TI HARDWARE AND COMPONENTS     M.G.C.S.

| CABLES | £ p |
|---|---|
| DISK DRIVE CABLE EXT X 2 | 08.00 |
| DISK DRIVE CABLE EXT X 2 LONG | 09.00 |
| DISK DRIVE CABLE INT S/H | 02.50 |
| DISK DRIVE CABLE INT X 2 | 08.00 |
| DISK DRIVE CABLE EXT X 2 | 08.00 |
| DISK DRIVE POWER SPLITTER | 05.50 |
| PARALLEL PRINTER CABLES | 08.50 |
| COMPONENTS | |
| DISK CONTROLLER CONNECTORS 34 WAY | 01.25 |
| DISK DRIVE CONNECTORS 34 WAY IDC | 01.10 |
| DISK DRIVE POWER CONNECTORS IN or EX | 01.00 |
| 9 PIN "D" CONNECTOR & COVER | 01.00 |
| PARALLEL PRINTER CONNECTORS | 03.00 |
| 16 PIN PRINTER CONNECTORS | 01.00 |
| P.E.B PUSH BUTTON ON/OFF SWITCH HEAVY DUTY REPLACEMENT | 01.75 |
| PROGRAM HEADERS 8 WAY (FOR SETTING DISK DRIVES) | 00.50 |
| SMALL BLACK PLASTIC BOXES 72x45x22 mm | 01.00 |
| 16 way RIBBON CABLE IDC PER FOOT | 00.50 |
| IN4148 TYPE DIODES FOR JOYSTICKS ETC | 00.05 |
| DISKS AND ANCILLIARIES | |
| 50 CAP DISK BOX WITH LOCK | 07.50 |
| 100 CAP DISK BOX | 10.50 |
| 140 CAP DISK BOX | 12.50 |

5.25 DSDD DISKS 96tpi 100% ERROR FREE
LIFETIME GUARANTEE

| 25+ | 50+ | 75+ | 100+ |
|---|---|---|---|
| 36p | 34p | 32p | 30p |

| HARDWARE | NON MEMBERS | MEMBERS |
|---|---|---|
| | £ p | £ p |
| JOYSTICK INTERFACE TI to 2 ATARI | 09.50 | 07.00 |
| CONSOLES RECONDITIONED | 25.00 | 20.00 |
| DISK DRIVES DS SD VARIOUS FROM | 45.00 | 30.00 |
| POWER SUPPLIES | 10.00 | 07.50 |
| MODULATORS | 10.00 | 07.50 |
| JOYSTICKS TI | 10.00 | 07.50 |
| JOYSTICKS QUICKSHOT I (ATARI TYPE) | 06.50 | 05.00 |
| CASSETTE LEADS | 07.50 | 05.00 |
| PAR PRINTER LEAD | 12.00 | 08.50 |
| MINIMEMORY BATTERIES | 01.25 | 01.00 |

CARRIAGE leads etc £00.50.
disk boxes etc £02.00
Cheques etc Made out to M.Goddard
Overseas cheque or money order drawn on London
Overseas carriage at cost call for details

MIKE GODDARD COMPUTER SUPPORT     TEL 0978 843547
"SARNIA" CEMETERY RD. RHOS. WREXHAM. CLWYD.

# TI*MES

TI99/4A USERS GROUP (U.K.) CONTACTS LIST:

Chairman: Gordon Pitt: (Tel 0922 476373)
259 Sneyd Lane, Bloxwich, WALSALL, W Midlands, WS3 7LS
General Secretary: Jim Ballinger:
5 Offerton Avenue, DERBY, DE3 8DU. Tel 0332 772612
Membership Secretary & Telecoms: Peter Walker:
24 Bacons Drive, CUFFLEY, Herts. EN6 4DU. Tel 0707 873778
Treasurer: Alan Rutherford. Tel 0625 524642
13 The Circuit, WILMSLOW, Cheshire, SK9 6DA
Publicity Officer: Christine Bennett. Tel 061 430 7298
20 Oak Avenue, Romiley, STOCKPORT, Ches (Bus 061 430 7645)
TI*MES Editor: Alan Bailey Tel 01 509 1053
14 Shelley Grove, LOUGHTON, Essex, IG10 1BY
Librarians:
Cassette: Tim Anderson (Tel 041 337 1291)
Ground Fllor (Left), 47 Apsley Street, GLASGOW, G11 7SN
Disk: Stephen Shaw (also Journal Exchange & VP)
10 Alstone Road, STOCKPORT, Cheshire, SK4 5AH
Hardware (& Projects): Mike Goddard (Tel 0978 843547)
'Sarnia',Cemetary Road, Rhos, WREXHAM, Clwyd, LL14 2BY
Modules: Edward Shaw: (Tel 0538 360382) 5.00 to 8.00 pm.
Crow Holt Farm, Basford, LEEK, Staffs, ST13 7DU
Publications: Mike Curtis (Tel 0209 219051 )
21 Treliske Rd, Roseland Gardens, REDRUTH, Cornwall, TR15 1QE.

CONTENTS

(contd over)

ISSUE NO. 27     WINTER 1989/90

## EDITORIAL

Here is another fully packed edition of your magazine. In the continuous search for improved service to you the members of our group, we need to make greater resources available for other purposes by reducing the cost of the magazine in your hands.
We are trying a new printer, and doing our best to avoid copying costs. Please help by sending your copy for the magazine in carefully prepared good black type, starting 15mm from the top edge of A4 paper,extending from 15mm from the left hand edge to not closer than 15mm to the right hand edge, and down to not closer than 22mm to the bottom edge. Stephen Shaw has offered to help in case of difficulty, including typing for anyone without this facility.
Please note the next copy date is 1st.March 1990.

## DISCLAIMER

The opinions expressed by the contributors to this magazine are their own, and are not necessarily those of the group or its committee.

## BLOXWICH WORKSHOP

Gordon Pitt tells me that his next workshop is scheduled to take place on Saturday 31st.March 1990. Entrance £3. Please contact Gordon for details.

## BUMPER ISSUE

The next issue will be extra large to accomodate a flood of interesting material from Stephen Shaw.

## CONTENTS CONT.

## MINUTES OF TI99/4a USERS GROUP(UK) A.G.M.
## ROMILEY COMMUNITY CENTRE 17th.JUNE 1989

**In the Chair.** Clive Scally

**Apologies** were received from P.Walker,T.Anderson,P.Brooks and C.Mehew.

1.**Minutes of previous A.G.M.** were proposed by A.Bailey and approved nem.con.

2.**Matters arising** - The lateness of the start of the A.G.M. was criticised. It was suggested that in future a time for the formal proceedings should be decided and published with the details of the venue.

3.**Chairman's Address** - Clive explained that he wished to step down, as he could no longer devote as much time as he felt this position required. A copy of his reasons for resigning were published in the last edition of TI*MES. He thanked all the Committee for their hard work, and the membership for their support.

4.**Vice-Chairman's Address** - Stephen thanked everyone present for making the effort to attend, particularly those who had travelled from abroad especially for the A.G.M.
He then went on to explain the origins of the User Group, and Clive's part in it. Finally he proposed the Clive should be offered the post of Honorary Vice-President together with Life Membership. This was seconded by Mike Goddard, and was agreed unanimously.

Clive thanked members for the offer of this honour, which he was pleased to accept. He then moved that the Election of Officers be taken next.

5.**Election Officers** - The following were elected:-

Chairman...........................Gordon Pitt
At this point Clive vacated the Chair.
Vice-Chairman......................Stephen Shaw
Treasurer..........................Alan Rutherford.
General Secretary..................Jim Ballinger

| | |
|---|---|
| Membership Secretary,Telecommunications.. | Peter Walker |
| Editor TI*MES........................... | Alan Bailey |
| Hardware,D.I.Y.Projects................. | Mike Goddard |
| Cassette Librarian...................... | Tim Anderson |
| Modules Librarian....................... | Edward Shaw |
| Publications Librarian.................. | Michael Curtis |
| Disc Librarian,"Rambles",Journal Exchange. | Stephen Shaw |
| Publicity Officer....................... | Christine Bennett |
| Programming Languages................... | VACANT |

6.Committee Member's Reports

Stephen Shaw submitted a financial report showing a healthy balance after transferring £200 to the club reserve, and having carried much of the financial extras involved in the A.G.M. organisation.

He pointed out that Journal Exhanges were still on the decline, though still providing useful contacts.

Alan Rutherford distributed copies of his Financial Report, showing the position (as at June 1st) of a balance of £1926.46 available, after paying expenses including the cost of TI*MES production and circulation (£2276.94). It should be noted that effect of neither the full effect of the increased subscription, nor the possible decrease in membership can yet known accurately. He points out that his figures had not been audited, as no Auditor has been appointed by the club.(Peter had offered to take on the position pending such an appointment, but this will be a matter for the Postal Ballot now I think [Sec]).

Jim Ballinger said that whilst the year had been very satisfactory on the whole, he had once again to report that the response to his circulars, particularly where ballots were asked for, was still far from adequate. He once again asked all Committee members to please record their vote and opinions on each issue.

Peter Walker's report was read to the meeting. He reported that the membership level had fluctuated, it started at 209, rising to 224 by the end of the first year. Then 48 members did not renew, level down to 176. This rose to 198 as a result of adverts in magazines and publicity gained from the Alternative Micro Show.

Writing on 12.6.89, he wrote that of the 137 members due to renew, only 75 had done so at that date, so there was the possibility that a further shrinkage may need to be faced. He was circulating members yet to renew. He was also worried that only about 50% of enquiries resulted in a susequent membership form being returned.

Of current renewals(75)37 made no comment about the group, which one assumes is good news, while 30 specifically praised the group as currently run. Comments noted :-

1. Poor print quality ( Very much improved see last issue-Sec)
2. Print too small
3. Can we see Group finances (detail print out available now-Sec)
4. Want libraries at Micro Shows/AGM
5. More D.I.Y.
6. More on MiniMemory.
7. Summary of available hardware e.g. RAMDISKS
8. Better arrangements for Expansion System sales.
9. Pay subs by Standing Order or Direct Debit.
10. Graphics on front cover (including Group Logo)
11. Pros/Cons of Standalone 32K for unexpanded owner.
12. How to fit 8k RAM into EA or TIW for Funnelweb.
13. DIY tips for non-disc owners.
14. Where can equipment be repaired?

(Mike is going to be busy!-Sec)

Peter repeated his request for someone to help provide material to interest the console only owner in TI*MES

Alan Bailey reported that, now that he had retired he was unable to make use of facilities formerly available to him, and this may result in a small increase in the price of printing TI*MES.

Mike Goddard reported that it had been quite a good year, particularly for repairs and sales. He reminded the meeting that he was able to offer members both repair and hardware supply services.

Tim Anderson wrote to report that he had drawn up a questionaire with the intention of raising interest in the library, which had not been used a great deal. He thinks that occasional delays in the Glasgow sorting office cannot help. He promises to let Committee members know what the response to his initiative is.

After purchasing C60 tapes for compilations and stamps etc, he has a balance of £1.41.

Edward Shaw reported that events had shown that members prefer to buy rather than borrow modules, producing a nett profit of £146.95.

Christine Bennett reported that she had written to several magazines and had letters-ads. published definitely in Micro-Mart and Computer Shopper, but only Peter could determine what response, if any, these had generated.

She would continue to write to the magazines in the hope that some interest will continue to ensue.

7.Matters arising.

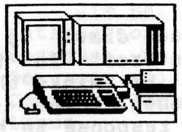
Geoffrey Coan resigned from the Committee. Since leaving the university (with an honours degree in Computer Science and Methematics) it has been apparent for some time that pressure of work was preventing him from doing as much with the TI as he would wish. Pleased to read that he remains a member and we may yet see more of his work in TI*MES. He was always a very pleasant and courteous person to deal with, and will be sadly missed at our Committee meetings.

Gordon Pitt pointed out that a Computer Workshop was to be held at Bloxwich on July 2cnd, and outlined the Communications availability.

The Chairman of the Dutch T.I.Group (Mr.Harmsen) advised all present of the Group Fair to be held at Neimegen, and invited those interested to attend.

The Meeting closed with a vote of thanks to the Chair.

Jim Billinge



# MEMBERSHIP NEWS

from Peter Walker
Membership Secretary

Since the last issue of TI*MES we welcome to TIUG(UK): M. Elshelli, Jan Moller, Michael Crowther, Barrie Clark, Fiona Kempson, Harry Frost, Andrew Harding, Colin Astbury, Alfredo Cevolini, J Clarke and Alan Foggett.

Here are some further lists of members who agreed to their names being published. Why not get together and form a local group?

## SOUTH-EAST MIDLANDS

| | |
|---|---|
| MARTIN L ARCHER | 3 TELFORD WAY, BLAKELANDS, MILTON KEYNES, MK14 5LB |
| PETER G Q BROOKS | 96 BANBURY ROAD, OXFORD, OX2 6JT |
| SIMON DORRICOTT | 27 LONDON ROAD, BEDFORD, MK42 0PB |
| DAVID G HEWITT | 53 KENNET CLOSE, BERINSFIELD, OXFORDSHIRE, OX9 8QE |
| IAN D JAMES | 36 HARRISON CRESCENT, BEDWORTH, WARWICKSHIRE, CV12 8SL |
| STEVEN MARTIN | 6 BEDFORD COTTAGES, GREAT BRINGTON, NORTHAMPTON, NN7 4JE |
| NEIL WILSON | 46 BEDFORD ROAD, HOUGHTON CONQUEST, BEDFORD, MK45 3NA |

+ 3 others

## SOUTH-WEST MIDLANDS

| | |
|---|---|
| BARRIE CLARK | SHANTAY,53 CLEEVE DRIVE, CLEEVE, AVON, BS19 4NP |
| KENNETH F HUGHES | 220 BROADLAND DRIVE,LAWRENCE WESTON, BRISTOL, AVON, BS11 0PN |
| RICHARD A SIERAKOWSKI | RUSHOLME,ELCOT LANE, MARLBOROUGH, WILTSHIRE, SN8 2BA |

+ 4 others

## CHILTERNS

| | |
|---|---|
| STANLEY J PHILLIPS | 56 WINDMILL AVENUE, MARSHALSWICK, ST ALBANS, HERTFORDSHIRE, AL4 9TF |
| CHRISTOPHER ROWSON | 5 HILLCREST ROAD, SHENLEY, RADLETT, HERTFORDSHIRE, WD7 9DB |
| NICHOLAS P SUMMERS | 36 DRAYTON ROAD, TILEHURST, READING, BERKSHIRE, RG3 3PH |

+ 7 others

## NORTH THAMES

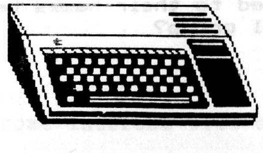
| | |
|---|---|
| DAVID R CHAMBERS | 134 FOREST LANE, FOREST GATE, LONDON, E7 9BB |
| DAVID T DEACON | 28 KETTON HOUSE,SUTTON WAY, NORTH KENSINGTON, LONDON, W10 5ET |
| M A A ELSHELLI | 83B JAMESTOWN ROAD, CAMDEN TOWN, LONDON, NW1 7DP |
| IAN P KILGOUR | 3 CARLTON COURT,65 STUDLEY GRANGE ROAD, HANWELL, LONDON, W7 2LU |
| KEITH RANSON | 154 LAMPITS, HODDESDON, HERTFORDSHIRE, EN11 8DX |
| PETER J WALKER | 24 BACONS DRIVE, CUFFLEY, POTTERS BAR, HERTFORDSHIRE, EN6 4DU |

+ 5 others

ODDS AND ENDS.................... Mike Goddard

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

While on holiday recently in Weymouth Dorset we visited the Shipwreck and Diving museum at the old harbour and imagine our surprise when we discovered a TI-99/4a working for its living demonstrating a coastguard rescue simulation program,since then I also have an undocumented report of another helping to run a Taxi business in Rhyl I wonder how many more there are out there still slaving away does anybody else know of any more ?.



# CONSOLE ONLY CORNER

BY PETER WALKER

In the last issue of TI*MES, I discussed conversion programs and promised to publish the best conversion program I received. In fact, only one was submitted by a member called S. Shaw - hmm familiar name that.... Anyway, Stephen's program is a disk based routine, so this is discussed elsewhere.

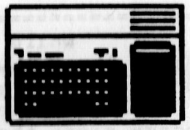
In the earliest days of computers, the clumsy valve based machines were often programmed to carry out lengthy mathematical tasks, such as calculating pi to hundreds of decimal places or finding prime numbers. Indeed, calculating prime numbers is a task which is still regularly used as a speed 'benchmark' between computers. As we all know, the TI99/4A Basic is far from being fast, so I was interested to see how long it would take on our machine to display all prime numbers under 1000.

Now prime numbers are those that have no factor other than the number itself and 1. These are 2,3,5,7,11,13,17.... The first program I tried is given below. For each number between 2 and 1000, we divide it by each number less than the original number's square root. If any division results in an exact integer then it is not a prime. The program, written for Extended Basic, takes 2 min 53 secs to run.

```
100 FOR N=2 TO 1000
110 FOR J=2 TO SQR(N)
120 IF INT(N/J)*J=N THEN 150
130 NEXT J
140 PRINT N;
150 NEXT N
```

In an attempt to speed up this program, we might decide that it is a waste of time to try every number as a potential divisor: it is only necessary to try prime numbers as divisors. So, as primes are found, we create an array to store them and only chose divisors from this list. We would get a program like this..

```
100 DIM P(1000) :: P(1)=2 :: L,J,N,M,Q=1
110 !@P-
120 FOR N=2 TO 1000 :: M=SQR(N)
130 FOR J=1 TO L :: Q=P(J) :: IF Q>M THEN 160
140 IF INT(N/Q)*Q=N THEN 170
150 NEXT J
160 PRINT N; :: P(L+1)=N :: L=L+1
170 NEXT N
```



All very well, only snag is that this program takes 3 min 3 secs to run - so much for speeding up the routine. Even removing the FOR-NEXT loop in line 130 (since it is always exited via the IF at the end of the line) actually further slows down the program. Do note though the calculation of the square root outside the FOR-NEXT loop in order to avoid repetitive calculation. What can we do to produce a faster program?

The answer is to go back to basics. Another traditional way of finding prime numbers is to write down all the numbers from 1 to 1000. You then cross out all the multiples of 2, ie 4,6,8,10.. Then cross out all the multiples of 3, ie 6,9,12,15.. 4 will already be crossed out so can be ignored and is also thus shown as not being a prime. When you have reached 1000, then the numbers not crossed out are the prime numbers. The program below achieves this. The elements of the array P() are set to -1 (TRUE) if "crossed out" and primes can be displayed as we progress.

```
100 DIM P(1000) :: J,K=0
110 !@P-
120 FOR J=2 TO 999 :: IF P(J) THEN 140 ELSE PRINT J;
130 FOR K=J+J TO 1000 STEP J :: P(K)=-1 :: NEXT K
140 NEXT J
```

The good news is that this program runs in just 37.6 seconds. Indeed it gives the impression of being speed limited not by the loop calculations but by the inherent delay in displaying numbers on the screen itself. The moral is clear: explore several methods of achieving your program's objectives and don't assume that elegance or obviousness results in greatest speed. Can anyone beat 37.6 seconds? You might like to try converting this routine into console Basic. This takes 52.8 seconds - can this be improved?

Peter Walker

## ODDS AND ENDS.................... Mike Goddard

I recently read a letter in Micropendium from a UK reader asking wether it was possible to run US stand alone type hardware in the UK because of the difference in mains voltage well the answer is fairly easy. All computer equipment runs internally on one or more of the following DC voltages +5 -5 +12 -12 provided by a power supply irrespective of wether the mains is 110 117 120 220 240 250 volts AC or 50 or 60 cycles per second so the simple answer is to find out what type of power supply the equipment uses and then find the UK equivalent and as most stand alone hardware uses the small plug into the mains type it is easy to obtain these from places like TANDY or similar electronic goods retailers. I think this answers the problem if not it is probably quicker to drop me a line than to use Micropendium as a communications media.

Mike

## MODULES MODULES MODULES MODULES

APPEAL TO ALL DISK DRIVE OWNERS........

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

---

The latest list of modules available for purchase follows! please remember that often there are only one or two modules of each title held and it helps if you can give alternatives when ordering. Terms are strictly first come, first served.

| | | | |
|---|---|---|---|
| ADVENTURE and PIRATE TAPE | 5.00 | TEACH EX BAS CASSETTE | 1.00 |
| DEFENDER | 4.00 | TERMINAL EMULATOR II | 5.00 |
| DONKEY KONG | 3.50 | TI LOGO II INCL MANUAL | 12.00 |
| EARLY READING | 2.50 | VIDEO CHESS | 8.00 |
| EDITOR ASSEMBLER (NO MANUAL) | 22.50 | BLASTO | 3.50 |
| HANGMAN | 2.00 | INDOOR SOCCER | 3.50 |
| TI INVADERS | 3.50 | TONBSTONE CITY | 4.00 |
| MUSIC MAKER | 5.00 | CONNECT 4 | 3.50 |
| MINIMEMORY (NO MANUAL) | 22.50 | | |
| PROTECTOR | 4.00 | | |
| PERSONAL REPORT GENERATOR | 3.00 | PERSONAL RECORD KEEPING | 3.00 |
| HOUSEHOLD BUDGET MANAGEMENT | 3.00 | MICROSURGEON | 4.00 |
| SHAMUS | 3.50 | PARSEC | 3.50 |
| VIDEO GAMES 1 | 3.50 | | |

PURCHASING MODULES FROM THE LIBRARY

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

---

Application to loan/purchase modules.

Name:.............................. Modules required:...........
Address:........................... ......................
................................... ......................
................................... ......................
I enclose cheque/PO for £......(as indicated on the list) & post to
Please make cheques payable to TIUG(UK).
Foreign orders can only be accepted if a BANKERS DRAFT is enclosed drawn in STERLING on a LONDON bank. It also helps if a little extra is added on for postage overseas.

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

---

CASSETTE REVIEWS....... NICKY GODDARD

===

All of the games reviewed here are available from the group cassette library at the current library terms.

---

STAR RATING GUIDE.
One star = terrible,Two stars = "OK",Three stars = quite good, Four stars = very good, Five stars = Brilliant..........

---

BLACKBEARDS TREASURE LIBRARY NO.G36

A few seconds after you've pressed play to load the game the computer says 'BLACKBEARDS TREASURE'(you don't need the speech synthesizer to say that) then starts to load.

Your five man diving team has found the lost treasure of Blackbeard, the pirate, but it is on the ocean bed in a shark infested sea and is guarded by two octopusses.

BEWARE though because the sharks havn't been fed since the last time someone played the game and they love to eat divers. Joystick 1 or E,S,D,X will move your diver off the boat and about the water. If your diver is moving down in the water and you press the left or right arrow keys(left or right on your joystick)it will decrease his downwards motion and increase his sideways motion. The diver will move slower on the sandy bottom of the sea than in the water. If you place the divers hand on the treasure he will pick it up. He will hold a maximum of 3 treasures at a time but each treasure will slow down his maximum ascent speed. To add the divers treasure to the score simply bring the diver up to the middle of the boat and he will automatically climb aboard.

For every ten treasures you bring up the level will increase by increasing the number of sharks and increasing to speed of the sharks and octopusses. A very good game for the joysick or the keyboard which I will play alot more in the future
STAR RATING *****

---

BLAST IT LIBRRAY NO.G38

The aim of this game is to defuse a number of bombs by visiting each in turn with a remote controlled robot.The robot cannot cross it's guide wires because if it does it ends the game. The bombs are defused from the top between the 'HORNS'.

They MUST be destroyed in numerical order. Use the arrow keys(E,S,D,X)to move the robot. The score for defusing the bomb is the bomb number mutiplied by the time left and then divided by ten. So the quicker you defuse the bomb the higher the score. If you haven't defused all the bombs when the time has run out the remaining bombs explode, then the computer gives you your score and tells you if your score is high enough to go into the hall of fame. You must score 400 points or over to get into the hall of fame. A good keyboard game a bit simple but enjoyable.
STAR RATING ***

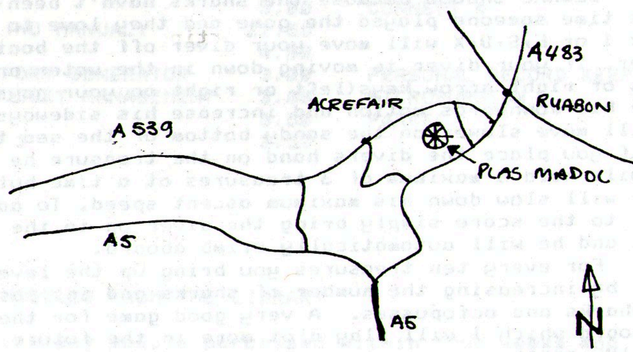
## WORKSHOP NEWS.................. mike Goddard

The next CHESTER/WREXHAM workshop will take place on Saturday 24th February at the PLAS MADOC LEISURE CENTRE in the AQUA LOUNGE this centre boasts an indoor heated pool, sport facilities and a full cafeteria it is at Acrefair on the A539 Llangollen road south of Wrexham the nearest train station is Ruabon situated on the Wrexham/Shrewsbury line and is about 5mins walk away. By bus take any Llangollen bus from either Wrexham or Chester.

We hope by then to have an operational Zeno board to demonstrate also an autodialler and mains control via the cassette motor port or if there is anything anybody would like to see or discuss specifically I'll do my best to arrange it.

Stephen Shaw has generously offered to demonstrate the latest software from the disk library and/or Tutorials on the more isoteric software packages such as Funlweb etc.

This meeting will be open to anybody with an interest in the TI-99/4A wether a member of the group or not and we hope as many of you as possible try to get along.



## ODDS AND ENDS.................... Mike Goddard

If any of you who read the article by Jan Alexandersson in issue 26 on the Myarc HFDC card were wondering about suitable power supplies for hard drives there are some ASTEC type AC9231-01 power supplies about on the UK surplus market which will deliver 2.5 amps at 12 volts plus 6 amps at 5 volts and for about £7.50 are very good value for money.

Mike

# TI-BASE TUTORIALS

Peter Walker reviews some help for TI-BASE users from our disk library

For those of you disk owning members that haven't yet discovered TI-BASE, you don't know what you're missing. For those who have and are confused by its complexity and the need to learn a completely new language, our disk library has 4 disks from Martin Smoley of the NorthCoast 99ers. These contain 9 tutorials on TI-BASE which were originally published in the NorthCoast 99ers magazine. The first of these are based on ver 1 of TI-BASE, but nevertheless there are many tips to pick up. For those of you cautious of ordering the disks without knowing what's on them, I give here a resume of what they contain.

Tutorial 1 - Introduction, Printing, Sorting, Command Files, label printing.

Tutorial 2 - Editing records, Command File tips, using 2 databases together, nested Command Files, maths, ver 1.02 control codes.

Tutorial 3 - CASE, selection menus, Boolean logic, label printing program, speeding up Command Files, Printer control codes.

Tutorial 4 - Transferring data from one database to another.

Tutorial 5 - Complete membership list program (building on modules from earlier tutorials)

Tutorial 6 - Corrections to Tut 5, Ver 2 introduction, plus another club program.

Tutorial 7 - More on Ver 2, general advice, printer controls.

Tutorial 8 - More on printer controls, 2-across label program

Tutorial 9 - Club disk library program, graphics using X-type data.

The disks contain other helpful tutorials on using Basic to assist database conversion and all the command and data files mentioned in the articles.

Your intrepid membership secretary has not yet converted the TIUG membership database to TI-BASE. The main reason is speed. Command Files are not as fast as bespoke programs (even those written in Basic), though I have to admit that there is very little that TI-BASE can't do. Program development is not easy or fast: there are many occasions when routines don't seem to work as intended and TI-BASE does not possess an informative set of error messages. As Stephen has noted before, you can't exit looping programs using WHILE by any means other than the original WHILE conditions. So you can't exit on an IF within a WHILE loop. Those of you trained on Basic will find the need to declare variables using the LOCAL statement difficult at first; others of you used to declarative languages will accept this with ease.

A few tips from my own experience. Nested Command Files can slow down working speed but are useful in testing program modules. Also variables declared in sub-routines only take up memory space during execution of the sub-routine. Using several small command files can be edited using the TI-BASE editor, while larger files need to be edited using TI-Writer or Editor-Assembler.

Looping routines using WHILE work fastest if they loop only within one command file. The READSTRING command cannot read screen text displayed under the input field (unlike Basic's ACCEPT AT). This prevents one from designing one's own edit screens, though original entry screens are possible. If you use the normal edit screen within a command file, it is very easy when pressing "BACK" to complete editing for this to abort the command file. You must press it very quickly. I have found the lack of a SEG$ function in TI-BASE a problem though it is possible to get over this in certain cases.

While I have not converted the TIUG membership database into TI-BASE, I have done so for another fairly large membership database which my wife uses. As I said above, I have found that almost every possible function can be achieved, but command file execution can be very slow at times. I am using TI-BASE as an adjunct to the membership database, such as producing the membership area lists printed elsewhere in this issue. The task is well suited to TI-BASE: printing records if meeting the condition AREA=X and PRINT=Yes (ie those who have agreed to be printed). The main problem is that my membership database is in Display Fixed 255 format and has the right format for TI-BASE save the fact that TI-BASE requires its datafiles to have Internal format even though its data is stored in Display form. I have thus had to write a program which alters the header sector to change the format type and then a TI-BASE command file to recover and resort the file.

In conclusion, the Smoley disks are worth going through but they are only a beginning and there is no substitute for learning the hard way!

Peter Walker

```
100 CALL INIT
110 CALL LOAD("DSK1.PU/03")
120 CALL LINK("LOOP",1)
130 DISPLAY AT(12,1)ERASE ALL:"TURN ON DIALLER": :"THEN PRESS ENTER"
140 CALL KEY(3,K,V):: IF V-1 OR K()13 THEN 140
150 DISPLAY AT(12,1)ERASE ALL:"INPUT NUMBER"
160 ACCEPT AT(14,1)VALIDATE(DIGIT)BEEP:P$
170 CALL LINK("LOOP",2)
180 CALL WAIT(1500)
190 FOR J=1 TO LEN(P$)
200 K$=SEG$(P$,J,1)
210 P=POS("1234567890",K$,1)
220 CALL LINK("PULSE",P)
230 CALL WAIT(800)                     TO RELEA
240 NEXT J
250 DISPLAY AT(12,1)ERASE ALL BEEP:"  PICKUP PHONE & PRESS KEY": :"
SE DIALLER": :"**** SWITCH OFF DIALLER ****"
260 CALL KEY(3,K,V):: IF V-1 THEN 260
270 CALL LINK("LOOP",1)
280 STOP
290 SUB WAIT(D)
300 CALL SOUND(D,30000,30)
310 CALL SOUND(1,30000,30)
320 SUBEND
```

autodialler

## TELEPHONE DIALLER PROJECT 11

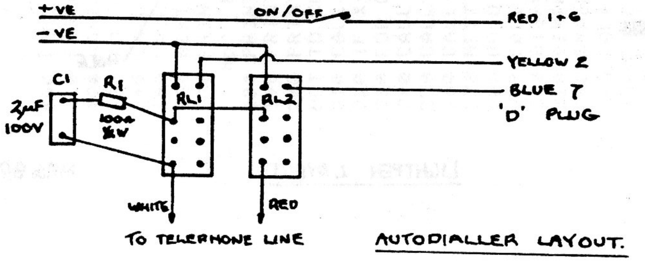
Peter Walker & Mike Goddard

This is the second part of the article which Peter started in issue 26. There isn't a great deal of hardware needed to actually run the dialler and probably choice or quality of components is more important.

The relays Peter used in his prototype were TANDY general purpose 12v types part number 275 247 the ones I used were FUJITSU DPCO miniature 1.25 amp available from Verospeed part number 258-57341C both types utilise the normally open contacts but almost any small relay should work provided it has normally open contacts and you can find a power supply to run it. My version uses 5volt relays and is powered by three penlight cells. The only type of relay to avoid are reed types as the contact break time is too long and they also generate a lot of contact bounce.

Another device which might be difficult to find is the 2 Mf line suppressor capacitor this is readily available from Verospeed as indeed are all the components needed for the project. I thought the project was too small to warrant a printed circuit board or even vero board and found the easiest method of construction was to stick the main components (relays and capacitor) upside down onto a piece of insulation board with blutak or something similar then simply solder the connections needed to the component tags direct. The line connections to a modern BT plug are the two out side connections usually coloured red and white ignore the other connections.As Peter mentioned with this circuit an on/off switch should be included this is easiest fitted in the live connection to the "D" plug connector to the cassette port and will effectively disable the circuit until needed. Shown below is a typical layout of hardware and some addresses of component suppliers.

Also with this article is another version of the software which reminds you to switch off the dialler which is extremely important because when the computer is on but not calling for the cassette motor ports they are actually on which apart from engaging your telephone line will also keep your relays on which if you use batteries will flatten them in no time!! If you have already typed in the program then you only need to change line no 250. Good luck both Peter and I look forward to any feedback from these articles and will try and sort out any problems as they occur.



AUTODIALLER LAYOUT.

| | |
|---|---|
| RL1&2 | MINIATURE RELAYS TO SUIT VOLTAGE |
| C1 | 2uF 100V CAPACITOR |
| R1 | 100Ω ¼W RESISTOR |
| | ON/OFF SWITCH |
| | 'D' CONNECTOR |

## THE LIGHTPEN AGAIN !!..........

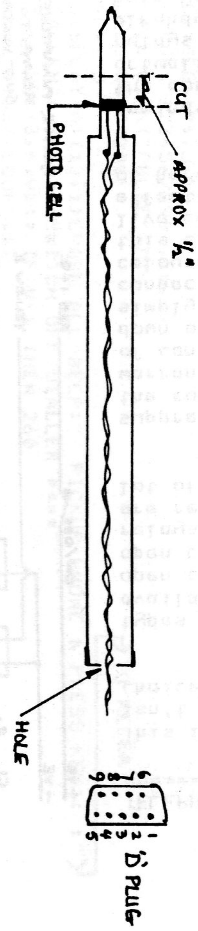
Mike Goddard

Yes you're right it is the same lightpen as before but with one important difference the software has been debugged completely by John Murphy of D.O.R.T.I.G and this version does work it was demonstrated to me by Trevor Tabener when I was on holiday earlier this year in Dorset. The idea is to place the "pen" on a series of dots randomly placed on the screen. There are two levels of difficulty which appear to be smaller dots and a different level of scoring for the harder level.

It is not a true lightpen in the sense that it doesn't read the raster of the TV or monitor screen but only responds to light so it is better used in reduced lighting and not direct sunlight etc.

The construction details are below and almost any CDS photocell should work "Tandy" types should be OK but not phototransistors as they will probably be too sensitive. It should be wired between 7 & 9 of the joystick port and a second one can be wired between 2 & 5 any old pen case can be used provided the p/cell can fit inside and should be glued into place. The connecting wire should be thin screened and about 1M long.

Mike



LIGHTPEN LAYOUT

MRG 89

LIGHTPEN SOFTWARE ORIGINALLY FROM THE SAN FRANCISCO 99ERS

DEBUGGED BY JOHN MURPHY OF D.O.R.T.I.G

```
100 CALL CLEAR :: CALL SCREEN(2)
110 FOR X=1 TO 10 :: CALL COLOR(X,16-X,1):: NEXT X
120 RANDOMIZE
130 FOR X=0 TO 2
140 CALL CHAR(96+8*X,"3C7EFFFFFFFF7E3C")
150 CALL COLOR(9+X,1,1)
160 NEXT X
170 PRINT " `hp`hp     `hp    `hp`hp`  hp`  `  p`h   ` p`h   hp`  h hp`      h  p`h
   hp` h  hp`  h    p`h"
180 PRINT " hp`   h  hp`  h   p`h  hp`  h hp`h   p`h  hp`   `   p`h   `   p`h"
190 PRINT "`hp`hp      `hp      p`h": : : :
200 PRINT : :"`hp`hp`hp`hp`hp`hp`hp`hp`hp`p   TI 99    LITE PEN     hh (TOUCH
DOT TO CONTINUE) p`ph`ph`ph`ph`ph`ph`ph`ph`ph`"
210 CALL JOYST(1,X,Y)
220 IF X<>0 OR Y<>0 THEN 290
230 FOR C=1 TO 3
240 CALL COLOR(9,7-4*(C=1)-8*(C=2),1)
250 CALL COLOR(10,7-4*(C=2)-8*(C=3),1)
260 CALL COLOR(11,7-4*(C=3)-8*(C=1),1)
270 NEXT C
280 GOTO 210
290 CALL SOUND(100,440,0):: CALL CLEAR :: SC=0
300 DISPLAY AT(12,9):"EASY      HARD"
310 DISPLAY AT(16,7):"SELECT DIFFICULTY" :: DISPLAY AT(18,1):"EASY=LARGE DOTS, H
ARD=SMALL."
320 CALL HCHAR(12,9,112):: CALL HCHAR(12,19,104)
330 CALL COLOR(11,16,16,10,2,2)
340 FOR I=1 TO 10
350 CALL JOYST(1,X,Y)
360 IF X=4 THEN CALL MAGNIFY(2):: GOTO 440
370 NEXT I
380 CALL COLOR(11,2,2,10,16,16)
390 FOR I=1 TO 10
400 CALL JOYST(1,X,Y)
410 IF X=4 THEN CALL MAGNIFY(1):: GOTO 440
420 NEXT I
430 GOTO 330
440 CALL CLEAR
```

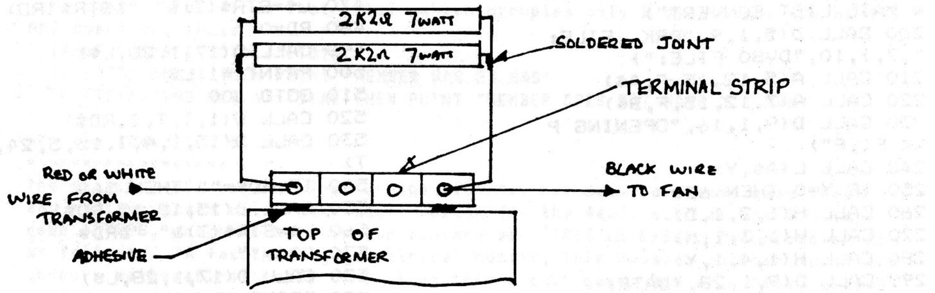
```
450 CALL SPRITE(#1,96,16,92,124)
460 DISPLAY AT(16,6)BEEP:"TOUCH DOT TO START"
470 CALL JOYST(1,X,Y):: IF X<4 THEN 470
480 CALL SOUND(-100,220,5):: CALL SOUND(-100,888,0)
490 CALL CLEAR
500 FOR L=1 TO 20
510 CALL SPRITE(#1,96,16,INT(RND*170)+1,INT(RND*230)+10)
520 N=0
530 CALL JOYST(1,X,Y)
540 IF X=4 THEN 560
550 N=N+1 :: GOTO 530
560 SC=SC+N :: CALL SOUND(-100,440,5)
570 DISPLAY AT(1,1):"SCORE ";SC
580 FOR X=1 TO 100
590 NEXT X
600 NEXT L
610 FOR Z=1 TO SC STEP 10
620 CALL SOUND(-100,Z+110,0)
630 NEXT Z
640 CALL CLEAR :: CALL SPRITE(#1,96,16,150,123)
650 DISPLAY AT(10,6):"YOUR SCORE IS ";SC :: FOR D=1 TO 500 :: NEXT D :: DISPLAY
AT(18,3):"TOUCH DOT TO PLAY AGAIN."
660 IF SC>150 THEN 670 ELSE DISPLAY AT(12,1):"YOU CAN'T FOOL ME! YOU      CHEATE
D!" :: GOTO 740
670 IF SC>200 THEN 680 ELSE DISPLAY AT(12,1):"VERY GOOD!" :: GOTO 740
680 IF SC>225 THEN 690 ELSE DISPLAY AT(12,1):"WOW! YOU NAMED RAMBO?" :: GOTO 740
690 IF SC>250 THEN 700 ELSE DISPLAY AT(12,1):"NOT BAD, BUT YOU PAY SOMEONETO SWA
T FLIES FOR YOU?" :: GOTO 740
700 IF SC>300 THEN 710 ELSE DISPLAY AT(12,1):"YOU NEED PRATICE!" :: GOTO 740
710 IF SC>350 THEN 720 ELSE DISPLAY AT(12,1):"HAVE YOU CONSIDERED LUDO?" :: GOTO
740
720 IF SC>400 THEN 730 ELSE DISPLAY AT(12,1):"HELLO! ANYBODY AWAKE OUT    THERE?
" :: GOTO 740
730 DISPLAY AT(12,1):" TRY POINTING THE PEN AT THE DOT"
740 FOR Z=1 TO 500
750 CALL JOYST(1,X,Y)
760 IF X=4 THEN 790 ELSE 770
770 NEXT Z
780 CALL CLEAR :: DISPLAY AT(12,1):" DOTS ALL FOLKS!" :: END
790 CALL DELSPRITE(ALL):: FOR D=1 TO 500 :: NEXT D :: GOTO 200
```

## SILENCING THE BOX.......

Mike Goddard

I've given some time recently to trying to silence the expansion box. The first job was to establish how noisy the fan is on its own so I removed it from its mounting and gave it a run and guess what ? its not noisy at all it is the method of mounting that causes the problems because the fan blows outwards as the air passes through the holes in the back of the box it causes turbulence with the fan blades (a bit like blowing and moving your fingers in front of your mouth) and therefore produces noise. As the fan cannot be mounted the other way round. I then tried fitting a 240 volt fan of the same size but blowing into the box the resulting noise was only slightly quieter and higher pitched so really just as irritating. Next I thought of reducing the voltage to the original fan using the alternative transformer tappings available in the mark 1 box again results were disappointing with only a very slight reduction in noise. Then I remembered Ross Bennetts modification of a 1k0 15watt resistor in series with the fan after a visit to my local TV dealer I fitted a resistor and bingo "silence" the box was so quiet I could hear the faint hum of the transformer so this does seem to be the best method if you want a quiet box the only snag I could find was that 1k0 15watt resistors aren't very common and I in fact used two 2k2 7watt resistors in parallel which in fact gives 1k1 14watt but as the fan only consumes 11watts there is plenty of leeway. I mounted my resistor in a piece of strip connector and stuck it with hot melt glue to the top of the transformer so the resistor sticks up into the airflow of the fan to help cool the resistor as it does get fairly hot in use the other little refinement I added was a small single pole switch mounted in the back of the box and connected across the resistor so that when it is switched on full power is restored to the fan giving in effect two speeds.

Mike



BOX SILENCER LAYOUT MRG 89

# CONVERSION ROUTINES

BY PETER WALKER & STEPHEN SHAW

As mentioned elsewhere, only one person responded to my request for good conversion programs. Stephen submitted a program written in "Enhanced Basic" (that resident in the Personal Record Keeping (PRK) and Stats modules) to convert a PRK data file into DV80 format. I noted two small problems with the routine. Firstly using CALL G on an empty record does not cause the input variable to be set to "". Also the disk file CLOSE statement was omitted. I have therefore decided to publish a slightly improved version of the program, which also demonstrates the use of CALL A and CALL D routines. It converts PRK data files to TI-Writer compatible DV80 files for use in Mailing Lists. (refer to page 113 of the TI-Writer Reference Guide).

My thanks to Stephen; where are all the rest of you budding programmers?

Peter Walker

```
100 REM PRK TO TIW MAIL LIST
 CONVERSION
110 REM BY STEPHEN SHAW & PE
TER WALKER
120 REM INSERT PRK MODULE
130 REM  FIRST DO THIS IN IM
MEDIATE MODE
140 REM  CALL P(N) N>256*SEC
TORS IN PRK FILE
150 REM NEW
160 REM CALL FILES(1)
170 REM NEW
180 CALL CLEAR
190 CALL D(1,1,28,"PRK TO TI
W MAIL LIST CONVERT")
200 CALL D(5,1,9,"PRK  FILE:
",7,1,10,"DV80 FILE:")
210 CALL A(5,12,15,F,A$)
220 CALL A(7,12,15,F,B$)
230 CALL D(9,1,16,"OPENING P
RK FILE")
240 CALL L(A$,Y)
250 IF Y=0 THEN 650
260 CALL H(1,2,1,D)
270 CALL H(1,3,1,M)
280 CALL H(1,4,1,Y)
290 CALL D(9,1,28,"DATE:
/")
300 CALL D(9,6,2,D)
310 CALL D(9,9,2,M)
320 CALL D(9,12,4,Y)
330 CALL H(1,5,0,F)
340 CALL D(11,1,28,"FIELDS:"
)
350 CALL D(11,8,4,F)
360 CALL H(1,6,0,R)
370 CALL D(11,14,13,"RECORDS
:")
380 CALL D(11,22,4,R)
390 CALL D(13,1,28,"OPENING
DV80 FILE")
400 OPEN #1:B$
410 FOR I=1 TO R
420 FOR T=1 TO F
430 CALL H(1,10,T,TP)
440 IF TP=1 THEN 520
450 CALL G(1,I,T,Z,RD)
460 CALL D(15,1,4,I,15,5,4,T
,15,10,4,RD)
470 L$=STR$(T)&" "&STR$(RD)
480 RD=0
490 CALL D(17,1,28,L$)
500 PRINT #1:L$
510 GOTO 600
520 CALL G(1,I,T,Z,RD$)
530 CALL D(15,1,4,I,15,5,24,
T)
540 IF RD$="" THEN 560
550 CALL D(15,10,15,RD$)
560 L$=STR$(T)&" "&RD$
570 RD$=""
580 CALL D(17,1,28,L$)
590 PRINT #1:L$
600 NEXT T
610 PRINT #1:"*"
620 NEXT I
630 CLOSE #1
640 STOP
650 CALL D(17,1,28,"ERROR CA
LL P(N)",19,1,28,"N WAS TOO
SMALL")
```

# RAMBLES

RAMBLES
For TI*MES issue 26, Jan 1990

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!).

Rambles is based on any letters or queries received, and if you don't write to me, it is based on what I am doing with my computer at the time of writing... there is so much still to do with this computer. People changing to other computers have different needs to me, and may perhaps not spend so much time programming- or perhaps they spend more time trying to come to grips with rather less friendly operating systems and languages...

Write to me! Queries - and articles- from console only users are very welcome!
==============

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 wher you are dealing with disk data files...

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

The index to the first 26 issues of TI*MES has 1000 items listed! and takes up a fair bit of paper when printed out. Readily available on disk either in TI Base format for TI Base owners! or I MAY be able to produce a DV80 text format, but I'm not too sure about a printed format due to cost...

Older members will not be too surprised (or indeed may be very surprised?) to learn that from the entire five copies of TIHCUC Newsletter, I managed to find 36 entries for an index!

===========

## CORRECTION. ISSUE #17.

Yes yes, that was all a long time ago, it just takes you that long to key programs in and tell me they won't work... this could win a prize for the longest time taken to find and squash a bug?

Back in March 1985 Lehigh 99ers published a program based on my articles in TI*MES Issue 6, and this was reprinted in the Sydney Group Newsletter in October 1985. It finally made it to TI*MES 17, and can be found on Page 58.

It is a music program with no Call Sounds, and only 5 CALL LOADs.

One error seems to have ocurred in paste-up, the other I cannot explain. Here are the lines to change:

```
 590 DATA 0295D2010491D0AD1101
 1120 DATA 04FF9FBFDFE0
```

THE PROGRAM WILL NOW RUN CORRECTLY. Remember to press ENTER when the blinking cursor appear around screen top left!

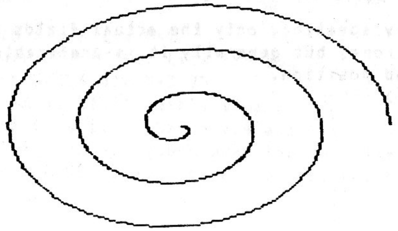
===============

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. Internatinally 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 machine code 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.

---> more --->

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.)

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

## CHAOS

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
```

Notice that after all these years there remain ample modules? (Apart from temporary local outages). Recent modules have no date on them. Truth in the rumour that TI continue to manufacture in secret!????

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?

(continued--->)

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 mang 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 warming 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...

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

```
1 ! STRAIGHT-LINE CALCULATOR
    TINYGRAM by Jim Peterson
    Accepts input such as
    6+6-11*2+3/4
2 T,F=0 :: C$="+-*/" :: ACCE
PT AT(12,1)ERASE ALL VALIDAT
E(NUMERIC,C$):F$ :: L=LEN(F$
):: FOR J=1 TO L :: X$=SEG$(
F$,J,1):: P=POS(C$,X$,1):: I
F P=0 THEN 5
3 IF F=0 THEN T=VAL(SEG$(F$,
1,J-1)):: F=1 :: A=J+1 :: P2
=P :: GOTO 5
4 V=VAL(SEG$(F$,A,J-A)):: A=
J+1 :: GOSUB 7 :: P2=P
5 NEXT J :: V=VAL(SEG$(F$,A,
255)):: GOSUB 7 :: DISPLAY A
T(12,L+1):"=";STR$(T)
6 DISPLAY AT(24,1):"PRESS AN
Y KEY" :: CALL KEY(0,K,S)::
IF S=0 THEN 6 ELSE 2
7 IF P2=1 THEN T=T+V ELSE IF
 P2=2 THEN T=T-V ELSE IF P2=
3 THEN T=T*V ELSE T=T/V
8 RETURN
```

That's all, folks!

Genial Computerware is no more. JP Software has arisen, with the same products from a new address. JP Software is at 2390 El Camino Real, #107, Palo Alto, Ca, USA, 94306. Mail orders must be supported by international money orders.

^^^^^^^^^^^^^^^^^^^^

John Guion, designer of the Multi-Mod, and upgrades for TI RS232 and TI Disk Controller Cards, died age 22 in September 1989 in a car accident. He designed the P-Gram card, designed a 32k ram for the 16 bit bus, and helped improve the Horizon Ram Card. His active support for our machine will be missed. There is no news regarding continued support or supply of the products he distributed himself.

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

AN APPRECIATION...

When the TI99/4 was first launched, there was very little support for it. Early owners enjoyed the support of the International TI99/4 Users Group in Bethany, Ohio, led by Charles LaFara. The first UK group of TI owners, TI Home, was among many beneficiaries of this group, and received a number of software items to start off a UK software library for the use of members.

The first software librarian for the Bethany group was Guy-Stefan Romano. In 1981 Guy commenced the Amnion Helpline, a telephone advice service *free* for TI99/4(a) users.

Later, as the Bethany group expanded to 100,000 members, the only possible operation was as a commercial group, and Guy left to continue library support on his own. Guy operated a telephone advice line to US TI owners, called AMNION HELPLINE, entirely free of charge. Guy helped TI owners right from the start, and continued to do so until they chose not to seek his help any longer. He wrote recently that over a six month period he had had no requests for software and only a couple of phone calls.

I gather the Amnion Helpline was financed by an ancient Swiss company but as with all Swiss corporate matters, the details are foggy.

Despite his long term and outstanding contribution to the TI Scene, Guy is not mentioned in the excellent book "The Orphan Chronicles", possibly due to his part in assisting with the book. There is a reference merely to "one close former-IUG insider".

UK disk owning members can find a file on the disk library catalogue called AMNION which contains selected items from Guys large collection of software.

Guy was born in France, with Ph.D's in Romance languages, Germanic languages, and fine arts, with a Masters in Chinese. He was fluent in 17 languages! He spent time teaching languages, and commenced work on a computer based translation machine in the 1950's

Guy knew England well. In employment he was an "internationally known potter" and a ceramic chemist - he spent some years learning his trade in the Staffordshire potteries. He was also a graduate chef. Guy developed a unique high temperature stoneware for cooking, marketed as AMNION WARE, which I gather was used by Elizabeth David (did I get that name right?) and David Mellor. He also wrote some cookery books.

Guy died in August 1989 age 57. His love care and devotion to all TI99/4 and 4A owners will be missed, but his contribution will be remembered.

Here is an odd recipe from Guy to remember him by...
...Here is a Hawaiian quick bread that is my downfall. Eat this still slightly warm with scads and scads of butter (and put a black wreath on your bathroom scale). This recipe serves... never enough...

KONA BREAD.
Preheat oven to 350F
Cream together 1 cup of raw sugar and 1/2 cup of butter. (Beat until fluffy).
Add: 2 eggs, 3/4 cup ripe bananas (bananas should be almost to the point where you are contemplating throwing them away - black and rather mushy. Why? Because almost all of the starch is converted to fruit sugar by this time.).
Mix together 1 teaspoon baking soda, 2 teaspoons baking powder, 1 1/4 cups flour. Then gently fold into banana mix. DO NOT OVERMIX!
Pour into buttered loaf pan and bake about 45 mins. GORGE YOURSELF.
PS. To gild the banana, so to speak, you may add 1/2 cup chopped macadamia nuts or pecans to this.

(American measures!)
======================

An 80 column card is available for the TI99/4A, compatible with the 80 column version of Funlweb. For details and ordering information, contact the manufacturers:
Dijit Systems, 4345 Hortensia Street, San Diego, CA, USA, 92103, or 'phone (619) 295 3301.
Note that you will need a suitable monitor!
=====================

```
100 ON WARNING NEXT :: DISPL
AY AT(3,10)ERASE ALL:"KALKUL
ATOR":"":"Input 1st value an
d Enter.":"Input other value
s preceded":"by +,-,* or / a
nd Enter." ! by Jim Peterson
101 DISPLAY AT(8,1):"Input =
 and Enter to get":"final re
sult."
110 R=14 :: C=1 :: ACCEPT AT
(12,1):N :: V=N :: F=1 :: N$
=STR$(N):: GOSUB 200
120 ACCEPT AT(12,1)VALIDATE(
"+-*/=",NUMERIC):N$ :: A=POS
("+-*/=",SEG$(N$,1,1),1):: G
OSUB 200 :: IF A=0 THEN 120
:: IF A=5 THEN 160
```

```
130 ON ERROR 140 :: N=VAL(SE
G$(N$,2,LEN(N$)-1)):: GOTO 1.
50
140 CALL SOUND(200,110,5,-4,
5):: C=C-LEN(N$):: DISPLAY A
T(R,C):"" :: RETURN 120
150 IF A=1 THEN V=V+N :: GOT
O 120 ELSE IF A=2 THEN V=V-N
 :: GOTO 120 ELSE IF A=3 THE
N V=V*N :: GOTO 120 ELSE IF
A=4 THEN V=V/N :: GOTO 120
160 DISPLAY AT(R,C):STR$(V):
: F,V=0 :: GOTO 110
200 DISPLAY AT(R,C):N$ :: C=
C+LEN(N$):: IF C>20 THEN C=1
 :: R=R+1 :: RETURN ELSE RET
URN
```
=======================

# EX BAS ONLY PLOTTING- 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(T,16,2) :: NEXT T
...
```
 YOUR GRAPHICS PROGRAM THEN FOLLOWS. Ignore any initiallisation 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 conecting 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.

## Mini-Memory Madness

By Peter Hutchison

I decided to write about the Mini-Memory module which must be one of the most under used modules around (hardly a mention anywhere) as really it is considered as a bit of a damp squid. Well, I'm going to turn that on its head: Mini-Memory is one of the most useful cartridges around, why?

What does it have to offer?

* 4K of directly accessible RAM (not VDP ram in the console)
* 4K of ROM containing routines for 7 new BASIC CALLs and XML machine code routines e.g. Floating Point arithmetic.
* 6K of Graphics ROM containing GPL machine code routines e.g. Device control and trigonometry arithmetic.
* A Line-By-Line Assembler and a LINES demo program.
* Easy Bug for debugging m/c programs and examining/changing memory.

What can you do with 4K of RAM?

The TI-99/4A has 16K of built in RAM but, and a big but it is, is that NONE of it is directly accessable by the TMS9900 processor, everything has to go through scratch pad RAM and that what makes the Texas SLOW! With 4K of RAM you can write super-fast machine code programs with access to all display modes i.e. standard Graphics mode, 80-column Text mode, Multicolor mode, Bit-Map mode and Sprites!!!

You can also save you favourite BASIC program in RAM using SAVE MINIMEM (if it will fit in 4K) and load it up within seconds by OLD MINIMEM ready to run - no slow disk or tape loading!!! You could use it as file space and access it randomly like a disk - useful for storing data temporarily for example extra screens for games or storing adventure locations descriptions.

What use is the Line-By-Line Assembler?

Well, lets face it the LBL assembler is no match for the Editor/Assembler and it takes up most of the 4K RAM itself leaving 768 bytes for your own programs but it does not requires disk drives although 32K RAM expansion is necessary (I d'don't have any though). The LBL converts the assembly directly into m/c as you type it in, and uses directives and labels. It can use the 32K RAM expansion for much larger programs (the new Zenoboard will come in useful here!). What kind of programs could you write: scrolling routines, routines to access some of the other display modes allowing use of proper graphics, small games and routines for use by BASIC such as real lowercase letters, the list is endless.

How can I use it from BASIC?

There are 7 new CALLs available from TI BASIC when the module is inserted:

| | |
|---|---|
| CALL INIT | To initialize the module and 32K if connected. |
| CALL LOAD | To load programs from disk and poke CPU RAM. |
| CALL PEEK | To examine the contents of CPU RAM. |
| CALL POKEV | To poke VDP RAM e.g. the screen. |
| CALL PEEKV | To examine the contents of VDP RAM. |
| CALL CHARPAT | Returns the hex pattern of a character. |
| CALL LINK | Runs m/c programs and pass parameters. |

The above subprograms can be used to speed up those slow TI BASIC programs by replacing all those slow CALL V/HCHARs with CALL POKEVs and CALL GCHARs with CALL PEEKVs.

For example, the familiar text print at routine can be replaced with

```
1000 FOR C=1 TO LEN(TEXT$)
1010 CALL POKEV(A+C-1,ASC(SEG$(TEXT$,C,1))+96)
1020 NEXT C
```

where C is the count of characters, TEXT$ contains the text to be printed and A is the address of screen (0-767). Notice that an offse of 96 must be added to character value in POKEV due to different nature between BASIC and machine level.

There are a few programs available in the Tape Library using Mini-Memory: MM Racer, Desert Car (using scrolling routines), Disassembler (recommended for m/c programmers) and Joy Sketch.

If you do buy Mini-Memory (#38 from Database) it is definitely recommended to get the Editor/Assembler Manual (#15) or Introductio to Assembly Language (#10) which will help to teach beginners. Their is also Artist (#7) for artists amongst TI owners. It is definite must for bored BASIC programmers (like me)!!!

6 Moorlands View
Free School Lane
Savile Park
Halifax
HX1 2XQ

## Treasure Trail

by Peter Hutchison.

Writing games in BASIC can be rewarding but also a challenge as you have to fit everything in 14K, Extended Basic helps a lot though with multi-statement lines. I have written 3 adventure games in the past:-

a) Challenge of the Dark Empire.
Has 36 locations (on square map),16 commands using verb/noun system and descriptions in DATA statements read into an array. The game was easy (can be completed in a session) and linear. Originally in TI-BASIC, converted to another computer.

b) Challenge of the Dark Tower.
Has 32 locations and 19 instructions also verb/noun format. Descriptions also in DATA and read into an array, limited length of descriptions. Improved game. TI BASIC again.

c) Halley's Comet.
Has 73 locations and 29 instructions in verb/noun format! Descriptions were also short but compacted data for moving between locations into 8 digit strings e.g. NNSSWWEE. Flags are characters in one string, DEF functions used. Game much harder and more enjoyable. Written in Extended BASIC.

Problems with writing adventures:

1) Descriptions take up a LOT of memory. They should not be in DATA statements AND in arrays. Best methods are to either READ them from DATA statements (use RESTORE at beginning of blocks of DATA to speed it up) or read data into arrays from tape (probably best method).
2) Objects stored in an array with a number indicating whether they are at a given location, being carried, or hidden from view.
3) Flags should be single characters stored in a string and accessed through SEG$. Arrays of strings and especially numbers are too memory hungry.
3) Parser should be restricted to Verb Noun format as it reduces code overhead and therefore allows new commands to be added. Also you should include variations of same command e.g. TAKE and GET, DROP and PUT, SHOOT and KILL, N and NORTH etc.

There are a few BASIC games available in the Library such as Adevnturemania(G4), Cake/Adv (G49), Cave Maze(G52), StarShip Supernova (G112), Haunted House/Wumpus(G127)?, Mania(G160), Santa & the Goblins (G208) hopefully I'll be able to have a good look how good they play and what techniques they've used.

Now for some tips on the Strange Odyssey adventure, which I have not completed - nearly there though:

1) Can't get off ledge? !pumJ
2) Boulder blocking your way? ti yortseD
3) Don't know what to do in hex room? PLEH dna dor enimaxE
4) Rigilian hound attacks you? ti nutS
5) Getting squashed on mining colony? muesuM eht tisiV
6) Can't open hatch? ynoloc gninim eht tisiV
7) Can't find fifth treasure? sieklaw rof dnuoh ekaT
9) Running out of air? enihcam neila esU

Now can someone help me, please?
1) How do I repair the broken Power Crystal?
2) What use is the White Button?
3) What use is the Black Emptiness?

Replies as allways to: 6 Moorlands View, Free School Lane, Savile Park, Halifax HX1 2XQ

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 homebase, 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 cheap LW portable radio for 3.50, which sells for 5.99 in Manchester- the catch with this being the 2.5mm headphone socket- not the usual 3.5mm socket!

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

```
1 REM FREEFORM ART BY        STEPHEN SHAW 1982          FOR MYARC XB NOV 1989
2 REM BASED ON WORK BY       JEREMY RUSTON
3 REM
4 RANDOMIZE
5 XX=INT(RND*110+40) :: YY=INT(RND*110+40) :: LL=INT(RND*110+40) :: MM=INT(RND*1
10+40) :: UU=10-RND*20 :: VV=10-RND*20
6 PP=10-RND*20 :: QQ=10-RND*20
7 CALL GRAPHICS(3)
8 FOR K=1 TO 30
9 X2=XX :: Y2=YY :: X3=LL :: Y3=MM :: GOSUB 16
10 IF XX+UU>150 OR XX+UU<40 THEN UU=-UU
11 IF YY+VV>150 OR YY+VV<41 THEN VV=-VV
12 IF LL+PP>150 OR LL+PP<39 THEN PP=-PP
13 IF MM+QQ>148 OR MM+QQ<38 THEN QQ=-QQ
14 XX=XX+UU :: YY=YY+VV :: LL=LL+PP :: MM=MM+QQ :: NEXT K :: CALL WRITE(1,24,3,"
SPACE FOR ANOTHER")
15 CALL KEY(0,A,B) :: IF A=32 THEN RUN ELSE 15
16 DX=X3-X2 :: DY=Y3-Y2 :: IF (DX=0)+(DY=0) THEN 20
17 IF ABS(DX)>ABS(DY) THEN 19
18 FOR LCV=Y2 TO Y3 STEP SGN(DY) :: R=INT(.5+LCV) :: C=INT(.5+X2+DX/DY*(LCV-Y2))
 :: CALL DOT(R,C) :: NEXT LCV :: RETURN
19 FOR LCV=X2 TO X3 STEP SGN(DX) :: C=INT(.5+LCV) :: R=INT(.5+Y2+DY/DX*(LCV-X2))
 :: CALL DOT(R,C) :: NEXT LCV :: RETURN
20 IF (DX=0)-(DY=0) THEN 21 ELSE RETURN
21 IF DY=0 THEN 23
22 C=INT(.5+X2) :: FOR LCV=Y2 TO Y3 STEP SGN(DY) :: R=INT(.5+LCV) :: CALL DOT(R,
C) :: NEXT LCV :: RETURN
23 R=INT(.5+Y2) :: FOR LCV=X2 TO X3 STEP SGN(DX) :: C=INT(.5+LCV) :: CALL DOT(R,
C) :: NEXT LCV :: RETURN
24 STOP
25 SUB DOT(R,C) :: CALL POINT(1,R,C) :: SUBEND
26 END
```

# TETRIS

TETRIS

Tetris is a computer program from the Soviet Union, now converted for most computers and even released as a coin-op machine, it's THAT good. Very simple in concept, it can take a moment or two to realise in play what you are meant to do- different shapes fall from above, and by rotating them, and moving them left and right, you have to do all you can to pack them so tightly that complete rows with no holes are made up - when you do the whole stack moves down as that row disappears.

The first TI version was TI TRIS, the version that follows is TetrIs and comes from the September 1989 issue of MICROpendium. Just think, if you had a subscription you could have been playing TetrIs some months ago! As with all programs with DATA statements, take very great care keying this in! This printout is from a program which has been up and running correctly, and can therefore be guaranteed free from bugs!

The disk library also has a machine code version from Germany. The original program was by Soviet citizen Alexey Pajitnov.

```
100 CALL START(E)
110 DIM Z$(23),Z(26),A(18,3)
,B(18,3):: RANDOMIZE :: C$="
JKL; UQSDFAR" :: Z(24)=4095
:: CALL MAGNIFY(4):: CALL CL
EAR :: FOR I=0 TO 6
120 READ N(I),C(I):: CALL CO
LOR(I+8,2,C(I)):: NEXT I ::
FOR I=0 TO 18 :: FOR J=0 TO
3 :: READ A(I,J),B(I,J):: NE
XT J :: NEXT I
130 FOR I=68 TO 143 :: READ
X$ :: CALL CHAR(I,X$):: NEXT
 I :: CALL CHAR(41,RPT$("FF"
,8))
140 FOR I=0 TO 23 :: Z$(I)=R
PT$(")",10):: Z(I)=2049 :: N
EXT I :: V=E :: D,P=24 :: U=
0 :: GOSUB 450 :: CALL VCHAR
(1,12,41,240)
150 CALL KEY(5,M,W)
160 P=0 :: Q=4 :: J=INT(RND*
7):: S=J*2 :: J8=J*8+89 :: I
F J>3 THEN S=S-1+2*(J-4)
170 GOSUB 470 :: T=0 :: X=1
:: Y=Q*8+81 :: CALL SPRITE(#
1,K,C(J),X,Y)
180 IF Z(O)AND 2^(Q+Y1)OR Z(
X2)AND 2^(Q+Y2)OR Z(X3)AND 2
^(Q+Y3)OR Z(X4)AND 2^(Q+Y4)T
HEN 430
190 FOR I=1 TO V :: CALL KEY
(5,M,W):: IF M<0 THEN 350 EL
SE ON POS(C$,CHR$(M),1)+1 GO
TO 350,210,280,230,340,250,3
30,440,210,280,230,340,330
```

```
200 REM
210 Q=Q-1 :: IF Z(P)AND 2^(Q
+Y1)OR Z(P+X2)AND 2^(Q+Y2)OR
 Z(P+X3)AND 2^(Q+Y3)OR Z(P+X
4)AND 2^(Q+Y4)THEN Q=Q+1 ELS
E Y=Y-8
220 CALL LOCATE(#1,X,Y):: GO
TO 350
230 Q=Q+1 :: IF Z(P)AND 2^(Q
+Y1)OR Z(P+X2)AND 2^(Q+Y2)OR
 Z(P+X3)AND 2^(Q+Y3)OR Z(P+X
4)AND 2^(Q+Y4)THEN Q=Q-1 ELS
E Y=Y+8
240 CALL LOCATE(#1,X,Y):: GO
TO 350
250 Y1=2^(Q+Y1):: Y2=2^(Q+Y2
):: Y3=2^(Q+Y3):: Y4=2^(Q+Y4
):: GOSUB 450 :: P=D-X4
260 IF (Z(P)AND Y1 OR Z(P+X2
)AND Y2 OR Z(P+X3)AND Y3 OR
Z(P+X4)AND Y4)=0 THEN P=P+1
:: GOTO 260
270 P=P-1 :: CALL LOCATE(#1,
P*8+1,Y):: GOTO 380
280 S=S-1 :: T=T-1 :: IF T<0
 THEN T=N(J)-1 :: S=S+N(J)
290 GOSUB 470
300 IF (Z(P)AND 2^(Q+Y1)OR Z
(P+X2)AND 2^(Q+Y2)OR Z(P+X3)
AND 2^(Q+Y3)OR Z(P+X4)AND 2^
(Q+Y4))=0 THEN CALL PATTERN(
#1,K):: GOTO 350
310 S=S+1 :: T=T+1 :: IF T=N
(J)THEN T=0 :: S=S-N(J)
320 GOSUB 470 :: GOTO 350
```

```
330 CALL KEY(5,M,W):: IF W<>
0 THEN 330 ELSE V=V+(V>1)::
GOSUB 460 :: GOTO 350
340 CALL KEY(0,M,W):: IF W<>
1 THEN 340
350 NEXT I :: P=P+1 :: IF P+
X4>=D THEN 370
360 X=X+8 :: CALL LOCATE(#1,
X,Y):: GOTO 190
370 IF (Z(P)AND 2^(Q+Y1)OR Z
(P+X2)AND 2^(Q+Y2)OR Z(P+X3)
AND 2^(Q+Y3)OR Z(P+X4)AND 2^
(Q+Y4))=0 THEN 360 ELSE P=P-
1 :: GOSUB 450
380 D=MIN(D,P):: FOR I=0 TO
3 :: W=Q+B(S,I):: M=P+A(S,I)
:: Z(M)=Z(M)+2^W :: Z$(M)=SE
G$(Z$(M),1,W-1)&CHR$(J8)&SEG
$(Z$(M),W+1,10)
390 CALL HCHAR(M+1,W+11,J8):
: NEXT I :: CALL DELSPRITE(#
1):: FOR I=MIN(P+3,23)TO P S
TEP -1 :: IF Z(I)<4095 THEN
420 ELSE J=I :: M=I-1
400 Z(J)=Z(M):: Z$(J)=Z$(M):
: DISPLAY AT(J+1,10):Z$(J)::
 IF Z(J)>2049 THEN J=J-1 ::
M=M-1 :: GOTO 400
410 U=U+INT(500/V):: GOSUB 4
60 :: I=I+1 :: P=P-1 :: D=D+
1
420 NEXT I :: GOTO 160
430 H=MAX(H,U):: DISPLAY AT(
1,20):USING "#########":H ::
 CALL DELSPRITE(#1):: GOTO 1
40
440 DISPLAY ERASE ALL:"HIGH
SCORE IS ";MAX(U,H):: STOP
450 U=U+INT((24-P)*100/V)
460 DISPLAY AT(3,20):USING "
#########":U :: V=MIN(V,MAX(
1,9-INT(U/5000))):: DISPLAY
AT(3,4)SIZE(2):10-V :: RETUR
N
470 X2=A(S,1):: X3=A(S,2)::
X4=A(S,3):: Y1=B(S,0):: Y2=B
(S,1):: Y3=B(S,2):: Y4=B(S,3
):: K=68+S*4 :: RETURN
480 DATA 2,15,2,7,2,14,1,16,
4,11,4,4,4,5
490 DATA 0,0,0,1,0,2,0,3,0,1
,1,1,2,1,3,1,0,0,0,1,1,1,1,2
,0,2,1,1,1,2,2,1
```

```
500 DATA 0,1,0,2,1,0,1,1,0,1
,1,1,1,2,2,2,0,1,0,2,1,1,1,2
,0,1,1,0,1,1,1,2
510 DATA 0,1,1,1,1,2,2,1,0,0
,0,1,0,2,1,1,0,2,1,1,1,2,2,2
520 DATA 0,0,1,0,1,1,1,2,0,1
,0,2,1,1,2,1,0,0,0,1,0,2,1,2
,0,2,1,2,2,1,2,2
530 DATA 0,2,1,0,1,1,1,2,0,1
,1,1,2,1,2,2,0,0,0,1,0,2,1,0
,0,1,0,2,1,2,2,2
540 DATA FFFFFFFF,,FFFFFFFF,
,OFOFOFOFOFOFOFOF,OFOFOFOFOF
OFOFOF,,
550 DATA FFFFFFFFOFOFOFOF,,O
OOOOOOFOFOFOF,,OOOOOOOOFOF
OFOF,OFOFOFOF,FOFOFOFOFOFOFO
FO,,OFOFOFOFFFFFFFFF,,FOFOFO
F,
560 DATA OFOFOFOFOFOFOFOF,,O
OOOOOOFOFOFOFO,FOFOFOF,OFOF
OFOFOFOFOFOF,,FOFOFOFOFOFOFO
F,,OFOFOFOFFFFFFFFF,,OOOOOOO
OFOFOFOF,
570 DATA OFOFOFOFOFOFOFOF,OF
OFOFOF,OOOOOOOFOFOFOF,,FFFF
FFFFOFOFOFOF,,FOFOFOF,,OOOOO
OOOFOFOFOF,,FOFOFOFOFOFOFOF
,FOFOFOF
580 DATA FOFOFOFOFFFFFFFF,,O
OOOOOOFOFOFOF,,OFOFOFOFOFOF
OFOF,OFOFOFOF,FOFOFOF,,FFFFF
FFF,,FOFOFOFOFOFOFOFO,,,OFOF
OFOF,FOFOFOFOFOFOFOF,FOFOFOF
590 DATA OOOOOOOOFFFFFFFF,,F
OFOFOFOFOFOFOFO,,OFOFOFOFOFO
FOFOF,OFOFOFOF,,FOFOFOF,FFFF
FFFFOFOFOF,,FOFOFOF,,OFOFOF
OF,,FOFOFOFOFOFOFOF,FOFOFOFO
600 END
```

```
20000 SUB START(E)
20010 DISPLAY AT(1,1)ERASE A
LL:"TETRIS by":"Steven Karas
ek 1989":"from Computer Brid
ge JUne89"
20020 DISPLAY AT(5,1):"Move
shapes:   Key or Key":"
 LEFT-----J-------S  ":"
 RIGHT----L-------F"
20030 DISPLAY AT(9,1):"Rotat
e Shapes ANTI CLOCKWISEONLY
with keys K or D"
```

```
20040 DISPLAY AT(12,1):"Spac
e bar to drop shape and scor
e!":"Pause = Key ; or A"
20050 DISPLAY AT(16,1):"Leve
l increases every 5000":"":"
Start at level(1-9):"
```

```
20060 ACCEPT AT(18,21)VALIDA
TE(DIGIT)SIZE(1):E$ :: IF E$
="" THEN 20060 ELSE E=VAL(E$
)
20070 E=10-E
20080 SUBEND
```

FUNCTIONAL SPECIFICATIONS FOR THE 99/4 DISK PERIPHERIAL

CONSUMER GROUP LUBBOCK, TEXAS 79414

DATE: MARCH 28, 1983 VERSION 2.0

SUPPORTED FILE MANAGEMENT OPTIONS

The disk peripheral supports most of the options in the File Management Spec. for the 99/4 Home Computer.
The supported options include:

Sequential and Relative record (random access files)
Fixed and Variable length records
Internal and Display file types
Out, Input, Update, and Append access modes
Program Load and Save functions

The I/O routines supported by the disk peripheral are:

OPEN - Open an existing file for access. This routine must have the drive number or the disk name and the filename to open.
CLOSE - Close a file for access. The PAB can be released and the disk peripheral software deallocates some buffer area in VDP memory. Since the number of files that can be open at once is limited it is advised each file is closed as soon as it is no longer needed.

--------------------

DSKX.FILE-ID
DSK.VOLNAME.FILE-ID
X is the drive ID number (1-3), "volname" is a volume name ID and "file-id" is an individual file ID. Both "volname" and "file-id" can be strings of up to 10 characters long. Legal characters for these strings are all the ASCII characters, except the "." character and the space.
The first form of the file name specification shows the direct drive ID option. The user can specify either DSK1, DSK2, or DSK3 as drive numbers. Only the specified drive is searched for the given file-id.
The second form of the file name specification is the symbolic form. The disk drive is not explicitly assigned, but assigned through the volume name ("volname"). All drives are searched in sequence for the given volname, i.e. DSK1 first, then DSK2, then DSK3. The first drive with the given volname on its disk will be used for the file-id search. It is allowed to use two or more disks with the same volname in the system, however, if the specified file-id doesn't exist on the first drive with the given volname, the other disk drive(s) with the same volname will not be searched.
Whichever form is used, the file-id has to be unique for the selected drive, i.e. if a new file is created, the file-id used must differ from all other file-ids on that drive, or the existing file will be replaced by the new one, unless it is protected.
The file-id in the OPEN statement has to correspond to a data file. If the file was created by a SAVE command, an OPEN for that file will give an error, unless the file is opened for OUTPUT mode, in which case the program file will be replaced by the new data file.
------>-------->-------->----->

FILE ORGANIZATION Option

The two file organizations the user can specify are:

1. SEQUENTIAL - Access the file in sequential order, comparable to tape-access. The file may be accessed in any of the four I/O modes. Record type may be specified as FIXED or VARIABLE. File type may be specified as INTERNAL or DISPLAY.
2. RELATIVE - Access the file in random order. The open mode can be any of the available four modes, and the record type must be FIXED, and may be either INTERNAL or DISPLAY. Due to BASIC limitations, the combination RELATIVE and APPEND is not supported. This combination is trapped out as an error.

The default file organization is SEQUENTIAL.

Both the SEQUENTIAL and RELATIVE specifications can optionally be followed by an initial record allocation specification. This spec. indicates the number of records to be allocated initially. In case the record length has been specified as VARIABLE, the allocation will be made for maximum length records.
The number of records initially allocated has to be less than 32767, in order to stay within the record addressing range of a file management system.

----------------------------

RECORD-TYPE Option

The record-type option is used to specify the size of each record in the file. This size can be either FIXED, all records have the same length, or VARIABLE, with a maximum length optional. If the file organization specified is RELATIVE, the only legal type is FIXED, which is also the default for relative record files.
Both the FIXED and the VARIABLE options can be followed by an expression indicating the actual or maximum record length. Since the length is used to reserve buffer space in the BASIC interpreter, a user is advised to select the length as precisely as possible. Larger record lengths mean fewer variables can be used by BASIC.
The disk peripheral defaults the record lengths for both the FIXED and VARIABLE options to 80 characters.The default record-type for SEQUENTIAL files is VARIABLE; for RELATIVE files it is FIXED.
If a file is opened for any I/O mode other than OUTPUT, and it already exists, the record length, has to match the stored length. If no record length is given the DSR will default to the stored length.
The maximum record length for FIXED records is 255, and for VARIABLE length records it is 254.

--------------------------

FILE-TYPE Option

The file-type option can be used to specify the format of data to be stored in the file. There are two formats available:

1. DISPLAY - Stores data in readable format, i.e. as it would be printed on a printer. If the data has to be read back by the machine, this format is not recommended.
2. INTERNAL - Stores data in machine readable format. Since most files on the disk will be read by machine, this format is recommended. It relieves the user of storing separate data like quotes and commas in the file in order to make it suitable for an INPUT command. It avoids the overhead of converting the internal machine representation for the numbers and strings into a representation that is readable for humans and vice versa.

----> more --->

Again, if the file exists and the I/O mode is not OUTPUT, the specification has to match the value stored at file creation. BASIC uses DISPLAY as a default, which means that if data is stored in INTERNAL format, the user always has to indicate this in the OPEN command.

---------------------------

### FILE-LIFE Option

BASIC only recognized the PERMANENT option as a file-life specification. Since it is also the default it can be eliminated.

----------------------------

### CLOSE

The key word DELETE is optional with the CLOSE statement. In case DELETE is specified, the file is not only disconnected from the file number, but the disk space taken up by the file is released, and the file-id is erased from the disk's catalog. This means the file can no longer be accessed, not even with an OPEN statement (see DELETE statement).

A few examples of the CLOSE statements are:

```
CLOSE #240 - close the file associated with #240
CLOSE #240:DELETE - same as above, but also deletes the file
```

-------------------------

### PRINT STATEMENT

The PRINT statement can be used to write information out to a file that has been previously OPENed. The PRINT statement can only be used for files that have been opened for access in either OUTPUT, UPDATE, or APPEND mode. A PRINT to a VARIABLE record length file will always set a new EOF mark, causing data behind the current record to be lost.
The general form of the PRINT statement is:

```
PRINT #file-number[,REC record-number][:print-list]
```

For a detailed description of the PRINT statement, refer to the 99/4 BASIC Language User's Reference Guide.

----------------------------

### INPUT STATEMENT

The INPUT statement can be used to read information from an existing file. The INPUT statement can only be used for files that have been OPENed for access in either INPUT or UPDATE mode.

The general form of an INPUT statement is:

```
INPUT #file-number[,REC record-number]:variable-list
```

------------------------

### RESTORE STATEMENT

The RESTORE statement repositions an open file to its first record, or at a specific record if the file is opened for RELATIVE mode and the RESTORE contains a REC clause.

The general form for the RESTORE statement is:

```
RESTORE #file-number[,REC record-number]
```

### EOF FUNCTION

The EOF function can be used to test for the end of file during I/O operations. Three conditions are indicated by the EOF routine:

```
0 Not EOF (End of File)
1 Logical EOF (End of File)
-1 Physical EOM (End of Medium)
```

Physical EOM can only be detected if the device is at its physical end and the file is at its logical end.
The general form for the EOF function is:

```
EOF(file-number)
```

The EOF indication only has meaning in the case of sequential access to files, since for random access the next record to be read or written cannot be determined from the current one. Therefore, the EOF subroutine will assume that the next record to be read/written is the sequentially next record.
The logical EOF indicates that the next sequential read/write operation will attempt to access a record outside the current file. In general this indication will only be used for read operations.
Because of pending BASIC INPUT conditions, it is possible that the EOF subroutine indicates "EOF", even if the next INPUT statement will yield no EOF error, since it can read data from the current record. Something similar can happen if it indicates "no EOF" and the next INPUT statement reads more than one record. In this case the INPUT might be terminated with an error. To avoid this type of situation, the user is advised to use only non-pending INPUT statements (INPUT statements without a trailing comma), so that each record corresponds to one INPUT statement.
For random access to files, the EOF subroutine can only give meaningful results if the access is converted to "semi-sequential" access, i.e. if the record is positioned through a RESTORE statement and then sequentially accessed through any I/O statement without REC clause specification. After the RESTORE the EOF subroutine will indicate that the condition for the next record is (EOF, EOM or available), without issuing an I/O error.
Note that there is one EOM condition that cannot be detected by the EOF subroutine. This condition occurs when the datablocks on a disk become so scattered that not enough datablocks can be allocated for a file. In this case a PRINT operation will be aborted with an I/O error, even though there is enough space available on the disk, and the EOF function does not indicate an EOM condition.

---------------------

NOTE that the software file protection does not offer any protection against complete disk re-initialization. The only way to avoid file loss in that case is to "write protect" the disk itself by placing a tab over the notch on the right side of the disk. This will disallow any write operation to the disk, giving a hard error as soon as the disk is being accessed for write operations. Notice that this type of write protection is only intercepted on the actual write operation. The disk software will not disallow destructive access to the disk until the moment it actually tries to modify part of the disk.

---------------------

### FILES SUBPROGRAM

The default number of files that can be opened simultaneously is three. To modify this number, the FILES subprogram has been provided. The syntax for this subprogram is:

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

CALL FILES(x)
NEW
Where "x" is a number from 1-9, indicating the number of files that can be opened at once. Arithmetic expressions and variables are not allowed.
The NEW command following the FILES call has to be considered as part of the FILES call, since FILES will destroy some pointers used by the BASIC interpreter.

### WARNING

The usage of the FILES subprogram in a BASIC program is not allowed, and doing so will cause strange results. Likewise a call to FILES without a NEW command immediately following it may cause strange results, ranging from loss of program TO LOSS OF DATA ON DISKETTES. The only way to avoid this is to use the FILES subprogram only in the above defined manner!

The FILES subprogram will check only for the above defined syntax.

CALL FILES(2)*2 will execute the same as CALL FILES(2)

The disk has a standard overhead buffer of 534 bytes. Each open file adds 518 bytes to this buffer area for the disk. If this would leave the user with a buffer of less than 2K bytes as may occur in a 4K system, the files program will return with an INCORRECT STATEMENT error.
In case of a syntax error before the right parenthesis (")"), an INCORRECT STATEMENT will occur.
*************************************************************
READ - Read a logical record from an open file.
WRITE - Write a logical record to an open file.
RESTORE/REWIND - relocate the file read/write pointer to a given location in the file. For sequential files this can only be the beginning of the file, whereas for relative record files, the file read/write pointer can be relocated to any logical record in the file by giving the record number.
LOAD - Load a program file into VDP memory. The disk peripheral will check the correct file type before the program is loaded (see section 4.7).
SAVE - Save a program in VDP memory onto the named disk file. The disk peripheral does not check for legal BASIC memory images, so this routine, like the LOAD routine, can be used for transfering binary memory data to and from disk files. Note that the disk file is marked as a program file however, so that files created with a SAVE command can only be read with a LOAD command.
DELETE - Delete the indicated file from the given disk, delete frees up the space occupied by the file for future use.
SCRATCH RECORD - This function is not supported by the disk peripheral.
STATUS - Indicates current status of a file. This includes the logical and physical EOF flags and the protection flag.
---------------------------

### INTERFACE TO BASIC

This section will provide a general overview of how the disk peripheral presents itself to the BASIC user. For the BASIC related details the reader is referred to the Home Computer BASIC Language Specs.
---> ---> ---> more ---> --->

### OPEN STATEMENT

The BASIC OPEN statement allows the user to access files stored on accessory devices, such as the disk peripheral. It provides the link between a file name and a BASIC file number. Once the file has been OPENed.

The general form of the OPEN statement is:

OPEN #file number:"file name"[,option[,option[,...]]]

In which "option" can be any of the OPEN options available to the user. The user can select the following options:

File organization - SEQUENTIAL or RELATIVE
Open mode - INPUT, OUTPUT, APPEND, or UPDATE
Record type - FIXED or VARIABLE
File life - PERMANENT
========================

### FILE NAME SPECIFICATIONS

In order to indicate which drive and which file on it the user wants to access, he must specify a file name in the OPEN statement. This can be in either of two forms:
DSKx.file id or DSK.volname.file id
==========================================================
The actual number of Allocatable Units (AUs) allocated can be computed by using the following rules:

1. VARIABLE length records have an overhead of one byte per record plus one byte per AU.
2. Logical records never cross AU boundaries, i.e. an integer number of logical records has to fit in an AU.

A direct result of these rules is that the maximum length of VARIABLE length records is limited to 254 (2 less than the AU size).
Initial allocation of a file is done to avoid scattering of data blocks over a diskette. NOTE: Initial allocation does NOT change the End of File markers, i.e. if 100 records have been initially allocated, the file will still have its EOF set at record 0!!
The initial allocation is only used if a file is opened for OUTPUT mode or if a nonexisting file is opened for UPDATE or APPEND mode. It is ignored if the file is opened for any other case.

### OPEN-MODE OPERATION

BASIC accepts four access modes:

1. INPUT - Data in a file can only be read. The file has to exist before it can be read.
2. OUTPUT - Data can only be written to a file. A new file is created if one does not exist. If one of the same name exists it will be overwritten unless it is protected.
3. APPEND - Data can only be written at the end of the file. If the file does not exist already this mode is equivalent to OUTPUT. Due to the limitations of the console, this mode can only be used for VARIABLE length records.
----> continued --->

4. UPDATE - Data can be both written and read. If the file does not exist, it is created. Otherwise data in an existing file can be read and/or changed and new data can be added or old data can be deleted. UPDATE mode is generally used for files OPENed in RELATIVE mode, although SEQUENTIAL is permitted. VARIABLE length record files can be OPENed in UPDATE mode, however, once a new record is written, all the original data behind this record will be lost. This mechanism is mainly intended for use in intermediary files, i.e. first the data is written out, then it is read back without closing the data file.
Note that for UPDATE mode, it is never possible to decrease the size of a file. A re-write will only reset the EOF markers, without releasing the datablocks.

The default OPEN mode is UPDATE, i.e. the file can be both read and written.
.BP

```
OPEN #1:"DSK1.FILEA"
```

This statement will open a file called "FILEA" on the disk in drive #1 and its file reference number in BASIC is 250. The atrtributes assigned to this file are:

File-organization - SEQUENTIAL
Open-mode - UPDATE
Record-type - VARIABLE
File-type - DISPLAY
File-life - PERMANENT
Record length - If none existed before it will be 80 else it will be equal to the length of the file when it was created.

```
OPEN #24:"DSK.MASTER.TABLES",INPUT,RELATIVE,INTERNAL
```

Opens a file called "TABLES" on a disk called "MASTER". Drives will be searched in sequence till on is fopund with the disk call "MASTER", then it will be searched for a file called "TABLES". If it exsists it will be made accessable to BASIC otherwise you get an error. The specification for this file are:

File-organization - RELATIVE
Open-mode - INPUT
Record-type - FIXED
File-type - INTERNAL
File-life - PERMANENT
Record-length - is equal to the stored length for the file "TABLES".

```
OPEN #1: "DSK3.TESTDATA",OUTPUT,FIXED 40, INTERNAL,RELATIVE
```

Creates a random access file called "TESTDATA" on drive 3. If it exists already, it is overwritten with the new data. The attributes for this file are:

File-organization - RELATIVE
Open-mode - OUTPUT
Record-type - FIXED, 40 characters
File-type - INTERNAL
File-life - PERMANENT

```
OPEN #1:"DSK1.",INTERNAL,FIXED 38,INPUT
```

This command will open the catalog file for sequential input. For more information see section 5.

### CLOSE STATEMENT

The CLOSE statement closes the association between the BASIC file-number and the file. After the CLOSE statement is performed, BASIC can no longer access that file, unless it is OPENed again.

The general form of the CLOSE statement is:

```
CLOSE #file-number[:DELETE]
```

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

### DELETE STATEMENT

The DELETE statement may be used to remove files that are no longer needed from the disk. This will free up space allocated for the file.
The general form for the DELETE statement is:

```
DELETE "file-name"
```

The DELETE statement is a statement for which no previous OPEN is required. Therefore it is possible to DELETE a file which is still OPEN for access. If this happens, any future reference to the file, including a CLOSE, will give an error. An example of the described sequence may be:

```
100 OPEN #2:"DSK1.FILE",OUTPUT
110 PRINT #2:"HELLO"
120 DELETE "DSK1.FILE"
130 CLOSE #2
```

Here line 130 will give an error, since the file no longer exists at that point in the program.

-----------------------------

### 4.7 OLD Command

The OLD command allows for retrieval of previously stored programs from a disk. The program must have been stored with a SAVE command, since the disk software will not allow loading of a data file with the OLD command.
The general form for the OLD command is:

```
OLD file-name
```

Since OLD is a system command that cannot be used in a program, the file-name can be an unquoted string.

```
OLD DSK1.PROGRAM
```

Is perfectly legal.

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

### SAVE COMMAND

The SAVE command can be used to save the current program in the 99/4 onto a disk file, which then can be reloaded with an OLD command.
The general form for the SAVE command is:

```
SAVE file-name
```

!!!!!!!!!!!continued-->!!!!!!!!!

CATALOG FILE ACCESS FROM BASIC

The BASIC user can access a disk catalog like a read-only disk file. This disk file has no name and is of the INTERNAL, FIXED length type. An example of a CATALOG file OPEN is:

```
OPEN #1:"DSK1.",INPUT,INTERNAL,RELATIVE
```

Since BASIC will automatically default the record length to the current value, it is recommended that the user not specify this length. If he wishes it is 38. Every other record length will result in an error.
The CATALOG file acts like it is protected, as it will only allow INPUT access. An attempt to open the CATALOG file for any other mode will result in an error.
The data in the CATALOG file is written in the standard BASIC INTERNAL format. Every record in the file contains four items: one string and three numerics. The string indicates the name of the disk, containing up to 10 characters. The numeric items indicate the following:

1. Record-type - Always zero for this record.
2. Total number of AUs on the disk - for a standard 40-track disk this should be 358.
3. Total number of free AUs on the disk.

Record numbers 1-127 contain information about the corresponding file in the CATALOG. Non-existing files will give a null-string as the first item, and 0s for the remaining three items. Existing files will indicate the file name in the string item, and the following in the numeric items:

File-type - negative if file is protected.
1 DISPLAY/FIXED datafile
2 DISPLAY/VARIABLE datafile
3 INTERNAL/FIXED datafile
4 INTERNAL/VARIABLE datafile
5 Memory image file (e.g. BASIC program)

Number of AUs allocated by the file.
Number of bytes per record.
A type 5 file will always indicate a 0 in its third item, since the number of bytes per record has no meaning.

*********************************

FILE PROTECTION

Auser may select to protect any of the files on a disk. This can be done with the disk manager package.
The effect of the protected file is that the system disallows any type of destructive access to that file, the following actions are disabled.

SAVE to a protected file.
OPEN a protected file in a mode other than INPUT.

--------------------------

Error codes follow..... --->

I/O ERROR CODES

I/O errors detected by the disk peripheral software are always indicated by BASIC in the following format:

* I/O ERROR xy [IN 111]
The digits "xy" indicate the type of error that has occurred. The first digit (x) indicates the I/O routine in which the error occurred. The following I/O routine codes can be given:
0 Error in OPEN routine
1 Error in CLOSE routine
2 Error in READ routine
3 Error in WRITE routine
4 Error in RESTORE routine
5 Error in LOAD routine used during OLD
6 Error in SAVE routine
7 Error in DELETE routine
9 Error in STATUS routine used in EOF

The second digit (y) indicates the type of I/O error that has occurred. There are 8 different codes with the following meaning:

0 BAD DEVICE NAME - the device could not be found
1 DEVICE WRITE PROTECTED - unprotect the disk and try again
2 BAD OPEN ATTRIBUTE - one or more open options were illegal and didn't match the file characteristics.
3 ILLEGAL OPERATION - should not be generated by BASIC for the disk peripheral. Indicates usage of non-existing I/O code.
4 OUT OF SPACE - a physical end of the file was reached, there was insufficient space on the disk to complete the operation.
5 ATTEMPT TO READ PAST EOF
6 DEVICE ERROR - a hard or soft device error was detected. This may occur if the disk was not initialized or was damaged, the system was powered down during disk writes, the unit did not respond, etc.
7 FILE ERROR - the indicated file or volume doesn't exist; the file type doesn't match the access code (program file versus data file).
======THE END=======

Note: Above disk text may not be complete and may not be in the original order of presentation. This document has not been presented by TI!

# HELP! MY "CALL SAY" DOESN'T

by Guy S. Romano

One of the most common calls we receive at Library Services has to deal with apparent problems with the Speech Synthesizer which in reality are not "problems" but rather a lack of understanding of just how the fantastic speech capabilities of the TI Computer work. To clarify the point let us draw some very broad and imprecise analogies. Let us call the console the "brain" of the whole system. Then by extension the Speech Synthesizer can be called the larynx or "voice-box" of our computer. Our own voice-boxes will not work alone; they need something to drive them to vibrate and create sound. So too with the Speech Synthesizer unit. It also requires something to cause it to create sound. So too with the Speech Synthesizer unit. It also requires something to cause it to create sound. Back then when the 99/4A was the 99/4 and little existed for the computer, this "driver" was simple since no choice was possible. The force that made speech possible was a Command Module called the Speech Editor. With it one could use the Speech Synthesizer to create a very realistic sounding voice in programs and use speech in conjunction with ALL the other capabilities of the TI Computer. (To the best of our knowledge, the 99/4A is still the only small computer that allows the combination of speech, graphics, and music all in the same program.)

CONTINUED→

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

## HENON MAPPING

Here is a fractal program for TRITON SUPER XB module owners...

```
100 REM TRITON XB
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 A from -4 to +4
270 COSA=COS(A):: SINA=SIN(A
) !
{-.1 to +.8 is full map }
{ step .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.
20
290 FOR Y=-.1 TO 0.8 STEP .2
0
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*COSA-(SY-SX*SX)*SI
NA
350 SY=SX*SINA+(SY-SX*SX)*CO
SA
360 SX=XX
370 PLTX=SX*90+91 :: PLTY=SY
*90+91
380 CALL LINK("MOVE",PLTX,PL
TY)
390 CALL LINK("DRAW",PLTX,PL
TY)
```

CONTINUED→

With the advent of the 99/4A other wonderful Command Modules came on the scene; Extended Basic and Terminal Emulator II among others. Texas Instruments gave us a great bargain in Extended Basic, they took all the speech capabilities of the old Speech Editor and incorporated them into the Extended Basic module. Now we had two "drivers" for the Speech Synthesizer and they both worked the same way, but they still were limited to a rather small vocabulary. When TI came out with the Terminal Emulator II which was primarily designed for use with a modem for computer hookups by phone, they threw in another present, a different and marvelous speech capability totally unlike what came before. The TE II now allowed for UNLIMITED speech capability with none of the limitations or restrictions of Speech Editor or Extended Basic. Now "we" could say anything and even change intonation patterns and voice pitch. TI, the forerunner of computer speech, gave us a really powerful and versatile unit neatly contained in one little module for an almost ridiculously low price. As is true, however, as in all other facets of life, with choices comes some inequity. The speech functions of Speech Editor/Extended Basic and Terminal Emulator II are not compatible with one another and their functions cannot be used interchangeably from program to program. But this seeming inequity or incompatibility is only superficial since with a little work on our part, we can change a program written for Speech Editor to one for TE II quite easily.

CONTINUED→

```
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
```

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.

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
INNER 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 28
0
230 XX=SX*COSA-(SY-SX*SX)*SI
NA
240 SY=SX*SINA+(SY-SX*SX)*CO
SA :: SX=XX
250 PLTX=SX*90+91 :: PLTY=SY
*90+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
```

END

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

The Speech Editor/Extended Basic type of speech "driver" uses statements like CALL SPGET(,,,) (not often used) and CALL SAY(XXX). Nothing else is required except that you MUST use only the resident vocabulary. Words may be made to seem more natural sounding by hooking words together with plus signs; i.e. "HOW+ARE+YOU" and then certain phrases were created that are in common usage and these were indicated by the use of the number sign, "#" as in "#TRY AGAIN#".

Terminal Emulator II functions as if it were opening a file (the same thing you do when you "SAVE" a program on cassette, etc. As you can begin to see it is quite simple to convert "CALL SAY" to TE II. Where the older format called "CALL SAY(HOW+ARE+YOU")" TE II used the form of "PRINT #1:"HOW ARE YOU"

Let us say that you have bought a program that needs the Speech Synthesizer but when you RUN it you get a message that says BAD NAME IN XXXX (where XXXX is the line number). If you immediately look at the line XXXX it will probably have a "CALL SAY" in it. So here is what to do:

OLD PROGRAM LINE

XXXX CALL SAY ("THAT+IS VERY+GOOD")

NEW LINE FOR TE II

XXXX PRINT #1:"THAT IS VERY GOOD"

What you are removing is the CALL SAY and the parentheses plus any of the linkers (plus signs of "#"'s) and replacing them with TE II you are using a file structure you have to "open" that file before your computer will recognize it. Therefore, somewhere at the beginning of the program you must insert a line that reads:

XXXX OPEN #1:"SPEECH", OUTPUT

CONTINUED →

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 - the 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
140 REM
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 rotati
on 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
290 CALL POINT(0,X*48+127,Y*
48+127)
```

CONTINUED →

If you want to be thorough you can then add a line at the end of the program which reads:

XXXX CLOSE #1

although this is not necessary. When you make these conversions keep in mind that you can now change the message in quotes to ANYTHING you want since there is no limit on the vocabulary with TE II. "SAVE" this converted program on a new tape for use later.

"OK", you may say, "but even though I have a Speech Synthesizer, I do not have a TE II and am not planning to use the program NOW!!!" Your wish is granted quite simply. Go through the program and look for lines with the "CALL SAY" or even the "PRINT #1:" if the program is written for TE II. Get into EDIT mode — type the line number, then press "FCTN" and "X" simultaneously. The cursor will be flashing over the first letter in the line (anything after the line number). Then type "REM" and press enter. Now when the computer looks at that line and it sees that "REM" it will ignore it and go on to the next line. Later when you get a TE II, etc. You can remove the REM's and make the necessary conversions and then take full advantage of the wonderful world of SPEECH that Texas Instruments has given us.

HAVE A NICE CHAT WITH YOUR COMPUTER!

END

FIRST PUBLISHED MAY 1983
in ENTHUSUAST 99 Vol 1 No 1

```
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(MINX,
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
```

END

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)
```

---> continued --->

```
430 NEXT N :: NEXT K :: GOSUB 40
440 REM 3D PLOT.
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
```

----> CONTINUED ---->)

```
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
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.

MIKE POSKITT reports a source for Alphacom Thermal paper as E.E.C.Ltd., 18-21, Misbowne House, Chiltern Hill, Chalfont St. Peter,Bucks..(0753 888866). 1 roll £3.50, 5 roll pack £12.95, plus P&P. He also reports the successful use of a 40/80 switchable half-height double sided Mitsubishi disk drive as 1&2, as he already has an internal drive as no.0. Mike would like to draw your attention to TIs educational software as shown in the appended table:

| | | |
|---|---|---|
| Early Learnin; | Preschool (2-5 Years) | Early Learning Fun<br>Early Logo Learning Fun |
| Reading | Early Elementary (5-7 Years) | Early Reading<br>Reading Fun |
| Spelling | | Hangman |
| Math | | Number Magic<br>Addition/Subtraction I<br>Addition/Subtraction II<br>Numeration I |
| Art | | Video Graphs |
| Reading | Middle Elementary (8-9 Years) | Beginning Grammar<br>Reading On<br>Reading Roundup |
| Spelling | | Scholastic Spelling Levels 3 & 4 |
| Math | | Multiplication I<br>Meteor Multiplication<br>Division I<br>Alligator Mix<br>Minus Mission<br>Alien Addition |
| Reading | Late Elementary (10-12 Years) | Reading Flight<br>Reading Rally |
| Spelling | | Scholastic Spelling Levels 5 & 6 |
| Math | | Demolition Division<br>Dragon Mix<br>Numeration II |
| Music | | Music Skills Trainer<br>Computer Music Box |
| Math | Early Elementary to Junior High (5-14 Years) | Addison-Wesley<br>Computer Math Games II, III, IV, VI |
| | | Milliken Math Series:<br>Addition, Subtraction, Multiplication, Division, Integers, Fractions, Decimals, Percents, Laws of Arithmetic, Equations, Measurement Formulas |
| Computer Programming | | TI LOGO II |
| Logic | | Video Chess |
| Typing | | Touch Typing Tutor |
| Physical Fitness | | Physical Fitness |
| Business | Junior High to Adult | Market Simulation (Disk) |
| Computer Programming | | Teach Yourself BASIC<br>Beginner's BASIC Tutor<br>Teach Yourself Extended BASIC |

```
 X   X   BBBB      # 14
  X X    B   B
   X     BBBB      By
  X X    B   B     Jim
 X   X   BBBB      Swedlow
```

[This article originally appeared in the User Group of Orange County, California ROM][Edited by S Shaw]

PRODUCT REVIEW: TI-WRITER INSTRUCTIONS AND HINTS

This is crammed with ideas, hints, suggestions and encouragement. It is 101 sectors long!

You should print it thru both the Text Formatter and the Text Editor. That way you can see how things were done. Like, for example, printing in compressed type with a line width of 130 characters.

I would only quibble with one suggestion: that you start your file on line 0001. I leave that line blank as it is lost if you do a Recover Edit.

While it is aimed at the beginner, I would rate this as quite valuable to any TI-Writer user. This program should be in our library by the time this article is published.

This program is released as FREEWARE. If you find it useful, send a few dollars to the author, Dick Altman. He earned it as this must have taken quite a bit of time and effort to write.

(Available from your disk library as TI WRITER TUTOR)

DISK*LABEL

This month's program is a revision of DISK*LABEL. The printer commands are now at the beginning of the program with full annotation (see lines 180 to 230). You can easily make it work on any printer that prints superscripts (half height letters).

I have made a number of changes in the program. It should be harder to crash now. Also, it will always read the first 23 files on a disk (even if there are not that many files on the disk). This avoids a problem some protection schemes cause. For example, this version works with Advanced Diagnostics.

I recommend that you NOT put the label directly on the disk. I have tried two different types, and both tend to peel. Very bad news if one comes off inside your drive!! Put the labels you got with your disks on the disk and these on the sleeve.

Enjoy.

```
100 ! DISK*LABEL
110 ! VERSION XB.2.1
120 ! BY JIM SWEDLOW
130 ! BASED ON A PROGRAM BY
      P.C.B. AND W.A.R.
140 ! 09 NOV 85
150 !
160 B,@=1 :: CALL CLEAR :: D
IM F$(144),T$(144),Y$(4):: F
OR I=0 TO 14 :: CALL COLOR(I
,16,@):: NEXT I :: CALL SCRE
EN(5)
170 GOTO 180 :: A,C,D,J,C$,D
$,E$,DS$,CN$,SS$,LF$,IN$ ::
!@P-
```

```
180 E$=CHR$(27)! ESCAPE
190 DS$=E$&"G" !
    DOUBLE STRIKE
200 CN$=CHR$(15)! CONDENSED
210 SS$=E$&"S"&CHR$(0)!
    SUPERSCRIPT
220 LF$=E$&"3"&CHR$(12)!
    CHANGE LINE FEED TO
    12/144 INCH
230 IN$=E$&"@" !
    INITALIZE PRINTER
240 Y$(@)="DF" :: Y$(2)="DV"
 :: Y$(3)="IF" :: Y$(4)="IV"
 :: D$="DSK1." :: OPEN #2:"P
IO"
```

```
250 DISPLAY AT(7,10):"DISK*L
ABEL": : :"Check the positio
n of the   labels before sta
rting."
260 DISPLAY AT(14,6):"Labels
/Disk:  1":"       Drive:
DSK1":"       Comment:"
270 DISPLAY AT(20,@):"ENTER
<P>rint <C>hange or":"
<Q>uit          P"
280 ACCEPT AT(21,20)VALIDATE
("CQPcpq")SIZE(-@)BEEP:E$ ::
 IF E$="" THEN 280 ELSE I=AS
C(E$):: E$=CHR$(I+32*(I>81))
290 IF E$="Q" THEN CALL CLEA
R :: CLOSE #2 :: STOP ELSE I
F E$="P" THEN 330 ELSE DISPL
AY AT(20,@): : :
300 ACCEPT AT(14,20)SIZE(-2)
VALIDATE(DIGIT)BEEP:E$ :: IF
 E$="" THEN 300 ELSE B=MAX(V
AL(E$),@)
310 ACCEPT AT(15,20)SIZE(-@)
VALIDATE("12")BEEP:E$ :: IF
E$="" THEN 310 ELSE D$="DSK"
&E$&"."
320 ACCEPT AT(17,6)BEEP:C$ :
: GOTO 270
330 C=0 :: DISPLAY AT(20,@):
"Initializing": : :: OPEN #@
:D$,INPUT ,RELATIVE,INTERNAL
 :: INPUT #@:F$(C),I,I,I ::
T$(C)="FREE "&STR$(I)
340 DISPLAY AT(22,@):F$(C);"
  ";T$(C):: IF C=127 THEN 37
0 ELSE INPUT #@:F$(C+@),I,J,
J
350 IF F$(C+@)="" THEN IF C>
23 THEN 370 ELSE C=C+@ :: GO
TO 340
360 I=ABS(I):: C=C+@ :: IF I
=5 THEN T$(C)="Prog" :: GOTO
 340 ELSE T$(C)=Y$(I)&STR$(J
):: GOTO 340
370 CLOSE #@ :: DISPLAY AT(2
0,@):"Printing": : : :: FOR
A=@ TO B :: J=0 :: D=8
380 E$="   " :: PRINT #2:DS$
;CN$;F$(0);E$;C$;E$;T$(0);SS
$;LF$: :
390 FOR I=J+@ TO J+D :: PRIN
T #2:F$(I);TAB(12);T$(I);TAB
(18);F$(I+D);TAB(29);T$(I+D)
;TAB(35);F$(I+2*D);TAB(46);T
$(I+2*D):: NEXT I
400 J=J+24-6*(D=10):: IF C>J
 THEN D=10 :: PRINT #2: : :
:: GOTO 390 ELSE PRINT #2:IN
$
410 NEXT A :: FOR A=@ TO C :
: T$(A),F$(A)="" :: NEXT A :
: GOTO 270
```

++++++++++++++++++++++++++++++

------------------------------

----------------

```
100 REM AUTO GENERATION OF A
PICTURE SEQUENCE FOR COMIC
SHOW V 4.0
110 RAD@=26
120 REM FOR JBM103
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=MYLAT*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)
470IF Y>TWOPI THEN Y=Y-TWOPI
 :: 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,O
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 ram disk 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"
```

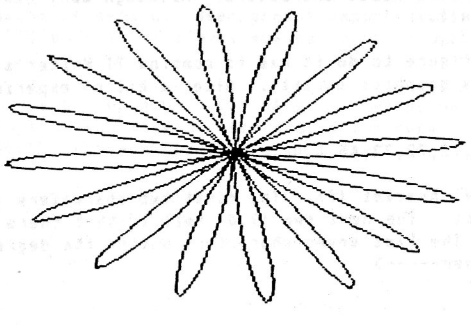
The catch is if the program bombs and you are returned to GRAPHICS(1) mode, the screen is blank! You must type NEW (and lose the program) to see text mode again!

Having generated a set of pictures you can then animate them using COMIC SHOW Vn 4.0 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 DSK2.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 DSK2.P1_P
GO
```

You end up with a memory image machine code program which will run from Editor Assembler Option 5 or equivalent.

Stephen



## TI BITS * Number 13 & 15
By Jim Swedlow

[This article originally appeared in the User Group of Orange County, California ROM] [Edited by S Shaw]

### TI WRITER TIP

Find String (FS) is a powerful tool for finding something in a document. Just hit FCTN 8 and then enter FS. Your TI Writer gives you this prompt:

```
    FIND enter /string/ :
```

You enter your string and use the slash as limiters. If you want to find the word "John", you would enter /John/. If you wanted to find John only when it is used as the last word in a sentence, you would enter /John./.

Should the "John" you find not be the one you wanted, you would go back to command mode and enter FS again. You will find /John./ still there. You just press enter and the search resumes.

Lets say, however, that now you want to find the word "Mo". But /John./ is on your screen. You could delete /John./. You could type in Mo but then you would have this:

```
    /Mo/n./
```

Need you worry about the text after the second slash? No. Your TI Writer only searches for the information between the first and second slash. It ignores everything to the right of the second slash.

You will have a problem with that if you use Replace String, but that is another story.

### TRICK QUESTION

Answer to the last trick question: How many birthdays does the average man have? One -- you celebrate it many times but you are born once.

### MAKING A DEGREE MARK IN TI WRITER

An owner in Huntsville Texas wrote me and asked if I know how to make TI Writer type a degree sign on a TI Impact Printer (it is a Epson MX80). A degree mark is not one of the standard ASCII characters. Although many newer printers can print it, the MX80 can't.

The only way I could figure to do it was to combine TI Writer's transliterate command and the MX80's graphics ability. After a bit of experimentation, I hit on this:

```
 .TL 91:27,76,7,0,48,72,72,72,48,0,0
```

This redefines the left bracket ([). The first two characters (27,76) tell the MX80 to invoke graphics. The next two (7,0) tell it that there will be seven graphics characters. The last seven characters define the degree mark.

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

This is not a perfect solution as, if you right justify, the right margin will be a bit uneven. It should work, however, on most Epson and compatible printers.

### SOME THOUGHTS ON WORD PROCESSORS

Of late I have occasion to use a number of word processors on other machines. I learned word processing on TI Writer and I wanted to see how the 4A stood up.

TI Writer is limited by the 4A's design. Eighty columns and a full keyboard make text management (warning: buzz phrase alert!!) much easier. Otherwise, TI Writer fairs well.

Just about anything you can do with the big name packages, you can do with TI Writer. Sometimes it is a bit harder, but it can be done. TI Writer is a powerful and flexible tool. It has some abilities, like transliterate, that are superior to other word processors.

The others are slicker because they have much more memory available. They can do things with one or two key strokes that take five or ten with TI Writer - but they can be done on the 4A.

If you are writing a book, it might be worth the cost to move up. [I wrote MY book ENTIRELY with TI Writer and had no difficulty or trouble AT ALL, shifting paragraphs and sentences around all over the place. Stephen Shaw.] But for correspondance, writing this column and similar jobs, TI Writer can do anything you need it to do. And that's a fact.

### BLACK FRIDAY PLUS FIVE

It was five years ago that TI announced that they were droping the 4A. October 28, 1983. A date that changed everything for 4A owners.

We moved from the main stream of computing to a cul-de-sac. Software and hardware became scarcer and scarcer. Retailers dropped from many to only a few.

And yet a cul-de-sac is not a bad place to live. In fact, they are preferred. Ours turned out to be pretty good. Five years later, exciting software continues to appear. This has been the year of graphics applications, with many innovative programs comming out.

There are some signs of strain. User groups report declining membership and money problems. TI owners are slowly moving to other machines (often with three letters).

The end, however, is not upon us. Our 4A still has strong support from retailers, developers, publishers, user groups and owners. I expect to be writing on the sixth anniversitie of black Friday.

FOUR-A/TALK Random ramblings about things TI. by Bill Gaskill
May 1989

DISCOVERIES:

- Art Green, the Canadian assembly language wizard who wrote the Macro Assembler programs, has released V4.3 of his TI-Writer upgrade. To be real honest with you, I didn't even know that there was a V1.0! None the less, Art has done as neat a job with TIW as he did with Macro Assembler, except that the TI-Writer upgrade is better documented. Or maybe it's that I understand TI-Writer better than I do assembly language. Whatever. Art's TIWV4.3 upgrade is a FairWare offering that is available at yesterday's price of $10 (suggested donation). A copy may be ordered from your user group disl librarian.

But wait! I'm going to tell you WHY you should send the $10 to Art and get a copy.

1. Like the Mike Ballman, Ed Jones and John Johnson TIW rewrite for the Horizon Ram Disk, Paolo Bagnaresi's BA-Writer, Tom Knight's TK-Writer and the McGovern's FunnelWeb system, the RAG version of TI-Writer has its own loader, so you don't need the TI-Writer module. The RAG loader lists A-Editor, B-Formatter or C-Utility.

2. Unlike the Ballman, Jones and Johnson collaboration, but similar to the McGovern's FunnelWeb system, the RAG version allows you to "install" the program to your system. That means that you can configure the program with your defaults for the;

-printer name.
-screen colors.
-tab settings.
-word wrap on/off.
-line number display on/off.
-defined character set.

3. The Show Directory function shows a catalog display much like the original one in the TIW module, which is a vast improvement over the one used in the BJ and J version.

4. The RAG version has several new commands such as QQ for an immediate quit without further prompting, Ctrl comma to go to the top of the file, Ctrl period to go to the bottom of the file, a PC (printer control) command has been added that allows control codes to be sent directly to the printer without changing the line count. It also allows you to setup a printer without having to use transliterates.

5. You can also define your own underline character, boldface character, required space character and mailing list control character. Art has also added a Conditional Page break option that tells the formatter program to do a page eject if less than a certain number of lines are left on the current page.

6. Perhaps one of the neatest features is the Chain File option that allows multiple floppy disks to be used in the formatter. This means that you can have a file that is hundreds of pages long, on several floppies, and still have the file printed as a single document. The CF option causes TIW to prompt you to insert the next disk and then press <ENTER>. Once that is done, processing resumes. Wow!

7. The formatter program is faster and more compact. In fact, it is only one file in the RAG version instead of two as it is in the original TI-Writer, the BJ and J version, BA-Writer, FunnelWeb etc.

8. Loaders for Extended Basic, Mini-Memory, the original TI-Writer cartridge and the SuperCart are included.

Many other "little" nicities have been added to make the RAG SOFTWARE version of TI-Writer my word processor of choice. It loads and works marvelously from my Horizon Ram Disk, though like all other TIW clones I have, it will not catalog my hard drive. It resists the loss of characters on word wrap better than anything I have seen to date, and it also scrolls text or windows the screen faster than anything I have seen to date, regardless of the size of the file.

I haven't mentioned much about the formatter improvements, but I will say that there are several. I also haven't mentioned the speed with which the program operates in general. Cursor speed is NOTICEABLY quicker as are block operations such as COPY and MOVE. Overall, the program is "smooth". Do yourself a favor and look into the Art Green TI-Writer V4.3. I know that you think a word processor is a word processor and all TIW clones are the same. NOT SO in this case.

NEWS:

- Andi Wise, editor for the newsletter at the Eugene, Oregon 99/4A Users Group, Box 11313 Eugene, Or. 97440, has compiled the most complete list of 99/4A Users Groups I have seen to date. There are over 500 listings in the data base, both past and present, U.S. and foreign. She really has gone to alot of effort to provide us with this much needed resource. SouthWest 99er BJ Mathis, who also has an excellent UG data base, contributed as an information source to the project. Andi compiled the data base in Mark Beck's Creative Filing System.

Not intending to compete with Andi, I have adapted her data to TI-Base so that TI-Base owners could also have access to the information, in a command file programmed environment. The TI-Base version is available in your club's library. Andi is asking a paltry $5.00 ShareWare fee for her file in CFS format. If you procure the TI-Base version, I ask that you also send her $5.00. Please do not send any money to me for the TI-Base version. Andi did the work and deserves the financial support for her Users Group.

- In case you haven't heard, Asgard's Chris Bobbitt is stirring things up with an offer to support the development of a prototype of the "Next Generation Computer" for the TI Community. From what I have read, it appears that he has lost faith in the Geneve or in Myarc, or has gotten tired of waiting for the Geneve to be a complete machine. It will be interesting to see what comes of it.

- Texaments have released MICROdex for TI-Base. All you TI-Base owners now have a chance to see the first third-party application available for your TI-Base data manager. The MICROdex libraries are available for $14.95 for MICROdex I and $9.95 for MICROdex II or $22.95 for both. Please include $8.00 S/H (UK). MICROdex is available from;

Texaments
53 Center Street
Patchogue, New York 11772
516-475-3480

- Back toAsgard again. Pro Page 99 by Ed Johnson. According to Chris Bobbitt, Asgard president, Pro Page will let you compose a full 8 1/2" X 11" page at once, with up to 28 pictures of any size on the page, and they may be placed anywhere you want. Also, both large and small type fonts [two FIXED sizes] for text will be supported as well as line drawing. Utilities to convert TI-Artist fonts and instances into Pro Page format will be included, with other conversion utilities planned for Picasso to Pro Page 99 format also. Wow! The newsletter editors out there ought to have a field day with this product [maximum 66 characters across the page!!!] Of course it does column layout of text files too, as well as importing and exporting of TI-Writer DV/80 files.

TI-99/4A REFERENCE LISTS:

I discovered Mike Wright's TI-99/4A Reference Lists while at the Fest-West in San Diego. Because I had to leave Sunday morning, and didn't get to spend the whole weekend there, I just barely noticed Mike's product in the Genial ComputerWare booth in a last minute sweep of the Fest before leaving. I bring the topic up again, after having read and re-read the lists, because I am convinced that they are the most complete, and the most professionally presented resource of their kind available to the 99/4A community. At $5.00 plus a couple of dollars to cover the cost of shipping and handling, they are a bargain. There must be hundreds of hours of research invested in them to come with the 40 plus pages of laser printed information, all of which has been verified by actual product. The list also contains the most complete and accurate description of books for the 99/4A I have seen since Barry Traver's list. If you are a 99/4A enthusiast, you WILL WANT the TI-99/4A Reference Lists by Mike Wright, 45 Centerville Drive, Salem, New Hampshire 03079. Honest!

TRIVIA:

Did you know that...

-The Peripheral Expansion Box project ordered by Don Bynum, designed to do away with the cumbersome chained peripherals setup of the 99/4, was officially completed in January 1982?

-In December 1983 Louisville, Colorado 99er Jim Robinson tried to start the International 99/4A Users Group complete with a bi-monthly newsletter named 4A Forum? I never heard of it again, but that doesn't mean it didn't succeed. Has anyone else heard of it or been a member?

-AtariSoft once listed Robotron: 2084, Stargate and Super Storm in their advertisements as being available for the 99/4A? They certainly showed us a lot of support after the "bailout" by TI, but I have yet to run across those titles in a 99/4A retailer's catalog or anywhere else. Mike Wright's TI-99/4A Reference Lists don't show them either.

-Charles LaFara's International 99/4 Users Group published only seven issues of Enthusiast 99, despite being in existence from September 1980 until May 1985? The magazine issues were May, July, September and November 1983, and January, March and a May/June 1984 issue.

-While most of us are familiar with four of the product designators used with the 99/4A line of computers, TI actually had seven of them? PHA-Accessories, PHD-Diskettes, PHL-Libraries, PHM-Modules, PHP-Peripherals, PHT-Cassette tapes and PHV-Value packs.

-Bill Bies, author of the Zaxxon clone "Arcturus", also wrote a Centipede clone named "Arthropod"? Wonder what Bill is doing these days? Sure would like to see him back amongst the active 99ers.

-The hottest 99/4A joystick today is the Epyx 500XJ. [UK NAME is KONIX SP EDKING- now updated to KONIX NAVIGATOR]. TexComp's Jerry Price advises that it has the most advanced design of any joystick available. It ought to really make MunchMan turn those corners. I bought one but don't have it yet.

-Cities named DANVILLE out number all other community names in the number that have 99/4A Users Groups. Yep! There is a joke out here in the west that no matter what state you go to west of the Mississippi River, you will find a body of water named Beaver Creek. It must be the same kind of thing for towns named Danville. There are four Danvilles, the Danville 99ers in Kentucky, the East Central Illinois Users Group in Illinois, the Southside 99/4A Computer UG in Virginia and the Susquehanna Valley 99ers in Pennsylvania. The next most common community name seems to be Springfield, with groups in Illinois, Missouri and Virginia.

-Aside from the Danvilles and Springfields, there are some other interesting community names where 99/4A User Groups can be found. How about Red Deer, Whitefish, Horseheads or Papatoetoe for starters?

Until next time...

And an ingenious tinygram version of Wheel of Fortune, in the West Penn newsletter.

```
1 ! *** FORTUNE OF WHEELS **
  *        A TINYGRAM       *
  * by Mike & Ed Machonis*
  *************************
2 CALL CLEAR :: INPUT "ENTER
 THE MYSTERY PHRASE    ":M$
:: CALL CLEAR :: L=LEN(M$)
3 D$=RPT$(CHR$(30),L):: FOR
J=1 TO L :: IF SEG$(M$,J,1)<
>" " THEN 4 ELSE D$=SEG$(D$,
1,J-1)&" "&SEG$(D$,J+1,L)
4 NEXT J :: PRINT D$
5 T=T+1 :: PRINT :"TRY No.";
T:;: :: INPUT "TYPE LETTER O
R ENTIRE PHRASE":A$ :: IF LE
N(A$)>1 AND LEN(A$)<L THEN 5
6 W=L+1-T :: IF A$=M$ THEN 9
7 FOR J=1 TO L :: IF SEG$(M$
,J,1)=A$ THEN D$=SEG$(D$,1,J
-1)&A$&SEG$(D$,J+1,L)ELSE 8
8 NEXT J :: PRINT :D$ :: GOT
O 5
9 FOR J=1 TO M :: CALL SOUND
(200+J*10,330+40*J,0):: NEXT
 J :: PRINT :"YOU WIN ";STR$
(W);",000 WHEELS!":;: :: INP
UT "PRESS ENTER TO PLAY AGAI
N":G$ :: T=0 :: GOTO 2
 =========   ============
```

# G

## G GRAPHICS

The G language is really quite interesting, especially in its remarkably clever error handling, which makes it possible to do things which would be very difficult in any other language we have. G is in the Disk Library and requires 32k and ExBas. It operates from DV80 command files, and below you will find three of these to try out. They originate from the magazine of the Dutch user group:

```
REM APESOFT DEMO
:START
I=0
N=30
FOR Z=1 TO 4 N=N+28
SET N N
FOR D=5 TO 50 STEP 5
DRAW D
I=I+90 ANGLE I
NEXT D
NEXT Z
FOR I=1 TO 700 NEXT I CLS
:CIRCLE
FOR R=2 TO 42 STEP 2
ARC 64 60 R R 0 360
NEXT R
FOR I=1 TO 500 NEXT I CLS
:PYRAMD
ANGLE 45
SET 100 100
FOR A=1 TO 36 STEP 2
BOX 100+A 100+A A A
BOX 100-A 100-A A -A
NEXT A
FOR I=1 TO 700 NEXT I CLS
:OKTAGN
D=2 Y=0 Z=0
FOR S=28 TO 108 STEP 4
SET S 42
Y=Y+90 ANGLE Y
D=D+2
FOR I=1 TO 8
Z=Z+45 ANGLE Z
DRAW D
NEXT I
NEXT S
FOR I=1 TO 500 NEXT I CLS
```

continued↗

```
:ZYLIND
FOR Y=120 TO 40 STEP -3
 ARC 100 Y 48 24 0 361
NEXT Y
FOR R=48 TO 2 STEP -2
 ARC 100 Y R R/2 0 360
NEXT R
FOR I=1 TO 500 NEXT I CLS
:LEAF
FOR P=0 TO 90 STEP 5
 SET 100 100
ANGLE P
DRAW P
NEXT P
FOR I=1 TO 800 NEXT I CLS
:THREAD
D=2 Z=125 T=128
FOR S=1 TO 126 STEP 5
Z=Z-5
T=T-D
SET S 120 TO T Z
NEXT S
Z=125 T=1
FOR S=126 TO 1 STEP -5
Z=Z-5 T=T+D
SET S 120 TO T Z
NEXT S
FOR I=1 TO 800 NEXT I CLS
GOTO :START
====================
```

```
REM CIRCLE 2
SCREEN 3 COLOR 1
FOR I=0 TO 4
FOR X=1 TO 170 STEP 5
ARC 128 96 X X 0 360 NEXT X
FOR I=1 TO 500 NEXT I
FOR I=1 TO 500 NEXT I CLS
REM CIRCLE 3
FOR N=0 TO 200 STEP 4
ARC N N N N 0 360
NEXT N
FOR I=1 TO 500 NEXT I CLS
REM KLEURLIJN
FOR I=1 TO 25
RND 1 Z RND 13 A
A=A+2 RND 200 X
COLOR A SET X X TO X+X X+X
NEXT I
FOR I=1 TO 500 NEXT I CLS
REM LIJNFIGUUR
R=3 S=3 P=12 Q=12
FOR A=1 TO 241 P=P+R Q=Q+S
SET 128 96 TO P Q
IF P>249 THEN R=-3
IF Q>190 THEN S=-3
IF P<6 THEN R=3
IF Q<10 THEN S=3
NEXT A
FOR I=1 TO 500 NEXT I CLS
```

CONTINUED →

```
REM LINES
FOR N=1 TO 9
FOR I=1 TO 25
RND 14 C RND 191 Y RND 250 X
COLOR C
SET X Y TO Y 65
NEXT I
CLS
NEXT N
FOR I=1 TO 500 NEXT I CLS
REM SATURN
COLOR 2
FOR I=1 TO 30 STEP 4
 ARC 125 84 I 25 0 360
NEXT I
FOR I=8 TO 20 STEP 7
 ARC 125 84 40 I 45 317
NEXT I
FOR I=1 TO 192 STEP 8
SET 0 0 TO I 192
SET 255 0 TO I+62 192
NEXT I
FOR I=1 TO 210 STEP 12
SET 0 0 TO I 56
SET 255 0 TO I+42 56
NEXT I
FOR I=1 TO 500 NEXT I CLS
REM SCHILLEN
RND 10 P P=P+10
FOR I=P TO 106 STEP 10
RND 15 K RND 15 L C=C+1
COLOR C
 ARC 127 106 I I 0 360
ARC 127 106 I-10 I-10 0 360
NEXT I
FOR I=1 TO 500 NEXT I CLS
STOP
----------
```

```
:START
REM KLOK
ARC 43 60 38 38 0 360
ARC 120 60 38 38 0 360
ARC 197 60 38 38 0 360
I=270
FOR N=1 TO 75
I=I+5
IF I=360 THEN I=0
TRACE 1
SET 43 60 ANGLE I DRAW 36
TRACE 0
SET 43 60 ANGLE I DRAW 36
TRACE 1
SET 120 60 ANGLE I DRAW 36
TRACE 0
SET 120 60 ANGLE I DRAW 36
TRACE 1
SET 197 60 ANGLE I DRAW 36
TRACE 0
SET 197 60 ANGLE I DRAW 36
NEXT N
FOR I=1 TO 700 NEXT I CLS
```

MORE ↗

Continued from Column 2

```
REM SCHERM
CLS Y=-3
FOR X=0 TO 600 STEP 10
Y=Y+3 SET X 0 TO 600 Y
SET X 180 TO 600 180-Y
SET 0 Y TO 600-X 0
SET 0 180-Y TO 600-X 180
NEXT X
Y=103
FOR X=0 TO 300 STEP 10
Y=Y-3
SET X 90 TO 300 Y
SET X 90 TO 300 180-Y
SET 300 Y TO 600-X 90
SET 300 180-Y TO 600-X 90
NEXT X
FOR I=1 TO 600 NEXT I CLS
REM VLINDER
R=3 S=3 P=12 Q=12
FOR A=1 TO 241
C=C+1 P=P+R Q=Q+S
SET 128 96 TO P Q
IF P>249 THEN R=-3
IF Q>190 THEN S=-9
IF P<6 THEN R=6
IF Q<10 THEN S=9
NEXT A
FOR I=1 TO 900 NEXT I CLS
GOTO :START
===========
```

Thats enough of G for now! Don't forget to send me ANY graphics you produce, using any TI language or program - or indeed, send me any programs you produce. Must be someone else out there programming huh?

##############################

This is a JBM103/TI XB version of one of the above routines, much modified to make up for the lack of equivalent error handling- JBM103 is available from the group disk library.

```
100 CALL LOAD(-31890,56,0)::
CALL LOAD(-31964,56,0)
110 CALL LINK("CLEAR") :: CA
LL LINK("SCR2")
120 REM SCHERM
130 Y=0
140 FOR X=0 TO 190 STEP 3 ::
Y=Y+1
150 CALL LINE(X,0,190,Y)
160 CALL LINE(X,60,190,60-Y)
170 CALL LINE(0,Y,190-X,0)
180 CALL LINE(0,60-Y,190-X,6
0)
```

CONTINUED↗

```
190 NEXT X
200 Y=34
210 FOR X=0 TO 100 STEP 3 ::
Y=Y-1
220 CALL LINE(X,30,100,Y)
230 CALL LINE(X,30,100,60-Y)
240 CALL LINE(100,Y,190-X,30
)
250 CALL LINE(100,60-Y,190-X
,30)
260 NEXT X
270 GOTO 270
280 SUB LINE(A,Y,C,Z)
290 B=Y*3+30 :: D=Z*3+30
300 CALL LINK("LIGNE",16,A,B
,C,D)
310 SUBEND
320 END
```