with a supplier subject to several recent complaints supplying a program carrying a false and misleading title of little use.... .

Ryte Data is also responsible for a "TI99/4A Compatible" scam, making moves which might make you think they were intending to produce a TI compatible, no they were just doing some market research to see if the Myarc Geneve was worth supporting...

My recommendation: Do not buy the so-called compiler, and do not send any money to Ryte Data..

You may order with confidence from Tenex, Texaments, Not Polyoptics, SSI, Jim Peterson. .

The Fortran compiler mentioned in the last issue has been reported on by a couple of US users, who seem to feel that it is a good product for the money, and make the point that it is a SUBSET of ANSI 1977 FORTRAN, and not Fortran IV as advertised. .

There are no character variables, which were only introduced in the '77 version, but character data can be handled using integer variables and arrays.. Some of the supported features are integer, floating point, double precision, if/then/else/elseif/endif, dowhile/enddo, as well as just IF and DO! Full multi dimensional array handling, extensions to support screen handling and TI CALLs..

The editor is reported to be about as poor as some main frame Fortan editors, but you can use TI Writer, the compiler detects errors OK but it not so hot on helping you find and correct them- again some main frame compilers are just the same! Preliminary reports are good. Any UK Fortran users out there to give us a full user report?.

Supplier is Tenex and the price is US\$50..

MACHINE CODE THROUGH EXTENDED BASIC....

Do you program machine code links for ExBas?.

Ever suffer odd prangs- screen image table or pattern descriptor table or color table suddenly going whacko while everything else carries on? Two hints:.

From anon, reprinted in April 1987 MSP 99 Newsletter:.

"WARNING!!! Handling your own DSR PABs and buffer areas appears to be VERY dangrous in the XB environment. You can very easily destroy data in VDP memory currently in use by the XB interpreter. One way of handling this is to OPEN the file in XB and use the value at >833c to find the PAB and buffer area allocated by the file (see page 302 of the E/A manual). OR, save VDP memory you are going to use to RAM and then restore it to the original value after the DSRLNK".

"Also, be advised that the Get String Space routine provided via the GPLLNK appears not to work in the XB environment. EVENTUALLY (usually after the first garbage collection is forced) VDP memory in use by the XB interpreter will get corrupted and.... results may be unpredictable!".

CONSOLE PROBLEMS?. Check the voltages.

Need a new module connector? Console lock ups? Color gone west?.

Reports from Southwest 99ers, Tucson, through MSP 99 Newsletter are that some ageing power supplies (the hot ones in the console!) are allowing their output voltage to drift upwards (somebody can tell me which bit is going- capacitor or resistor or diode or something!) such that measured output voltage on the nominal 5V DC line has been up to 5.4V. Reducing that voltage back to the preferred 5.1V DC appears to solve an awful lot of ageing problems..

If anyone has problems, and fixes them with a revitalised (or new) internal power supply please write and tell me, as this looks an interesting area for maintaining console life! .

If your internal power supply PCB has a code on it 1053214-2, the 12v line may be unstable, jumping from 9.5 to 11.5 volts! The PCBs marked 1053201 seem to have good 12v lines..

We have a member with a MYARC DISK CONTROLLER CARD looking for fixes to the various programs which dont like it. MGs EXPLORER has a special version for the Myarc disk controller, but Advanced Diagnostics (and several freeware programs) wont work. Does anyone know WHY and is there a simple fix or is it hopeless? PLEASE WRITE!.

=====.
INITIATIVE....

Library disk H3+H4 contains a machine code program that copies the data base for the Adventure module from cassette to disk. It seems everyone who sends for it writes to tell me I have missed the copy program off... I haven't, and the program and document files are clearly called COPY! It just is not possible to send out paper documentation on public domain/ freeware items, and quite often documentation and instructions are minimal (sometimes they are overwhelming!).

Look for files in DV80 format- they are either machine code source code- which may include instructions- or they are straight forward instructions. Files in PROGRAM format MAY be in Basic or Extended Basic or Machine Code or merely comprise a memory image of data. Funlweb will tell you if they are machine code or Basic/XB..

IV254 files may be long Extended Basic programs..

DF80 files are usually machine code to be loaded with LOAD AND RUN..

PROGRAM files of 33 sectors, with several files differing only by the last letter, are machine code memory image files- you load only the first file, using RUN PROGRAM FILE or similar.

The Adventure modules use PROGRAM type files but they are neither Basic nor Machine code. They are a data base in memory image format, used by the program in the module.

And if you are interested in adventures on disk for your Adventure module why not send for the Disk Library Catalogue- send two disks and return postage!!!.

TURBO PASC 99 PROGRAM: .

.
Very very few of our readers will own TP99 right now- but for those of you with disk drives, here is a lengthy sample of a program written in TP99, to enable you to see how it appears, and for those of you with TI Pascal, you can see how the syntax varies. Professional Pascal users may be interested to see how arrays are passed to procedures!.

.
This is the first ever Pascal program written by me, and is based on a TI Basic program in the book "Dynamic Games for your TI99/4A" by Scott Vincent. I have wanted a flexible LIFE simulation in machine code, but not found any suitable, so I have taken advantage of Pascal to see what I can do!.

.
In this version, you first set up the LIFE starting position by keying A for life and SPACE for no life, to fill a 12x12 grid. I have not put any cursor control in- it just moves over each line in turn, left to right. .
The outer frame has fixed status- it will affect the inner 10x10 grid, but no changes will occur in the frame: if you put life there it will stay there! This enables you to experiment with variations on the normal isolated life universe. To help spot the frame, the cursor is a + in the frame and a ? in the 10x10 grid..

.
You have the option of allowing automatic changes in generations, or keying them. If you select automatic, you may return to the set up section just by pressing a key. If you select keyed progress, to get back to the set up screen involves quickly pressing two different keys, one after the other..

.
There is one "bug" left in- the counter G is of integer type but there is no test for maxint, so after some 32000 odd generations the program will bomb out!.

.
LIFE patterns may:.

Die out. .

Reach a stable pattern..

Cycle between patterns- changing between two patterns is the most common, but I have created a 4-pattern cycle..

As a variant of the above, a pattern has been found in which a "bullet" is regularly produced from a cluster, and moves off screen..

.
Here is the Pascal code, entire, and as successfully compiled....

```
{ PASCAL CODE =====}
```

```
PROGRAM life;
```

```
VAR k,z:INTEGER;
```

```
    m,n:ARRAY[12,12] OF INTEGER;
```

```
PROCEDURE rules;
```

```
VAR k,status:INTEGER;
```

```
BEGIN
```

```
cls;
```

```
writeln(" LIFE");
```

```
writeln(" first set up start position");
```

```
writeln(" Put life forms into a");
```

```
writeln(" 10 x 10 grid ");
```

```
writeln(" by pressing A or space");
```

```
writeln(" (The 10x10 grid is in a frame-");
```

```
writeln(" making a 12x12 grid in total:");
```

```
writeln(" If you place life in the frame");
```

```
writeln(" it will never die, and can ");
```

```
writeln(" constantly create new life if");
```

```
writeln(" the proper life conditions are met");
```

```
writeln(" ");
```

```
writeln(" There is no cursor control.");
```

```
writeln(" Generation starts after 144th");
```

```
writeln(" key press");
```

```
writeln(" When prompted indicate if you");
```

```
writeln(" want to prompt each generation");
```

```
writeln(" with a Keypress, or let the");
```

```
writeln(" program go full tilt.");
```



```

writeln(" ");
writeln("    Press any key to continue");
writeln("    Stephen Shaw Jan 1988");
REPEAT
    key(3,k,status)
UNTIL status=1;
END;

PROCEDURE draw(VAR m,n:ARRAY[12,12] OF INTEGER);
VAR row,col:INTEGER;
BEGIN
    FOR row:=1 TO 12 DO
        BEGIN
            FOR col:=1 TO 12 DO
                BEGIN
                    m[row,col]:=n[row,col];
                    cursor(row+2,col+5);
                    IF m[row,col]=1 THEN
                        write("0")
                    ELSE
                        write("~");
                END;
            END;
        END;
    END;

PROCEDURE think(VAR m,n:ARRAY[12,12] OF INTEGER);
LABEL rerun;
VAR c,g,k,s,t,y,z:INTEGER;
BEGIN
    cls;
    cursor(2,2);
    writeln("  Press K to key generations");
    writeln("  Or any other key for auto");
    writeln(" ");
    writeln("    When all life ceases,");
    write("    Press a key to restart");
    REPEAT
        key(3,y,c)
    UNTIL c=1;

    { y = 75 TO key prompt }
    cls;
    g:=0;
    REPEAT
        g:=g+1;
        FOR k:=2 TO 11 DO
            BEGIN
                FOR z:=2 TO 11 DO
                    BEGIN
                        c:=m[k-1,z-1]+m[k-1,z]+m[k-1,z+1];
                        c:=c+m[k,z-1]+m[k,z+1];
                        c:=c+m[k+1,z-1]+m[k+1,z]+m[k+1,z+1];
                        IF (m[k,z]=0) AND (c=3) THEN
                            n[k,z]:=1;
                        IF (m[k,z]=1) AND ((c=2) OR (c=3)) THEN
                            n[k,z]:=1;
                        IF (m[k,z]=1) AND ((c>3) OR (c=0)) THEN

```

```

        n[k,z]:=0;
        key(3,t,s);
        IF s=1 THEN
            GOTO rerun;
        END;
    END;
    cursor(24,2);
    write("generation ",g);
    draw(m,n);
    IF y=75 THEN
        REPEAT
            key(3,k,s)
            UNTIL s=1;
        UNTIL false;
    rerun:
    END;
PROCEDURE setup(VAR k,z:INTEGER;VAR m,n:ARRAY[12,12] OF INTEGER);
VAR j,jj:INTEGER;
BEGIN
    REPEAT
        REPEAT
            cursor(k+2,z+5);
            IF (k=1) OR (k=12) OR (z=1) OR (z=12) THEN
                write ("+")
            ELSE
                write("?");
                REPEAT
                    key(3,j,jj);
                    UNTIL ((j=32) OR (j=65)) AND (jj=1);

                IF j=65 THEN
                    BEGIN
                        cursor(k+2,z+5);
                        write("0");
                        m[k,z]:=1;
                        n[k,z]:=1;
                    END
                ELSE
                    BEGIN
                        cursor(k+2,z+5);
                        write("~");
                        m[k,z]:=0;
                        n[k,z]:=0;
                    END;

                z:=z+1;
                UNTIL z=13;
            z:=1;
            k:=k+1;
            UNTIL k=13;

        END;
    { This is the actual program!:- }
    BEGIN

```

```

cls;
rules;
REPEAT
  k:=1;
  z:=1;
  cls;
  text;
  setup(k,z,m,n);
  think(m,n);
UNTIL false;

END.

```

=====

It takes about a second for each generation- much faster than is possible in TI Basic! It would be simple to amend the size of the life universe, but the larger the universe the longer the creation time!

=====

.....**REVIEWS:**..

EPYX 500XJ JOYSTICK by Konix :

(Sold in the U K for 12 pounds as the KONIX SPEEDKING).

Purchased from Tenex for US\$15.95. They also sell the Wico BOSS Joystick for US\$16.95- the Boss was on sale in Manchester at Christmas time for just .11.4.95! The Epyx 500XJ joystick is made in the UK! and it requires an Atari to TI adaptor (\$4.95 from Tenex)..

(My wife is becoming concerned at my collection of joysticks, but each has its plus features!). Most important note: The Epyx 500XJ joystick is contoured to be held in the hand- but ONLY the left hand. It cannot be used in the right hand. The unique shaping does make it quite easy to hold- I have fairly small hands (7 1\2 for the technically minded!) and suspect a child would not find it so easy..

The joystick operates true microswitches, not the common leaf contacts in so many other joysticks. The action is positive with good loud clicks! Only a small amount of movement is required, and diagonals are fairly easy to get. The trigger button is part of the special contouring and is operated by the middle finger of the left hand (if your hands are larger you could possibly use the first finger!). Trigger action is positive and quite rapid firing is possible, although I found my finger tiring rapidly when faced with long spells of rapid fire! (It is preferable if the program allows you to just hold the fire button down rather than requires separate pushes!)..

In general, a well made and comfortable joystick, taking priority over my Prostick II joystick, which is relatively uncomfortable to hold- although the Prostick can be held in either hand..

.....
Book: COMPUTE!'S BEGINNERS GUIDE TO ASSEMBLY LANGUAGE ON THE TI99/4A by Peter Lottrup. US\$13.95 from Tenex. 262 pages, paperback. Copyright date is 1985!.

Another assembly language book based on the Mini Memory module plus line by line assembler- but also applicable to the main Editor Assembler package. Helpful detail on actually USING the LBLA.- the instructions TI supplied were inadequate to say the least!

Apart from being of inestimable value to anyone without expansion, but with a mini memory module, wanting to write short machine code routines with it, this book is also of great value to anyone learning TI machine code who is using

some other assembler, such as the Editor Assembler. Each instruction is very carefully explained and illustrated- except that again there is no explanation of XOP. Does ANYONE have a book or magazine article explaining XOP please!.

Addressing modes are described briefly but usefully. The sections on using KSCAN (and Joysticks!) and Sprites are very useful. Moving variables between Basic and Machine Code is explained- but NOT passing Arrays. Sounds and Graphic Modes are explained. .

One serious black mark- throughout the book, direct addresses are used to Mini Memory routines. eg BLWP @>6018. No Ed/As equivalents are given in the text, and to save a page, there is no appendix. You will NEED to have the mini memory manual if you wish to understand this book if using Ed/As. And no mention is made of the different environment of Extended Basic- which is also very badly described in the EdAs manual..

Summary: An EXCELLENT book for the mini memory owner, and a good book if you are learning machine code with Editor Assembler: provided you also have the Mini Memory manual!.

....

A SHORTCUT THROUGH ADVENTURELAND VOL 2: INFOCOM...ISBN 0-88190-358-2. (c)1984.

Paperback, 189 pages, by Lynn,Ashley and Sloan, published by DATAMOST.

Covers the first ten Infocom adventures, from Zork 1 to Infidel. For each adventure there are several sections. First there is a general introduction, which tells you what you are supposed to be doing! including where applicable a list of treasures you need to collect. Then there are general hints, including odd restrictions on what you can do where, which will save you wasting time trying to do impossible things! Also some important problems are solved-eg what do you do to get past the troll in Zork 1, and how do you kill a Cyclops (you can't!). .

Next comes ROOM NOTES, a detailed LIST of the rooms, their names and notes on what they are there for, including sometimes detailed hints on problem solving in a particular room. Next item is a detailed HOW TO WIN section- essentially a list of what you must do to win. And finally a room plan..

You may notice when loading Infocom games a REVISION number- my Zork I is revision 75! - as a result of changes to the Infocom adventures, a detailed book such as this one does NOT enable you to zoom through an adventure, as inevitably there have been changes to the detail! Those you must sort out for yourself! This book will give you a good start though and MAY help you with a really difficult problem..

Reading this book convinced me of two things- 1. NO WAY could I possibly complete ANY Infocom adventure. 2- I would not like to be left alone with anyone who had alone and unaided completed an Infocom adventure! The book does however allow you to get the set task out of the way, enjoy the text, and then have a little harmless fun generally exploring and trying to do silly things (though how you could be sillier than some of the intended solutions I do not know!). .

My copy was second hand, and I dont know of a source, but watch out for it!.

NOTE: Infocom are no longer producing their adventures for the TI, and have run out of stocks of some titles.

....

A CRASH COURSE IN PASCAL. by Donald M Monro. Published by EDWARD ARNOLD. 1985..

ISBN 0 7131 3553 0. .7.50 or so. Paperback, 202 pages..

This PASCAL textbook is ideal for anyone trying out the Turbo Pasc 99 package from Texaments. With so many Pascal books now dealing with TurboPasc (the Borland language not the TI one), it is nice to find this book dealing exclusively with ISO PASCAL, the international minimum standard for Pascal. .

Commercial versions of Pascal naturally differ from the ISO PASCAL, otherwise no-one would buy them! Typically, features are added, and some things are expanded or made easier. This is true of TI TurboPasc99, but you will find

yourself able to use at least half the examples in this excellent book..

I found the text very readable, and with TurboPasc99 loaded and working, easy to sail through. Adjusting for the variations in TP99 was no problem- but I did miss not having TYPE available!.

TP99 does not have packed arrays- it has strings instead. Although type is not supported you CAN pass arrays to procedures. In the absence of type, records and files are not available (as types), nor are pointers. Despite these departures from ISO Pascal, at least one half of this inexpensive book on ISO Pascal is directly of use to TP99 users, and it almost certainly represents the best value reference book you can buy to use with TP99..

USCD Pascal is an extension of ISO Pascal, and TI Pascal owners may also find this book of interest. As it is a UK publication, the price is very reasonable and its value is great!.

Pascal was developed (in Switzerland) as a language for TEACHING (rather like Logo!) but was then taken up as a serious language in its own right. Because of its origin it is quite a good language to learn after Basic! Once you have an initial grounding in Pascal, you should not find C so hard! And a grounding in Pascal and C will take you very far these days! A formal (eg incomprehensible) description of ISO Pascal can be found by buying a copy of British Standard BS 6192:1982. The BRITISH standard is the definitive one for English versions of Pascal, and derived from it are ANSI/IEEE 770 X3.97-1983 and ISO 7185-1982..

....

FONT WRITER 2 by J P Hoddie, published by Asgard and widely available from the USA for about US\$25..

Font Writer has already been reviewed. Version 2 goes a long way beyond that! In essence we have a replacement formatter program for TI Writer files, with additional "dot" commands to let us incorporate pictures into the text, and to change fonts. Speed is... SLOW! .

Font Writer is not as powerful as The Printer Apprentice, but it is faster and easier to use- eg the products complement rather than compete..

Font Writer 2 can work with most TI ARTIST or CSGD fonts. Graphics can be either TI Artist Pictures (_P) or Instances (_I) or GRAPHX pictures..

Graphics may be inverted and/or framed. Pictures can have a left margin from 0 to 28, while instances can have their own margins. Instances and text can be on the same line- eg text can have ONE instance on the same line- not quite as flexible as TPA! Print density and width- affecting all output- are selected with dot commands. Fonts are also selected by dot commands- allowing each line of text to be a different font (but not mixed fonts on one line). Text can be inverted (eg white on black- an excellent way to get through your ribbon!)..

Most Useful Utility: DISK DUMP. If you have a disk with a mixture of CSGD Fonts, TI Artist Fonts, TI Artist Instances and Pictures, Graphx Pictures- this utility will read the disk and prepare a file to run through the Formatter which will print out all the disk contents! Fonts are printed with as many letters as will fit on a line! File names are given with each item. One problem here: using fanfold paper you may find pictures split by the form feed! Edit the text file to avoid this if you can..

Other Utilities: BANNER which allows you to create huge banners using your font collection- NB: ONLY uses CSGD fonts in /CH and /GR files. However... there is also a utility to transfer TI Artist fonts into CSGD fonts included!.

The banner may have up to 252 characters. (/GR images are not magnified at all)..

Also included is a handy FONT MANAGER to merge fonts. Fonts are stored in memory so some very large fonts may not fit! The FONT MANAGER can convert between _F files and /CH files..

Also we have conversion routines for changing from /GR to _I files, from /DT files to _I files, from _I to /GR files, and from _I to /DT files (eg between TI Artist and CSGD)..

There is a FILE MANAGER which is a sort of disk manager- it lets you catalog, delete files, rename files, change file protection, and copy DV80 files..

There is a FONT EDITOR which allows you create your own fonts or modify existing fonts- you may call up and amend individual characters if you are not happy with them! Then use Font Manager to merge the amended character back into the original font file. This is quite a handy font editor to use..

There is also a macro facility for use in the font editor, but it strikes me as being a bit fiddly. All in all, a useful package of programs here, well worth the asking price..

....

THE PRINTER APPRENTICE TOOLBOX by McCann Software. Widely available in the USA at around US\$22.50.

This "companion" to THE PRINTERS APPRENTICE is only JUST a companion, as it can do a great deal of work on its own. You only need the main package to prepare a SCHEDULE for the formatting of a mixed page of different text/images. This package can, on its own, do the following- items beginning with an asterisk * are really for use with the main package:.

*See on screen the page make up of a formatted page- as our screens are not dense enough for WYSIWYG, we have boxes in place of actual text/graphics. I found this item unhelpful and misleading and recommend it is avoided! Go by the numbers provided by the SIZE command and formatter schedules..

FORMAT text- includes a mini editor or can use DF80 files produced with TI Writer. Uses special TPA fonts! (see later!). You can now expand fonts without limit other than printer capability. And expanded fonts do not have to be solid blocks- you may design pixel shapes (nice). Letters can be distorted by magnifying width and height differently. Most useful..

CONVERT TI ARTIST and CSGD fonts to TPA SDSH fonts. Also you can bring in TI ARTIST instances or pictures for use with the TPA Scheduler. Instances and Pictures can be cropped to remove unwanted bits..

*Border Builder creates a border chosen from many designs (design your own!) to put around text or graphics, and the border is placed around the output using the Scheduler from the main package. The Toolbox can however just print out a frame for you..

Create GRAPH PAPER! by placing BOXES of small dimension close to each other!. All in all a good package. Essential if you have TPA. If you buy the Toolbox, the main TPA will help you get the most out of it but is not essential..

.....

.

MORE REVIEWS!!!.

ART-CONVERT...Trio+ Software, about US\$10 from Tenex and other US suppliers..

This utility will transfer your TI Artist instances into a disk format which you can print out using TI WRITER Formatter- for instance, you can develop a fancy letter heading to start your letters off!.

I was especially interested in this program as I wrote one very similar about a year ago- based upon programs in The Smart Programmer and Super 99 Monthly. My program, ART/FORM, is on Utility 7, in the disk library, and allows you to invert the image if you wish, to choose from three print densities, and to mark the left tab. ART-CONVERT lacks the image-inversion but adds the ability to use an 8-bit high TI Artist font instead of your usual printer font. Maybe I'm biased here! Suffice to say you can get UTIL 7 which has lots of other programs on it as well for a lot less than the cost of ART-CONVERT!.

NB: When printing graphics with the Formatter, the printer name must end with .CR, and if you USUALLY use a printer name ending with .LF, then instead you must add a line feed at the end of every line of text you want printed (you can do this automatically by formatting the text to disk instead of printer as a first pass)..

....

SINGS DATA 1 from TRIO+ Software, includes SINGS-PLAY. US\$10 from TRIO+ and other US Sources..

This program follows on from the FREWARE program TI SINGS, available from the disk library- DATA 1 has NOT been released as freeware! SINGS DATA 1 contains a number of "rock songs" for your TI plus Speech Synth plus TE2 module to sing..

The style is of course a trifle odd, but how many other computers do you know that can sing I HEARD IT THROUGH THE GRAPEVINE or THE HOUSE OF THE RISING SUN! The program SINGS-PLAY enables you to select which songs on the disk you wish to hear, and to select the order of play. Having done this, the songs will play in sequence, rather like a juke-box! SINGS-PLAY will work with any song file produced using the TI SINGS creation program..

If you have TE2 and speech synth, the cost is low. Why not try it!.

....

LEGENDS. by Donn Granros and Ed Johnson. Released through Asgard Software. \$25 from Tenex..

How to describe this program? Hmmm. You need a long time... fortunately there is a save option! Most important of all: NEVER use your master disk as a play disk. ALWAYS use backups! The program amends the disk as it goes! Take a party of adventurers out to explore this stricken planet, and if someone is killed, go back to recruit more! This is a graphic adventure on a large scale, with monsters and dungeons. .

This is NOT an Infocom type adventure. There is an element of strategy involved, but a great deal of luck. Perseverance and patience are needed. By fighting many monsters you build up your experience and wealth to the point where you can take on the really evil ones! The closest similar program I can think of is SECRET OF KZIRGLA by Kuhl Software, now revamped and improved as THE DUNGEONS OF REMZAK. But LEGENDS is one heck of a program- immense! With an awful lot to explore and do and a great deal of fighting too. If you have a lot of spare hours, and would enjoy lots of random adventuring, you may like this one. I dont think sheer size is a guide to quality, and the advertising is WELL over the top- I repeat, you cannot equate this program with an Infocom adventure (and be sane and honest that is!). It is quite different. If I find myself with many empty hours I may play it...

The price (for two disks) is not bad considering the many many hours that must have gone into preparing this monster, but I wonder about the ability of the program to hold and maintain your interest....

...

UpDate: **SPAD XIII by Not Polyoptics**, available widely in the USA:.

Version 2 adds a RED BARON option- this is a more intelligent enemy. PLUS the ability to quickly zoom to choice locations- quickly teleport to the German airfield, bomb the hangar, and teleport back home before you get hit by flack! (Cheating isn't it!). You can now see your plane from outside- viewed from the South and slightly above. And at game end you have an option to restart instead of the program just terminating. Plenty of improvements. It is an interesting experience to taxi up to a tree and stop with the tree growing out of your plane!.

....

OLDIE: An old XB game has been mentioned in print several times recently.

History first... **DIABLO**, by Manual Constantinidis, was released in TI Extended Basic by the US firm of EXTENDED SOFTWARE in 1983, when it was advertised for US\$19.95, and stated to be "1st place winner of the TISHUG Australia Contest". Yes, the author was a member of the Sydney user group..

DIABLO involved the movement of tiles containing various shapes of track, in order to keep a moving ball continuously supplied with track. As it moved over a tile, the track it moved over disappeared, so you were left with less and less track! The object was to clear the screen, so that the ball had moved over

every piece. Diabolic! It was sold in the UK by Timeless Software.

DIABLO is NOW on sale again..."Diamond Games, a German software company, marketed in the UK under the Robtek banner. By Manuel Constantinidis, updated to live within the half-a-megabyte machines, with a Commodore version to follow". DIABLO for the Amiga or ST is advertised at .19.95 and .14.95 respectively. The C64 version will be .7.95 on tape or .9.95 on disk. These new versions appear to be exactly like the TI original, apart from a slightly different tile layout, and the ability to pause/save/load..

When a program written FOR the TI99/4A by a user group member makes it onto the Amiga, it is a time for celebration- and pause to consider just how good our old programs really are!.

...

HIGH GRAVITY: .

Way way back... December 1982... I enjoyed a game on a Hewlett Packard computer in which you aim a rocket in a chosen direction, with the intention of it reaching a certain point. Its course is affected by the presence of various planets, which pull it off course by gravity....

Well, I can now play this game on my TI! HIGH GRAVITY is a good disk full from Asgard Software (recommend you buy from Tenex or some other reliable source!). You may select how many planets you want on screen, and then solve the "random" screens thrown at you. OR you can set up your own screens- and save them for future solution! The program comes with many such screen files! You may experiment with gravity- and as the rocket leaves a trail behind it, perhaps even use gravity to produce an interesting art form!.

After waiting so long I can only recommend the program to you. It is more of a program to experiment with than a game, and you may not play it often after the first week, but it really is an excellent program!.

...

HIDDEN POWERS OF DISK FIXER (book) \$8.95 from Tenex..

Written by Bill Gronos, and intended as the manual for the Navarone Module "Disk Fixer", this book is also available separately, and is of value for use with any disk sector access program (there are several in the User Library). .

This book has some interesting tricks to perform with sector access- like making a Basic program unlistable and unresquenceable. How to remove or deal with various types of protection (at sector access level- does not deal with initialisation protection devices). And most important of all, how to recover information from a disk which suddenly becomes uncooperative, just before you make that back up copy you've been meaning to do for the last five years....

A mere 52 pages, with six forming the Disk Fixer Manual, this little book has some interesting information..

It is not however an official manual- the information it contains is gleaned- and may not be totally correct or complete. In particular, the method for putting program lines into line number order is unduly tedious- instead of editing every single line in the program, you can use your console to do the work for you- save in merge format and then type new and merge back in. All program lines now sit in memory in line number order.

Interesting note- although I have seen the book advertised for many years, it contains a copyright date of 1987. It may be that only the format is copyright- placing a copyright date which comes after original publication implies the text is not so protected..

A useful little book to have if you ever need it - and one day you just may!.

.....

=====

Submitted July 1988: RAMBLES..
Welcome to the July 1988 issue of RAMBLES by Stephen Shaw.
10 Alstone Road, STOCKPORT, Cheshire, UK, SK4 5AH.

CORRECT TO N PLACES.....

Rounding is easy- just use:.

RESULT=INT(NUMBER+0.5).

But if you want to display a number correct to a certain number of digits? A little more difficult but not impossible..

The GENERAL format is:.

RESULT=INT(NUMBER * P +0.5)/P.

Where P=the power of ten of the number of places to be rounded..

Thats easy isn't it? To round to two places, P= 100 (10^2)..

The general format will only work for positive numbers. For numbers which may be either negative or positive, the formula becomes:.

RESULT=INT(ABS(NUMBER)*P+0.5)*SGN(NUMBER).

Here us a tiny utility sub program for you- for TI BASIC just remove the first and last lines and GOSUB instead....

```
100 SUB PLACES(NUMBER,PLACES,RESULT).
110 P=EXP(PLACES*LOG(10)).
120 RESULT=INT(ABS(NUMBER)*P+0.5)/P*SGN(NUMBER).
130 SUBEND.
```

Now to find out what 23456 rounded to 3 places is you would use:.

```
1 CALL PLACES(23456,3,RESULT).
2 PRINT "Result:":RESULT
3 STOP
```

MAURICE E T SWINNEN of Mid Atlantic 99ers. 1986.

=====
MACHINE CODE FROM TAPE:

Did you know that CALL INIT:: CALL LOAD("CS1") works! Of course first you have to transfer those DF80 files from disk to cassette. And some of the lines may need shortening (you do remember that cassette data files don't use 80 byte records!)- dropping off the checksum is a good start there!.

eg from ..9BD3A87F131F 0001.
 ..5A47F196F 0002.
 ..A127F1BFF 0003.

shorten to:.

 ..9BD3A8F 0001.
 ..5A4F 0002
 ..A12F 0003

-eg remove the 7 tag and the next four characters. Leave the final F.

Read each record from disk and transfer the first 64 characters to tape- you will need to note how many lines are to be transferred or use the EOF function. The tape file is saved as DISPLAY, FIXED 64. You will also need to know the starting name to LINK to..

Then of course, there is the TIME element. Take a look at how many lines your average DF80 file has... a 54 line file, loaded from tape, takes 6 minutes. Now work out how long it is going to take to transfer SUPER DEMON ATTACK this way... and WHERE are you going to find a cassette THAT long!.

Just one more capability the TI99/4A has that has not been documented. .

Now we have the FUNLWEB LOAD program, tape owners can find it much easier to load machine code into their 32k rams using memory image files. Much faster..

Editor Assembler module owners can just use the LOAD PROGRAM FILE option and indicate CS1. - it really was clever of TI - the ONLY loader that TI supplied to enable cassette owners to load machine code into expansion ram, and it is sold on a high priced package- module, huge manual, and two disks.

Thanks to George Meldrum, TISHUG. May 1988..

=====.

ON BREAK LOAD DSK1.LOAD

SOURCE CODE. Author and original publication not known..

Assemble this little utility into a non-compressed DF80 file and you can load it into XB with a CALL LOAD. Then it will auto-boot "DSK1.LOAD" whenever a running XB program breaks for ANY reason - a little more powerful than ON BREAK NEXT. Try it!

```
DEF CHECK.
CHECK  MOVB @>8344,@>8344 * >8344 IS 0 IF XB NOT IN RUN MODE.
      JEQ  NORUN .
      B    *R11          * EVERYTHING OK SO RETURN ELSE:.
.
NORUN  CLR  @>83C4        * TURN THIS ROUTINE OFF (ISR HOOK).
      LI  R1,>6372      * XB GROM START.
                        * MAY NOT WORK WITH SOME XB VERSIONS.
      MOVB R1,@>9C02    * WRITES >63 TO GROM WRITE ADDRESS REGISTER.
      SWPB R1 .
      MOVB R1,@>9C02    * WRITE >72 TO GROM WRITE ADDRESS REGISTER.
      B    @>006A      * EXECUTE XB.
      AORG >83C4.
      DATA CHECK      * PLACE THIS ROUTINES ADDRESS AT >83C4.
      END.
```

=====.

"We do not know what is going on in England. we know the Queen is still kicking and alive (God save Her!), but we are afraid the TI-99/4A is dead there." Huh?. Written by Paolo Bagnaresi, Milan, Italy, in August 1986, to Ottawa, Canada. Guess our reputation hasn't got to Italy just yet then!.

=====

INFOCOM adventures are not all FULLY logical- there is a RANDOM element in some of them which means that sometimes you die and sometimes you live... and an interesting command to type in to your INFOCOM adventures is \$VE try it! . There is a #RAND command in Lurking Horror, which expects a number before/after it, not sure what it does but I think it may determine the path when you come to a random choice. HITCHHIKER has a total vocabulary of 969 words- have you found them all yet? KILL ADAMS? Some odd commands, purpose unknown include YXXY and ZZMGCK - the latter may just be an end of file dummy. SUSPENDED has a vocabulary of 680 words, but you can complete it with just 35, that's real overkill!.

=====.

MICROPENDIUM:

If you don't yet subscribe to MICROpendium, why not? The cost is reasonable- pries advertised in March 88 copy (I get mine by seaimail!) are US\$23.50 per year by sea mail and US\$37 by airmail. Send US\$ International Money Order, bought from Lloyds or Barclays Banks, to:.

MICROpendium, P O Box 1343, Round Rock, TX, USA, 78680..

This is no fly by night magazine. The March 88 issue is their 50th monthly issue.

Regular articles on Basic, c99, Geneve, and plenty of reviews. Back issues are \$2 each by seaimail to subscribers only. .

=====.

DV80 FILES TI WRITER CAN'T READ:

Any time you see a Display Variable 80 file on a disk, it is always a good idea to take a look at it with TI Writer, as there is a good chance it is either documentation or source code—which may contain documentation. In almost every case a DV80 file should load with TI Writer, whatever is in it, text or data or anything..

However, there are a few text files coming out of Europe that you CANNOT load with our version of TI Writer—our European friends are using a different and incompatible version of TI Writer, VERSION 2.0. If their text files are saved to disk with PF there is no problem, but using SF adds tab information..

In order to produce those odd European characters, printers use ASCII codes outside the usual range of Vn 1 of TI Writer, so Vn 2 was modified to accept them, and the tab info had to be modified as well. And as their tab data is outside the capabilities of our version to handle, the result is a console lock up. Curing the problem was difficult, until our membership secretary dealt with it....

In the April 88 issue of EAR99's newsletter, he presented a program to amend the tabs on a Vn 1 TI Writer file—largely to demonstrate the way the tabs are saved. What was interesting was that by appending a new tab set, the original set is "replaced". If it works for version one, why not try appending a version one tab set onto the end of a version two file?.

It works! The following routine is a much modified form of his program. If you cant load any DV80 file, amend it with this trifle and try again!.

```
100 REM MAKE VN 2 TI WRITER .
      FILE ACCEPTIBLE TO VN 1 .
      based on an idea by      .
      Peter Walker. UK. 1988. .
110 DISPLAY AT(2,2)ERASE ALL.
: "INPUT NAME OF FILE TO BE":.
"MODIFIED:" .
120 DISPLAY AT(6,2): "DSK1." .
130 ACCEPT AT(6,5)SIZE(-12):.
FILE$ :: FILE$="DSK"&FILE$ .
140 OPEN #1:FILE$,DISPLAY,VA.
RIABLE 80,APPEND .
150 A$=CHR$(128)&CHR$(134)&C.
HR$(128)&CHR$(212)&RPT$(CHR$.
(213),16)&CHR$(128)&CHR$(136.
) .
160 PRINT #1:A$ .
170 CLOSE #1 :: PRINT "DONE".
180 END .
===== .
```

PR BASE HINTS:

Sorting and so on are based on an ASCII STRING and everything works according to the ASCII values of the characters. Thus while 4 comes after 2, 22 will come before 4. Use leading zeroes on numbers you are sorting by—then you will correctly find the sort as 02,04,22..

Selective Indexing search works on your input UP TO the first space, so that "good day" will only work on "good". To use the whole thing, you must insert a question mark, thus "good?day"..

PLEASE will someone write tutorials for us for PRBASE and CFS!.

=====.

AXIOM PRINTER?.

By this time some owners of AXIOM printer may discover that their printers are not entirely behaving themselves. In particular, your computer may give you error messages indicating the printer isn't there! The problem lies in the thin fiddly wires they used which are PUSH FITTED in their little connectors. In due course the wires will either pull out, or more likely, break near the end, and contact is lost..

The solution is to remove the whole ribbon, remove about a half inch off the end, and then refit- soldering is recommended, but CAREFULLY..

=====.

CASSETTE CARE: .

The Manchester Central Library have published an excellent leaflet on care of cassettes and cassette players, which is highly relevant to Cassette Users. I have extracted the juicy bits for you....

MUSTS: .

1. CLEAN your recorder regularly- if you use a "wet cassette" you must still clean separately the capstan and pinchwheel, as these special cleaning cassettes only clean the heads properly. See notes later!.
2. Keep cassettes in cases, away from heat and magnetic fields and damp..
3. Before putting a cassette in the machine, rotate a spool with your finger to make the tape fully taut..
4. Never use C120's and for best results stick to C60s or shorter. C120s are very likely to stretch and snap, and even C90s may come apart quite quickly. The thinner tapes are more likely to snarl up your machines..
5. Cassettes are not hammers. Observe care..

.

Detail: .

A cassette tape is a VERY thin thing, coated with an even thinner oxide coating, and even under the best conditions, the best tape will shed its coating little by little... onto the surfaces of your recorder, where they stick and begin to scratch your tapes and even more coating comes off.....

The PINCHWHEEL is the larger rubber-like wheel which rotates pulling the tape onto the take up spool, while the CAPSTAN is the thin metal wheel which presses the tape against the pinchwheel. If you dont clean these properly, tape is liable to snarl up in your player. The only way to clean them is with COTTON BUDS, dipped in special head cleaning fluid or meths. Some alcohols may be suitable but NOT isopropyl alcohol, widely used for cleaning disk drives!, as isopropyl alcohol can cause uneven swelling in the pinchwheels.

Clean the heads to avoid undue wear, and losses of audio quality (dullness) which may stop the player being computer compatible..

CLEAN EVERY 10 HOURS PLAYING TIME. .

If you use a head cleaning tape, discard it after ten uses, after that it just spreads dirt around..

Tape heads become magnetised over a period of time- as the tape passes over them, the heads gradually build up a magnetic charge from the tape. This could produce increased background hiss, and might eventually result in tapes becoming partly (and irrevocably!) erased while being played. Demagnetise your heads AT LEAST once a year. Special cassette-shaped devices can be bought and are safest to use. .

(With thanks to Manchester City Council-Cultural Services).

=====.

ATARISOFT BUG:.

In case you just find one, the early Atarisoft modules for the TI99/4A do not all work properly in SOME consoles. Atari downloaded the "large character set" from grom 0, but used a direct address instead of the indexed address, and yep, some consoles are 4 bits out on the direct address. It means those characters look kinda Japanese....

====.

MILTON BRADLEY BUG (and others):.

Some MB games intended to use speech have a bit of a problem when loaded from disk dumps into the 32k ram. The delay in testing for speech is just not long enough (see EdAs manual pages 349/350), so sometimes the games will fail to identify that you have a speech synth, and give no speech. .

====.

TI FORTH DEBUG.

The following corrections have just surfaced from Ottawa- passed on without testing. Change a COPY of your Master Disk!.

Screen 58 line 10 to read:.

VDPMD @ 4 < IF SMTN 80 0 VFILL 300 ' SATR ! ENDIF.

Screens 53,54,54 line 1 should end SETVDP2..

Screen 59 line 9 change 00FF to 00FE.

====.

WHAT'S IN A FILE?.

Disks for the TI can have a number of different file types, which can serve several different purposes. How can you tell which is which? In some cases, you are reduced to trial and error! But the following notes may help:.

PROGRAM FORMAT:.

Unwisely named, as not all PROGRAM files are programs. This type of file is better described as MEMORY IMAGE- just a byte by byte image of a particular area of memory in the computer, which MAY be a program in basic or machine code, or some kind of data- graphics, adventure data and so on..

FUNLWEB will identify basic and most machine code images for you- use SD from TI Writer, and once the directory is on screen press the equals (=) key. The right hand column will now be marked BA or EA as appropriate..

Other memory image files can be identified as follows:.

File name ends in _P or _C: A picture to load with TI Artist or MAX/RLE..

File is 54 sectors long: MAY be a picture to load with Graphx or MAX/RLE..

File is 25 sectors but doesn't end in _P or _C: MAY be graphics for CERT99..

File is 8 sectors: May be a graphics screen for Fractal Explorer..

File names are identical except for last letter:.

Type 1:	TYPE 2:.
INVADERS	INVADERS
INVADERT	INVADERS1
INVADERU	INVADERS2
Ed/As Option 5	Gram Kracker
Machine Code program	Machine Code program..

If you place the files on a blank new disk, and inspect the header (first 3 bytes) on Sector 22, with a sector editor, as supplied with Funlweb, you may also determine:.

TUNNELS OF DOOM: 0406 0504 0400.

Scott Adams Adventures: 2020 2020 2020.

Module ram required: If third byte is >8000 or >7000.

If the first byte is 0000 you MAY be dealing with graphics such as a CHARA1 file or a GRAPHX picture...

On standard E/A Option 5 files, where there is more than one file, eg INV1,INV2 and so on, the first byte on the LAST file is >0000 while all preceding files have a first byte of >FFFE (Gram Kracker files are >FF05,>FF06 etc).

***.

DISPLAY VARIABLE 80 .

Almost all DV80 files can be loaded into TI Writer for inspection. If they won't load they may have been prepared with the European version of TI Writer, just use the little utility program given elsewhere in this issue to force them to load!.

DV80 files are most commonly:.

TEXT- documentation..

SOURCE CODE - which may contain instructions!.

GRAPHICS: .

Files ending _I,_F,_S are for TI ARTIST..

5 sector and 2 sector files MAY be for Picasso..

Files may be RLE graphics..

***.

DISPLAY FIXED 80:.

Usually used for machine code object files, to load with XB or EA modules. Can also be loaded into TI Writer for inspection..

36 and 68 sector files MAY be fonts for THE PRINTERS APPRENTICE..

***.

DISPLAY FIXED 128:.

Used for RLE GRAPHICS..

MAY be ARCHIVED files which need unpacking..

Could be a special format machine code file requiring a special loader..

***.

INTERNAL FIXED 128:.

Used for JOYPAINT FONTS..

Also used for COMPRESSED ARCHIVED files which need uncompressing..

***.

DISPLAY VARIABLE 163:.

Used for Extended Basic MERGE files..

type MERGE DSK1.FILENAME..

***.

DISPLAY FIXED 254 is used for Draw a Bit graphics..

***.

INTERNAL VARIABLE 254 is used for:.

LONG Extended Basic programs.(Use OLD DSK1.FILENAME as usual).

Data for Creative Filing System.

Data for TRIO SINGS program..

Data for CSGD- watch for file names ending in /CH and /GR etc..

***.

DISPLAY FIXED 255:.

Used for INFOCOM data files, usually GAME1 and GAME2..

Used for Super Disk Cataloguer data files..

.

That will give you a start anyway!.

=====.

.

SECOND HAND PRICES.

What is a good price for second hand gear? What somebody will pay for it... seriously, I have seen adds quoting prices which seem intended to stop anyone even haggling, so high are the prices asked..

If you are selling TI gear, consider what YOU would NOW pay for it yourself!

When five hundred will pay for a complete computer with monitor disk drive and everything else, with high res graphics, faster processor and so on than the TI, why should you expect a TI owner to lay out nearly a thousand on a fully fitted PEB! Everything you are selling is likely to be AT LEAST five years old and probably heavily used! May I suggest a limit of 25% of purchase price for

second hand goods- many will be worth very much less than this though. New games and educational modules are readily available NEW for under three pounds! AND if any members ARE giving up their TIs (shame shame) please do consider donating your equipment to the Group, in support of remaining members..

=====
.....

News of further cable problems, this time the hefty cable to the PE Box, which like the Axiom cable is liable to fractured wires if mistreated or subjected to frequent movement. And if your console moves around a lot....

A possible solution... BOOTS have opened a section for the disabled (ALL branches can order all products if not stocked) and one interesting item is a non-slip mat made by Dycem. It seems to be some sort of very grippy plastic, and has its main use as a bottle grip for opening bottles, but also useful for trays and so on. And very suitable to place between a console and a table top, to very effectively stop console movement. Essential if small keyboard users are around, and essential for stand-alone users. Try it!.

The Dycem product is not to be confused with so-called non-slip rubber mats you can get for typewriters, which very often are more slippery than nothing!. It is VERY grippy!.

=====

.....

Macroassembler?.

The disk library continues to hold the Macro Assembler by Art Green, but what is a macro?.

Example of use. Your source code frequently has code like:.

```
CI R0,168.  
JL LABEL1.  
B @LABEL2 .
```

LABEL1

- that is, check a register value, and jump depending on its value..

Using the macro assembler, you can use a supplied macro, IF, as follows:.

```
IF R0,GE,168,LABEL2.
```

LABEL1

Why? It can ease the writing process! This assembler is faster in operation than TI's, and also has a nicer LISTing format..

(sample code from Mike Dodd, K Town 99ers)

=====

As there seems to be a lot of reinventing of wheels going on in the States, with a few "exciting" new offerings not being quite as good as what has gone before, may I draw your attention to programs in the Disk Library which can:.

LOAD a DV80 file into a running program FAST..

RUN a program LISTed to a DV80 file!.

Change compressed/uncompressed DF80 object files to the alternative format..

Remove the autostart on some DF80 files..

Embed a DF80 file into an ordinary XB program for cassette loading!.

Move a PROGRAM format machine code file onto tape ready to be loaded with EdAs module or, using Funlweb, using XB..

Create a PROGRAM format m/c file from a DF80 file, without having to insert SFIRST, SLAST, and SLOAD labels..

Extract lines from an XB program FAST..

Set an alarm clock to remind you when to finish computing...

and so much more.....

Take another look at the disk library listing, and advise your needs!.

=====

PROGRAM SEARCH:

Here is a tiny program which will search through an XB program for a variable name, keyword, or quoted string, or line number, or just about ANY text- for example, "FOR I=1 TO" or "CALL KEY(3" or "IF A>B AND"..... up to 100 occurrences will be listed for you..

The list on the left starting with line 1 is in

XB and should be saved in . |
MERGE format. |

To find a section of an XB program, FIRST Load the program then MERGE the xb program which starts at line 1 below, into it,

Then type:
32000 and the thing you are searching for, eg
32000 "A" (string only)
32000 A (string + var. name)
32000 GOSUB (all GOSUBs)
there are a few things XB will not permit you to type in. There is also an exception- line numbers.

To find a NUMBER 100 use:
32000 100
to find any reference to LINE NUMBER 100 however (for instance, GOTO,GOSUB, IF, etc) you must use:

32000 GOTO 100
The first entry of GOTO on line 32000 is ignored. To search for all GOTOs, use:
32000 GOTO GOTO

...
The program:

1 CALL FIND :: !@P-
32000 !
32010 !@P+
32020 SUB FIND
32030 ! SEARCH FOR CONTENT OF LINE 32000
32040 DIM A(100)
32050 ON ERROR 32080 :: CALL LINK("FIND",A()) :: ON ERROR STOP
32060 FOR I=1 TO 100 :: IF A(I)=0 THEN STOP
32070 PRINT A(I); :: NEXT I
:: STOP
32080 CALL ERR(B,C) :: IF B=84 THEN PRINT "ERROR: INSERT PATTERN IN 32000" :: STOP
32090 IF B=57 THEN RETURN 32060 ! SUBSCRIPT ERROR

This is the source code which is to be assembled into the file FIND/OB:

```
DEF FIND
START EQU >8330 * START OF LINE NO TABLE
FINISH EQU >8332 * END OF LINE NO TABLE
XMLLNK EQU >2018 * FOR EX BAS USE !!!
NUMASG EQU >2008
FAC EQU >834A
GOTO BYTE 134,0 * Filter GOTO
FIND BLWP @FIND1
RT
FIND1 DATA FINDWS,FIND2
FINDWS BSS 32
FIND2 MOV @START,R1
GETPAT BL @MOVE
CI R6,32000 * PATTERN IN LINE 32000
JEQ GOTPAT
JL ERROR
AI R1,4
JMP GETPAT
ERROR LI R0,>2100 * DATA ERROR
BLWP @>2034
GOTPAT INCT R1
BL @MOVE
MOV R1,R8
MOV R6,R9
DEC R9
CLR R10
MOVB *R9+,R10 * PATTERN LENGTH TO R10
SWPB R10
CB *R9,@GOTO * GOTO FILTER
JNE NO
INC R9
DEC R10
NO MOV @FINISH,R2
DEC R2
CLR R0
LOOP C R2,R8
JEQ DONE
MOV R2,R1
BL @MOVE
MOV R6,R3
DEC R3
CLR R15
MOVB *R3+,R15
```


32100 IF B=14 OR B=135 THEN		SWPB R15	
CALL INIT :: CALL LOAD("DSK1		LOOP2 DEC R15	
.FIND/OB") :: RETURN 32050		JEQ NOMTCH	
32110 RETURN		MOV R3,R4	
32120 SUBEND		MOV R9,R5	
=====		MOV R10,R12	
It IS possible to transfer the		LOOP3 DEC R12	
machine code into CALL LOAD		JEQ MATCH	
format, and add it to the		CB *R5+,*R4+	
MERGE file, but on a long XB		JEQ LOOP3	
program, this adds consider-		INC R3	
ably to both the time taken		JMP LOOP2	
to MERGE the file in, and		MATCH MOV R2,R1	
then to RUN it.		DECT R1	
		BL @MOVE * LINE NO TO R6	
This code is from		MOV R6,@FAC * TO F/PT ACCUM	
STEVEN KARASEK and was first		BLWP @XMLLNK * TO F/POINT NUMBER	
published in:		DATA >20	
COMPUTER BRIDGE		INC R0	
Vol 5 No 11 November 86		LI R1,1	
		BLWP @NUMASG * LINE NO TO BASIC ARRAY	
=====		NOMTCH AI R2,-4	
		JMP LOOP	
The utility uses the on-board		DONE RTWP	
tokenisation facilities on		MOVE INC R1	
line 32000 and compares the		MOVB *R1,R6	
tokenised form to the entire		SWPB R6	
program in memory.		DEC R1	
NOTE: If your basic program		MOVB *R1,R6	
contains a line with !@P+ in		RT	
it, those symbols will almost		END	
certainly have to be removed			
in order for this utility to			
run properly!			

Retyped by sjs. Queries/complaints to S Shaw!

Repeated use of this program alternating with running of your XB program is possible just by removing lines 1 and 32000 to RUN YOUR XB program, and by re-inserting lines 1 and 32000 to again use the utility. The CALL LOAD will not be repeated provided your program does not have a CALL INIT in it... this will save on the frustrations of the speed of MERGEing....

+++++
=====

Here are some interesting routines by Dr Roy T Tamashiro, for XB plus 32k..

FLIP.

Lines 100 to 150 are a DEMO program. The FLIP routine is in lines 30100 onwards and may be saved as a merge file

```
100 CALL FLIP
110 CALL CLEAR
120 PRINT "FLIP DEMO": :
130 FOR I=33 TO 126 :: PRINT
  CHR$(I); :: NEXT I :: PRINT
140 CALL LINK("FLIP")
150 INPUT "Press enter":R$:
: GOTO 110
160 !
170 !
30100 SUB FLIP
30110 ! 1986 R TAMASHIRO
30120 ! COMPUTER BRIDGE
30130 ! NOVEMBER 1986
30140 ! XB + 32K
30150 !
30160 CALL INIT :: CALL LOAD
(16376,70,76,73,80,32,32,36,
244) :: CALL LOAD(8196,63,24
8)
30170 CALL LOAD(9460,2,0,4,0
,2,5,37,72,2,2,0,2,2,4,0,4,2
,1,40,56,4,32,32,44)
30180 CALL LOAD(9484,192,96,
40,56,6,5,213,65,6,193,6,5,2
13,65,5,192,2,128,6,240,21,5
,6,4)
30190 CALL LOAD(9508,22,239,
2,37,0,16,16,234,2,2,2,240,2
,1,37,64,2,0,4,0,4,32,32,36,
4,96,0,112)
30200 PRINT "FLIP LOADED"
30210 SUBEND
30220 END
```

UPSIDE DOWN DEMO

```
(-Merge FLIP into it!!!-)
100 CALL FLIP ! Merge it in!
110 DATA 1,"UPSIDE DOWN"
120 DATA 3,"by Roy Tamashiro
"
130 DATA 5,"from Computer Br
idge"
140 DATA 6,"November 1986"
150 DATA 15,"After a while,
everything"
160 DATA 16,"gets corrected.
.."
170 DATA 18,"It will wait fo
r you to"
180 DATA 19,"press the ENTER
key"
190 DATA 99,"DUMMY DATA"
200 RESTORE :: CALL UPSIDE
30000 GOTO 200
31500 SUB UPSIDE
31510 DIM W$(24) :: FOR I=1
TO 24 :: W$(I)=" " :: NEXT I
31520 READ R :: IF R<25 THEN
  READ W$(R) :: GOTO 31520
31530 CALL LINK("FLIP") :: C
ALL CLEAR :: FOR I=1 TO 24
31540 DISPLAY AT(24-I,14-LEN
(W$(I)/2):W$(I)
31550 NEXT I :: CALL DELAY
311560 FOR I=1 TO 24 :: CALL
  HCHAR(I,1,32,32) :: DISPLAY
  AT(I,14-LEN(W$(I)/2):W$(I)
  :: NEXT I :: CALL DELAY :: C
ALL LINK("FLIP")
31570 DISPLAY AT(24,1):"PRES
S ENTER" :: CALL KEY(0,K,S):
L IF K<>13 THEN 31570
31580 CALL CLEAR :: SUBEND
31590 SUB DELAY :: FOR D=1 T
O 1000 :: NEXT D :: SUBEND
31600 END
```

=====

The ANNUAL MEETING went of well (I quite like Derby! this year there was even a train DIRECT from STOCKPORT to DERBY! No such luck getting back, fortunately one of the Committee gave me a lift back - even more fortunately as I later found that trains from Sheffield to Manchester were NOT arriving at their advertised destination on that day due to track relaying! (Thanks)..

The turnout was probably pretty good considering our membership - perhaps we can look for recruiting a few more members to support us next year!. It was nice to see Francis Parrish once again, and have a good chin-wag. PARCO are still trading, and still have good stocks of the older modules, and limited stocks of the newer ones. Peripherals and expansion systems are clearly in demand (even if only limited demand, demand exceeds supply). Non-attendees

missed some first class bargains, and some ridiculous offers were not taken up (does everyone really have everything there is now? Or is 10p just too much?).. Many thanks to Parco for subsidising the costs of the event..

Nicer still to see the Geneve working- pity the highest-res graphics disks turned out to have been wiped (aaaagh!) but the display on the Phillips monitor was still very impressive. The 80 column word processor and Multiplan were also very nice. Pity about the cost... .

Good also to see Peter Brooks, a gentleman who has supported the TI in this country for about as long as anyone possibly could. And in addition to the committee members, managed to meet one or two members also.

A very pleasant day. We must get together again sometime.....

=====

Q: How do I transfer programs from CASSETTE to DISK?.

a. Before loading the tape, type in:.

```
CALL FILES(1).
NEW          then.
OLD CS1.
```

b. If it loads but wont run, you need to use XB. The utility VDPUTIL from the disk library will enable a TIB program to run in XB..

c. Send me the tape and a disk and I will MAKE it run from disk- 32k ram required! Tools available include Myarc XB, which is an excellent conversion tool, and procedures to shorten programs..

=====.

Q. How do I redefine the CURSOR in a running XB program?.

A. The answer to this query is on page 57 of TI*MES Issue 8, and comes from Jim Peterson of Tigercub Software. It REQUIRES 32k ram..

```
100 ! Define character N      .
with the definition you      .
require.Now proceed:         .
110 CALL CHARPAT(N,A$) :: FO. .
R J=1 TO 16 STEP 2 :: H$=SEG. .
$(A$,J,2) :: CALL HEXDEC(H$, .
D) :: T=T+1 :: H(T)=D :: NEX. .
T J                           .
120 CALL INIT :: CALL LOAD(8. .
196,63,248)                   .
130 CALL LOAD(16376,67,85,82. .
,83,79,82,48,8)              .
140 CALL LOAD(12288,H(1),H(2. .
),H(3),H(4),H(5),H(6),H(7),H. .
(8))                          .
150 CALL LOAD(12296,2,0,3,24. .
0,2,1,48,0,2,2,0,8,4,32,32,3. .
6,4,91)                       .
160 CALL LINK("CURSOR")      .
!program here ----- then: .
```

```
20000 SUB HEXDEC(H$,D) :: N=.
1 :: DEC=0
20001 FOR J=1 TO LEN(H$) :: .
A$=SEG$(H$,LEN(H$)-J+1,1) ::.
IF ASC(A$)>58 THEN HT=ASC(A.
$)-55 ELSE HT=VAL(A$)      .
20002 DEC=DEC+N*HT :: N=N*16.
:: NEXT J                  .
20003 IF DEC<=32768 THEN D=D.
EC ELSE D=-(65536-DEC)     .
20004 SUBEND                .
20005 END
```

=====

TI EDITOR ASSEMBLER BUG:

If you wish to use PIO as an output device for a LIST, you have to put a full stop after it to make it work - eg PIO. - due to an error in the ASSM2 file. The error lies in the DSRLNK routine which the assembler uses - it does not use the same routine that YOUR programs call! The assembler internally uses a modified VSBR routine which returns data to R0 instead of the usual R1, but whoever wrote the internal DSRLNK forget this! Consequently, the assembler looks at R1, and the program uses an incorrect name length when scanning for the device name! The period causes the routine to exit, and "catches" the error..

Want to fix it? If you still have the original ASSM2 file, use a sector editor to look for the code 04 20 AA BE 00 00 D1 C1. It is usually in the 8th sector of the file..

Change that last C1 into a C0 and the Assembler will work as it is SUPPOSED to! No need to add a full stop after PIO now!.

Courtesy Danny Michael (Dump and Neatlist author), SHOALS 99ERS..

=====.

From Tips from the Tigercub Number 8:.

A better conditional line jump:

Original:.

```
100 IF X=1 THEN Y=7 ELSE IF X=2 THEN Y=33 ELSE IF X=3 THEN Y=19 ELSE IF X=4 THEN Y=21..
```

Better:.

```
100 Y=VAL(SEG$("07331921",X*2-1,2)).
```

=====.

I tend to cycle round the programs in my collection - it takes quite a while - and from time to time rediscover some interesting programs. .

Music disks have not been popular in the disk library, although my son and I are quite fond of them. A POP MUSIC disk from Germany with four (non-pop!) musical items is very much liked, as are the several Sam Moore programs in XB, which are mainly on Amnion disks..

Speaking of music disks, a recent addition is a 77 minute rendition of J C Bach's Sonata for Pianoforte Opus 5 Numbers 1 to 6- on two disks of machine code! Think of the time spent coding it! And no authors name either..

Long long ago, when there were 20 members in the ONLY TI UK Users group, a software company called APEX TRADING was set up by Vince Apps, who also managed what may have been the first UK published TI Book (a collection of programs). These were early days, and Vince's programs were not exactly classics, but one I heard several complaints about I have just been playing, and find the complaints then heard quite unjustified. I mean- what sort of flight simulator can YOU write in TI Basic??? I think Vince's PILOT program was quite playable. It is certainly retained on MY playlist. (Anyone know what Vince is doing these days?)..

Possibly I am the only disk owner to play TI Basic games these days- but there really are several playable games in Basic and ExBas, especially if like me your reflexes are off, and after a days work you want to relax rather than wind up! .

Cassette owners- check out our cassette library this quarter, as I have supplied around a hundred (?) of the games I play to Tim, to beef up our collecton. What games do YOU enjoy playing?

=====.

David Caron of Ottawa reminds me of these tips for XB users....

1. In order to quickly tell the difference between I and 1 and between 0 and O when keying a program in - a source of frequent errors! - first key this in, in immediate (command) mode:.

```
CALL COLOR(3,16,0) :: ACCEPT AT(1,1):A$ .
```

... when you press enter, the cursor will appear at top left..

... Press CLEAR [FCTN 4] and you can begin to type in your program, and the numbers 0 to 7 will really stand out!

2. To REALLY annoy someone, with a VERY long sound in your program, instead of the usual meagre 4 seconds or so, and without having to add more CALL SOUNDS, add these lines to your program-and turn the noise (sorry, sound) on with a GOSUB 1000:.

```
1000 ! NOISE.  
1001 ON ERROR 1005.  
1002 CALL SOUND(-4150,110,0).  
1003 CALL SOUND(-9999,110,1).  
1004 GOTO 1006.  
1005 RETURN NEXT.  
1006 RETURN
```

(May not function on all XB modules... tell me if it doesn't on yours, and quote the serial number on the module and the error message!)..

=====

.....UTILITIES FOR PROGRAMMERS : **DISPLAYING PROGRAMS**

There are many ways of producing program listings, and many ways of manipulating programs. This little article is to show you one program, which has been manipulated with three others, so you will see the original and three variations. Not shown here is the 28 column format, produced by Tony McGovern's COLIST program, as that has been used many times in this magazine already..

COLIST, UNBASHER, and NEATLIST are all available from the User Group disk library, while SMASH is a commercial program available from Tenex..

First the original program- I am only showing a central portion of the program, rather than take up too much magazine space!.

```
1440 FOR X1=1 TO 4  
1460 R(X1)=ASC(SEG$(P$,X1,1))  
1480 NEXT X1  
1500 W=0  
1520 FOR I=1 TO 4  
1540 FOR J=1 TO 4  
1560 IF G(I)<>R(J)THEN 1640  
1580 W=W+1  
1600 R(J)=0  
1620 GOTO 1660  
1640 NEXT J  
1660 NEXT I  
1680 W=W-B  
1700 RETURN
```

continued.....

Once this program has been SMASHed, these lines look like this:.

```
.
1440 FOR X1=1 TO 4 :: R(X1)=ASC(SEG$(P$,X1,1)):: NEXT X1 :: W=0 :: FOR I=1 TO 4
:: FOR J=1 TO 4 :: IF G(I)<>R(J)THEN 1640
1580 W=W+1 :: R(J)=0 :: GOTO 1660
1640 NEXT J
1660 NEXT I :: W=W-B :: RETURN
```

Which can be made a little more meaningful by running it through NEATLISTER:.

```
.
01440 FOR X1 = 1 TO 4 ::
      R(X1) = ASC(SEG$(P$ , X1 , 1)) ::
      NEXT X1 ::
      W = 0 ::
      FOR I = 1 TO 4 ::
      FOR J = 1 TO 4 ::
      IF G(I) < > R(J) THEN 1640
01580 W = W + 1 ::
      R(J) = 0 ::
      GOTO 1660
01640 NEXT J
01660 NEXT I ::
      W = W - B ::
      RETURN
```

VARIABLES USED IN MAIN PROGRAM
(in lines 1440 to 1700 only)

G	01440
I	01440 01660
J	01440 01580 01640
P\$	01440
R	01440
W	01440
X1	01440

(The variable listing is available as an option with the listing, or on its own)..

We can also resplit the SMASHed program by using UNBASHER, which produces the following- note that the output has not been resequenced, but of course it could have been:.

```
.
1440 FOR X1=1 TO 4
1441 R(X1)=ASC(SEG$(P$,X1,1))
1442 NEXT X1
1443 W=0
1444 FOR I=1 TO 4
1445 FOR J=1 TO 4
1446 IF G(I)<>R(J)THEN 1640
1580 W=W+1
1581 R(J)=0
1582 GOTO 1660
1640 NEXT J
1660 NEXT I
1661 W=W-B
1662 RETURN
```

continued.....

You may not have a lot of control over the format of a program you purchase or receive from elsewhere- such as the library- but the utilities are available to change the format to something which may be easier to read/debug, or which perhaps occupies a little less memory space and runs a trifle faster..

.
=====.

"We would like everyone to know that the Ottawa TI-99/4A Users' Group recommends that anyone using DM1000 Vn 3.6 or higher go back to 3.5. It is the most stable version at the moment, and later versions may [ss: nice word: MAY!] contain potentially serious bugs. We apologise to anyone who may have experienced them, and hope that they will enjoy using 3.5 until our next release..

"Version 4.0, for example, is an escaped beta copy and should NOT be used under ANY circumstances. We hope to have a new version (thoroughly beta-tested) ready to go in the fall[1988]."[ss: Nothing heard since this- still at Vn 3.5 as at June 1990].

=====.

ENHANCED BASIC.

Or What You Can Do when you have the *Personal Record Keeping* or Statistics modules inserted into your console, or, if loaded from disk, if your console thinks they are inserted....

Last issue we dealt with the really easy bit, the equivalent calls to ACCEPT or DISPLAY data..

This issue we shall dive into the mysteries of CALL L (load), CALL P (partition), Call H (header), and Call G (getput)..

The best way to introduce these perhaps is with a working program, then in the next issue we can get down to a little more detail!.

Program first, then discussion. This program will use an already created PRK data file, and in the TI Basic environment will read that data and display it on the screen. It will demonstrate several of the calls in a fairly simple manner, so you can get used to the idea of using PRK data in a Basic program!.

.
If you have a disk system, free up memory by typing:.

CALL FILES(1).

NEW.

before you load and run this program! .

.
As this program occupies some VDP memory, you may find that a few PRK data files will be too long to load. Try deleting some records to make the data file fit..

.
Remarks will be in lower case between the program lines!.

.
BEFORE LOADING THE PROGRAM you must allocate an area of VDP memory for the data, by using the partition command, in command mode, thus:.

CALL P(10900).

NEW.

.
continued.....

Now load the following program and run it:.

```
.
first,load the prk file from disk or cassette .
100 CALL L("CS1",Y).
or call l("dsk1.filename") as you wish.
now to check to see if the file is loaded.
110 IF Y=0 THEN 500 .
no there was an error.
else carry on as follows.
how many fields in each record? lets find out.
120 CALL H(1,5,0,F).
where f will return number of fields..
130 CALL CLEAR.
140 CALL D(1,1,28,"# OF FIELDS:"&STR$(F)).
then we need to know how many records there.
are in the data file- think of a record as a page:.
150 CALL H(1,6,0,R).
160 CALL D(2,1,28,"# of RECORDS:"&STR$(R)).
now let's loop through each record and.
display it on the screen:.
170 FOR RCD=1 TO R.
taking each field in turn.
180 FOR FLD=1 TO F.
but does the field contain a number or a string?.
let's find out:.
190 CALL H(1,10,FLD,TYPE).
200 IF TYPE=1 THEN 240.
as type is 0 data is a number:.
get the data from field FLD of record RCD:.
210 CALL G(1,RCD,FLD,Z,RD).
and print the result on screen:.
220 CALL D(2+FLD,1,28,RD) .
and jump over string section.
230 GOTO 260
this is string section: .
240 CALL G(1,RCD,FLD,Z,RD$).
250 CALL D(2+FLD,1,28,RD$).
thats one field dealt with, on to the next:.
260 NEXT FLD.
and now all fields in the record are on screen.
lets give the record number and a pause.
270 CALL D(23,18,10,"RECORD:"&STR$(RCD)).
280 CALL D(24,1,28,"PRESS ENTER FOR NEXT").
290 CALL A(24,28,1,RC,RV$)
and on to the next record.
300 NEXT RCD.
310 PRINT "NO MORE".
320 GOTO 320.
500 CALL CLEAR.
510 PRINT "LOAD ERROR".
520 GOTO 520.
530 END.
```

.
CALL P sets aside an area of memory for the data to be stored in, and must of course be sufficient to hold the data, but for Basic use, we must also have enough room for our Basic program! After using CALL FILES(1), disk users have a maximum of 12768 bytes of VDP RAM free to split between Basic program and data

area, while cassette users have 13820 bytes available (see how much memory a disk system eats up!). By using 10900 for our data, we have left even disk users with at least 1868 bytes for our little program!.

.
CALL L just loads the data file into the partitioned area, the return variable following the device/file name is a 0 (zero) if an error occurs, for instance, the files is not on the disk, the data is too large for the partition, or other I/O errors. The device/filename may be a string variable if you wish, e.g. CALL L(A\$,A).

.
CALL H deals with the "header" information- the database specification you create when using PRK/Stats: what sort of data is held in each field and what name have we given the field?.

.
As used in the above example (there are many other uses!!!):.

.
CALL H(1,5,0,F).

CALL H(1,6,0,R).

CALL H(1,10,FLD,TYPE).

The first digit (1) is a READ instruction. We would use a 0 to write.

The fourth digit is a variable placed in or obtained from the header data..

The second digit (5,6, or 10 above) is the header item number, from 1 to 14, where 5 is the number of fields per record, and 6 is the number of records..

10 is the type of field, returning to the fourth item, a numeric variable, a 1 for characters (string), a 2 for Integer, a 3 for decimal, and a 4 for scientific notation..

We shall deal with the other 11 header items, and with writing to the header, in a later article..

.
CALL G deals with the actual data we have stored, and as used above (again there are other uses...):.

.
CALL G(1,RCD,FLD,Z,RD).

CALL G(1,RCD,FLD,Z,RD\$).

.
Again the first item (1) is a READ instruction, with 0 used to write..

The second and third items identify the record and field number that we are going to read/write, while the final item is the variable that the contents of that particular record/field will be placed in for us to use. Note that we must use a string variable for characters!

That fourth item (Z) is a numeric variable, which can be used to indicate a blank field. It takes a value 0 if data is found, and 1 if data is missing, that is, not entered!

When using CALL G to WRITE, the fourth parameter as above is not used, you just go on to the value you are writing, for instance:.

CALL G(0,RCD,FLD,"STRING INPUT").

That fully covers CALL G, apart from a sample of how to write!.

CALL S will Save the data area to cassette/disk, very easily indeed:.

CALL S(DEVICE\$,VARIABLE)

with variable again indicating success or failure!.

.
That leaves us with a great deal to look at with CALL H, and also we need to look at writing data, and (perhaps!) maybe even creating a data file WITHOUT using the PRK create file routine. Until later...

.
((Is anyone reading this please!!! Is this the right level/approach for you?)).

=====

GRAPHICS : COMPATABILITY

This article has been prompted by a very odd chart of the various Graphics programs for the TI which I came across in a US newsletter- odd because at the end of the day it failed to tell you very much and was decidedly biased!.

This article also follows -in a way- from the discussion of various formats of disk file..

Each type of file is referred to by means of a short abbreviation, details of which are given in the first section below:.

1. List of Formats:.

TI ARTIST- Fonts (_F files, referred to later as TIAF).
Pictures (_P and _C files, referred to as TIAP).
Slides (_S files, TIAS).
Instances (_I files, TIAI).
GRAPHX....Clipart, inc fonts (GC).
Pictures (GP).
CSGD.....Pictures (/DT files, CP).
Graphics (/GR files, CG).
Fonts-usual (/CH files, CF).
-care: see note at end!!.
Fonts-DocuPrinter (/DP files, CD).
Labels (/LB files, CL).
Letterheadings (/1L files, CH).
JOYPAINT...Pictures (JP).
Compressed pictures (JC).
PICASSO...Pictures (PP).
Fonts (PF).
Icons (PI).
BITMAC....Pictures (BP).
DRAW N PLOT..Pictures (DP).
DRAW A BIT 1.Pictures (DAB1).
DRAW A BIT 2.Pictures (DAB2).
MAX RLE....Pictures: DV80 files or DF128 files (MP).
PAINT N PRINT (Module)...PNPP.

Max RLE and PICASSO are available from the disk library.

NOTE: CSGD uses two different sets of /CH files. The font editor creates one set of /CH files, which then have to be converted to another type of /CH file for use. The /CH files referred to here are always the converted files. The conversion program is on CSGD Volume 1..

continued.....

2. MUTUALITY:.

This section indicates the types of file each graphics program can use from the above list, WITHOUT using an external conversion utility. The ability to both save and load can be assumed unless otherwise noted:.

MAX/RLE.....TIAP, GP, MP
TI ARTIST.....TIAP,TIAF,TIAS,TIAI,DAB1,DAB2,DP,GP.
GRAPHX.....GC,GP.

CSGD 1 AND 2.....CP,CF,CG.
CSGD 3.....CF,CG,CH,CL and LOAD ONLY CD..

PICASSO.....PP,PI,PF,TIAP.
Can also LOAD a TI Writer text file..
JOYPAINT.....JP
JOYPAINT PAL 2...JP,TIAP,JC.
Can also LOAD GP,DP .

Where more than one type is listed in the above section, conversions are possible as part of the main program, which is usually much faster..

3. GRAPHICS UTILITIES including external (eg separately loaded) conversion routines on main graphics disks:-.

THE PRINTERS APPRENTICE... Uses its own picture and font formats, can also use TIAP..

TPA TOOLBOX.....Uses TPA fonts and graphics, plus can convert into TPA format the following: TIAI,TIAF,TIAP,CF.

PRINT WIZARD.....Creates its own format from TIAI and TIAF.

FONT WRITER 2.....Uses, in various utilities, TIAF,TIAI,TIAP,CF,CG,GP.

CAN CONVERT: CG to TIAI, CP to TIAI, TIAI to CG and TIAI to CP.

PICASSO can convert an XB font to PF, or load a PF into an XB program..

Convert BP to PP. Make use of CF and CG files..

EXTENDED GRAPHICS PACKAGE. Requires Paint n Print module.

CSGD 1 can convert an XB screen into CP..

ARTIST EXTRAS (Texaments) can convert: CF or CD to TIAF, CG to TIAI, and CP to TIAI. Allows SUPERSKETCH to be used as an input device for TI Artist.

ARTCONVERT (Trio+) can convert TIAI and TIAF to TI Writer graphics..

ARTIST ENLARGER(Asgard) works with TIAF and TIAI..

GRAPHICS EXPANDER AND BIGTYPE (Genial) works with TIAF,TIAP, and TIAI..

GRAPHICS LISTER (Nameloc). TIAI.

PICASSO UTILITIES(Asgard). Ad description fails to indicate what this does..

DISPLAY MASTER (Inscebot). TIAP.

CSGD CATALOGER (sic) (Texaments). CG.CF.TIAF.TIAI.

GRAPHX SLIDE SHOW.(Asgard). GP.

DESIGNER LABELS. (Texaments) TIAI.

EXTENDED BUSINESS GRAPHS (Gt Lakes) JP.

CHART MAKER II (QS99). DP.

CALENDAR MAKER 99.(Asgard). TIAI.

PICTURE IT. (Merritt).(Disk Library).TIAI to Banner, XB, and TI Writer.
GRAPHIC LABELLER (disk library). CG.
JBM103 (Disk library) enables graphics to be loaded/saved to/from Extended Basic bit-mapped screens in TIAP format..
UTIL12 (Disk library) has a utility to convert from TIAI to Extended Basic program format- merge file, or listing to disk or printer..
UTIL 7 (Disk Library) has a utility to convert TIAI to TI Writer graphics..
UTIL17 (Disk library) has a utility to convert a segment (5x5 chars) of a GP to CG, and a utility to convert CG to TIAI and/or Extended Basic merge file.(Called XBGC)..
TASS (Disk Library) TIAP.GP.DAB2. Slide show.
COMIC 1 (Disk Library)A utility which enables you to create a machine code animation sequence from up to 100 TIA Pictures.(TIAP). The animation speed is adjustable as the program runs, and can be very fast indeed..
MYARC UTILITIES (Disk library). TIAP and GP to load into Myarc XB program.

Programs in MICROpendium. Back issues available to subscribers:

Feb 1987. CONVERT TIAP to/from Myarc XB.
July 1987. ROTATE. TIAI.
Oct 1987. Print utility. TIAI.

PICASSO and PICTURE IT are COPYRIGHT. Other Disk Library programs are FAIREWARE. All other programs are copyright commercial programs.

The de facto standard has been set by TI ARTIST- only graphics programs released before TI Arist lack TIA capabilities, apart from CSGD, although external utilities have been created to remedy that!.

As far as PRINTERS go, all these work with EPSON FX series printers or any printer which follows Epson commands- the usual commands used are:.

ESC * (8 pin bit image mode).
ESC K (480 DOT 8 PIN mode).
ESC L (960 dot 8 pin).
ESC Z (1920 dot 8 pin).
ESC A n (Line spacing in n/72 inch).
ESC l n (Left margin setting).

A few programs allow GEMINI printers to be used, but Gemini used two incompatible codings in their printers, and Gemini owners often report problems. A very few programs will support other printer codings..

The vast majority of the programs listed above remain available. Not a bad choice at all for an orphan computer, whose manufacturers left it with a VIDEO GRAPHS module which compares very badly with the above..

I have deliberately omitted a few simple programs such as Nortons Graphics Package, and an input device, SUPERSKETCH- but note that Supersketch can be interfaced to TI Artist. Any other omissions are due entirely to my ignorance of the products involved, or in the case of very new items, news was not with me when this was printed!

[Addendum: Since the above was written TI owners have been given access to MACPAINT files with both MACFLIX PRO and PIX PRO. MYART has also been made available to TI owners through XHi and SmArtCopy. There is also PAGE PRO from Asgard with its own format.]

TI WRITER DATABASE...

TI Writer can function as quite an effective database, if you use FS to locate a specific item. You can even use M to put your items into order....

The program which follows will enable you to use each line of TI Writer as a separate record, divided into up to 9 fields, and with up to 400 records. Not bad eh! Quite sufficient for simple database purposes..

This program is an XB sorting routine, which will sort the records by any one field.

The fields are identified in line one of the TI Writer file, by numbers located in the left-most column of each field. Numbers may be zero filled, or RIGHT justified, as follows:.

1	2	3	4	5.
TI*MES	JAN-MAR 88	43	PILOT 99	ARTO HEINO.
TI*MES	JAN-MAR 88	8	DISK DRIVES	STEPHEN SHAW.

Save the DV80 file using SF as usual, this program will strip the tab records out for you!.

Use of TI Writer as a database is of value as it is familiar, easy to use, and easily edited!.

SORT TI WRITER FILES:.

This program comes from COMPUTER BRIDGE, Vol 5 No 11, Nov 86, and was written by George Paschetto..

```
...
100 REM PRESCAN
110 GOTO 120 :: D$,X,Y,K,S,Z
,FN,FS,FL,L,S$,H,J,B$ :: CAL
L KEY :: CALL SOUND :: CALL
HCHAR
120 DISPLAY AT(4,2)ERASE ALL
:" See Instructions? Y/N" ::
DIM A$(400),F(10)
130 !@P-
140 ACCEPT AT(5,12)VALIDATE(
"YN"):D$ :: CALL CLEAR :: IF
D$="N" THEN 230
150 PRINT " This program was
written to sort files create
d with the TI Writer. The fi
le can be as long as 400 li
nes."
160 PRINT " The first line o
f the file must contain the
field numbers, like thi
s:"
170 PRINT "1 2 3 <-first
line":102 NJ Paul Peters":
"314 MO Carol Corrina":"622
AL Hu Noes":
180 PRINT " You may have up
to 9 fields but they must all
fit on one line.": :: INPUT
"<press enter>":D$ :: CALL C
LEAR
```

```

190 PRINT " The lines can be
up to the full 80 character
s long that TI Writer support
s. This program will only
sort one field at a time."
200 PRINT : " The sort is al
ways": "ascending order, with
the lowest value first."
210 PRINT " The file can be
used by TI Writer after this
program is through with i
t.":
220 INPUT "<press enter>":D$
:: CALL CLEAR
230 DISPLAY AT(12,1):"Input
device and file name:": " DSK
"
240 ACCEPT AT(13,5):D$ :: IF
D$="" THEN 240 ELSE OPEN #1
:"DSK"&D$,DISPLAY,VARIABLE 8
0,INPUT
250 CALL CLEAR
260 REM LOOK FOR 1ST LINE
270 LINPUT #1:A$(0) :: IF EO
F(1) THEN 610 ELSE IF A$(0)=
" " THEN 270
280 FOR X=1 TO 400 :: LINPUT
#1:A$(X) :: IF EOF(1) THEN
310
290 NEXT X
300 REM F()=field's position
s (count down 2 from eof():l
ast 2 lines are tabs)
310 X=X-2 :: CLOSE #1 :: FOR
Y=1 TO 9 :: F(Y)=POS(A$(0),
STR$(Y),1) :: IF F(Y)=0 THEN
330
320 NEXT Y
330 F(Y)=80 :: Y=Y-1 :: IF Y
THEN 350
340 PRINT "Can't find field
marker": "This was found inst
ead":A$(0) :: STOP
350 DISPLAY AT(10,5)ERASE AL
L:"Press:": "1. To Sort": "2.
To Save to disk": "3. To QUIT
"
360 CALL KEY(3,K,S) :: IF (K
<49)+(K>51) THEN 360 ELSE CA
LL SOUND(-20,880,0) :: ON K-
48 GOTO 380,540,580
370 REM GET FIELD TO SORT
380 DISPLAY AT(1,3)ERASE ALL
:"<choose field to sort>"
390 FOR Z=1 TO Y :: DISPLAY
AT(Z*2,1):"Field No:":Z;" ";
SEG$(A$(1),F(Z),F(Z+1)-F(Z))
:: NEXT Z ! continued /

```

```

400 CALL HCHAR(20,1,30) :: C
ALL KEY(3,K,S) :: FN=0 :: CA
LL HCHAR(20,1,32)
410 IF (K<49)+(K>48+Y) THEN
400 ELSE FN=K+48
420 REM SORTING
FN=FIELD NUMBER
430 REM
FS=FIELDS START
FL=FIELDS LENGTH
440 DISPLAY AT(12,5)ERASE AL
L:" Sorting..." :: FS=F(FN)
:: FL=F(FN+1)-FS :: K=X :: L
=INT(X/2)+1
450 IF L<>1 THEN L=L-1 :: S$
=A$(L) :: GOTO 470
460 S$=A$(K) :: A$(K)=A$(1)
:: K=K-1 :: IF K<1 THEN A$(H
)=S$ :: GOTO 350
470 J=L
480 H=J :: J=J+J :: IF J>K T
HEN A$(H)=S$ :: GOTO 450
490 IF J>=K THEN 510
500 IF SEG$(A$(J),FS,FL)<SEG
$(A$(J+1),FS,FL) THEN J=J+1
510 IF SEG$(S$,FS,FL)>=SEG$(
A$(J),FS,FL) THEN A$(H)=S$ :
: GOTO 450
520 A$(H)=A$(J) :: GOTO 480
530 REM SAVE ON DISK INC TAB
540 DISPLAY AT(12,1)ERASE AL
L:"Device and filename:": " D
SK";D$
550 ACCEPT AT(13,6)SIZE(-12)
:D$ :: IF D$="" THEN 240 ELS
E OPEN #1:"DSK"&D$,DISPLAY,V
ARIABLE 80,OUTPUT
560 FOR Y=0 TO X+2 :: PRINT
#1:A$(Y) :: NEXT Y :: CLOSE
#1 :: GOTO 350
570 REM QUIT
580 DISPLAY AT(12,1)ERASE AL
L:"<ARE YOU SURE? Y/N>"
590 CALL KEY(3,K,S) :: IF S<
>1 THEN 590
600 IF K=89 THEN STOP ELSE I
F K=78 THEN 350 ELSE 590
610 DISPLAY AT(22,1):"File n
ot properly organised": " <pr
ess enter>" :: INPUT D$ :: G
OTO 230
620 END

```

Submitted september 1988

9/88

.
RAMBLES.. October 1988. by Stephen Shaw. .
10 Alstone Road STOCKPORT Cheshire SK4 5AH.

Yet another TI programmer is no longer with us. Bill Knecht, from whom the disk library has three disks of music, passed away on July 9th, at the tender age of 41..

=====
MOONMINE module:.

How do you pick up objects- eg water?.

A: When the object appears at the bottom of the screen, hit SPACE. See that new little man? Walk him over to the object and press FIRE and he picks it up! To return to the ship you must first destroy that monster lurking on the screen!.

=====
SUPERSPACE I and II. These are modules which contain the Editor Assembler GROM chip, plus 8 or 32k of ram, battery backed. If you take the 32k version, you have to incorporate bank switching into your software to take advantage of the extra 24k. The standard 8k is at >6000 onwards, and can be used to store machine code programs, including Funlweb, which is supplied with a utility to place it into the module. .

Superspace also allows us to make use of machine code programs that just won't fit into the standard 32k ram. Thanks to Super Space I am now able to run the Infocom adventure LEATHER GODDESSES OF PHOBOS (which arrived in an anonymous brown paper bag from Canada..) and also the other adventures that Infocom released for the IBM PC but not for the TI99/4A. .

My Superspace arrived with a menu item on board already which allows me to use it as a substitute module for Myarc's XB module - always nice to have a stand by.

I am also able to load all the Scott Foresman READING modules from disk, and my favorite game, MANCALA - without first having to load Myarc XB. For cassette owners with 32k, they can also load MANCALA from cassette using this module..

Super Bug 2 by Edgar Dohmann is supplied with the module, and has a version to load into the module (filling 8k) to appear on the initial menu after the title screen. .

+++++++..

TI BASE : REVIEW OF VERSION ONE [Now into Vn 3, 1990].

Program: Sophisticated DATABASE. .

One disk, 34 page manual, function strip..

PUBLISHER: INSCEBOT. SUPPLIER: TENEX PRICE:US\$25.

.
MICROpendium has already published a lengthy and detailed review of this program, which I am sure everyone with a disk system has read - hmmm? So this will be a short overview and a handful of hints..

.
TI BASE is unique in the TI world in its "3D" capability: it is possible to use five databases simultaneously, and move data between them or create a report taking data from different databases. It can operate by means of a "command language" including the capability of an initial auto-loaded command file. .
The manual gives considerable technical detail on data base structure, and if you wish to do anything that the command language cannot handle, the information is there to help you out, so you can use a Basic or Machine Code utility of your own. A Command is available to make small patches in the machine code, to cover any minor revision. .

The manual describes the database as capable of:

up to 255 characters per FIELD..

up to 17 fields per RECORD..

up to 16129 records per DATABASE..

... subject to storage capacity on your storage medium!!!.

A database will be comprised of two storage files on disk: a structure file which describes the database, and a data file, which is in IF format, with pure data- no field separators, making it easy to interface to, provided you remember the difference between internal and display formats! A 255 byte disk sector will be used in accordance with the disk DSR for records up to 255 bytes- that is, you can have 12 records of 20 bytes in a single sector. For longer records the program carries out its own blocking..

Here are the commands you can use in a command file: APPEND BLANK, BOTTOM, DOCASE-BREAK-ENDCASE, CATALOG, CHANGE, CLEAR, CLOSE, COLOR, COPY, CREATE, DELETE, DISPLAY, DO, EDIT, IF-ELSE-ENDIF, WHILE-ENDWHILE, FIND, FORMAT, LOCAL, MODIFY, MOVE, PACK, PRINT, QUIT, READ, RECALL, RECOVER, REPLACE, RETURN, SCROLL, SELECT, SET, SORT, TOP, USE, WAIT, WRITE. The system variable eof is available for testing. Strings can be concatenated and trailing blanks trimmed. Maths is available- +,*,/,^,sqr,log,alog,sin,cos,tan,atan,<,>,<>,,=,~,NOT,AND,OR..

If like me you have not met the symbol ~, you may like to know that:

4 ~ 4 = 0

5 ~ 9 = -1

9 ~ 5 = 1

You are probably aware of boolean math, which in this database runs:

(9 = 9) = 1

(9 = 0) = 0

Also available are three "date" operators, DAY,MONTH,and YEAR. These are not explained at all in the manual! It is necessary to tell you that the database uses three types of data: Numbers, Characters and Dates. Wit the version I have, there is no apparent difference between Characters and Dates- both may be any length and contain anything!.

You can place a number or date into a character variable/field.

You can place a character string into a date field. .

You can place a date or character string into a numeric field ONLY if they only contain numbers..

The length of fields/variables has to be declared..

If you try to place more characters into a character string/date field or variable, your input will be truncated to the declared length - useful form of SEG\$ possible here!.

If you try to place a number into a field which is too small, all is lost. The field will display as an asterisk, and will have no function. Bear in mind that when declaring the size of a number, it is much better to overdeclare than to lose data. Although you are stuck with right justification, you can still move your number to/from character string variables and concatenate or trim on the string. There is a lot of power to play with in this program..

If your date variable/field is in the form MM/DD/YY that is, 02/28/88, or if the first eight characters of a character string are in this format, you may extract the individual elements- day, month and year, as follows:.

NUMBER=DAY(DATE) will set NUMBER to 28 in this example..

STRING=YEAR(DATE) will set string to 88 in this example..

This is quite a different database to use, and I did have a little trouble getting on top of it at first- notable hang ups for me were:.

Failing to notice that string handling must be carried out through an intermediate local variable. .

Failing to SET HEADING OFF when printing selected records!.

Undocumented features:.

SET is shown as used with =, which appears to be optional- see example below..

DELETE is indicated as marking a file as not in use until it is EITHER RECALLED or permanently lost by PACKing. In fact, if you SORT before PACKing, the delete marker is also removed, and the file is not permanently lost..

DATE on program startup can be any 8 characters.

NUMBER fields are RIGHT justified, and leading blanks cannot be stripped, so it will often be more convenient to use CHAR fields for numbers- unless you wish to use math..

CATALOG DSKn. will produce a different report with less information if there are any active local variables. The catalog disk command shows disk name, sectors used and free, and file names, in the order they were created, plus "file type" as a single digit, then number of sectors in file, and record length. The final two items are dropped if there is a local variable defined..

Make sure you make heavy use of the file copy command to back up files on your disk- it is remarkably easy to corrupt data/structure files!!! There IS a RECOVER command, which goes a long way to recovering data, but life is easier if you use pure copy files instead. Recovered files are probably better copied directly into a new pristine data base with an identical structure, using a command file, rather rely on the repair job to hold! .

Page Up and Page Down can also be carried on by using Function E and X. .

FIND is a single key search, from the top down to the bottom. It will find only the first occurrence- anything else you can do with the command language, possibly using a temporary intermediate database (five can be in use at once!). .

Assume data in sorted field is:.

- 1 IN .
- 2 INSTRUMENT.
- 3 INSTRUCT.
- 4 INSTRUMENTS.

then... FIND "IN" would locate record 1.

FIND "INSTRU" would locate record 2.

FIND "INSTRUM" would find record 2

FIND "INSTRUMENTS" would find record 4... and so on..

SORT is also on a single key, and works by creating an index- your records will retain record numbers in the order you actually keyed them in, and SORT OFF will restore the database to original record number order..

If you require to actually reallocate record numbers to sorted records, or to carry out a multiple key sort, then you can easily use the "SORT1 AND SPELL1" disk from the group library- this group has paid a licence fee to allow us to copy this disk for our members.

IBM users will be familiar with ECHO ON, and not be shy in using SET TALK ON..

The command files are rather like BATCH files, and can even be used to format a disk, catalog a disk, or copy or delete disk files!.

HINT: The function keys are described ONLY on the function strip. You may think it a good idea to note them also in the manual!.

.
INPUT screens are standard, but it is possible to create you own by using command files, and you can structure a database so that the user is hardly aware he is using TI BASE! You can READ an input from a screen location (similar to ACCEPT AT) and use this to choose some other command file to run or to select an event in a CASE structure. .

.
DISLIKES: .

It is not possible to print out database structure information. If you want a record of this information you must write it down. .

It is not possible to PRINT to disk. The PRINT command must go to a printer..

There is no way of picking up an error (to avoid built in error messages, and to suspend a possibly destructive sequence once things go awry), nor to test for the presence of a specific file on a disk. The disk catalog is to screen only..

.
Some examples of command files..

First, the basic data files I am using, looking like this when SORT is OFF and heading and record number are set ON:.

.
REC AUTHOR_SN INIT_FN TITLE.
0000 HEINLEIN ROBERT PUPPET MASTERS,THE.
0001 HEINLEIN ROBERT MOON IS A HARSH MISTRESS.
0002 HEINLEIN ROBERT A MENACE FROM EARTH, THE.
0003 HEINLEIN ROBERT DOOR INTO SUMMER, THE.
0001 HEINLEIN ROBERT STRANGER IN A STRANGE LAND.
.

Here is our first, simple, command file- parameters have been set, so that the command file can be run without making any assumptions, and the first comment line begins with an asterisk:.

.
* COMMAND TEST FILE A/C.
SET TALK OFF.
SET HEADING=OFF.
USE BOOKLIST .
* datafile booklist on default data disk.
CLEAR.
SORT ON TITLE
LOCAL TEMP C 60.
LOCAL IN C 2.
TOP.
WHILE .NOT. (EOF).
REPLACE TEMP WITH TRIM(INIT_FN) | " " | TRIM(AUTHOR_SN) | " " | TITLE.
PRINT TEMP.
READ 2,20,IN.
MOVE.
ENDWHILE.
WRITE 2,2,"END OF FILE".
CLOSE ALL.
RETURN.
.

and the resultant printout is:.

0003 ROBERT HEINLEIN DOOR INTO SUMMER, THE.
0002 ROBERT A HEINLEIN MENACE FROM EARTH, THE.
0001 ROBERT HEINLEIN MOON IS A HARSH MISTRESS.
0000 ROBERT HEINLEIN PUPPET MASTERS,THE.
0004 ROBERT HEINLEIN STRANGER IN A STRANGE LAND .
.

To illustrate the use of two databases at once, the following two examples show how, like basic, you can do lots of weird things, and still get the same result. Both examples below will produce exactly the same output as the first command file!.

We are using a second database, called TEMP, which has a single field in each record, called TEMP. Yes, you can use databases, variables, and fields, all with the same name, and this program can sort them all out!.

For these command files to work, we have previously CREATED the second database, and APPENDED a BLANK record to it for us to use..

To save a little space, the line [****] below has been used instead of that long line REPLACE TEMP WITH... from the above command file, but the actual command file would have to have the full original line!.

```
.
*COMMAND TEST FILE A/C                * COMMAND TEST FILE B/C
SET TALK OFF                          SET TALK OFF.
SET HEADING=OFF                       SET HEADING=OFF.
USE BOOKLIST                          USE BOOKLIST.
SORT ON TITLE                         SORT ON TITLE.
SELECT 2                               SELECT 2.
USE TEMP                              USE TEMP.
SELECT 1                               SELECT 1.
CLEAR                                  CLEAR.
LOCAL TEMP C 55                       LOCAL PUB C 55.
TOP                                    TOP.
WHILE .NOT. (EOF)                     WHILE .NOT. (EOF).
  [****]                               [****]
  REPLACE 2.TEMP WITH 0.TEMP          SELECT 2
  PRINT 2.TEMP                       REPLACE TEMP WITH PUB
* print field temp in slot 2          * first look for local TEMP, none, so
  MOVE                                * looks for field in current database
ENDWHILE                              PRINT TEMP
WRITE 2,2,"END OF FILE"              SELECT 1
CLOSE ALL                             MOVE
RETURN                                ENDFILE
                                      CLOSE ALL
                                      RETURN
.
```

When told to use TEMP, the program first looks for a local variable with that name - we can use 0.TEMP if we want it to look no farther! .

If there is no local variable called TEMP, and we have not specified 0.TEMP then it will look for a field called TEMP in the CURRENT database. We can also use a field in any data base that is in use by indicating which of the five slots it is in, without having to SELECT that slot, by modifying the name as follows: 2.TEMP etc..

We have to SELECT the database that we wish to MOVE on a record though. MOVE will only work on the current database - that is, the slot now in use..

This is but a brief discussion. If enough members purchase the program, and ASK ME, then more articles can follow, as this program is a beaut. Particular questions are also welcome..

Note the price- only US\$25, a mere \$5 more than the almost lamentable TOTAL FILER reviewed elsewhere. This program is a bargain. BUY a copy- don't pirate it. Let's keep goodies like this one coming in!.

VERY HIGHLY RECOMMENDED for almost any database user. .

SEPTMBER 1988:.

=====

TOTAL FILER : REVIEW.

Program by Warren Agee, Documentation by Chris Bobbitt..
Published by Asgard Software. Price about US\$20. One disk..
Written in c99. .
Effective documentation 8 pages..

-----.

Total Filer is a simple fixed-format data base. The documentation gives no technical details, so what follows is from experiment. The only technical information in the documentation is that the first field can contain 40 characters. It can actually contain 78..

This data base has a total of six fields. You CAN amend the field titles, but that is all. The first field can hold 78 characters, then fields 2, 3 and 4 can each hold 39 characters..

Field five is a special field, which holds "keywords". You can enter up to 78 characters, but once saved you can only view the first 39. All characters are active even though you cannot see them! More on keywords later..

Field six is a full screen- a little under 800 characters..

When entering data into a field with the capacity for more than one line of data, you have to press ENTER for each line- a bit of a pain if you don't want to put much into field six! Otherwise, text wraps around automatically..

This "database" will not allow you to amend the input screens nor to amend the output format (to screen or printer). There is no sorting. Just the "keyword" search. .

Imagine three records with these keyword fields:.

- 1: ONE THE CAT SAT ON.
- 2: TWO THE MAT WHILE.
- 3: THREE THE DOG HOWLED..

If you now search for the keyword THE, you will be given a list of all three records, but search on TWO and only record two is listed..

The keyword search also allows a "wildcard" character, the asterisk (the documentation refers to an astericks)(it looks like this:"*"). Thus we can search for *AT then both records 1 and 2 will be listed..

Each record is held in a separate disk file, thus there is a limit, imposed by the bit map method, of 127 records per disk, regardless of disk size. However, a single database may have records on more than one disk- you just have to search on every disk in sequence..

Thus TOTAL FILER is fairly unique in the TI world in permitting an endless database, and it also allows what appears to be the longest field (around 800 characters). I am not at all sure if this makes up for its fixed format, or its price. For only another five dollars, you can buy TI BASE.

The documentation dwells at some length on the intended use of TOTAL FILER, to keep track of articles in magazines or on disks, hence the preset field names, which are: Article Name, Author, Diskette, Miscellaneous, Keywords, and Abstract.

This is a software product you either want or not. If it does not suit your purpose, you can't alter it very much! It is fairly easy to use. .

Stephen Shaw

=====

END 9/88

RAMBLES. by Stephen Shaw..

.10 Alstone Road, STOCKPORT, Cheshire, ENGLAND, SK4 5AH..

I mentioned earlier this year a disk containing a **VAT ACCOUNTING MODULE**, an unreleased TI module, and indicated it had not been heard of. Correction. This module was listed as "forthcoming" in the FIRST issue of TIHOME (circulation 14) and was THEN not heard of again!.

At the Aston show, one member fought shy of obtaining a copy of Ray Kazmers Christmas Demo (free remember!) 'cos he had it. Oh no he did not - he had only the initial 1987 issue, which left little Woodstock crying for his present - the 1988 release has Woodstock retrieve the parcel and open it!

We presently have around 200 members, who as far as we can tell do not all have every TI product ever produced! so if you have any software or equipment not in use, why not offer it to your fellow members- or even to the Group! Drop the relevant librarian or myself a line indicating what you have and what you want. None of us have a lot of money these days, so please be reasonable on pricing! The group is not well funded and cannot afford to buy your equipment, but I am happy to stockpile it and I can at least repay you the postage cost!

=====

MACHINE CODE RAMBLES:

In an EXTENDED BASIC program is it possible for the program to sense that the **alpha-lock key** is depressed or not WITHOUT the user having to press a key?.

Yes- if you also have 32k ram. .

Here is the source code for a VERY short machine code routine:.

```

      DEF      ALPHA.
ALPHA  MOV     R12,@>FFFC      save old R!@.
      CLR     R12              9901 CRU Base = 0.
      SBZ    21                Signal alphaslock key line.
      TB     7                  Check alphaslock other side.
      JNE    STATE             jump if state = on.
      SETO   @>FFFE             state=off.
      JMP    JUMPA              As off skip next line.
STATE  CLR     @>FFFE             state=on.
JUMPA  SBO    21                stop sending to alphaslock key!.
      MOV     @>FFFC,R12        restore old R12.
* standard xb return now.
      SB     @>837C,@>837C      clear error for basic.
      B      @>0070             return to calling program.
      END    ALPHA.

```

To use this code, assemble it to a non-compacted object code, say DSK1.K/OB.

In Extended Basic, start your program with:.

10 ! K by M Gikow, Andover, MA.

11 ! August 1988. Use with K/OB.

12 CALL INIT :: CALL CLEAR :: CALL LOAD("DSK1.K/OB").

.....

YOUR PROGRAM HERE. THEN....

.....

now, an example of sensing the state of the alphaslock:.

see next page....

.....

```

100 GOSUB 4000 .
110 PRINT A.
120 GOTO 100.

.....
4000 CALL LINK("ALPHA").
4001 CALL PEEK(-1,A).
4002 RETURN.
5000 END.
.....

```

A word of further explanation may help... thanks to Mike Gikow for this:.

The keyboard is connected to the computer by a device (TMS 9901) that uses the Communications Register Unit (CRU) which is accessed by CRU instructions such as SBZ, SBO and TB (Set Bit Zero, Set Bit One, and Test Bit)..

The Technical Data Manual shows connections to the keyboard such that the alpha lock key is connected between the output for CRU bit 21 and the input for CRU bit 7..

When the switch is closed, the output can be read by the input..

This program tests the condition of the switch and writes a value of -1 or 0 to memory locations -1 AND -2. The CLR instruction produces the 0, and SET0 produces -1..

The call link statement in XB then tests memory location -1 (=>FFFE) for a value of 0 or -1. NOTE that the PEEK statement will set A=255 rather than -1! so you will see either a 0 or 255..

.
[At the time of submitting this article, I am not aware of prior publication of the above program, taken from a letter by Mike].

=====.

One heart felt cry, from more than one member, involves the printing of text files using TI Writer -

TI Writer NORMALLY sends a carriage return and line feed at the end of each line, and life is much easier if you switch the internal dip switches in your printer so that your printer does not itself add an automatic carriage return and line feed at the end of each line! If both the printer and TI Writer add a line feed, you end up with double spacing, whether you want it or no..

.
The TI RS232 card contains a number of software switches, and if you use RS232 as your printer name you will certainly know all about those! However PIO users generally have little call to use them and so remain unaware!.

.
You can instead of using just PIO, name your printer as PIO.CR or PIO.LF - if you use the Formatter, you may NEED to use PIO.LF even if PIO on its own is OK when printing from the Editor..

If you ever use a **Graphics program**, you will need to use **PIO.CR** to prevent an automatic carriage return every 80 characters- something that can make your graphics look a little untidy. .

In case of difficulty with line feeds, go through all the possible printer names and use the one that is best, be it PIO.CR or PIO.LF or possibly even PIO.CR.LF.

It makes life easier if you can switch the auto line feed off at the printer- consult your printer manual..

.
Another problem is that several members use printers which are not 100% compatible with the Epson control codes, and when printing text which has these embedded, you may meet problems, such as a printer freeze. Again, TI Writer has been written to take care of this problem. You can instruct TI Writer to print the text file but to strip out the control codes- that is to print only the "printable" characters ASCII 32 to 126- to do this, instead of using the output

device name of PIO you use C PIO - that is, a capital C followed by a space followed by the normal printer name..

.
And as a reminder, if you want TI Writer to save in DF80 format instead of DV80, select PF then type F DSK1.FILENAME - use the print file command instead of the usual save file, and add an F and a space in front of the output device name. You remember of course that TI Writer can load both DF80 and DV80 files! In fact it will even handle DF and DV files longer than 80, but only load the first 80 characters of each record. TI Writer is an unusually well written program!.

.
And remarks regarding TI Writer also hold true for Funlweb!.

=====

Reminder to **PR Base** users about the bugs in that database:.

1. Output Device Name:.

When you initialise a disk (any disk) the sectors are filled with >E5's..

When you select CREATE using PRBASE, the default output device name appears, IF you have nominated one previously. Otherwise you will see an apparently blank input field. Not so- what you cannot see is an input field full of >E5's, so if you type "PIO", the program will then try to use a device called PIO followed by several CHR\$(229)'s and tell you you don't have any such device!

This is mainly a problem when using PIO, as RS232 users usually make use of the cards software switches and put a full stop in there someplace- the card is so programmed that the excess >E5s are then ignored..

PIO users should type PIO and then fill the input field with blank spaces by holding space bar down! OR first use ERASE (FCTN 3) before entering the printer name.

.
2. Bill Warren is aware that some users with TI disk controllers have been unable to format a double sided disk using the PR Base utility. Their solution is simple- use another disk manager to format your disks! [I suspect that Bill's problem may have been with the standalone disk controllers which are for single sided use only?SS].

.
3. Mis-Alignment of columns in TAB reports:.

This time due to >00's.! When you type the header, do not use FCTN D to move to the right- it will leave >00's behind. You MUST type spaces with the space bar. Bill does not fill the screen with spaces, he used nuls!.

.
Note: Bill is NOT working on updates to PR Base. Version 2.0 is his last. However Mike Dodd has written a version 2.1 which makes normal use of sectors 0 and 1, but is incompatible with Vn 2.0 data files. Both versions are available from the disk library..

.

continued....

One disk library user had a problem using **HBMPRINT**, a utility to print the data from Home Budget Manager in various formats- it would only load with Editor Assembler. Now- this program was in Forth, which if you recall Versions 3.x of Funlweb, had to be flagged K=60 in the Funlweb load menu. That menu is no longer available, and it would appear that Funlweb is no longer able to produce that particular load environment..

No problem. Some time ago, the peculiar original TI Forth Load requirement was removed, and an amended format produced which would load as a standard Load and Run file- you can find it on the MiniMemory version of Forth in the disk library. All I had to do was swop the load routine and now HBMPRINT loads fine with Funlweb..

.
Some RAMBLES text can be found on GENie- refs 2676 and 2677!.

.
Stephen Shaw.

.
=====

.
REPORT ON ALTERNATIVE MICRO SHOW - ASTON (Near Birmingham) 12th Novmber 1988.

.
This show, held at the Aston Sports Centre, not far from Birmingham and easily reached by rail and motorway, was organised by EMSOFT, software publishers for the Tatung Einstein computer..

.
They had organised the Show for users of the Einstein, and in a gesture of solidarity, invited User Groups and Dealers connected with other orphan computers to participate - orphan computers do not have any coverage in magazines or other shows! Naturally, many of the 47 stands were connected with the Einstein computer, also many were generic, dealing with general computer supplies suitable for any computer - disks and paper and so on..

.
It was entirely appropriate for the TI99/4A to be represented, as the monitor supplied for the Tatung Einstein is the ONLY monitor that can be connected directly to the PAL TI99/4A, which has an unusual colour-difference video output..

.
The TI99/4A was represented by stands from TI99/4A Users Group (U.K.), by the East Anglia Region User Group, and by lone UK dealer Parco Electric. Parco had a good selection of modules available. The TI99/4A Users Group had a nice little robot, controlled by joystick, who was most polite and apologised nicely if he bumped into anything. His core program came from Ross Mudie and appeared in the September 1987 newsletter of the Sydney User Group (Australia).

.
Other micros represented were the MSX series, the Sinclair QL, the Lynx, the Enterprise (brain child of David Levy who had a hand in the TI99/4A Video Chess module), and the little Jupiter Ace, the only home computer to be sold with a language other than Basic on board- Forth 79. There was no User Group representation for the New Brain or the Memotech or the Dragon- many users and some dealers changed to the U.K. produced Dragon after dropping the TI99/4A..

.
One stand was taken by a general supplier who purchased odd orphan products, and TIUG was able to purchase a box of assorted modules at not a lot more than a couple of pounds each, including a five pack of the Adventure module..

.
The show was well attended and had good refreshment facilities, but never became too crowded. The TI99/4A owning visitors seemed to have mostly heard of the show from their own groups. The hoped for swelling of membership as a

result of the national publicity said to be provided by the organisers was not seen. Nevertheless, it was a rare meeting together of TI99/4A users, some of whom travelled very long distances indeed..

.
The organisers have booked the Bingley Hall, three miles from Stafford, for a two day show in 1989. Your reporter is not at all certain that the orphan groups and dealers can maintain a two day show, and the location is not known for its national shows- three miles from the nearest railway station and with bus services liable to cancellation instantly is not so good, and moving the show nearer the North - away from the more populous South East - is not usually a good thing for exhibitions. We shall see what happens next year!.

.
=====.
SECOND THOUGHTS ON **TI PLANNER (aka TI CALC)** from DATABIOTICS..

.
In putting together a long spreadsheet, a small period of slight panic was caused when the module filled the disk (any size disk!) and reported that it could not save the spread. .

Try loading.... no problem. Loaded... well, imperfectly... somewhat corrupted..

.
Many hours later, the precise nature of the problems comes to light, and now I am once more a very happy and satisfied user of this easy to use and reasonably fast spread sheet module..

.
First bug: Filling the storage medium! (Whatever its size)..

This is caused by placing anything on line W. Actually, if you only place something in cell W01 and leave the rest of line W blank, you merely have a corrupted spread! Anywhere else on line W and your storage medium will be filled and you will be told the save failed. .

In practice you may be able to load your data back in, delete all contents in line W, and carry on..

SO: Lesson 1: Never use Line W..

(It may be unrelated, but Control W is used to change the size of an EMPTY spread sheet. Pure coincidence. Is it ?)

.
Second bug: .

When writing a long basic program, you save it from time to time to protect yourself against lockouts! (After the first lock out you do anyway)..

If you do this with TI Planner, you could find yourself with a hopelessly corrupted spreadsheet unless defensive action is taken..

When saving to disk with 32k attached, the user may select from a spreadsheet measuring 50x50, 40x63, or 35 x 72..

.
If you start with a 50x50 spread, the FIRST time you save it, it is saved properly as a 50x50 spread. The next save will be 40x63, and the next will be 35x72. (Your screen will keep on telling you you have a 50x50 spread, it's only when you reload you find a different sheet size and corrupted formulae and possibly lost data.) .

If you start with a sheet measuring 35 x 72, the first save will be a correct 35 x 72, the second save will be 50x50 and the third save will be 40x63..

.
Examining the disk files, it is apparent that the first word is a flag which signals the sheet measurements. Apparently as you SAVE a sheet, an internal flag is incremented, and from then on your spreadsheet thinks it is a different size... UNLESS you reset the (register?) by reloading the sheet..

SO: Lesson 2: Everytime you save- to disk OR cassette, if you wish to do any more work on the sheet, immediately RELOAD it! .

.
These two defensive actions are now taken here, and no additional problems have been found. The spreadsheet works like a charm..

.
Further examining the files created by the module seems to show that EVERY time you save, the data is saved three times- once in each format, with the current actual format first. Upon reloading however, only the first format is read. This is especially noticeable when using cassette, as the save may occupy a count of 15 on your tape indicator, but the load will terminate on a count of five!.

.
Tape users may wish to note that 102 cells save to tape (in the UK) in about four minutes. .

.
Tape users without prior experience of using tape DATA files should also be aware that cassette data can be accepted by the computer in a corrupted form if the tape playback volume is slightly off- unlike the usual Basic programs, which will NOT corrupt without an error message.

.
Stephen Shaw. November 1988.

.
END 11/88 =====

SUBMITTED SEPTEMBER 1989

RAMBLES. Written for Oct 89 TI*MES by Stephen Shaw..

10 Alstone Road STOCKPORT Cheshire SK4 5AH.

.
The Annual Show, back in June, is now ancient history, but you may be interested in knowing that we had some 20 unexpanded visitors and 45 expanded visitors- given a membership of about half unexpanded, and some remarkable deals on modules (good modules, in quantity) the lack of unexpanded owners was rather a pity..

.
Jim Peterson sent over a box of disks for us to sell- duly opened by Customs, who rather oddly, took one disk out of its envelope, tore it, and put it back into its envelope? I cant see how Jim could have recorded on such a badly damaged disk, nor how a disk can be damaged and leave no mark on its tyvek envelope....

.
Texaments sent along over US\$400 worth of software for us to sell, but made the error of declaring a value of US\$100 on the Customs label. The package was opened for inspection and promptly seized. .

.
MICROpendium intended to send us some sample copies, but forgot. Laura mentions that they have **a mere 14 subscribers in the UK** (the majority recruited by me!), which is a very low figure even considering OUR membership number. This is a good regularly published and well established magazine covering all interests including basic only, and with timely news of new products. I have been recommending this magazine for four years now- with apparently little effect. .

.
Dolores of Harrison Software actually came along to be with us, all the way from Maryland (near Washington DC) and sold a few Music disks and a couple of word processors. Perhaps the members who purchased those would review them? Dolores has also submitted an article for TI*MES on her music programs- watch

out for it if and when space permits. .

Bringing over a couple of boxes of disks for sale, she had to pay VAT at the airport on the retail value, and was promised a refund on any unsold, on exit. On leaving the UK, she got a non-English speaking Indian customs officer, who took her receipt and was never seen again. When you have a Jumbo waiting for you you are not in a position to hang around! .

Dolores also suffered from other English bureaucratic bungling- while over here she wished to use our Public Record Office in London, and had a letter from them advising her that she could do so on production of her passport for id. When she got there they THEN insisted she produce a document from a University or Government Department vouching for the seriousness of her research (this is contrary to published information sold by the Government publisher HMSO). Does any member have access to the PRO please?.

Lory also wanted to visit the British Museum reading room, but they chose the day of her visit to put on show a vase smashed up some 145 years ago, and she was unable to get near the museum! She did manage to visit J C Bach's grave. .

Repeat- Lory would appreciate help with US/UK/German history around 1780/90, especially London, Hamburg and the Battle of Yorktown. Any lists of soldiers at the Battle? Any period music from any army, recorded or scored? Any access to Middlesex legal records of the period or details of legal practices especially costs? Activity of Catholics in the UK at this time, especially in London?.

Why another word processor? Bruce Harrison describes TI Writer (quite accurately) as a TEXT EDITOR. As he was used to using a Wang Word Processor, he felt the need for something a little more like that - the entire document is NOT stored in memory but is page-oriented and saved and loaded from disk in page sized chunks. It took him two years to produce the present product, and as of June 89 some two dozen copies had been sold, two in the UK - review please chaps! - Bruce says that it has sold on the ease of initial use. With a half page ad in Micropendium at US\$124, this little family firm are a little uneasy about getting back their expenditure... their wordprocessor has an interesting EMBED command, which allows you to type in-just like this:.

CHR\$(27)"A"CHR\$(8) or if you prefer just (27)"A"(8) and have the control characters for setting line spacing at 8 dots, placed in the document. They have also supplied conversion routines to and from TI Writer files, and can handle many standard Formatter commands- unlike TI Writer, their formatting is dealt with in normal printing rather than through a separate program..

Took time to look more closely at LISP99 (book review later if space) and it turns out to be a good implementation of MacLisp, just missing the Property List functions- also Quotient has the arguments reversed, and MINUS seems to be missed out! Otherwise you could say it was very like TI LOGO without the graphics!.

=====
1st August sees the 1988 Patent Design and Copyright Act becoming law- this is 12.50 from your local HMSO. It has an interesting clause- not only is making of unauthorised copies unlawful, but in addition it is unlawful to sell or provide a product (which could include a software disk) which is DESIGNED to make an unauthorised copy or which you have REASON to know that it will be used to make copies..

DM1000 and every other disk copier except the TI Disk Manager readily and easily ignore the simple copy protect features TI originally provided- however, TI provided an excellent track copier in their officially released TI FORTH, so mere track (and presumably sector) copying is probably lawful. The sellers of commercial copiers of protected programs (for other machines) are taking the view that they can continue to sell provided their product has a legitimate use and they do not advertise its unlawful possibilities. .

=====.

MINI-REVIEW: PagePro from Asgard:

This is a program which allows you to mix text and pictures on a page AND SEE THEM BOTH ON YOUR TV SCREEN together! There are a few limitations- the fonts and pics have their own format, but can be converted from TI Artist fonts and pics. The most significant limitation for me is that fonts MUST be either 8x12 or 16x24 pixels. .

Why is this a limitation? The standard 40 column screen used by TI Writer is based on characters 6x8 pixels. The standard 80 column printer also uses a standard character definition of 6 pixels width, so using PagePro you become limited to a maximum of 60 character from page edge to page edge- not too bad for a flyer or advert or if you have little to say, but a bit of a loss when you wish to cram a lot of information into a small space!.

Apart from that the program is remarkably easy to use. One other limitation- for best effect all characters should be centred in the 8x12 block, otherwise the printed effect can look- perfectly illustrated in the manual- like this:.

LOADI NG I NSTRUCTI ONS..

No, there are no spaces after the I's, it's just that in TI Artist format they are firmly placed at the extreme left of the 8x12 block, leaving rather a lot of nothing after them! It's up to you to reposition the characters before you translate them to PagePro format!.

PagePro has line wrap but not word wrap, perhaps making hyphenation a little easier! Import of DV80 text is possible WITH CARE!..

Not an expensive program at around US\$25 or so, and certainly the fastest in its field if you wish to mix text and graphics.

-----.

Speaking to Larry Connor I gather that all the extra goodies for TurboPasc 99 -eg Windows disk and Utilities disk giving access to graphics and sound- are due for release at this years Chicago show, and therefore should be available by our NEXT issue. All registered purchasers of TP99 will be advised. The reason for the delays is the difficulty of translating technical material from German to English, and at the same time retain the meaning and improve the readability. These should make TP99 into a very worthwhile language indeed. (UPDATE: Still not available June 1990)..

-----.

THERMAL TRANSFER RIBBONS or how to make your own T-Shirt Graphics:.

Some time ago, Personal Computer World Magazine gave the address of a California company who sold special printer ribbons which you could use to make your own transfers for your very own T-Shirts. In due course, TI supplier TEX COMP also made these available, but only in one package, whereby you had to buy a black ribbon with a set of coloring pens..

I have now located a UK source of these ribbons!.

For the Epson MX80/FX80 series the ribbons are eight pounds each including vat but exclusive of postage. They are fabric ribbons, but to produce a good image usage is limited to at most 100 images, and possibly as few as 20, depending on size and density..

BLACK ribbons are available as one-offs, but there is a minimum order of four-off for each of the coloured ribbons, which are inked to order. Colours available are red, brown, blue, green and purple..

.

To use, you print in the usual way onto a heavy smooth paper (70/100 gsm). The cloth should be 50/50 poly-cotton: pure cotton is unsuitable, more polyester may be better but causes difficulties when transferring the image as it scorches more easily..

Place foil on the ironing board, then your cloth, then the paper with the image, face down, then two sheets of plain paper to spread the heat. With your iron heated to "COTTON", heat the image for 30/40 seconds and the transfer is complete. Maximum image size for a normal domestic iron will be about 6"x6".

They also supply the heat transfer PEN set (you colour the PAPER you have printed on and then transfer the coloured image), and ribbons for many printers such as LX80, MX100, Citizen 120D, Panasonic 1080, Star NX10 etc..

Due to the minimum order for coloured ribbons I have only tried the black ribbon so far- this gives an initially brownish/black image but the brown washes out leaving a black image. Wash is "COOL" and ironing is not advisable but if required should be at lowest temperature and iron on back of image not face..

I have not had my ribbon long enough to comment on long term durability of the image, but my test piece has gone through one test wash without difficulty..

From the same supplier, further unusual items- if you have a printer which insists on tractor feed only, you may feel you cannot use lovely coloured or printed papers- not so. The choice of "continuous" paper is as follows:.

Parchment in Lavender, Natural and Turquoise (also packs available with seals and ribbon), Pastel colours-vanilla, orchid, pale blue, pale yellow, green; bright colours: yellow, red, blue, green, pink; also aqua, buff, ice blue, ivory and pearl gray to match printed papers below..

And printed paper(with pictures printed on) as bears going camping, bears holding hearts, bears with bow ties, butterflies, christmas geese, christmas morning, dinosaurs wearing sneakers, enchanted flower garden, holly border, jolly snowmen, lambs playing, seagulls soaring, biplane, lilies, mountainscape, ship, tropical beach. .

Matching envelopes available of course!.

Prices vary a little- for the printed papers you can buy as few as 50 for 8.95 or 250 for 16.95. 250 Pastel shades are 10.95..

Oh yes, the supplier....

MGA MICROSYSTEMS, Pear Tree, Appledore, Kent, TN26 2AR..

They do not take credit cards!

=====

Article by Bruce and Dolores Harrison for TI*MES:

Machine Code and Music:

NEW MUSIC FROM HARRISON SOFTWARE

Many people have done music programs on the TI-99/4A home computer, and many have done well at that, writing in Extended Basic. This story is about a new way of doing music programs in Assembly Language, and then making that music play as if it were Extended Basic. What we've done at Harrison Software has been a team effort between our resident musician and our resident assembly programmer. Neither could create this music alone.

The effort began a couple of years ago, when our musician (Dolores P. Werths) wanted to hear some music by Johann Christian Bach. That music, which is a set of six piano sonatas, is very rarely played, and was not available on records. She asked our resident assembly programmer (Bruce Harrison) for some help in getting the TI to play the music, so she could listen to it and study the compositions. After making some attempts in Extended Basic, which was really not satisfactory, the effort to do it all in assembly was started.

The team effort was absolutely necessary because the assembly programmer cannot read music, and the musician cannot deal with all the intricacies of assembly, including the unfriendly Hexadecimal number system. The programmer then made up a file of nothing but equates called the "NOTES" file. This file allows the musician to create music in her own quasi-musical notation, while the assembler takes care of converting the musician's input data into that mysterious Hexadecimal stuff. The notes file allows the musician to enter such statements as:

```
DATA D32  
BYTE V52
```

The musician means by this that a third-octave D is to be played by generator 2, and that volume level 5 is assigned to that generator. This concept, where all the available notes and volumes are entered as mnemonics, is still in use in all our music efforts, but we had another hurdle to overcome. That hurdle is the matter of timing the durations of the notes when playing. That was important because in the Baroque and Galante music, there are numerous musical ornaments, and these require very fast and precise timing to sound right.

In the TI Editor/Assembler book, there is a whole chapter devoted to sound, and the method recommended by TI was to create "sound lists" in main memory and to put these bytes in VDP Ram, then let the VDP produce the sound through the generators. Unfortunately, the timing of sound durations by the VDP is done using its vertical interval as the clock. This means that the timing is all done in 60ths of a second, and that proved to be a killing limitation for the ornaments in the music. In other words, one cannot properly execute 64th note triplets when the smallest duration for either a note or a rest is 1/60th of a second.

Fortunately, there was another method of doing things in assembly, which was mentioned in passing by TI in that giant book. That is the method of passing sound bytes directly to the sound generator chip at address Hex 8400. TI of course pointed out that when you do this, you must use your own timing loop to control the duration of the sound. This turned out to be the main ingredient of our success. We soon found out that our timing loop, even with purposeful time-wasting operations built in, would have to execute over 200 times for a 64th note at a Presto pace. This gave us the flexibility to execute all the ornaments flawlessly.

To make all that accessible to the musician, we created the concept of the "TEMPO" file. This too, like NOTES, was all equates, and required only one numeric entry to define the duration of the 64th note, then let the assembler derive the durations for 32nd, 16th, 8th, and so on, from that initial entry. Even the durations for triplet notes are entered as simple math expressions, which the assembler computes for us. Again, as with the notes, the durations were translated into easy mnemonics, so that the musician enters DATA Q for a quarter note, DATA E for an eighth, and so on.

continued....

That file of duration equates, which we call the TEMPO or simply T file, looks like this:

TEMP	EQU	250	(RANGE 4 THRU 2000)
SX	EQU	TEMP	(64TH NOTE)
T	EQU	TEMP*2	(32ND NOTE)
S	EQU	TEMP*4	(16TH NOTE)
E	EQU	TEMP*8	(8TH NOTE)
Q	EQU	TEMP*16	(QUARTER NOTE)
H	EQU	TEMP*32	(HALF NOTE)
TS	EQU	E/3	(TRIPLET 16TH)
TT	EQU	S/3	(TRIPLET 32ND)

Thus all the durations are derived from a single programmer entry. Notice that the range of possible values for TEMP is very broad, but values below 200 make the music very fast indeed, while values above 400 make for truly "dirge" playing.

Our resident musician is very careful about the markings given on the sheet music, and sometimes takes many attempts at the tempo number before she's satisfied that the music is being played "Allegro Assai" and not "Presto", or vice versa.

This musical integrity is one of the things that we insist on, and it's yet another reason that our music is a team effort. Many of the music programs we've seen by others have all kinds of superb graphics and special effects of one kind or another, but they are lacking the musician's touch, in that the rhythms are inaccurate, or the various ornaments are improperly executed. For us, that detracts from the enjoyment of the music. In all fairness, many of these are actually programmed in basic or extended basic, so not all of the musical faults can be laid at the programmers' feet. We know from our own experience that it's sometimes impossible to get "CALL SOUND" to do what you really want.

Of course there's more to the musician's touch than just timing. There's also the matter of dynamics and musical balance to consider. The dynamics are of course not anywhere near the available range of real musical instruments. The total range available on the TI sound generators is only 30 decibels, and because of the way we use volume settings, we are limited to only 28 decibels. Even so, we can produce dramatic changes in volume. Our resident musician is very careful in the use of large swings in volume. Rarely do the scores call for maximum Forte or minimum Piano.

For that matter, some of the sheet music we use gives no indication of dynamics at all. In some of those cases, we can listen to recordings of the music played by first-rate musicians, and try to emulate their style. In the Nutcracker Suite, for example, many cues were taken from a recording by the Philadelphia Orchestra under Eugene Ormandy. Of course there are many recordings of that work which could be used for guidance. In the cases of such works as Johann Christian Bach's Sonatas, Opus V and XVII, there are no recordings readily available, so the musician must study other works by that composer very carefully, to get the sense of how he would wish to have his music played.

Having a musician "on staff" here has made all the difference in the music we create. On occasion, she even finds mistakes in the printed scores, and with her knowledge of music is able to correct them. Sometimes the publisher or editor has misinterpreted an ornament indication, or even put in

a wrong note. Being able to detect and correct such errors makes our music sound better.

In much of the music we do here, the ornamentation is indicated simply by markings, and it is left to the performer to interpret the markings in a correct musical manner. Our musician has studied all this with great care, so that the actual playing of the ornaments is made smoothly and appropriate for its musical context. She often has to resort to working out the timings on graph paper, to make sure that the notes in the ornamentation work out properly timed with the longer duration notes in the bass line. One of the more challenging instances was in the Nutcracker Suite, where Tchaikovsky had a tied group of seven notes in the space of one quarter note. The timing for that had to be worked out on a grid where each square represented 1/28th note. Fortunately, once the timing had been worked out, the numbers could be expressed directly in the source file so that the assembler did the calculations of the actual durations. For example, if a note needs a duration of one seventh of a quarter note, we can simply enter Q/7 in a data statement, and let the assembler take the value assigned to Q and divide that by seven. Of course there is some "rounding error" in the calculation, since the assembler works only in integer numbers, but the effect is minimal because the numbers are large enough. The value for a quarter note is somewhere around 3800, so dividing by seven and leaving off the part beyond the decimal point is still well less than a one per cent error, and is not a noticeable difference when the music is playing.

That leads to another important point in the way we do our music. With our methodology we can make the computer play much faster than any human player could. We try very hard to keep within the "humanly possible" limits, including our interpretation of ornaments. If our musician feels that the computer is playing something that no human could match, she'll either slow down the tempo slightly or simplify the ornament so that it's within human capability. A good friend of ours who is a very advanced amateur flutist, and who also owns a TI, has been a "guinea pig" for much of our musical work. He provides some feedback on our musical aspects, and has often remarked on the "human" quality of our work, even though played by a machine.

Of course many of you still have some burning questions as to how this assembly language stuff gets to appear to the computer as if it were simple Extended Basic programs. This was not easy to accomplish the first time, but it's really not all that complicated.

In this example, we're showing what was done in one of our numbers from Anna Magdalena's Notebook, by Johann Sebastian Bach.

We use a bit of trickery which we call "submerging" assembly programs. First we assemble the program in the usual way with the E/A module. Next, we examine the first line of the object file, to discover the length of the program. That gives us a hexadecimal number, say for example >1DC4. We use that number to determine where the program needs to be loaded under an extended basic program. Extended Basic always loads XB programs in high memory, at a high enough address so that the area from >FFE7 through >FFFF is left open. What we then do is to subtract the length of the assembly program from >FFE6. For our example, >FFE6->1DC4 = >E222. We then go back to our source file and put in an Absolute Origin (AORG) directive at the beginning of the program, which we also insure is the entry point of that code. Now we re-assemble the program using that absolute origin. That's where the Loader will place the machine code in memory when we load this new object file.

Now the real tricky part begins. We take that address, subtract 16 from it to allow a small buffer zone in memory, giving us in this case >E212. We separate that number into its two bytes and convert each byte to a decimal number. Thus >E2 becomes 226 and >12 becomes 18.

Now we go into Extended Basic, do a CALL INIT, NEW, and then a CALL LOAD(-31952,226,18,226,18). This trick makes the Extended Basic interpreter behave as if the bottom of memory is already filled with program. Now we create a one-line program such as:

```
100 CALL LINK("AM7")
```

That one line program will serve as the link to our assembly code, but first we do the following in command (immediate) mode:

```
CALL LOAD("DSK1.ANMAG7")
```

The Extended basic loader now loads in the assembly object file ANMAG7 just after our 16 byte buffer into the higher locations in high memory. Now we simply do a save under another file name, and the machine code we just loaded in is saved to disk just as if it were part of that one-line program. All that would be fine except that the computer would not know how to execute that CALL LINK in line 100. That's where our second trick comes in.

In the LOAD program which puts the selection menu on the screen, we execute a CALL LOAD("DSK1.DEFAM"). That file DEFAM is a zero-length object file. That is, an assembly language program which contains no program as such. We use it merely as a vehicle to trick the loader into placing the addresses of all our 20 pieces into low memory at one whack. The source code for DEFAM looks like this:

```
DEF AM1,AM2,AM3,AM4
DEF AM5,AM6,AM7,AM8
(ETC), THEN
AM1 EQU >FCF4
AM2 EQU >FD4A
...
AM7 EQU >E222
```

And so on. Thus the absolute address for the assembly part of each piece is loaded along with its name in the lookup table, so that CALL LINK("AM7") causes the computer to branch directly to address >E222. That is why we insist that you run these music programs from their LOAD program, otherwise the computer will not know where to find them.

That leaves one more burning question to handle: How do we make the TI's sound generators behave like a piano, celesta, harp, and so on? Again the root of the process lies in the fact that we deal directly with the sound generators, and perform our own timing through a loop.

While that loop is executing, we can make it periodically reduce the volume on the generators that are active, and by tapering the time between these volume reductions, we can simulate the exponential decay of a struck piano string. We make different instruments by varying the overall time of the decay and the change of slope during that decay. The decay time is related to the tempo, so that in slower pieces the sustain is longer, while in faster pieces the faster decay prevents blurring of the sound. Of course the working out of the numbers for a new instrument effect takes some trial

and error, but we can now produce a pretty wide variety of such effects.

The one instrument we're still struggling with is the violin. That instrument family is characterized by a triangular waveshape which is difficult to simulate with the TI's square wave generators. Stay tuned, though, we're getting closer all the time.

The bottom line to all this is that we have created a new kind of music on the TI, and we're thrilled that other people enjoy it enough to buy our software. If you'd like more information, please feel free to write us at:

5705
MD 20781

=====
=====

BOOK REVIEWS..

Note on purchasing: All the books below can be ordered from ANY good book shop, including most branches of W H Smith, and all branches of their subsidiary Sherratt and Hughes (and Waterstones) as well as Dillons, Hatchards, Willshaw and independants. You will probably need to leave a deposit- the final price will depend on the price on delivery to the bookshop which is usually within three weeks of order. It may be worth ringing (free) The Book Company on 0800 010910 who supply many books post free by mail order..

THE ARMCHAIR UNIVERSE. A K DEWDNEY. W H Freeman and Co..

Paperback. 330 pages incl index. ISBN 0-7167-1939-8. About Pds 11.00.

This book has been cited in Rambles twice already- in association with Wa-Tor and with Fractals, so I thought it was time I bought a copy! It is based on a series of computer-related articles in Scientific American, and is full of interesting ideas on things to do with your computer. Section headings include Infinite Graphics, Mathemagadgets, Artificial Intelligence and Artificial Insanity, Life in Automata (imagine a game of 3d animated life...), Puzzles and Wordplay, Stimulation through Simulation, and Core Wars..

A good read for the computer literate, it also gives some ideas for programs to play with- with the emphasis on recreational programs rather than games or utilities. Very interesting reading.

LISP-A GENTLE INTRODUCTION TO SYMBOLIC COMPUTATION..

by David S Touretzky, Published by Harper and Row, 384 pages incl index.

Paperback. About Pds 15.00.

This is the book recommended for LISP99, and is based on MacLisp, as is LISP99 (available from your user group library!). An excellent introduction both to Lisp and to Lisp 99, easy to follow, apart from an awkward Chapter 0, and maybe a different approach to Chapters 1 and 2. Start with Chapter 3 with Lisp 99 loaded and ready! Only a chapter towards the end- Chapter 11 on Property Lists- is of no value with Lisp99..

Lots of exercises, WITH ANSWERS! All inputs to LISP99 are in ROUND BRACKETS, as shown in the book eg:.

(DEFUN DOUBLE (Y) (MULT 2)) defines function DOUBLE.

(DOUBLE 5) gives the output then of 10..

.
FRACTALS EVERYWHERE by Michael Barnsley. Hardback. 393 pages incl index..
Academic Press Inc. ISBN 0-12-079062-9. About Pds.27.00.

OK, this IS an expensive book, no doubt of that, but if your imagination is grabbed by fractal graphics, it is possibly the best value book you can buy, as it has lots of algorithms and IBM Basic listings, easily converted to Myarc XB, lots of sample graphics in black and white AND colour, and an appalling amount of pretty heavy going math, BUT presented in a pleasant chatty style which is not at all frightening! Lots of exercises but NO answers! My printer won't go anywhere near duplicating some of the math symbols used, whose names I do not know, but understanding the math can be left until after you have thoroughly explored the listings, there is so much to consider that even if only a little of the theory sinks in, you have greatly advanced your logical or mathematic thinking! Dont worry about things like "The random iteration algorithm is founded in ergodic theory"- it paints a pretty picture!.

.
THE SCIENCE OF FRACTAL IMAGES. Many authors! Pub. Springer-Verlag..
Hardback. 312 pages incl index. About PDS 23.00

This one has a foreword and a chapter by Mandelbrot himself, but also chapters by others extending well beyond the famous Mandelbrot set. This book is rather based upon the idea of looking at something and then determining a fractal approach to imaging it - not such light reading! but some very interesting colour plates. There are lots of algorithms presented as well, which appear to be in Pascal or a similar language. The math is much heavier than in Barnsley's book, with section headings such as "The spectral representation of random functions". Some interesting information in here though- watch out for fractal images in STAR TREK II next time you see it- watch for TINY triangles forming the image! Escher, the graphics artist, made use of fractals for some of his designs.... there is also a chapter on having an eye for fractals when taking photographs. This is a book for the more dedicated fractalist..

.
TI BASE BOOKS....

Well, not quite... but TI Base is a CLOSE approximation of the old database language called dBase II, and although most books are now on dBase IV plus, there remain a couple of books on dBase II around still..

At present TI Base lags behind dBase II in not having a report-form capability (understood to be in progress) and only limited screen set up commands- no form creating commands such as PICTURE..

The CHEAPEST book is:.

.
WORKING WITH dBASE II. M de Pace. BSP Professional Books..

Paperback. 175 pages incl index. ISBN 0-632-02305-8. About Pds 10..

Despite the differences between dBase II and TI Base, this book gives a useful introduction to creating and using databases in this type of environment. It is interesting to see that TI Base has more math operators, but fewer functions..

.
UNDERSTANDING dBase II. Alan Simpson. Sybex. 260 pages incl index..

Paperback. ISBN 0-89588-147-0. About Pds 22..

Quite a bit costlier, but I think its rather a better read, with clearer lay out and progression. This is my preferred book..

.
Incidentally- the TI 99/4A uses DSK1 where the rest of the world uses B:, so if you see USE B:MAIL, on the TI you would type USE DSK1.MAIL.

The disk library has a four disk set of TI Base tutorials by Smoley, with sample databases and command files, which is more relevant than any book, and a lot cheaper!.

.
There is a good Pascal book by Doug Cooper called OH PASCAL, which is appropriate to Pascal 99- I already have a good Pascal book so did not buy it, but I did buy a companion book-.

TEACHING INTRODUCTORY PROGRAMMING (with Oh Pascal 2nd Edition). Doug Cooper.. W W Norton and Co. ISBN 0-393-95447-1. Paperback. 173 pages. About Pds 5.00. -yes a computer book for a fiver!.

This book is intended for teachers using Oh Pascal as a text book, and while the chapter on Staff Meetings is utterly useless, what gives this book value is a series of Pascal test papers and homework assignments- about 107 pages of tests of one sort or another. There are NO answers but if you work your way through this lot, you should end up with a fairly solid idea of what Pascal is all about! .

.
COMPUTER SCIENCE LOGO STYLE. Three volumes. Brian Harvey. MIT PRESS. Paperback.. Volume One: Intermediate Programming. ISBN 0-262-58072-1.. 315 pages. About Pds 16.00.

The chapter on Logo versions lists no less than 18 versions, and the TI Logo gets no mention at all, which gives you an idea of how recent the book series is! Consequently, as the other Logo's have evolved away from the older TI Logo, some of the examples will not run without some alteration, if at all possible..

The series is an interesting up to date text on Logo though, well set out and eminently readable. Where there are major advances in Logo, perhaps the series may inspire a TI programmer to write a more up to date TI Logo in machine code?.
=====.

FRACTAL IMAGING PROGRAMS.

There are a number of ways of obtaining bit map graphic images on the TI99/4A, and while I have used Myarc XB for these programs, they should be readily adoptable to any other form of program that allows the use of the 192 x 256 pixel graphics screen, such as Triton XB, or JBM103 (available from the disk library, and requiring only TI XB and 32k), or the Australian graphics language G, or perhaps even a Geneve!.

The first program is a variation on a program I submitted in several formats for the last issue. This one uses two very slightly different forms of the basic function, and sets the variables at random. Scaling to the screen is automated and is based on the first two hundred points plotted..

.
The second program produces some interesting images, far more complex than the code may imply- a tree, a fern, a Serpinski triangle, and a black square. The square is of interest as depending on how many iterations you allow, that sets the density of the square (try it). Kicking in the random probability subroutine will demonstrate how to turn the black solid square into a set of patterns- one particular set of probabilities will give you a straight Serpinski Triangle! The routines for Q=5 and Q=6 are also squares, but the images are derived with (5) different base data and (6) preset different probabilities..

.
Instead of allowing random operation you can of course input data as you see fit. The CALL LINK("DUMP") is a machine code routine for Myarc XB which is available from the disk library which allows you to dump a Myarc XB bit map screen to an Epson printer via PIO..

```

1 ! MYARC XB FRACTAL PROG 1 .
2 ! EXPLORING THE MARTIN SET .
3 ! With alternate sub set .
4 ! from J de Rivaz .
5 ! Stephen Shaw UK 1989 .
20 ON ERROR 10000 .
40 RANDOMIZE .
60 CALL GRAPHICS(3) .
80 Q=INT(RND*2)+1 .
100 CALL WRITE(1,24,28,STR$( .
Q)) .
120 ON Q GOTO 1000,2000 .
1000 REM .
1020 A=(RND*5000-RND*5000)*R .
ND*RND :: B=15*RND*RND*RND : .
: C=(RND*5500-RND*5500)*RND* .
RND .
1040 CALL WRITE(1,2,3,STR$(A .
),2,9," "&STR$(B),2,18," "&S .
TR$(C)) .
1060 X,Y=0 .
1070 GOSUB 5000 .
1080 FOR L=1 TO 8000+INT(RND .
*5000)-(FLAG=1)*5000 .
1100 CALL POINT(1,INT(X*XS+X .
0),INT(Y*YS+Y0)) .
1120 XX=Y+SGN(X)*SQR(ABS(B*X .
-C)):: Y=A-X :: X=XX :: NEXT .
L :: GOSUB 12000 :: RUN .
2000 REM .
2020 A=(RND*4000-RND*4000)*R .
ND*RND :: B=15*RND*RND*RND : .
: C=(RND*3000-RND*3000)*RND* .
RND .
2040 X,Y=0 .
2050 GOSUB 6000 .
2060 FOR L=1 TO 8000+RND*600 .
0-(FLAG=1)*5000 .
2080 CALL POINT(1,INT(X*XS+X .
0),INT(Y*YS+Y0)) .
2100 XX=Y-SGN(X)*SQR(ABS(B*X .
-C)):: Y=A-X :: X=XX :: NEXT .
L :: GOSUB 12000 :: RUN .
5000 CALL WRITE(1,12,2,"one .
moment..."):: FOR L=1 TO 200 .
5010 GOSUB 9000 .
5020 XX=Y+SGN(X)*SQR(ABS(B*X .
-C)):: Y=A-X :: X=XX :: NEXT .
L :: GOSUB 9500 .
5030 CALL WRITE(1,12,2," .
"):: RETURN .
6000 FOR L=1 TO 200 .
6010 GOSUB 9000 .
6020 XX=Y-SGN(X)*SQR(ABS(B*X .
-C)):: Y=A-X :: X=XX :: NEXT .
L :: GOSUB 9500 :: RETURN .

```

```

9000 REM SUB SCALE .
9001 CALL WRITE(1,14,4,"WORK .
ING...") .
9002 MINX=MIN(MINX,X):: MAXX .
=MAX(MAXX,X):: MINY=MIN(MINY .
,Y):: MAXY=MAX(MAXY,Y) .
9004 SUMX=SUMX+X :: SUMY=SUM .
Y+X :: RETURN .
9500 AVX=SUMX/200 :: AVY=SUM .
Y/200 .
9505 SX=MAXX-MINX :: SY=MAXY .
-MINY .
9510 CALL WRITE(1,14,4," .
") .
9520 XS=(192/SX)*.5 :: YS=(2 .
56/SY)*.5 .
9521 IF XS+YS<.6 AND XS+YS>. .
35 THEN XS=XS*1.4 :: YS=YS*1 .
4 .
9525 X0=96-AVX*XS :: Y0=128- .
AVY*YS .
9526 IF XS+YS<.44 THEN XS=XS .
*3 :: YS=YS*3 :: FLAG=1 .
9600 RETURN .
10000 ON ERROR 10000 .
10001 RETURN NEXT .
10002 END .
12000 !ONLY IF JD/OBJ UTILIT .
Y IS LOADED .
12001 ! CALL LINK("DUMP",0,1 .
6) .
12002 OPEN #1:"PIO" :: PRINT .
#1:"A=";A:"B=";B:"C=";C:"XS .
=";XS;" YS=";YS:"X0=";X0;" .
Y0=";Y0: : : : : CLOSE # .
1 .
12003 RETURN .
20000 END .

```

```

1 ! MYARC XB FRACTAL PROG 2 .
2 ! SQUARE TRIANGLE TREE & .
3 ! FERN using Random Iter- .
4 ! ation Algorithm .
5 ! From FRACTALS EVERYWHERE .
6 ! by Michael Barnsley .
7 ! SPIRAL from FRACTAL .
8 ! REPORT 2, Keith Wood .
9 ! Q=8 by S Shaw. .
10 ! Stephen Shaw UK 1989 .
100 RANDOMIZE .
110 ON ERROR 780 .
120 CALL GRAPHICS(3) .
130 Q=INT(RND*8)+1 .
140 IF Q>4 AND Q<7 AND RND>..
5 THEN 130 .
150 ON Q GOTO 160,170,180,19.
0,200,210,220,230 .
160 RESTORE 240 :: GOTO 320 .
! TRIANGLE .
170 RESTORE 250 :: GOTO 320 .
180 RESTORE 260 :: GOTO 320 .
190 RESTORE 270 :: GOTO 320 .
200 RESTORE 280 :: GOTO 320 .
210 RESTORE 290 :: GOTO 320 .
220 RESTORE 300 :: GOTO 320 .
230 RESTORE 310 :: GOTO 320 .
240 DATA 0.5,0,0,0.5,1,1,0.3.
3,0.5,0,0,0.5,1,50,0.33,0.5,.
0,0,0.5,50,50,0.34,,,.
250 DATA .5,0,0,.5,1,1,.25,..
5,0,0,.5,50,1,.25,.5,0,0,.5,.
1,50,.25,.5,0,0,.5,50,50,.25.
260 DATA 0,0,0,.16,0,0,.01,..
85,.04,-.04,.85,0,1.6,.85,.2.
,-.26,.23,.22,0,1.6,.07,-.15.
,.28,.26,.24,0,0.44,.07,,.
270 DATA 0,0,0,.5,0,0,.05,.4.
2,-.42,.42,.42,0,.2,.4,.42,..
42,-.42,.42,0,.2,.4,.1,0,0,..
1,0,.2,.15,,.
280 DATA 0,.5,.5,0,0,0,.25,..
5,0,0,.5,0,.5,.25,.5,0,0,.5,.
.5,.5,.25,0,-.5,-.5,0,1,.5,..
25,,.
290 DATA .5,0,0,.5,1,1,.1,.5.
,0,0,.5,50,1,.2,.5,0,0,.5,1,.
50,.3,.5,0,0,.5,50,50,.4 ,,,.
300 DATA 0.846,-.308,.308,.8.
46,0,0,.9,-.163,-.163,.163,-.
.163,1,-1,.1,0,0,0,0,0,0,0.
,0,0,0,0,0,,.
310 DATA .45,0,0,.45,.55,0,..
25,.45,0,0,.45,0,.55,.25,.45.
,0,0,.45,.55,.55,.25,.45,0,0.
,.45,0,0,.25,,,.
320 FOR Z=1 TO 4+(Q=1) .
330 READ A(Z),B(Z),C(Z),D(Z).
,E(Z),F(Z),POSS .
340 P(Z)=P(Z-1)+POSS :: NEXT.
Z .

```

```

350 IF RND<.15 THEN GOSUB 59.
0
360 REM .
370 CALL WRITE(1,1,12,STR$(Q.
)) .
380 IF Q=6 THEN Q=2 .
390 X,Y=0 :: NUMITS=10000 .
400 FOR N=1 TO NUMITS .
410 REM .
420 REM .
430 K=RND+.00000001 .
450 IF K<P(2)THEN K=2 :: GOT.
O 480 .
460 IF K<P(3)THEN K=3 :: GOT.
O 480 .
470 K=4 .
480 REM .
490 REM .
500 NEWX=A(K)*X+B(K)*Y+E(K) .
510 NEWY=C(K)*X+D(K)*Y+F(K) .
520 X=NEWX :: Y=NEWY .
530 IF DX>390 OR DY>390 THEN.
RUN .
540 IF N<11 THEN 570 .
550 ON Q GOSUB 710,720,730,7.
40,750,750,760,770 .
560 CALL POINT(1,DY,DX) .
570 NEXT N .
580 CALL SOUND(200,200,1):: .
CALL LINK("DUMP",0,16):: RUN.
590 REM DIFFERENT PROBABILIT.
IES- RANDOM .
600 P(4)=1 .
610 IF Q=1 THEN P(3)=1 :: GO.
TO 640 .
620 P(3)=P(4)-RND*RND .
630 IF P(3)<.2 THEN 600 .
640 P(2)=P(3)-RND .
650 IF P(2)<.15 THEN 600 .
660 P(1)=P(2)-RND .
670 IF P(1)<0 THEN 640 .
680 CALL WRITE(1,24,8,"RANDO.
M PROBABILITY") .
690 RETURN .
700 END .
710 DX=X*2+40 :: DY=Y*1.92 ::
: RETURN .
720 DX=X*1.3+40 :: DY=Y*1.3+.
50 :: RETURN .
730 DX=X*40+120 :: DY=193-Y*.
19+5 :: RETURN .
740 DX=120-X*250 :: DY=200-Y.
*380 :: RETURN .
750 DX=X*150+60 :: DY=Y*150+.
30 :: RETURN .
760 DX=190-X*98 :: DY=80-75*.
Y :: RETURN .
770 DX=X*160+30 :: DY=Y*160+.
25 :: RETURN .
780 ON ERROR 780 :: RETURN N.
EXT .
790 END.

```

=====.
.G.

The Graphics language from Adelaide.
Available from the User Group Library..

.
These simple routines are taken from TIjdingen July 1989..
The headings are in Dutch, which I dont understand, but rather than omit credit
to anyone I will include them in full before the routine:.

Apart from routines using ARC which is not generally available, you should be
able to transfer these images to other languages making bit map or pseudo-bit
map graphics available such as Myarc XB, Triton XB, TI XB+JBM103, Logo etc etc..

.
1. 3D BOL : Bron MSX Computer Magazine 3e jaargang nr.16
Auteur:H.J.Smit,Amsterdam. Vertaald:Ronald van Kleunen Buys Ballotstraat 27,
4462 AM Goes. "Regio Zeeland".

```
FOR N=5 TO 70 STEP 5.  
ARC 190-N 100 N N 0 360.  
ARC 50+N 100 N N 0 360.  
NEXT N.  
STOP.
```

.
2. 3D-MSX : Heading as No 1 above..

```
:START.  
COLOR 7.  
FOR N=0 TO 7.  
A=N*5 ARC 128 85+A 85-A 85-A 0 360.  
NEXT N.  
FOR I=1 TO 600 NEXT I.  
CLS.  
COLOR 8.  
FOR N=0 TO 21.  
A=N*5 B=45-N*2 ARC 128 125-A B B 0 360.  
NEXT N.  
FOR I=1 TO 600 NEXT I.  
CLS GOTO :START.
```

.
3. ARTISTiek: Bron: KIJK Nr 10 okt 1987. Auteur:Gilbert Stokman Orgineel:MSX
Rest of heading as No1..

```
FOR X=0 TO 250 STEP 5.  
A=X/1 B=X+A SET X X TO B 180-X.  
SET 256-X X TO 256-B 180-X.  
C=X/2 SET 125 X TO 125-C 192.  
SET 125 X TO 125+C 192.  
NEXT X.  
STOP.
```

.
4. ATARI grafisch Bron KIJK No 8 aug 1986 Auteur Arvid Riem Lelystad .
Orgineel Atari 800XL (Kleuren verwijderd). Rest of heading as No1..

```
B=0.  
FOR L=1 TO 79.  
SET L+90 B TO 160-L B.  
SET L+90 B TO 160-L 190-B.  
SET L+90 190-B TO 160-L 190-B  
SET L+90 190-B TO 160-L B.  
B=B+6 IF B>190 THEN B=0.  
NEXT L.  
STOP.
```

```

.
5. ESCHER Driehoek van M C Escher Bron KIJK Nr 10 Oct 1988
Auteur Roy van Ginkel uit Zutphen Rest of header as Nr1..
SCREEN 5 COLOR 1.
SET 75 139 TO 92 150 TO 185 97 TO 185 72 TO 92 21 TO 75 30 TO 75 139.
SET 92 150 TO 92 62 TO 132 87 TO 110 100 TO 110 119 TO 185 72.
SET 132 87 TO 151 76 TO 151 76 TO 75 30.
SET 110 100 TO 110 74.
STOP.

```

```

.
6. DEMO1 Jan Knapen.
LINestyle 7 ARC 100 100 75 75 0 361.
LINestyle 5 ARC 100 100 50 90 140 501.
LINestyle 2 ARC 100 100 90 50 440 801.
LINestyle 4 ARC 100 100 60 60 0 361.
FOR I=1 TO 1000 NEXT I.
CLS.
SCREEN 15 FGND 4 BGND 9.
:NE.
REM :NE MUST BEGIN AT FAR LEFT MARGIN!.
FOR I=0 TO 4.
TRACE I SET 0 20*I+25 TO 255 20*I+25.
SET 20*I+25 0 TO 20*I+25 191.
NEXT I.
GOTO :NE

```

```

.
7. DEMO2 Jan Knapen.
SET 36 4 FOR I=0 TO 270 STEP 90.
ANGLE I DRAW 184 NEXT I.
ANGLE 0 SET 36 50 DRAW 184.
SET 36 96 DRAW 184 SET 36 142 DRAW 184.
ANGLE 90 SET 82 4 DRAW 184.
SET 128 4 DRAW 184 SET 174 4 DRAW 184.
FOR I=1 TO 5 PRINT 1 8*I "POS" NEXT I.
FOR I=15 TO 20 PRINT 1 8*I "NEG" NEXT I.
BOX 1 4 30 184 PATTERN 8440 FILL 9 12.
CLEAR 2 5 28 182.
PATTERN 8320 FILL 40 8 PATTERN 8320 FILL 88 10.
PATTERN 8320 FILL 136 12 PATTERN 8320 FILL 184 14.
PATTERN 8232 FILL 40 54 PATTERN 8240 FILL 86 54.
PATTERN 8248 FILL 132 54 PATTERN 8256 FILL 178 54.
PATTERN 8264 FILL 40 100 PATTERN 8272 FILL 86 100.
PATTERN 8280 FILL 132 100 PATTERN 8288 FILL 178 100.
PATTERN 8296 FILL 41 147 PATTERN 8304 FILL 86 146.
PATTERN 8312 FILL 132 146 PATTERN 8320 FILL 178 146.
INVERT 24 24 155 155.
STOP.

```

These demo routines also illustrate and amplify the rather limited documentation supplied with the G package!

```

.
=====
.

```


LOGO ROUTINES : with apologies to the original author whose name has become detached from the routines (sorry):.

TO DRAWIT.

TELL TURTLE HT CS CB 10 SC 13.

SXY -42 -42 PD.

LT 30 REPEAT 3 [LLL RT 60].

END.

.

TO LLL.

FD 15 RT 60 FD 100 RT 120 FD 85.

RT 120 FD 15 RT 60 FD 55 LT 120 FD 85 RT 120 FD 15 .

RT 60 FD 100.

END.

.=====.

.

TO TSTAR.

TELL TURTLE PD CB 4 SC 5 CS HT SXY -45 -47.

REPEAT 3 [BLOCK1 PU LT 30 BK 14 LT 90 FD 7 PD].

PU FD 22 RT 120 BK 15 PD.

REPEAT 3 [BLOCK2 PU LT 30 BK 18 LT 90 FD 6 PD].

PU FD 18 RT 120 BK 12 PD.

REPEAT 3 [BLOCK3 PU LT 30 BK 16 LT 90 FD 9 PD].

PAUSE 150 REPEAT 11 [STAR PAUSE 10 PR].

END.

.

TO STAR.

SXY -26 9 SH 0 FD 12 LT 60 FD 10 RT 120.

FD 9 LT 60 FD 10 RT 120 FD 11 LT 60 FD 10 RT 120.

FD 11 LT 60 FD 9 RT 120 FD 10 LT 60 FD 11 RT 120.

FD 10 LT 61 FD 11.

END.

.

TO BLOCK1.

REPEAT 2 [FD 21 LT 60 FD 16 LT 120].

RT 60 FD 16 LT 60 FD 21 LT 120 FD 16.

RT 60 FD 16 RT 120 FD 16 RT 60 FD 16.

END.

.

TO BLOCK2.

REPEAT 2 [FD 16 LT 60 FD 16 LT 120].

RT 60 FD 22 LT 60 FD 16 LT 120 FD 22 RT 60.

FD 16 RT 120 FD 22 RT 60 FD 16.

END.

.

TO BLOCK3 .

REPEAT 2 [FD 16 LT 60 FD 22 LT 120].

RT 60 FD 16 LT 60 FD 16 LT 120 FD 16.

RT 60 FD 22 RT 120 FD 16 RT 60 FD 22.

END.

.

=====

END SEPT 89

=====

This is yet another graphics program, in **Myarc XB**, which either draws flowers with varying numbers of petals, or it draws spirals. As it stands it selects at random, and will pause after approximately every 180 degrees for you to press space to continue or any other key to halt. After drawing terminates a star appears at screen bottom left. Press space to re-run program or any other key to print out (assuming you have loaded JD/OBJ from the disk library!)..

```
.
10  REM STEPHEN SHAW MYARC XB NOV 89.
100 XS=16 :: YS=21 :: Z=3.15
110 XOF=90 :: YOF=126
120 RANDOMIZE :: CALL GRAPHICS(3)
130 CALL POINT(0,XOF,YOF)
140 FLAG=0
150 ON ERROR 160 :: GOTO 170
160 ON ERROR 160 :: RETURN NEXT
170 MULT=INT(RND*32)+1
180 IF RND<0.3 THEN MULT=RND*12 :: IF MULT<1 THEN 180
190 IF MULT<15 THEN YS=YS/2 :: XS=XS/2 :: IF MULT<8 THEN YS=YS/3 :: XS=XS/3
200 IF MULT<4 THEN YS=YS/1.5 :: XS=XS/1.5
210 IF RND<0.3 THEN FLAG=1 :: MULT=0
220 TEMP=RND*0.12+0.01
230 FOR T=0 TO 6*PI+.05 STEP 0.025
240 IF T=Z OR T=2*Z OR T=3*Z OR T=4*Z OR T=5*Z THEN CALL PAUSE(P) :: IF P=20.
    THEN T=20
250 IF FLAG=0 THEN R=R+RR :: RR=SIN(T*MULT) ELSE R=TEMP+R
260 REM
270 IF RND<0.8 AND FLAG=1 THEN RR=SIN(MULT*T) ELSE IF FLAG=1 THEN RR=RR+TEMP
280 REM
290 CALL DRAWTO(1,SIN(T)*R*XS+XOF,COS(T)*R*YS+YOF)
300 NEXT T
310 REM
320 CALL WRITE(1,24,28,"*")
330 CALL KEY(0,A,B) :: IF B<1 THEN 330
340 IF A=32 THEN RUN ELSE CALL LINK("DUMP",0,16)
350 OPEN #1: "PIO"
360 PRINT #1: " ": "MULT=";MULT : "FLAG=";FLAG : "TEMP=";TEMP : " ": ""
370 CLOSE #1 :: RUN
380 STOP
390 SUB PAUSE(P)
400 CALL WRITE(1,24,4,"*")
410 CALL KEY(5,A,B)
420 CALL WRITE(1,24,4," ")
430 IF B<1 THEN 400
440 IF A=32 THEN P=0 ELSE P=20
450 SUBEND
460 END.
```

Nothing too grand really but if you want to draw spirals or flowers, thats how you do it!.

=====.

One thing I do find of great utility is to go back, again and again, over old magazines, books, programs, etc etc, as inevitably I will now find something of interest which I may not have done some years ago - and I may also rediscover something I found useful and have then forgotten. The demo program below is from an ancient copy of TIdings, published by the first UK user group TIHOME. Originally in TI BASIC, using a pseudo-hi res code by Peter Brooks, it is here presented in TI XB for use with the disk utility JBM103- it uses true bit map graphics. Easily transferred to Myarc XB for greater speed, although you can also use JBM103 in Myarc XB a little faster than in TI XB. Transferable to most other programs too.... DO NOT remove the rems in lines 10 and 20 until the program is properly running! The routine in lines 630-750 took over 30 minutes in the original TI Basic routine..

```

.
10 !ON ERROR 20 :: GOTO 100 .
20 !ON ERROR 20 :: RETURN NEXT .
30 GOTO 100 .
40 REM PAUSE .
50 CALL SOUND(994,200,30) .
60 CALL SOUND(50,200,30):: FOR T=1 TO 4 :: Y=SIN(4):: NEXT T .
70 CALL LINK("CLEAR"):: CALL LINK("SCR2") .
80 RETURN .
100 REM GRAPHICS PROGRAM IN EX BAS FOR JBM103 UTILITY .
110 REM LOAD JBM103 FIRST .
120 REM .
130 REM ROUTINES FROM Tidings, Feb 82, P BROOKS. .
140 REM In JBM103/XB by S Shaw Nov 89 .
150 CALL LOAD(-31890,56,0,"",-31964,56,0):: RANDOMIZE .
160 CALL CLEAR :: CALL LINK("CLEAR"):: CALL LINK("SCR2") .
161 GOTO 2000 .
170 REM SINE WAVE .
180 FOR C=1 TO 256 .
190 R=96+INT(0.5+SIN(C/16)*40):: CALL DOT(R,C):: NEXT C .
200 GOSUB 40 .
210 REM SIMULT SINE .
220 FOR C=64 TO 191 .
230 R=96+INT(0.5-SIN(C/16)*40) .
240 CALL DOT(R,C) .
250 R=96+INT(0.5+SIN(C/8)*40) .
260 CALL DOT(R,C):: NEXT C :: GOSUB 40 .
270 REM CIRCLES .
280 R=96 :: C=128 :: CALL DOT(R,C) .
290 V=4*ATN(1) .
300 FOR K=10 TO 40 STEP 10 :: FOR J=-V TO V STEP 1/K .
310 C=128+INT(0.5+K*SIN(J)) .
320 R=96+INT(0.5+K*COS(J)) .
330 CALL DOT(R,C):: NEXT J :: NEXT K :: GOSUB 40 .
340 REM ELLIPSES .
350 R=96 :: C=128 :: CALL DOT(R,C) .
360 V=4*ATN(1):: FOR K=5 TO 25 STEP 5 :: FOR J=-V TO V STEP 1/K .
370 C=128+INT(0.5+2*K*SIN(J)):: R=96+INT(0.5+K*COS(J)) .
380 CALL DOT(R,C):: NEXT J :: NEXT K :: GOSUB 40 .
390 REM NESTED ELLIPSES .
400 FOR K=1 TO 5 :: FOR N=1 TO 30 .
410 C=INT(128-(15+K^2.8)*COS(N/15*4*ATN(1))) .
420 R=INT(96+60*SIN(N/15*4*ATN(1))):: CALL DOT(R,C) .
430 NEXT N :: NEXT K :: GOSUB 40 .
440 REM 3D PLOT.

```

! CONTINUED.....

```

450 REM FUNCTION PLOTTED IS      IN LINE 490      .
460 REM ANGLE OF TILT IS IN     LINE 500         .
470 FOR X1=0 TO 48 :: SS=X1*X1 :: P1=SQR(2304-SS):: I1=-P1 .
480 R1=SQR(SS+I1*I1)/48         .
490 Q=(R1-1)*SIN(24*R1)! !!!!! This is function plotted .
500 YY=I1/3+Q*36 !!! TILT     .
510 IF I1=-P1 THEN M=YY :: GOTO 560             .
520 IF YY>M THEN M=YY :: GOTO 570             .
530 IF YY>=N THEN 600           .
560 N=YY                                   .
570 R=98-INT(YY):: C=128-X1 :: CALL DOT(R,C)   .
580 C=128+X1 :: CALL DOT(R,C)               .
600 I1=I1+4 :: IF I1<P1 THEN 480           .
610 NEXT X1 :: GOSUB 40                   .
620 REM                                     .
630 REM 3D PLOT/BANTHROP/PCW             .
640 H1=192 :: V=104 :: X1=H1/2 :: X2=X1*X1 :: Y1=V/2 :: Y2=V/4 .
650 FOR X5=0 TO X1 :: X4=X5*X5 :: M=-Y1     .
660 A=SQR(X2-X4)                         .
670 FOR I1=-A TO A STEP V/10             .
680 R1=SQR(X4+I1*I1)/X1                 .
690 F=(R1-1)*SIN(R1*12)! !!!!! Function plotted .
700 R=INT(I1/5+F*Y2)                   .
710 IF R<=M THEN 740                   .
720 M=R :: R=Y1-R :: C=X1-X5+32 :: CALL DOT(R,C) .
730 C=X1+X5+32 :: CALL DOT(R,C)         .
740 NEXT I1 :: NEXT X5 :: GOSUB 40       .
750 REM                                     .
760 REM RANDOM SYM PLOT1                 .
770 FOR A1=1 TO 3 :: FOR DOTS=1 TO 300   .
780 R=INT(RND*96)+49                   .
790 C=INT(RND*80)+113                  .
800 CALL DOT(R,C):: R=193-R :: CALL DOT(R,C):: C=257-C :: CALL DOT(R,C):: R=193- .
R :: CALL DOT(R,C)                     .
810 NEXT DOTS :: GOSUB 40 :: NEXT A1 :: GOSUB 40 .
820 REM                                     .
830 REM RANDOM SYM PLOT2                 .
840 FOR A1=1 TO 3 :: FOR DOTS=1 TO 150   .
850 R=INT(RND*96)+49 :: C=INT(RND*80)+113 :: CALL RSP(R,C) .
860 R1=INT(C/256*192):: C1=INT(R/192*256):: R=R1 :: C=C1 .
870 CALL RSP(R,C):: NEXT DOTS :: GOSUB 40 :: NEXT A1 :: GOSUB 40 .
880 REM                                     .
890 REM NESTED 8S                         .
900 V=4*ATN(1):: FOR K=5 TO 25 STEP 5 :: FOR J=-V TO V STEP 1/K .
910 C=128+INT(0.5+K*SIN(2*J)):: R=96+INT(0.5+2*K*COS(J)) .
920 CALL DOT(R,C):: NEXT J :: NEXT K :: GOSUB 40 .
930 REM NESTED DOUBLE LOOPS             .
940 V=4*ATN(1):: FOR K=4 TO 20 STEP 4 :: FOR J=-V TO V STEP 1/K .
950 C=128+INT(0.5+2*K*COS(J)):: R=96+2*INT(0.5+K*SIN(3*J)) .
960 CALL DOT(R,C):: NEXT J :: NEXT K :: GOSUB 40 .
970 REM                                     .
980 REM RANDOM EQUILATERAL      TRIANGLES .
990 FOR A1=1 TO 5                       .
1000 R=INT(RND*24)+84 :: C=INT(RND*32)+96 :: T=INT(RND*8)+4 .
1010 FOR J=1 TO T :: R=R+1 :: C=C-1 :: CALL DOT(R,C):: NEXT J .
1020 FOR J=1 TO 2*T :: C=C+1 :: CALL DOT(R,C):: NEXT J .
1030 FOR J=1 TO T :: R=R-1 :: C=C-1 :: CALL DOT(R,C):: NEXT J .
! CONTINUED.....

```

```

1040 NEXT A1 :: GOSUB 40
1050 REM
1060 RUN
30100 SUB DOT(R,C)
30110 CALL LINK("POINT",16,R,C):: SUBEND
31000 SUB RSP(R,C):: CALL DOT(R,C)
31010 R=193-R :: CALL DOT(R,C):: C=257-C :: CALL DOT(R,C):: R=193-R :: CALL DOT(
R,C):: C=257-C :: CALL DOT(R,C):: SUBEND
32000 END

```

You can extract individual routines, and experiment with different functions for the two 3d plots. Please write in with any pleasing functions you may find.. Stephen.

=====
RAMBLES. For TI*MES issue 26, Jan 1990. 1/90

Welcome to another issue of Rambles, and another new year, now more than ten years since our computer first saw the light of day, and still excellent new programs are being produced (eg TI ARTIST PLUS!)..

=====
BASIC....

To find a **SQUARE ROOT** you would use SQR, eg PRINT SQR(4) produces 2..

How about a **CUBE root** or higher?.

If you type: PRINT 4^0.5 you will get 2, and similarly if you type.

PRINT INT(8^0.333333+.00001) you will get 3..

To find an N root you need to use the form ROOT=NUM^(1/N), remembering to take account of small inaccuracies in the binary math!.

=====
Speaking of binary math, one aspect of Basic not exactly covered in your manuals is the use of **AND** in a compacted data type situation....

Lets say we have a user group whose members have different items of equipment. Lets allocate powers of two to each item of equipment:.

CONSOLE = 0.

EX BAS = 2.

32K = 4.

SSSD DISK= 8

DSSD DISK=16

PIO =32

MODEM =64 and so on and so on up to 32k.

Now add up the values of the items of equipment each member has, and obtain a total..

We can find out from that total- which occupies only 8 bytes! -by use of the AND operator, thus:.

INPUT "MEMBER TOTAL:":TOTAL.

IF TOTAL AND 2 THEN PRINT "MEMBER HAS EX BAS".

IF (TOTAL AND NOT 4)=TOTAL THEN PRINT "MEMBER DOES NOT HAVE 32K RAM".

Get the idea? Try it out!.

=====

The magazine ACE has nothing for the TI99/4a, but does have some game ideas, and reviews games which just might appear for the 4a... such as the Russian game TETRIS, which we have in our library as TITRIS in ExBas. Described in Ace as follows..."A fascinating geometrical oddity, this Russian puzzler turns the obscure mathematical topic of packing into a cult game. One at a time, shapes fall downwards into a rectangular playing area. Left to their own devices they'll pile up until they reach the top of the screen; your task is to guide them down and pack them tightly so that doesn't happen.... the brilliantly simple idea... means... well worth a look whatever your machine".(We in fact have two versions- TI Tris and Tetris, take your pick!)..

Also listed as "Classics" are Scrabble and Lurking Horror, both of which we have available on disk only- Lurking Horror requires a double sided drive. You must know Scrabble! Lurking Horror is an Infocom text only adventure (never released for the TI) and Ace writes "Infocom's tribute to H.P. Lovecraft and the horror-fantasy genre sends you into a cold sweat as you discover something very nasty lurking beneath your college laboratory. Superb text only game that defies you to play it after dark...".

=====.

.TEST 4

For those of you who like a little something to program, a slightly modified puzzle, taken from Ace Magazine....

How do you make five REFs FREE?.

eg REF + REF + REF + REF + REF = FREE ? When you replace each letter with a digit, so that it reads:.

251 + 251 + 251 + 251 + 251 = 1255..

Get the idea?.

Good. Now, what are the digits involved in order to make 18 REF's FREE?

Now something a little harder! There is another puzzle, involving a different number of REFs which still make FREE... how many REFs in this one!?.

Programs to solve this (with answers!) to Stephen please!.

===== .

.TEST 5

Like puzzles? OK here's another....

Make a circle of fifty apples say, and to make life easier let's say one of them is at the 12 O'clock position which we shall call position ONE for reference purposes!.

Starting at position one and counting clockwise, remove every tenth apple, continuing to count and remove until only TWO apples are left. What are there positions?.

Suppose we have 49 apples to start with, and do the same thing... what are the final two positions?.

Write a program to determine the answers to these two questions. Now try a generic program which allows you to input any number of apples and will yield the final two positions..

Thats test one! Which we can divide into two sections- fast in section one, and graphic with bells in section two!

Test two is to take a dozen different inputs to your generic program and see if you can determine any pattern which would allow you to speed up your program- so it doesn't actually have to go round and round removing the apples! IS there a pattern? If you find one, write it into a program and test it against the first program with another dozen different number of apples!.

Programs- for ANY section! - to Stephen please..

Submitted by an Australian member. Any UK members out there?).

=====.

.TI BASE.

Having spent five months indexing TI*MES, I then set about right justifying the page numbers by adding a zero in front of single digits, as follows:.

```
REPLACE PAGE WITH "01" ;FOR PAGE = "1".
```

I made the drastic mistake of not using a copy file, and the damned thing went and changed all pages 1,10,11,12,13...19 to 01. End of index to TI*MES for a little while...

Moral:.

Never assume any program acting on a file is going to work as you think it will- always use a copy file and retain your original file for some time!.

REMEMBER- when using TI Base, the comparison is with the first N letters only, where N is the length of the test string. THUS I should have used:.

```
REPLACE PAGE WITH "01" ;FOR (PAGE="1") .AND. (PAGE<"10").
```

.

=====.

MYARC EX BAS DIFFERENCE....

Well, hardly a bug, as it seems an improvement on TI Ex Bas handling....

I have been running and developing a domestic program in Myarc XB for quite some time. Running it in TI XB the other day I came across an error message I was not expecting. A little experimentation showed why....

In TI XB, when you type PRINT VAL("") you get an error..

In Myarc XB, when you type PRINT VAL("") you get 0, which is much better when you are dealing with disk data files....

=====.

Our 1989 AGM was attended by a rep of the Ottawa group. Know what he said about us- no new UK software. The UK is firmly identified as a TAKING country, giving little back to the world of 99ers. Is ANYONE programming ANYTHING in the UK? Some of our most famous programs are XB so dont be shy. Internationally Ron Johnson continues to be remembered for his Roman War Game program - though few UK users have looked at it. CARFAX ABBEY is another UK export. And of course the Billy Ball programs formerly from Stainless. All ExBas. Machine code is nice but not essential. Good ideas and lots of work are needed!.

=====.

Unless you caught Tomorrows World, you may have missed TI's latest innovation, as they paid POUNDS 2.50 PER COPY to advertise in a US magazine with a circulation of 200,000... yes folks, the first TALKING MAGAZINE ADVERT, with fifteen seconds of speech to advertise a talking chip... the advert is said to have repeated itself 650 times before the battery ran out- a little over two and a half hours of continuous synthesised speech! Sadly I could not find a copy of the magazine...

^^^^^^^^^^^^^^

CORRECTION: TI SORT..

In the last issue I made the assumption that as TI SORT was so very similar to SORT1 both in timing and specification, it was in some way derived from it. I have checked this out with TI SORT author Dennis Faherty, and I fully accept his claim to have written TI SORT entirely from scratch. The very close similarity to SORT1 is merely a coincidence of two good programmers having similar ideas! Apologies..

.TI SORT.

"The actual sort algorithms were based on a QUICK SORT method described by C A R Hoare in a book titled "Fundamentals of Data Structures" by Horowitz. The file handling methodology is a variation of creating multiple sorted files and merging them into the final output. The variation is that a single fixed file is used with sections of this file representing the multiple files. With the limitation imposed by TI on the number of files to have open concurrently, this seemed like a reasonable variation. It is also designed specifically for TI-BASE files, with other record types also being considered, to make it as

generically useful as possible. It was obvious even before TI-BASE was first released that it would probably need a supplemental sort program for data bases of any large size, and I had planned TI-Sort at that time."

.
Sorting TI BASE files with TI SORT is indeed a very easy and simple doddle, you don't really need the manual! The AID key is functional with help windows throughout, should you need them. Just one tiny niggle- when selecting the order of nesting, when sorting on more than one field, the order number is placed at screen left, and is not visible on my tv! Such screen echoing of input is not absolutely needed though. When sorting TI BASE data, TI SORT accesses the Structure file, and lists the fields for you to select from. And off it goes, making use of an intermediate file, rewriting the old file OR making a separate file if you prefer. .

.
As with SORT1, use of a RAM disk makes life a lot easier, else use of different disk drives speeds things up- eg intermediate file on a different drive to the input/output files..

TI SORT can cope-subject to your media limitations!- with up to 32767 records. ALL sizes of TI Base records are supported, but non-TI Base data is limited to 255 byte records or under. MAXIMUM 17 FIELDS PER RECORD. Nested sorts up to eight deep.

TI Sort will load using the standard Extended Basic LOAD from drive one, or using LOAD AND RUN from drive one, or using machinecode RUN memory image file from any drive..

On screen display of the sorting process is optional..

.
In addition to TI Base data TI SORT can handle the following other types of data- in every case you must create a structure for the data similar to that used by TI Base (and similarly by PRK). This just means you have to tell the program how many fields are in each record, of what type, and how long they are. Structure files so created may be saved for later use either with the same data or with other identically structured data. .

.
FIXED RECORDS- strictly speaking, only the actual fields you wish to sort on must be in fixed positions, but generally it is preferable if all data fields are of fixed length and position..

.
DELIMITED RECORDS- in which each field varies in length from record to record, and is merely separated from other fields by a special delimiter character, which can be any character you wish, so long as it is not included as part of a data item..

.
BASIC FILES- files created in a BASIC program, internal or display format. In this case embedded blanks may cause problems eg "THE RED SKY" as it cannot tell the difference between imbedded blanks and blanks between fields..

.
The program automatically (transparently) switches between three modes, depending on the amount of data- if it will all fit into available ram then the "scratch file" is not used, everything takes place in ram. For example 200 records of 80 bytes would be dealt with in ram..

.
After we run out of ram the use of sub files takes place, with a third mode available when you go over around 4000 records - but you really dont need to know all that at all!.

If you only do a little sorting, there is SORT1 from the library, with only the barest documentation. Or their is this very friendly program specially written for TI Base, called TI SORT, which I recommend to you.

=====.

TI BASE...to come....

Dennis Faherty advises that he will do everything in his power to get more string handling into Version 3, including something like SEG\$, and he is also considering the possibility of adding bit manipulation into the logical operators (AND and OR) to allow dense data packing..

(Dennis is the father of TI Artist author Chris, who first wrote TI Artist-version 1- when he was (gulp) a mere stripling of 14.).

=====.

.C H A O S.

There have been books on it, there have been TV programs on it and New Scientist has run articles on it... this is the idea that the fluttering of a butterfly's wing can have a causative relationship with a deadly hurricane on the other side of the world..

Impossible?....

Look at this really simple program:.

```
100 INPUT "START (0>1):":S.
110 INPUT "MULTIPLIER:" :M
120 FOR L=1 TO 20.
130 S=M*S*(1-S).
140 PRINT S,.
150 NEXT L.
160 END.
```

Everything in that little simple program is deterministic- there is no use of RND anywhere. We are only taking the progression 20 steps..

Now run the program with S=0.4 and M=2..

Notice anything. Any chaos here? No. Lets try again....

Now run the program with S=0.4 and M=3.25.

What now? Any chaos or is it still fairly simple?.

OK lets go the whole hog this time. Set S=0.4 and M=4.

Looking at the last print, can you guess what the next figure is going to be if we alter the loop size? You COULD work it out of course, but is there a pattern? .

There is a pattern actually. At M=2.8 we have a stable output after a while, at M=3.3 there is a stable pattern of two results, at M=3.5 there is a repeating pattern of 4 results- doubling all the time, and- here is chaos- as M grows larger, the difference in the size of M to change the size of the repeated pattern grows smaller very quickly. At M=5 you wont see the size of the pattern, but it grows a great deal if you then set M=5.0000000000000001 - the equivalent of a butterfly wing beat in an active atmosphere!.

=====.

.TI BASE TIP.

Want to make use of DATE? When you start TI Base, you have to enter a date in the format MM/DD/YY, and you have access to this as follows:.

-in a command file-.

```
LOCAL DATE C 8.
REPLACE DATE WITH .DATE..
DISPLAY DATE.
PRINT DATE.
WRITE 2,15 "Date is ",DATE.
RETURN.
```

=====.

INCIDENTALLY... if you do not find in TI*MES that for which you seek... perhaps it is because nobody knows you want that information? Drop me a line if there is a topic you want covered... as specific as possible! Actual coverage will depend to some extent on how many pages adequate coverage requires- we cannot print a 200 page tutorial in a 64 page magazine alas! - and the pressure on available space which requires every issue to have material left over!.

=====

A warning tale from down under of a large company with lots of expensive IBM and clone machines... and a staff turnover and training record that ensured nobody really knew how to use the things, most managing merely to use one or two standard programs, and then perhaps only just... faced with a task a little non-standard, and with a time deadline, can you guess the solution... yes, a TISHUG member took along his expanded TI99/4A and quickly and efficiently tackled the job required. .

Another Australian user provided computer support for a 5 day horse event, providing all manner of reports and printouts... .

=====

end sept 89

=====

DECEMBER 1989 ISSUE 27

Henon mapping

Here is a fractal program for Triton Super Extended Basic module owners.

```
100 REM TRITON SEB
110 REM BEFORE LOADING TYPE:
120 REM CALL FILES (2)
130 REM NEW
140 REM CALL INIT
150 REM CALL DRAWNPLOT
160 REM CALL LINK("GCLEAR")
170 REM AND OFF YOU GO ...
180 REM
190 REM S SHAW 9/89
200 REM FROM FRACTAL REPORT
210 REM ISSUE 4
220 REM HENON MAPS
230 REM ANDY LUNNESS, BURY
240 REM
250 REM
260 A=RND*4 ! experiment with
    h A from -4 to +4
270 COSA=COS(A):: SINA=SIN(A)
    ) !
(-0.1 to +0.8 is full map)
(step 0.2 is about right.)
(smaller steps take much
longer but give finer
detail)
(larger steps lose detail
fast)
280   FOR X=-.1 TO 0.8 STEP 0.
290   FOR Y=-.1 TO 0.8 STEP .2
300     SX=X
310     SY=Y
(500 loop required for full inner plot)
```

```

(larger loops seem to add little )
320 FOR IT=1 TO 500
330 IF SX>500 OR SY>500 OR S
X<-500 OR SY<-500 THEN IT=50
0 :: GOTO 400
340 XX=SX*COXA-(SY-SX*SX)*SI
NA
350 SY=SX*SINA+(SY-SX*SX)*CO
SA
360 SX=XX
370 PLTX=SX+91 :: PLTY=SY+91
380 CALL LINK("MOVE",PLTX,PL
TY)
390 CALL LINK("DRAW",PLTX,PL
TY)
400 NEXT IT :: NEXT Y :: NEX
T X
(after 15-30 mins ...)
410 CALL LINK("SHOW")
420 CALL LINK("GDUMP","PIO.C
R")
430 RUN

```

=====

The same thing in ordinary Myarc XB:

```

100 ON ERROR 110 :: GOTO 120
110 ON ERROR 110 :: RETURN N
EXT
120 STP=0.2 ! MAX DETAIL
130 LOOP=500 ! COMPLETE INNE
R LOOP
140 RANDOMIZE
150 A=RND*4-RND*2.5
160 COSA=COS(A):: SINA=SIN(A
)
170 CALL CLEAR :: CALL GRAPH
ICS(3)
180 FOR X=-.1 TO 0.8 STEP ST
P
190 FOR Y=-.1 TO .8 STEP STP
200 SX=X :: SY=Y
210 FOR IT=1 TO LOOP
220 IF SX>LOOP OR SY>LOOP OR
SX<-LOOP OR SY<-LOOP THEN 2
80
230 XX=SX*COXA-(SY-SX*SX)*SI
NA
240 SY=SX*SINA+(SY-SX*SX)*CO
SA :: SX=XX
250 PLTX=SX+91 :: PLTY=SY+91
260 CALL POINT(1,PLTX,PLTY)
270 NEXT IT
280 NEXT Y :: NEXT X
290 CALL LINK("DUMP",0,16)
300 OPEN #1:"PIO" :: PRINT #
1:"A=";A;" LOOP=";LOOP;"STP=
";STP:" ":" " :: CLOSE #1
310 RUN

```

==

You can use JBM103 to make interesting animated sequences, either vary the "step" in the loop to gradually lose or build up detail, or vary the value of A slightly, to slowly distort the image. This slow distortion is perhaps even more interesting than separate images! Try varying the STEP by about .005, or varying the value of A by .002 or .001 per frame.

=====

Random stars

I am not entirely happy with this coding, which I am sure can be improved to give more acceptable images than at present, but this one is a start. It yields similar images to the Rose program published a few issues ago. Some of the images will be unduly simplistic, some will not be at all nice. Some with plots partly off screen will be horribly distorted due to the way our error trapping is working G handles such things much better! (See later!).

This code is again in **Myarc XB**:

```
1 RANDOMIZE
2 ON ERROR 3 :: GOTO 100
3 ON ERROR 3 :: RETURN NEXT
4 TRACE
100 REM STAR FRACTALS
110 REM L J VERSCHUEREN
120 REM FRACTAL REPORT 4
130 REM MYARC XB S SHAW 89
150 CALL GRAPHICS(3)
160 N=INT(RND*8)+3 ! Number
of points
170 R=.5+RND*.5 ! Reduction
180 F=INT(RND*5)+1 ! "fracti
onation"
182 CALL WRITE(1,4,4,STR$(F)
)
190 FOR Z=1 TO INT(RND*8)+1
191 READ A !angle of rotatio
n in radians
192 NEXT Z
200 DATA .25,.5,.75,1,2,3,4,
5
201 RESTORE
210 IF INT(N/2)=N/2 AND INT(
A/2)<>A/2 THEN 182
220 C=1
230 IF N/2-INT(N/2)=0 THEN I
F A<>INT(A)THEN CALL GETC EL
SE IF A/2-INT(A/2)<>0 THEN C
=2
240 A=A*PI/N :: K=N-1
250 Y=RND*.5-RND*.5 ! "start
ing height"
260 REM S=SCALE
270 S=.7
280 X=-.5*S :: H=F-1
```

! continued....

```

290 CALL POINT(0,X+127,Y+127
)
291 CALL WRITE(1,24,4,"WAIT"
)
292 F=0
296 T1=C*N*K^H-1 :: LP=MIN(T
1,35):: IF LP<12 THEN RUN
297 GOTO 800
298 CALL WRITE(1,24,4,"
",4,3,"      ",5,3,"      ")
300 F,M,B,G,J=0
310 FOR I=0 TO C*N*K^H-1
320 M=I :: B=I*A :: G=0
330 IF M/K-INT(M/K)=0 AND G<
H THEN G=G+1 :: M=M-K :: GOT
O 330
340 J=H-G :: X=X+R^J*COS(B)
350 Y=Y+R^J*SIN(B)
360 CALL DRAWTO(F,Y*MULT+CO,
X*MULT+CO)
361 IF F=0 THEN F=1
370 NEXT I
380 REM
390 FOR I=1 TO 600 :: NEXT I
400 RUN
410 REM
430 C=A-INT(A):: IF C>0.5 TH
EN C=1-C
460 STOP
500 END
800 REM
810 FOR I=0 TO LP
811 CALL WRITE(1,5,5,STR$(I)
)
820 M=I :: B=I*A :: G=0
830 IF M/K-INT(M/K)=0 AND G<
H THEN G=G+1 :: M=M-K :: GOT
O 830
840 J=H-G :: X=X+R^J*COS(B)
850 Y=Y+R^J*SIN(B)
860 MINX=MIN(MINX,X):: MAXX=
MAX(MAXX,X):: MINY=MIN(MINY,
Y):: MAXY=MAX(MAXY,Y)
870 DIFX=ABS(MAXX-MINX):: DI
FY=ABS(MAXY-MINY):: SC=MAX(D
IFX,DIFY):: MULT=110/SC :: S
C1=MAX(MAXX,MAXY):: SC2=MIN(
MINX,MINY):: CO=190/ABS(SC1-
SC2)/1.3+20
873 NEXT I
876 X=A1 :: Y=A2
880 GOTO 298
900 SUB GETC(A,C)
901 C=A-INT(A):: IF C>0.5 TH
EN C=1-C
902 C=2/C
905 SUBEND
=====

```

=====

MARCH 1990

3/90

REPORT ON THIRD ALTERNATIVE MICRO SHOW, November 11th 1989..
In a field a few miles from Stafford....

The first AMS was in Birmingham, the second in London. This third show was in the Staffordshire County Show Ground, in a rather isolated location some miles from Stafford, with remarkably little signposting, and no public transport! . The advertising for this event was as usual minimal, but even more so this time, as route directions/map were not given.

Given the location, the organisers decided that it was proper for the stand footage rate to be quadrupled over their show in the centre of London. As a result of both cost, and remote location, many exhibitors at London did not exhibit in Stafford- including TIUGUK. The TI99/4A was represented by the East Anglia group, and by Database (Frances Parrish). Poor Frances- after setting off at 4am from his deep South home base, he gave up, packed up, and moved out at around 3pm, as trade was truly abysmal..

Our East Anglia friends booked one six foot table, and filled it with two expanded systems (you try it!!!), then got a second table - there was a lot of vacant space!- to display their spare modules, books and hardware for sale. Naturally their table - stand number 1 (ONE) - was the focus for TI99/4A owners who had a good chat..

Elsewhere, the show was quite remarkable for the variety of produce available. For the first time I met up with - and purchased- toasted corn (USA), fried yellow peas (India) and fried Cassava crisps (Costa Rica). I also bought a LW portable radio for 3.50 (binned 2022), which sells for 5.99 in Manchester- the catch with this being the 2.5mm headphone socket- not the usual 3.5mm one!.

I COULD have purchased some second hand BETA video recorders, an NTSC/PAL tv set, a LASER, some hankies, die cast cars, Christmas cards, biros, cuddly toys, a lovely solar driven music chip- like you get in greetings cards- think of it, whenever the sun shone, continuous music..... - good choice of portable cassette players, and much more besides! Even the usual offers of cheap disks and listing paper..

With stallholders generally cramped onto small tables in a huge hall with lots of spare room, I rather missed seeing which user groups had turned up apart from EAR. A most odd exhibition, which has moved on a long way from AMS1, which was organised by the Einstein users, who sought to join with other orphan computer user groups, to share a meeting which would otherwise be too expensive. The other user groups seems to have been rather pushed aside now, and I see that AMS4, Nov 90 in the same venue, is set to support also the PC - is it still an ALTERNATIVE Micro Show?.

Stephen Shaw. 13.11.89.

=====

HI RES - UNEXPANDED.

This is for our 50% unexpanded owners, and will allow you to sample many of the graphic routines I have (and will) present which really call for bit-map graphics. This routine allows you to plot in "pseudo" high resolution- you may program as though you had bit map graphics, but in fact what we are doing is continually redefining characters, which means the full screen area is not available, and it is a little slower than other means used! It is not possible to say when you will "run out of ink" as TI Logo, which uses the same idea, so aptly puts it. The routine is written to prevent crashes is you do run out of ink, but in itself will not check that R and C are within the screen boundaries. It is in Extended Basic and is by Gary Harding..

Where in the other programs we have:.

CALL DOT(1,R,C) you should use CALL PLOT(R,C,S).

CALL LINK("POINT",N,R,C) you should use CALL PLOT(R,C,S).

etc etc. .

.

Your program must commence:.

1 CALL SCREEN(2) :: S=31 :: CALL HCHAR(1,1,S,768).

2 FOR T=1 TO 14 :: CALL COLOR(1,16,2) :: NEXT T.

....

.

YOUR GRAPHICS PROGRAM THEN FOLLOWS. Ignore any initialisation for other graphics languages, such as CALL LOADS and CALL LINKS or CALL GRAPHICS

.

....

Now the subroutine which should be at the END of the program:.

31010 SUB PLOT(R,C,S).

31020 Y=INT(R/8+.875) :: X=INT(C/8+.875).

31030 H\$="0123456789ABCDEF".

31040 B=C-X*8+8 :: P=2*R-16*Y+16+(B<5)

31050 IF B>4 THEN B=B-4

31060 CALL GCHAR(Y,X,H)

31070 IF H>31 THEN 31100 ELSE IF S=143 THEN SUBEXIT.

31080 S=S+1 :: D\$=RPT\$("0000",4) :: CALL CHAR(S,D\$).

31090 CALL HCHAR(Y,X,S) :: H=S :: GOTO 31110.

31100 CALL CHARPAT(H,D\$).

31110 N=(POS(H\$,SEG\$(D\$,P,1),1)-1)OR(2^(4-B)).

31120 D\$=SEG\$(D\$,1,P-1)&SEG\$(H\$,N+1,1)&SEG\$(D\$,P+1,16-P).

31130 CALL CHAR(H,D\$) :: SUBEND.

31140 ! ROUTINE BY GARY HARDING

31150 ! FROM TIDINGS OCTOBER 1982

32000 END.

.

Any programs you see which use some form of DRAW TO or LINE are merely connecting up to spots on the screen, and you can probably work out a way of connecting them by means of a looped routing with several CALL PLOTs..

.

This program will just cease drawing when out of ink! You must reset S=31 to reuse the routine.

.

|||||||

Fractal graphics

THIS ARTICLE WAS PRINTED IN ISSUE 28, BUT WAS RADICALLY CURTAILED. THIS IS THE FULL TEXT.

Spherical shapes - planets - balls

The first set of programs generates planets or balls this enables you to create animated sequences just a little bit like the NASA pictures! Unlike the skeleton globes you often see generated by computer, frequently see-through wire frames only, these programs generate a true planet, and map all parts but only show what should be shown. Thus you may go into any orbit you wish, polar or equatorial or (with tiny adjustment) any other. You may approach or depart from the planet, and can set the speed of planetary revolution, and even decide whether the poles are to be vertical or horizontal. With a tiny tweak you can set the poles at an angle.

The global mapping algorithm is by Mark Datko, July 1989, and appeared in Issue 4 of Fractal Report, which is available by subscription only, for UK residents at £10 for six issues (irregular but about 6 issues a year) from Reeves Telecommunications Laboratories Ltd

The first listing is in **Myarc Extended Basic** and requires that language - a program for standard TI XB plus JBM103 (from disk library) follows.

(ON ERROR takes care of off-screen plots.)

```
100 !ON ERROR 110 :: GOTO 12
```

```
0
```

```
110 !ON ERROR 110 :: RETURN
```

```
NEXT
```

```
120 RANDOMIZE
```

```
130 CALL GRAPHICS(3)
```

```
140 REM SEEDS:
```

(k high, a low seems to give more detail.

Experiment!)

```
150 K=RND*0.5 :: A=RND*0.01
```

```
160 CALL WRITE(1,1,1," "&STR  
$(K)&" "&STR$(A))
```

```
170 REM DENSITY OF IMAGE:
```

```
180 NUMITS=42
```

```
190 REM VIEWPOINT:
```

```
200 MYLAT=RND*90
```

```
210 MYLONG=RND*90
```

```
220 CALL WRITE(1,2,1," "&STR  
$(MYLAT)&" "&STR$(MYLONG))
```

```
230 RADIUS=64
```

```
240 PIBY28=PI/28
```

```
250 PIPT8=PI+0.8
```

```
260 TWOPI=PI*2
```

```
270 TWOPI10=TWOPI/10
```

```
272 TWOPI10=PI*2/10
```

```
280 CONRAD=0.0174533
```

```
290 MYLAT=MYLAT*CONRAD
```

```
300 REM This sort of thing r  
educes processing time - hon  
est ...
```

```
310 SINMYLAT=SIN(MYLAT)
```



```

320 COSMYLAT=COS(MYLAT)
330 MYLONG=MYLONG*CONRAD
340 FOR H=1 TO 8
350 CALL DCOLOR(H+2,1)
360 FOR V=TWOPI10 TO TWOPI10
*8 STEP TWOPI10
370 X=H*3 :: Y=V
380 FOR I=1 TO NUMITS
390 X=X-K*SIN(Y)
400 Y=Y+X*(1-A*X)
410 IF Y>TWOPI THEN Y=Y-TWOP
I :: GOTO 410
420 IF Y<0 THEN Y=Y+TWOPI ::
GOTO 420
430 LAT=(X-14)*PIBY28
440 LONG=Y+PIPT8
450 COSLAT=COS(LAT)
460 SINLAT=SIN(LAT)
470 LONG=LONG-MYLONG
480 SINLONG=SIN(LONG)
490 CLCL=COS(LONG)*COSLAT
500 IF CLCL*COSMYLAT+SINLAT*
SINMYLAT<0 THEN CALL POINT(1
,26+RADIUS*(SINLONG*COSLAT+1
),22+RADIUS*(1+CLCL*SINMYLAT
-SINLAT*COSMYLAT)+30)
510 NEXT I
520 NEXT V
530 NEXT H
540 CALL WRITE(1,24,1," PRES
SPACE FOR ANOTHER")
550 CALL KEY(5,A,B):: IF NOT
B THEN 550
560 RUN

```

Now in TI Extended Basic using JBM103:

```

100 REM PLANET FOR JBMI03
110 REM requires jbml03 disk
from group library
120 CALL LOAD(-31890,56,0)
130 CALL LOAD(-31964,56,0)
140 CALL CLEAR
150 RANDOMIZE
160 K=RND*0.5
170 A=RND*0.01
180 NUMITS=30
190 MYLAT=RNDO
200 MYLONG=RNDO
210 RADIUS=60
220 PIBY28=PI/28
230 PIPT8=PI+0.8
240 TWOPI=PI+PI
250 TP10=TWOPI/10
260 CONRAD=0.0174533
270 MYLAT=MYLAT*CONRAD
280 SINMYLAT=SIN(MYLAT)

```

```

290 COSMYLAT=COS(MYLAT)
300 MYLONG=MYLONG*CONRAD
310 CALL LINK("CLEAR")
320 CALL LINK("SCR2")
325 CALL SCREEN(16)
330 FOR H=1 TO 8
340 COLR=H+2
350 FOR V=TP10 TO TP10*8 STE
P TP10
360 X=H*3 :: Y=V
370 FOR I=1 TO NUMITS
380 X=X-K*SIN(Y)
390 Y=Y+X*(1-A*X)
400 IF Y>TWOPI THEN Y=Y-TWOP
I :: GOTO 400
410 IF Y<0 THEN Y=Y+TWOPI ::
GOTO 410
420 LAT=(X-14)*PIBY28
430 LONG=Y+PIPT8
440 COSLAT=COS(LAT)
450 SINLAT=SIN(LAT)
460 LONG=LONG-MYLONG
470 SINLONG=SIN(LONG)
480 CLCL=COS(LONG)*COSLAT
490 IF CLCL*COSMYLAT+SINLAT*
SINMYLAT>=0 THEN 510
500 CALL LINK("POINT",COLR,2
6+RADIUS*(SINLONG*COSLAT+1),
50+RADIUS*(1+CLCL*SINMYLAT-S
INLAT*COSMYLAT))
510 NEXT I :: NEXT V :: NEXT
H
520 REM
530 REM SAVE PICS
540 PIC=PIC+1
550 PIC$="PIC"&STR$(PIC)&" P
"
560 S$="DSK2."&PIC$
570 CALL LINK("SAUVE",S$)
580 CALL LINK("SCR1")
590 GOTO 140
-----

```

To create a COMIC SHOW animation:

```

100 REM AUTO GENERATION OF A
PICTURE SEQUENCE FOR COMIC
SHOW v4.0
110 RAD@=26
120 REM FOR JBMI03
130 RANDOMIZE :: K=RND*0.5 :
: A=RND*0.01
140 REM requires jbm103 disk
from group library
150 CALL LOAD(-31890,56,0)
160 LAT@=135
170 LON@=5

```

```

180 CALL LOAD(-31964,56,0)
190 CALL CLEAR
200 RANDOMIZE
210 REM
220 REM
230 NUMITS=30
240 RAD@,RADIUS=RAD@*1.06
250 OFF=55-RADIUS/3
260 LAT@,MYLAT=LAT@+RADIUS/8
270 LON@,MYLONG=LON@+8+RADIU
S/20
280 PIBY28=PI/28
290 PIPT8=PI+0.8
300 TWOPI=PI+PI
310 TP10=TWOPI/10
320 CONRAD=0.0174533
330 MYLAT=IYLAT*CONRAD
340 SINMYLAT=SIN(MYLAT)
350 COSMYLAT=COS(MYLAT)
360 MYLONG=MYLONG*CONRAD
370 CALL LINK("CLEAR")
380 CALL LINK("SCR2")
390 CALL SCREEN(16)
400 FOR H=1 TO 8
410 COLR=H+2
420 FOR V=TP10 TO TP10*8 STE
P TP10
430 X=H*3 :: Y=V
440 FOR I=1 TO NUMITS
450 X=X-K*SIN(Y)
460 Y=Y+X*(1-A*X)
470 IF Y>TWOPI THEN Y=Y-TWOP
I :: GOTO 470
480 IF Y<0 THEN Y=Y+TWOPI ::
GOTO 480
490 LAT=(X-14)*PIBY28
500 LONG=Y+PIPT8
510 COSLAT=COS(LAT)
520 SINLAT=SIN(LAT)
530 LONG=LONG-MYLONG
540 SINLONG=SIN(LONG)
550 CLCL=COS(LONG)*COSLAT
560 IF CLCL*COSMYLAT+SINLAT*
SINMYLAT>=0 THEN 580
570 CALL LINK("POINT",COLR,0
FF+RADIUS*(SINLONG*COSLAT+1)
,OFF+RADIUS*(1+CLCL*SINMYLAT
-SINLAT*COSMYLAT))
580 NEXT I :: NEXT V :: NEXT
H
590 REM
600 REM SAVE PICS
610 PIC=PIC+1
620 PIC$="P"&STR$(PIC)&"_P"
630 S$="DSK2."&PIC$
640 CALL LINK("SAUVE",S$)
650 CALL LINK("SCR1")
660 GOTO 190

```

I ran the above program in Myarc XB, which can be used with JBM103, as follows:
Place on your ramdisk:
TIVDP from your Myarc XB disk, SCRO from the JBM103 disk, and the above program,
say PROGRAM.

Now type:

```
CALL LOAD("RD.TIVDP")::CALL LOAD("RD.SCRO") :: RUN "RD.PROGRAM"
```

The catch is if the program bombs and you are returned to GRAPHICS(1) mode, the screen is blank! You

must type NEW (and lose the program) to see text mode again!

Having generated a set of pictures you can then animate them using Comic Show v4.0 from the disk library. Here is a sample command file for use with that utility...

```
FP DSK2.P1_P
SC 31
SN DSK3.GREENBALL
MP
AP DS K2 .P1 P
KW 50
AP DSK2.P2_P
KW 3C
AP DSK2.P3_P
KW 3C
AP DSK2.P4_P
KW 3C
AP DSK2.P5_P
KW 3C
AP DSK2.P6_P
KW 3C
AP DSK2.P7_P
KW 3C
AP DSK2.P7_P
KW 3C
AP DSK2.P8_P
KW 3C
AP DSK2.P9_P
KW 3C
AP DSK2.P10_P
KW 3C
AP DSK2.P11_P
KW 65
AP DSK2.P11_P
AP DSK2Y1_P
GO
```

You end up with a memory image machine code program which will run from E/A 5 or equivalent.

.END 3/90

=====

.

BOOK REVIEWS.....

All of the following books are published by Interface Publications, and are available from: Interface Publications Ltd.,.

As at January when this review was written, stocks of the Scott Vincent books were limited (80 copies each) so do not delay if you would like them!.

DYNAMIC GAMES FOR YOUR TI99/4A by Scott Vincent..

I have already reviewed this a while back, but did not then know that it was still available! so a brief mention is appropriate. I used one of the programs in this book to demonstrate Turbo Pasc 99 a few issues back. 159 pages, 30 programs, all in TI Basic. Each program has brief comments. The programs are fairly simple but well done for all that, and the book provides a good source of ready to key in TI Basic programs, which can if required be decorated with fancier graphics or extra bits and pieces, or even used as inspiration when trying out a new language such as Turbo Pasc 99....

MAKING THE MOST OF YOUR TI99/4A. By Scott Vincent. June 1984. 256 pages (!!)..

An addition to the TI Basic manual, with lots more examples, this book commences, neatly, with two longish programs to key in! There are numerous programs, and quite a fair bit of interesting text, and it must rank as one of the better books on TI Basic. The comment that User Groups might purchase your programs (coupled with a lack of contact addresses!) is odd, but is just about all I can take issue with..

PROGRAM DESIGN: THE ART OF STRUCTURED PROGRAMMING by Peter Juliff. May 1984. 247pages. Peter Juliff is/was the Head of the Department of Computing at Victoria College, Australia..

Do not let the title put you off- this book is readable and relevant! It is worth the price for the cartoons- you can probably guess the illustration for file processing, but can you guess how to illustrate stacks? I have spent many happy hours debugging and rewriting programs written by other people, and also have long programs of my own which have been subject to continuous development, so by a process of need, I have become rather fond of the style of programming this book proposes. .

Not perhaps a book for the raw beginner, nevertheless, it is probably better read before you develop any odd programming practices- but never too late if you have! Some of the concepts are by their nature quite heavy, but the author manages to make it all so readable. If you don't get it all in one go, just go back and start again... IF you are interested in programming -especially commercially- do read this book. There are appendices covering Basic and Pascal too..

=====

MINI MEMORY MEANDERING MEMORIES.....

Back in Issue 6, we carried a bit of a glut of articles on Mini Memory usage in TI Basic, on Graphics, Speech and Sound. In later issues we put these together into a demo program, and also covered the use of VDP registers... .

After such a long break, with a few changes of owners, and possibly more mini memory owners than in 1984, it does not seem a bad idea to repeat a little now... In this issue a brief look at graphics.

Mini Memory adds a few commands to TI Basic, including the ability to read and write directly to **VDP ram** using **CALL PEEKV** and **CALL POKEV**. .

The VDP Ram is used to hold your TI Basic program, but also contains the screen contents, and colour and character definitions. There is also room for sprites (add that to the list of things to come!!!).

Here is the program, which will run in TI Basic provided you have the Mini Memory or Editor Assembler module installed. With only slight variation, you could use Extended Basic, but you will need to use the PEEKV/POKEV machine code utility we have already published, and you will need 32k ram to put it in. The extra ram is not required for mini mem or ed/as..

```
.
100 CALL CLEAR.
110 PRINT "    ]" .
120 PRINT "    ]]" .
130 PRINT "    ]]]" .
140 REM ] IS FCTN T.
150 CALL PEEKV(1152,A,B,C,D,E,F,G,H).
160 CALL POKEV(1232,A,B,C,D,E,F,G,H).
170 FOR T=1152 TO 1231.
180 CALL PEEKV(T,A,B,C,D,E,F,G,H).
190 CALL POKEV(1512,A,B,C,D,E,F,G,H).
200 NEXT T.
210 GOTO 170.
220 END.
```

TRY IT!!!.

Now the explanation. The character definitions are stored in vdp ram sequentially, each taking up 8 bytes. Character 48 is stored at addresses 1152 1153 1154 1155 1156 1157 1158 and 1159. Then from 1160 comes the definition for character 49, and after that 50 and so on. (Note to machine code programmers- all numbers used here are decimal!)..

If you wish you can replace lines 150 and 160 with these- the effect is exactly the same:.

```
150 CALL CHARPAT(48,A$).
160 CALL CHAR(58,A$).
```

As the definition of character 58 begins at memory address 1232..

Lines 100 to 130 place some character 93's on screen, and in the T loop we shall continually redefine this character, whose definition is held starting from vdp address 1512.

In order to achieve a smooth scrolling effect, it has been necessary to have a character defined as a number zero at both ends of the numbers- before 1 and after 9! This is what lines 150 and 160 do..

Within the T loop we are reading the definitions of the numbers, but not as normal, with CALL CHARPAT which looks at 8 byte blocks, but with direct addressing, which enables us to increment by one memory address at a time, for an interesting effect. Changing from number 1 to number 2 takes a 8 cycles through the loop- and as each cycle reads 8 bytes at a time, that means we are actually reading 64 bytes in order to go from number 1 on screen to number 2 on screen. And of course we are also performing a similar number of writes, making a total of 128 memory operations. All things considered the speed isnt too bad!.

=====.
RAMBLES.. by Stephen Shaw. March 1989.. for TI*MES #24.
10 Alstone Road STOCKPORT, Cheshire, SK4 5AH.

A note for international dissemination- the Chicago group have discovered that they have had real problems in the past year- their postal distributor was ill and mail was not forwarded (resulting in members being struck off the data base); they have changed their membership database (resulting in more members being dropped off), they have changed their BBS phone number; their "hotline" telephone number has been inactive; their Chairman has been very seriously ill; and they are onto their third membership secretary in little over a year, as one became bogged down in his paid work, and the other suffered severe health problems.... their brand new President Don Jones is very sorry and please would anyone offended please drop him a line now as matters are now in hand..

.
Anyone think we have problems now? Life for ANY user group is made very much easier the more members actually contribute!.

.
The Chicago group- possibly one of the largest "User" groups for any computer, even have a member in Tianjin, so now we can add China to the list of countries with a working orphan!.

.
The Chicago group have also listed the TI peripherals which did not make it (not in any quantity anyway!)... how many of you have heard of:

1. Modem CARD (Novation Cat 300 baud modem on a card)-no TE2 needed!
2. IEEE48 controller card..
3. VCR Controller..
4. Debugger card (code for which is in DEBUG for ED-AS)..
5. Grom library peripheral..
6. Hex bus controller (to connect battery operated RS232/pio, colour printer and wafertape drive)..
7. Special PEB interface card for the 99/8..
8. TIs own hard disk controller card..

and of course there were (and still are) a number of cards produced by third parties for all sorts of things. Now perhaps you can understand why the PEB has room for 8 cards???

.
Depending on how you look at the records the Chicago group has from 600 to 800 members!.

.
Following on from the product announcement in the last issue of TI*MES about the new wordprocessor, PRESS, the Lima (Ohio) group reporting on a get together organised by the Chicago group, saw a demo of an unfinished product- additional info is that the spell-checker is on THREE DISKS. Files are NOT saved in the usual Text style of DV80 format, but utilities are provided to convert between PRESS and TIWRITER formats. [Update: As at June 1990 no news of Press]...

.....
From the Hunter Valley in Australia, Tony McGovern advises of a "fundamental flaw" in the ROS for the HRD (=ram operating system and horizon ram disk respectively!). NO problem for you if you have all TI peripherals and stick to them... problems if you have the Myarc disk controller, the Diji AVPC card or an Australian made ram-board. Tony's comments refer to Version 7.3 ROS and earlier, and he has advised both interested parties of the difficulty. .

.
Liz Lawrence, age 11, has three favorite games: TI Runner, Jumpy, and Flooraway. One of these was written in the UK. Two of these are in machine code- the UK program is in pure and simple TI BASIC. Don't give up on Basic too fast! Liz has scored 34500 on Jumpy..

.
From earlier days....

```
100 CALL INIT.  
110 CALL LOAD(-31879,0).  
  
...  
200 CALL PEEK(-31879,A).  
210 PRINT "TIME=";A/50;"SECONDS".           [On NTSC consoles use /60]  
220 GOTO 220.
```

Requires minimem or ed/as plus 32k ram or XB plus 32k ram..
You can put code between lines 110 and 200, and time how long that code takes to process- maximum time is about 5 seconds then it goes back to zero..

.....
Got a DF80 machine code program that does not autostart, and need to know the name to make the !@#\$\$ program start? Well, the easy thing to do is load FUNLWEB and load the file with that- Funlweb then gives you a list of DEF names, and it is usually the first you choose. The harder way is:.

```
Using Editor Assembler OR Minimemory, load the DF80 file by:.  
CALL INIT.  
CALL LOAD("DSK1.FILENAME").  
(remember you MAY need more than one file to make the program work!)..
```

```
.  
Now, type:.  
CALL PEEK(16176,A,B,C,D,E,F).  
PRINT CHR$(A);CHR$(B);CHR$(C);CHR$(D);CHR$(E);CHR$(F).  
and there is the start name to use....
```

[from an anonymous bulletin board, dated 8th June 1987]

Extra Terrestrial Mail... just before **Christmas** I received a newsletter from the Chicago group. Not too odd you may think. Well, it WAS the **April** 30th issue.....

What do you mean you don't subscribe to MICROpendium department... and look what you are missing....

In the October 88 issue (I keep saying, I get mine by seaimail, ok!):.
Articles on: Basic (Using Trace); c99(arrays); Configuring Funlweb; The Geneve; Review of Bunyard Hardware Manual (recommended); and several mini reviews and lots of ads. .

In the November 88 issue....
Machine code- amending a CHARA1 file easily; c99(using filenames); Geneve, Review of TI Base Vn 2.0, micro reviews and lots of ads! news that NAVARONE lost all interest in their TI products as of Dec 1987 and will not consider any support for them (even if you bought them November 30th 1987!); .

What is the use of the Editor Assembler module if you have Funlweb... well you do get a nice machine code manual with it.... apart from that, you also get the source code for TOMBSTONE CITY on disk... and of course you can bung in 8k of ram and have yourself a SUPER SPACE module!!.

.


```
TTTTT  EEEE  SSS  TTTT
  T    E    S    T
  T   EE   SS   T
  T    E     S   T
  T   EEEE  SSS   T
```

My reader has written in asking if I can stimulate some interest by setting tests! Well, I'll try anything once... so, is there anyone living out there? Here is a little test to find out....

The object of this exercise is to stimulate a little active programming, and as it may have been some time since you last did any of this, we shall start off with something nice and easy..

No prizes. No closing date. As many entries as you like. Any language you like- as many languages as you like! In each issue the best response will be printed- and I am not entering, so if no one enters, you won't see any results, so there! Entries on disk or paper..

TEST #1:.

Here is an example first:.

Take a four digit number (3025).

Separate it into two halves, of two 2-digit numbers (30 and 25).

Add these two numbers (30 + 25 = 55).

Square the result (55 x 55 = 3025).

And we have our first number!!.

REQUIRED: A PROGRAM (in any language) which will print out (to screen) every four digit number for which this is true.

This is a programming test in which the problem is spelt out. Now you must put together the best problem solving algorithm you can and program it!!! If you find BASIC easy, send in your Basic entry then try to do it in another language- you have several to try: XB, Forth, Pilot, Lisp, Logo, Assembly, c99, TP99, what have you?!.
 ===== .

NEW USERS: .

Cassette troubles.....

- a. The remote control (the small 2.5mm plug) works on about half the worlds recorders- the other half use a different polarity, which matters as the switching is electronic, not mechanical. If you can switch the wires in the socket around it should work..
- b. 'cos electronic switching is used, there may be a voltage drop across the electronic device used, and if you are using batteries and the voltage is low, the drop may be enough to prevent proper operation. Mains power is highly recommended but it may still - rarely! - be necessary to leave the remote plug out..

+++++

REVIEW: JOYSTICK: KONIX NAVIGATOR.

14.99 each, inc post and packing from: Konix Products,

A few issues back I reviewed the Konix Speedking joystick (sold in the USA as the Epyx 500XJ). Konix, despite their name, are a little Welsh company who seem to have the joystick market well sown up- with millions of Speedking joysticks out there, their latest addition just had to be obtained!.

-----> continued

The Navigator Joystick is 99.9% perfect. The only question in my mind is why were all joysticks not made this way eight years ago? It is a real joy to use, and will quickly help you to double your previous high scores!.

(That .1% reduction is for the shiny surface of the grip by the fire trigger, the remainder of the unit is textured for a happier hold. It would have been nice to have a textured front face as well)..

Like the Speedking the Navigator uses microswitches and a steel shaft, so a long life seems secure! .

The Navigator joystick I have seems just a little more "positive" than the "Speedking" I have- a little more pressure is needed- but not a great deal.. You may recall I mentioned that the Speedking was for holding in the left hand only-as the fire button was on the right hand side.. The Navigator solves this problem by placing the fire button in the centre and by redesigning the shape of the body..

The Navigator joystick will sit easily between the thumb and first finger, with the thumb on one side and the other fingers on the other side, while the fire trigger falls neatly under the end of the first finger. The joystick can be used for prolonged periods without fatigue. It does not use up any desk space, as so many joysticks do, and this is a GOOD THING for me 'cos I don't have any desk space to plonk a giant joystick on! The Navigator is held in one hand while the other hand controls the stick..

This joystick has an "autofire" switch. I have seen these advertised and wondered what they do- it appears that they allow you to hold the fire button down, and send a train of fire pulses--- something which we ancient TI99/4A owners really have little use for, as off the top of my head I cannot think of any TI program which requires you to repeatedly press fire- we seem to have autofire built in. Yet again, TI gave us something noone else seems to have thought of! If you do switch the autofire on when using this joystick with the TI, it has no effect on most games, but on some games with rapid joystick scans, it can get in the way, allowing you only a small number of repeated shots in Parsec for instance (making it impossible to overheat the laser!). In general you would use it with autofire off..

The joystick has TWO D style 9 pin plugs. Use the BLACK one with your TI99/4A and also use one of the ATARI TO TI joystick converters..

This is an excellent product, very strongly recommended, and it's Welsh..

TI BASE UPDATE...

TI BASE is now into Version 2.00.[update: Vn 3 as of June 1990].

New V2 items/augmentations include:.

Sorting can be on up to 8 nested fields.

READ can be used with quoted strings and there is a new.

READSTRING..

There is a CONVERT to more easily amend database structures..

MEMORY gives details of memory usage..

EJECT is put in from dBase II..

SNAP is a screen dump command..

Printer output can now be directed to disk..

There is a limited file print command- to print command files..

TRACE to printer has been added..
There is slightly more memory available for your use..
Global conditional changes can be made..
Printer control codes can be incorporated and used in PRINT directives..
SUM command, which can also be conditional/global.

Thus a product which was already superb is even better. The new rather fatter 66 page manual has more examples of usage too. AND the price remains at US\$25,.

A fuller review of the revised database appears in the November 1988 issue of MICROpendium, and back issues are available to current subscribers. The final grade was A-, the minus coming from Ease of Use, as the database is designed for maximum power and flexibility - options which usually require some sacrifices in user friendliness. The reviewer concluded that "It is difficult if not impossible to measure TI Base against the competition since it doesn't have any competition".

=====
MACFLIX by J Peter Hoddie..

Published by Genial Computerware for US\$15 plus shipping..
[updated now to Macflix Pro and revised- original reviewed below].
Another graphics standard is born for the TI! How would you like to have a single graphics disk file to print out an A4 size picture in one pass, with a resolution of 700 pixels by 480 pixels (or thereabouts)?
MACFLIX is NOT a drawing program!!! It is a program which allows you to view and print high resolution pictures produced on the Mac using MacPaint, and also adds the ability to transform parts of the Mac picture to TI Artist format..
If you were to split a Mac picture up into bits for TI Artist and then print them using Artist options 1.3.8., you can transform an A4 portrait into (with some sticky tape) a fair sized poster some 33 odd inches long - naturally pixels are larger on this!.
What use is it? On its own, not a lot, unless you have access to Mac pictures! Which brings in a query: would you like the disk library to add MAC pictures, in addition to the RLE pictures? Let me know if interested. The MAC pictures I have at present are quite impressive..
- Correct, you cannot see a full Mac picture on a TI screen in one go! You have to window around the thing until you have the bit you want, then save that screen as a TI Artist pic, OR you can print the whole thing (Epson and Prowriter printer standards only).
- The maximum disk file size that this program can cope with is about 90 to 100 sectors, so some Mac pictures may not fit- the bottom parts of the pictures are lost on larger files..
- Geneve compatible, and can save in both My-Art formats too..

A good and inexpensive program..
=====

QUICK RUN written by Travis Watford (of RLE fame).. ASGARD SOFTWARE. .
Interesting utility this one! It "takes a snap shot" of a running ExBas program, and then you reload it in "running" mode- complete with prescan done, variables filled with values from data statements or external files, and so on.

It depends very much on the original program, but with SOME programs you can save considerable loading time by using this utility. With other programs, the extra loading time caused by the longer disk file and the different format, may result in a slightly longer loading time. .

continued....

You insert into your XB program a CALL LINK line which will snap the program when it reaches that point (must not be immediately after a CALL SOUND though, and a few other restrictions...)..

The instructions tell you to load the machine code part of the utility in immediate mode before you load your program, but in the first program I tried it out on, it was necessary to place the CALL LOAD immediately before the CALL LINK in the program itself, as it contained some non-relocatable machine code, which kept obliterating QUICK RUN if it was loaded as instructed!.

A public domain demonstration disk is available- from the Group library- with five demo programs saved with Quick Run. I only have two of these in the original Extended Basic, and these compare as follows:.

VENUS (music by Sam Moore):.

Time from RUN "DSK.." to first note:.

TI Ex Bas = 32 seconds.

Quick Run = 22 seconds..

Not a huge saving is it? But it is a saving..

AXEL F(Music)..

TI Ex Bas version = 91 seconds..

Quick Run version = 23 seconds..

Not bad eh? The ExBas program had to load data from two external disk files, which the Quick Run program did not need to do..

I tried the utility out on a particularly bothersome program, from way back when, written in TI Basic back in 1981, a version of FIDDLER ON THE ROOF with a very good animated graphic..

In order to run it in XB it was necessary to merge in the VDPUTIL which is on one of the disk library utility disks..

Running in XB, the original program, which had to do a prescan, load the machine code of VDPUTIL, and read data from a disk file, took all of 127 seconds!.

Running with QUICK RUN, this program is up and running in just 25.5 seconds! although it now occupies 92 disk sectors rather than the original 76 sectors..

As this is an even better demo than the ones Asgard have made available, you can obtain from the group disk library both the original and the snapped version of FIDDLER by asking for QUICK RUN DEMO DISK 2..

An especially nice thing about quick run- the RUN part is placed in the public domain so you can distribute programs saved with the utility without bothering about royalties or piracy. .

It is also possible to chain XB programs, just as before, but with the advantage of the rapid start, and as before you can specify a line number to run from..

An XB program saved and then loaded with QUICK RUN can be LISTED in the normal manner..

Potentially a good time saving utility. Nice one..

=====.

COLUMN ATTACK. By Chris Bobbitt. Written in Fortran 99.. ASGARD SOFTWARE.

Loads with usual EdAs/XB loaders..

You may recall a game featured in this magazine, called Spider Bop, which we gave you first in XB then in C99. Well this program is very similar but with less graphics, less playability and at least one bug!.

THE defined graphic is your spaceship at the base of the screen, shooting solid lines dropping from the top of the screen. They grow in several columns, at random, and you must shoot them all the way to the top of the screen to remove them. If they reach a lower part of the screen they start shooting at you - if you are underneath when they shoot you lose a ship. There are two

columns on my version which CANNOT be destroyed once they reach the screen bottom, and the only exit is suicide!

As an introduction to the capabilities of Fortran this is not a good product- it is similar in many ways to 99VADERS written in TI Basic by Not Polyoptics, but that had excellent graphics, and high playability, which the slightly faster speed of this game does not replace..

I have wiped my version of this program, I dont think it is worth cluttering up my disk collection even as a novelty item. It isn't novel!

Not recommended..

=====.

TI ARTISTS COMPANIONS..

No 1 (5 disks) US\$18.

No 2 to 9 (2 disks each) US\$10 per set.. TEXAMENTS..

These disks are filled with fonts and instances for TI ARTIST. Altogether over 100 fonts! and more than 400 instances! (mostly small but some larger).. Quite a lot of artwork ready for you to use - even if you are unlikely ever to use half of it! How do you review fonts? There are lots here of all sorts of shapes and sizes. .

It is perhaps worth mentioning that the documentation for Companion #7 contains information not exactly relevant to the disks that go with it- instances are illustrated which do not appear on the disk and vice versa, and the file list differs from the disk contents. This does not reduce the value of the disks in any way..

TI Artist itself is \$20 from the same source..

=====.

SECTOR SHARING..

By Mark Shafer. Bluegrass 99 Computer Society Inc...

Retyped by S Shaw..

[This is a complex but unusual article, addressing an odd problem in an odd way. Lateral thinking isn't in it.... sjs].

They say necessity is the mother of invention. And in this case, I'm the father. I think I've discovered something you'll find intriguing. Take a look at the following disk catalogue:.

DSK1 - DISKNAME= FNWEB/4*1.

AVAILABLE=117 USED= 241.

FILENAME SIZE TYPE P.

AS 33 PROGRAM.

AT 22 PROGRAM.

CF 31 PROGRAM.

CG 25 PROGRAM.

D1 33 PROGRAM.

D2 33 PROGRAM.

D3 29 PROGRAM.

DU 33 PROGRAM.

DV 33 PROGRAM.

DW 29 PROGRAM.

LOAD 31 PROGRAM

If you don't notice anything strange, add up the sizes of the files and

compare that to the number of sectors used. How did I do that? Why did I do that? That's what I'm here to tell you..

What I've done is to make it so that some files take up the same space as other files, which is the concept I call sector-sharing..

First, let's get into why I did it. I have Disk Utilities by John Birdwell. One of its features is the ability to change to default system setup. The trouble was sometimes I will want the defaults to be one way and sometimes I will want them another.

Now, I could change the setup in the program when I need to, but this is some trouble. The ideal solution would be to have two (or more) copies of the program on the disk, and boot the one with the defaults I want at the time. But I only have one SSSD disk drive, so I clearly don't have the room to do this if I want to add to the above disk the Funlweb files CHARA1, EA,ED, EE,QD, SL,SYSCON and UL!

Just like limited memory can lead to tight coding, limited disk space can lead to creative disk utilisation..

All I wanted to do was to change the first sector. So I got the idea to create a file that would have a different first sector, but share the rest of the sectors with the original file!.

The steps to do this, I believe, can be done in any order. Basically it goes like this: creating the new header sectors, creating the modified sector, updating the disk catalog, marking the used sectors, and renaming the new files. The beauty is that Disk Utilities itself can handle all of the above in one session, but I suppose any sector editor and disk manager will do..

So let's create the new header sectors first. Each file on a disk has to have a sector that identifies the type of file it is and where it is on the disk. The first step here is to find out what sectors are free. One way to do this is to look at sector 0 starting at byte >38, look for non-F's, and figure out what sectors correspond to the blank bits. Or you could use Disk Utilities to print the disk report and figure out what sectors are contained in no file. In my case, sectors >13 to >16 were available. You could put them anywhere, but the normal thing is to put header sectors in the >02 to >21 range..

Next I need to know where on the disk the files I'm going to "copy" are, as well as where their header sectors are. The disk report has this information. So now you edit the header sector of these files. So to the first one first. You need to change two things on it. Change the name to something that would fall at the end of the disk catalog. This way we don't have to insert when we change sector >1. I called my new files ZX,ZY, and ZZ.

For the file that has the modified sector, you need to change the segments starting at byte >1C. Insert three bytes at this point. This may be a little more difficult with some sector editors. Put in the following three bytes at >1C: yz 0x 00 where xyz is the sector we're going to create in step 2. In my case it was >16 so I inserted 16 00 00. Then add one to the next byte, so if it's >57, make it >58..

This process makes it so this file is in the same place as the original file except its first sector is different. If you're changing a sector in the middle, this is a bit more difficult. When you save it back, put it at the firstavailable sector you found. For the remaining header sectors, I just simplechanged their names and saved them to the next available sectors since they are to share exactly the same sectors..

Once you've got that done, the rest of it is a cinch. To create the modified sector, simply edit the sector you wish to change, make the appropriate changes, and save it to the free sector you indicated at step one (eg >16 in my case). Normally the sectors contained in a file are higher than >21 but I didn't have any free in that area..

Next it's time to change the disk catalog at sector 1. Simply put the header sectors you created at step one at the first available 0000 in sector one. I appended 0013 0014 0015 to add my three new files..

The next step is to tell the disk what sectors we've used. With Disk Utilities, you just use the Mark Sector feature. With others, you may have to figure out what bits they correspond to in sector 0 and make the changes yourself..

The last step is to rename the new files what you really want to call them. I called mine D1,D2, and D3. The last bytes have to be consecutive so that they load as one continuous program. After this, I had to configure Funlweb to be able to load my new program. So now, when I run Disk Utilities, I have the choice of the options in DU or the options in D1. They both load just fine..

But there are some consequences. There's the problem of copying. If you try to copy a sector-sharing disk by file, the duplicate will unshare them!

Also, you may get an out-of-space error. So to copy such a disk you should use a sector [or track-sjs] copier. Then there's the problem of what happens if you want to copy the sector sharing files but not the whole disk- if you can't do a direct copy sector x to sector x, I would recommend that you find a way to sector copy the whole disk, and delete the files you didn't want.

Or you could copy only one of them and start the operation over again on the new disk..

So to make a copy of this 95 sector program, it took only four additional sectors! Three for each new header sector, and one for the modified sector. Shorter files would need even fewer additional sectors. I could go on and make another version of this program, but I think I'm happy with just two. I wonder if I have any other files I can do this to....

[[Disk Utilities is available from the User Group library. If you can follow this by no means simple article you will quickly learn how to use Disk Utilities - which does have simpler applications - and you will certainly learn very effectively how your disk system works- we have already covered sector layouts in previous issues, this article really hammers it all home! sjs]].

=====.

PROGRAMMING HINT- NOT BASIC.....

This note is appropriate to any language other than Basic- and is prompted by a member having problems with the command language of TI BASE..

His code was along the lines of:.

```
WHILE RESPONSE="Y".
  INPUT PROCESS.
  WRITE X,Y "ANOTHER?".
  READSTRING RESPONSE.
  IF RESPONSE="N".
    CLOSE DATA.
    RETURN.
ENDIF.
ENDWHILE.
```

Repeated processing of this type of code was resulting in an error message 'too many if while or case levels'.

How too many levels- only one surely? One WHILE/ENDWHILE and one IF/ENDIF... no? NO! Actually too many whiles under certain circumstances!

Basic is unique in computer languages in that it allows you to get away with code like this:

```
100 FOR T=1 TO 1000.  
110 CALL KEY(1,A,B).  
120 IF B=1 THEN 140.  
130 NEXT T.  
140 PRINT "DONE";T
```

in any other language- including the TI Base command language, this would spell disaster. The general rule of programming is that you do not jump out of a conditional loop until the condition is met- GOSUB can provide us with a Basic example:

```
100 GOSUB 1000.  
110 END.  
1000 FOR T=1 TO 1000.  
1010 CALL KEY(1,A,B).  
1020 IF B=0 THEN 100 .  
1030 NEXT T  
1040 RETURN.
```

if you run this, you will after a little while of not pressing any keys, receive the Basic equivalent message to the one you are getting with TI Base- each GOSUB is remembering where you jumped from, and unless you exit with a RETURN, you just keep remembering more and more return addresses until....

(I really have seen quite a lot of programs coded like that last bit of code!)

The BASIC program, in a format other languages can use, is:

```
100 FOR T=1 TO 600.  
110 CALL KEY(1,A,B).  
120 IF B=1 THEN T=600.  
130 NEXT T.  
140 PRINT "DONE";T.
```

Basic actually clears the T flag when you jump out- other languages wont, they will keep waiting for the condition to be met, and if the code is processed frequently, memory will quickly go!

That Basic allows the first form, and teaches a method inapplicable to any other form of programming, has resulted in Basic receiving a lot of criticism from professional programmers. TI went a long way to meeting criticisms by offering in ExBas sub-programs with local variables- long before Acorn thought of it!

continued.....

The first bit of code should be configured as below- we do not jump out of the conditional WHILE loop until the condition is met...

```
.  
WHILE RESP="Y".  
  INPUT PROCESS.  
  WRITE X,Y "ANOTHER?".  
  READSTRING RESP.  
ENDWHILE.  
CLOSE DATA.  
RETURN.  
.
```

If you compare this to the first form, you will see it does the same thing, but actually uses the test in the WHILE. And as we only leave the WHILE loop when the condition is met, you don't eat up memory by nesting WHILE loops when you don't need to! (The code is also shorter)..

This is one case where early experience with TI Logo would have helped, as that would rapidly have taught the perils of iteration (that's the technical name for going around in circles!). TI LOGO actually counts the "level"s as you go down....
=====.

REVIEW: .

TI RUNNER LEVEL EDITOR. by EB SOFTWARE.

Advertised for US\$18.95

Review by Stephen Shaw.

This review was commissioned by MICROpendium, and appears here by virtue of our exchange agreement with them. Thanks are expressed to MICROpendium for the review copy. They chose not to print this review but instead printed a favorable review by another....

TI RUNNER is a well known popular game on the TI, with versions available on disk (50 screens), tape (40 screens) and module (25 screens), the disk version from EB SOFTWARE (or by virtue of paid licence, from this group) , the module version by Databiotics (called Star Runner) and the cassette version, for XB plus 32k ram, from licensed user groups (including our own group)..

NOTE: The following text refers to the disk which was supplied to MICROpendium for review, and does not apply to the current commercial release. -Please see the later part of the article for a brief look at the latest release supplied directly to me by EB Software..

This EDITOR allows users to create their own screens for use with the Disk version, creating merry mazes with the three types of brick- those you can make holes in, those you cannot, and those you fall through! and of course the ladders, including the ladders which only appear when you have collected all the treasures you need to.

Supplied on disk, and requiring XB plus 32k ram, the program was supplied with 9 pages of A4 documentation, stapled at top left. The disk contains the original 50 levels in one file, and a further 40 advanced levels in another file. These are provided for you to edit if you wish. You can of course create a second games disk by copying your original TI Runner program to a fresh disk, and copying the ADVANCED file to it, renaming it LEVEL28. .

LE (TI Runner Level Editor) allows you to create entirely new files of screens, or to copy individual screens into a new file in a new order, as well as

editing existing screens..

This product is of interest to any TI Runner player who has made it to screen 50 (anyone?) or who finds it quite impossible and would like to edit a problem screen to make it easier! .

When using any product of this nature it really is wise to make up those suggested back up copies of your disks, and edit only the back up copies, never a master copy!.

The documentation and program caused me a few sleepless nights, but most of the difficulties have now been solved and can be passed on to you. The game disk received was NOT marked as a Beta test version, but was received as a final commercial offering. EB software have received an advance copy of these corrections and it is to be hoped that subsequent sales will contain them!.

To deal with the three sections of the program in turn....

1. LEVEL MANAGEMENT... this is where you edit a screen, add a screen, copy a screen, or delete a screen, all very straight forward! .

However if you choose to add or edit a level, the documentation requires a little elucidation:.

The first selection when you choose to add/edit a screen is CHANGE COLOR- this selection must be chosen also if you wish to amend (or see) the number of treasures that must be collected to complete a screen. You will need to know how many treasures the screen needs, so use this option!.

Changing colors is a little odd- instead of entering 1 to 16 to input colors, the program requires you to enter CHR\$(48) to CHR\$(63)- that is, 0 to ?. which is indeed displayed on screen, but is not a normal method of thinking of colors. This can be amended to a more normal operation with very little code, and is merely an indication of poor design..

The documentation lists a number of keys to press to select the drawing character- the various types of ladder, treasure and so on, which can then be placed by cursor keys or joystick. The documentation fails to mention that instead of pressing keys 1 to 9 to select a character, you can select space to erase characters! The auto-repeat from left to right OR from top to bottom is a nice touch in this section. When placing characters using keyboard or joystick, they move 8 pixels at a time..

The documentation indicates that you may leave the editor by pressing enter- it would have been nice to be told in advance that this does NOT immediately leave the level but rather then allows you to place first your own man and then the three baddies..

The men can be moved by keyboard or joystick, but if you move them by joystick they move only 1 pixel at a time. The documentation carries the following text:.

"You must be VERY careful when placing the men with the joystick since they may act strangely when the .TI-Runner. game is played.". Thats it, that is your warning- no further explanation! No I can't explain this cryptic comment (I am useless at crosswords too...) except to suggest that it may be possible to place a man such that he may be one pixel into a brick! It is safe to place the men one pixel above the bricks- provided their heads are not thereby placed one pixel into an overhead brick!.

BUG: If you placed the last man using the joystick fire button, you had distinct problems in carrying on using the program! You could not enter anything to select from subsequent menus unless you wishes to enter the default value, and if you scrubbed that out could not reenter it.....

continued.....

This caused me some headaches. What was happening was that although the ACCEPT AT routine for the menu was perfectly alright, the console was STILL mapping the keyboard as CALL KEY(1...), something you cannot emulate in BASIC, but the routine to move the men is in machine code, and that code is not resetting the key scan flag on exit! Fortunately you can reset the keyscan flag in basic, so this bug is easily resolved..

EDIT the program EDITOR as follows:

Into the prescan lists at the start of the program add the variables @ and @@, and the CALL KEY. Then add a new line as follows:.

```
1545 CALL KEY(3,@,@@).
```

and THAT bug is quite squashed!!!.

When editing or creating a new screen, there are several things you must watch out for- it is up to you to ensure that the screen really is completeable..

Remember I advised you to note how many treasures were required to finish a screen? Two quotes from the docs will explain this:.

"You can enter 5 treasures, but have six on the screen and fool the player into going for impossible treasures".

"A level created with more treasures on the screen than are needed to win that level will behave strangely. When playing this type of level, the .TI-Runner. man will not be able to get up any ladders if more than the set amount of treasures are taken. At this point the level is impossible to win"..

I consider this a design fault- the editor should have been able to note the number of treasures required, and flashed a warning if you tried to leave the editor with too many (or just as bad, too few) treasures on screen. It is not fatal, just requires CARE!.

We now come to:.

2. FILE MANAGEMENT. Which contained a very interesting bug indeed, pure Extended Basic, and the first time I have met him. Sub-option 1 should allow you to "change drive number"- this allows you to create a new file on a blank disk. LE uses a data file SCRLST to record the different level files on your disk, faster than using the disk directory! So when you select a new disk, it is necessary for the file SCRLST to be created. My copy of the program steadfastly refused to do this, preferring to terminate the program..

This error was caused as follows:.

The program has one ON ERROR routine..

Specific errors are caught by making use of CALL ERR..

Change Drive Number caused an error if the program could not read from a file SCRLST on the new disk. .

This error was tested for by a test for the LINE NUMBER the error occurred in. .

Know something? When your program has:.

```
1000 CALL ERR(A,B,C,D).
```

```
1010 IF D=2030 THEN....
```

where D refers to a potentially difficult program line, if you then resequence your program, that line reference test in line 1010 is NOT amended, and your error check fails to work!.

This is what happened, and was spotted as a REM line very rarely causes errors (ever?). So, again you must EDIT the disk file EDITOR: .

Change line 2910 to read:.

```
2910 IF Z=1710 THEN EMPTY=1 :: etc etc.
```

And that is that bug squashed..

continued.....

Finally, Option 3, PREPARE A SCREEN FILE FOR PLAY. I can understand (almost) why this option was provided, but my best suggestion is that you do not use it. The LE creates a data file with a name chosen by you. In order to use that file, its name must be changed to LEVEL28, the name that the TI RUNNER program expects!.

Option 3, calmly and coolly creates a new file LEVEL28 and then copies the data across. We are talking here about a 100 sector data file!.

It is very much faster to use your favorite disk manager to change the file name and copy it over to a duplicate program disk, remembering to call the disk name TR..

There is a further error in the documentation supplied at this point. "Be prepared to swap disks if you have only one disk drive". This option will not ask you to, nor indeed let you. Your disk manager will..

Subject to bug squashing it is a simple matter to use this program, and the difficulty is not in using the program but rather in designing new screens! In any case, you have an extra 40 screens to work your way through (wow are they hard!)..

In summary, subject to the changes and explanations here provided, this is a product that TI Runner aficionados need. The product can be improved in design and documentation, but the price is not very high.

REPORT CARD:.

Heading:	As supplied:	Debugged:.
PERFORMANCE.....	C.....	A.
EASE OF USE.....	D.....	B.
DOCUMENTATION.....	C.....	A
VALUE.....	A.....	A
FINAL GRADE.....	C.....	A-.

Following on from the above review, written a few weeks back, I have now received notification from EB SOFTWARE that the version sent to MICROpendium **for review was a BETA TEST copy** (there was no indication of such in the program)..

EB SOFTWARE have sent to me a disk claiming itself as VERSION 1.3 of the Editor. In addition to an ADVANCED set of screens to play with your TI RUNNER program there is also a further 50 screens - so this disk would add two further sets of screens for you to play! regardless of whether you use the editor..

Despite finding some bugs in the original review copy, I was at least able to make it work..

I regret to report that while Version 1.3 has every appearance of working correctly, the files that it produces are highly unreliable. Typical faults include incorrect behaviour of hidden ladders- which sometimes work while invisible, and sometimes fail to appear when all the treasure is taken, and the unreliable operation of the bad guys, one or more of whom now run along the screen horizontally, regardless of any walls or lack of walls, or of any obstructions..

Using the CREATE SCREEN option makes things even worse, causing severe corruption to the playing screen..

If Version 1.3 is the current commercial version then I regret I must downgrade all gradings to E-. In printing a favorable review later, MICROpendium did NOT mention any version number!!!!!!!
more-->

Having now received two versions of the program, both of which failed to work when received I must recommend caution! I WAS able to make the first version work, but the fault in the current commercial version seems to be in the machine code section. If you send for the commercial version and it fails to work, at least you have the two sets of screens! and if you send me your ORIGINAL purchased disks, I can at least send you a modified and working beta test version of the editor, as first reviewed above! (Return post and packing please!)

I have also tried a FREeware editor- the first disk received was apparently blank, and the second one, while apparently creating new screens, also failed to produce screens that would play correctly. .

Stephen Shaw.

=====

TURBO PASC 99 TUTORIAL

By Stephen Shaw..January 1989..

This article is going to start in a way which I hope will show you that TP99 is an extremely easy language to actually use, and by no means difficult to program in. The better a TI Basic programmer you are, the easier you will find TP99- and looking at TP99 may even make you a better TI Basic programmer!.

We are going to start with a program every TI owner has, from the manual supplied with the console- look up GOTO in your manual and you will find a little TI Basic program:.

```
100 REM HOW MANY GIFTS ON.  
THE 12 DAYS OF CHRISTMAS?.  
110 GIFTS=0.  
120 DAYS=1.  
130 COUNT=0.  
140 COUNT=COUNT+1.  
150 GIFTS=GIFTS+1.  
160 IF COUNT=DAYS THEN 180.  
170 GOTO 140.  
180 DAYS=DAYS+1.  
190 IF DAYS<=12 THEN 130.  
200 PRINT "TOTAL NUMBER OF G.  
IFTS IS";GIFTS.  
210 END.
```

(The total number of gifts is 78)..

That is a fairly short program to convert, but by no means simple as you will see...

To start off using TP99, first insert your Editor Assembler module- at the time of writing TP99 insists on using only this module as direct access is made to the ROM. You can also use (as I do) a Super Space module, which has the Editor Assembler ROM inside it..

Select the Editor Assembler option 5. RUN PROGRAM FILE and with your TP99 disk in drive number one, press ENTER- this will automatically load the file UTIL1 and the following files, which contains the editor and the compiler..

The opening screen will tell you that your workfile is empty, and the cursor is down at bottom left (quite invisible on my tv set, which cuts off that left hand column!)

.
Type in ED and press enter. The cursor is now at top right (still invisible to me!). If you cannot see it, press space bar a few times!.

.
Now you are ready to type in the code which follows. You can type it all in with the alpha lock in the up position, and do not need to use the SHIFT key at all... the capitals you see listed have been put there automatically by the TP99 editor..

.
The left hand column margin is set from the column you start typing- all subsequent new lines will start off underneath the first character of the last line, UNLESS you move the cursor with the arrow keys. .

.
The indentations in the listing are not required, but help you to follow the code, and are in general accordance with the usual Pascal practice. The most common error in typing TP99 code in is omitting those semi colons at the end of so many of the lines. They are essential! And you may experience a little difficulty with the equals sign... note that we have here used := (colon equals). There is a little more to it than that but lets keep it easy this time!.

.
In transferring the Basic code to Pascal, there is one difficulty. Pascal does not have a GOTO, nor any means of branching back into previous code - indeed, Basic is a little unusual in allowing such freedom to jump backwards and forwards, and for that reason met with some hostility from professional programmers! That does not concern us- what does is making the transfer!.

.
Examining the Basic program, there are three jumps, so we can connect with arrowed lines a jump from 160 to 180, and two backwards jumps, from 170 to 140 and from 190 to 130..

.
To deal with the backward jumps we have to use one of the LOOPing procedures in Pascal, and have quite a variety:.

DO WHILE, FOR...NEXT, and REPEAT UNTIL..

As with Basic, there are many ways of doing the same thing, and what I show here is just one way of converting the Basic program- there are other equally valid ways..

.
As we have two backward jumps, that means two loops, an inner loop (140 to 170) which increments COUNT and DAYS. The loop is controlled by a test of equality between COUNT and DAYS..

.
The outer loop (130 to 190) in addition to the inner loop, resets COUNT=0, increments DAYS, and is controlled by testing the value of DAYS against a value of 12..

.
Outside these loops we have the variables initialised and the result printed..

continued.....

Now lets take a look at how that can be put into Pascal code:.

```
PROGRAM xmasgifts;
VAR gifts,days,count : INTEGER;
PROCEDURE innerloop(VAR count,days,gifts : INTEGER);
BEGIN
  WHILE count <> days DO BEGIN
    count := count+1;
    gifts := gifts+1;
  END;
END;
PROCEDURE outerloop(VAR count,days,gifts : INTEGER);
BEGIN;
  WHILE days < 13 DO BEGIN
    count := 0;
    innerloop(count,days,gifts);
    days := days+1;
  END;
END;
BEGIN
  CLS;
  gifts := 0;
  count := 0;
  days := 1;
  outerloop(count,days,gifts);
  writeln(" total no of gifts",gifts);
  readln(days);
END.
```

Pascal requires that we start with a program name, that is at the top, and although required is of little consequence..

Next we have declared the global variables- these variables, in this position, are available to every part of the program. In addition to these it is possible to declare in each procedure, local variables, which like variables in Extended Basic user written sub-programs, can have different values to the "outer" global variables. All the numbers we are dealing with in this program are integers, so for greater speed and reduced memory usage we declare the variables as INTEGER (this is a good language for learning not to spell that word with an extra R!)..

Declaring variables is necessary in Pascal, and many other languages. In Basic it is taken care of with the PreScan, which TI gave us so much more control over in Extended Basic. It sets aside a bit of memory to hold the values!.

We then have the two loops, which I have here written as separate PROCEDURES. In Pascal, a procedure must appear in the code BEFORE it is called so we first have the innerloop, then the outerloop. .

After the identifying word PROCEDURE and the name of the procedure, we list the variables that are being transferred to the procedure, and as we wish to pass

them back to the calling routine, we add the word VAR at the start of the list. The type of the variable is also declared in this list. Not very different to an EXTENDED BASIC CALL....

The innerloop should be fairly easy to compare to the Basic program. We are incrementing both COUNT and DAYS, and then when count equals days, returning to the calling routine..

The outerloop is not that hard... just that instead of testing for DAYS<=12, we jump out of the loop when DAYS=13, which is just another way of saying the same thing! You can see here how one procedure is using a previously defined procedure, as we use the procedure innerloop..

After the two PROCEDURES follows the actual but of program!!! I have added the input (READLN) at the end so you can see the answer! It requires you to input a number before leaving the program and returning you to the title screen..

Once you have typed all this in using the EDITOR, press BACK (FCTN 9) to return the cursor to the bottom of the screen, and do a test compile by entering C0 (then press ENTER). If compilation is complete, you may save the code by entering SA DSK2.DAYS/P and then compile to disk with C0 DSK2.DAYS/S..

You may of course use your own file names! I use /P to indicate PASCAL source code, and /S to indicate 9900 Source code..

If you have an error, the program will place you in editor mode with the cursor near to the error point- it will actually appear at the point at which an error has become apparent, so a missing semi colon may result in the cursor appearing after the first word in the line below that which does not have that necessary punctuation. Check out the manual and your code! It is fairly straight forward.

In preparing this article I left off two semicolons! and also miscounted my BEGIN/ENDs, which have to balance out..

You now have on disk your Pascal source code in DV80 format, and some 9900 source code, also in DV80 format. Now you have to assemble it! My TP99 disks had the assembler ready for use, so just choose the Editor Assembler option ASSEMBLE, and with a source code file of DSK2.DAYS/S, and an object file of say DSK2.DAYS/O, off you go. I choose to use just the R specifier when assembling (habit!). .

NOTE: When assembling, you MUST have your TP99 disk in drive one, as the source code (DAYS/S) has an include directive to copy in RUNLIBEQ from the TP99 disk!!!.

Your assembled object code is now ready to run- but will NOT run on its own! You MUST use the LINKER! Choose the Editor Assembler option 5. RUN PROGRAM FILE and input the name DSK1.LK99..

The "module name" to input once the linker is operational is DSK2.DAYS/O - that is, the object code file. As there is only one module, press enter next, then you are asked if you want a memory image file- type Y, and input the file name- I chose DSK2.DAYS/MI..

Then when asked if you want to RUN the program, type Yes! .

NOTE: You must at this stage have in drive one your TP99 disk, as the LINKER . uses the file RUNLIB..

Once the memory image file has been prepared and the program run, your disk directory may look something like this:

DAYS/MI	3 Program)	Memory image produced with
DAYS/MJ	4 Program)	LK99
DAYS/O	6 Dis/Fix 80	=	Object code from TI Assembler
DAYS/P	4 Dis/Var 80	=	Pascal source from TP99 editor
DAYS/S	6 Dis/Var 80	=	9900 source from TP99 compiler

Yes, two files for the memory image form!

The memory image files will run without having to use the LINKER, but INSIST that RUNLIB is in DSK1, and also REQUIRE the use of the Editor Assembler module (or at least its ROM!).

TURBO PASC99 is a major language for the TI99/4A - and will be a lot better when we have access to graphics and sound! - and really is very easy to use!.

Best wishes, Stephen Shaw.

=====.

(Author unknown. Found on disk...)

XB SCREEN COLORS

QUICK AND EASY COLOR . (Extended Basic plus 32k ram required)

Here's a super short, super FAST, assembly routine for X-BASIC that allows you to change screen and character colors instantaneously! .

There's lots of possible uses for the thing including games, but the real feature of this program is that it changes the color of the EDIT Mode screen as well! Yep, no more black on cyan if you don't want to! .

HOW DOES IT WORK? .

The color change is inserted into the USER-DEFINED INTERRUPT and is constantly "re-performed" every 1/60 of a second. This makes it seem like the EDIT Mode screen has been changed. In order to return control of the color commands CALL SCREEN and CALL COLOR, you must load the User-Defined Interrupt with zeros. . (e.g. CALL LOAD(-31804,0,0). Any use of CALL COLOR or CALL SCREEN while the routine is operational will just cause the screen to flash briefly..

DEMO PROGRAMS .

Along with the program that loads in the original routine, below is a demo routine to show off your new screen colors. .

A SIMPLE CALL LOAD WILL DO IT .

continued.....


```
420 PRINT "DATA LINES ENTERED  
WRONG. CHECK AGAINST LISTING!"  
430 END.
```

```
PROGRAM #2: CLRDEMO1/X  
-----
```

```
100 ! *****  
110 ! * *  
120 ! * COLOR CHANGE *  
130 ! * *  
140 ! * DEMONSTRATION *  
150 ! * *  
160 ! *****  
170 !  
180 ! 11/84  
190 !  
200 ! SUBFILE99  
210 !  
220 ! *-----*  
230 ! NOTE:  
240 ! YOU MUST HAVE  
250 ! ALREADY LOADED &  
260 ! RUN "SCRNCOLR/X"!  
270 ! *-----*  
280 !  
290 CALL CLEAR :: RANDOMIZE  
:: DIM C$(15).  
300 M$="Screen Color Change  
Demo".  
310 ! .  
320 FOR C=0 TO 15 :: READ  
C$(C):: NEXT C.  
330 ! .  
340 !  
350 DISPLAY AT(1,1):M$:DISP  
LAY AT(2,1):RPT$("-",LEN(M$)  
)  
355 DISPLAY AT(10,1):"FRGRND  
COLOR:"  
356 DISPLAY AT(16,1):"BKGRND  
COLOR:"  
360 ! .  
370 FC=INT(RND*15)  
380 BC=INT(RND*15)  
390 DISPLAY AT(10,14):C$(FC)  
400 DISPLAY AT(16,14):C$(BC)  
410 !  
420 CVAL=16*FC+BC.  
430 CALL LOAD(9460,CVAL)  
440 !  
450 GOTO 370  
460 !  
470 DATA Transparent,Black,Me  
dium Green,Light Green
```

480 DATA Dark Blue,Light Blue,
Dark Red,Cyan .
490 DATA Medium Red,Light Red,
Dark Yellow,Light Yellow.
500 DATA Dark Green,Magenta,
Gray,White .
510 END .

< * End of file * >

=====
end 4/89

=====
submitted september 1990

9/90

RAMBLES. for TI*MES 30 / September 1990.

.
Greetings and welcome to another Rambles, a collection of odds and sodds related to the TI99/4A. Your comments and requestes and queries are very welcome.

.
First an important update on ASGARD- a major publisher of software for the TI, who nevertheless managed to get all my four orders wrong and therefore carries a warning tag. In past issues I indicated you could obtain Asgard programs via Comprodine, alas this is no longer the case.

.
=====
. ANCIENT HISTORY.

Our local library regularly sells off unwanted books and recently cleared their old computer books (including Townsends TI99/4A book). I picked up two product guides from 1982 and 1984..

I bought my first TI back in 1981, so the '82 book is a good record of what I had to choose from- the TI was just changing from 99/4 to 99/4a. Comparing the 1984 and 1982 books, only the ZX81 was still available two years later- pick any other computer in 81 or 82, and you would have been orphaned by 84!.

Here was the scene in 82:.

The TI99/4A was selling (for a day or so, the price started to crash downwards from here on!) at 399 pounds! Seen as competition, and producing a price reduction later, was the 4k VIC20 at 185 (22x23 char display!). Here are the more well know 1982 consoles in price order:.

rs.

5k VIC20 (22x23 display)	185.00	.
8k OSI C1P with 8k basic,	250.00	.
12k Acorn ATOM with only intger basic and mono o/p.	250.00	.
16k Video Genie EG3003 with 12k basic and 64x16 display.	289.00	.
4k AIM 65 with a 20 x 1 (One) display! and 1k basic,	315.00	.
32k Acorn Proton (forerunner of BBC-B)	335.00	.
8k Commodore Pet 4008 with 8k basic. and mono monitor	378.00	.
32k ATARI 400 with rubber keyboard.	395.00	.
16k TI99/4A for	399.00	.
4k TANDY TRS80/1 incl tv and recorder.	430.00	.

By the time the 1984 book came out, the TI99/4A had been withdrawn, and the Dragon, which Boots had switched to selling, and which some TI owners rather too quickly switched to, had just been withdrawn as the liquidator was called in! Of the 1984 computers, apart from the price drops and improved specs, it is interesting to look around the computer stores and see that some are still commercially supported, and indeed, much to Commodores chagrin, one the C64 is

still earning large sums of money, six years on! Properly supported could the TI have kept going as the C64 has? You betcha! The 1984 computers....

A real curiosity to start- anyone remember the 4k Tandy MC10 at 70.00- it had no program editing facilities!!! If you wanted to alter a program after typing it in, you had to type it in again. Made debugging a chore!.

48k Spectrum 130.00	20k CGL (Sord) M5 150.00.
16k Colour Genie 168.00	48k Oric Atmos 170.00.
48k Alphasatronic 170.00	16k Spectravideo SV318 186.00.
64k Commodore 64 199.00	32k Acorn Electron 199.00.
48k Lynx 225.00	64k Amstrad CPC464 incl grn monitor 229.00.
64k Dragon 225.00	64k Atari 800XL. 250.00.

At this time you could have picked up a TI99/4A for around 50 or less. .

There were others which did not last long enough to make it into either book, and of course others which were well outside this sort of price range..

Update on the scan of George in last issue- and the cartoon scan in this issue, timed to coincide with a new series on BBC TV- Star Trek, The Next Generation.. (Notung used the ST-TNG scan on the cover of their ST TNG Calendar manual):

The scans have been done by Ray Kazmer of Sunnyvale, California, and the procedure now used is as follows:.

Scan using IBM PC clone and clean up using FIRST PUBLISHER. Version 3 allows perfect ratio on the PC! Save in MAC format..

Transfer to TI99/4A using PC-TRANSFER, which despite a habit of scrambling pictures most of the time, is said to be faster than using a direct wire hook up with comms programs on both machines..

And once on the TI, the MAC format pics can be printed using PIX PRO and transferred to other formats such as TI Artist if size permits..

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

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

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

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

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

```
1 CALL INIT :: CALL PEEK(-31
952,A,B,C,D):: SL=C*256+D-65
539 :: EL=A*256+B-65536 :: F
OR X=SL TO EL STEP -4 :: CAL
L PEEK(X,E,F,G,H):: ADD=G*25
6+H-65536 :: PRINT *256+F .
2 CALL LOAD(ADD-1,0):: NEXT
X :: STOP :: !@P-
```

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

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

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

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

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

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

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

```
.
1 CALL INIT :: CALL PEEK(-31952,A,B,C,D):: SL=C*256+D-65539 :: EL=A*256+B-65536
:: FOR X=SL TO EL STEP -4 :: CALL PEEK(X,E,F,G,H):: ADD=G*256+H-65536 :: PRINT E
*256+F
2 I=1 :: CALL PEEK(ADD-1,V):: IF V THEN 6
3 CALL PEEK(ADD+I,V,W):: IF V THEN I=I+1 :: GOTO 3
4 FOR Y=SL TO EL STEP -4 :: CALL PEEK(Y,E,E,E,F):: IF E*256+F-65536=ADD+I+2 OR W
=0 OR ADD+I>-3 THEN CALL LOAD(ADD-1,I+1):: GOTO 6
5 NEXT Y :: I=I+1 :: GOTO 3
6 NEXT X :: STOP :: !@P-
```

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

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

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

```
* PROGRAM TO REENTER LINE .
* LENGTH VALUES TO A BASIC PROG .
* author wishes to be anon .
* FROM TISHUG JULY 90 .
* .
* RUNS ONLY FROM XB+32K .
* load corrupted program into xb .
* CALL INIT .
* CALL LOAD("DSKn.FILENAME") .
* CALL LINK("UNFIX") .
* .
DEF UNFIX .
AORG >2500 .
UNFIX LWPI USRWS .
MOV @>8330,R1 .
MOV @>8332,R2 .
C R1,R2 .
```

```

        JHE  FIN
*
        INC  R2
        S    R1,R2
        SRL  R2,1
        CI   R2,BUFMAX
        JGT  FIN
*
        SRL  R2,1
        MOV  R2,@BUFLEN
        INCT R1
        LI   R4,BUFF
*
LOOP    MOVB *R1+,*R4+
        MOVB *R1+,*R4
        DEC  R4
        DEC  *R4
        INCT R4
        INCT R1
        DEC  R2
        JNE  LOOP
        LI   R5,>FFE8
AGAIN   MOV  @BUFLEN,R3
        LI   R2,ZERO
        LI   R1,BUFF
        DECT R1
*
NEXT    INCT R1
        C    *R1,*R2
        JL   SKIP
        MOV  R1,R2
SKIP    DEC  R3
        JNE  NEXT
*
        MOV  *R2,R6
        MOV  *R1,*R2
        S    R6,R5
        DEC  R5
        SWPB R5
        MOVB R5,*R6
        MOV  R6,R5
        DEC  @BUFLEN
        JNE  AGAIN
*
FIN     LWPI >83E0
        CLR  R0
        MOVB R0,@>837C
        RT
*
USRWS  BSS  32
BUFLEN BSS  2
ZERO   DATA >0000
BUFMAX EQU  >1800
BUFF   BSS  BUFMAX
*
        END

```

A great deal faster! This machine code is for loading with ExBas ONLY!

Why do we need to have the length right for editing? 'Cos if the length byte is 255 bytes, when you delete (or edit) the lines, the computer will delete 255 bytes, and if the line is shorter, you are going to lose other data, probably partial lines, which will be very sad!.

Oh yes... if the corrupted program comes from elsewhere, after making it listable, take a look for any lines that look like our first two lines above, if they are there you need to remove them or they will make the program unlistable once more!

COMPETITION from Doug Moller, no prizes just send in your code! .

1. At various times the hands of a clock with trisect its face- there will be a 120 degree angle between the hands. Assume all hands start at 12 and all three hands move smoothly. When will the angles between the hands be exactly 120 degrees? OK harder one- this next clock has a ticking second hand, which moves only in one second jumps but the other hands move smoothly....

2. A journey of 300 miles is undertaken in a car rented at \$3 per hour plus fuel at \$2 per gallon. What is best (eg cheapest) speed? To be totally unreal, assume fuel consumption is 30mpg at 30mph, 20mpg at 60mph and 15mpg at 75mph. eg an increase in speed of 3mph means one gallon lasts 1 mile less.

This issue omits several regular contributions, as my annual holidays fell in the last two weeks before press date, and unfortunately, in the previous month to that my father was not too well in hospital, leaving me little time to put this together! Still we have the full 60 pages and perhaps back to normal content next time- not that anyone ever writes to me to say they want to see more XB tutorials or more Tigercub or more reviews or more... (or less!)..

MODULE REVIEW: TRIS- Famous Game!. Asgard Software, 1989. Main author Jim Reis..

Who cannot know this game? The TI module is a perfectly good emulation, in which tetromino shapes fall, can be rotated and dropped, and must be formed into a solid horizontal line. You score by how quickly you line the shapes up and drop them into place. Game ends when the shapes stack up to the top due to your having left empty spaces in horizontal lines. .

This is very similar to the Hulpke disk version, except that you may not amend the "next shape" (which is optionally displayed in the module version) and the module version benefits by automatically increasing in speed as play progresses- Hulpkes version plays constantly at the same level..

A good module for unexpanded owners who cannot play the Hulpke disk version, this is also a good module for all games players. .

Music and sound are credited to Ken Gilliland, Bruce Harrison, Jim Reiss.

HOW GOOD IS IT? My wife -Cathy- has not played a module game before- she is hooked on this one, belting the living daylights out of a joystick and not infrequent *expletive deleted*s! My six year old doesn't get too high a score at the easiest level but still likes a go. I play it. Look- if you ever play games this is one for you. If you have never played a game on your TI, this is still one for you. Buy it. OK!.

Recommended..

GAME REVIEW- DISK- BALLOON WARS- 1985. ASGARD SOFTWARE.

Written by John Morrison..

Quite an old game this one, I've seen it advertised but not reviewed. Its an ExBas game, but uses disk data files, so is not suitable for tape. .

My disk was quite unusual, having a disk jacket a trifle floppier than the disk inside it. I was surprised when my drive read it (and copied it onto a more substantial disk!). The program is list protected but most disk owners know what to do about that! .

Your task is to pilot a hot air balloon from screen left to right across several screens, using fuel as you go! There are enemy troops below with a nasty habit of firing at you- you may avoid their shots and you may bomb them. On some screens, after removing the enemy, you may land for repairs and restocking. .

The documentation is in error- it is joystick down to release a bomb, and joystick up to drop a snabag (not vice versa)..

This program is not exactly sophisticated, and the graphics are very simple, but it is an interesting game which presents some simple strategic decisions to be made. If you like a game to can get into quickly, you'll like this one..

=====
=====.

UTILITY REVIEW- TI BASE VERSION THREE- DISK. TEXAMENTS. US\$25+ \$8 p and p..

We have already covered earlier versions, this one is fully compatible with earlier ones but spreads its wings in many new directions! .

Of particular interest is the ability to place command files into VDP memory- and save and load them as memory images- for really rapid operation, more so if you are using a ram disk for the actual data! There is also the availability of using this option to create macros- making up your own commands from the ones supplied..

The report generator is possibly misnamed, it is a database of command files, and while it can certainly be used to produce screen and printed formats, it is not limited to that use. The major limit would be the number of commands you can fit into the database, but with the macro facility already indicated above, this additional facility gives the programmer considerable power to hide TI Base behind a fast and friendly user interface- the user need never know TI base is in use, apart from the loading screen!.

Inverse characters are available, but the documentation is incomplete. You can turn all text inverse with the command SET INVERSE ON but you may also mix normal and inverse text by leaving SET INVERSE OFF and inserting characters with ASCII codes 128 higher than usual- eg an inverse A is character 193 instead of 65. These characters are used on Epson printers as the italic set. You cannot enter a character 193 into a command file using TI Writer, you must use the TI Base editor, which has another new option, familiar to PC users as the ALT key: hold down CTRL and key in the three numbers of the character you want (eg 193) and that character will appear on screen. .

(As the TI Base editor is limited to small command files, we have in the disk library a utility to merge such command files into longer ones)..

Of course, as a program/language becomes more powerful it tends to become harder to use, and indeed, to become underused as few bother, but TI Base can be used without command files at all, for the early owner, leading upwards to a very professional database presentation for the more experience programmer who wishes to impress (very often its the experienced programmers who leave the stitches showing!)..

If you want a database, this is the one! Can work with one SSSD disk drive, works better with two disk drives, and is happy with ram disks or hard drives..
=====.

MUSIC TAPE REVIEW: THE FUNNIEST COMPUTER SONGS

Various artists.

US\$12 inc p&p from Vince Emery Productions. San Fransisco,..

Not a mention of a TI99/4a on this but well worth having if you are a computer phreak, and essential if you are a computer pro! Most of the tracks on this tape are in the folk/blues tradition, and are well presented and recorded, making for fun listening. Tracks include....

HALs song... you need to have seen the film 2001 to appreciate this one. It ends with "I'm ready for my first lesson"- remember now?.

Please, Mr Compatibility... it helps to know a little about computers to appreciate the humour here, like the request to connect a printer using the parallel port via a co-ax cable. Dont understand that? Never mind there are other tracks....

I built a better model than the one at Data General- the music here is from Gilbert and Sullivan, "He is the very model of a modern major general"-was that Iolanthe?- a high tech patter song. Try it for yourself!.

Stuck Here, Threes, and S-100 Bus.

I'm a Mainframe, Baby- blues style..

Uncle Ernies Used Computers Babbages Birthday Bargain Bash- a more modern patter song..

Mushrooms... you know, keep em in the dark!.

Engineers Rap- this one is a bit like a thorn, being totally different to any other track, yes your actual computer rap!.

Killer Byte Blues...more blues..

Do it yourself (this had Cathy-my wife-and I distinctly laughing. A different way to own a computer just like the ones IBM makes....

(Sit in the LOTUS position to hear this, make yourself WORDPERFECT and do not dBase yourself. Ready... then its 1-2-3...)..

This is a good fun tape, recommended..

=====.

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

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

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

```

int num[2500];

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

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

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

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

for(count=0;count<=800;count++)
{puts(" ");}
locate(3,1);count=0;
}
}
}

```

```

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

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

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

```

DEF SEVAC

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

MULT  INC 2
      CLR 4
      LI 0,DIG   [DIGIT STORE]
C     INC 4
      MOV *0,5
      MPY 7,5
      A 3,6
      DIV 10,5
      MOV 5,3
      MOV 6,*0+
      C 4,1
      JNE C
      CI 3,0
      JEQ D
      MOV 3,*0
      CLR 3
      INC 1
D     B *11

TEST  CLR 4
      LI 0,DIG-2
E     CLR 8
F     AI 4,6
      C 4,1
      JGT RTN
      AI 0,12
```

```

MOV *0,6
C 6,7
JNE F
G DEC 4
CI 4,0
JEQ H
DECT 0
MOV *0,6
C 6,7
JEQ G
H INC 4
C 4,1
JGT RTN
INCT 0
MOV *0,6
CI 6,7
JNE E
INC 8
CI 8,6
JNE H
RTN B @PRINT
B *11

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

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

```

```

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

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

VMBW  SWPB 0
      MOVB 0,@>8C02
      SWPB 0
      MOVB 0,@>8C02
      CLR 4
M      MOV *3+,1
      MOVB 1,@>8C04
      INC 4
      C 15,2
      JEQ N
      SWPB 1
      MOVB 1,@>8C04
      INC 4
      C 4,2
      JNE M
N      B *11

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

DIG   BSS 400

      END
=====..

```

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

```

      DEF SEVHI

SEVHI LI 1,1      [NO. OF DIGITS]
      LI 7,7
      LI 10,10
      LI 9,-2
      MOV 1,@DIG
      CLR 2      [POWER]
      CLR 3      [CARRY]
A      BL @MULT

```

```

        CI 1,6
        JNE A
B       BL @MULT
        BL @TEST
        CI 2,10000
        JNE B
        MOV 9,0
        AI 0,3
        LI 2,10
        LI 3,TXT-2
        BL @VMBW
        DATA >10FF

MULT   INC 2
        CLR 4
        LI 0,DIG [DIGIT STORE]
C       INC 4
        MOV *0,5
        MPY 7,5
        A 3,6
        DIV 10,5
        MOV 5,3
        MOV 6,*0+
        C 4,1
        JNE C
        CI 3,0
        JEQ D
        MOV 3,*0
        CLR 3
        INC 1
D       B *11

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

```

```

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

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

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

VMBW SWPB 0
      MOVB 0,@>8C02
      SWPB 0
      MOVB 0,@>8C02
      CLR 4
M     MOV *3+,1
      MOVB 1,@>8C04
      SWPB 1
      MOVB 1,@>8C04
      INC 4
      C 4,2
      JNE M
N     B *11

TXT  TEXT 'END OF RUN'

DIG  BSS 400

=====
      END

```

even more....>

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

```
.
TXT2 TEXT 'IS'
PROGRAM SEVPAS;

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

PROCEDURE PRINT;
VAR Z:INTEGER;
BEGIN
    WRITELN("SEVEN TO THE POWER OF",POW," IS :");
    WRITELN;
    FOR Z:=MAXDIG DOWNT0 1 DO
    WRITE(SEG(CIS(DIG[Z]),2,1));
    WRITELN;
    WRITELN
END;

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

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

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

```

END;
IF CARRY<>0 THEN
  BEGIN
    DIG[MAXDIG+1]:=CARRY;
    MAXDIG:=MAXDIG+1;
    CARRY:=0
  END;
POW:=POW+1
END;

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

```

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

.
 =====
 end 9/90
 =====

MORE FOLLOWS.....

RAMBLES by Stephen Shaw . for TI*MES January 1991.

Welcome to another issue of RAMBLES and a happy new year to you all. Your comments and letters are always welcome, and your queries too. I have been asked to write some summary type comments on the **various XBs** and also on what a brand new disk owner should be looking for- in this last case my comments are additional and supplemental to Peter Walkers discussion in the last issue..

Firstly, all versions of XB are based on TI XB, and are merely enhancements to it. All versions other than TI seem to have added some form of bit-map graphics- where you can draw in hi resolution..

TI EXTENDED BASIC VERSION 100 is very rare, being the first, and contains a number of bugs, the more serious (for TI) being the incredible ease with which a non-expanded owner can remove the proprietorial "PROTECTION" feature which you can have with XB which prevents programs being saved or listed. Version 100 also worked on the basis that you always had 28 sprites in action, and was consequently rather slow unless (being expanded) you told it otherwise with a CALL LOAD..

=====.

TI VERSION 110 is the more common version and forms the standard for which almost all XB programs are written. The most serious omission being the lack of bit map graphics..

=====.

then came **MECHATRONICS XB** from Germany, a heavy module which tended to get a little hot. It had a very powerful but extremely complicated set up for bit map graphics, which, like TI Logo, was NOT bit map graphics but only redefined characters, consequently you could quickly run out of ink.

=====.

Next on the scene was **MYARC XB** which has been around in a number of versions, the latest and definitive version being Vn 2.12 - this has the disadvantage of REQUIRING the Myarc 512k ram card, as the ExBas itself resides in RAM. This does allow for some increase in speed. . The Myarc XB has good true bit map graphics, variable screen windowing with both 32 and 40 column text modes. .

Myarc XB is in the form of a module which contains a loader plus ram, a disk, and a chip for the ram card..

Added extras include integer and real variables- with integer variables processing a little faster than real ones. You may set a default disk drive and subsequently omit "DSK1." when using OLD and SAVE or RUN. Because more VDP is available, Myarc XB can load AND run those very long TI Basic programs that will NOT load in TI XB, AND the TI Basic extra character sets ARE available in Myarc XB..

CALL CHAR is available from 0 to 255! and you may have 32 sprites. There is a full range of bit map graphics commands- draw, drawto, circle, point, rectangle, fill, write, dcolor, and uniquely you can determine if a pixel is on or off..

You may RUN many program format machine code programs directly. DF80 files written for TI XB may require you to use a utility program to reset VDP registers or may not function at all..

Myarc XB allows you to run many memory image files which otherwise would require a SuperCart (EdAs with 8k ram) as the Myarc module contains ram in the correct location.

Myarc has several unique abilities but does require the Myarc ram card- which is pretty handy anyway!.

The disk library has some utilities for Myarc XB.

=====

The latest Extended Basic version was **TRITON SUPER EXTENDED BASIC** which amalgamated a number of TI XB mods prepared for Gram device owners. Thus you have such useful new commands as CALL CHIMES, CALL ALOCK, CALL ALL, a very useful CALL CAT, CALL CLOCK (runs slow in the UK), GOSUB and GOTO with variables, but perhaps its most important features would be the enhanced editing features, which allow you to move, copy, delete or resequence PORTIONS of an XB program quite rapidly- a major use of my module.

Also it has included on ROM a commercial machine code bit map graphics program called DRAW N PLOT. This one allows you to plot bit map graphics, but only off screen- only when completed can you SHOW them, although there is a cursor on screen when you SHOW which allows direct editing. Graphics can be saved and printed and the format is compatible with TI ARTIST. There is a major conflict in use of VDP between the graphics and an XB program and hang ups may be frequent, especially if you PRINT the pic.

Triton XB requires 32k ram for the graphics feature.

There is a graphics program available (CLASS) for Triton XB..

=====

With all this talk of graphics it is worth adding that the program **THE MISSING LINK** will work with all versions except Myarc to add true bit map mode graphics, with plotting in or out of windows. It does not bat an eye if you draw a line to a point off screen, and can dump a screen to printer at any time. Picture files on disk are compatible with TI Artist. TML is incompatible with the CLOCK command of Triton XB..

TML is now my language of first choice for graphics, but suffers in not being able to tell me if a pixel is on or off.

+++++

NEW DISK DRIVE OWNER? .

The world is your oyster, with many programs available at very low cost which compare well with PC programs costing LOTSD more..

THE standard graphics program is TI ARTIST at about US\$25, well supported with many disks of graphics, compatible with Triton XB, The Missing Link, RLE and so on..

THE MISSING LINK is well worth having especially if you have any interest in Bit Map Graphics. It too is US\$25 and a demo disk is available from the Group Disk Library..

TI BASE is THE database for the TI, a fully featured database at US\$25 which would cost lots more for any other computer. It is very close to dBase II. You can use it very simply or get stuck into its command language, micros, and so on. NOT easy, if you want something easy stick to the limited format PERSONAL RECORD KEEPING module!.

The only significant word processor is TI Writer, which is a line orientated text processor with mail merge capability. It has been much enhanced in the form of FUNLWEB, available from the disk

library, which includes also the TI Assembler plus a disk manager including sector editor.

If you have a disk drive you must get **Funlweb**, and print out all the docs. You may not use all its features! but they are there..

Funlweb provides one "environment" for operating in- another is BOOT (disk library) which gives you an easy to configure menu for your disk, as well as the ability to catalogue and print from the menu screen. .

All disk owners should have MCOPIE (disk library) which will organise your disks in a more efficient manner, making disks with more than 16 files operate more quickly and with less wear and tear..

After that the choice is yours and depends on what YOUR interests (and storage space or pocket money) will run to. The group has an extensive disk library which is well worth perusing- disks are inexpensive, and the catalogue even cheaper- just send three disks and return postage!.

Perhaps a quick list of the programs I have on my main two utility disks may help- remember I have a strong interest in graphics!. In addition to the usual Funlweb package I have:.

T-SHELL, an XB utility which gives you disk catalog and file read functions in the background while you are programming in XB. Available from the disk library..

SQUEEZER a graphics utility which makes 4 tiny TI Artist pics from one full screen one, with different density levels for you to pick the best. (Disk library)..

ARCHIVER- packs and unpacks backup disks for more storage on less disks. Must unpack to use! (Disk library)..

UNBASHER- makes sense of XB programs with multistatement lines taken to excess, helps to unbug them! (Library)..

VDP- allows you to use those early TI Basic programs which use Character Sets 15 and 16, in ExBas (library)..

EXTRACTOR- allows you to quickly take out a part of an XB program for use elsewhere (Triton XB can also be used instead!)(library)..

RLE- graphics display and print program, TI Artist compatible..

ARTDISK and PHOTO from Harry Brashear (commercial) to print TI Artist pics..

TEXTLOADER allows you to take a program in DV80 text format and load it directly into the computer, no editing required. (Library)..

MCPIX and PIXPRO are commercial graphics programs that enable you to see and print pics in McFlix format..

~ ~ ~ ~ ~ ~ ~ ~

Referring to Peters article, the following programs are in the disk library:.

Funlweb; TI Writer Vn 4.5; Multiplan Vn 4 (module rqd); PR Base; Fast-Term; EA5Loader & Textloader; Ace; Systex; Archiver; Star; JBM103; Neatlist; Paragon EDP; .

Full up to date catalogue in return for three disks and return postage only! .

Commercial offerings -which you must buy!- are:.

TI BASE (Texaments or Inscebot US\$25); TI ARTIST PLUS! (same suppliers and price); Graphx (available?); Page Pro 99 (Asgard); Oak Tree DEP (available?); The Missing Link (Texaments US\$25);

Semi commercial offering:.

TELCO. US\$20 +overseas mail (say US\$25 to 30 altogether):.

Charles Earl, 34 McLeod Str, OTTAWA, Ontario, CANADA, K2P 0Z5..

=====
=====.

If you do not have a complete set of TI*MES, back issues are well worth having for reviews of old programs which are still available and which you wont know about unless you get those back issues, some of which are in rare supply (some indeed sold out) so contact Peter Walker today!!!.

ISSUE 8 contains a TI Basic program which demonstrates the power of MiniMemory from Basic including constant music; TI Writer tips; first lesson in TI Forth; .

ISSUE 9 contains a second article on TI Forth; an important article on VDP Registers for use with TI Basic and mini memory (also now applicable to XB using VPOKE utility); single pixel drawing program in TI Basic for mini memory module..

ISSUE 10: Call Sound with TI Forth; An important FASTER ExBas version of the unduly slow SIN function good for 0 to 1.7 radians (say 97 degrees); Speech with TI Forth; Faster trig for TI Forth; .

ISSUE 11: Tutorials on cassette data files and extended basic and TI Forth; connecting a printer to TI PIO socket; and changing a VDP register using only TI Basic and a cassette recorder- no modules required at all! including use of SPRITES (without automotion) in good old TI Basic using CALL CHAR....

ISSUE 14: c99for beginners; map of TI Basic program in VDP RAM and on disk; DIY expansion box; .

ISSUE 16: Editing a CHARA1 disk file; new CALL COINC to see if sprite is near an "on" dot in usual 32 column screen; illustration of speeding up machine code by using scratch pad ram; TI Writer graphics; .

=====
=====.

IS TI EXPANSION EXPENSIVE?.

Certainly it can cost quite a lot of money to chase down an expansion box, with prices in the region of 150-200 for a populated box. However, you must also consider the cost of the programs which you may then purchase, and compare the cost of those programs to similar (or less efficient!) programs for a new computer such as a PC. A PC may be faster and may have more memory, but do you need it? Often all the extra memory is fully occupied by programs which need the memory due to being inefficiently written and/or containing a multitude of features you will never need!.

Consider the software when you consider the hardware!!!.

=====
=====.

WHAT IS EXPANSION???

Here in the UK TI never really tried to sell anything other than the console, and extra hardware was almost never seen in the retailers. With many of our present members in the position of having console only, a brief word of explanation may be handy!.

There are two forms of extra hardware- "stand alone" and "peripheral cards". The latter require that you have the Peripheral Expansion Box, or PEB, a large and very heavy steel box which has a power supply and room for up to 7 cards and a full height disk drive. Many members use two half height drives but you have to watch the power requirements if you do that!.

.
At the "low end" you may purchase modules which have plugs coming out of them which plug directly into a printer (parallel port) for word processing or a spreadsheet. .
The Speech Synthesiser is a small device that plugs into the right hand socket and allows speech with a suitable module- TE2 or XB. .
There are "stand alone" devices which plug into the right hand socket and allow you to connect a printer (usually these are "parallel" devices, the other standard to RS232)..
You may find "stand alone" 32k memory expansion- and if you only add 32k ram, we can supply several machine code programs on cassette for you..

The cards for the PEB offer a wide variety, including the fairly standard printer interface card (RS232 and parallel on board)..

The disk drive controller card, to which you attach up to 3 single sided or double sided single density (TI controller) or single or double density (Myarc and Corcomp controllers) 40 track disk drives. The Myarc controller may also use 80 track drives. It is possible indeed to interface with almost any standard of drive, but the standard required for interchange is single density (90k single sided, 180k double sided)..

The older stand alone TI disk controller will only function single sided single density..

Ram cards allow you to use RAM chips as though they were disk drives, allowing much faster access- and some may allow you to store the contents even when you switch the computer off, using battery power supplies. Power drain may be high, and there is always a risk of corruption with these- always keep a disk backup! Ram cards may also allow you to use them as printer buffers (you fill the ram quickly with text and this is then sent to the printer as required while you carry on with something else). Myarc ram card allows you to use Myarc Extended Basic as well..

There is a USCD Pascal card (very slow language!) and even a FORTI music card allowing more sound channels. GRAM cards are similar to RAM cards but are configured for use with the special TI internal language GPL..

You can even buy a hard disk controller to operate with a hard disk or two! A hard disk is an expensive storage medium which has much more room than a floppy disk, typically these days 20 million bytes is the minimum compared to 90 thousand bytes for a single sided single density floppy disk! The cost per byte is low but when a hard disk becomes inoperable you of course lose rather more data than when a floppy disk decides not to work!..

A MOUSE is a drawing/pointing device which requires special disk software and may -depending on make- plug into the joystick port or RS232 printer port. A TRACKBALL was once made for the joystick port, this was a special version for the TI only made by Wico. .
There was the SUPERSKETCH drawing tablet, which connected a drawing arm through two variable resistances to a drawing program in the module- it was possible to cut off the tablet and attach a mouse instead..

The MBX UNIT was a complex box of tricks which would have allowed use of a more complex (three plane) joystick, and for a few modules allowed rather primitive speech control. It also had a touch tablet face for switching, and offered speech without the speech synth, but only with the special MBX modules.

Consoles may also be modified with the addition of a LOAD INTERRUPT switch which causes the console to tear off to a specified memory address when pressed to do a program located there. You need to put the program there and tell the console where it is! The auto-reset when you insert a module can be switched by a manual switch- you can also prevent a module resetting the console. Ask Mike Goddard for details, which are in his book of things to do to a console (10% ok Mike?)..

*****.

One item of hardware mods omitted from Mikes excellent compendium (plug) is a **Load Interrupt Switch**- this appeared on page 46 of issue 12. As I now have some information on a program many of you have which can use it! here once more are the details.....

At any time you can by means of a hardware switch tell the computer to stop what it is doing and go to a specific memory location and run the program it finds there. This has been used for example for dumping screen displays to printer, although it is also done (better) using a software interrupt, whereby the computer runs the main program and keeps taking time off to say scan the keyboard for a command key- effectivly doing two things at once. .

TI have provided us with a program which can use a load interrupt switch, although they did not tell us about it! It is the DEBUG program that comes with editor assembler, and this is how you utilise it....

Insert the Editor Assembler module, and have the disk with DEBUG on it in drive one. Choose TI Basic and type in and run this program:.

```
100 CALL INIT.
109 REM SEE NOTES BELOW RE LINE 110!.
110 CALL LOAD(8228,96,0) .
120 CALL LOAD("DSK1.DEBUG").
130 CALL LOAD(-4,131,224,112,190).
140 CALL LOAD(8228,160,0).
150 PRINT "PRESS Q THEN ENTER".
160 CALL LINK("DEBUG").
170 END.
```

This loads DEBUG. Enter BYE and select an option of Editor Assembler- eg LOAD AND RUN. While the LOAD AND RUN (D/F 80) program you have selected is running- say a game! - press your load interrupt switch and you will be in DEBUG and can have a look around!..

For the technically minded, pressing the switch causes the computer to do a BLWP to >FFFC where >FFFC contains the WS pointer and >FFFE contains the Program Counter. Line 130 above puts the addresses into these locations..

Now, if you have DEBUG in memory, then load a program, where is debug going to be? Hmmm? The listing above places DEBUG out of the way into ram area >6000 - which you need to provide! Either by modifying an editor assembler module by adding 8k ram or by use of a SUPER SPACE module or similar. Or omit lines 110 and 140, use only relocatable code, and hope there are no clashes... Line 110 resets FFAH (First Free Address in High Memory) to >6000 while line 140 changes it back to >A000..

REMEMBER WHEN...

A year ago, back in issue 27 (page 20) I set a couple of puzzles. No response still to TEST 4 but here is a response to TEST 5 which - as you have forgotten it! - is to set up a circle of things and count them off removing every tenth one. If you start at position 1 and the first you remove is number ten, what are the numbers of the last two left?.

```
.
100 ! JOHN SEAGER
110 OPTION BASE 1 :: DIM ARRAY(100)
120 CALL CLEAR :: INPUT "NUMBER OF ITEMS 3>100:":TOTAL :: IF (TOTAL>2)+(TOTAL<10
1)<>-2 THEN 120
130 FOR I=1 TO TOTAL :: ARRAY(I)=I :: NEXT I
140 OLDPOS,NEWPOS,MISS=1
150 ARRAY(NEWPOS)=ARRAY(OLDPOS)
160 MISS=MISS+1 :: NEWPOS=NEWPOS+1 :: OLDPOS=OLDPOS+1
170 IF OLDPOS>TOTAL THEN TOTAL=NEWPOS-1 :: OLDPOS,NEWPOS=1
180 IF TOTAL=2 THEN 210
190 IF MISS<10 THEN 150
200 OLDPOS=OLDPOS+1 :: MISS=1 :: GOTO 170
210 PRINT "":"REMAINING:" :ARRAY(1),ARRAY(2)
220 END
```

=====.

REMEMBER 1983? Back when TIHOME was extinguished just BEFORE TI pulled the plug, leaving TI owners in the UK with NOTHING to fall back on?.

I was interested in reading a review of US computer magazines by Bill Gaskill to see that amidst the likes of COMPUTE!, ENTHUSIAST 99, 99ER MAGAZINE, and THE SMART PROGRAMMER, there was TIDINGS - apart from the rather glossy IUG magazine, the ONLY UG publication to be listed. How nice I thought..

Now I have a copy of "FREE SOFTWARE FOR YOUR TI99/4A" published sometime in 1984 by ENRICH DIV/OHAUS in the US of A..

This now very out of date book lists the IUG- which produced its last magazine in June 1984- as having 80,000 members (!!!!) -all TI owners - and gives the late Guy Stefan-Romanos San Fransisco address as the IUG Library..

Of much more interest to us is the entry for the UK, where the last issue of TIHOME was dated May 1983! There is an interview with Paul Dicks, indicating a membership of 2000. The article makes remarkable reading for those of us in from the start and is quite interesting history for you newer owners!.

"Although there are now twenty to thirty thousand TI owners in the British Isles, Paul's group is 'virtually the only resource that TI owners have in England'.

"TI users face several obstacles, Paul explains. 'Although the price of the TI home computer is reasonable, the government encourages the BBC Acorn and won't give schools subsidies to purchase TI 99/4 computers. Also, the price of peripherals is almost prohibitive-we're really working on trying to improve this situation. At this time, there are probably only 12 disk drive owners in the entire British Isles! [-ss here- I got my disk drive in Autumn 1981]. Third party software is just beginning to appear [in Nov 82 I was selling software by Not Polyoptics, PS Software, Norton Software, Pewterware, and others! By May 83 to this was added Roach Software, Oak Tree Software, FFF Software, Kuhl Software, Maple Leaf Microware- all advertised in TIDINGS- 5 page ad in May 83 issue!!! and there were also ads by UK houses Lantern and

Apex]- but it's still difficult to even get a copy of 99er magazine from any source other than our service [never mentioned this service in TIDINGS- first I have heard of it!!!ss]..

"To help British TI users, Paul founded a by-mail support group that offers serices and helps TI owners contact others in their area to form local users groups..

"TIHOME offers members a resource for public domain software ,some computer supplies, publications, local contacts, and technical information..

....
"TIHOME publishes a 100-page magazine six times yearly, which members receive as part of their annual dues. The magazine offers editorials, comments, and advertisements by members; detailed software and book reviews; and in depth tutorials and program listings. A recent issue features LINE DRAWING ROUTINE as part of a discussion on TI hi-res graphics, reviews of TI Writer and game software, Beginners Basic, tutorials on LOGO and Assembly Language, comments on computer shows and other activities, an in depth article on concepting and programming adventure games, and reports from local users groups all over the British Isles..

"The TI99/4 is still the best made computer in its price range, even in England, Paul comments. Despite the obstacles, true British pluck has paid off. TIHOMES membership went up from 180 to 2000 in a year, and its still growing!.

[By the time the book was published TIHOME had been dead for at least 7 months! The comment on membership would place the interview at around April 1983- about the time of the very last magazine. Only the last two magazines ran to 102 pages. The LINE DRAWING ROUTINE mentioned as being in a "recent issue" actually appeared in the last one, May 83.]..

A curious interview especially in regard to its unfortunate timing!.

=====
In April 1982 there were 180 members in TIHOME. In August 1990, there were 158 members in TIUGUK..

=====
We have previously advised of the difficulty of using the Editor Assembler module which insists that if you wish to use LIST DEVICE as PIO you must put a full stop after it..

Mike Poskitt has reminded me that if you wish to RUN PROGRAM and type CS1 to load a memory image machine code program then also you must add a full stop at the end, to enable multi-file programs to be loaded (some memory image format machine code programs may be on three four five or more files!)..

And it is appropriate to remind you that if you have a cassette recorder, extended basic, and 32k ram, but no disk system, you can obtain a number of machine code games ON TAPE from our cassette librarian, and if the demand is there, more can be made available. It is possible to load maybe 70% of machine code programs from cassette, including about 90% of games. .

=====
The Southern User Show on Jan 25th is one I shall TRY to get to- weather and rail maintenance permitting. It would help a great deal if anyone travelling by car past Stockport could give a lift, as the rail interchange at London makes my journey into a rather long five hour one (each way!). So all you Southerners, this is your chance to come and ask awkward questions and say what you want in Rambles!.

~~~~~

Asgard News V2 No2 -undated!!! but after July 89 - gives the following version numbers and suggests return of master disk to Asgard with a small fee (typically \$3 or \$4 but add overseas mail):.

High Gravity(2.3); Picasso 2 (2); Page Pro (1.5); TOD Editor(3.0).

There is a list of famous 4Aers including a Belgian, a German, an Italian, and an Australian. You may believe that there are no UK names listed, but under what circumstances could Tony McGovern have been omitted???

There are the usual many complaints of piracy. I would be grateful if anyone can tell me the name or even country of the "Large dealer of TI software in northern Europe" who is involved in large scale commercial piracy of Geneve (or TI) software.

Bobbitt is talking about international court action- which could only be justified by REALLY large scale piracy. It seems to have passed me by...

=====.

### **BASIC PROGRAMMING.**

Nice bit of code here for you to look at. Let's suppose we have CAT=6 and DOG=9, how do we change those round? We could set up a temporary variable like this:.

```
10 CAT=6 :: DOG=9 .
20 PET=CAT :: CAT=DOG :: DOG=PET.
30 PRINT CAT;DOG.
```

But we can save a little variable storage space and produce more interesting code like this...

```
10 CAT=6 :: DOG=9.
20 CAT=CAT+DOG :: DOG=CAT-DOG :: CAT=CAT-DOG.
30 PRINT CAT;DOG.
```

Try it- it works! And saving variable names is of great value if you are programming for THE MISSINK LINK, which only has limited VDP space for variable storage..

Purely in the interests of science, ExBas programmers can tackle this another way, but it is slower:.

```
10 CAT=6 :: DOG=9.
20 CAT=CAT XOR DOG :: DOG=CAT XOR DOG :: CAT=CAT XOR DOG.
30 PRINT CAT;DOG.
```

SEE PAGES 42 TO 44 OF YOUR EXBAS MANUAL and note that this use of XOR will only work if both variables are less than 32767!.

=====.

### **REVIEW: TI BASE UTILITIES:.**

**COMMAND FILE EDITOR and. EVENTS CALENDAR. by WILLIAM GASKILL.**

I received these - as a surprise! - on disk, together with issue V1N5 of THE TI-BASE USER, a newsletter on TI Base published by Bill, for a mere US\$22 per year to UK subscribers (payable in US funds). You may well feel that the \$22 is worth it for these utilities alone. Seeing how Bill puts his command files together is a real education, and may inspire you to bigger and better things with your copy of TI BASE (commercial database published by Inscebot and also available from Texaments)..

In the last issue of TI\*MES you may have seen mention of a utility to merge two command files together- the on board editor only allows a limited length command file, even though TI Base can in fact handle longer. As the 40 column editor allows the use of inverse characters, not available in DV80 command files, the MERGE utility allowed you to prepare several short files and then stick them together for use!.

Having stuck them together how to amend them though! Too long to load into the on board 40 column editor!!!! CFE to the rescue, a well-featured editor especially to amend those merged files- though it will not allow insertion of reverse video characters, it will not delete them either!.

The CFE utility is in EXTENDED BASIC and uses a 40 column machine code utility by Brad Snyder. There are the usual editing facilities, including disk catalogue, save, load, insert/delete line [ a little slow those two! ], insert/delete character etc. You can dip back to XB command mode then type RUN again, should that be necessary though you will need to reload your text. Very nicely written and very useful..

EVENTS CALENDAR is intended as a demonstration of using TI Base and the command files are well worth printing and examining. It is also a simple but effective diary system, with three fields- date (MM/DD/YY), day (MON), and event (65 chars). You can check to see if anything is listed against the date you entered when TI Base was loaded, or against any date, or day, and there is a FIND utility for the EVENT field, which can find HOLIDAY if you search for OLI! There is also a QUERY EDITOR which is worth a close look as it is something you may wish to copy in your own creations..

If you have TI base, the newsletter (with these utilities) has to be an essential purchase, and if you do not have TI Base, why not consider it!.

=====.

**REVIEW- REMIND ME..** JP SOFTWARE. US\$15.00 plus p&p..

My personal life is so complex that my appointments for the year easily fit onto a single sheet of paper, but if I ever needed to keep a diary, this program is one I might consider, certainly if I did not have TI Base and Events Calendar! .

The basis of the program is as follows: You select a month/year and that month is displayed on screen in usual calendar format. You may move around to select a specific day, and check to see what is listed for the day, or edit the entry for the day, which is virtually a full screen of data. .

You may print out the entries for any day or series of days within the displayed month- print outs are in two columns of 40 characters. And there is a splendid FIND routine which operates quite quickly. Lets say you wanted to find every day in the month with the word LUNCH in the data. Type in the word and if LUNCH appears anywhere in the data for a day, a tick will appear in the calendar box, and you can move the cursor to the box and check the full data. If the date has any data there is a blob in the box, empty boxes have no data..

A fairly simple utility- useful utilities generally are simple!- and if you need computer assistance to keep track of yourself, this one is well worth looking at..

=====.

**REVIEW- BOOK- COMPUTERS PATTERN CHAOS AND BEAUTY.**

Clifford A Pickover.. Hardback. 391pages. ISBN 0-86299-792-5.

PUBLISHER: ALLAN SUTTON PUBLISHING. Around 25.00.

Not a cheap book this, but a thoroughly splendid read for anyone interested in the odd graphics programs I have offered from time to time in TI\*MES, including some in this issue from this book..

The 27 pages of references are really useful, especially the details of the very many newsletters and magazines you can look to for more graphics experimentation.

There are 17 chapters, and sub headings include: .

Bach Beethoven The Beatles; Cartoon Faces in Education; FM Synthesis of Speech; Image Processing of the Shroud of Turin; Is there a double smoothly undulating integer?; Wild Monopodial Tendril Plant Growth; A note on osculatory packing.....

Yes there are long words in there! but used with skill and even if you failed math and cannot follow the equations, there is much of general interest.

The "recipes" for programs are very very basic program descriptions, quite adequate to work up into graphics programs you can play with. And there are some beautiful illustrations, in mono and colour (and an address for poster versions!- one appeared in a Daily Telegraph supplement in August)..

This IS a serious book, one worth looking at again and again, which raises all sorts of questions and ideas of a computer related type. If you are not afraid to wet your feet, it is a good book to start with in exploring serious and semi-serious graphics, and maybe to explore the possibilities of computers a little more.

My copy contained an error in Pseudocode 10.1 on page 153..

theta=theta+angle(i) should read:.

theta=theta+angle(j)

Recommended for the adventurous!.

=====

.

### **SOFTWARE REVIEW- ROCK RUNNER** by Eric LaFortune

Distributed by Asgard Software.

REQUIRES: Disk drive, 32k ram, EDITOR ASSEMBLER MODULE (no other module works!)

.

This bit is written by Asgard:.

" You are about to experience one of the most remarkable arcade games ever written for the TI99/4A computer, and certainly the best new game ofor the TI in years. Not only isRock Runner a great game, it isalso a technical feat that pushes the 4a to the limits. Written on a cassette system with the minimemory module, Rock Runner utilizes a graphics mode never before seen in any other TI-99/4A or Geneve program- game or otherwise. Dubbed "half bitmap" this new graphics mode mixes the color capabilities of bit map mode with the speed and flexibility of pattern mode graphics to allow an unprecedented amount of full colour animated graphics! Combine this with excellent sounds, beautiful animation, and fantastic action, and you get a game that puts to shame anything on the Atari, Nintendo, Commodore or anything on any other home computer. Rock Runner is a herculean effort, a stunning introduction to the work of this young author."

This next bit is written by Mike Poskitt:. ....WOW! .

Superlatives aside this IS indeed a very enjoyable game..

The object of Rock Runner is to move your character around the playing area using joystick 1 or 2, picking up a specific number of diamonds to enable you to move to the next level. The playing area scrolls in four directions and the total area is about nine times the visible screen area..

As the character moves he digs a tunnel through which various monsters (from level two upwards) can persue him. His progress is also impeded by rocks which may fall on him if he tunnels beneath them. In later levels other obstacles are encountered (such as expanding acid pools) and you need to drop bombs and lay traps to destroy monsters which will then turn into the diamonds needed to progress to subsequent levels..

If this isnt enough to keep the adrenaline flowing, you are also fighting against a time factor..

There are 15 levels and you may start at any level, which saves the boredom of repeating easier levels at future attempts..

The accompanying instruction leaflet is more than adequate and even offers advice on strategy..

All in all this is an excellent machine code game..

=====.

PART TWO.

This article is a revised version of one which originally appeared in the newsletter of the Australian group TISHUG..

III. SUBPROGRAM PARAMETER LISTS.

In the last chapter we saw how subprograms fitted into the overall workings of Extended Basic. In this chapter we are going to go into the details of writing subprograms. Most of the fiddly detail here concerns the construction of the parameter lists attached to CALL and SUB statements, and some of the little traps you can fall into..

Any information can be transmitted from the CALLing program to the CALLED subprogram via the parameter list, and anything not transmitted this way remains private for each program, with the exception of the DATA pool which is equally accessible to all. If something is mentioned in the parameter list then it is a two-way channel unless special precautions, provided for in XB, are taken. In this case the CALLing program can inform the subprogram of the value of a variable, but not allow the CALLED program to change the value of the variable as it exists in the CALLing program. Arrays however, numeric or string, can't be protected from the follies of subprograms once their existence has been made known to the subprogram through the parameter list..

Let's for starters take a very simple but useful example, where a program needs to invoke a delay at various points. Now some BASICs (and TI LOGO) have a built-in function called WAIT. XB doesn't have this command so you have to program it. It can be done by a couple of CALL SOUNDS or with a FOR-NEXT loop. Let's use an empty loop to generate the delay, about 4 millisecc. each time around the loop, and place the loop in a subprogram..

```
..  
230 CALL DELAY(200).  
..  
670 CALL DELAY(200/D).  
..  
990 CALL DELAY(T).  
..  
..  
3000 SUB DELAY(A):: FOR I=1 TO A :: NEXT I ::SUBEND.
```

This is easier to follow when editing your program than using a GOSUB, and you would need to enter the subroutine in every subprogram since GOSUBbing or GOTOing out of a subprogram is verboten. Also it's less messy than writing the delay loop every time. The example shows several different CALLs to DELAY. The first supplies a number, and when DELAY is CALLED, the corresponding variable in the SUB list, A, is set to 200. This is a particular example of the kind of CALL from line 670 where the expression 200/D is first evaluated before being passed to DELAY to be assigned to A. Variable D might for instance represent the level of difficulty in a game. The CALL from line 990 invokes a numeric variable T, and A in the subprogram is set to the value of T in the CALLing program at the time when the CALL is executed..

Nothing untoward happens to T in this example, as the DELAY subprogram does nothing to change A. Now it may not matter in this instance if T did not retain its value after the subprogram CALL. Suppose instead the delay was to be called out in seconds. Then a subprogram on the same lines DELAYSEC might go.

```
..
230 CALL DELAYSEC(2).
..
990 CALL DELAYSEC(T).
..
4000 SUB DELAYSEC(A):: A=A*250.
4010 FOR I= 1 TO A :: NEXT I :: SUBEND.
```

Now after DELAYSEC has been executed with the CALL from 990, T will have value 250 times its value before the CALL. This won't be a bother if you don't use T again for its previous value. If the CALLing program specifies a numeric constant as in line 230, or a numeric expression, the change in A in the subprogram has no effect on the main program. Suppose you can't tolerate T being changed in line 990 (and this kind of thing can be a source of program bugs). You will find that XB allows for forcing T to be treated as though it were an expression, thus isolating T from alteration by the subprogram, if T is enclosed in brackets in the CALL (not SUB) list. Suppose DELAYSEC is also called from line.

```
970 CALL DELAYSEC((T)).
```

If this CALL in line 970 is followed by the CALL from line 990, T not having been altered in the meanwhile, the same delay will be obtained, but if the order of CALLs were reversed the second delay would be ~250 times the first. In the language of XB this is known as "passing by value" as distinct from "passing by reference". This can only be done for single variables or particular array elements, which behave like simple variables in CALL lists. Whole arrays cannot be passed by value, but only by reference. Expressions and constants can only be passed by value, and its hard to see what else could be done with them. In the example as written, a different variable name was used in the SUB, but if you remember the little experiment in the last chapter you'll see that it wouldn't make any difference if T had been used in the SUB list instead of A.

Now let's complicate things a little by flashing up a message on the bottom line of the screen during the delay interval..

```
..
200 CALL MESSAGE(300," YOUR TURN NOW").
..
270 CALL MESSAGE(T,A$).
..
..
3000 SUB MESSAGE(A,A$):: DISPLAY AT(24,1):A$.
3010 FOR I=1 TO A :: NEXT I :: DISPLAY AT(24,1):"".
3020 SUBEND.
```

The SUB parameter list now contains a numeric variable and a



string variable in that order. Any CALL to this subprogram must supply a numeric value or numeric variable reference, and a string value or string variable reference, in precisely the same order as they occur in the SUB list. In the little program segment above, line 200 passes constants by value and line 270 passes variable references. There is no reason why one cannot be by value and one by reference if so desired..

This process can be extended to any number of entries in the parameter list, provided the corresponding entries in the SUB and CALL lists match up entry by entry, numeric for numeric, string for string. The XB manual does not say so explicitly, but it appears that there is no limit apart from the usual line length problems, on the number of entries in the list. This is the only apparent difference between the parameter list in XB subprograms and the argument lists for CALL LINK("xxxxxx", , ... ) to machine code routines in XB, and Minimemory and E/A Basics..

One little freedom associated with built-in subprograms is not available with user defined subprograms. Some built-ins, such as CALL SPRITE permit a variable number of items in the CALLing list. Parameter lists in user defined subprograms must match exactly the list established by the SUB list or an error "INCORRECT ARGUMENT LIST in ..." will be issued. To compensate for this inflexibility user defined CALLs allow whole arrays, numeric or string, to be passed to a subprogram. .

Complete arrays may be passed by reference only. Individual array elements may be used as if they were simple variables and may be protected from alteration by bracketing in the CALL list..

An array is indicated in the parameter list by the presence of brackets around the array index positions. Only the presence of each index need be indicated as in A(). MATCH(,,) indicates a three-dimensional array MATCH previously dimensioned as such, explicitly or implicitly. Don't leave spaces in the list. If the subprogram needs to know the dimensions of the array these must be passed separately (or as predetermined elements of the array). TI Basics are weaker than some others in that they do not permit implicit operations on an array as a whole, a very annoying deficiency..

Arrays may be DIMensioned within subprograms. This will introduce a new array name to the program, and an array or variable name from the SUB parameter list can't be used or an error message will result. In the following code the main program passes, among other things, an array SC to subprogram BOARD (perhaps a scoreboard writing routine in a game)..

```
100 DIM SC(2,5) :: .....
..
450 CALL BOARD(P,A$( ),SC( )).
..
4000 SUB BOARD(P,A$( ),S( )):: DIM AY(5):: .....
      .... :: CALL REF(P,AY( ),S( )).
..
4080 SUBEND.
5000 SUB REF(V,A( ),B( )):: .... :: SUBEND.
```

BOARD generates internally an array AY() which is passed to another subprogram REF (maybe this resolves ties) along with SC( ), which BOARD knows as S( ), and REF in its turn as B( ) -- the same name could have used in all places. There is however no way that the main program or any subprogram whose chain of CALLs doesn't come from BOARD can know about

the array AY(). This would hold equally well for any variable or array, string or numeric, first defined within BOARD and whose value has not been communicated back to the CALLing program via some other variable mentioned in BOARD's parameter list. .

By following this line of reasoning you can see that there is no way for a subprogram whose chain of CALLs does not come through BOARD to know about array AY(). The only way around this is for AY() to be DIMensioned in the main program (even if this is its only appearance there) and the message passed down all necessary CALL-SUB chains..

This idea of DIMensioning an array only within a subprogram is particularly useful if the array is to READ its values from DATA statements and to be used in the subprogram. This could be done again from any other subprogram needing the same data, without having to pass its name up and down CALL-SUB chains. Remember that DATA statements act as a common pool from which all subprograms can READ. If the array values are the results of computations then these values must be passed through the CALL parameter lists..

For completeness note that , although the XB manual has nothing to say about it, IMAGE statements for formatting PRINT output are accessible from any part of a program in the same way as DATA statements and not confined to the subprograms in which they occur as are DEF entries..

It is not necessary to have any parameters in the list at all. Subprograms used this way can be very helpful in breaking up a long program into more manageable hunks for ease of editing. We shall also see in later chapters that there can be other benefits as well..

One more XB statement for subprograms remains, the SUBEXIT. This is not strictly necessary as it is always possible to write SUBEND on a separate line and to GOTO that line if a condition calling for an abrupt exit is satisfied. Like a lot of the little luxuries of life however, it is very nice to have and makes programs much easier to read and edit. It does not replace SUBEND which is a signal to the XB pre-scan to mark the end of a subprogram. SUBEXIT merely provides a gracious and obvious exit from a subprogram (awkward in some Pascals for instance). The next chapter will demonstrate typical examples of its use. .

---

Note inserted by Stephen: Please do read these articles very carefully. Tony is an excellent ExBas programmer, and these notes should be of assistance to every ExBas owner. Tony wrote this as a result of what he considered to be the appalling quality of some ExBas programs he has seen! Now back to Tony.....

---

#### **IV. USEFUL SUBPROGRAM EXAMPLES.**

In the previous chapter we used as an example a DELAY subprogram which could, with a little refinement, be used to substitute for the WAIT command available in some other languages. You can extend this idea to build up for yourself a library of handy-dandy subprograms which you can use in programs to provide your own extension of the collection of subprograms that XB offers..

For our first example let's take one of the more frustrating things that TI did in choosing the set of built-in subprograms. If you have Minimemory or E/A you know that the keyscan routine, KSCAN, returns keyboard and joystick information simultaneously, while XB forces you to make separate subprogram CALLs, KEY and JOYST, to dig it out..

Since these GPL routines are slow it is difficult to write a fast paced game in XB that treats keyboard and joysticks on an equal footing as is done by many cartridge games. On the other hand in games where planning and not arcade reaction is of the essence there is no reason why the player(s) should be forced to make a once-and-for-all choice and not be able to use either at any stage of the game..

The subprogrammers approach to this problem, once he realised that it can be done (and we have commercial XB games where the writers haven't) is to write the game using joysticks, but replacing JOYST by a user defined sub-program JOY which returns the same values as JOYST even when keys are used..

The first step in telling whether keys or joysticks are being used is to check the keys, and if none have been pressed then to check the joysticks. If a key has been pressed then its return, K, has to be processed so that the direction pads embedded in the keyboard split-scan return the corresponding JOYST value. A subprogram along the lines of the one used in TEX-BOUNCE does just this..

```
900 SUB JOY(PL,X,Y):: CALL KEY(PL,K,ST):: IF ST=0.  
      THEN CALL JOYST(PL,X,Y):: SUBEXIT.  
910 X=4*((K =4 OR K=2 OR K=15)-(K=6 OR K=3 OR K=14)).  
920 Y=4*((K=15 OR K=14 OR K=0)-(K=4 OR K=5 OR K=6)).  
930 SUBEND.
```

PL is the player (left or right joystick or side of the split keyboard) number and is unaltered by the procedure. The simple-minded approach for converting K to (X,Y) values by using the XB logic operators (one of the more annoying omissions from console Basic) seems to work as well as any. The subprogram as written checks the keys first but balances this out by putting the processing load on the key return..

This is as good a time as any to sharpen your own skills by working out alternative versions of this procedure, and also by writing one for mocking up a substitute CALL KEY routine to return direction pad values even if a joystick is used..

[ TEX BOUNCE aka TXB is available from our Disk Library! ]

~~~~~

more....

=====

How many of us have come to the end of another evening of cheerfully going through those familiar cassettes of games and half-finished programs by gazing fondly at our unexpanded system (even that phrase left us feeling somewhat emasculated) and thought: where do I go from here? Should the faithful, old TI be abandoned in favour of an Atari, a Mac, an Apricot - or maybe just one of those nice and cheap Amstrads? Or should I continue with the orphaned machine by splashing out vast sums on a PEB and all the associated items? Either way I'd be able to move into the word processing league that I'd been promising myself for all those years and have something tangible to show from the time spent tapping away in the spare room!.

Being the financially cautious sort (if I wasn't I wouldn't have a TI!) I decided first to consult Francis 'Parco' Parrish to ask for advice on the costs on expanding the TI. The long and the short of my investigations were that a couple of years ago my interest in the TI was re-kindled by buying a module that Parco called MINI-WRITER III, but is labelled and documented as WORDWRITER+..

For around #80 I received the module (of the usual cartridge type), which has a built-in output port, and a made-up cable to connect directly to the parallel input of a printer. It is, in fact, made by DataBioTics Inc. (c)1987 with a program written by Todd Kaplan. Not having seen TI WRITER, the word processing capabilities of this device did nevertheless seem pretty close to the TIW text editor facilities I'd read about..

The first thrill is to enter the world of the 40 column text mode. By using the 40x24row windowing technique you can create your documents up to 80 columns wide. .

On first entering the program you are presented with a prompt line consisting of:.

Edit Tabs Quit Files SearchH Lines Purge.

with the next line as the Command line for your abbreviated choice. To start on your new page you may change the preset tabs and left/right margins, otherwise simply press E for Edit. The arrow keys are 'live' to move the cursor around the screen (ie. function E,S,D,X) and the window is moved around the text by using CTRL E or FCTN 6(window up),CTRL X or FCTN 4(window down),CTRL D or FCTN 5(window right) and CTRL S or FCTN 5(twice) (window left). .

Text can be inserted using fixed or word-wrap (CTRL 0), and the text closed up after an insertion by reformatting either relative to the left margin (CTRL 2) or relative to the cursor with CTRL R. FCTN 8 inserts a blank line whilst CTRL 8 starts a new paragraph by placing a Carriage return and a Line Feed on the next line. Inevitably characters, lines and blocks of text can also be deleted. .

Particularly impressive are the facilities which enable the user to search for a nominated string, ie. a word or phrase, and to replace strings with different ones at all or selected occurrences. Lines of text can be copied and can be easily moved around to different positions on the page..

more....

Text can be saved to cassette or disk. Using the SaveFile -SF- option will store your outpourings in the usual Dis/Var (ie text) mode and by specifying start and finishing line numbers it will save only that part of the file. An interesting feature is that files can also be saved using SaveMem -SM- or SaveFix -SX- options. SM saves in memory image, ie program, mode thereby reducing the space needed, whilst SX stores in Dis/Fix format. Similarly files (or parts of files) can be loaded using LF,LM, or LX..

If you do have the facilities to save to disk another useful capability would be that of Show Directory (SD). This displays all files held on the specified disk, giving details of the stored format, sectors used and size of each file..

Finally, of course, you can print your file through the module interface or through to any other printer you may have connected to your computer. Printer control characters can be entered (escape sequences) to give access to your printer's special commands, eg. underlining or changing fonts. .

It should also be noted that for most of the functions obtained by going through the command line, there is also the option of, what is termed in the perfectly adequate accompanying handbook, the Expert Mode. For example, instead of FNCTN 9 & PF to get to PrintFile, this can be entered merely by keying CTRL P..

So, in summary, if you've been looking for a way to get a lot more out of your dusty, old TI by expanding into the word-processing field, but have shied away from a fully expanded system, this comprehensive module may be just the thing for you.

[Ashley now has a PEB and TI Writer. In order to load the Wordwriter+ DV80 files into TI Writer it is necessary to use the short conversion program in TI*MES #21, or to use VIEW to discover the last line number and then LOAD all lines EXCEPT that one- both these tricks divert TI Writer from the non-compatible Wordwriter+ control codes in the final record.]

.
=====

GRAPHICS PROGRAMS: .

.
While these are all presented ready to run with Extended Basic plus the disk utility THE MISSING LINK (\$25 + \$8 post from Texaments) the programs can be readily converted for use with any other utility that allows bit map graphics such as Triton XB, Myarc XB or the disk utility JBM103 (from disk library)..

.
The first program although one of the smaller ones, produces some most interesting results. I have used a multiplier of 1.2, but at some stages it pays to take it a little more slowly. In effect you are backing away from a wall which is covered with circles, and as you back away, there is a repetitive effect to the patterns. The circular patterns are technically known as aliases, as the computer is not really plotting circles at all.....

.
Reminder: If you write a program with TML loaded and save it to disk it will be in IV254 format. If you load an old program to use with TML and it is not in IV254 format it should be converted using the procedure on page 24 of TML manual.

```

1 ! XB + THE MISSING LINK .
2 ! .
100 REM CIRCLES .
110 REM JE CONNETT/PWH
    MOON/S SHAW 1990 .
120 SIDE=20 ! 10>2000 value .
130 REM .
140 CALL LINK("CLEAR") .
150 FOR I=1 TO 150 :: FOR J=
1 TO 150 .
160 X=I*SIDE/150 :: Y=J*SID
/150:: C=INT(X*X+Y*Y):: D=C/
2 :: IF D-INT(D)>.1 THEN 180 .
170 CALL LINK("PIXEL",I+20,J.
+20) .
180 NEXT J :: NEXT I .
187 !
188 ! next bit just stores .
pictures! omit if required .
189 !
190 PIC=PIC+1 :: A$="DSK2."&.
STR$(PIC) .
200 CALL LINK("SAVEP",A$) .
201 !
202 !
203 ! lets look from a .
little farther away... .
204 !
210 SIDE=SIDE*1.2 :: GOTO 14
0 .
220 END

```

The following program produces a random dot pattern on the screen and then first enlarges it AND rotates it slightly, and then just rotates it. See what effect these minor manipulations have. There are possibilities for some interesting animations here!.

The program to be found in the right hand column is based on work by M Feigenbaum in 1979 and produces some interesting pictures, depending on the limits you set when the program starts. Sometimes looking VERY closely (maximum magnification) will produce an interesting result, at other times it pays to stand back a little.

Towards the right of the overall plot is chaos, on which can be found some overlying non-chaotic detail. The number of "invisible" plots can be varied- failing to plot the first few dots concentrates the display on certain features. Reducing the invisible range will produce a little more detail- if you wish to look at all dots (invisible=0) then try using the 16 colour mode and using a different plot colour for the first 14 plots say- or use all the colours at different plotting stages in the inner loop. More detail will be seen!

Have fun! This program is a true fractal as the same shapes keep reappearing..

.

```

1 ! XB + THE MISSING LINK .
Stephen Shaw Aug 1990 .
2 ! from COMPUTERS PATTERN .
CHAOS AND BEAUTY by Clifford
A Pickover .
3 !
100 REM RANDOM DOT PATTERNS-
MOIRE DOT PATTERN .
110 REM AFTER PICKOVER. .
120 SEED=4 :: RANDOMIZE SEED
130 ! ANGLE OF ROTATION IN .
DEGREES: .
140 AN=2 :: AN=AN*PI/180
150 ! SIZE OF PLOT IN PIXELS.
:- .
160 SZ=140 :: MD=SZ/2
170 FOR I=1 TO 3000
180 RANDX=RND*SZ :: RANDY=RN.
D*SZ .
190 CALL LINK("PIXEL",RANDX+.
5,RANDY+5) .
200 REM
210 RANDXX=1.11*((RANDX-MD)*.
COS(AN)+(RANDY-MD)*SIN(AN))+.
MD .
220 RANDY=1.11*((RANDY-MD)*C.
OS(AN)-(RANDX-MD)*SIN(AN))+M.
D .
230 RANDX=RANDXX .
240 CALL LINK("PIXEL",RANDX+
5,RANDY+5) .
250 NEXT I .
260 REM .
270 CALL LINK("PRINT",180,20.
,"ANY KEY FOR NEXT") .
280 CALL KEY(5,A,B):: IF B<1.
THEN 280 ELSE CALL LINK("C
LEAR") .
290 REM
300 ! different pattern .
310 ! this time we have no
enlargement, just rotation. .
320 SEED=4 :: RANDOMIZE SEED
330 ! ANGLE OF ROTATION IN .
DEGREES: .
340 AN=2 :: AN=AN*PI/180 .
350 ! SIZE OF PLOT IN PIXELS.
:- .
360 SZ=140 :: MD=SZ/2
370 FOR I=1 TO 3000
380 RANDX=RND*SZ :: RANDY=RN.
D*SZ .
390 CALL LINK("PIXEL",RANDX+.
5,RANDY+5) .
400 REM
410 RANDXX=1.00*((RANDX-MD)*.
COS(AN)+(RANDY-MD)*SIN(AN))+.
MD .

```

```

420 RANDY=1.00*((RANDY-MD)*C.
OS(AN)-(RANDX-MD)*SIN(AN))+M.
D .
430 RANDX=RANDXX
.
440 CALL LINK("PIXEL",RANDX+.
5,RANDY+5) .
450 NEXT I
.
460 REM
.
470 CALL LINK("PRINT",180,20.
,"ANY KEY TO END") .
480 CALL KEY(5,A,B):: IF B<1.
THEN 480 ELSE STOP .
490 END
.
.

```

```

.
1 ! XB+ THE MISSING LINK
STEPHEN SHAW AUG 1990 .
2 !
3 ! FEIGENBAUM DIAGRAM
Becker & Dorfler / Leon
Heller :- .
FRACTAL REPORT #10 AUG90 .
4 ! .
100 CALL LINK("CLEAR") .
110 CALL LINK("PRINT",1,25,"
VERTICAL AXIS FROM: TO:")::
CALL LINK("PRINT",21,25,"HOR
IZONTAL AXIS FROM: TO:") .
120 A1$="1.140" :: A2$="1.17
1" :: A3$="2.530" :: A4$="2.
586" :: CALL LINK("INPUT",1,
104,BOTTOM,7,A1$) .
130 CALL LINK("INPUT",11,147
,TOP,7,A2$):: CALL LINK("INP
UT",31,103,LEFT,7,A3$)::CALL
LINK("INPUT",31,147,RIGHT,7,
A4$):: CALL LINK("CLEAR") .
138 !
139 ! Follow these lines:-
140 IF RIGHT-LEFT<.0000001 T
HEN LEFT=LEFT+RIGHT :: RIGHT
=LEFT-RIGHT :: LEFT=LEFT-RIG
HT .
150 IF TOP-BOTTOM<.0000001 T
HEN TOP=TOP+BOTTOM :: BOTTOM
=TOP-BOTTOM :: TOP=TOP-BOTTO
M .
160 IF (TOP=BOTTOM)OR(LEFT=R.
IGHT)THEN 100 .
170 CALL LINK("LINE",1,1,1,2.
40):: CALL LINK("LINE" ,1,24
0,192,240):: CALL LINK("LINE
",192,240,192,1):: CALL LINK
("LINE",192,1,1,1) .
180 WIDE=240 :: DEEP=190
190 CALL LINK("PRINT",3,12,"
<"&STR$(LEFT)&" ^BTM="&STR$(
BOTTOM)):: CALL LINK("PRINT"
,2,170,">"&STR$(RIGHT)) .
200 CALL LINK("PRINT",180,20.
,"TOP="&STR$(TOP)) .
210 VISIBLE=50 :: INVISIBLE=.
50 .
220 SCALE=(RIGHT-LEFT)/WIDE
230 REM
240 FOR RANGE=1 TO WIDE
250 K=LEFT+RANGE*SCALE
260 P=0.3
270 FOR I=0 TO INVISIBLE :: .
P=P+K*P*(1-P):: NEXT I .
280 FOR I=0 TO VISIBLE

```

```

290 ROW=(P-BOTTOM)*DEEP/(TOP.
-BOTTOM) .
300 COL=(K-LEFT)*WIDE/(RIGHT.
-LEFT) .
310 CALL LINK("PIXEL",ROW,CO.
L) .
320 P=P+K*P*(1-P)
.
330 NEXT I
.
340 NEXT RANGE
.
350 GOTO 350
.
360 END
.
.

```



```

.
1 ! XB + THE MISSING LINK
2 ! Stephen Shaw August 90
3 !
4 ! from COMPUTERS PATTERN .
CHAOS AND BEAUTY 1990 by
Clifford A Pickover..
5 !
6 ! Produces patterns from a.
sequence of turtle like
variables- repeats of.
length/angle/number of times .
7 ! Random values below will.
often go off screen as no
scaling is used. Plan and .
fill variable arrays with.
your own values! .
8 !
100 CALL LINK("CLEAR")
110 CT=0
120 RANDOMIZE :: SETS=INT(RN.
D*5+1):: CALL
LINK("CLEAR"):: CALL
LINK("PRINT",180,
50,STR$(SETS)) .
130 REM LENGTH ARRAY
140 DIM L(10)
150 FOR I=1 TO SETS
160 L(I)=(INT(RND*8)+18)
170 NEXT I
180 REM ANGLE ARRAY
190 DIM A(10)
200 FOR I=1 TO SETS
210 ANG=RND*360 :: IF ANG<12.
OR ANG>350 THEN 210 ELSE IF
(ANG>160)AND(ANG<210)THEN 21
0 .
220 A(I)=ANG*PI/180
230 NEXT I
240 REM REPEAT ARRAY
250 DIM R(10)
260 FOR I=1 TO SETS
270 R(I)=INT(RND*5+3)
280 CT=CT+R(I)
290 NEXT I
300 OLDX,X=90 :: OLDY,Y=110
310 THETA=0 :: LL=0
320 FOR I=1 TO CT*1.3333
330 REM
340 FOR J=1 TO SETS ! # SETS
350 REM
360 FOR K=1 TO R(J)
370 CALL LINK("PRINT",2,3,ST.
R$(I)&"|"&STR$(J)&"|"&STR$(K.
)) .
380 LL=LL+1 ! # OF VERTICES
390 REM

```

```

400 THETA=THETA+A(J):: IF TH.
ETA>=2*PI THEN THETA=THETA-2
*PI ! 360 IN RADIANS .
410 X=L(J)*SIN(THETA)+X
.
420 Y=L(J)*COS(THETA)+Y
.
430 CALL LINK("LINE",OLDX,OL.
DY,X,Y):: OLDX=X :: OLDY=Y .
440 NEXT K
.
450 NEXT J
.
460 NEXT I
.
470 CALL LINK("PRINT",180,30.
,"ANY KEY FOR ANOTHER") .
480 CALL KEY(5,N,B):: IF B<1.
THEN 480 ELSE RANDOMIZE :: G
OTO 100 .
490 END

```

```

.
1 ! XB + THE MISSING LINK
2 ! STEPHEN SHAW AUG 1990
3 !
4 ! How random is a sequence
of numbers? This program
uses the RND function but
can take any list of values,
ranged accordingly.
5 !
6 ! from Computers Pattern
Chaos and Beauty 1990 by C A
Pickover.
7 !
8 ! If series is by chance
pattern should stay on
screen, preferably with in
central circle (oval!).
9 !
100 RANDOMIZE
110 SEED=INT(RND*256):: RAND.
OMIZE SEED
120 CALL LINK("PRINT",3,200,.
STR$(SEED))
130 CALL LINK("BOX",1,1,192,.
240):: CALL LINK("CIRCLE",96
,120,35)
140 X,OLDX=96 :: Y,OLDY=120
150 CALL LINK("PIXEL",X,Y)
160 REM
170 C=INT(RND*8)
180 ON C+1 GOSUB 230,240,250.
,260,270,280,290,300,310,320
190 CALL LINK("LINE",OLDX,OL.
DY,X,Y):: OLDX=X :: OLDY=Y
200 GOTO 170
210 REM
220 STOP
230 X=X-3 :: Y=Y+3 :: RETURN
240 Y=Y+3 :: RETURN
250 X=X+3 :: Y=Y+3 :: RETURN
260 X=X-3 :: RETURN
270 X=X+3 :: RETURN
280 X=X-3 :: Y=Y-3 :: RETURN
290 Y=Y-3 :: RETURN
300 X=X+3 :: Y=Y-3 :: RETURN
310 RETURN
320 RETURN
330 END

```

more.....

If you enjoyed the speed of the bifurcation plot already given, try this one, which has similarities to the affine plots we have given in earlier issues- it is slow! It is based on a code in Clifford Pickovers recent book reviewed elsewhere in this issue. .

```

.
100 ! BIFURCATION PLOT .
GENERATOR - PICKOVER .
110 CALL LINK("CLEAR"):: RAN.
DOMIZE .
120 MN=56 :: MX=74 ! FULL-IS.
H PIC WOULD BE 0 TO 125-
CHAOS RULES FROM 59 UP..
130 BETA=5 ! LOW VALUE 3 .
MAKES CHAOS FARTHER AWAY
HIGHER VALUE INCREASES CHAOS .
140 RS=180 ! PLOT RESOLUTION
150 X0=1.95 ! START VALUE FO.
R Xt [t=0] .
160 N=90 ! ITERATION COUNT .
USE HIGHER FOR MORE CHAOS
MORE PATIENCE NEEDED .
170 RSC=RS/(MX-MN):: CS=48
180 FOR LA=MN TO MX STEP (MX.
-MN)/RS .
190 CALL LINK("PRINT",1,120,.
STR$(LA)) .
200 X=X0
210 FOR I=1 TO N+10
220 X=LA*X*(1+X)^(-BETA)
230 IF I<11 THEN 250
240 CALL LINK("PIXEL", (LA-MN.
)*RSC+11,X*CS) .
250 NEXT I
260 NEXT LA
270 CALL LINK("PRINT",1,1,"M.
N ")::CALL LINK("PRINT",1,18
,SEG$(STR$(MN),1,3)).
280 CALL LINK("PRINT",1,48,"
MX "):: CALL LINK("PRINT",1,
75,SEG$(STR$(MX),1,3) ) .
290 CALL LINK("PRINT",1,110,.
"B "):: CALL LINK("PRINT",1,
120,SEG$(STR$(BETA),1,3 )) .
300 CALL LINK("PRINT",1,160,.
"X0 "):: CALL LINK("PRINT",1
,180,SEG$(STR$(X0),1,4) ) .
310 CALL KEY(5,A,B):: IF B<1.
THEN 310 .
320 MN=RND*150 :: MX=MN+RND*.
100 :: BETA=3+RND*3 :: X0=RN
D*3+.01 :: CALL LINK("CLEAR"
):: GOTO 160 .
330 END ..

```

=====.

An earlier program printed patterns of circles using a simple modulus method of deciding if a pixel was on or not. The following program can produce much more complex patterns but again uses a modulus decision making method. The second simpler form uses variable values which produce a very regular pleasing pattern..

This program is to experiment with. Instead of using SIN(X)+SIN(Y) try other formulae! And let me have any good patterns - formulae and variable values..

By all means set R higher, it just takes longer!. And you can plot "to the right" or "to the bottom" by using say FOR I=50 TO R+50 and then when LINKing to PIXEL instead of I use I-49 for the plot. ..

I especially like the patterns when B1=-5, B2=-10, G=66, A=9, and M=2.. What happens to a pattern when you halve or double the value of G and leave other variables unchanged?.

To see how amending G or M varies the pattern, try say putting in a new line:.

```
205 A=INT(I/12)+2 or .
205 M=INT(I/20)+2 .
```

Have fun, experiment, and share your discoveries!.

If you don't have the Missing Link (commercial) program, these programs are easily adopted for ANY program that allows BIT MAP (PIXEL) plotting!.

```
95 ! PATTERN MAKING PROGRAM .
IN TI XB REQUIRES THE
MISSING LINK. Can be re -
written for other bit map
languages/utilities. .
96 ! from egg tile generator.
page 242 of Computers Chaos
Pattern and Beauty by.
Clifford A Pickover.
97 ! for ti by s shaw 1990
98 !
100 RANDOMIZE
109 ! R= number of pixels .
down and across of pic. .
110 R=110
118 ! do not have to use INT.
function except for M. .
119 ! b1,b2=phase shift of .
sine wave, minor adjustments .
120 B1=INT(RND*5)+6*-1
130 B2=INT(RND*9)+12*-1
139 ! g= frequency of sine .
wave, main pattern
determinator. .
140 G=INT(RND*140)+60
149 ! a=degree of disorder, .
low a=good order .
150 A=INT(RND*6)+4
```

```
159 ! m=modulus, degree of .
pattern density- m low=more
pixels on. Also affects
pattern. .
160 M=INT(RND*2)+2
170 CALL LINK("PRINT",20,160.
,STR$(B1)):: CALL LINK("PRIN
T",40,160,STR$(B2)):: CALL L
INK("PRINT",60,160,STR$(G)) .
180 CALL LINK("PRINT",80,160.
,STR$(A))::CALL LINK("PRINT"
,100,160,STR$(M)) .
190 REM
200 FOR I=1 TO R :: FOR J=1 .
TO R .
210 X=B1+(G*I)
220 Y=B2+(G*J)
230 Z=A*(SIN(X)+SIN(Y))
240 C=INT(Z)
250 IF C/M=INT(C/M)THEN CALL.
LINK("PIXEL",I+10,J+10) .
260 NEXT J :: NEXT I
270 CALL LINK("PRINT",172,20.
,"ANY KEY") .
280 CALL KEY(5,Z,X):: IF X<1.
THEN 280 .
290 GOTO 100
```

=====.

Or in a much simpler form:.

```
.
110 R=110
120 B1=-6
130 B2=6
140 G=314
150 A=5
160 M=3
170 !
180 !
190 !
200 FOR I=1 TO R :: FOR J=1
TO R .
210 X=B1+(G*I)
220 Y=B2+(G*J)
230 Z=A*(SIN(X)+SIN(Y))
240 C=INT(Z)
250 IF C/M=INT(C/M)THEN CALL
LINK("PIXEL",I+10,J+10) .
260 NEXT J :: NEXT I
270 CALL LINK("PRINT",172,20.
,"ANY KEY") .
280 CALL KEY(5,Z,X):: IF X<1.
THEN 280 .
290 GOTO 100
```

This next program is definitely fractal in nature but also fairly short, indeed I will first present it as a one liner - this requires THE MISSING LINK but can be translated easily to any other language giving pixel graphics:.

```
10 FOR X=1 TO 170 :: FOR Y=1 TO 180 :: T=((X/32)^2)*Y/32 :: IF T AND 1 THEN
CALL LINK("PIXEL",X,Y) :: NEXT Y :: NEXT X :: GOTO 10.
```

The final GOTO 10 is just to hold the image on screen. Instead of AND 1 try AND 2 or AND 4 or AND 8 - we are using the XB AND operator not as the usual LOGIC operator, but as a BINARY operator, which works ONLY on numbers up to 32700 odd..

The /32 part is just to give us a reasonable pattern, by magnifying the image for any particular set of screen widths/depths- it keeps the numbers below 2^{16} exceeding which will give you a numeric overflow..

Here is a more refined form of the program:.

```
.
90 CALL LINK("CLEAR"):: CALL LINK("PRINT",1,1,"TOP")
100 CALL LINK("INPUT",1,31, TOP$, 4, "1"):: TOP=VAL(TOP$)
105 CALL LINK("PRINT",11,1,"BOTTOM:")
110 CALL LINK("INPUT",11,61, BTM$, 4, "200"):: BOTTOM=VAL(BTM$)
115 CALL LINK("PRINT",21,1,"LEFT")
120 CALL LINK("INPUT",21,35, L$, 5, "1"):: LEFT=VAL(L$)
125 CALL LINK("PRINT",31,1,"RIGHT:")
130 CALL LINK("INPUT",31,51, R$, 5, "200"):: RIGHT=VAL(R$)
140 VS=180/(BOTTOM-TOP):: HS=200/(RIGHT-LEFT)
150 CALL LINK("CLEAR")
160 FOR X=TOP TO BOTTOM STEP 1/VS :: FOR Y=LEFT TO RIGHT STEP 1/HS :: TEMP=INT((
(X/32)^2)*Y/32)
170 IF TEMP/2<>INT(TEMP/2)THEN CALL LINK("PIXEL", (X-TOP+1)*VS, (Y-LEFT+1)*HS)
180 NEXT Y :: NEXT X
190 CALL KEY(5,A,B):: IF B<1 THEN 190 ELSE CALL LINK("CLEAR"):: RUN
```

.
In this version- still not too long is it? - we are checking whether the integer of a number is odd- this is the same as AND 1. This allows us to use numbers well over 2^{16} !!! The refined form allows you to input the number for the top, bottom left and right row, then scales the display to 180 x 200. The display is not very satisfactory for widths/depths of under 20. At a width/depth of 3000 the pattern is decidedly chaotic, using smaller widths you will find that the pattern is almost repetitive and grows smaller as the numbers mount. The very definition of a fractal pattern. ENJOY!.

.
=====

.....

a chaotic program! (Shock, horror!).. **FACES**

.
What this program does is draw a face! The facial features are set by ten parameters. .

.
The program is written in XB for use with The Missing Link but it can easily be modified for use with other graphics programs that allow pixel addressing. The image is small enough to be used with the TI BASIC listing given in an earlier issue..

Reminder: If you write a program with TML loaded and save it to disk it will be in IV254 format. If you load an old program to use with TML and it is not in IV254 format it should be converted using the procedure on page 24 of TML manual.

Notice how much work the computer is doing to draw a smiling or frowning mouth!.

.
90 RANDOMIZE
99 Z=150
100 ! PICKOVER B3 P327 FACES
110 RANDOMIZE
120 FOR I=1 TO 10
130 P(I)=5+INT(RND*6)-INT(RND*6):: CALL LINK("PRINT",I*10+40,120,STR\$(P(I)))
140 NEXT I
150 CALL HEAD(P(1))
160 CALL EYE(P(2),P(7),P(8))
170 CALL PUPIL(P(3),P(7))
180 CALL EYEBROW(P(4))
190 CALL NOSE(P(5))
200 CALL MOUTH(P(9),P(6),P(10))
210 CALL LINK("DUMP") ! to printer
220 FOR I=1 TO 500 :: NEXT I :: CALL LINK("CLEAR"):: RUN
400 SUB HEAD(P1)
401 Z=150
404 DEF CD(X)=COS(X/180*PI)
407 DEF SD(X)=SIN(X/180*PI)
410 EX,EY=0 :: R=30
420 IF P1>5 THEN EX=(P1-5)*2 ELSE IF P1<5 THEN EY=ABS((P1-5)*2)
430 FOR T=1 TO 360
440 X=(R+EX)*CD(T)+50
450 Y=(R+EY)*SD(T)+50
460 IF T=1 THEN CALL LINK("PIXEL",Z-Y,X)
470 IF T<>1 THEN CALL LINK("LINE",Z-OY,OX,Z-Y,X)
480 OX=X :: OY=Y
490 NEXT T
500 SUBEND

! continued....

```

510 SUB EYE(P2,P7,P8)
511 Z=150
514 DEF CD(X)=COS(X/180*PI)
517 DEF SD(X)=SIN(X/180*PI)
520 EX,EY=0 :: R=5
530 IF P2>5 THEN EX=(P2-5)*2 ELSE IF P2<5 THEN EY=ABS((P2-5)*2)
540 P7=P7-5
550 P8=(P8-5)/2
560 FOR T=1 TO 360
570 X=(R+P8+EX)*CD(T)+40-P7
580 Y=(R+P8+EY)*SD(T)+60
590 IF T=1 THEN CALL LINK("PIXEL",Z-Y,X)
600 IF T<>1 THEN CALL LINK("LINE",Z-OY,OX,Z-Y,X)
610 OX=X :: OY=Y
620 NEXT T
630 FOR T=1 TO 360
640 X=(R+P8+EX)*CD(T)+60+P7
650 Y=(R+P8+EY)*SD(T)+60
660 IF T=1 THEN CALL LINK("PIXEL",Z-Y,X)
670 IF T<>1 THEN CALL LINK("LINE",Z-OY,OX,Z-Y,X)
680 OX=X :: OY=Y
685 NEXT T
690 SUBEND
700 SUB PUPIL(P3,P7)
701 Z=150
710 PS=P3/5 :: IF PS=0 THEN PS=.1
720 FOR K=PS TO 0 STEP -.2
730 CALL LINK("CIRCLE",Z-60,40-P7,K):: CALL LINK("CIRCLE",Z-60,60+P7,K):: NEXT K
740 SUBEND
800 SUB EYEBROW(P4)
801 Z=150
810 Y1,Y2=70
820 Y1=(P4-5)+Y1
830 Y2=Y2-(P4-5)
840 CALL LINK("PIXEL",Z-Y1,35)
850 CALL LINK("LINE",Z-Y1,35,Z-Y2,45)
860 CALL LINK("PIXEL",Z-Y2,55)
870 CALL LINK("LINE",Z-Y2,55,Z-Y1,65)
880 SUBEND
900 SUB NOSE(P5)
901 Z=150
910 P5=(P5-5)/2
920 CALL LINK("PIXEL",Z-55,50)
930 CALL LINK("LINE",Z-55,50,Z-45-P5,46)
940 CALL LINK("LINE",Z-45-P5,46,Z-45-P5,54)
950 CALL LINK("LINE",Z-45-P5,54,Z-55,50)
960 SUBEND
1000 SUB MOUTH(P9,P6,P10)
1001 Z=150
1010 P9=P9-5
1020 X1=40-P9 :: Y1=35
1030 X2=60+P9 :: Y2=35
1040 X3=(X2-X1)/2+X1
1050 Y3=(P6-5)+35
1060 REM
1070 FOR K=1 TO 2
1080 IF K=2 THEN Y3=Y3+P10/2
1090 OX=X1 :: OY=Y1

```

! continued....

```

1100 D=X1^2*(X2-X3)+X1*(X3^2-X2^2)+X2^2*X3-X3^2*X2
1110 A=(Y1*(X2-X3)+X1*(Y3-Y2)+Y2*X3-Y3*X2)/D
1120 BB=(X1^2*(Y2-Y3)+Y1*(X3^2-X2^2)+X2^2*Y3-X3^2*Y2)/D
1130 C=(X1^2*(X2*Y3-X3*Y2)+X1*(X3^2*Y2-X2^2*Y3)+Y1*(X2^2*X3-X3^2*X2))/D
1140 REM
1150 FOR I=X1 TO X2
1160 X=I :: Y=A*I^2+BB*I+C :: CALL LINK("LINE",Z-OY,OX,Z-Y,X):: OX=X :: OY=Y ::
NEXT I
1170 NEXT K
1180 FOR K=1 TO 400 :: NEXT K
1190 SUBEND
1200 END

```

.
.=====

MICROpendium will be ending its seventh year of publication with the January 1991 issue, and starting its eighth year with February 1991 making it one of the longest running TI-based publications around. A few user groups have a continuous print record of longer duration (TI*MES is one, but only just, with this our 31st quarterly issue!) but MICROpendium is the clear winner of "commercial" publications..

MICROpendium had a paid circulation of about 3600 at the start of 1990, and if you do not yet subscribe, additional UK subscribers are welcome! The dollar exchange rate at the time of writing makes it a good buy. .

The seamail subscription is US\$30 per year (12 issues, 40 pages each) but seamail may take six weeks, and you may receive one issue before the one that preceded it! Airmail is only US\$42 per year..

P O BOX 1343, ROUND ROCK, TX, USA, 78680.

=====
And for those of you not subscribing (fie) the following news was in the October issue of MICROpendium:.

With both TENEX and TRITON withdrawing from the TI supply scene, only TEXAMENTS and TEXCOMP are left apart from direct-supply producers such as Asgard. .

LOU PHILLIPS is said to have another job and a new baby to take care of, leaving little time for MYARC work- reporting on Delphi, Beery Miller reported from a telephone call with Lou that there are to be no more major soft or hardware items Geneve related, that there are virtually no Geneves available for sale, no news on the HFDC, and HFDC and Geneve cards are being sent out for repair to third parties.

RAY KAZMER and KEN GILLILAND are said to have joined in a new software venture under the name NOTUNG SOFTWARE (any Wagnerites out there?)..

MCCANN SOFTWARE (The Printers Apprentice) have moved to: 4411 North 93rd Street, OMAHA, NE, USA, 68134..

McCann are producing no new TI software; TPA-MDOS version is not compatible with MDOS .97H and no amendments are to be made to TPA-MDOS to make it compatible with any further revisions of MDOS..

.
Had you subscribed you would have known all this is October!.

=====
.

Recalling TI*MES #30 page 53 No 2:.

```
.
100 REM TI*MES COMPETITION.
110 REM ISSUE 30 P53.
120 REM PETER HUTCHINSON.
130 MILES=30.
140 RENT=3.
150 GALLON=2.
160 CHEAP=9999.
170 SPEED=0.
180 FOR SP=1 TO 117.
190 X=SP/3.
200 MPG=40-X.
210 COST=MILES/SP*RENT.
220 FUEL=MILES/MPG*GALLON
230 TOTAL=COST+FUEL.
240 IF TOTAL>=CHEAP THEN 270.
250 CHEAP=TOTAL.
260 SPEED=SP.
270 NEXT SP.
280 PRINT "CHEAPEST SPEED IS".
290 PRINT STR$(SPEED);&"MPH AT A COST OF".
300 PRINT "$"&STR$(CHEAP).
310 END .
```

=====.....
TI WORLD NEWS by Jim Peterson .
AUGUST 1990 .

The Spring 1990 catalog of All Electronics Corp. lists many brand new original TI parts at remarkably low prices, especially on quantity purchases - TI keyboards for \$3.50, R/F modulators for \$5, heat sinks at 3 for \$1, etc. The address is P.O. Box 567, Van Nuys CA 91408, phone 800-826-5432.

Computer Buyer's Guide is now called Vulcan's Computer Monthly. It contains zillions of ads for computer products, and a monthly TI column by Barry Traver - which was missing in August, probably because he didn't get it to them on time. It is still not available on very many newsstands, but the annual subscription price of \$12 (delivery in US!) is a steal. However, I pity the mail carrier - if he ever had to deliver that magazine plus Computer Shopper and the MicroCenter catalog on the same day, he would surely have a hernia. I have suggested to Vulcan's that they buy a computer for their bookkeeping department - they double-billed me for my subscription, and then for my ad!.

.
*****.

DIJIT is no longer taking orders for their AVPC 80-column card, but will continue to support those they have sold..

*****.

Another XBasic programming environment, called Multi-Mode XB, is available from Disk Only Software. It was written by Jean Marleau and, like the others, consists of CALL LINKs to assembly. However, it offers some features not found in any other. For a complete review, see the article by Barry Traver in Vulcan's Computer Monthly for September 1990.

*****.

Paul Scheidemantle is publishing the P&A Express, which appears to be a short newsletter with news about new graphics utilities.

.
*****.

Corcomp has been taken over by International Diversified Technologies Inc.

They will continue to manufacture and repair Corcomp products. For repairs, call (714) 965-4450 to get a return material authorization. This is a tape, so have your information ready.

~~~~~W

This is another old program from mine very own hand! It has an interesting history both pre- and post- its 1982 original publication date! .

Before I got bogged down with a computer I had a serious interest in gaming, including the purchase of copies of old patents of games. Back in the 19th Century (!) there was a patent for a puzzle composed of two interlocking rings!

In a rush for a program for the magazine Computer and Video Games (which in 1982 published lots of listings!) I put together a simulation. Around the same time (this was just post-Rubiks Cube) another Hungarian Endre Pap produced a physical puzzle of two rings, while Israeli Ivan Moscovitch also produced one. Endre had planned versions with more than two rings...(could this have produced Rubiks Clock?)..

My program appeared in C&VG in Nov 1982 with copyright reserved, & Stainless Software sold the program commercially (well it was in the catalogue, I'm not sure if any were sold!)..

I now look through GET MORE FROM THE TI99/4A by Garry Marshall, (c) 1983, and find the same simulation- my program had 38 small segments, Gary's has 30; mine had four colours, Gary's three; mine had bidirectional rotation, Gary's only clockwise. However looking at the code shows that he was not merely altering (making easier!) my program but also making reference to my rather hasty (and hence not the best!) code. My origination WAS acknowledged by the way, but I was not asked for consent nor paid anything!

The program is now public domain, but it pleases me that the program was considered of such value! Here is the original as published in 1982, with no amendments. Key it in carefully and see if you can solve it- or produce some other pleasing pattern....

```
100 DIM RA(20),CA(20),RB(20),CB(20)
120 CALL CHAR(128,"FF80BE8888B880FF")
140 CALL CHAR(129,"017D417D057D01FF")
160 CALL CLEAR
180 CALL HCHAR(20,26,128)
200 CALL HCHAR(21,26,129)
220 FOR C=96 TO 120 STEP 4
240 CALL CHAR(C,"FF818181818181FF")
260 NEXT C
280 PRINT "HUNGARIAN SQUARES"
300 CALL COLOR(10,2,16)
320 CALL COLOR(11,2,12)
340 CALL COLOR(12,2,6)
360 CALL COLOR(13,2,13)
380 CALL COLOR(14,5,14)
400 PRINT " "(C) STEPHEN SHAW 1982": : "A MANIPULATIVE PUZZLE"
420 PRINT " ": : "PRESS ANY KEY TO CONTINUE"
440 FOR C=1 TO 6 ! continued....
```

```

460 RA(C)=1
480 RB(C)=5
500 CA(C)=C*2+1
520 CB(C)=C*2+7
540 NEXT C
560 CALL KEY(0,A,B)
580 IF B=0 THEN 560
600 CALL CLEAR
620 PRINT "TWO LARGE SQUARES,A & B ARE FORMED OF SMALL COLORED      SQUARES"
630 PRINT "THERE ARE TEN SMALL SQUARES OF TWO COLORS AND NINE OF    TWO OTHER COL
ORS"
640 PRINT "THE OBJECT IS TO FORM ONE OFTHE LARGE SQUARES OF":"ALTERNATELY COLORE
D SMALL"
650 PRINT "SQUARES-USING THE COLORS OF WHICH THERE ARE TEN SMALL    SQUARES"
660 PRINT "AND ALSO FORMING THE OTHER  LARGE SQUARE OF ALTERNATE    SMALL SQUARES
"
670 PRINT " ":"OR YOU CAN TRY TO FORM YOUR OWN PATTERNS!"
700 PRINT " ":"PRESS ANY KEY TO CONTINUE"
720 CALL KEY(0,A,B)
740 IF B=0 THEN 720
760 CALL CLEAR
780 FOR C=1 TO 4
800 RA(6+C)=C*2+1
820 RB(6+C)=C*2+5
840 CA(6+C)=13
860 CB(6+C)=19
880 RA(C+16)=11-C*2
900 RB(C+16)=15-C*2
920 CA(C+16)=3
940 CB(C+16)=9
960 NEXT C
980 CALL HCHAR(4,29,128)
1000 CALL HCHAR(5,29,129)
1020 FOR C=1 TO 6
1040 RA(C+10)=11
1060 RB(C+10)=15
1080 CA(C+10)=15-2*C
1100 CB(C+10)=21-2*C
1120 NEXT C
1140 PRINT "PRESS A OR B...<"
1160 PRINT "PRESS 1 TO MOVE CLOCKWISE":"& 2 TO MOVE ANTICLOCKWISE"
1180 FOR C=1 TO 7
1200 CALL HCHAR(RA(C),CA(C),96)
1220 CALL HCHAR(RB(C),CB(C),104)
1240 NEXT C
1260 CALL HCHAR(RB(8),CB(8),112)
1280 FOR C=9 TO 17
1300 CALL HCHAR(RA(C),CA(C),120)
1320 CALL HCHAR(RB(C),CB(C),112)
1340 NEXT C
1360 FOR C=19 TO 20
1380 CALL HCHAR(RA(C),CA(C),96)
1400 CALL HCHAR(RB(C),CB(C),104)
1420 NEXT C
1440 CALL HCHAR(RA(18),CA(18),96)
1460 CALL HCHAR(3,5,ASC("A"))
1480 CALL HCHAR(13,17,ASC("B"))
1500 REM ACTUAL MOVEMENT      HERE      *****      ! continued.....

```

```

1520 CALL HCHAR(21,18,30)
1540 CALL HCHAR(21,18,32)
1560 CALL KEY(0,A,B)
1580 IF B=0 THEN 1520
1600 IF A=ASC("A")THEN 1660
1620 IF A=ASC("B")THEN 2220
1640 GOTO 1500
1660 CALL HCHAR(22,31,30)
1680 CALL HCHAR(22,31,32)
1700 CALL HCHAR(23,31,30)
1720 CALL HCHAR(23,31,32)
1740 CALL KEY(0,A,B)
1760 IF B=0 THEN 1660
1780 REM MOVE SQUARE A
1800 IF A=ASC("1")THEN 1860
1820 IF A=ASC("2")THEN 2040
1840 GOTO 1660
1860 REM MOVE A CLOCKWISE
1880 CALL GCHAR(RA(1),CA(1),TEMPA)
1900 FOR C=1 TO 19
1920 CALL GCHAR(RA(C+1),CA(C+1),TEMPB)
1940 CALL HCHAR(RA(C+1),CA(C+1),TEMPA)
1960 TEMPA=TEMPB
1980 NEXT C
2000 CALL HCHAR(RA(1),CA(1),TEMPA)
2020 GOTO 1500
2040 REM MOVE A ANTICLOCKWISE
2060 CALL GCHAR(RA(20),CA(20),TEMPA)
2080 FOR C=19 TO 1 STEP -1
2100 CALL GCHAR(RA(C),CA(C),TEMPB)
2120 CALL HCHAR(RA(C),CA(C),TEMPA)
2140 TEMPA=TEMPB
2160 NEXT C
2180 CALL HCHAR(RA(20),CA(20),TEMPA)
2200 GOTO 1500
2220 CALL HCHAR(22,31,30)
2240 CALL HCHAR(22,31,32)
2260 CALL HCHAR(23,31,30)
2280 CALL HCHAR(23,31,32)
2300 CALL KEY(0,A,B)
2320 IF B=0 THEN 2220
2340 IF A=ASC("1")THEN 2400
2360 IF A=ASC("2")THEN 2580
2380 GOTO 2220
2400 REM MOVE B CLOCKWISE
2420 CALL GCHAR(RB(1),CB(1),TEMPA)
2440 FOR C=1 TO 19
2460 CALL GCHAR(RB(C+1),CB(C+1),TEMPB)
2480 CALL HCHAR(RB(C+1),CB(C+1),TEMPA)
2500 TEMPA=TEMPB
2520 NEXT C
2540 CALL HCHAR(RB(1),CB(1),TEMPA)
2560 GOTO 1500
2580 REM MOVE B ANTI-          CLOCKWISE
2600 CALL GCHAR(RB(20),CB(20),TEMPA)
2620 FOR C=19 TO 1 STEP -1
2640 CALL GCHAR(RB(C),CB(C),TEMPB)
2660 CALL HCHAR(RB(C),CB(C),TEMPA)      ! continued.....

```

```

2680 TEMPA=TEMPB
2700 NEXT C
2720 CALL HCHAR(RB(20),CB(20),TEMPA)
2740 GOTO 1500
2760 END
2780 REM (C)1982
2800 REM BY STEPHEN SHAW
2820 REM 10 ALSTONE ROAD
2840 REM STOCKPORT  CHESHIRE
2860 REM SK4 5AH
2880 REM
2900 REM *****
2920 END

```

=====

How to describe this **graphics program**? We are if you like preparing a potato (or lino) cut pattern composed of two quarter circles. These can be printed in two possible ways, one being a 90 degree rotation of the other. The computer program places them randomly to produce some interesting patterns. The program given uses RND to select numbers between 0 and 1, which are then converted to 0 and 1 values. There are two fairly obvious shapes created- circles and dumbbells (two circles connected by a thin lobe). If our RND truly is random, we may expect that if we count the total number of tiles then there will be one eightieth that number of dumbbells, and a trifle over one twentieth that number of stand alone circles. .

Instead of using RND we could say PEEK into memory locations and convert the values we find to binary (011100 etc) and use those 1's and 0's to produce a quite different pattern! Or try various uses of RND eg RESULT=(RND+RND/2) or RESULT=RND\*RND and so on. Sent print outs to me if you find anything pleasing!!.

```

100 ! TRUCHET TILE GENERATION      from Pickover p.333      .
110 ! for XB+TML                    .
120 ! S SHAW   OCT 1990              .
130 ! easily converted to   any other pixel graphic language/utility.. .
140 !                                .
150 RANDOMIZE                        .
160 XOFF,YOFF=0.0                    .
170 DX,DY=12 :: R=DX/2                .
180 JUMP=180/DX :: OLDDATA=0          .
190 P1,P0=0.5000 ! 50/50              .
200 REM                               .
210 FOR J=1 TO 100*100 ! 100X100 DISPLAY .
220 RESULT=RND                        .
230 IF OLDDATA=0 THEN IF RESULT<P0 THEN D=0 ELSE D=1 .
240 IF OLDDATA=1 THEN IF RESULT<P1 THEN D=1 ELSE D=0 .
250 IF D=1 THEN GOSUB 340 ! UP        .
260 IF D=0 THEN GOSUB 510 ! DOWN     .
270 XOFF=XOFF+DX                      .
280 IF (J/JUMP)=INT(J/JUMP)THEN YOFF=YOFF+DY :: XOFF=0 .
290 IF YOFF>200 THEN GOTO 290 ! KEEP WITHIN SCREEN BOUNDARY .
300 OLDDATA=D                          .
310 NEXT J                             .
320 STOP                               .
330 REM                               .
340 CALL LINK("PRINT",180,20,"UP      ") .
350 IF OLDDATA=0 THEN XOFF=XOFF+DX     ! more -->

```

```

.
360 FOR I=90 TO 180 STEP 10
370 X=R*COS(I/180*PI)+XOFF
380 Y=R*SIN(I/180*PI)+YOFF
390 IF I=90 THEN OLDX=X :: OLDY=Y ELSE CALL LINK("LINE",OLDX,OLDY,X,Y).
:: OLDX=X :: OLDY=Y .
400 NEXT I
410 REM
420 REM
430 XC=XOFF-DX :: YC=YOFF+DY
440 FOR I=270 TO 360 STEP 10
450 X=R*COS(I/180*PI)+XC
460 Y=R*SIN(I/180*PI)+YC
470 IF I=270 THEN OLDX=X :: OLDY=Y ELSE CALL LINK("LINE",OLDX,OLDY,X,Y).
:: OLDX=X :: OLDY=Y .
480 NEXT I
490 RETURN
500 REM
510 CALL LINK("PRINT",180,20,"DOWN")
520 IF OLDDATA=1 THEN XOFF=XOFF-DX
530 FOR I=0 TO 90 STEP 10
540 X=R*COS(I/180*PI)+XOFF
550 Y=R*SIN(I/180*PI)+YOFF
560 IF I=0 THEN OLDX=X :: OLDY=Y ELSE CALL LINK("LINE",OLDX,OLDY,X,Y)::.
OLDX=X :: OLDY=Y .
570 NEXT I
580 REM
590 XC=XOFF+DX :: YC=YOFF+DY
600 FOR I=180 TO 270 STEP 10
610 X=R*COS(I/180*PI)+XC
620 Y=R*SIN(I/180*PI)+YC
630 IF I=180 THEN OLDX=X :: OLDY=Y ELSE CALL LINK("LINE",OLDX,OLDY,X,Y).
:: OLDX=X :: OLDY=Y .
640 NEXT I
650 RETURN
660 END
.

```

Truchet you ask? (OK you dont ask...). **This idea dates back to... 1704!**.

It is fun to save this as TI ARTIST pics and then use the FILL to prove that this is a two colour pattern only. Also, by its nature, you can always align one printout with another, top or bottom or sides, to make your own very unique wallpaper, gift wrap, book covering etc. The pattern is also suitable for an all-over fabric print and can be used with either very strongly contrasting colours or with two close shades of the same colour for a more subtle effect.. Have fun!.

=====

more follows....

. (This was SUBMITTED 12/90, not used, and resubmitted 3/91)

### **BULL RUN: .**

A program loosely based on a simple strategic war simulation by Tim Hartnell, converted to a one player TI game by Stephen Shaw, 1990..

.  
This strategic game is played out on a 6x6 grid, with rows identified by the letters A/F and columns identified by the number 1 to 6..

.  
At game start the human player allocates a total of 100 men around the 36 squares which may have from 0 to 9 men in each..

.  
Once the human player has allocated all his men the computer will allocate his (without referring to the human placement)..

.  
The computer, having the handicap of having no brains! will always go first..

.  
Each player in turn selects one square as the centre of a battle, which will rage over all adjacent squares- a maximum of nine squares in the centre, or if the battle is centred on one of the four corner squares, only four squares are involved. Six squares are involved for any other edge square..

.  
The total men for each player is calculated for all these active squares, and the winner has the most! The loser sees all his men in those active squares wiped out. .

.  
Play terminates if one player has more than double the number of men left than his opponent, or after 18 battles. The winner controls the most squares at game end, and could do so with less men than the loser has..

.  
It is easy to play for a draw on territory, so as a penalty, in the event of each player controlling the same territory as the other the game is awarded to the computer. .

.  
As written winning can be difficult! Various adjustments can be made to make the computer play more intelligently! or less so. I shall leave that to you dear reader! Enjoy..

.  
Basic programmers may wish to note how the board is quickly placed on screen, and how player input is handled..

.  
Stephen Shaw.

```
100 REM
110 REM based on FIRST BULL RUN by Tim Hartnell 1984
120 REM for TI99/4A by s shaw october 1990
130 REM winner has most territory at end not most troops
140 REM
150 REM JUST NEEDS EX BAS!
160 REM
170 REM PLACE UP TO 9 TROOP IN EACH TERRITORY TROOPS IN BLOCK OF
9 ARE COUNTED FOR RESULT
180 REM
190 CALL CLEAR :: GOSUB 290 :: T=0 :: RANDOMIZE
200 GOSUB 350
210 GOSUB 610 :: IF FLAG=1 THEN 1110
220 GOSUB 820
230 GOSUB 610 :: IF FLAG=1 THEN 1110
240 GOSUB 730 ! continued....
```

```

250 BATL=BATL+1 :: IF BATL=9 THEN 1110 ELSE 210
260 ,5
270 REM HUMAN SELECTS
280 REM
290 REM SETUP
300 CALL COLOR(11,15,15,12,16,16)
310 A$="ppppzzzzppppzzzzppppzzzz" :: B$="zzzzppppzzzzppppzzzzpppp"
320 FOR T=1 TO 14 STEP 6 :: DISPLAY AT(T,1):A$:A$:A$:B$:B$:B$ :: NEXT T
330 FOR T=1 TO 6 :: DISPLAY AT(T*3-1,25):CHR$(64+T) :: DISPLAY AT(19,T*4-
2):CHR$(48+T) :: NEXT T
340 RETURN
350 REM PLACE
360 DISPLAY AT(20,1):"YOU MUST PLACE 100 MEN" :: TOT=100
370 DISPLAY AT(21,1):"ROW COL MEN"
380 CALL HCHAR(21,7,32) :: CALL KEY(5,A,B) :: CALL HCHAR(21,7,30) :: IF A<65 OR
A>70 THEN 380 ELSE ROW=A-64 :: CALL HCHAR(21,7,A)
390 CALL HCHAR(21,15,32) :: CALL KEY(5,A,B) :: CALL HCHAR(21,15,30) :: IF A<49
OR A>54 THEN 390 ELSE COL=A-48 :: CALL HCHAR(21,15,A)
400 CALL KEY(5,A,B) :: IF A>20 THEN 400
410 CALL HCHAR(21,23,32) :: CALL KEY(0,A,B) :: CALL HCHAR(21,23,30) :: IF A<48
OR A>57 THEN 410 ELSE MEN=A-48 :: CALL HCHAR(21,23,A)
420 IF ARRAY(ROW+1,COL+1)>0 THEN TOT=TOT+ARRAY(ROW+1,COL+1) ::
ARRAY(ROW+1,COL+1)=0
430 IF TOT-MEN<0 THEN 410
440 DISPLAY AT(22,1):TOT-MEN;" LEFT TO PLACE" :: TOT=TOT-MEN ::
ARRAY(ROW+1,COL+1)=MEN
450 CALL HCHAR(ROW*3-1,COL*4,A)
460 IF TOT=0 THEN 480
470 GOTO 370
480 COMP=100 :: FOR ROW=1 TO 6
490 FOR COL=1 TO 6
500 A=INT(RND*9*RND+1) :: IF COMP-A<0 THEN 500
510 COMP=COMP-A :: CALL HCHAR(ROW*3-1,COL*4+2,A+48) :: DISPLAY
AT(24,1):"COMPUTER HAS ";COMP;" LEFT"
520 GRID(ROW+1,COL+1)=A :: IF COMP=0 THEN 590
530 NEXT COL
540 NEXT ROW
550 IF COMP=0 THEN 590
560 FOR COL=7 TO 2 STEP-1 :: FOR ROW=2 TO 7 :: IF GRID(ROW,COL)<9 THEN
GRID(ROW,COL)=GRID(ROW,COL)+1 :: COMP=COMP-1
570 CALL HCHAR((ROW-1)*3-1,(COL-1)*4+2,48+GRID(ROW,COL)) :: IF COMP=0 THEN 590
580 NEXT ROW :: NEXT COL
590 CALL HCHAR(20,1,32,160)
600 RETURN
610 REM CHECK END?
620 YOURTOT,COMPTOT,FLAG=0
630 FOR ROW=2 TO 7
640 FOR COL=2 TO 7
650 YOURTOT=YOURTOT+ARRAY(ROW,COL)
660 COMPTOT=COMPTOT+GRID(ROW,COL)
670 NEXT COL :: NEXT ROW
680 IF YOURTOT<16 OR COMPTOT<16 THEN FLAG=1
690 IF YOURTOT<COMPTOT/2 THEN FLAG=1
700 IF COMPTOT<YOURTOT/2 THEN FLAG=1
710 DISPLAY AT(21,1):" ":YOURTOT;" COMP=";COMPTOT :: CALL DELAY
720 RETURN
730 REM HUMAN SELECTS
740 DISPLAY AT(22,1):"BATTLE CENTRE: ROW COL":YOURTOT;COMPTOT ! more---->

```



```

750 CALL HCHAR(22,21,30) :: CALL KEY(5,A,B) :: CALL HCHAR(22,21,32) :: IF A<65
OR A>70 THEN 750 ELSE ROW=A-64 :: CALL HCHAR(22,21,A)
760 CALL HCHAR(22,27,30) :: CALL KEY(5,A,B) :: CALL HCHAR(22,27,32) :: IF A<49
OR A>54 THEN 760 ELSE COL=A-48 :: CALL HCHAR(22,27,A)
770 ROW=ROW+1 :: COL=COL+1
780 IF BATTLE(ROW,COL)=1 THEN DISPLAY AT(21,1):"FOUGHT THERE ALREADY" :: CALL
DELAY :: GOTO 740
790 BATTLE(ROW,COL)=1
800 GOSUB 930
810 RETURN
820 REM COMP SELECTS
830 FOR A=1 TO 16 :: ROW=INT(RND*4+3) :: COL=INT(RND*4+3) :: GOSUB 1210
840 IF CT>UT AND BATTLE(ROW,COL)=0 AND UT>3 THEN 900
850 NEXT A
860 FOR ROW=7 TO 2 STEP-1 :: FOR COL=2 TO 7 :: GOSUB 1210
870 IF CT>UT AND UT>0 AND BATTLE(ROW,COL)=0 THEN 900
880 NEXT COL :: NEXT ROW
890 ROW=INT(RND*6+2) :: COL=INT(RND*6+2) :: IF BATTLE(ROW,COL)=1 THEN 890
900 DISPLAY AT(22,1):"COMPUTER FIGHTS AT ";CHR$(ROW+63);COL-1:" ":" "
910 IF BATTLE(ROW,COL)=1 THEN 820
920 BATTLE(ROW,COL)=1 :: CALL DELAY
930 REM LETS COUNT
940 GOSUB 1200
950 DISPLAY AT(21,1):" ":"YOU:";UT;" COMPUTER:";CT
960 CALL DELAY
970 IF CT=UT THEN DISPLAY AT(23,1):" -NO BATTLE-" :: GOTO 1100
980 IF UT>CT THEN DISPLAY AT(23,1):"YOU WIN" ELSE DISPLAY AT(23,1):"YOU LOSE"
990 IF UT>CT THEN GRID(ROW-1,COL-1),GRID(ROW-1,COL),GRID(ROW-
1,COL+1),GRID(ROW,COL-1),GRID(ROW,COL+1),GRID(ROW+1,COL),GRID(ROW+1,COL-1)=0
1000 IF UT>CT THEN GRID(ROW,COL),GRID(ROW+1,COL+1)=0
1010 IF CT>UT THEN ARRAY(ROW-1,COL-1),ARRAY(ROW-1,COL),ARRAY(ROW-
1,COL+1),ARRAY(ROW,COL-1),ARRAY(ROW,COL+1),ARRAY(ROW+1,COL+1)=0
1020 IF CT>UT THEN ARRAY(ROW,COL),ARRAY(ROW+1,COL-1),ARRAY(ROW+1,COL)=0
1030 CALL DELAY
1040 FOR ROW=2 TO 7 :: FOR COL=2 TO 7
1050 CALL HCHAR((ROW-1)*3-1,(COL-1)*4,ARRAY(ROW,COL)+48)
1060 CALL HCHAR((ROW-1)*3-1,(COL-1)*4+2,GRID(ROW,COL)+48)
1070 NEXT COL :: NEXT ROW
1080 DISPLAY AT(22,1):"CALCULATED NEW TROOPS":"" :: CALL DELAY :: CALL
HCHAR(22,1,32,32)
1090 RETURN
1100 CALL DELAY :: RETURN
1110 REM FINAL PRINT OUT
1120 DISPLAY AT(21,1):"":"FINAL TROOPS:";"YOU:";YOURTOT;" COMPUTER";COMPTOT
1130 FOR ROW=2 TO 7 :: FOR COL=2 TO 7
1140 IF ARRAY(ROW,COL)=0 THEN CW=CW+1
1150 IF GRID(ROW,COL)=0 THEN UW=UW+1
1160 NEXT COL :: NEXT ROW
1170 IF UW>CW THEN DISPLAY AT(24,1):"YOU WON MOST TERRITORY" ELSE DISPLAY
AT(24,1):"COMPUTER WON MOST TERRITORY"
1180 DISPLAY AT(20,1):"PRESS ANY KEY TO RE-RUN" :: CALL KEY(5,A,B) :: DISPLAY
AT(20,1):"PRESS ___ KEY TO RE-RUN" :: IF B<1 THEN 1180 ELSE RUN
1190 CALL KEY(5,A,B) :: IF B>0 THEN RUN
1200 REM CHECK COMP CHOICE
1210 UT,CT=0
1220 CT=GRID(ROW-1,COL-1)+GRID(ROW-1,COL)+GRID(ROW-1,COL+1)+GRID(ROW,COL-
1)+GRID(ROW,COL)+GRID(ROW,COL+1)+GRID(ROW+1,COL+1)+GRID(ROW+1,COL)
1230 CT=CT+GRID(ROW+1,COL-1) ! still more ---->

```

```

1240 UT=ARRAY(ROW-1,COL-1)+ARRAY(ROW-1,COL)+ARRAY(ROW-1,COL+1)+ARRAY(ROW,COL-1)
1250 UT=UT+ARRAY(ROW,COL)+ARRAY(ROW,COL+1)+ARRAY(ROW+1,COL+1)+ARRAY(ROW+1,COL)
+ARRAY(ROW+1,COL-1)
1260 RETURN
1270 SUB DELAY :: FOR T=1 TO 700 :: NEXT T :: SUBEND

```

=====

Jim Swedlow writes:

## **BASIC PROGRAMMING TIPS**

### **ASC and SEG\$**

A common coding for determining the ASCII value of the first character of a string is:

```
ASC(SEG$(A$,1,1))
```

This can be simplified by omitting the SEG\$ function:

```
ASC(A$)
```

ASC always returns the ASCII value of the first character of the string. You will get an error if A\$ is a null string (if A\$="").

Suppose you want to lop off the first four characters of a string. You might do this:

```
A$=SEG$(A$,5,LEN(A$)-4)
```

Our 4As, however, do not compare the third value in the SEG\$ function (the length of the new string) to the length of the old string. Therefore, this works just as well:

```
A$=SEG$(A$,5,255)
```

Use 255 as that is the maximum length of a string variable. If the length of A\$ is less than 5 even if it is zero the new A\$ will be a null string (but no error).

---

### **POStion**

One way to ask for a menu selection is to ask the user to input the first letter of his/her choice. Suppose that the options were <C>hange, <P>rint or <Q>uit.

You might do this:

```

190 ACCEPT AT(10,10)SIZE(-1)
    VALIDATE("CPQ"):A$ ::
    IF A$="C" THEN 230 ELSE
    IF A$="P" THEN 340 ELSE
    IF A$="Q" THEN 980 ELSE 190

```

A simpler way would be to use POS:

```

190 ACCEPT AT(10,10)SIZE(-1)
    VALIDATE("CPQ"):A$ ::
    ON POS("CPQ",A$,I) GOTO
    230,340,960

```

If your user inputs a null, the POS function will return a value of 1 and control will transfer to the first line in the GOTO list. If this is a problem, do it this way:

```

190 ACCEPT AT(10,10)SIZE(-1)
    VALIDATE("CPQ"):A$ ::
    IF A$="" THEN 190 ELSE
    ON POS("CPQ",A$,I) GOTO
    230,340,960

```

(Source: a Tigercub program)

----

## DIMs and SUBPROGRAMS

Will this program work?

```
10 DIM A(5)
20 CALL SETUP(A())
30 SUB SETUP(B())
40 FOR I=1 TO 10
50 B(I)=I
60
NEXT I
70 SUBEND
```

Answer: It will crash in line 50 when I=6. Once A has been DIMensioned in line 10. The process of calling SETUP in line 30 transfers the DIMension to B within SETUP. In essence, there has been a DIM B(5) inside SETUP.

Note that a STOP is not needed at the end of line 20. Your 4A will not execute a SUBPROGRAM unless it is CALLED.

-end 12/90

SUBMITTED MARCH 1991

3/91

RAMBLES by Stephen Shaw. For TI\*MES April 1991 .

.  
Greetings and welcome once more. Your enquiries are welcome, on any aspect of TI computing (except assembly on which I am an illiterate!), comments on and requests for Rambles are very welcome too. .

### EXTENDED BASIC BUG: .

I am grateful to Bruce Harrison of Harrison Software for pointing this one out in the December issue of Micropendium- if you have a disk system, and have a disk of XB programs which has a program called LOAD, your system will automatically load the LOAD program from DSK1 when you select Extended Basic. It also kicks RANDOMIZE into touch, not only for the LOAD program but also for any program that the LOAD program loads and runs. If the computer manages to find a LOAD program, then your RND will always provide the same sequence every time you boot up, even if your program has a RANDOMIZE..

Try it! Put in a little program on your disk like this:.

```
100 RANDOMIZE.
110 FOR T=1 TO 6.
120 PRINT INT(RND*100).
130 NEXT T.
140 RUN "DSK1.LOAD".
```

and watch the six numbers repeat again and again, in defiance of the RANDOMIZE command..

Thus if you use a LOAD menu program to put in XB programs such as games or graphics or what have you, you may as well forget about RANDOMIZE... except....

Bruce has provided an answer- insert:.

```
70 CALL INIT.
80 CALL PEEK(-31880,[,]).
90 CALL LOAD(-31808,[,]).
```

and now try it... see, different numbers.....

( Yes Virginia you can use [ and ] as variable names!).

.  
NOTE that you must omit the CALL INIT if it has already been used for example to load an assembly utility such as The Missing Link else you will destroy the utility! Call Init is also not required if you use the Triton version of XB titled Super Extended Basic on the module label and (c)1987 TYPC..

Bruce tells us this fix can be attributed to Harry Wilhelm and dates back a couple of years..

-----.

From January MICROpendium, the sad news of the death of JOHN BIRDWELL, age 41. John contributed DSKU (Disk Utilities) to the TI community. .

=====.

In January I received an ASGARD catalogue, and listed below are the current offerings. .

This listing is not a recommendation and orders are sent at your risk, but please contact me if you don't receive an order within six weeks, and especially if you don't receive an order and your credit card has been charged!.

The issue number of a TI\*MES review and the page numbers are quoted as appropriate!.

.  
TRIS MODULE. ITEM E01A. \$19.95 REVIEW #30 p46 .  
ROCK RUNNER. Item No: E05. DISK requires EdAs and j/s. \$12.95. Review #31 p47.  
Tournament Solitaire. Item No: E06. Disk. BRAND NEW. 7 solitaire games. \$14.95  
Waterworks. Item No E07. Disk. Req j/s. BRAND NEW. \$12.95.  
Disk of Dinosaurs. 2 disks. It No G10. \$9.95  
Disk of Pyrates. 4 disks. It No G!1. \$9.95.  
Yet Another Paint Program. For 80 column card owners. Disk. It No G15. \$29.95.  
Page Pro 99. Disk. It No:P01. \$24.95. Review #26 p11. .  
Music Pro. Disk. It No.P15. \$17.95. Review #28 p48 .  
QUICK RUN. Disk. It No: U03. \$9.95 Review #24 p61  
Beyond Video Chess. It No U04. \$9.95 Review #28 p45 .  
Pix Pro. It No U06. \$14.95. Review #28 p46 .  
Orphan Survival Handbook. Book. It No T03. \$9.95 Review #17 p12 .  
Asgard Mouse. Plugs into RS232. Req Disk drive. It No H04. \$49.95  
Midi Master. Module+disk, req RS232. Item P01. \$44.95 .  
High Gravity. Disk. It No E20. \$4.95. Review #20 p23  
Balloon Wars. Disk. It No E21. \$3.95 Review #30 p46 .  
Column Attack. Disk. It No E22. \$3.95 .  
Font Writer 2. Disk(specify DISK). It No G16. \$9.95 Review #20 p2 .  
Total Filer. Disk. It No P22. \$6.95 Review #22 p50 .  
This lengthy list is only a selection from a very large range!.  
ASGARD SOFTWARE P O BOX 10306 ROCKVILLE MD USA 20849.  
(Contact MARTIN BLYTHE at DATABASE for these- he recently listed both Rock Runner and Waterworks at just Ten Pounds each)..

=====.

I have also received a catalogue from **NOTUNG SOFTWARE** which lists:.

THE RING COMPANION. 2 disks. text and instances (Rackham) re The Ring. \$8.  
SON OF THE DISK OF DINOSAURS. An animation, text and instances. 2 disks. \$12..  
Ye Cock Ale. Beer bottle labels and recipes. \$7.  
Filmlib. A database command set for TI Base for video collectors. \$7.  
Fonts and Borders. Vol 1. \$7. Vol 2 \$7. Vol 3:\$7..  
1991 Star Trek TNG Calendar. \$10.  
1991 KBGB Girlie Calendar. \$5..

=====.

And another catalogue -must be the season for them- from COMPRODINE, at

All products are on disk, and where appropriate the TI\*MES issue in which a review appeared is indicated in (brackets)..

Backsteine, a new breakthrough fame (32) at \$10. Living Tomb( ) \$15.

Warzone(28)..\$10. .

Artist Printshop(28)..\$25. Giant Art Posters(28)..\$15

Certificate 99 \$7, Cert 99 Companion 1 and 2 at \$7 each.

Joypaint \$7, Joypaint Pal \$7. Extended Business Graphs \$7

Border Maker for Artist Printshop \$10. and also 30 (thirty) SSSD disks of small graphics (3/4" square) which print out with TI Writer EDITOR.

=====

**From New Scientist Christmas issue....**

**Santa** delivers to a very long road. He notices that at one house, when he removes the first digit and then squares the remaining digit he is back to the original number. Here, let's do this one for you:.

125.....25 x 25 = 125.

What is really quite amazing is that when he gets to the 25th house number which satisfies this rule, he finds it is his own house! What number is his house?.

=====.

**Review: The Turing Omnibus (book)**

The Turing Omnibus: 61 Excursions in Computer Science, by A.K. Dewdney. Hardback, 415 pages. W.H. Freeman and Co. ISBN 0 7167 8154 9.

We have previously reviewed the earlier book, The Armchair Universe, and this one follows on from that, but this time the articles are a little heavier and we do not have the simple algorithms to give us fast graphics output.

Rather a set of articles loosely connected to computing, the reading and understanding of which just may help you to resolve computing problems faster or easier.

Something of a recreational math book. Some of the chapter headings include: Error correcting codes; karnaugh maps; the fastest sorts of sorts; the chaitin-kolmogoroff theory ...

This book is intended as a "semi popular compendium of computer science" rather like a touring bus (pun intended) visiting selected landmarks of computer science. You have been warned!

**Review: The Magic Machine (book)**

The Magic Machine: A Handbook of Computer Sorcery, by A. K. Dewdney, paperback, 357 pages. W.H.Freeman. ISBN 0 7167 2144 9

This 1990 book is a little easier to take. It has a few easy to translate algorithms to turn into programs, and in theory 16 out of 28 chapters could give rise to a program of some sort, although some could be hard to sort out!

Once more, lots of general reading to improve your approach to problems! Including even a mention of transactional analysis; other headings include: Balls in boxes; Trains of thought (some classic recreational math puzzles); Palmiters protozo; Catching biomorphs.

This book is a collection of recreational material that was previously published in Scientific American under the heading "computer recreations."

**Review: Backsteine** From Comprodine, By Quinton Tormanen. Game.

Well that is the worst thing about this offering out of the way anyway, with a high score of 0 out of 10 for the awful name (unless you are German, or speak the language, when you will know the title is Bricks).

This program was written in an English-speaking country (sort of) for sale in same. No excuse.

It is a Breakthrough type program where you move a bat at screen bottom and a bouncing ball batters away at a wall of bricks, and if the ball hits the floor it is lost.

Not original. However the implementation is absolutely superb, well up to modern arcade standards, with the ball and some bricks throwing shadows on a patterned background.

CONTINUED.....

Bricks take one two or three hits to knock out, and may drop one of six different bonus prizes, which make your bat bigger or affect the ball in some way. There are also some bricks which cannot be destroyed. The game comes with an assortment of 50 screens (made up of different patterns of bricks) and there is an editor to make your own.

Game play is with joystick one for one player. Editing can be with joystick one, or with more control (e.g. slower) with joystick 2 or even ESDX keys. The graphics and implementation of this game are exceptional. You may not be playing it non-stop for the next year, but it is an excellent game to while away an hour or so every now and then, and contemplate just what our ancient machine is capable of ... and was capable of in 1980. Why did TI not release modules of this quality! Highly recommended.

**Review: Extended Business Graphs** From Comprodine. Disk.

This program originated with Great Lakes Software. It is quite old, and rather shows its age. Basically you key in your data, and have a few graphing options. Graphs can be printed out in a single format, and cannot be saved to disk! in any format!

A little restricted perhaps, but there are not many graphing programs available, and this new price is not bad.

**Review: Picture it** From Comprodine. Disk. USD \$10.

A graphics utility program with several options, none of them very fast, but quite useful for all that.

Dating back to 1987, this program is by Rodger Merritt, who runs Comprodine. The program disk has several utilities...

**BANNERS** to print Instances up to 12 lines high and/or text in a specially supplied 8 inch high font, sideways, much enlarged. You get to choose which ASCII character represent a "pixel" so there is plenty of room to vary contrast.

Banners seen at Romiley and Chester AGMs en England were prepared with this. Instances can be text prepared with any TI-Artist font of course!

**INSTANCES** can transfer a TI-Artist instance into an XB program, either as redefined characters or as defined sprites. The output file is a MERGE format D/V 163 file.

**TI-WRITER** and you can save Instances to disk in a format which prints out via the TI-Writer Formatter (uses a lot of Transliterations). If you need the program, you need it!

=====

\*\*\* more.....

I have taken out a subscription to a pretty expensive magazine called Recreational and Educational Computing which at 30p per page is not too cheap. Issues are bi-monthly

Here is a puzzle from one issue:

In the land of make believe there is an eccentric jailer who takes the following action on New Year's Day:

At minute 0 all 100 doors are locked shut.

At minute 1 he goes to every cell and opens the door.

At minute 2 he goes to door 2 and, if it is open, closes it. If it is shut he opens it. Then he goes to door 4 and does the same thing and so on to every second cell.

At minute 3 he begins at cell 3, and reverses the state of its door, carrying on with every third cell (reversing 3, 6, 9, 12, etc), and so on and so on and so on.

At minute 100 he just reverses cell 100.

Now then ... how many prisoners can walk free, i.e. which cell doors are open?

\*\*\*

With reference to the puzzle set on p53 of issue #30 regarding the *three hands of a clock*:

For ease of reference, refer to the position of all three hands in terms of the clock face reading in hours that is if the second hand is pointing to the hour number 3, we refer to the position of that hand as 3.

The easiest way to approach this problem is to consider firstly just two hands the hour hand and the minute hand. Instinctively we can visualize these two hands for an angle of 120 degrees (or 4 hours!) about 24 times in 12 hours and then repeat.

When the hour hand moves from 12 to 1, a passage of one hour, the minute hand moves from 12 to 12, moving through 12 hours (hour numbers!). Thus the difference in movement between the two hands is  $12-1=11n$ .

Our problem requires that the difference between the hour and minute hands is 4 hours (on the clock face) so we must solve for  $11n=4$ . Because a clock face is circular (you noticed!) there is more than one value of  $n$  we can use.

We can now write a program for values of  $n$  from one upwards until 12 hours have elapsed and the numbers repeat. It is necessary to continually reduce the number to keep it within the 12 hour period.

Then, having our 22 values when the hour and minute hands are exactly 4 hours apart we can examine the position of the second hand to see where it lies at these very limited number of times. If it is precisely four hours from each of the hour and minute hand, our problem is solved!

```
100 FOR N=4 TO 96 STEP 4
110 IF N/12=INT(N/12) THEN 180 ! hands coincide
120 HOURS=N/11
130 MINS=12*HOURS
140 IF MINS>12 THEN MINS=MINS-12 :: GOTO 140
150 PRINT HOURS ! note 0.09' is 1/11th
160 PRINT MINS
170 PRINT "====="
180 NEXT N
```

and then

```
1 REM THREE HANDED CLOCK
```

```
2 REM DATA=TIME ELAPSED FOR HOUR AND MINUTE HAND
```

---->>>

```

    TO FORM 120 DEG ANGLE EXPRESSED IN HOURS
    ELAPSED ON HOUR HAND
3 REM S SHAW JAN 1991
4 REM TI-99/4A Extended Basic
5 REM
100 DATA 0.36363636
110 DATA 0.72727272
120 DATA 1.45454545
130 DATA 1.81818181
140 DATA 2.54545454
150 DATA 2.90909090
160 DATA 3.63636363
170 DATA 4.0
180 DATA 4.72727272
190 DATA 5.09090909
200 DATA 5.81818181
210 DATA 6.18181818
220 DATA 6.90909090
230 DATA 7.27272727
240 DATA 8.0
250 DATA 8.36363636
260 DATA 9.09090909
270 DATA 9.45454545
280 DATA 10.18181818
290 DATA 10.54545454
300 DATA 11.27272727
310 DATA 11.63636363
320 DATA 99,99
330 READ HOURS :: PRINT "====="
340 IF HOURS=99 THEN STOP
350 MINS=12*HOURS
360 IF MINS>12 THEN MINS=MINS-12 :: GOTO 360
370 PRINT SEG$(STR$(HOURS),1,6)"-"&SEG$(STR$(MINS),1,6)
380 SECS=60*HOURS
390 IF SECS>1200 THEN SECS=SECS-1200 :: GOTO 390
400 IF SECS>60 THEN SECS=SECS-60 :: GOTO 400
410 PRINT SECS/5 ! CLOCK FACE READOUT IN HRS
420 COUNT=COUNT+1
430 IF INT(COUNT/8)=COUNT/8 THEN 440 ELSE 450
440 FOR PAUSE=1 TO 4000 :: NEXT PAUSE
450 GOTO 330
460 END

```

This is a pretty fast way of sorting this problem out. The ticking second hand? Well expressed in terms of the hour numbers, a second is 0.2 hours, so if one of the 22 solutions comes within 0.2 of satisfying our needs, maybe ticking does have some significance ... ?

Much harder query: If the perfect solution is not possible, can we find the time that the hands are closest to our requirement? For this purpose, differences are added regardless of their sign:

```

    We require: 120 120 120
    We get, say: 119 118 123
    Error 1 + 2 + 3 = error 6.

```

I do not have an answer to this one and look forward to hearing from you.  
My thanks to Walter Allum for correspondence received on this one!

=====



## Better Basic programs

A simple puzzle: I somehow manage to find a £1 (100 pence) and wish to purchase items which are priced at 14p, 17p, 22p, and 39p. In what combinations may I purchase these items so that I exactly spend my pound?

We can quickly get our calculators out, work out the maximum number of each I can buy for a pound, and come up with code like this:

```
100 FOR A=0 TO 7
110 FOR B=0 TO 5
120 FOR C=0 TO 4
130 FOR D=0 TO 2
140 IF 14*A+17*B+22*C+39*D=100 THEN PRINT A;B;C;D
150 NEXT D
160 NEXT C
170 NEXT B
180 NEXT A
```

Can you follow the reasoning behind this code? This little program takes about 25.4 seconds to find the four possible answers. We know our console is not the fastest in the world, so can we speed up this work?

The first improvement is interesting curious too. This simple amendment:

```
100 FOR D=0 TO 2
110 FOR C=0 TO 4
120 FOR B=0 TO 5
130 FOR A=0 TO 7
140 IF 14*A+17*B+22*C+39*D=100 THEN PRINT A;B;C;D
150 NEXT A
160 NEXT B
170 NEXT C
180 NEXT D
```

produces nearly a 10% speed-up, and runs in about 23.2 seconds. Most odd.

We can however speed things up a great deal more by applying a little thought to the problem. If we buy a 39p item we have only 61p left which at most allows us to buy 3 items at 22p. In this case the C loop does not need to test for a value of 4. If we buy two 39p items we have only 22p left and can only buy one 22p item.

The following code reduces the number of steps in the loop by taking account of this sort of limitation. The first line is slightly wasteful but will make the flow a little clearer:

```
100 FOR D=0 TO 100/39 ! think about it
110 FOR C=0 TO (100-39*D)/22
120 FOR B=0 TO (100-39*D-22*C)/17
130 A=INT((100-39*D-22*C-17*B)/14)
140 IF 14*A+17*B+22*C+39*D=100 THEN PRINT A;B;C;D
150 NEXT B
160 NEXT C
170 NEXT D
```

This code, would you believe, gives us the same four answers in just 2.82 seconds or thereabouts. However we can still speed things up a little. Look at lines 130 and 140. Provided A does have an integer value our problem has a solution, and we can simplify this part like this:

```
130 A=(100-39*D-22*C-17*B)/14
140 IF A=INT(A) THEN PRINT A;B;C;D
```

This improves the speed to around 2.24 seconds about the time we saved with our first improvement.

continued.....

Just a little thought before coding a problem can save much un-needed work by the computer and speed things up to a degree which is greater than the difference between a 4 MHz computer and a 33 MHz computer all other things being equal, which they never are!  
Thanks to Recreational and Educational Computing Newsletter, Volume 1 No. 4.

=====  
=====

Tenth anniversary

Easter 1981 I keyed in my first program on an NTSC TI-99/4 borrowed from Texas Instruments (those were the days) and subsequently received an invoice from TI dated June 1981 for a fully expanded TI-99/4 sidecar system the UK PAL version was then available, but with sound from a little loudspeaker in the console, as TI had not then worked out how to re-jig the modulator, made for European PAL, to work properly on UK PAL. They never did get it entirely right! despite many changes of modulator design.

I became a founder member of the Texas Instruments TI-99/4A User Group UK, and by January 1982 had a program published in Computer and Video Games magazine (still going, but no listings these days). I was also a founder subscriber to 99'er Magazine, and was soon making my first purchase of modules from the USA. They were so much cheaper there than TI was selling them for here!

As the year progressed I wished to share the software I was buying, and did the proper thing by establishing licensing arrangements and sold programs as Stainless Software, and so we went on ... upgrading to a PEB and 99/4A in due course, continuing to write for anyone who wanted my material.

In due course, the collected writings were assembled into a book I was asked to write it by a commercial publisher, who made the first payment before a word was written. Written entirely with TI-Writer, it brought in a little money, all of which was spent on new programs and books!

I love games, especially strategic games. I enjoy programming. And the TI has so many superb programs, so many languages to explore (still discovering new things about Basic!). I have a stack of computer books full of programming ideas, and hundreds of programs to write, if I can ever find the time (anyone need programming ideas, mainly strategy or puzzle games?).

My TI has at this moment clocked up 11,044 hours, and has outlived the TV I bought to use with it! I am quite happy to go on using it for another ten years, even longer maybe, only an irreparable breakdown is going to separate us!

=====

## REVIEWS:

**Review: The Ring Companion.** Notung Software.

This two-disk set is a companion to the music disks by Ken Gilliland, "Richard Wagner's Greatest Hits" and "Richard Wagners Der Ring Des Nibelungen". We managed no less than three version of The Ring on TV over Christmas/New Year, so it may be a little more familiar to you than last year!

continued.....

This disk set contains:

A huge picture of The Ring, with a dwarf and Rheinmaiden, composed of eight TI-Artist pictures formatted for printing with TI-Artist Plus! but any other \_P printer will do. Plus a picture of Richie Wagner. Plus 12 pictures and Instances of Ring cartoons by Arthur Rackham. Plus some text about The Ring (roots, characters, Wagner). Plus a program to demonstrate 30 "leitmotifs." Quote a package for lovers of Wagner.

**Review: Son of The Disk of Dinosaurs.**

Notung Software, This also consists of 2 disks. A follow up to the Disk of Dinosaurs published by Asgard, this disk set contains a further landscape to place dinosaurs into, and 25 TI-Artist Instances of individual dinosaurs, which Notung Software used to very good effect in a very unusual Christmas card!

There is a further 62-frame cartoon adventure with Thug, but this time using Comic Show 4.0, so you cannot slow down the rather rapid movement! Plus text on the dinosaurs, and a quiz which allows you to build a dinosaur by answering correctly 10 questions. The source for a TI-Artist Instance of a Pachycephalosaurus!

=====

**Review: Jiffy Flyer** From Comprodine. Disk. USD \$10 + USD \$4.00 postage.

Rather an old program but I did not seem to have reviewed it yet. It is a remarkably simple program! It enables you to quickly prepare a single sheet poster, using a border (several are part of the program); a large type heading, a small type announcement over several lines (several fonts are part of the program), and either an internal border or four corners made up of the same graphic, which you load from disk the small pictures are to be in CSGD format, several are supplied, and the Disk Library has many more.

And that really is that!

The format is pretty fixed, which makes for a fairly straightforward and easy to use program the more powerful a program, the less easy it is to use (The Printer's Apprentice is about 10 times harder to use than this program!). A very similar utility can be found in Artist Printshop from the very same publisher, with extra facilities, but priced at USD \$25.

Artist Printshop is the more versatile but if your budget is tight, this little utility, at less than half the price, may well suit.

**Review: Certificate 99** From Comprodine. Disk. USD \$7.00 + USD \$4.00 postage.

This is, again, a rather old program, which allows you to fill an 8.5" x 11" or A4 sheet of paper with a computer-produced certificate, with a choice of borders, a choice of graphics, a choice of fonts, and a choice of signatures (perhaps Mrs. Thatcher is no longer appropriate?). There are support disks available from both Comprodine and Notung with extra fonts, etc. If you need to produce certificates, this is the most directly applicable TI program, and pretty inexpensive too!

=====

=====.

## CUFFLEY...

The meeting at CUFFLEY went well in January, although not as well attended by SOUTHERN users as could have been hoped. It was mentioned that Cuffley being actually North of London was to be considered a Northern show... hmmm. Those who did attend had a good chat amongst themselves and there were some real bargains to be picked up for those who wanted them..

Cuffley was attended by about 38 TI users - (The Sydney group in Australia, now starting its tenth year, also had 38 attending its AGM in December 90...!). There was good representation from the DORTIG Dorset group, and Martin Blythe from Database was there..

## SUPPLIES...

In the last few months I have had requests for software items which Martin can supply from stock, which puzzles me a little. I am not in any form of TI business, and have no intentions of semi-commercial supply. Attempts to help members with group purchases have met with far too little response to make it worthwhile.

And Edward Shaw (see cover) has a good stock of modules available second hand. Edward or Martin can (between them) obtain virtually anything you may want!.

In extremis- and it really shouldn't be too often! - if you cannot find what you want, drop me a line, but note I do not have stocks, and you must expect severe delays! I am not competing with Edward and Martin- merely supplementing them, as I do have US contacts who can assist with odd items.

I have also been taken to task for not pointing out in very blunt language that you really can still obtain a very very large number of items of software on module or disk, and almost every module ever made is still available from somewhere. Before you are tempted to look at another brand of computer, consider what software you need. There is a very good chance you can obtain an adequate working system for your TI, probably at a much lower cost!

MICROpendium has been mentioned in almost every issue. They review new items and tell you where to get them. There is also regular advertising of older material especially TI modules at low low prices..

The cost of modules in the States can be very low, and this does make the overhead costs of such long distance purchases seem very high. For example, if you buy a US\$ International Money Order from the Royal Bank of Scotland there is a commission charge (as at Feb 1991) of Three Pounds. NOT a lot for a \$200 order but rather steep for a US\$5 order! It is the same piece of paper no matter what value is written on it, with all the same work to do! There is a similar problem with postage- a single disk MAY cost only \$5 by airmail, but you can usually expect to pay \$8/\$10 postage on most orders. A large package by SEAMAIL may be \$30 or more... so don't forget MARTIN, who does all the work for you as well as covering overhead costs - which include storage and buying his food!.

If you wish to order directly, it is not difficult, you buy an IMO or you quote your credit card number (TexComp take credit cards, most other suppliers are too small). In due course the postman delivers your package, and MAY ask you to pay 15%+costs on the total invoice value before handing the package over (most packages for low values are allowed through without charge). If you have difficulties with any supplier, please write and tell me and we can spread the word and maybe even offer some assistance. Note you may have to wait two months for a package!.

.=====.

Some time ago I printed a membership application form for the Sydney group, which was utterly ignored by everyone. They now have a second UK member to join me! They publish a 24 page (size A4 but still small print!) magazine 11 times a year and overseas subs are A\$45 by slow seamail or A\$60 by airmail. Their articles include machine code, hardware, and Forth as well as some you are already familiar with.  
and another plug for.

MICROPENDIUM,

Seamail US\$30, Airmail US\$42, and MICROpendium take credit cards! Quote card expiry date and say MASTERCARD for Access and VISA for members of that group as well as the card number of course. Monthly, 40/48 pages, now in 8th year..

MICROpendium gave TI\*MES a good review in their January 1991 issue!.

Also in their January issue a note that COMPRODINE was not only still bringing out new goodies but had on board programs from Great Lakes Software such as JoyPaint and Certificate 99 at very low prices..

=====.

Some welcome comments at last on the graphics routines- after a long silence it appears there are enough of you out there to keep some graphics programs coming! Thanks for writing everyone, it is appreciated..

=====.

**TI WRITER et al....**

Our friends in the East Anglia group have passed on some good TI Writer tips, culled from other newsletters, notably....

... CTRL 8 places a carriage return and adds a line. It is faster to type ctrl 8 then fctn 3 than it is to type say (ctrl u, fctn r, ctrl u) should you need to insert carriage returns in already written text..

... when you enter LF Funnlweb will place your cursor after DSK- that is, on the disk number. If you type LF SPACE SPACE enter, the cursor will appear after the dot eg after DSK1. or whatever. Backspace has a similar effect to space-wherever the cursor is when you press enter, it will be in a similar position on the device/filename..

=====.

A WARM WELCOME to our new members STATESIDE who are very welcome to drop me a line should they wish to share there knowledge and experience and goodies.... and if there is ANY U.K. product you want but cannot get let me know! (eg Red Dwarf paperback!). Every member is of course welcome to write and say what they like or dislike about Rambles, to ask questions and so on. If a personal reply is needed an SAE (or dollar bill) is always useful..

=====.

TI\*MES #31 update... re page 26....

NO response from Paul Scheidemantle, despite sending a couple of IRCs. Could be a postal error, or I got his address wrong, or he don't want to know...???.

MICROpendium Jan 91 issue indicates at least some CorComp owners are not finding it easy to get IDT to repair failed CorComp cards..

-----.

MORE.....

```

1 REM WHAT IS NEXT NUMBER...
2 ! 3 5 7 9 ?
3 ! 9 16 25 36 ?
4 ! 8 27 64 125 ?
5 ! 4 7 11 16 22 29 ?
6 !
7 ! write a program to find      the number ? in each      case.
8 !
9 ! Ecker.86. Shaw.91.
100 CALL CLEAR
110 DIM A(30,30)
120 PRINT "INPUT A SERIES OF NUMBERS"
130 PRINT "INPUT -99 TO TERMINATE- THISWILL NOT FORM PART OF SERIES"
140 J=1
150 INPUT A(1,J)
160 IF A(1,J)=-99 THEN A(1,J)=0 :: J=J-1 :: GOTO 190
170 J=J+1
180 GOTO 150
190 N=J
200 FOR K=2 TO N
210 FOR J=1 TO N+1-K
220 A(K,J)=A(K-1,J+1)-A(K-1,J)
230 NEXT J
240 NEXT K
250 A(N,2)=A(N,1)
260 FOR L=N-1 TO 1 STEP -1
270 A(L,N-L+2)=A(L,N-L+1)+A(L+1,N-L+1)
280 NEXT L
290 PRINT "NEXT NUMBER IS ";A(1,N+1)
300 GOTO 120
301 ! not all series will      be solved with this      program.

```

=====

## PRINTER GRAPHICS..

We have had a number of programs of screen graphics, but details of printer graphics seem a little scarce, and as the manuals tend to be a little unhelpful, this item is directed in that direction. .

The EPSON standard is now covered by almost all "mainstream" printers, and to some extent almost all dot matrix printers sold today are compatible with the commands we shall be discussing first, although the second method may not be so widespread..

Firstly let us look at the way the Epson graphics mode works. On our computer characters are typically defined in an 8 x 8 block, in a manner fairly well known to most TI owners- consult your TI manual if in doubt! The TI defines its characters in two horizontal blocks, so that a straight line character could be defined as "FF00000000000000" while a solid 8x8 block would be "FFFFFFFFFFFF FFFF" or a straight vertical line "8080808080808080". The printer has a print head with 9 pins arranged vertically, and characters are defined in a vertical manner to allow faster printing..

On the Epson, characters are nominally 8 dots high, with an option of shifting them down one row to use the ninth (bottom) printing pin. We shall here discuss only the use of the "normal" eight pins, and refer to the eighth pin from the top as the "bottom" pin- eight pin bit image graphics by definition do not use the ninth pin! .

The bottom pin is allocated a value of 1, the next one up a value of 2, then 4 then 8 then 16 then 32 then 64 and finally the top pin a value of 128. We add up the values of all the pins we wish to strike in a single column and send this value to the printer- so a straight vertical line would require us to send a value of 255, eg CHR\$(255)..

Sending several columns to print a character needs a little rethinking from the TI approach however, as the Epson graphic standard uses a half column. Imagine that the usual TI columns are numbered 1,3,5,7 and so on. These print exactly next to each other. The Epson standard allows us to print dots half-under each other, like this:.

|                 |                      |
|-----------------|----------------------|
| TI:             | EPSON:.              |
| XXX XXX XXX XXX | XXX XXX XXX XXX XXX. |
| XXX XXX XXX XXX | XXX XXX XXX XXX XXX. |
| XXX XXX XXX     | XXX XXX XXX XXX .    |
| XXX XXX XXX     | XXX XXX XXX XXX.     |
| XXX XXX         | XXX XXX XXX XXX .    |
| XXX XXX         | XXX XXX XXX XXX .    |

Thus the printer allows characters to be a little less square than the TI forces us to be. Note however that the intermediate half column pins cannot be used as well as an adjacent normal column pin when using quadruple density which is ESC \* 3 or double speed double density which is ESC \* 2 !(Thus if we use pin value 4 in column 1 we cannot use pin value 4 in column 2, but we can use pin value 2 or 8 in column 2)..

The "bog standard" Epson 8 bit graphic uses 480 dots along each line and uses the command ESC K... as you will see in the program which follows. Note that an exact equivalent to ESK K is ESC \* 0- so instead of:.

```
PRINT #1:CHR$(27);"K";.... you could use:
PRINT #1:CHR$(27);"*";CHR$(0);...
```

Generally you will be printing graphics using more than 80 columns of dots so to avoid an auto line feed we open the printer up by adding .CR to its name, and make our program sane the line feeds at the end of each graphic line..

Our program is using a graphic with a defined shape in a 6x6 grid, so we need to tell the computer to move the line up 6 dots at a time (Line 110)..

In line 140 we tell the printer to use the 8 pin bit image mode with 480 dots across, and that after each ESC K sequence we are going to send (218)+(1\*256)=474 defined columns -after which we will tell it to move up a line (line 260). Note than by not using the full width, and by sending say 200 columns of dots, the printer then switched to text mode for the remainder of the line- we can send mixed text and graphics, or even use graphics mode for micro adjustment of character positions..

The program is based on a Truchet tile generation.

```
100 OPEN #1:"PIO.CR"
110 PRINT #1:CHR$(27);"A";CHR$(6);.
120 REM lines are 6 dots high.
130 RANDOMIZE .
140 A$=CHR$(27)"K"&CHR$(218)&CHR$(1) .
150 REM sets up for 474 dot columns to follow..
160 [$=CHR$(8)&CHR$(16)&CHR$(32)&CHR$(65)&CHR$(2)&CHR$(4).
170 ]$=CHR$(8)&CHR$(4)&CHR$(2)&CHR$(65)&CHR$(32)&CHR$(16).
180 PRINT #1.
190 PRINT #1:A$;.
200 FOR WIDTH=1 TO 79. ! CONTINUED
```

```

201 REM 6 x 79 = 474.
210 IF RND<.5 THEN 240.
220 PRINT #1:[$;..
230 GOTO 250.
240 PRINT #1:]$;
250 NEXT WIDTH.
260 PRINT #1:CHR$(10);CHR$(13);.
270 GOTO 190.
280 END.

```

.  
Please note that this graphics program is written in pure TI BASIC and only requires a printer and appropriate TI->Printer interface to operate. Instead of ESC K try using instead ESC \* 0 to ESC \* 6 or any other ESC \* modes your printer may support. The above program does not use adjacent horizontal dots and so all modes may be used..

.  
The Epson range of printers can be found with "definable fonts" which allows you to redefine the character definitions held in the printer. This is a little more arduous. The TI Basic program below prints about the same pattern as the above program, but instead of using different ESC \* modes you may use character modes such as condensed, expanded, Pica, Elite, and so on. Your printer must support the commands ESC & as well as ESC %. On the FX80 printer it is necessary to switch OFF the on board print buffer, as this is used to store the new definitions. See your printer manual under "download characters"..

```

100 OPEN #1:"PIO.CR".
110 PRINT #1:CHR$(27);"A";CHR$(6);.
120 PRINT #1:CHR$(27);"p";CHR$(1); .
121 REM proportional mode required to get correct horizontal spacing for 6x6
graphic..
130 RANDOMIZE.
140 A$=CHR$(27)"&"&CHR$(0) .
150 B$=CHR$(0)&CHR$(0)&CHR$(0)&CHR$(0)&CHR$(0).
160 [$=A$&"["&CHR$(6)&CHR$(8)&CHR$(16)&CHR$(32)&CHR$(65)&CHR$(2)&CHR$(4)
&CHR$(8)&B$.
170 ]$=A$&"]]"&CHR$(6)&CHR$(8)&CHR$(4)&CHR$(2)&CHR$(65)&CHR$(32)&CHR$(16)
&CHR$(8)&B$.
180 PRINT #1:[$&]$.
181 REM redefines characters [ and ].
190 PRINT #1:CHR$(27);"%";CHR$(1);CHR$(0);.
191 REM tells printer to use new definition instead of usual character.
200 FOR WIDTH=1 TO 131.
210 IF RND<.5 THEN 240.
220 PRINT #1:"[";.
230 GOTO 250.
240 PRINT #1:"]";.
250 NEXT WIDTH.
260 PRINT #1:CHR$(10);CHR$(13);.
270 GOTO 200.
280 END.

```

.  
Have fun. Stephen Shaw 1990.  
.=====

CONTINUED...



.  
 In Dr Pickover's excellent book COMPUTERS PATTERN CHAOS AND BEAUTY he gave a short routine for producing a graphic result from pascals triangle, which did not work for me so I left it. I found the original article in his Journal of Chaos and Graphics for August 1988 and this encouraged me to have another look. The program is based on plotting pixels when an array element takes a zero value, but as the program started off initialising the array to zero values, and as MOD(zero,N) is always zero, I kept getting a solid black triangle... then I thought of initialising to a value of 1, which is appropriate to the triangle... what is pascals triangle? It is formed of numbers like this:.

```

      1
     1 1
    1 2 1
   1 3 3 1
  1 4 6 4 1
  
```

OR

```

  1.
 11.
121.
1331.
14641.
  
```

.  
 The right hand form is merely a reformat which makes computer graphics easier! Note that each number is the sum of each of the two numbers above it, assuming a value of 1 along each side..

.  
 The triangle has a number of fascinating mathematical applications, the one I first met with in school was the result of  $(A+B)^N$ . If N is two, that is we multiply  $(a+b)$  by  $(a+b)$  then we get  $a^2+2ab+b^2$ , a 1-2-1 pattern.. Similarly,  $(a+b)^3$  is  $a^3+3a^2b+3ab^2+b^3$ , a 1-3-3-1 pattern, and so on down the triangle..

There are many other applications, but here is a graphic one for the computer. Lets examine each number in turn, and if it is exactly divisible by a number of our choosing, turn a pixel on in that position or leave it off.. Thus looking to the right hand format, and using a divisor of 2, we would have a pattern, using 0 for pixel off, X for pixel on, of:.

```

  0.
 00
0X0
0000
0XXX0 and so on.
  
```

Let's put that into a program!.

```

1 REM PASCALS TRIANGLE BY MODULAR MEANS.
2 ! FROM CLIFFORD PICKOVER. Modified S Shaw Dec 90.
3 ! for TI99/4A + ExBas + The Missing Link..
4 ! try different values if DIV..
90 DIV=4.
100 CALL LINK("PRINT",90,15,"MOD "&STR$(DIV)).
110 DIM P(190),C(190).
120 FOR L=0 TO 190 :: P(L)=1 :: NEXT L.
130 FOR N=2 TO 190 :: FOR R=2 TO N.
140 CALL MOD(P(R)+P(R-1),DIV,C(R)).
150 IF C(R)=0 THEN CALL LINK("PIXEL",R,N).
160 NEXT R .
170 FOR L=1 TO 190 :: P(L)=C(L) :: NEXT L.
180 NEXT N
190 GOTO 190.
200 SUB MOD(A,B,OP) :: OP=INT(A-B*INT(A/B)).
210 SUBEND.
  
```

.  
 MORE.....

or if you wish to look at a much larger triangle we can output straight to our  
epson compatible printer like this:.

```
.
100 DIM P(470),C(470).
110 G$=CHR$(27)&"K"&CHR$(470-256)&CHR$(1).
120 FOR L=0 TO 470 :: P(L)=1 :: NEXT L.
130 OPEN #1:"PIO.CR".
140 PRINT #1:CHR$(27);"A";CHR$(1);CHR$(10);CHR$(13).
150 DIV=15.
160 FOR N=2 TO 470 :: PRINT #1:G$&CHR$(0)&CHR$(0).
170 FOR R=2 TO N :: CALL MOD(P(R)+P(R-1),DIV,C(R)).
180 IF C(R)=0 THEN PRINT #1:CHR$(2); ELSE PRINT #1:CHR$(0);.
190 NEXT R.
200 FOR L=2 TO 470-N :: PRINT #1:CHR$(0); :: NEXT L.
210 FOR L=1 TO 470 :: P(L)=C(L) :: NEXT L.
220 PRINT #1:CHR$(10);CHR$(13).
230 NEXT N.
240 CLOSE #1 :: STOP.
250 SUB MOD(A,B,OP).
260 OP=INT(A-B*INT(A/B)).
270 SUBEND.
.
=====.
```

more content follow.....

INFOCOM was a company that made text-only adventures of mind boggling complexity, and many were released for the TI99/4A. Later larger adventures required a double sided disk, and were not officially released for the TI but can be found. Some of these were so large they required also extra RAM at >6000. Here is a full list of Infocom adventures for the TI..

```

CODE: YEAR: AUTHOR: LEVEL: REQ: NAME:.
A1.....83.....15.....3.....S.....INFIDEL.
A3.....84.....15+12...2.....S.....CUTTHROATS.
A4.....86.....7.....2.....D.....HOLLYWOOD HIJINX (Treasure hunt).
C1.....86.....17.....2.....R.....LEATHER GODDESSES OF PHOBOSS.
H1.....87.....5.....-.....D.....LURKING HORROR (Login 99. Password TI. Clue)
K1.....84.....16+10...1.....R.....SEASTALKER.
M1.....82.....14.....4.....S.....DEADLINE .
M2.....83.....16.....2.....S.....WITNESS.
M3.....84.....5.....3.....S.....SUSPECT.
M4.....86.....11.....2.....D.....BALLYHOO (at the circus!).
M5.....86.....16+10...1.....R.....MOONMIST.
R1.....87.....1.....-.....D.....PLUNDERED HEARTS.
S1.....82.....5.....4.....S.....STARCROSS.
S2.....83.....15.....4.....S.....SUSPENDED.
S3.....83.....17.....2.....S.....PLANETFALL.
S4.....84.....17+4....2.....S.....HITCHHIKERS GUIDE TO THE GALAXY.
S6.....87.....17.....-.....D.....STATIONFALL.
Z0.....85.....3.....1.....D.....WISHBRINGER.
Z1.....80.....14+5....2.....S.....ZORK 1.
Z2.....81.....14+5....3.....S.....ZORK 2.
Z3.....82.....14+5....3.....S.....ZORK 3.
Z4.....83.....14+5....2.....S.....ENCHANTER.
Z5.....84.....17.....3.....S.....SORCEROR.
Z6.....85.....5.....4.....D.....SPELLBREAKER.
--.....--.....--.....--.....S.....SAMPLER

```

Authors: 1. Amy Briggs 3. Brian Moriarty. 4. Douglas Adams 5. Dave Lebling.  
7. Dave Anderson. 10. Jim Lawrence. 11. Jeff O'Neill. .  
12. Jerry Wolper. 14. Marc Blank. 15. Michael S Berlyn..  
16. Stu Galley. 17. Steve Meretzky..

Level: Some games were released without level indication. 1=Easiest 4=Expert..

Req: S-Standard TI disk system. D-Double sided disk drive. .

R-Double sided disk system plus ram at >6000 ..

Note that Code A2 was not used, and some codes are not available on TI format..

In 1986 Infocom was bought by Activision, now calling itself Mediagenic. They have now formally written off their purchase for nine million dollars, and while they still own the name, there will be no more Infocom adventures as they are now known. (Information taken from Oct/Nov issue of New Atari User)..

=====.

more.....

## DESIGNS FOR FUN.

Originally written for TI Basic by Peter Brookes and published in TIDINGS in June 1982. Rewritten for Extended Basic plus The Missing Link by Stephen Shaw 1990..

Main Menu: .

0. Set Defaults... resets all parameters to the original default values..
1. Check Settings.. displays current parameter values.
2. Select Pattern Area.. each block of pattern may be set to an area of 1, 4, 9, 16 or 25 characters- from 8x8 pixels to 40x40 pixels.
3. Select Colours.. Only operational if The Missing Link is loaded and set up for 16 colour use. Sets screen, foreground and background colours..
4. Select Bias.. Allows selection from 16 predefined biases-for more use option 5. See description below.. (Default 01234789CEF).
5. Create Bias.. Allows input of a bias string from 2 to 112 characters, composed of the numbers 0 to 9 and the letters A to F..

The program is designed so that the selection of combinations at random is made from the bias string. If the string contains the entire range of combinations (0 to F) the program will randomly select any of the 16 possible combinations of 4 pixels. If however the bias string contained only 1248 the program would only be able to select from those patterns, thus biasing the result..

6. Reset Randomiser.. It is the nature of the beast that it may become cyclic-probably with a large cycle! Using this option will break the cycle and shift the pattern generation to another range..

7. Select Mode.. either continuous or pattern will stay on screen until you press a key..

8. Continue.. get the patterns going..

.  
During pattern generation:.

Press and hold key S to return to main menu..

User prompted pattern change: any key except S & A for next pattern..

To print: Hold CTRL and FCTN together..

To save in TI Artist format: Once the pattern is on screen and before the definitions at screen bottom start to change, press and hold key A. Indicate drive number, then file name (up to 8 characters, omit \_P). Easier to use if user prompt mode is specified..

.  
Patterns saved in TI Artist format can then be cut out into Instances and used for texture or repeated over the full screen as required before printing. Can be used for book jackets or fly leaves or even wallpaper! Enlarge and use as a crosswork design. etc etc..

.  
As listed the equivalent character definitions are given at screen bottom- if you would like to save your TI Artist files without these then amend the following line:.

2730 FOR @=1 TO 24 :: etc etc.

Or to view a full screen pattern after the new "characters" have been displayed, make the following amendment:

2630 FOR L=1 TO 24.

The characters are numbered from left to right first then from top to bottom, the number pattern at right. indicates how the 5x5 grid is numbered!

|    |    |    |    |
|----|----|----|----|
| 1  | 2  | 3  | 4  |
| 5  | 6  | 7  | 8  |
| 9  | 10 | 11 | 12 |
| 13 | 14 | 15 | 16 |

.

```

100 REM  DESIGNS FOR FUN          (C) PETE BROOKS 1982 V1.0
110 REM  TO TML S SHAW 1990
120 CALL LINK("CLEAR") :: CALL LINK("CHSIZE",8,8) :: OPTION BASE 1 :: DIM
A$(24),B$(40),C$(2) :: D$="0123456789ABCDEF" :: E$="084C2A6E195D3B7F" ::
F$="54433322" :: C$(1)="CONTINUOUS"
200 C$(2)="USER-CUED" :: DEF A(B)=INT(RND*B)+1 :: FOR C=96 TO 127 :: CALL
LINK("CHAR",C,"00013B3323033F7F") :: NEXT C :: GOTO 2270
260 FOR C=1 TO D*8 :: B$(C)=" " :: NEXT C :: E=LEN(G$) :: FOR C=1 TO D*4 :: FOR
F=1 TO D :: H$=SEG$(G$,A(E),1) :: B$(C)=H$&B$(C)&SEG$(E$,POS(D$,H$,1),1) :: CALL
KEY(1,G,H) :: IF G=2 THEN 550
360 IF G=1 THEN GOSUB 2670
370 NEXT F :: B$(D*8-C+1)=B$(C) :: NEXT C :: I=96 :: FOR C=1 TO D*8 STEP 8 ::
FOR F=1 TO 2*D-1 STEP 2 :: H$=" " :: FOR J=C TO C+7 :: H$=H$&SEG$(B$(J),F,2) ::
CALL KEY(1,G,H) :: IF G=2 THEN 550
480 NEXT J :: CALL LINK("CHAR",I,H$) :: GOSUB 2540 :: I=I+1 :: NEXT F :: NEXT
C :: GOSUB 2610
550 RETURN
560 FOR C=1 TO 24 :: A$(C)=" " :: NEXT C :: FOR C=1 TO D :: FOR F=1 TO D :: A$
(C)=A$(C)&CHR$(95-D+C*D+F) :: NEXT F :: NEXT C :: FOR C=1 TO D :: FOR F=1 TO
VAL(SEG$(F$,D,1)) :: A$(C)=A$(C)&A$(C) :: NEXT F :: A$(C)=SEG$(A$(C),1,28) ::
NEXT C
700 FOR C=D+1 TO 24 :: A$(C)=A$(C-D) :: NEXT C :: RETURN
740 FOR C=1 TO 10 :: CALL LINK("PRINT",C*8-7,1,A$(C)) :: NEXT C :: RETURN
780 D=3 :: GOSUB 560 :: K=1 :: G$="01234789CEF" :: L=15 :: M=15 :: N=2
850 CALL SCREEN(L) :: REM
870 CALL LINK("COLOR",N,M) :: REM
890 CALL SOUND(100,1220,0) :: RETURN
910 CALL SOUND(100,1220,0)
920 CALL KEY(0,G,H) :: IF H<1 THEN 920
940 O=POS(D$,CHR$(G),1) :: IF O=0 THEN 920
960 CALL SOUND(100,1220,0) :: RETURN
980 CALL LINK("CLEAR") :: CALL LINK("PRINT",2,20,"CURRENT SETTINGS") :: RESTORE
2480 :: FOR C=1 TO L :: READ I$ :: NEXT C :: CALL LINK("PRINT",12,20,"SCREEN IS
"&I$) :: RESTORE 2480 :: FOR C=1 TO M :: READ I$ :: NEXT C
1090 CALL LINK("PRINT",22,20,"BACKGROUND IS "&I$) :: RESTORE 2480 :: FOR C=1 TO
N :: READ I$ :: NEXT C :: CALL LINK("PRINT",32,20,"FOREGROUND IS "&I$)
1150 CALL LINK("PRINT",42,20,"BIAS IS "&G$) :: CALL LINK("PRINT",82,20,"PATTERN
AREA IS "&STR$(D*D)) :: CALL LINK("PRINT",92,20,"MODE IS "&C$(K))
1170 CALL LINK("PRINT",122,20,"PRESS 'S' TO CONTINUE")
1180 CALL KEY(1,G,H) :: IF G<>2 THEN 1180
1200 CALL SOUND(100,1220,0) :: RETURN
1220 CALL LINK("CLEAR") :: CALL LINK("PRINT",20,20,"PATTERN AREA") :: CALL
LINK("PRINT",30,20,"PRESS: FOR:") :: FOR C=1 TO 5
1250 CALL LINK("PRINT",30+C*9,20,STR$(C)&" "&STR$(C*C)&" "&("&STR$(C)&" X
"&STR$(C)&")) :: NEXT C :: CALL LINK("PRINT",120,20,"YOUR SELECTION ?:") ::
CALL SOUND(100,1220,0)
1290 CALL KEY(0,G,H) :: IF(G<49)+(G>53)THEN 1290
1310 CALL SOUND(100,1220,0) :: D=G-48 :: CALL LINK("PRINT",140,90,D) :: GOSUB
560 :: RETURN
1360 CALL LINK("CLEAR") :: RESTORE 2480 :: CALL LINK("PRINT",12,20,"SELECT
COLOURS") :: CALL LINK("PRINT",22,20,"PRESS: FOR:") :: FOR C=1 TO 16 :: READ I$
1420 CALL LINK("PRINT",C*8+26,23,SEG$(D$,C,1)&" "&I$) :: NEXT C :: CALL
LINK("PRINT",160,40,"SCREEN COLOUR ?:") :: GOSUB 910 :: L=0
1470 CALL LINK("PRINT",160,220,SEG$(D$,L,1)) :: CALL
LINK("PRINT",169,20,"BACKGROUND COLOUR ?:") :: GOSUB 910 :: M=0
1500 CALL LINK("PRINT",169,220,SEG$(D$,M,1)) :: CALL
LINK("PRINT",178,20,"DIFFERENT FOREGROUND COLOUR ?:")
1510 GOSUB 910 :: IF M=0 THEN 1510      ! more....

```

```

1530 N=0 :: CALL LINK("PRINT",178,220,SEG$(D$,N,1)) :: FOR C=1 TO 200 :: NEXT
C :: RETURN
1580 CALL LINK("CLEAR") :: RESTORE 2520 :: CALL LINK("PRINT",10,20,"BIAS
SELECTION") :: CALL LINK("PRINT",20,20,"PRESS: FOR:") :: FOR C=1 TO 16 :: READ
G$
1630 CALL LINK("PRINT",C*8+22,30,SEG$(D$,C,1)&"      "&G$) :: NEXT C :: CALL
LINK("PRINT",170,20,"YOUR SELECTION ?:") :: GOSUB 910 :: CALL
LINK("PRINT",170,220,SEG$(D$,0,1))
1680 FOR C=1 TO 200 :: NEXT C :: RETURN
1710 CALL LINK("CLEAR") :: CALL LINK("PRINT",10,10,"BIAS  CREATION") :: REM
1740 CALL LINK("PRINT",22,20,"USE 0 TO 9 AND A TO F")
1750 REM
1760 CALL LINK("PRINT",32,20,"INPUT BETWEEN 2 & 112 DIGITS LONG") :: CALL
LINK("PRINT",62,20,"PLEASE ENTER YOUR STRING :- ") :: CALL
LINK("INPUT",70,20,G$,112)
1790 CALL LINK("CLEAR") :: CALL LINK("PRINT",100,20,"CHECKING...") :: IF
LEN(G$)>1 THEN 1850
1820 CALL LINK("PRINT",109,20,"YOU'VE MADE A MISTAKE") :: CALL
LINK("PRINT",119,1,"YOU'LL HAVE TO INCREASE THE LENGTH OF YOUR STRING") :: GOTO
1760
1850 FOR C=1 TO LEN(G$) :: H$=SEG$(G$,C,1) :: IF H$<>" " THEN 1900
1880 CALL LINK("PRINT",109,20,"PLEASE DO NOT INCLUDE SPACES") :: GOTO 1750
1900 IF(H$<"0")+(H$>"9")*(H$<"A")+(H$>"F")THEN 1750
1910 NEXT C :: CALL LINK("PRINT",150,50,"STRING CHECKED") :: FOR C=1 TO 200 ::
NEXT C :: RETURN
1960 CALL LINK("CLEAR") :: RANDOMIZE :: CALL SOUND(100,1220,0) :: RETURN
2000 CALL LINK("CLEAR") :: CALL LINK("PRINT",10,20,"SELECT  MODE") :: CALL
LINK("PRINT",20,20,"PRESS: FOR:") :: FOR C=1 TO 2
2030 CALL LINK("PRINT",40+C*10,40,STR$(C)&"      "&C$(C)) :: NEXT C :: CALL
LINK("PRINT",80,30,"YOUR SELECTION ?:")
2060 CALL KEY(0,G,H) :: IF(G<49)+(G>50)THEN 2060
2080 K=G-48 :: CALL LINK("PRINT",90,120,K) :: CALL SOUND(100,1220,0) :: FOR C=1
TO 200 :: NEXT C :: RETURN
2140 CALL LINK("CLEAR") :: GOSUB 850 :: GOSUB 740
2170 GOSUB 260 :: CALL KEY(1,G,H) :: IF G=2 THEN 2260
2200 IF G=1 THEN GOSUB 2670
2210 ON K GOTO 2170,2220
2220 CALL KEY(1,G,H) :: IF H<1 THEN 2220
2240 CALL SOUND(100,1220,0) :: GOTO 2170
2260 RETURN
2270 GOSUB 780 :: GOSUB 560
2290 CALL LINK("CLEAR") :: CALL SCREEN(15) :: RESTORE 2500 :: CALL
LINK("PRINT",10,20,"MAIN  MENU") :: CALL LINK("PRINT",20,20,"PRESS: TO:") :: FOR
C=0 TO 8 :: READ I$
2350 CALL LINK("PRINT",C*9+28,30,STR$(C)&"      "&I$) :: NEXT C :: CALL
LINK("PRINT",168,30,"YOUR SELECTION ?:") :: CALL SOUND(100,1220,0)
2390 CALL KEY(0,G,H) :: IF(G<48)+(G>56)THEN 2390
2410 G=G-47 :: CALL LINK("PRINT",176,140,G-1) :: CALL SOUND(100,1220,0) :: FOR
C=1 TO 200 :: NEXT C :: ON G GOSUB 780,980,1220,1360,1580,1710,1960,2000,2140
2470 GOTO 2290
2480 DATA TRANSPARENT,BLACK,MEDIUM GREEN,LIGHT GREEN,DARK BLUE,LIGHT BLUE,DARK
RED,CYAN,MEDIUM RED
2490 DATA LIGHT RED,DARK YELLOW,LIGHT YELLOW,DARK GREEN,MAGENTA,GREY,WHITE
2500 DATA SET DEFAULTS,CHECK SETTINGS,SELECT PATTERN AREA,SELECT COLOURS,SELECT
BIAS,CREATE BIAS
2510 DATA RESET RANDOMISER,SELECT MODE,CONTINUE      ! more>>>

```

```

2520 DATA F2481F4812F8124F1248,02468ACE13579BDF,FFEECC00,FEC8,01234789CEF,
1248,8CEF137F,11333311333311C8
2530 DATA 842184218421,124812481248,9AC953F,11224488,07E07E07E,DB00DB00,
969696F08181,1032547698BADCFE
2540 REM PUT CHARDEFS ON SCREEN
2550 CALL LINK("CHSIZE",6,7) :: P=I-95! 1 TO 25
2560 Q=8*INT(D*D/2+2)! 1 TO 13 *8
2570 R=193-Q+INT((P-1)/2)*8 :: S=1-((I/2)<>INT(I/2))*114 :: CALL
LINK("PRINT",R,S,STR$(P)&":"&H$) :: CALL LINK("CHSIZE",8,8) :: RETURN
2610 REM DISPLAY NEW PATTERN
2620 T=(192-Q)/8 :: FOR C=1 TO T :: CALL LINK("PRINT",C*8-7,1,A$(C)) :: NEXT
C :: RETURN
2670 REM TO ARTIST FILE
2680 CALL LINK("PRINT",1,196,"DSK") :: CALL LINK("INPUT",1,220,U,1) :: CALL
LINK("PRINT",1,194,"FILE ") :: CALL LINK("INPUT",9,170,J$,8) :: J$="DSK"&STR$
(U)&". "&J$
2730 FOR U=1 TO 3 :: CALL LINK("PRINT",U*8-7,1,A$(U)) :: NEXT U :: CALL
LINK("SAVEP",J$,1) :: RETURN

```

NB- Make sure your program on disk is in IV254 format or loading the program will corrupt The Missing Link.

=====

```

1 ! Bloggs Store is giving away a free playing card with every purchase. The
y have the same number of each of the 52 cards.
2 ! Cards are given at "true" random.
3 ! on average how many purchases do I need to make to get a full set
of cards?
4 ! Phillips 86. Shaw 91.
100 ITEMS=52
110 FOR K=1 TO ITEMS
120 S=S+1/K
130 NEXT K
140 PRINT "BUY ";S*ITEMS
150 STOP

```

=====

=====

=

content continues....

.  
Yet another odd magazine to come my way is THE JOURNAL OF CHAOS AND GRAPHICS,  
edited by Clifford A Pickover. Sample copy available from:  
Clifford A Pickover, .

.  
This program below is from Volume 3, August 1988 issue, and produces a  
different sort of pattern. It is an opportunity for me to show you how to use  
the MOD function on the TI- instead of  $A=MOD(B,C)$  on some other computer, I  
have used a subprogram and the form  $CALL MOD(B,C,A)..$

.  
Also I have used the DEF function to prepare COS and SIN functions COSD and  
SIND which expect to find degree measures instead of the usual radians. It is a  
good way of finding out how efficient DEF is too- just try replacing the SIND  
and COSD functions and replacing them with the ordinary COS and SIN functions  
but amend the variable in accordance with the DEF - eg instead of  $COS(A)$  use  
 $COS(A/180*PI)$  in every SIND and COSD line. Any faster?

..  
1 REM **ANALYTIC COMPUTER ART**  
2 ! JOE JACOBSON  
3 ! APT 1009 675 E STREET RD  
4 ! WARMINSTER PA USA 18974  
5 ! fluted scallops  
6 ! The Journal of Chaos & Graphics August 1988  
7 ! For TI99/4A plus ExBas + The Missing Link by Stephen Shaw UK Jan  
91  
8 !  
100 DEF SIND(X)=SIN(X/180\*PI)  
110 DEF COSD(X)=COS(X/180\*PI)  
120 OS=99 :: RANDOMIZE  
130 L=INT(RND\*60)+3  
140 CALL LINK("PRINT",180,220,STR\$(L))  
150 FOR B=16 TO 76 STEP 2  
160 FOR A=0 TO 360 STEP 5  
170 GOSUB 250  
180 IF A>0 THEN 210  
190 CALL LINK("PIXEL",OS+X,Y+OS):: OLDX=X :: OLDY=Y  
200 GOTO 220  
210 CALL LINK("LINE",OS+OLDX,OLDY+OS,OS+X,OS+Y):: OLDX=X :: OLDY=Y  
220 NEXT A  
230 NEXT B  
240 GOTO 290  
250 R=B\*(1+.25\*ABS(SIND(L\*A)))! this is the function plotted:R=f(x):=polar coord  
inate curve.  
260 X=R\*COSD(A)  
270 Y=R\*SIND(A)  
280 RETURN  
290 CALL LINK("PRINT",180,20,"+")  
300 CALL LINK("DUMP")  
305 CALL LINK("CLEAR")  
310 RUN

.  
=====.



The disk library has recently received the text of the New Testament on disk. I thought it would be nice to have the computer read it to me... hence the following program. It has been put together specifically to deal with the Bible text, and only takes account of the letters A to Z, in upper OR lower case..

It is written to be used with the Terminal Emulator 2 (TE2) module and of course requires the speech synthesiser peripheral.. Using TE2 we program in Basic, and cannot use the XB form of LINPUT A\$ to read the text from disk. As some punctuation commonly found in text represents "end of record" when using Display format files, we have added a comma to the INPUT line, which ensures that we miss no text. In XB LINPUT ignores the commas and so on (=LINE INPUT)..

Apart from the need for the comma after A\$ there is nothing too strange about this little program. As TE2 only "speaks" upper case we have to go through the string to alter it all to upper case, and for the purpose of the intended usage, certain characters are excluded from the string to be spoken. The chapter and verse numbers used in the Bible are very out of place if you insert them as you read!.. Hope you find this of use!.

```
.
100 REM READ DV80 FILE
110 REM WITH EMBEDDED          COMMAS
120 REM AND LOWER CASE
130 REM AND USE TE2 TO READ
140 REM IT OUT LOUD.
150 REM THIS PROGRAM ONLY
160 REM SPEAKS A-Z and a-z
170 REM NUMBERS ARE REMOVED
180 REM RECOMMEND THAT TEXT    IS PLACED ON A RAM DISK    IF AT ALL POSSIBLE F
OR    SMALLER PAUSES.
190 REM
200 REM S SHAW FEB 91
210 REM
220 OPEN #1:"DSK3.LUKE1",INPUT
230 OPEN #2:"SPEECH",OUTPUT
240 INPUT #1:A$,
250 IF LEN(A$)<1 THEN 240
260 PRINT A$
270 A$="_"&A$
280 FOR T=1 TO LEN(A$)
290 A=ASC(SEG$(A$,T,1))
300 IF A>57 THEN 320
310 IF A<>32 THEN 360
320 IF A<95 THEN 350
330 A=A-32
340 IF A=61 THEN 360
350 B$=B$&CHR$(A)
360 NEXT T
370 A$=B$
380 B$=""
390 PRINT #2:A$
400 IF EOF(1)THEN 420
410 GOTO 240
420 CLOSE #1
430 CLOSE #2
440 END
```

## TI-Writer hints, tips, and answers

Compiled by Stephen Shaw

(F refers to a tip for the Formatter while E refers to a tip for the Editor section. Equivalents include Funnelweb.)

1F. Outdenting. This is the reverse of "indenting." It will allow the first line of a paragraph to be started further to the left than the remaining lines in the paragraph. Example: .LM12;IN7;RM71 causes the first line to start at column 7 and subsequent lines to begin at column 12.

2E. When using Replace String you should use word wrap OFF if the document lacks carriage returns or you will end up with one huge paragraph. With word wrap ON, Replace String will reformat from the amended word to the end of the paragraph.

3E. This is for use in Funnelweb Text Editor mode (not TI-Writer). Want to convert the character case (upper to lower; lower to upper)?

i. Upper to Lower depress CTRL . (period).

ii. Lower to Upper depress CTRL ; (semi-colon).

By keeping the keys depressed the auto-repeat function will take effect and every character the cursor passes over will be converted (you only need hold the , or ; once auto repeat takes over you can let go of CTRL).

4E. To save part of a document insert in front of the filename the first line number you wished saved then either a comma or a space and then the last line number you wished saved. Example: 5 30 DSK1.MINUTES will save lines 5 to 30, inclusive, to disk drive #1 under the name MINUTES.

5E. To "get rid" of the line numbers on the left of the screen press FCTN 0 (zero). To get them back press FCTN 0 again.

6E. If you erase a line in error, press CTRL 1 (Oops) and your line will be restored. Note: This usually only works if you have not pressed any other keys after the accidental erasing. There are a few exceptions, however. I have often had it happen that after editing several parts of a line and then making a mistake I tried to recover by using CTRL 1. By doing so I found that Oops! returned the entire line to what it was like before I started editing it!

7E. When using the "Search" command remember that the search is only from the point that the cursor is located. Therefore to search the total document the cursor must be on line 1 before you go to the Command mode.

8E. To backspace beyond the left margin press CTRL Y. This will temporarily disable the left margin. It will also disable the right margin in latest Funnelweb only. In both cases the cursor should be next to the margin.

9F. When using the Header or Footer command with the page number it is possible to have no value printed (such as for the introduction, etc) by using the .PA format command with a value of zero. The page numbering will begin on the following page. A .PA at the end of each page will delay the numbering further.

10E. TI-Writer can save a file in other than the normal D/V 80 format by using the PF command and either putting an "F" in front of the filename (i.e. F

DSK1.MYFILE) or by putting a "C" in front of the filename. "F" causes a file to be created in DISPLAY/FIXED 80 format. "C" strips any control characters from the file as it is sent.

11E. TI-Writer can be used as a database. Each line must be a record and set up exactly the same. For example if the data was names, addresses and phone numbers then all names must start in the same column; all addresses must start in the same column and all phone numbers must start in the same column. There can be no lines which are blank or which have other type of information on them (i.e. titles). Then using the program Sort Utility (by D.R. Romer and J. Clulow) you can sort this file. Once sorted, which is done very quickly, titles can be added if you are printing it.

12E. There are CTRL keys equivalents to most FCTN keys, plus a few others. For example to tab to the right you can go FCTN 7 or CTRL W (Funnelweb also has CTRL Z). To tab back (to the left) you can go CTRL T (there is no equivalent FCTN key).

13E. If you must go to the bottom of your text (and it is very long), instead of paging down simply go to the Command Line and press S for "Show Line" and at the prompt "Enter Line Number" just type E and press ENTER. E is a valid line number for the last line (end) of a document. This feature is active in all the commands requiring you to enter a line number.

14F. If you wish to prevent the form feed at the end of printing when using the Formatter then make the last line of your text ".PL 1". This will suppress the form feed, but note: do not forget to reset PL if you have another document to print.

15F. You can string the Formatter commands on the same line separated with a semicolon. Example: .LM 10;RM 70;IN +5;FI;AD  
There does not have to be a space between the Formatter command and the number which follows it. In other words, the commands .LM 4 and .LM4 are equivalent.

16F. If you are having problems with Formatter commands, make sure they are UPPER case letters.

17E. To avoid a Buffer Full notice, just keep saving the file as it gets larger, then use the SD command to see the file size. The Buffer is usually becoming full at 92 sectors. (Note: Due to the use of Run Length Encoding in the buffer area, a full buffer may occur at very different file sizes depending on nature of text.)

18F When using the .CE command you must also use the .LM and .RM commands because .CE centers the text between the SET margins, not necessarily the middle of your paper.

19F. The Formatter also ensures that you have two spaces after each period. To control this when you are typing such things as "Mrs. E Smith" or "1023 N. Queen Street" then use the "^" sign in place of the space after the period. Example:  
Mrs.\*E Smith, 1023 N.^Queen Street

20F. If you must have a dot in column one of your text, transliterate it. ".TL 124:46" will allow FCTN A to print a period. The reason for avoiding the dot (.) is because when the text is passed through the Formatter the whole line following the dot will be erased. If you are not passing it through the Formatter then there will be no problems.

21F. To create a file without line feeds yet formatted, do the following:

- i. Use the Formatter to Print the text to disk.
- ii. Go back to Editor, load the formatted text, and do a Print File (PF) replacing PIO with C DSKn.filename.

22F. If you wish to include a program listing in your document, instead of retyping it into TI-Writer just LIST the program to diskette using the following command:

```
LIST "DSKn.filename"
```

This will save the program in D/V 80 format which allows it to be read by TI-Writer. You can now load this file into TI-Writer and place the carriage return character at the end of each program line.

23E. If you wish to place a Carriage Return at the end of a line of text (a line without the return will usually occur when you have inserted blank lines in the text and then put text on them) then place the cursor at the end of the text and press CTRL 8. This will place a Carriage Return where you want it and insert a blank line below. If this line is not wanted you can delete it with FCTN 3. Or, alternatively, enter Special Character Mode by pressing CTRL U, press SHIFT M and then leave the Special Character Mode by pressing CTRL U again. This may seem like a lot, but when you get used to it you can perform the three-step operation in less than a second!

24E. You can get a printout of your file with line numbers when printing from the Editor mode by placing an "L" and a space before the printer name in the command instruction. Example: L PIO  
This will eliminate the last 6 characters at the end of each line (columns #75 to #80) therefore keep your line lengths to a maximum of 74.

25E. Did you know that you can type anything you want after a carriage return on the same line and it will not print, but it will Save. This is great for text notes for screen reading.

26E. When typing a document which uses certain long words or phrases frequently, then a time saver (and also added insurance against typing errors) is to type the words in short form or initials (i.e. TI-Artist could become TIA; The Ottawa TI-99/4A User Group could become TOTIUG). When you are finished with the document use the Replace String function (RS) to change the words back to the full spelling (e.g. /TIA/TI-Artist/). Care must be taken in three forms when using this:

- i. The search only begins from the spot that the cursor is on, so to do the whole document insure that the cursor is on line 1 before starting.
- ii. The search will locate all occurrences of the string. Therefore if the string searched for is "at" it will find word "at" and also the "at" in "cat" and "that", etc. So before telling the machines to "Change all occurrences automatically," be sure this condition cannot arise. If you are not confident of this it is best to walk through and change each separately as it is found.
- iii. As a reformatting will be done wherever a change is made, it would be wise to review the document afterwards to be sure that it is still formatted correctly.

27E. When using the Find String command you can specify which column range to search. Example: 5 15 /text/ will look for the string "text" in the columns 5 through to 15 inclusive. (Also available with Replace String.)

28F. If your printer does not have a slashed zero and you want to print it out that way use the following Transliteration:

```
.TL 48:48,8,47
```

This will cause the normal zero (48) to be printed; then backspace (8); and then a printing of a slash (47).

29E. Two files can be loaded into the Editor (assuming the total size is not too large for memory) by loading in the first file then doing a LF and entering E DSKn.YYY (where n = disk # and YYY = second file). This will load the second file after the end of the first file.

30E. If you do not like the windowing when using the 80-column format, then set the margins for 0 and 39 and turn off the line numbers (FCTN 0). When you are finished, reset the left and right margins to what you desire and reformat each paragraph.

31E. You can merge sections of a second file into the current document by the following entry using the LF command: 25 7 12 DSK1.YYY  
This will load lines 7 to 12 (inclusive) from file YYY to the current document after line 25.

32E. If you are using Funnelweb 4.1 or greater, after you have the directory on the screen (using the SD command) you will be able to see how many lines are in a file by marking the file and then requesting (V)iew. The line count will be shown at the bottom of the screen as you read through the document.

=====

Reminder re The Missing Link programs:

Because The Missing Link uses most of the VDP ram, in order to load programs into the console they must be saved to disk in the IV254 format - if you load a normal "Program" file the result will be at best some corruption of the image.

Rabbits and Foxes

Now for a graphic program requiring The Missing Link.

```
100 ! RABBITS AND FOXES
110 ! Dolores Garcia, Spain
120 ! Dr M Ecker, PA, USA
130 ! S SHAW FOR TI-99/4A, X
B+TML, April 91
140 CALL LINK("CLEAR"):: A,B
=5
150 CALL LINK("PRINT",A,B,"R
abbits and Foxes"):: A=A+12
:: B=5
160 CALL LINK("PRINT",A,B,"P
lease input the variables re
quested below. Suggested val
ues are given for a 'TYPICAL
' response. Try others!")
170 CALL LINK("PRINT",70,12,
"RABBIT BIRTH RATE:")
180 CALL LINK("PRINT",80,12,
"RABBIT DEATH RATE (Excl. ea
```

```

ten ones!):")
190 CALL LINK("PRINT",100,12
,"RATE OF EATEN RABBITS:")
200 CALL LINK("PRINT",110,12
,"FOXES DEATH RATE:")
210 CALL LINK("PRINT",120,12
,"FOXES BIRTH RATE:")
220 CALL LINK("INPUT",70,150
,A,4,".04"):: IF A<.00001 TH
EN A=.4
230 CALL LINK("INPUT",90,150
,B,6,".00005"):: IF B<LE-9 T
HEN B=.0005
240 CALL LINK("INPUT",100,15
0,C,4,".002"):: IF C<.00001
THEN C=.002
250 CALL LINK("INPUT",110,15
0,D,4,".03"):: IF D<.00001 T
HEN D=.04
260 CALL LINK("INPUT",120,15
0,E,5,".0002"):: IF E<1E-8 T
HEN E=.0002
270 CALL LINK("PRINT",140,12
,"RABBITS START AT:")
280 CALL LINK("PRINT",150,12
,"FOXES START AT:")
290 CALL LINK("INPUT",140,12
0,R,4,"300")
300 CALL LINK("INPUT",150,12
0,F,4,"30")
310 CALL LINK("CLEAR")
320 P=F*R
330 R=(1+A-B*R)*R-C*P
340 F=(1-D)*F+E*P
350 P=F*R
360 IF ER>500 THEN CALL LINK
(L80,L80,"---."):: GOTO 360
370 IF NOT(F>0 AND R>0 AND F
<200 AND R<640)THEN ER=ER+1
:: GOTO 330
380 CALL LINK("PIXEL",2+4*F,
R*.7+2):: CALL LINK("PRINT",
181,1,"RABBITS:"&STR$(INT(R)
)&" ")
390 CALL LINK("PRINT",181,11
0,"FOXES:"&STR$(INT(F))&" ")
400 ER=0
410 GOTO 330
420 END

```

=====

more....

=====.....

.....

for TI\*MES 33..

First item of news comes in a note from Dr Clifford Pickover, who advises me that his magazine Chaos and Graphics is now published commercially as a segment of "COMPUTERS AND GRAPHICS" from Pergammon Press. He also says he is to have a further book published by the end of the year, so that is one I shall be looking for..

===== .

.SAFE HEX - USE DIGITS.

You all know that 666 is the number of the Beast, but did you know His name? 666 is a SMITH number. Look:..

666 = 2 x 3 x 3 x 37 and.

2 + 3 + 3 + 3 + 7 = 18 which is also 6 + 6 + 6 !!!!!.

A Smith number has been defined by Harold Smith's brother in law, Prof Albert Wilansky, as being a non-prime number, with prime factor digits (excluding one as usual) adding to the same sum as the digits of that number..

666 is also a PSmith, as it is Palindromic- the same backwards as forwards. It is not a Smith brother however as 665 and 667 are neither of them Smiths..

.....

Remembering that prime numbers are excluded! there are 81 Smiths between 0 and 2000. Can you derive a program to list them?.

Can you derive a 3x3 magic square containing only Smith numbers?-columns and rows and diagonals all add up to the same total..

The highest known Smith number has 3681893 digits... a bit beyond our machine perhaps... it has been proven that there is an infinite number of Smiths and PSmiths but the proof is awaited that there are an infinite number of Smith brothers.....

=====.

**.UPDATE- WATERWORKS..**

In the last issue I mentioned this disk based game from Asgard, and later flagged it as bugged. .

Here is a fuller review.... .

The disk comes with a short manual, but the bulk of the rules are on disk to be viewed and/or printed out - instructions fill some 49 screens, and each prints out to the usual 24 lines. Thats a lot of docs to print out..

For the avoidance of any doubts, I did print it all out and read it before I played the game!.

For those of you who do not know, I have a seven year old in the house, who likes to see new games, and have them explained to him.

This makes a pause key useful, but in any case means that initially a little more time can be taken on a game! Having to print out all those docs did not please him too much....

And both he and I were a trifle (well, a lot actually) unhappy when after a fairly lengthy loading procedure, the game locked out on us- a full lock out with all keys inoperative, while an animated sprite ran havoc over the screen.....

The game? To put it as simply as possible, you run a pipeline from the top left of the screen to the bottom left. Your pipeline must be complete and leakfree and must pass through an increasing number of marked points. As the game progresses you get more obligatory locations to serve, and a long list of obstacles and hazards.. The documentation indicates a time limit to complete ALL floors and suggests an average time of 12 minutes per floor- ample for the first few floors..

When I play it the game locks out on the first screen after about two minutes. This is NOT a time-out end of life, it is a fully fledged lock out, requiring the game be reloaded to play again. This condition has been confirmed to me by other players. The problem in my opinion is sheer bad programming..

If you work fast enough you can get on to other screens. .

I subsequently found that if I obtained T-junction pipe and laid that, although I could then buy the needed end caps to seal off the pipe end NOT going to the termination point, I was unable to place the end stop. This pertains to the first six screens I got up to..

I also found that by a "legal" use of the hammer to destroy pipe wrongly laid, my score leapt upwards somewhat, and instead of a legal maximum of under a thousand I suddenly had a score of over 900,000.

Once more, in my opinion a mark of an appallingly written game..

At this point I gave up. .

There are numerous other bugs in this program (chewing gum fails to lose its grip after the documented 60 seconds maximum for example) but as rebooting is such a CHORE I have not explored them very thoroughly..

My supplier, Martin Blyth of Database, and (reported by Martin) the publisher, Chris Bobbitt of Asgard Software, consider this game to be of merchantable quality. In my opinion it is not..

It is possibly the worst commercial software I have seen in the last five years.

=====.....

Increasingly I find members are not quite aware of how the disk library operates... not too surprising as we have many new expanded members since this topic was last covered!.

All over the world there are TI owners who program. Some of their programs are well worth sharing, although commercial distribution may not be appropriate..

There is a need to hear of these programs, to obtain them, and to encourage more of them!.

Keeping in touch with the TI world is becoming increasingly more difficult, as the numbers of owners and programmers decreases, and keeping in direct contact with known active programmers is essential if we are to benefit from their work. This does cost money. .

Encouraging programmers to write, by showing appreciation both by comment and by financial reward, is essential if the TI is to continue to thrive. This can cost a lot of money, but not as much as if we had to rely on full priced commercial products for another computer! .

Typically, having heard of a useful utility, the disk librarian (me) will write directly to the author, with a suitable financial inducement- typically five to seven pounds- and in due course the disk will arrive for assessment. IF it is useful the program enters the disk library. NOT all submissions are added. A typical Utility disk, with programs from several authors, may represent expenditure of fifty pounds or more!!!.



These funds do not appear. They come from library charges, and the fewer members using the library, the fewer new programs can be added. There are user groups who spend a lot of time swapping programs with each other.

There are difficulties attached to this- often the programs are corrupted or incomplete (the cause of many complaints!); often they are old and long since updated and improved or debugged. More seriously, swapping programs between user groups is NOT an inducement to authors to write programs- they NEED user contact, and financial contributions are always welcome! -and makes it harder for new programs to enter the chain.

I will restrain myself from commenting on TI owners who sit around waiting for goodies to arrive without any payment or activity on their part and complain that the TI is dead 'cos nothing arrives! As with all things you get out what you put in. Put nothing in you get nothing out. The UK is already well known as a land of freebooters, contributing little in the way of programs, and (with some notable and honorable exceptions!) making no donations for faireware. If the TI is to maintain itself, we need to put in!

=====.

## **.GRAPHICS IMAGES.....**

There is a much increased number of programs for graphics on our computer right now, and they do NOT all print out the same!.

This article is addressed to answering the questions:.

1. How many sizes are available for printouts.
2. How much distortion is produced?.

The latter point is most valid.... if you draw a perfect square ON SCREEN how will it look on the printer? If you draw a perfect square by pixel count, how will it appear on a print out? If you ask most graphics programs to produce a circle, you end up with a rather fat squashed looking oval! If you print that to look like a true circle, then what looks like a square on screen will end up being tall and thin....

First a list of the programs, and some of their various print sizes, marked with an indicator to make listing later less space consuming:.  
Units of measure are for a sample square. To produce full picture sizes you need to know that a full screen printed as item A below measures 68mm x 119mm..

The first measure given is the height of the square, and if the picture is rotated when printed, this will become-on paper- the width. .

The ratio quoted is the height divided by the width, with a low number resulting in a low fat squashed picture, while a high number will produce a tall thin elongated picture..

To produce a good print of a "circle" or pixel-count square, look for a value of between 1.1 and 1.2, while to produce a reasonable copy of what is on screen look for values between 0.97 and 1.00..

|                                                                   | HEIGHT | WIDTH | RATIO  |
|-------------------------------------------------------------------|--------|-------|--------|
| A. TI ARTIST VERSION 2. Print options 1-1-8.....                  | 36     | 37    | 0.973. |
| B. TI ARTIST VERSION 2. Print options 1-2-8.....                  | 73     | 74    | 0.986. |
| C. ARTIST PHOTO 2. SIZE A.....                                    | 18     | 16    | 1.125. |
| D. ARTIST PHOTO 2. SIZE B.....                                    | 36     | 30    | 1.200. |
| E. ARTIST PHOTO 2. SIZE C.....                                    | 73     | 61    | 1.197. |
| F. ARTIST PHOTO 2. SIZE D.....                                    | 110    | 93    | 1.183. |
| G. TI ARTIST Vn3. UP. 1/2. 1 pic.....                             | 70     | 75    | 0.933. |
| H. TI ARTIST Vn3. ROTATED. 1/2. 1 PIC.....                        | 88     | 59    | 1.492. |
| I. TI ARTIST Vn3. UP. 1/2. 2 PICS.....                            | 35     | 37    | 0.946. |
| J. TI ARTIST Vn3. UP. 1/2. 3 PICS.....                            | 35     | 19    | 1.842. |
| K. TI ARTIST Vn3. ROTATED. 1/2. 2 PICS.....                       | 44     | 29    | 1.517. |
| L. TI ARTIST Vn3. ROTATED. FULL. 3 PICS.....                      | 22     | 59    | 0.373. |
| M. MAX RLE.....only one size!.....                                | 36     | 37    | 0.973. |
| N. JOYPAINT PAL. NORMAL.....                                      | 36     | 37    | 0.973. |
| O. JOYPAINT PAL. DOUBLE.....                                      | 88     | 62    | 1.419. |
| P. TIPS 1.8 CARD 3/4 and LABEL.....                               | 18     | 18    | 1.000. |
| Q. TIPS 1.8 CARD-IMAGE.....                                       | 73     | 74    | 0.986. |
| R. TIPS 1.8 SIGN-POSTER-IMAGE & PROCESS-PIC-TOTEM...146...149.... | 146    | 149   | 0.980. |
| S. TIPS 1.8 PROCESS-PIC-BANNER.....                               | 176    | 123   | 1.430. |

Now in order of image height: And in order of ratios:.

|                    |                 |
|--------------------|-----------------|
| for sampled square | 0.373....L.     |
| 18mm...C,P         | 0.933....G.     |
| 22.....L           | 0.946....I.     |
| 35.....I,J         | 0.973....A,M,N. |
| 36.....A,D,M,N     | 0.980....R.     |
| 44.....K           | 0.986....B,Q.   |
| 70.....G           | 1.000....P.     |
| 73.....B,E,Q       | 1.125....C.     |
| 88.....H,O         | 1.183....F.     |
| 110....F           | 1.197....E.     |
| 146....R           | 1.200....D.     |
| 176....S           | 1.419....O.     |
|                    | 1.430....S.     |

All measurements taken to nearest millimetre. Printer friction fed so also allow a little for slippage..

I have not listed the excellent package SMARTCOPY by Alexander Hulpke as that is a very versatile program allowing fine adjustment to printed ratios, especially for printing larger images where pixel multipliers can be very finely tuned.

.....

===== END OF JUNE 1991

From TI\*MES 34:

Reminder re The Missing Link programs:

Because The Missing Link uses most of the VDP ram, in order to load programs into the console they must be saved to disk in the IV254 format - if you load a normal "Program" file the result will be at best some corruption of the image. If you write a program while TML is loaded and then SAVE it to disk it will be in IV254 format.

## ATTRACTION BASINS

Here is a truly fractal program which, using The Missing Link and Extended Basic, only takes about 12 hours or so to plot a graphic of  $120 \times 200$  pixels!

The total graphic is vertically reflected about 0, so if 0 is at screen center you can plot top and bottom parts at the same time to halve plotting time...

This program is all about "attraction basins" which I do not understand, and while the program can probably be modified for other functions, I do not know how! What I know is that it plots a truly fractal shape which we can zoom in on.

```
100 ! TO DRAW AN ATTRACTION
BASIN USING NEWTON FORMULA 0
N:
110 ! F(z)=z^(-3)-1
120 !
130 !
140 ! J C TOPHAM ! Fractal R
eport 13 ! Feb 1991
150 ! for ti+tml by S Shaw m
arch 1991
200 WIDE=242 :: HEIGHT=190
210 !
220 !
230 CALL LINK("CLEAR")
239 ! actual pixels plotted:
240 ACROSS=200 :: DOWN=120
246 ! area to be plotted
247 ! whole image is within
X=-2 to +2
248 ! Y= -1.75 to +1.75
249 ! Y is vertical and is r
eflected about 0.
250 XMIN=-1.1 :: XMAX=0.9
260 YMIN=-1.70 :: YMAX=0.30
270 !
280 !
290 RY(1)=-SQR(3)/2
300 RY(2)=SQR(3)/2
310 RY(3)=0
320 !
330 DX=(XMAX-XMIN)/ACROSS
340 DY=(YMAX-YMIN)/DOWN
350 XPOS=INT((WIDE-ACROSS)/2
)
360 YPOS=INT((HEIGHT-DOWN)/2
)
370 !
380 FOR YP=0 TO DOWN
390 FOR XP=0 TO ACROSS
400 CALL LINK("PRINT",181,12
,STR$(YP)&":"&STR$(XP))
410 YN=YMIN+YP*DY
420 XN=XMIN+XP*DX
430 FOR ITER=1 TO 30
```

```

440 GOSUB 650
450 XM=(A*C+B*D)/9
460 YM=(B*C-A*D)/9
470 IF XM*XM+YM*YM>1000 THEN
  620
480 !
490 !
500 IF ABS(YN-YM)>1E-2 THEN
590
510 IF ABS(XN-XM)>1E-2 THEN
580
520 FOR I=1 TO 3
530 IF ABS(RY(I)-YM)>1E-2 TH
EN 560
540 CALL LINK("PRINT",1,200,
STR$(ITER)&" "): : IF ITER/20
INT(ITER/2)THEN ITER=30 : :
GOTO 570
550 CALL LINK("PIXEL",YPOS+D
OWN-YP,XPOS+XP): : ITER=300
560 !
570 NEXT I
580 !
590 !
600 XN=XM : : YN=YM
610 NEXT ITER
620 NEXT XP
630 NEXT YP
640 END
650 ! INVERT Z3
655 X2=XN*XN : : X3=X2*XN : :
X4=X2*X2 : : Y2=YN*YN : : Y3=Y
2*YN : : Y4=Y2*Y2
660 A=4*XN-X4+6*X2*Y2-Y4
670 B=4*YN-4*X3*YN+4*XN*Y3
680 C=3 : : D=0
690 RETURN

```

```

=====
=====

```

Fractal Graphics with The Missing Link by Stephen Shaw

### **Mandelbrot plot**

This is exceedingly slow if you want a detailed plot! However, a minimum detailed plot over a limited screen area is not too slow! Maximum detail and you could be looking at several days to complete a screenful.

This program does work, and cries out for some machine code! Some means of saving the picture to disk or printer would be required that is perhaps the hardest bit.

User should input area to be plotted, and possibly be given a choice of what method is to be used for deciding if a pixel is to be plotted or not see comments in listing! and the degree of inner detail required.

listing follows....

```
100 ! MANDELBROT PLOTTER
110 ! VERY SLOW!
120 ! FOR EX BAS + THE MISSING LINK
130 !
140 CALL LINK("CLEAR")
150 !
160 ! DESIGN LIES IN AREA
170 ! PMIN -2.25 PMAX 0.75
180 ! QMIN -1.50 QMAX +1.50
190 !
200 ! use smaller area for more detail - ensure there IS detail there though!
210 !
220 ! Concentrate on areas very close to central creature!
230 !
240 A=240 :: B=180 ! maximum values
250 !
260 ! A=width plotted B=height plotted in pixels
270 !
280 ! M= number of iterations per point. Needs to be enough to reach CMAX at centre of beast.
290 ! 100 is often used but lower values can usually be safely used.
300 !
310 !
320 A=160 ! A=width
330 B=150 ! B=height in pixels
340 M=71 ! MAX ITERATIONS
350 ! P=REAL Q=IMAG
360 PMIN=-.250 ! DEFINE AREA
370 PMAX=-.20 ! TO BE
380 QMIN=-0.83 ! DRAWN
390 QMAX=-0.79 !
400 !
410 ! CMAX must be a power of 2, eg 2,4,8,16,32,64,128..
420 ! The higher the value the more detail close to the beast.
430 ! 16 gives minimum acceptable detail, 32 is reasonable for an odd/even test, while 64 and higher give maybe too much detail.
440 ! If using 64 or higher you must use a logarithmic scale with greater gaps as K
```

```

gets larger... eg
450 ! 64...pixel on. 53 to 6
3...pixel off K= 44 to 52..
.pixel on K= 36 to 43...pixe
l off K= 29 to 35...pixel o
n...
460 !
470 ! or use IF INT(K^.8)/2=
INT(INT(K^.8)/2) THEN pixel
on... or some other fraction
al power
480 ! note that processing t
ime really does get longer a
s CMAX increases!
490 !
500 !
510 CMAX=64 ! DETAIL, 8,16,3
2,64,128,256 ETC
520 !
530 DP=(PMAX-PMIN)/(A-1)
540 DQ=(QMAX-QMIN)/(B-1)
550 FOR NP=1 TO A
560 FOR NQ=1 TO B
570 P=PMIN+NP*DP :: Q=QMIN+N
Q*DQ :: K,X,Y=0
580 ! LOOP
590 !
600 !
610 XN=X*X-Y*Y+P
620 Y=2*X*Y+Q
630 X=XN :: K=K+1
640 !
650 IF X*X+Y*Y>M THEN 690
660 IF K=CMAX THEN 690
670 GOTO 580
680 !
690 IF K/2=INT(K/2)THEN CALL
LINK("PIXEL",NQ,NP)! SEE C
OMMENTS ON CMAX ABOVEI
700 !
710 CALL LINK("PRINT",170,20
0,STR$(NQ)&STR$(K)&" ")
720 NEXT NQ
730 NEXT NP
740 CALL LINK("SAVEP", "DSK2.
PIC2")
750 GOTO 750

```

```

=====
=====

```

Fractal Graphics with The Missing Link by Stephen Shaw  
**The Golden Ratio**

```
1 ! THE GOLDEN RATIO - 1: 1.618...
2 ! removing a square from a rectangle with sides in this ratio leaves a smaller rectangle which also has sides in the Golden Ratio
3 ! -points dividing sides lie on a logarithmic spiral which can be found in shells, and in art works by da Vinci, Dali - even in
4 ! the Parthenon.
5 ! Fractal in nature - it keeps getting smaller or larger depending on which way you go...
6 ! program written by Ashley Tilling for JBM103 converted by S Shaw for TML.
7 !
8 ! Due to our not having square pixels the rectangle on screen is NOT in the golden ratio as viewed. Ah well...
9 !
99 ! remove ! from line 100 for use with JBM103.
100 ! CALL LOAD(-31890,56,0)
:: CALL LOAD(-31964,56,0)
110 AX=12 :: AY=8
120 CL=200 ! side length
130 CALL LINK("CLEAR")!for jbm103 CALL LINK("SCR2") also
.
140 RL=INT(CL/1.618)
150 FOR I=0 TO 1 :: FOR J=0 TO 1
160 CALL LINK("LINE",AX+RL*I,AY+CL*I,AX+RL*J,AY-CL*(J=0))
)
162 ! CALL LINK("LIGNE",16,AX+RL*I,AY+CL*I,AX+RL*J,AY-CL*(J=0))
170 NEXT J :: NEXT I
180 FOR K=1 TO 3
190 M=CL-RL :: N=INT(CL*.236)
:: P=INT(CL*.146)
195 CALL LINE(AX,AY,AX,AY)
200 CALL LINE(RL,M,1,M)
210 CALL LINE(N,1,N,M)
220 CALL LINE(1,N,N,N)
230 CALL LINE(P,M,P,N)
```

more....>

```
240 AX=AX+P :: AY=AY+N :: CL
=P :: RL=INT(CL/1.618)
250 NEXT K
260 GOTO 260
270 SUB LINE(A,B,C,D)
280 REM ! LINES for JBMI03 I
NSTEAD of CALL LINK("LINE"..
and CALL LINK("PIXEL"...
290 IF (A=C)*(B=D)THEN AX=A
:: AY=B :: SUBEXIT
300 CALL LINK("LINE",AX+A,AY
+B,AX+C,AY+D)
310 ! CALL LINK("LIGNE",16,A
X+A,AY+B,AX+X,AY+D)
320 IF B=D THEN 350
330 L=ABS(B-D)-1
340 ST=-PI*(B>D):: FI=PI/2+P
I*(D>B):: GOTO 370
350 L=ABS(A-C)
360 ST=-PI/2-PI*(A>C):: FI=P
I*(C>A)
370 IF L<3 THEN SUBEXIT
380 FOR J=ST TO FI STEP -1/L
390 Y=AY+D+INT(.5+L*SIN(J))
400 X=AX+C+INT(.5+L*COS(J))
410 CALL LINK("PIXEL",X,Y)
420 ! CALL LINK("POINT",16,X
,Y)
430 NEXT J
440 SUBEND
450 END
```

=====  
=====

more content follows:



## Polynomial equations

```
1 ! Newton's method for solving polynomial equations.
2 ! eg find z when  $4Z^6-2Z^3+z-1=0$ 
3 ! sorry about the input format, you just need to enter
   the multiplicand for each power of z and its sign if negative (6z or -6z)
4 ! enter the default zero if that power is not in the equation.
5 ! one numeric answer will usually be given but some formulae may have more than one answer- try varying the seed
6 ! in line 290, value of X
7 ! to produce different answers.
8 ! a few equations MAY not be solvable with this program
9 ! amend input method if using multipliers over +99 or under -9 or if you wish to use higher powers (remember to DIM the array).
10 !
11 !
100 CALL CLEAR :: V$="0123456789-+"
110 DISPLAY AT(1,1):"NEWTONS METHOD TO SOLVE Polynomial Equations"
120 DISPLAY AT(3,4):" Dr M Ecker 1987.5 Shaw for TI 1991"
130 DISPLAY AT(6,1):".Your equation is in the form  $5Z^5+0Z^4-2Z^3+0Z^2-2Z+9=0$ "
140 DISPLAY AT(12,1):" 0 Z^9 0 Z^8 0 Z^7 0 Z^6"
150 DISPLAY AT(13,1):" 0 Z^5 0 Z^4 0 Z^3 0 Z^2"
160 DISPLAY AT(14,1):" 0 Z 0 = 0"
170 ACCEPT AT(12,2)SIZE(-2)VALIDATE(V$):A(9)
180 ACCEPT AT(12,8)SIZE(-2)VALIDATE(V$):A(8)
190 ACCEPT AT(12,14)SIZE(-2)VALIDATE(V$):A(7)
200 ACCEPT AT(12,20)SIZE(-2)VALIDATE(V$):A(6)
210 ACCEPT AT(13,2)SIZE(-2)VALIDATE(V$):A(5) ! more-->
```

```

220 ACCEPT AT(13,8)SIZE(-2)V
ALIDATE(V$):A(4)
230 ACCEPT AT(13,14)SIZE(-2)
VALIDATE(V$):A(3)
240 ACCEPT AT(13,20)SIZE(-2)
VALIDATE(V$):A(2)
250 ACCEPT AT(14,2)SIZE(-2)V
ALIDATE(V$):A(1)
260 ACCEPT AT(14,6)SIZE(-2)V
ALIDATE(V$):A(0)
270 FOR T=0 TO 9 :: IF A(T)<
>0 THEN N=T
280 NEXT T
290 TLD,TL=.00000000001 :: X
=.800 ! INITIAL GUESS
300 FOR K=0 TO N
310 Y=Y+A(K)*X^K
320 NEXT K
330 IF ABS(Y-0)<TL THEN 410
340 FOR K=1 TO N
350 YD=YD+K*A(K)*X^(K-1)
360 NEXT K
370 IF YD=0 THEN X=X+.01 ::
GOTO 400
380 X=X-Y/YD
390 DISPLAY AT(22,1):"Next i
terate:"X;" "
400 Y,YD=0 :: GOTO 300
410 DISPLAY AT(20,1):"***DON
E***":"<any key for another"
420 DISPLAY AT(16,3):"Z=";X
430 CALL HCHAR(18,6,42,6)
440 CALL KEY(5,P,B)
450 IF B=1 THEN RUN
460 CALL HCHAR(18,6,32,6)
470 CALL HCHAR(18,6,45,6)
480 GOTO 430
491 ! Recreational & Educati
onal Computing
492 ! 909 Violet Terrace
493 ! Clarks Summit PA
494 ! USA 18411
495 ! 1991 sub Europe US$36
or
496 ! send $12 up for sample
copies
500 END

```

I gather this method is best for orders up to a power of 6 but better methods are available for higher orders - so I am told.

=====

```
Adding Fractions1 ! ADDING FRACTIONS
2 ! result is not reduced e.
g. 12/16 would usually be shown as 3/4
100 CALL CLEAR
110 DISPLAY AT(10,5):"--- +
--- = ---"
120 ACCEPT AT(9,5)SIZE(3)VALIDATE(DIGIT):A
130 ACCEPT AT(11,5)VALIDATE(DIGIT)SIZE(3):B
140 ACCEPT AT(9,11)SIZE(3)VALIDATE(DIGIT):C
150 ACCEPT AT(11,11)SIZE(3)VALIDATE(DIGIT):D
160 GOSUB 230
170 DISPLAY AT(9,16):USING "####":N
180 DISPLAY AT(11,16):USING "####":L
190 DISPLAY AT(14,1):"ENTER KEY FOR ANOTHER"
200 DISPLAY AT(1,1):"NORMAL RESULT=";A/B+C/D
210 ACCEPT AT(24,12):A$ :: GOTO 100
220 STOP
230 FOR X=2 TO B*D
240 IF INT(X/B)<X/B THEN 260
250 IF INT(X/D)=X/D THEN 270
260 NEXT X
270 L=X
280 N=INT(L/B)*A+INT(L/D)*C
290 RETURN
```

=====

more follows....

## PI GENERATOR

PI is the ratio of a circle's circumference (distance all the way round) to its radius (distance from center to edge). In Extended Basic we have a fixed variable PI we can use. In TI Basic we can use the function ATN to derive PI. In Assembly Language or most other languages we are on our own! We could approximate PI to 3.14, but what if greater accuracy is required? This little program derives PI to as many places as our language can support, without using any trigonometry, and is quite remarkably fast!

You may note the use of AND 2, which is using AND as a "logical operator," not too well covered in the manual. In Extended Basic, type in:

```
FOR T=1 TO 20 :: PRINT T;T AND 2 :: NEXT T
and inspect the result!
```

```
1 ! CALCULATE PI
2 ! QUICKLY AND ACCURATELY
3 ! from W.F. DOSSET, AUSTIN
  , TX, 25 feb 88
4 ! for TI-99/4A S. SHAW 3/9
  1
5 ! from John Machin, 1706
7 ! n=7 for 13-digit accurac
y of TI-99/4A Basic
8 ! n>7 if language goes bey
ond 13 digits!
9 ! e.g. n=35 for 53 digits
10 !
100 N=7 :: K=-1
110 K=K+1
120 I=-((2 AND 2*K)-1)
130 U=U+4*I/((2*K+1)*5^(2*K+
1))
140 V=V+I/((2*K+1)*239^(2*K+
1))
150 IF K<=N THEN 110
160 W=U-V :: PI2=4*W
170 ! DONE
180 PRINT PI2,PI
190 PRINT(PI2-PI)*1000, ! CO
MPARE with on-board variable
200 PRINT(PI2-(4*ATN(1)))*10
00 ! COMPARE with TI Basic's
  calculation
210 PRINT " ";(PI2-3.14)*1
00;:: DISPLAY AT(24,2)SIZE(4
):"3.14" ! displays ALL the
digits actually in the varia
ble!!!=13 incl dec point.
220 PRINT " 3.14159265358979
-etc" ! more accurate actua
l value for comparison!
230 END
999 ! from The Rec Newsleter , March 1988, Volume 3 No. 2
```

The program is not derived from the standard mathematical series, stretching off to infinity, but from an approximation published around 1706, a little before computers were around! This, courtesy of Mr. Dossett, has yielded a rather fast program!

=====

## Primes

Continuing with math related topics, prime numbers- that is numbers which are divisible only by themselves and 1. Asked to write a program to list prime numbers I suspect most of us would start from 1 and then try dividing every number by every number to see if it was divisible...

The first listing is fairly straight forward, but uses a few tricks to speed things up. In particular, if we know that a number is NOT divisible by two, it cannot possibly be divisible by any higher power of two, such as 4- or indeed any multiple of two. So we can check just the odd numbers, and also not bother testing for divisors greater than the first power of a known prime.

```
100 ! First 100 primes -neat
ly-
110 ! from Loren Krienke via
    The Rec Newsletter, March 1
    988, Volume 3 No. 2
120 ! for TI-99/4A S. SHAW
130 L=100 ! number required
140 DIM P(100)! P(L)-holds f
ound primes, used as divisor
s
150 DIM S(100)! S(L)-squares
    of primes, reduces number o
f divisions needed
160 P(1)=2 ! first prime
170 PRINT P(1);
180 T=1 ! initialize loop
190 FOR I=2 TO L ! test non-
even numbers:
200 T=T+2
210 FOR J=1 TO I-1
220 CALL MOD(T,P(J),R):: IF
R=0 THEN 200
230 IF T>S(J)THEN 235 ELSE 2
40
235 NEXT J
240 P(I)=T ! found prime, st
ore it
250 S(I)=T*T ! GREATER SPEED
260 PRINT T; ! PRIME
270 NEXT I ! LOOP
280 STOP
290 ! 18 July 1986
300 ! Loren Krienke
310 ! San Diego, CA
320 !
330 SUB MOD(A,B,C)! C=A MOD
B becomes CALL MOD(A,B,C)
340 C=INT(A-B*INT(A/B))
350 SUBEND
```

A similar but very much more opaque program follows, which has no obvious connection with the above, but it produces the same output!!! Check out the speed comparison between the two programs. Any difference? You may also wish to try looking for a larger number of primes!

```
100 ! First 100 primes -quickly-
110 ! Dr H.B. Phillips from
The Rec Newsletter, March 19
88, Volume 3 No. 2
120 DIM P(300),X(12)
130 A=0 :: B=1 :: D=0.5 :: E
=180
140 M=100 :: L=3 :: F=0
150 ! increase M for more- also
increase DIMs.
160 PRINT 2;:: C=B :: IF M=B
THEN END
170 L=INT((M/C)*L+F):: N=L+L
+B
180 FOR I=B TO INT((SQR(N)-B
)*D):: PP=P(I)
190 IF PP=B THEN 230
200 IF PP=A THEN PP=I+I+B ::
PRINT PP;:: P(I)=PP :: C=C+
B :: IF C=M THEN END
210 IF X(I)=A THEN X(I)=(PP*
PP-B)*D
220 FOR J=X(I) TO L STEP PP :
: P(J)=B :: NEXT J :: X(I)=J
230 NEXT I :: IF F=0 THEN S=
I
240 FOR I=S TO L
250 IF P(I)=A THEN PP=I+I+B
:: PRINT PP;:: P(I)=PP :: C=
C+B :: IF C=M THEN END
260 NEXT I :: F=(M-C)*L/E ::
S=L+B
270 GOTO 170
```

And no, I have no idea how this program works! When faced with a problem, the programmer usually has more than one way of attacking it, and it also usually happens that by restating the problem a faster solution can be found.

=====

## Worm

This is a tricky little program for The Missing Link. In this format it is a demo program, but if you leave the worm invisible all the time (by leaving Pen Erase on all the time) then you have a nasty little trick especially if you disable BREAK and QUIT. This is the fun version.

```
100 A,B=10
110 CALL LINK("PRINT",A,B,"The Missing Link is a disk based utility which allows the Extended Basic programmer to make use of many ")
120 CALL LINK("PRINT",A,B,"enhanced commands, including true bit map graphics, which in turn allows more powerful PRINT commands, and of ")
130 CALL LINK("PRINT",A,B,"course, full access to sprites.")
140 CALL LINK("PRINT",A,B,"This program demonstrates graphics outside a small window which, worm fashion, eat up the text! Enjoy..")
150 ! WORM
160 ! Scientific American, Dec 87 by S.L. Wentworth
170 ! TI-99/4A BY S. SHAW 91
180 ! XB + THE MISSING LINK
190 DIM XC(25),YC(25)
200 XC(2),YC(2),XC(1),YC(1)=100
210 D=0
220 T=1
230 RANDOMIZE
240 CALL LINK("WINDOW",80,80,110,120,1)
250 CALL LINK("PRINT",7,7,"WORM"):: CALL LINK("PRINT",16,7,"HOLE")
260 CALL LINK("REVWIN")
270 WT=T
280 CALL MOD(T,25,T):: T=T+1
290 CALL LINK("PE")
300 CALL LINK("CIRCLE",XC(T),YC(T),4,0)
310 CALL LINK("PD")
320 C=RND
330 IF C<.5 THEN D=D+.1745 ELSE D=D-.1745
340 X=XC(WT)
350 Y=YC(WT)
360 NX=X+4*COS(D)
370 NY=Y+4*SIN(D)
380 IF NX>189 THEN NX=NX-189 ! continued
```

```

390 IF NY>240 THEN NY=NY-240
400 IF NX<2 THEN NX=NX+189
410 IF NY<1 THEN NY=NY+240
420 XC(T)=NX
430 YC(T)=NY
440 !
450 CALL LINK("CIRCLE",NX,NY
,4,0)
460 CALL LINK("PD")
470 GOTO 270
480 SUB MOD(A,B,C)
490 C=INT(A-B*INT(A/B))
500 SUBEND

```

=====

### Fractional plus and minus

This is the promised fractional plus and minus program. I will not tell you how long it took me to get the displays like that ...

```

100 ! FRACTIONAL + AND -
110 ! R. Caldwell, Jan 91
120 ! FOR TI-99/4A BY
130 ! S. Shaw Apr 91
140 DIM Q(102)
150 DISPLAY AT(1,1)ERASE ALL
:"FRACTIONAL + & -"
160 DISPLAY AT(7,1):"-----"
"

170 DE,X=1
180 ACCEPT AT(6,2)VALIDATE(D
IGIT,"+-")SIZE(4):N(X)
190 IF N(X)=0 THEN 310
200 ACCEPT AT(8,2)VALIDATE(D
IGIT)SIZE(4):D(X)
210 IF D(X)=0 THEN 310
220 DP(X)=N(X):: DE=DE*D(X)
230 CALL HCHAR(6,1,32,12)
240 CALL HCHAR(8,1,32,12)
250 DISPLAY AT(5,1):"PLUS:"
260 DISPLAY AT(3,1):"ENTER 0
TO TOTAL"
270 X=X+1 :: DISPLAY AT(7,15
):"ITEM";X :: DISPLAY AT(8,1
5):"-MAX 10-"
280 IF X=11 THEN 310
290 GOTO 180
300 REM
310 F,X=X-1
320 FOR S=1 TO F :: FOR J=1
TO F
330 Z=Z+1 :: K=S+J-1
340 P(S,J)=K-F*INT(K/F):: IF
P(S,J)=0 THEN P(S,J)=F
350 Q(Z)=P(S,J):: NEXT J ::
NEXT S
360 FOR X=1 TO F :: Y=F*X-F+
1 :: FOR C=1 TO F-1 ! more--->

```



```

370 Y=Y+1 :: DP(X)=DP(X)*D(Q
(Y)):: NEXT C
380 NU=NU+DP(X):: NEXT X ::
DD=2
390 DISPLAY AT(10,1):"SUM IS
:" :: DISPLAY AT(11,5):NU;" /
";DE
400 IF NU/DD=INT(NU/DD)AND D
E/DD=INT(DE/DD)THEN NU=NU/DD
:: DE=DE/DD :: GOTO 400
410 DD=DD+2+(DD=2)
420 IF ABS(NU)>DE THEN A=ABS
(NU)ELSE A=DE
430 IF DD<=SQR(A)THEN 400
440 IF NU/DE=INT(NU/DE)THEN
540
450 IF NU>0 THEN WN=INT(NU/D
E):: NU=NU-WN*DE :: PN=1
460 IF NU<0 THEN WN=INT(NU/D
E)+1 :: NU=ABS(NU-WN*DE):: P
N=2 :: IF WN=0 THEN NU=-NU
470 REM
480 DISPLAY AT(13,5):USING "
#####":NU
490 IF PN=1 THEN DISPLAY AT(
14,1):WN :: NU=NU+WN*DE
500 IF PN=2 THEN DISPLAY AT(
14,1):WN :: IF WN<0 THEN NU=
WN*DE-NU
510 PN=0
520 DISPLAY AT(14,5):"-----
_"
530 DISPLAY AT(15,5):USING "
#####":DE
540 DISPLAY AT(18,4):NU/DE
550 DISPLAY AT(24,4):"ANY KE
Y FOR ANOTHER"
560 CALL KEY(5,G,H):: IF H<1
THEN 560
570 RUN
580 END

```

=====

TI\*MES Issue 35: Winter 1991-1992

May I remind you any program yet published by Regena, in any publication, is available on disk or tape from the author.

Additionally, the following books are available from Regena:

Programmer's Reference Guide to the TI-99/4A. 358 pages. USD\$15.00.

First Book of TI Games. (29 listings). USD \$13.00.

Basic Programs for Small Computers (including TI-99/4A) (40 programs). US\$13.00 and on disk ...

All programs by Regena published in MICROpendium in 1989 for \$5 (one disk):

San Diego Guide; Bake a Cake with print option; printing random math problems; knit a sweater for 25" to 35" chest using different gauges; 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):

San Diego Guide; Bake a Cake with print option; printing random math problems; knit a sweater for 25" to 35" chest using different gauges; 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 Theorems;

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: Disk of 16 game programs for TI-99/4A, by Regena**

Cost: USD \$10 From: Regena, USA.

Sixteen programs on disk for \$10 cannot be 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.

Color Code is not unlike Master Mind, 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!).

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. 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. Poker Solitaire is very like the Pewterware program "Challenge Poker" but without the wild card! You arrange cards in a 5 x 5 grid to try to score poker hands. Overall a good collection of programs with something for everyone. Everything will run in TI Basic, and only two require VDP or BXB to run in Extended Basic. Worth looking at.

=====  
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 so here is an extract from the publisher's announcement:

### **Computers and the Imagination by 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 visualization 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 color. Publisher: Alan Sutton Publishing. ISBN: 0 86299 999 5.

### **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(10*RND);
130  NEXT T
140  RUN "DSK1.LOAD"
```

and save this little program on to a new disk in Drive 1 as "LOAD". Now re-boot 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 and 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 ALSAVE, 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
* 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
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
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

KEYIN
    CLR @KEYADR            CLEAR KEY-UNIT
    MOVB @>83D7,@>83C1 TAKE THE SCREEN TIMEOUT'S LOW BYTE INTO SEED + 1
    CLR @STATUS            CLEAR GPL STATUS
    BLWP @KSCAN            SCAN KEYBOARD
    LIM 2                  ALLOW INTERRUPTS
    LIM 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
                           CONTINUED>
    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 Editor/Assembler or Mini Memory 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 ALSAVE, EXTENDED BASIC
* TWO CALL LINKS ARE INCLUDED
* 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
* 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
                        CONTINUED....
        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
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

```

=====

## Pre-review: Scud Busters, 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.

Bruce Harrison is a first-rate Assembly Language programmer, and the quality of programming can be seen in the highest speed of the defence missile! 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.

An interesting touch is that from pressing fire to the time your missile arrives at the marked point is virtually a constant time. This makes for interesting strategy you can move the aiming dot to far screen right to make your defense 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.

=====

Riemann Square graphic program for The Missing Link plus 32k ram  
by Stephen Shaw

This program illustrates how to project a graphic on to 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 do not 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 confusing, 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 March 9, 1991 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 (August 1991)

This program allows you to view the globe from any direction. The direction is given by three inputs, which are the angle to the center of the sphere from each of the three planes (left-right, up-down, towards-away) passing through the center. 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. CAST
LE-SMITH 3/91 for TI by S. S
HAW 8/91
110 ! XB + 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 SHO
RT(A(T)):: NEXT T
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 THE
N 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)))
310 CALL LINK("PRINT",180,1,
"PLOBAC="&STR$(PLOBAC)&"(0=N
O PLOT ON BACK)")
320 !
```



```

330 !
340 CALL LINK("CIRCLE",101,1
01,70)
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=-SQ
UARE TO SQUARE STEP SQUARE/1
0 :: GOSUB 630 :: NEXT XC
530 XC=SQUARE :: FOR YC=-SQU
ARE TO SQUARE STEP SQUARE/10
:: GOSUB 630 :: NEXT YC
540 YC=SQUARE :: FOR XC=SQUA
RE TO -SQUARE STEP -SQUARE/1
0 :: GOSUB 630 :: NEXT XC
550 XC=-SQUARE :: FOR YC=SQU
ARE 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
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 T
HEN 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(OLDY
,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 THE
N SUBEXIT
790 OLX=OLDX+101 :: X=X+101
:: OLY=OLDY+101 :: Y=Y+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.

Projecting a TI Artist Picture onto the Sphere:

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 Expansion Memory 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 D/V 80 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).

The program is used to create a D/V 80 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.MYARTIS
T") ! [from disk library]
120 CALL GRAPHICS(3) ! set
Bitmap Mode
130 CALL LINK("LOAD","DSK1.

```

```

PIC") !load pic-do not add _
P
140 REM
150 OPEN #1:"RD.DATA" ! RD
is optional device
160 FOR ROW=1 TO 20
! name for Myarc ramdisk
170 FOR COL=1 TO 160 ! appr
ox size of overscan
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 MYAR
TIST is available from the d
isk library.

```

The data is saved by the above program in a D/V 80 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 10 pixel co-ordinates per program DATA line ...

```

100 CALL GRAPHICS(3) ! MYARC
XB ONLY
110 CALL INIT :: CALL LOAD("
DSK2.MYARTIST")
120 CALL LINK("LOAD", "DSK1.P
IC")
130 OPEN #1:"RD.DATA",DISPLA
Y,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$(R
OW)&" "&STR$(COL)&" "&STR$(V
AR)&" ")
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 T
0 PRINT GRAPHIC EVERY WHICH
WAY ALL AT ONCE
2 ! THIS LISTING FOR MYARC X
B, BUT ANY PIXEL ADDRESSABLE
LANGUAGE CAN BE USED
3 !
4 ! DATA FILE TO BE MERGED I
NTO PROGRAM
5 !
100 CALL GRAPHICS(3)
110 READ A,B

```

```

120 CALL POINT(1,A,B) :: CAL
L POINT(1,180-A,240-B) :: CA
LL 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 REFER
S TO END OF DATA MARKER INSE
RTED AS LAST 2 ITEMS
500 GOTO 500

```

=====

### 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:

$$X^2 - 4 = 0$$

is true if X equals either 2 or -2, that is, two solutions.

If we make the x,y location 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 color 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 colors 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 "colors". 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 gray. 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 POSITIO
N
2 ! Paul Gailiunas in Fracta
l Report 17
3 ! for ti99/4a with ex bas
and The Missing Link by Step
hen Shaw Sept 1991
4 ! easily amended for any p
ixel addressable language
5 ! note particularly how th
e effect of five colours is
given on a monoscreen/printe
r
6 ! this effect can be used
on any other graphic program
needing up to 5 colours
7 ! including white
8 ! formula is in lines 190
and 200 . Experiment with ot
her polynomials or otherform
ulae.
9 ! move centre by setting X
0 and Y0. Range is half the
value of a side. Reduce rang
e for magnification- well wo
rth while!
10 ! takes a long time to pl
ot!
11 !
80 RANDOMIZE
90 ON WARNING NEXT! AVOID DI
VIDE BY ZERO ERRORS
100 R1=-1 :: R2=-.5 :: R3=.5
:: R4=1
110 X0=2*RND-2*RND :: Y0=2*R
ND-2*RND
120 CALL LINK("PRINT",1,1,ST
R$(X0)&":"&STR$(Y0))
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=X0+(X-120)*RANGE/120!
scaling
170 YI=Y0+(95-Y)*RANGE/95
180 FOR I=1 TO 20! give up a
fter 20 as unlikely to conve
rge
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)
210 Z=(XI*FY-YI*FX)/(FY-FX)
220 XI=YI :: YI=Z
230 IF ABS(XI-YI)<0.00001 TH

```

```

EN 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 DO
T(X+1,Y) :: CALL DOT(X,Y+1)
:: CALL DOT(X+1,Y+1) :: RETU
RN
330 CALL DOT(X,Y) :: CALL DO
T(X+1,Y+1) :: RETURN
340 CALL DOT(X,Y) :: RETURN
350 CALL DOT(X,Y) :: CALL DO
T(X+1,Y) :: CALL DOT(X,Y+1)
:: RETURN
360 END
370 SUB DOT(X,Y)
380 CALL LINK("PIXEL",Y,X)
390 SUBEND

```

=====

**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 is different- it 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 symmetrical about horizontal and vertical axes at point 0,0. 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 "colors" on a mono screen or printer dump, with five textures, here representing 16 colors.

MORE.....

```

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 REN
DITION OF PAUL GAILIUNAS
15 !
16 ! FOR EX BAS + THE MISSIN
G LINK
17 ! BUT CAN USE ANY PIXEL A
DDRESSABLE LANGUAGE
18 !
100 CALL LINK("CLEAR")
101 ! line 110 sets upper,lo
wer,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 A
BS(X)<50 OR ABS(Y)<50 THEN C
ALL DOT(NX,NY,K):: K=16 :: G
OTO 230 ELSE K=16
230 NEXT K :: NEXT NY :: NEX
T NX
239 CALL LINK("SAVEP", "RD.PI
C")
240 GOTO 240
250 SUB DOT(X,Y,K)
260 ON K GOTO 270,280,290,30
0,270,280,290,300,270,280,29
0,300,270,280,290,300,270,28
0
270 CALL LINK("PIXEL",X,Y)::
SUBEXIT
280 CALL LINK("PIXEL",X,Y)::
CALL LINK("PIXEL",X+1,Y+1):
: SUBEXIT ! CONTINUED

```



```

290 CALL LINK("PIXEL",X,Y)::
  CALL LINK("PIXEL",X+1,Y)::
CALL LINK("PIXEL",X+1,Y+1)::
  SUBEXIT
300 CALL LINK("PIXEL",X,Y)::
  CALL LINK("PIXEL",X+1,Y)::
CALL LINK("PIXEL",X,Y+1):: C
ALL LINK("PIXEL",X+1,Y+1)
310 SUBEND
320 ! fractal report costs t
en uk pounds in uk for 6 iss
ues from
330 ! reeves telecom labs lt
d
340 ! west towan house Port
htowan 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!

=====

Items submitted for TI\*MES 41:

ARTICLE by Stephen Shaw

Hi everyone. Not much written by me this issue due to family illness, but a brief bit for you... .

.  
We will start with another graphics program, this time a RECURSIVE one. Our versions of Basic do not support recursion, but as long as there is enough memory we can stack up our subroutines, which is how this program operates..

.  
Every time you GOSUB, the computer remembers where you left from UNTIL it returns, so lots of GOSUBS without any returns can end up with a MEMORY FULL error, but with a small program like this, we can stack up the GOSUBS far enough to produce a perfectly adequate recursive program:.

**1 ! KOCH CURVE .**

2 ! recursive pattern .

3 ! written for TI ExBas plus The Missing Link .

4 ! by stephen shaw march 1993 .

5 ! from "fractals for the classroom" by peitgen,jurgens, and saupe.. Springer-Verlag..

6 ! .

7 ! after Helge von Koch, Arkiv for Matematik, 1904 .

8 ! .

100 CALL LINK("CLEAR"):: RANDOMIZE .

110 ! CALL LINK("PRINT",41,12,"Input peak offset.  
(suggest 0.29)"):: CALL LINK("INPUT",71,81,R).

120 R=RND\*.52 ! 0<R<1 -rem out if using line 110. .

130 CALL LINK("CLEAR"):: R\$="R"&STR\$(R):: CALL LINK("P.  
RINT",140,100,R\$) . ! MORE....

```

140 L=5 :: XL(L)=10 :: XR(L)=220+10 :: YL(L),YR(L)=120 .
150 GOSUB 190 .
160 CALL LINK("PRINT",170,20,"DONE-press a key") .
170 CALL KEY(5,A,B):: IF B<1 THEN 170 .
180 END .
190 IF L>1 THEN 220 .
200 CALL LINK("LINE",YL(1),XL(1),YR(1),XR(1)) .
210 GOTO 390 .
220 L=L-1 .
230 XL(L)=XL(L+1):: YL(L)=YL(L+1) .
240 XR(L)=.333*XR(L+1)+.667*XL(L+1) .
250 YR(L)=.333*YR(L+1)+.667*YL(L+1) .
260 GOSUB 190 .
270 XL(L)=XR(L):: YL(L)=YR(L) .
280 XR(L)=.5*XR(L+1)+.5*XL(L+1)-R*(YL(L+1)-YR(L+1)) .
290 YR(L)=.5*YR(L+1)+.5*YL(L+1)+R*(XL(L+1)-XR(L+1)) .
300 GOSUB 190 .
310 XL(L)=XR(L):: YL(L)=YR(L):: XR(L)=.667*XR(L+1)+.333*XL(L+1) .
320 YR(L)=.667*YR(L+1)+.333*YL(L+1) .
330 GOSUB 190 .
340 REM RIGHT BRANCH .
350 XL(L)=XR(L):: YL(L)=YR(L):: XR(L)=XR(L+1):: YR(L)=YR(L+1) .
360 GOSUB 190 .
370 L=L+1
380 RETURN
390 RETURN
400 RETURN ! do we need all these? check and delete if required!.

```

=====

-----.

#### PROGRAMMING BASIC....

.  
Way back in TI\*MES Issue 34, Autumn 1991, I gave a program that would add fractions, but not reduce the result (Listing One, below), and asked if you could better it..

.  
I also, in the same issue, gave a more complex program that would reduce the result (Listing Number Two, below)..

.  
These entries created the usual lack of response from the UK, but have stirred up quite a response from our overseas friends..

.  
In MICROpendium March 1993 issue, Dean S Mah of Red Deer, Alberta gave some modifications to listing one- see listing three below, to be read in conjunction with list one..

.  
Meanwhile one of our leading programmers has been attacking the problem from several angles- see listings four onwards..

.  
Consider the problem. eg to add 11/13 to 17-32 etc and display the result as a fraction (some fractions cannot be displayed too well as a decimal!)..

.  
Try to solve the problem..

.  
Look at the various attempts below. Which program is the easiest to follow? Which program uses the least memory? And which program gives the fastest result? Which of these three options would you prefer?.

.

Now for the listings, in 80 column format to reduce space....

-----  
LISTING ONE- THE ORIGINAL:

```
.
1 ! ADDING FRACTIONS
2 ! result is not reduced      eg 12/16 would usually
be shown as 3/4 .
100 CALL CLEAR .
110 DISPLAY AT(10,5):"---- + ---- = ----" .
120 ACCEPT AT(9,5)SIZE(3)VALIDATE(DIGIT):A .
130 ACCEPT AT(11,5)VALIDATE(DIGIT)SIZE(3):B .
140 ACCEPT AT(9,11)SIZE(3)VALIDATE(DIGIT):C .
150 ACCEPT AT(11,11)SIZE(3)VALIDATE(DIGIT):D .
160 GOSUB 230 .
161 ! PRINTER CHECK: WHEN YOU SEE #### USE SHIFT 3.
170 DISPLAY AT(9,16):USING "####":N .
180 DISPLAY AT(11,16):USING "####":L .
190 DISPLAY AT(14,1):"ENTER KEY FOR ANOTHER" .
200 DISPLAY AT(1,1):"NORMAL RESULT=";A/B+C/D .
210 ACCEPT AT(24,12):A$ :: GOTO 100 .
220 STOP .
230 FOR X=2 TO B*D .
240 IF INT(X/B)<X/B THEN 260 .
250 IF INT(X/D)=X/D THEN 270 .
260 NEXT X .
270 L=X .
280 N=INT(L/B)*A+INT(L/D)*C .
290 RETURN .
```

-----  
LISTING TWO- reduces answer... .

```
100 ! FRACTIONAL + AND - .
110 ! R CALDWELL JAN 91 .
120 ! FOR TI99/4A BY .
130 ! S SHAW APR 91 .
140 DIM Q(102) .
150 DISPLAY AT(1,1)ERASE ALL:"FRACTIONAL + & -" .
160 DISPLAY AT(7,1):"----+----" .
170 DE,X=1 .
180 ACCEPT AT(6,2)VALIDATE(DIGIT,"+-")SIZE(4):N(X) .
190 IF N(X)=0 THEN 310 .
200 ACCEPT AT(8,2)VALIDATE(DIGIT)SIZE(4):D(X) .
210 IF D(X)=0 THEN 310 .
220 DP(X)=N(X):: DE=DE*D(X) .
230 CALL HCHAR(6,1,32,12) .
240 CALL HCHAR(8,1,32,12) .
250 DISPLAY AT(5,1):"PLUS:" .
260 DISPLAY AT(3,1):"ENTER 0 TO TOTAL" .
270 X=X+1 :: DISPLAY AT(7,15):"ITEM";X :: DISPLAY AT(8,15):"-MAX 10-" .
280 IF X=11 THEN 310 .
290 GOTO 180 .
300 REM .
310 F,X=X-1 .
320 FOR S=1 TO F :: FOR J=1 TO F .
330 Z=P+1 :: K=S+J-1 .
340 P(S,J)=K-F*INT(K/F):: IF P(S,J)=0 THEN P(S,J)=F .
350 Q(Z)=P(S,J):: NEXT J :: NEXT S .
360 FOR X=1 TO F :: Y=F*X-F+1 :: FOR C=1 TO F-1 .
370 Y=Y+1 :: DP(X)=DP(X)*D(Q(Y)):: NEXT C .
```

```

380 NU=NU+DP(X):: NEXT X :: DD=2 .
390 DISPLAY AT(10,1):"SUM IS:" :: DISPLAY AT(11,5):NU;"/";DE .
400 IF NU/DD=INT(NU/DD)AND DE/DD=INT(DE/DD)THEN NU=NU/DD ::
DE=DE/DD :: GOTO 400 .
410 DD=DD+2+(DD=2) .
420 IF ABS(NU)>DE THEN A=ABS(NU)ELSE A=DE .
430 IF DD<=SQR(A)THEN 400 .
440 IF NU/DE=INT(NU/DE)THEN 540 .
450 IF NU>0 THEN WN=INT(NU/DE):: NU=NU-WN*DE :: PN=1 .
460 IF NU<0 THEN WN=INT(NU/DE)+1 :: NU=ABS(NU-WN*DE):: PN=2 ::
IF WN=0 THEN NU=-NU .
470 REM CHAR ##### IS SHIFT 3 .
480 DISPLAY AT(13,5):USING "#####":NU .
490 IF PN=1 THEN DISPLAY AT(14,1):WN :: NU=NU+WN*DE .
500 IF PN=2 THEN DISPLAY AT(14,1):WN :: IF WN<0 THEN NU=WN*DE-NU .
510 PN=0 .
520 DISPLAY AT(14,5):"-----" .
530 DISPLAY AT(15,5):USING "#####":DE .
540 DISPLAY AT(18,4):NU/DE .
550 DISPLAY AT(24,4):"ANY KEY FOR ANOTHER" .
560 CALL KEY(1,G,H):: IF H<1 THEN 560 .
570 RUN .
580 END .

```

===== .  
LISTING THREE- modifications to LIST ONE,.  
from Micropendium March 1993:.

```

165 CALL REDUCE(N,L).
300 SUB REDUCE(A,B).
310 X=A :: Y=B :: IF X<Y THEN TEMP=X :: X=Y :: Y=TEMP.
320 CALL MOD(X,Y,REMAIN).
330 IF REMAIN>0 THEN X=Y :: Y=REMAIN :: GOTO 320.
340 A=A/Y :: B=B/Y :: SUBEND.
350 SUB MOD(A,B,REMAIN).
360 REMAIN=A-INT(A/B)*B :: SUBEND .
370 END.

```

----- .  
LIST FOUR- WITH MACHINE CODE LINK (machine code source at end): .

```

90 CALL INIT :: CALL LOAD("DSK1.FRED/O") .
100 CALL CLEAR .
110 DISPLAY AT(10,5):"--- + --- = ---" .
120 ACCEPT AT(9,5)SIZE(3)VALIDATE(DIGIT):A .
130 ACCEPT AT(11,5)SIZE(3)VALIDATE(DIGIT):B .
140 ACCEPT AT(9,11)SIZE(3)VALIDATE(DIGIT):C .
150 ACCEPT AT(11,11)SIZE(3)VALIDATE(DIGIT):D .
160 GOSUB 230 :: CALL LINK("FRED",N,L) .
170 DISPLAY AT(9,16):USING "####":N ! # IS SHIFT 3 .
180 DISPLAY AT(11,16):USING "####":L .
190 DISPLAY AT(14,1):"ENTER KEY FOR ANOTHER" .
200 DISPLAY AT(1,1):"NORMAL RESULT=";A/B+C/D .
210 ACCEPT AT(24,12):A$ :: GOTO 100 .
220 STOP .
230 FOR X=2 TO B*D .
240 IF INT(X/B)<X/B THEN 260 .
250 IF INT(X/D)=X/D THEN 270 .
260 NEXT X .
270 L=X .
280 N=INT(L/B)*A+INT(L/D)*C .

```

```

290 RETURN .
.
.
----machine code source:.
.
* FRED/S .
* TO REDUCE FRACTIONS .
* CODE BY B. HARRISON .
* PUBLIC DOMAIN .
* .
DEF      FRED .
NUMASG EQU >2008 .
NUMREF EQU >200C .
XMLLNK EQU >2018 .
CFI     EQU >12B8 .
CIF     EQU >20 .
        CLR R0 .
        LI  R1,1 .
        BLWP @NUMREF .
        BLWP @XMLLNK .
        DATA CFI .
        MOV @>834A,@NUMER .
        MOV @>834A,@TNUM .
        INC R1 .
        BLWP @NUMREF .
        BLWP @XMLLNK .
        DATA CFI .
        MOV @>834A,@DENOM .
        MOV @>834A,@TDEN .
        C   @DENOM,@NUMER .
        JLT IMPROP .
        MOV @NUMER,@LIMIT .
        JMP PROPER .
IMPROP MOV @DENOM,@LIMIT .
PROPER .
        INCT R0 .
RPT     C   R0,@LIMIT .
        JGT ANSWER .
TRIAL  MOV @NUMER,R3 .
        CLR R2 .
        MOV @DENOM,R5 .
        CLR R4 .
        DIV R0,R2 .
        MOV R3,R3 .
        JNE NXT .
        DIV R0,R4 .
        MOV R5,R5 .
        JNE NXT .
        MOV R2,@TNUM .
        MOV R4,@TDEN .
NXT     INC R0 .
        JMP RPT .
ANSWER .
        CLR R0 .
        LI  R1,1 .
        MOV @TNUM,@>834A .
        BLWP @XMLLNK .
        DATA CFI .

```

```

        BLWP @NUMASG .
        INC R1 .
        MOV @TDEN,@>834A .
        BLWP @XMLLNK .
        DATA CIF .
        BLWP @NUMASG .
        LWPI >83E0 .
        B @>6A .
WS      BSS 32 .
NUMER  DATA 0 .
DENOM  DATA 0 .
TNUM   DATA 0 .
TDEN   DATA 0 .
LIMIT  DATA 0 .
END .

```

=====.

```

LIST FIVE- moving back to pure XB-.
amend the basic portion of list 4 as below: .
1 ! ADD FRACTIONS WITH REDUCTION .
2 ! LINE 290 REMOVED & NEW LINES 300-360 ADDED BY B. HARRISON .
3 ! ORIGINAL PROGRAM BY S. SHAW .
290 REM .
300 LIM=MIN(N,L):: TN=N :: TL=L .
310 FOR Y=2 TO LIM .
320 IF N/Y<>INT(N/Y)THEN 350 .
330 IF L/Y<>INT(L/Y)THEN 350 .
340 TN=N/Y :: TL=L/Y .
350 NEXT Y .
360 N=TN :: L=TL :: RETURN .
=====

```

LISTING SIX- modifications to listing 4: .

```

.
90 ON WARNING NEXT .
290 REM
300 TN=N :: TL=L .
310 FOR Y=2 TO MIN(N,L) .
320 IF N/Y>INT(N/Y)THEN 350 .
330 IF L/Y=INT(L/Y)THEN TN=N/Y :: TL=L/Y .
340 NEXT Y :: N=TN :: L=TL
350 RETURN .

```

=====.

```

LISTING SEVEN- complete listing: .
2 ! MODIFIED FOR REDUCTION AND SPEED ENHANCEMENT BY BRUCE HARRISON .
4 ! VERSION ADFR6 OF 1 APRIL 93 .
90 ON WARNING NEXT .
100 DISPLAY AT(10,5)ERASE ALL:"--- + --- = ---" .
110 ACCEPT AT(9,5)SIZE(3)VALIDATE(DIGIT):A .
120 ACCEPT AT(11,5)SIZE(3)VALIDATE(DIGIT):B :: IF B=0 THEN 120 .
130 ACCEPT AT(9,11)SIZE(3)VALIDATE(DIGIT):C .
140 ACCEPT AT(11,11)SIZE(3)VALIDATE(DIGIT):D :: IF D=0 THEN 140 .
150 FOR L=MAX(B,D)TO B*D STEP MAX(B,D) .
160 IF INT(L/B)<L/B THEN 180 .
170 IF INT(L/D)=L/D THEN 190 .
180 NEXT L .
190 N=L/B*A+L/D*C .
200 FOR Y=2 TO MIN(N,L) .

```

```

210 IF N/Y>INT(N/Y)THEN 230 .
220 IF L/Y=INT(L/Y)THEN N=N/Y :: L=L/Y :: GOTO 200 .
230 NEXT Y .
240 DISPLAY AT(9,16):USING "####":N .
250 DISPLAY AT(11,16):USING "####":L .
260 DISPLAY AT(14,3):"PRESS ENTER FOR ANOTHER" .
270 DISPLAY AT(1,1):"NORMAL RESULT=";A/B+C/D .
280 CALL KEY(0,K,S):: IF S<>1 THEN 280 ELSE IF K=13 THEN 100 .
=====

```

There you have it. There are almost always MANY ways to tackle ANY programming problem. Some will be better than others because they are easier to follow OR use up less memory OR work faster. The programmer must decide his priorities and program accordingly..

So called programmers using macroassemblers with vast routine libraries already written tend to end up with programs which are hard to follow, hard to modify, are slow, and use up lots of memory, which is why PC users now tend to go for machines with 4MB of memory running at 32Mhz or more! You don't need all that if you know how to program. .

And you can avoid using up 29k just to do a PRINT "HI"! .

~~~~~end of file~~~~~

Submitted for TI*MES 42:

I have previously quoted the puzzles from New Scientist magazine, many of which are very suitable for computer solution. This one is Enigma 725, set by Robert High in New Scientist of 3rd July 1993..

In the following diagram each letter stands for a digit. Each digit is represented only by one letter..

Example first with sample values:.

```

  T=1   E=3   A=2.
  A=2   T=1.

```

In the second row, the values are derived by taking the differences between the two letters above, ignoring any minus sign, so that taking the first two letters T and E, the difference is 2 which is A. These values apply only to this example!!!

Now solve this:.

```

  P  E  R  F  E  C  T.
  P  U  Z  Z  L  E.

```

I can tell you that T does not equal 1, and that there is only ONE solution. A

quick Extended Basic program I threw together here found the answer in 55 seconds and proved there was no other solution in 75 seconds. Can you beat that?

=====

```
1 REM WITHOUT RUNNING THIS
2 REM PROGRAM- WHAT DOES

3 REM IT DO?
4 REM CAN YOU DO IT MORE
5 REM EFFICIENTLY?

6 REM
7 REM
8 REM -A DEMONSTRATION TO
9 REM SHOW THAT YOU CAN DO
10 REM THINGS THE HARD WAY
11 REM OR
12 REM THE EASY WAY.
13 REM
14 REM X AND Y TO BE >0
15 REM
100 INPUT "FIRST NUMBER:":X
101 X=ABS(X)
110 INPUT "SECOND NUMBER:":Y
111 Y=ABS(Y)
120 Z=0
130 IF X/2=INT(X/2)THEN 150
140 Z=Z+Y
150 X=INT(X/2)
160 Y=2*Y
161 PRINT X;
170 IF X<>0 THEN 130
180 PRINT Z
190 END
```

=====.

=====

MORE FOLLOWS....

COMPILER....

.
I have right now a true compiler which is capable of producing code for the following extended basic program:.

```
100 B=48.3
110 A$="THIS WAS A VERY LONG TEST"
120 S=1 :: R=20 :: T=4
130 FOR I=-1 TO -R STEP -S :: FOR J=2 TO 30 STEP T

140 GOSUB 190
150 NEXT J :: NEXT I
160 FOR K=10 TO 1000 STEP 20 :: PRINT K :: NEXT K
170 PRINT B;A$
180 DISPLAY AT(12,7)BEEP:"TEST COMPLETE" :: STOP
190 PRINT I;J :: RETURN
```

.
This program demonstrates the ONLY parts of the new compiler which at this moment works, further extensions are to come..

.
The only extended basic function which is ruled out at present is the use of user sub programs (eg CALL MYPROG)..

.
PRINT uses a lot of processor time and the compiled program runs about twice as fast - if PRINT is removed, the compiled program runs about six times faster..

.
When I say "true compiler" I mean a program which takes an ordinary extended basic program as input (in merge format) and outputs assembly source code for an assembler to work upon. By additional processes, the output is made into a "hidden code" extended basic program- which could load from cassette if the 32k ram was present..

.
Here are the steps:.

1. Write your XB program and SAVE it to DISK in MERGE (DV163) format..
2. Load and run the compiler, which will ask you for the name of your saved XB program and ask for an output file name..
3. The compiler will create disk files: a. Assembly source code for assembling; b. Assembly data file which will be accessed while assembling. c. A merge format extended basic program to have the object code inserted into it..
4. Assemble the source code using Art Greens assembler. This takes as input the source file, accesses the data file, and various supplied "library" files, and outputs machine code object code..
5. Load Harry Wilhelms extended basic program HML (supplied) which takes as input the object file. Merge in the shell file and you are left with what looks like an odd extended basic program, which runs quite quickly..

.
CONTINUED....

At the time of writing supported features are: Variable equates (S=1);
FOR-NEXT-STEP (slightly limited at present); PRINT (numeric variables only at
present); GOSUB; GOTO; END; STOP. A limited form of IF...THEN, which like basic
can at present only jump to a line number, and no ELSE right now. .

.
At present there is no limit to the number of FOR-NEXT loops you can stack. .

.
Perhaps a little limited right now but a good start..

.
The compiler is being put together by Bruce Harrison. Watch for more news..
.

.....
.....