TI*MES

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

Chairman: Gordon Pitt Tel.0922 476373
259 Sneyd Lane, Bloxwich, WALSALL, West Midlands. WS3 7LS
General Secretary: Jim Ballinger Tel. 0332 772612
5 Offerton Ave., DERBY. DE3 8DU
Membership Secretary & Telecoms, TI*MES Back Nos.: Peter Walker Tel. 0707 873778
24 Bacons Drive, CUFFLEY, Herts. EN6 4DU
Treasurer: Alan Rutherford Tel.0625524642
13 The Circuit, WILMSLOW, Cheshire. SK9 6DA
Publicity Officer: Phil Trotter Tel.0642 817356
80 Martonburn Rd. MIDDLESBROUGH, Cleveland. TS4 2TH
Programming: Mark Wills Tel.0743 64177
37 Abbots Rd., Monkmoor Estate, SHREWSBURY. SY2 5PZ
TI*MES Editor & Distribution: Alan Bailey Tel. 081 508 1053
14 Shelley Grove, LOUGHTON, Essex. IG10 1BY
Librarians: Cassette: Nicky Goddard Tel. 0978 843547
Sarnia, Cemetary Rd., Rhos, WREXHAM, Clwd. LL14 2BY
Disk: Stephen Shaw (& Journal Exchange & V.P.) Tel.061 432 6097 (after 8pm)
(letters preferred) 10 Alstone Rd., STOCKPORT, Cheshire. SK4 5AH
Modules: Edward Shaw Tel.0538 360382 (5pm to 8pm)
Crow Holt Farm, Basford, LEEK, Staffs. ST13 7DU
Publications: Mike Curtis Tel. 0209 219051
21, Treliske Rd., Roseland Gdns., REDRUTH, Cornwall. TR15 1QE
Hardware & Projects: Mike Goddard Tel.& Address as Nicky Goddard above.

MAGAZINE CONTENTS

ISSUE NO. 30 AUTUMN 1990

Magazine Contents (Change)

ISSUE NO. 30 **AUTUMN 1990**

EDITORIAL

Hello there! Here's another fun packed issue of your favourite magazine! Or anyway information packed. As I hope you will find some of the articles are designed to answer queries raised by members either on renewal forms or directly. If you need any answers on other points why not write or phone me or any other contact? The answers you get are bound to interest other members so I will get useful articles for the magazine!

On the subject of writing for us may I ask again for typescript or any other matter to be not more than 181mm wide. If your printer paper is wider, as it will be since printers seem to be USA sized rather than European, naturally enough, please make the left hand margin 15mm, since I then only have to trim one side of the paper. Incidentally Peter tells me that a suitable format is given by TI-Writer command LM 5;RM 73;FI;AD.

DISCLAIMER

Views expressed in this magazine are those of the contributor him or herself and not necessarily supported by the Committee.

NEXT COPY DATE

Please let me have any copy for the next issue of the magazine by 1st.December.

TIUG

TI-99/4a USERS GROUP (UK)

MINUTES OF ANNUAL CENERAL MEETING HELD 26th MAY 1990
Held at the NORTHGATE ARENA.CHESTER

Gordon Pitt in the Chair

Opening the meeting Gordon Pitt welcomed all members attending, remarking that although the Committee members at the table had met him on several occasions since his appointment as Chairman, he took the opportunity of introducing himself to other Group members.

Apologies for absence had been received from:-

John Harris who regretted that owing to pressing domestic commitments he was unable to attend this year. He assures us that he remains a committed member of the group, and wishes all attending an enjoyable day.

Alan Bailey (Editor/Publisher of TI*MES) who cannot attend due to travel problems. He makes plain that he is prepared to continue to serve in his present posts if required, or to take the responsibility of publication and distribution only if someone else wished to take on the Editor function.

Item 1. MINUTES OF 1989 AGM Were read and approved. As the minutes of each AGM are printed in TI*MES it is proposed that in future they should be taken as read at future AGMs. There were no matters arising

Item 2.REPORTS FROM OFFICERS :-

Gordon Pitt (Chairman) reported he had several ideas that he was keen to see mounted by the group, but they were of course dependant on the support evinced by members. He was keen on seeing the group take a greater part in the activities of continental clubs for example, and felt that the benefits to be gained outweighed the expected costs. He had represented the TIUG(UK) at such a meeting during the last year, and outlined some of the projects that were being pursued by continental

groups at present, including a card for the PEB that would enable compatibility with IBM, an 80 column card as well as Geneve support. He had attempted to promote ties between the various TI groups in Britain, and felt that he had met with some success. He concluded by listing the very active contintal groups, remarking that it was quite remarkable the number of new developments that were "on the cards" for a computer developed long enough ago to be considered outdated by many, and noting the increase in members owning expanded systems. A very good year he thought.

Stephen Shaw (Vice-Chairman,Disc Librarian,Journal Exchange) reported a good year, with the fall off in membership not being as great a he had feared since our computer was orphaned.(see Membership sec's report.) He presented his annual financial detail report on the Disc Library, which showed that after all expenses (books,freeware, donation,stationery,xerox ,postage etc, together with show and meeting costs) the balance at bank stood at £160.6p at 30.4.90 after donating £100 to the Group Central funds.(the relevant figures for the whole of 1989 were £469.52 and £200 respectively). He noted that many members did not seem to use the disc library as much as one would expect with the number of members who now possess the necessary equipment, which resulted in a loss in real terms this year so far(£308.83 being taken from the 1989 balance to compensate for this loss). The decline in the Journal exchange activity had continued, his opinion is that of the 3 major journals still being exchanged, we had reason to be proud of TI*MES.

Jim Ballinger (General Secretary) reported a satisfactory year, saying that at the last meeting of the committee a suggestion had been floated that a second "show" might be organised during the current year. This is included as an item on the agenda. During the last year he had met with some difficulty in performing his office for the group, and had tendered his resignation to the committee at their last meeting. The support offered by them had led him withdraw his resignation, but while he quite enjoyed the work felt that the physical limitation might make him ineffective in the future. He intended to stand this year if required, but thought should be given to a potential replacement.

The Chairman on behalf of the committee said that Jim had worked very hard as Secretary, and everybody appreciated his work for the committee.

Alan Rutherford (Treasurer) said that he had circulated the accounts, and showed a bright situation, close scrutiny revealed that this impression was perhaps misleading, but we were not doing too badly. Membership had fallen slightly, but the effect of the last increase in membership fees and the economies in the printing of TI*MES had dramatically improved the financial

position, and their full effect has not yet shewn itself. He felt confident that further increases would not be needed this year. Mike had taken over full financial control of the Hardware and repair section, which had proved successful, and he was talking to Edwad about a similar arrangement with him over the Modules Library. He had re-arranged the report so as make it coincide with the AGM coverage, a much easier format to read to gain the financial trends. Some extracts - (Figures in brackets cover 1988/89)

Balance £1812.07 (£894.33) TI*MES costs £1205.47 (£1664.44)
Subscriptions £2154.00 (£2064.00)

PeterWalker (Membership Sec,Telecoms) confirmed that membership had fallen off slightly, but the steady input of new members offset this to some extent. He noted that the members renewing their subscriptions had reported most favourably of the services provided for them ,which seems to meet their needs. Members either renewed without comment, or expressed their satisfaction, and it is good to read that we "have it about right". A few members felt that parts of TI*MES were not easy to read but the print could only be improved by using larger print which must mean less articles to read. The current membership level was 175, but until the renewals due at the end of the month is known (42 had not yet replied) it could not be clear what the figure would be then. He noted that several of the members involved were present at the AGM, and he would be delighted to receive a cheque from them. His guess was that by the end of the summer period we would be down to 140,but felt that this number would represent the real solid core of very enthusiastic members, as others drifted to other machines. He paid tribute to Stephen's activities in placing adverts in suitable magazines which had meant that up to this quarter we had just about as many new members as we had lost. Overall, he was not too dismayed at the membership trends, and he had no problems. The Telecommunications side, he was pressing ahead with, and progress was being maintained. He hoped that members at the meeting would put foward to him any ideas they had for improving the service, and he would be glad to meet those people he knew only through their letters.

Tim Anderson (Cassettee Librarian) had been forced by pressure of business interests, to resign from the position, recently and Nicky Goddard had agreed to carry on the service since then, and had done so with great success. Nicky felt unable to present a report on such short experience.

Edward Shaw (Modules Librarian) reported the the demand for modules had been about the same as last year, there was no sign of it falling off. Income for the year amounted to was £447.40 (modified to bring this figure in line with the Treasurer's new presentation [AGM to AGM]) this figure will be £550.50) with new module purchases of £532.81. The present stock of modules is

valued at £233 (resale value). Demand was brisk he commented.

Mike Goddard (Hardware and Projects) Mike felt that members were aware of the DIY and projects he had workd on since these had been published in TI*MES. He had also co-operated with Peter in the realisation of some of his (Peter's) designs. Further work was in hand and would be published as soon as completed. Those present will have seen that he, in addition to carrying out repairs and reconditioning services, he had built up a stock of new material and equipment for members convenience. A satisfactory year he felt.

The Chairman announced that with the Editor(Alan Bailey), and Publications Librarian (Michael Curtis)unable, and with no word from the Publicity Officer(Christine Bennett) that the reports were now concluded, and members of the committee then answered questions put from the floor.

He then asked for nomination for the various offices, and the following were elected by show of hands:-

ITEM 3 ELECTION OF OFFICERS

ALL appointments were unopposed,but the names of Proposers and Seconders will be supplied on request.

Chairman.....Gordon Pitt

Vice-Chirman.....Stephen Shaw

General Secretary.....Jim Ballinger

Treasurer.....Alan Rutherford

Membership Sec.Telecoms.....Peter Walker

TI*MES editor,Publisher.....Alan Bailey

Disc Librarian,Journals.....Stephen Shaw

Cassette Librarian.....Nicky Goddard

Modules Librarian.....Edward Shaw

Hardware and Projects.....Mike Goddard

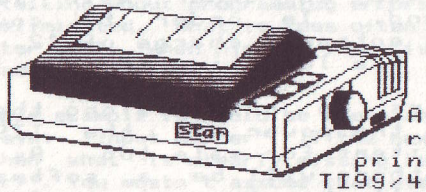
Publications Librarian.....Michael Curtis

Publicity Officer.....Phillip Trotter

Programming Officer.....Mark Wills

ITEM 4. Discussion on suggested second Annual Show. A discussion was held on the advisability of mounting a second show this year, and a vote revealed overwhelming support for the idea by members present. The venue for such a meeting was discussed, Peter has listed membership levels for areas. It was agreed that the committee would consider the proposition from all angles at it's next meeting, taking note of the feelings expressed by the members present.

ITEM 5.(Any other business)
Members raised questions, including the workshops held by the WM group at Bloxwich, but no propositions were raised. The Chairman thanked all members for attending and declared the meeting closed.



PRINTER SURVEY

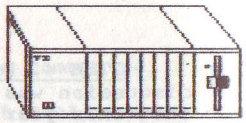
A number of members have requested information on printers suitable for the TI99/4A. In order to meet this request, I would like to publish an article on printers and would like to hear from as many printer-owning members as possible in order to collate your experiences. I would like the following information:

Name & Make of printer
Still available? Price?
Type (eg Dot matrix, Daisywheel, Inkjet, Laser)
Type of feed (Friction, Tractor or both)
Special Features (fonts, graphics, char sets)
Compatibility with other makes (eg Epson, IBM, Micronics, Axiom, Okidata, DEC)
Any interfacing problems with TI99 (RS232, PIO)
Any problems with common software (eg TI-Writer, TI-Artist)
What printer would you buy now if you had to replace?

I hope to hear from as many of you as possible

Peter Walker

SOUTHERN USER SHOW AND WORKSHOP



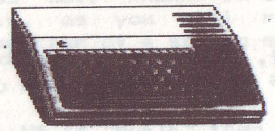
SATURDAY 26th JANUARY 1991

Recent AGMs and User Shows have all been in the Midlands and the North of England in recent years. Apart from our appearance at the London Alternative Micro Show, members in the South have not had a good opportunity to come to our shows and meet other 99ers. Next January all that changes, as we are holding a Southern Area Users Show and Workshop in my home village of Cuffley. The date is 26th January 1991, 11am to 5pm. The venue is Cuffley Hall. Cuffley is conveniently situated just north of the M25 and close to the M1, A1 and A10, so this venue should be accessible to a wide range of our members, not just those in the Home Counties. For those who wish to travel by rail, Cuffley is just 30 minutes by train from London, Kings Cross, which itself is well connected to other main line stations by Underground. In order to cover costs, there will be an entrance charge of £2. A bar will be open during the lunchtime period.

We expect many members to bring along their systems and demonstrate the power of the TI99. We hope there will be items for sale too. As an additional attraction, there will be a software competition. There will be two prizes of £10 each (or free membership for a year), one for cassette console only and one for disk, for the best new original program in any language. The only stipulation is that the author must demonstrate his program at the show and submit it for inspection.

Further details of this event will appear in the next issue of TI*MES. I hope to see as many members as possible at the Show!

Peter Walker



CONSOLE ONLY CORNER

BY PETER WALKER

When writing a program it is always worth remembering that it may be used by others and they might not necessarily react in the same way as you, the programmer. It is therefore worthwhile taking a few precautionary steps and employing what is generally called 'defensive programming'.

Firstly, do document your program so that others know what it does and how to operate it. Better still, try to provide on-screen help and, if this is long, make its viewing an option at the beginning of the program.

Secondly, make sure that as far as possible any run-time errors are trapped so that the program doesn't crash or emit 'warnings'. I'm not here talking about programming errors - we must assume that these have been eliminated. What is less often thought about is that the program operator will, when his/her input is requested, react in a way that you hadn't predicted.

A simple example is would be where an input is required and the screen displays: "Input number". If the next statement is an INPUT or ACCEPT AT then what is clearly missing is the phrase ".. and then press <Enter>". You mustn't assume that everyone knows to do this and in any event it is often the case that a single digit number is input by a CALL KEY which doesn't require the use of the Enter key.

Another problem arises with the following:

```
300 PRINT "Press Yes or No"
```

What is the operator supposed to do? Enter "Yes" or press "Y" (or the negative option)? Now look at this:

```
300 PRINT "Continue? Press Y/N"
310 CALL KEY(5,K,V)
320 IF V=1 THEN 310
330 IF K=89 THEN 500 ELSE 600
```

The problem here is that, if the alpha lock key was up, then pressing "Y" will set K to 121 not 89 and so the wrong branch will be taken. One way round this would be:

```
330 IF (K=89) + (K=121) THEN 500 ELSE 600
```

(ExBas owners could use IF K=89 OR K=121 THEN ...)

However, a far neater way is to use:

```
310 CALL KEY(3,K,V)
```

Key unit 3 interprets all key presses as upper case, so only K=89 needs to be tested.

If the operator is asked to pick from a menu of, say, 4 items by pressing 1, 2, 3 or 4 then you might program this as follows:

```
400 CALL KEY(3,K,V)
410 IF V-1 THEN 400
420 IF (K<49)+(K>52) THEN 400
430 ON K-48 GOTO 500,600,700,800
```

The check in line 420 ensures that no key press other than 1-4 will be allowed to proceed to 430 where any other value would cause an error. Another approach to input checking is as follows:

```
400 CALL KEY(3,K,V)
410 IF V-1 THEN 400
420 ON POS("1234",CHR$(K),1)+1 GOTO 400,500,600,700,800
```

This method is neater and also works well when the menu options are not numbers but letters eg A,B,C,D (Use ON POS("ABCD"..) and is the only sensible checking method when the letters are not alphabetically consecutive such as C,R,Q (As in Continue, Return or Quit).

A different sort of problem arises when you use numbers on a menu but the choices exceed 9. One cannot then use CALL KEY, but rather INPUT or ACCEPT AT must be used to input a one or two digit number. Error trapping could be implemented as follows:

```
400 INPUT "Input No then press <Enter> ":N
410 IF (N<1)+(N>12) THEN 400
```

This is not as neat as the CALL KEY trap since input of, say, 13 will cause the Input line to be repeated. A further problem is caused if the operator inputs a string instead of a number. This causes a 'honk' and a STRING-NUMBER MISMATCH warning which won't stop the program but is not a friendly way of trapping the problem. (As you can see we are talking about "Idiot Proofing" here!).

For those with ExBas, a better way of trapping unwanted key presses is through the use of validating with ACCEPT AT. For example:

```
400 DISPLAY AT(1,1):"Input No then press <Enter>"
410 ACCEPT AT(1,29)VALIDATE(NUMERIC):N
```

This ensures that only numbers can be input. But there is still one last possibility of error. If the operator presses <Enter> before inputting anything, a string-number mismatch will again occur since the numeric N cannot be equal to "". If you want to trap against this, you can use the following:

```
410 ACCEPT AT(1,29)VALIDATE(NUMERIC):N$
420 IF N$="" THEN 410 ELSE N=VAL(N$)
```

It is not just when inputting choices that errors can occur. It is always worth ensuring that 'divide by zero' errors are trapped:-

```
200 IF N=0 THEN 500
210 X=Y/N
```


Less well known is an error that can occur with the ASC function. This, as you will recall, returns the ASCII value of the first character of a string eg:

```
200 N=ASC(N$)
```

If N$ is "ALFIE" then N becomes 65. But if N$ is the null string "" then the ASC function will cause the program to crash, so it worth trapping as follows:

```
190 IF N$="" THEN N=-1 :: GOTO 210
200 N=ASC(N$)
210 ......
```

If you do have ExBas then you have another powerful method of error trapping through the statements ON ERROR, RETURN and CALL ERR. However, I have seen even experienced programmers use these incorrectly so it is worth explaining in detail how they should be used.

The statement ON ERROR 1000 sets up a condition such that any subsequent error, instead of halting the program, will cause the program to branch to a subroutine at line 1000. There are two important facts to remember. Firstly, an error branch is a subroutine, that is, it is like GOSUB rather than GOTO. Such a branch must return to the calling program via a RETURN statement. As we will see however, there are 3 forms of RETURN used with ON ERROR instead of the single version used with GOSUB. Secondly, the branch condition to line 1000 remains valid until either:

-another ON ERROR is executed

or

-an error causes the branching to the specified line.

In other words, ON ERROR sets up a 'pending' condition that is cancelled once used. So, if you want to continue error trapping after an error has occurred, you must execute another ON ERROR. However, you are not forced to do this, contrary to what one might glean from many of the manuals. Nor is it necessarily the best idea to rely on resetting the ON ERROR condition within the error subroutine itself, since the target routine might depend on where the subroutine returns to.

Lets look at the structure of a typical error trap:

```
100 ON ERROR 1000
110 REM YOUR PROGRAM
120 A=B/C
130 PRINT A

1000 PRINT "DIVIDE BY ZERO!"
1010 A=X
1020 RETURN NEXT
```

At line 100 we protect against division by zero at line 120. When C=0 an error occurs and the program branches to line 1000 and a default for A is set. RETURN NEXT causes the routine to return to line 130, the next line after the one where the error occurred. There are 3 types of RETURN:-

RETURN - this returns to re-execute the line where the error occurred.
RETURN NEXT - this returns to the line after that which caused the error.
RETURN XXX - this returns to a specific line XXX.

Now the important issue is this: given that the error subroutine must return in an appropriate way for the error that occurred, you mustn't expect to use one subroutine that can recover from all the errors that might happen. So you may want to have specific subroutines protecting specific lines where errors might arise. For example:-

```
100 ON ERROR 1000
110 REM 1000 recovers errors from here forward.
200 ON ERROR 1500
210 REM 1500 recovers from here onwards.
```

To clarify the issue of when the three types of RETURN should be used, study the following:-

```
100 ON ERROR 1000
110 INPUT "N? ":N
120 PRINT "N SQUARED IS ";N*N
140 PRINT "SQUARE ROOT N IS ";SQR(N)
150 PRINT "LOG N IS ";LOG(N)
160 GOTO 110

1000 ON ERROR 1000
1010 RETURN NEXT
```

In this example if the value of N causes any error to occur, the faulty line is passed over. Try with N=-1.

RETURN alone is used here:

```
100 ON ERROR 1000
110 INPUT "FILE? ":F$
120 OPEN £1:F$, INPUT
130 .....

1000 PRINT "CAN'T OPEN FILE ";F$
1010 INPUT "TRY AGAIN ":F$
1020 ON ERROR 1000
1030 RETURN
```

If the file can't be opened at line 120, then the filename is requested again at line 1010, control is returned to line 120.

Finally, RETURN XXX might be used as follows:-

```
100 ON ERROR 1000
110 REM START OF THE PROGRAM
120 ..

1000 PRINT "An error has occurred - program will restart"
1010 RETURN 100
```

Notice here that the error trap is reset outside the subroutine.

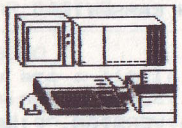
So, we have seen that the error recovery subroutine needs to be compatible with the error that called it. There is another way around this problem, which involves the use of CALL ERR. This routine returns the error type and line number where the error occurred. It is then possible to return in a manner appropriate to the error. For example:-

```
100 ON ERROR 1000
1000 CALL ERR(A,B,C,D) ! D IS LINE NUMBER
1010 IF D=210 THEN X=2 :: RETURN 200
1020 IF D=300 THEN PRINT "WRONG!" :: ON ERROR 1200 :: RETURN
1030 IF D=500 THEN RESTORE £1 :: RETURN NEXT
1040 PRINT "ERROR ";A;" AT LINE ";D :: STOP
```

In this example, the right RETURN is provided for errors occurring at lines 210, 300 & 500. Other errors cause a printout and stop. An important point to note is that the line numbers 210, 300 and 500 in lines 1010, 1020 and 1030 are not affected by a RESEQUENCE, so either avoid resequencing or remember to adjust these line numbers yourself after a resequence.

Well, I hope that explains how to remove or contain errors in your programs.

Peter Walker



MEMBERSHIP NEWS

From Peter Walker

Membership Secretary

Since issue 29, we welcome the following new members: Martin Ross, Eddy Carter, John Murphy, H. McAlroy, Erland Ericsson, Derrik Rush and Gerald White. Following the announcement in the last issue of TI*MES concerning back issues, let me repeat the present situation:

Available at £2 (members only): 4, 12, 13, 19, 24, 26
Available at £1 (members only and if ordering at least 5 issues): 8-11, 14, 16, 18, 20, 22, 25 & 27. (otherwise cost is £2)
Non-members: all available issues £3 each.

PLEASE ensure you add sufficient money to cover return postage! 5 issues together costs £1.80 to post.

Elsewhere in this issue there are details of a Southern Area User Show and Workshop. I would like as many members as possible in the Home Counties area, or beyond, to assist by bringing along their system to help with demonstrations. Please let me know if you can assist in this.

Peter Walker

CASSETTE LIBRARY REPORT..... NICKY GODDARD

Since the last cassette library report I have had quite a good response to the advert about surplus stock cassettes. There is a list of more at the end of this article.

It is because of the response that I have decided to lower the cassette library prices to the following. Please note that these prices are per programme.

80p IF YOU SEND A CASSETTE
£1 IF YOU DON'T SEND A CASSETTE
POSTAGE INCLUDED

I hope to get quite a good response to my decision.

Another notice is that due to him moving house, Mark Wills was unable to do the cassette reviews so I have done them for him.

IAL OFFER SPECIAL OFFER SPECIAL OFFER SPECIAL OFFER SPECIAL OFFER SP

BUY ANY OF THESE SURPLUS STOCK CASSETTES FOR 50P EACH OR 6 FOR £2.50

POSTAGE = 30p FOR THE FIRST CASSETTE AND 10P FOR EACH CASSETTE THEREAFTER

UTILITIES -

BEGINNERS BASIC TUTOR, WYCOVE FORTH, WYCOVE FORTH E/A VERSION, LINE BY LINE ASSEMBLER, GAMES WRITERS PACK 1, GAMES WRITERS PACK2, TEACH YOURSELF BASIC, PERSONAL FINANCIAL AIDS, TI LOGO SAMPLER.

GAMES -

OLDIES BUT GOODIES 1, CORE, KAT TRAXX, PILOT, ROBOPODS, FUN PACK 3, LASER TANK, RI SKI, ABM CONTROL/CAVERN HUNT*, DREAMER, BLAST IT, BATTLE OVER TITAN, DEVILS ISLAND/RUSSIAN ROULETTE*,GOBLIN CAVES/ ANAGRAM*.

ADVENTURE TAPES -

TUNNELS OF DOOM.

* SAME TAPE.
PLEASE SEND ALL ORDERS TO N.GODDARD,.
'SARNIA',
CEMETERY ROAD,
RHOS,
WREXHAM,
CLWYD,
NORTH WALES LL14 2BY.
PLEASE MAKE ALL CHEQUES etc, PAYABLE TO TI USER GROUP UK.

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.....

TIME TRAVELLER LIBRARY NO. G47

You are a time traveller and your duty is to gather stones on a strange planet.

While you are collecting the stones you must avoid little red, green and white monsters because if you stand on them (!) you lose a life.

You have 6 possesions, 3 lives and 3 time machines.

The game ends when you run out of lives.

The time machines can be used to escape to a totally different time, when things get tricky. When all the time machines have been used the game ends.

When you walk you leave a trail of 'RADIO-ACTIVE TIME-SLIME' . If you step on the time slime you lose a life. The edges of the edges of the screen are also covered in 'RADIO-ACTIVE TIME-SLIME' so be warned.

A good Basic keyboard game.
STAR RATING ***

Q*BONO LIBRARY NO. G51

You are a little blue object with 5 lives and your aim is to jump (using keys Q,E,Z,C)from square to square turning the square you land on blue and gaining points as you go along.

At each side of the triangle of squares there are 2 soap bar shaped multicoloured objects which carry you back to where you started if you jump on them.

But watch out for the white snake because if he jumps on you you lose a life.

If you jump over the side of the triangle of squares you lose a life as well.

A very good Extended Basic keyboard game.
STAR RATING ****

ON THE HARDWARE FRONT..... MIKE GODDARD

I must apologise to all those who expected a constructional article this quarter but I really haven't had time to prepare and test anything so I will try to produce a bumper bundle for the next issue. This is not to say things have been standing still far from it and there are several articles in the pipeline one is an excellent project from Britain so it would appear all is not lost.

I will try to answer some of the hardware points raised by members in response to the questionnaire on the membership renewal form I don't know who raised them so if it seems familiar it is probably yours!!.

Assistance in buying hardware required.

In general the group is too small to make large enough bulk purchases of hardware but if anybody does require anything specific or is having trouble obtaining anything please get in touch.

Would like to see MAPLIN style interfacing projects.

Well most Maplin style projects apertain to certain other machines allied to a Knight of the realm and an ex barrow boy. Seriously again we don't have a large enough membership to make such projects viable but I will endeavour to tailor projects to specific requirments if at all possible.

Can the Horizon horizon RAM disk be obtained in this country.

As far as I am aware nobody actually stocks these but several people such as Gordon Pitt, Richard Seirakowski and not forgetting DATABASE which is now run by Martin Blyth who will either supply such goods to order or will assist all they can in their purchase.

Suitability of printers for the TI.

Almost any Epson or Epson compatible printer should work on the TI ones I know to operate OK are EPSON MX80 (the original TI badged printer),EPSON FX80, EPSON RX80FT+, MANNESMAN TALLY 80+ PANASONIC KXP1081. The printers to avoid are IBM compatibles and TANDY types although some of these can be used with the correct modifications.

I think that about covers the direct hardware points but no doubt I'll be put right.

The most welcome and interesting visit from a member I've had this year was Chris Noon from Durban South Africa, no he didn't come 3000 miles just to see me but called in while here on business nice to see you Chris pop in any time!

MORE ABOUT MINIMEMORY
by John Stocks

Although there have been many articles in TI*MES concerning the Minimemory, complete working Assembly Code programs have been conspicuously absent and, despite many letters written, I have been unable to find anybody who is active in this field. Stephen Shaw's recent 'sixes of sevens' problem provides a striking example of what can be done, so I am writing this article in the hope of stimulating (or provoking) some of our Minimemory owners into getting out of their armchairs and actually producing something.

Firstly, here are some tips and general advice for newcomers. Ignore those people or books telling you how easy-peasy it all is. This is nonsense, but the difficulties are far from insuperable. Like learning to swim or play the piano or even ride a bicycle, a fair amount of dedication and effort is required, but the pleasure and satisfaction of mastering a difficult task make it well worthwhile. The fact that these activities come naturally to some people does not entitle those gifted ones to patronise the rest of us by telling us how simple it is, thereby implying that the rest of us are thickies!

One essential is a good guidebook and the one I am most familiar with is "Assembly Language on the TI99/4A" by Peter Lottrup. This seems to me a splendid effort. The book by Amrouche and Didi is riddled with misprints, translator's errors (it was written in French) and with opaque explanations, which make it more of an obstacle course than an instruction manual. However, it is still worth buying for its excellent range of original programs. If you are patient enough to work through them methodically, line by line, and satisy yourself that you understand what each item is doing, you will learn some invaluable programming lessons.

Most beginners will be aware of the importance of the "registers", sixteen designated memory locations in which mathematical and other operations can be performed. You can in fact specify these yourself, but it is usually best to use the recommended locations at >70B8 onwards. It is essential to use the registers for the more complex operations such as multiplication and division, but the simpler processes such as addition and subtraction can be carried out carried out in any memory location you like. Amrouche and Didi do not seem to be aware of this, but see page 256 of Lottrup's book for details. The important point is to plan the use of registers very carefully in order to avoid false operations. If, for example, you branch to a subroutine with the BL directive, the return address is placed in R11 whether you like it or not. If therefore you wish to branch from a subroutine to another subroutine, you must move the R11 information to another place before you branch, remembering to return it to R11 when the branch operation is over. Similar shuffling around is frequently necesaary, although no fewer than sixteen registers are actually available. VMBW for example insists on using registers 0, 1 and 2, so if you have any information stored there, you must get it out of the way p.d.q.

When you have developed your program and are looking for faults, it is useful to bring operations to a halt. Amrouche/Didi recommend the rather mysterious entry >2C56 in two successive program lines. A better one is >10FF, which simply means 'jump to where you are'. This gives a lockout of course, but if you switch off and then on again, all the memory locations are available for you to

inspect. Several 'disassembler' programs are available which will tell you exactly what your machine code says, so if you have a printer these will be a real boon in bug-hunting. If you have no printer, you should keep a detailed and accurate record of everything you do.

One final point : it is often tempting to assume that the computer has made a mistake, but you should not deceive yourself. Lockouts etc. are always caused by faulty programs, so when you hit trouble, look, look and look again!

Turning now to the program itself : the solution is delivered in just over two seconds, as opposed to six minutes in the Basic version that Stephen Shaw has published. (In Turbo-Pascal it takes a very creditable ten seconds.) As you can see, the program breaks down very conveniently into blocks consisting of a framework and subroutines. The framework block takes us up to the eleventh power without automatic testing. (The lower powers can be tested visually, so there is no point in testing them in the program.) Higher powers are tested after each multiplication stage has been carried out. The multiplied result of each power is stored at locations from >7118, for the smallest digit, upwards. It is admittedly wasteful to allocate two 8-bit memory locations for each digit, but it does give simplification of the program through the straightforward numbering sequence. Once the required group of six sevens is identified, we print it and, after a delay, return to the Minimemory captions.

Even those totally unfamiliar with Assembly Code are urged to punch it all in and convince themselves that it really works. The thought of all that arithmetic in just two seconds is really mind-boggling!

Experimenters might care to investigate further by eliminating the full printout routine and just displaying the 'valid' powers. A program developed for the Expansion Memory shows that there are no fewer than 52 powers giving the required group of six sevens, up to the ten thousandth power of seven, but it takes one and and a half hours to deliver the goods! If you produce something similar for the Minimemory, remember that it will eventually overwrite the start of your program at >7D00, so watch out!

PROGRAM TITLE : "SEVENS"
==========

```
      PROGRAM FRAMEWORK

7D00  LWPI >70B8          LOCATE REGISTERS
7D02
7D04  LI R1,1             TOTAL NUMBER OF DIGITS
7D06
7D08  LI R7,R7            MULTIPLICATION CONSTANT
7D0A
7D0C  LI R10,R10          DIVISION CONSTANT
7D0E
7D10  MOV R1,@>7118       LOAD SMALLEST DIGIT
7D12
7D14  CLR R2              POWER REACHED
7D16  CLR R3              CARRY
7D18  BL @>7D2C           MULTIPLY BY SEVEN
```



```
7D1A
7D1C  CI R2,>B            ELEVENTH POWER REACHED?
7D1E  JNE >7D18           MULTIPLY AGAIN
7D20
7D22  BL @>7D2C           MULTIPLY
7D24
7D26  BL @>7D54           TEST FOR SEVENS
7D28
7D2A  JMP >7D22           MULTIPLY AGAIN

      MULTIPLY BY SEVEN

7D2C  INC R2              POWER
7D2E  CLR R4              NUMBER OF DIGITS MULTIPLIED
7D30  LI R0,>7118         LOCATION OF SMALLEST DIGIT
7D32
7D34  INC R4
7D36  MOV *R0,R5          MOVE DIGIT FOR MULTIPLICATION
7D38  MPY R7,R5           MULTIPLY DIGIT BY SEVEN
7D3A  A R3,R6             ADD CARRY - IF ANY
7D3C  DIV R10,R5          DIVIDE PRODUCT BY TEN
7D3E  MOV R5,R3           QUOTIENT CARRIED OVER
7D40  MOV R6,*R0+         REMAINDER STORED IN PLACE OF ORIGINAL DIGIT
7D42  C R4,R1             ALL DIGITS MULTIPLIED?
7D44  JNE >7D34           IF NOT, MULTIPLY NEXT DIGIT
7D46  CI R3,0             ANY CARRY?
7D48
7D4A  JEQ >7D52           IF NOT, RETURN
7D4C  MOV R3,*R0          IF SO, STORE AS NEW DIGIT -----
7D4E  CLR R3              ----- CLEAR CARRY -----
7D50  INC R1              ----- AND INCREMENT NUMBER OF DIGITS
7D52  B *R11              RETURN

      TEST FOR SEVENS

7D54  CLR R4              SERIAL NUMBER OF DIGIT TESTED
7D56  LI R0,>7116         LOCATION OF SMALLEST DIGIT, MINUS TWO
7D58
7D5A  CLR 8               NUMBER OF SUCCESSIVE SEVENS
7D5C  AI R4,6             TEST EVERY SIXTH DIGIT
7D5E
7D60  C R4,R1             HAS END OF COMPLETE NUMBER BEEN PASSED?
7D62  JGT >7D94           IF SO, RETURN
7D64  AI R0,>C            LOCATION OF NEXT SIXTH DIGIT
7D66
7D68  MOV *R0,R6          INSPECT DIGIT
7D6A  C R6,R7             IS IT A SEVEN?
7D6C  JNE >7D5C           IF NOT, TRY NEXT SIXTH DIGIT
7D6E  DEC 4               IF SO, INSPECT PREVIOUS DIGIT
7D70  DECT R0             LOCATION OF PREVIOUS DIGIT
7D72  MOV *R0,R6          RETRIEVE PREVIOUS DIGIT FOR INSPECTION
7D74  C R6,R7             IS IT A SEVEN?
7D76  JEQ >7D6E           IF SO, MOVE FURTHER BACKWARDS
7D78  INC R4              IF NOT, MOVE FORWARD AND START COUNTING
7D7A  C R4,R1             HAVE WE PASSED THE FINAL DIGIT?
7D7C  JGT >7D94           IF SO, THEN RETURN
7D7E  INCT R0             IF NOT, MOVE TO LOCATION OF NEXT DIGIT
7D80  MOV *R0,R6          RETRIEVE NEXT DIGIT FOR INSPECTION
7D82  C R6,R7             IS IT A SEVEN?
7D84  NOP                 (NO OPERATION - AN OVERSIGHT ON MY PART!)
7D86  JNE >7D5A           IF NOT, THEN MOVE ON AND START LOOKING AGAIN
7D88  INC R8              IF SO, CLOCK IT UP IN REGISTER 8
```

```
7D8A  CI R8,6              HAVE WE CLOCKED UP SIX SEVENS YET?
7D8C
7D8E  JNE >7D78            IF NOT, INSPECT NEXT DIGIT
7D90  B @>7D96             IF SO, PRINT AND TERMINATE
7D92
7D94  B *R11               RETURN TO FRAMEWORK

      PRINT AND TERMINATE

7D96  MOV R1,R14           MOVE DIGIT TOTAL OUT OF HARM'S WAY
7D98  MOV R2,R6            DITTO FOR POWER
7D9A  LI R0,>182           POSITION ON SCREEN FOR PRINTING TEXT
7D9C
7D9E  LI R2,14             NUMBER OF LETTERS TO PRINT
7DA0
7DA2  LI R1,>7E22          STORE LOCATION OF TEXT FOR PRINTING
7DA4
7DA6  BLWP @>6028          MINIMEM SUBROUTINE : PRINT TEXT
7DA8
7DAA  LI R0,>195           SCREEN LOCATION FOR SECOND TEXT BLOCK
7DAC
7DAE  LI R2,2              LENGTH OF DITTO
7DB0
7DB2  LI R1,>7E30          STORE LOCATION OF DITTO
7DB4
7DB6  BLWP @>6028          PRINT SECOND TEXT BLOCK
7DB8
7DBA  LI R0,>193           SCREEN LOCATION TO PRINT POWER
7DBC
7DBE  CLR R5               NECESSARY FOR DIVIDING INTO R6
7DC0  DIV R10,R5           DIVIDE POWER BY TEN
7DC2  MOV R6,R1            REMAINDER GIVES REQUIRED POWER DIGIT
7DC4  AI R1,>30            CONVERT NUMBER TO ASCII CODE
7DC6
7DC8  SWPB R1              SHIFT NUMBER TO MOST SIGNIFICANT BYTE
7DCA  BLWP @>6024          MINIMEM SUBROUTINE TO PRINT SINGLE CHARACTER
7DCC
7DCE  DEC R0               MOVE TO PREVIOUS SPACE ON SCREEN
7DD0  MOV R5,R6            UNPRINTED RESIDUE OF POWER
7DD2  CI R6,0              HAVE WE PRINTED ALL POWER DIGITS?
7DD4
7DD6  JNE >7DBE            IF NOT, DIVIDE AND PRINT AGAIN
7DD8  MOV R14,R0           TOTAL DIGITS IN FINAL NUMBER
7DDA  AI R0,>1C0           ESTABLISH SCREEN LOCATION OF SMALLEST DIGIT
7DDC
7DDE  MOV R14,R4           TOTAL DIGITS TO BE PRINTED
7DE0  INC R4
7DE2  LI R5,>7118          STORE LOCATION OF SMALLEST DIGIT
7DE4
7DE6  DEC R0               MOVE BACK ONE SPACE ON SCREEN
7DE8  DEC R4
7DEA  MOV *R5+,R1          RETRIEVE DIGIT FOR PRINTING
7DEC  AI R1,>30            CONVERT TO ASCII
7DEE
7DF0  SWPB R1              SHIFT ASCII CODE TO MSB
7DF2  BLWP @>6024          PRINT SINGLE CHARACTER
7DF4
7DF6  CI R4,1              HAVE WE PRINTED ALL DIGITS?
7DF8
7DFA  JNE >7DE6            IF NOT, PRINT NEXT (HIGHER) DIGIT
```



```
      TIME-DELAY LOOP

7DFC  LI R0,>FFFF
7DFE
7E00  LI R1,11
7E02
7E04  DEC R1
7E06  DEC R0
7E08  CI R0,0
7E0A
7E0C  JNE >7E06
7E0E  CI R1,0
7E10
7E12  JNE >7E04
```

```
      TERMINATION ROUTINE

7E14  CLR R0
7E16  MOVB R0,@>837C
7E18
7E1A  LWPI >83E0
7E1C
7E1E  B @>70
7E20
7E22  TEXT 'SEVEN TO POWER'
7E30  TEXT 'IS'
7E32  AORG >701C
701C  DATA >7E32           FFAM
701E  DATA >7FF8           LFAM
7020  AORG >7FF8
7FF8  TEXT 'SEVENS'        PROGRAM TITLE
7FFE  DATA >7D00           PROGRAM ADDRESS
8000  END
```

TREASURE TRAIL

Peter Hutchinson.

VOODOO CASTLE tips. (Adventure Module adventure).
CAN'T READ PLAQUE.....ecalp kard a ni ti dear
Test tubes a bother.....yromra eht tisiv
Stuck in jail cell.....was a deen ouy?
What is bag for.....gnir eht thiw od uoy did tahw
JuJu man is mumbling..... mih ot netsil
Where is 2nd charm.....puos emos dnif

PYRAMID OF DOOM tips...
Cobra proving deadly.....cisum emos yalp
Mummy strangling you....taeh eht ecuder
Mirror room a problem....od nem dnilb od tahw
Metal bar a problem.....daeh ruoy esu
Oyster an obstacle.....ti deef
Wheres treasure room...lufesu si enots
A treasure short.....nobrac fo eciep a yrt
or perhaps.....gnihtemos was
Purple worm food?.....ti erongi
Rats difficult.....meht deef tnod

QUERIES...
Does anyone know what to do in MANIA?

Pyramid of Doom....
How do you get up the hole above the ledge? What use is the blood stained altar? How do I find che pharoah? What use are... the rope, bandages,chopping block?

Fully solved if you need help are: Strange Odyssey, Adventureland, The Count, Voodoo Castle, Pirate Adventure.

Please write: 6 Moorlands View, Free School Lane, HALIFAX, HX1 2XQ Telephone 0422 355857

==========

RAMBLES STOP PRESS...

TI BASE: Vn 3.00 wont allow an X in a field name! Version 3.01 has a problem with DATE. Following fix for 3.01 from Bill Gaskill- insert into your setup file:

```
CHANGE F84C 06A0
CHANGE F84E FFD8
CHANGE FFD8 0420
CHANGE FFDA 32E0
CHANGE FFDC 2912
CHANGE FFDE 045B
```

Bill Gaskill's address is:
2310 CYPRESS CT, GRAND JUNCTION, CO
U.S.A. 81506

as always only change a COPY of the program, leave the master disk untouched!
BILL GASKILL has produced a TI BASE newsletter, called THE TI BASE USER and at time of writing it is up to issue 4.
A subscription to the first volume of 6 issues is US$22 (in US dollars!) to UK subscribers including postage. Bill wrote a tutorial for Vn 1 (available from the library). The first two issues of his newsletter cover Version 2 (still applicable to Vn 3) and then move on to Vn 3. Each issue is 12 pages A4. Recommended to TI Base users!

==========

Next issue of Rambles- reviews of CLASS and FasTrans by Bill Harms, REMIND ME by Johnson, and a new Fractal book and of course more Fractal programs. Send me three disks and return post for full disk library catalogue! Stephen.

MACHINE CODE

TUTORIAL 1 - THE BEGINNING

by Mack McCormick
74206,1522

Here are the objectives of this first tutorial: 1. To introduce you to the Hexadecimal and Binary Numbering systems. 2. To introduce you to the assembler instruction format. 3. To introduce you to addressing modes. 4. First program. Adding two numbers and displaying them on the screen. 5. How to assemble.

Just a few words on Assembly language before we begin. It's not as difficult as you may believe. You will be communicating with the microprocessor at the first level above machine language, assembler. As you know, the machine actually communicates in binary 0's and 1's, on or off. Assembler allows us to talk to the machine in a language we can understand (Although I'm sure the uninformed would disagree). With these tutorials I will assume no prior knowledge of assembler or other number systems. Please bear with me, I won't insult your intelligence and things will become more complex soon. Stick with the tutorials. Read every book about assembler you can get your hands on. I will publish a bibliography of books soon. Don't get discouraged! Compuserve is a difficult medium thru which to provide assistance. I promise to answer your questions and if I don't know the answer, I'll find someone else that can. Please make this an interactive process, as we grow and learn with each other.

Numbering Systems

Hexadecimal (HEX) and binary are merely different base numbering systems for counting. It's important we understand both of these systems in addition to base 10 or the decimal system because assembler uses all three. I will tell you up front that I use a calculator designed for these numbering systems usually but we need to understand the principles also. If you want to get a calculator, and I recommend that you do, there are several inexpensive models on the market. I use the Casio solar powered fx-451 scientific calculator for $35. It supports HEX, OCT, BIN, LOGICAL OPERATORS, and all scientific functions. Works great! Craig Miller and others have also published programs which will allow you to use your computer but this has the disadvantage of requiring you to load another program every time you need to make a calculation, a real pain.

Binary Number System

As already mentioned, binary is the native language for your computer. Everything eventually gets converted to binary. Let's look at a decimal number first. As you know decimal means powers of 10. Each number represents a power of ten. For example 4175:

```
10^3=1000  4 X 1000=4000
10^2= 100  1 X  100= 100
10^1=  10  7 X   10=  70
10^0=   1  5 X    1=   5
                  ------
                  4175
```

Binary numbers can be 1 or 0 only, hence base 2. The individual number is called a bit. A group of eight of these is called a byte. To convert the binary number 00001011 to decimal follow the same procedures you used with the decimal number:
Ignore any leading zeros.

```
2^3=8      1 X 8=8
2^2=4      0 X 4=0
2^1=2      1 X 2=2
2^0=1      1 X 1=1
                ---
                11
```

To make it easier to communicate with the computer we most often use HEX. From now on I will use a > to indicate a number is in hex. Hex is base 16. That is a number may be 0 thru 15. To represent numbers greater than 9 we use letters of the alphabet. 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F. Just remember to use >A for 10 and count to 15 ending with >F. Let's convert >394F to decimal:

```
16^3=4096   3 X 4096=12,288
16^2= 256   9 X  256= 2,304
16^1=  16   4 X   16=    64
16^0=   1   F X    1=    15
                   ---------
                   14,671
```

The largest number you may represent in one byte is >FF or decimal 255. The largest value in a word (two bytes) is >FFFF or 65,535. Enough on numbering systems for now, we'll cover minus numbers (twos compliment) and additional points as we encounter them in programs.

Assembler Instruction Syntax.

Like every computer language there are certain rules we must follow for inputting instructions. Unlike BASIC, assembler will not give you a warning or error until you assemble the program. Here's the general syntax:
LABEL OPCODE OPERAND COMMENT
labels must begin in the 1st column and may be up to 6 characters long. One or more spaces follow. Next is the OPCODE. This is the actual instruction to be performed followed by one or more spaces. Next are one or more operands or data for the instruction to operate on followed by one or more spaces. Finally is an optional comment which may extend to column 80. Each time we use a new instruction I will fully explain it.

Addressing Modes.

There are five general addressing modes and one special addressing mode used for assembler instructions. We will examine each one in detail as we encounter them in a program. There is one type of addressing we need to look at now. We are going to be operating on individual bits, bytes, and words of memory. Think of the computers memory as a series of consecutive small pieces of memory laid out end to end. We can address any single byte of memory by hanging a label on it but frequently we must address a byte of memory some distance from that label. Think of it like an array. To get to the 5th byte from the label we could say LABEL+4. We used 4 instead of 5 because we must start counting from 0. Think of it like OPTION BASE 0 in BASIC. Lot's more on this later.

First Program. I strongly recommend you enter the program manually by typing it in instead of just cleaning it up using TI-Writer or Editor/Assembler. The only way to gain experience programming is to practice.
I've place the program separately in DL3 to meke it easier to read. It's name is PRO1.ASM.

Program explanation . These comments supplement the comments contained in the program itself. Any statement with an * in column 1 is a comment and you may enter anything else on that line. One fairly unique thing about the 9900 microprocessor in the TI-99 is the ability to designate your own workspace registers anyplace in memory or more than one set at a time. Think of registers as 32 consecutive bytes of memory that are used as your scratch paper for calculations. Thirty-two bytes is of course 16 words of memory. Because this is a 16 bit (1 word) machine (something many of your friends can't brag about) that gives us 16 registers to use for our computations. We place an R in front of the number to designate that we are refering to a register. For example, R0 is register zero and R15 is register fifteen (really the 16th word of memory because we started counting with zero.) Here's the detailed explanation of the program:

```
      DEF START
```

DEFines the entry point of the program so the computer may find it. Places the name START in the Reference/Definition table. More on this next time.

```
      REF VSBW,VMBW
```

REFerence refers to console routine the program will use. In the advanced tutorials we'll create our own utilities.

```
WSREG  BSS >20
```

WSREG is the label we decided on for our workspace registers. Could have been any label 1 to 6 characters long. Block Starting Symbol (BSS) sets aside a block of memory, unititialized for use as our workspace. >20 means set aside >20 or 32 bytes (16 words) for R0 to R15.

```
X      DATA 10
```

Initializes one word of memory to 10 (0010). Hangs the label X on that word.

```
START  LWPI WSREG
```

Load Workspace Register Immediate (LWPI). Tells the computer to use the 32 bytes of memory for our workspace which begin at WSREG. START is the entry point (beginning) of our program.

Logic for clear routine. Writes the space character >20 or 32 to the screen 768 times to the screen to clear it. R0 is the counter. R1 contains the space character. We increment R0 to point to the next screen location and check it against 767 to insure we haven't gone too far.

```
      CLR  R0
```

CLeaRs the contents of R0 to zero.

```
      LI   R1,>2000
```

Load Immediate R1 with >2000.

```
LOOP   BLWP @VSBW
```
Branch and Link Workspace Pointer to the Video Single Byte Write Routine. Brances to the console routine for writing single bytes of information to the screen. R0 always contains the address on the screen to write to. Briefly, there are 768 screen positions 24 rows X 32 Columns=768. This routine writes to VDP RAM in the screen image table (SIT) which is 768 bytes long. Any ASCII value you write to the SIT is displayed on the screen. For example to display the number 3 at row one column one, R0 would have 0 in it (because we begin counting at 0) and R1 would contain >3300 or 5100 in it. Note the number to be written is in the left byte of the word. The VSBW routine always writes on the Most Significant Byte and disregards the LSB. Here's the easy way to remember it. R0 always contains the address in VDP RAM. R1 always contains the address or data in CPU RAM.

```
     INC   R0
```
INCrement R0 by one. Add one to the contents of R0.

```
     CI    R0,767
```
Compare Immediate whats in R0 to 767.

```
     JLE   LOOP
```
Jump Less than or EQUAL to the label LOOP. IF R0<=767 THEN GOTO LOOP.

Logic for the addition routine. We add the two number together. Because only ASCII numbers may be displayed on the screen we must add >30 to each byte before we display it. In this case our number is 37. We must place a 3 and 7 on the screen. To do this we divide 37 by 10 resulting in a quotient of 3 and remainder of 7. We add >30 to the 7 to make ASCII >37. We move this value to the right most byte of our ANS word. We then divide 3 (old quotient) by 10 resulting in 0 quotient and 3 remainder. We again mask up by >30 and shift it left 8 bits (1 byte) in the register. We then move this byte to the left (MSB) of ANS. ANS looks like >3337 when we're finished.

```
     A @Y,@X
```
Adds two words of memory. Places the sum in the second operand. May also use registers (eg. A R0,R1). Adds whats at the word of memory with label Y to whats at the word of memory label X.

```
     DIV   @TEN,R5
```
DIVides uses two registers. In this case R5 and R6. Divides whats in R6 by whats at TEN or 10. The quotient is placed in R5 and the remainder at R6. That is why we clear R5 before we divide.

```
     MOV   R6,@ANS
```
MOVe the contents of R6 to whats at the label ANS.

```
     SLA   R6,8
```
Shift Left Arithmetic. Shift the contents of R6 left 8 bits (1 byte) to the MSB. Fills the shifted out positions with 0.
```
     MOVB  R6,@ANS
```
MOVe Byte moves the Most Significant Byte (MSB) or leftmost to the word at ANS without disturbing the LSB of ANS.

Logic for display on the screen routine. R0 contains the position on the screen to display the answer. Found by SCREEN ADDRESS=((ROW-1)*32)+(COLUMN-1). In this case 366. R1 contains the address of the beginning of the data to write to the screen. In this case R1 contains the address of ANS. R2 contains the number of bytes to write beginning at the VDP address in R0 and the CPU address in R1.


```
     JMP   $
```
Instructs the computer to JuMP to the current location of the program counter. Same as 100 GOTO 100. This locks up the computer so you may see the result. If you want to see how quickly the computer executes place an * in column 1 in front of this instruction and reassemble.

Logic for the Return to the Calling Routine. Clears the GPL status byte at >837C. Much more on this important byte later. Loads the workspace pointer back to the GPL workspace and branches to the routine at >0070. END is a directive to inform the assembler there are no more instructions.

How to Assemble.

If you have the Molesworth book refer to page 42 or page 30-36 in the Editor/Assembler manual. Here's a brief step by step.
1. Select EDITOR/ASSEMBLER from the main menu. Place your editor assembler disk A in drive 1.
2. Select 1-EDIT from the E/A menu.
3. Select 2-EDIT from the EDIT menu.
4. Enter your program just as shown. You may omit any comments if you desire.
5. Press FCTN ESCAPE twice to return to the EDIT title screen.
6. Select 3-SAVE. Answer Y to the VAR/80 prompt. Enter your source file name such as DSK2.SOURCE. If you only have one drive place another disk in drive one first and use DSK1 instead of DSK2.
7. Press FCTN ESCAPE to return to E/A title screen.
8. Select 2-ASSEMBLE. Answer Y to the load assembler question. Insure the E/A disk A is in drive 1.
9. At the SOURCE FILE NAME enter the same file name you used in 6 above. ie. DSK2.SOURCE, press enter.
10. At the OBJECT FILE NAME enter DSK2.OBJECT and press enter.
11. Press enter at the LIST FILE NAME. More on this feature next time.
12. At the OPTIONS prompt enter R. Press enter. R means you used R in front of your register numbers in the source code. You may also enter CLST. C is compressed object code (will not load from X/B loader). L is a source listing if you entered a LIST FILE NAME at the prompt. S prints the symbols and registers used in the program on your source list. T prints the full text string in your source listing. More on these features later. Assembler executing will appear followed by 0000 errors (you hope). Press enter.
13. Press 3-LOAD AND RUN.
14. At the FILE NAME prompt enter your OBJECT file name ie. DSK2.OBJECT. Press enter. Press enter again to get to the PROGRAM NAME prompt.
15. This is the name we DEF in our program in this case START.
16 Your program will execute.

SUMMARY.
I realize this has been long but there has been much to cover to get started. Don't get discouraged. We'll go at this together. I strongly recommend you study these references in your Editor-Assembler manual and experiment on your own. Until next time, "ASSEMBLER EXECUTING".
Page 20-36 Using the Editor-Assembler Cartridge.
Page 39 Sec 3.1 Registers
Page 46-48 Source Statement Format
Page 53 Predefined Symbols. ($)
Page 57-62 Sec 4.1.1, 4.1.4, 4.2, 4.4 Addressing.
------> continued---->>

```
Page 80          Add Instruction
Page 85          Add Immediate
Page 88          Divide
Page 90          Increment
Page 107         Branch
Page 115         Jump Less Than or Equal.
Page 143         Compare Immediate
Page 163         Load Immediate
Page 165         Load Workspace Pointer Immediate
Page 166         Move Word
Page 168         Move Byte
Page 200         Shift Left Arithmetic
Page 212         Block Starting Symbol
Page 225         Data
Page 227         DEF
Page 228         REF
Page 248         VSBW,VMBW
Page 329-330     Graphics Mode Tables
Page 394-396     Numbering Systems
Page 428-429     ASCII character set
Page 442         Other Returns
```

[Stephen here.... would you like the rest of this tutorial? Please write and tell me. No letters means thats all you get!]

```
*****************************************
*                                       *
* THIS IS THE FIRST PROGRAM FOR THE     *
* BEGINNER ASSEMBLER TUTORIAL.          *
* IT CLEARS THE SCREEN, ADDS TWO NUMBERS *
* TOGETHER AND DISPLAYS THE SUM IN THE  *
* CENTER OF THE SCREEN.                 *
* HERE'S WHAT IT WOULD LOOK LIKE IN     *
* EXTENDED BASIC:                       *
* 10 CALL CLEAR                         *
* 20 X=10                               *
* 30 Y=27                               *
* 40 X=X+Y                              *
* 50 DISPLAY AT(12,15):X                *
* 60 END                                *
*    BY MACK MCCORMICK 74206,1522       *
*****************************************
```

```
* THIS PART OF THE PROGRAM IS THE INITIALIZATION *

      DEF  START  THE PROGRAM NAME IS START
      REF  VSBW,VMBW CONSOLE ROUTINES WE ARE GOING TO USE
WSREG BSS  >20    SETS ASIDE A BLOCK OF 32 BYTES FOR USE AS WORKSPACE REGISTERS
X     DATA 10     COULD HAVE USED >A INSTEAD OF 10 (LIKE X=10)
Y     DATA 27     (Y=27) COULD HAVE ALSO SAID Y EQU 0027
TEN   DATA 10
ANS   DATA 0        WORD TO PUT ANSWER IN. INIT TO 0.
*    ----> continued ---->
```



```
* PROGRAM BEGINS HERE
*------------------------------*
START  LWPI WSREG    LOAD WORKSPACE POINTER IMMEDIATE. POINT TO OUR WORKSPACE.

* CLEAR THE SCREEN
       CLR  R0       CLEARS R0 TO ZERO (BEGINNING OF SCREEN IMAGE TABLE)
       LI   R1,>2000 LOAD IMMMEDIATE R1 WITH >2000. VSBW ROUTINE WRITES THE LEFT
LOOP   BLWP @VSBW    BYTE IN R1 ALWAYS. IN THIS CASE >20 OR 32 OR SPACE CHR
       INC  R0       ADD 1 TO R0
       CI   R0,767   COMPARE IMMEDIATE R0 TO 767
       JLE  LOOP     IF ITS LESS THAN OR EQUAL JMP (GOTO) LOOP

* ADD THE NUMBERS TOGETHER AND CONVERT TO ANCII
*------------------------------*

       A    @Y,@X    ADDS X TO Y AND PLACES RESULT IN X
       MOV  @X,R6    MOVE WHATS AT X TO R6
       CLR  R5       CLEAR R5
       DIV  @TEN,R5  DIVIDES 10 INTO 37. QUOTIENT IN R5. REMAINDER R6.
       AI   R6,>30   ADD IMMEDIATE ANCII OFFSET TO R6
       MOV  R6,@ANS  MOVE CONTENTS OF R6 TO THE WORD ANS
       MOV  R5,R6    MOVE CONTENTS OF R5 TO R6
       CLR  R5       CLEAR R5
       DIV  @TEN,R5  DIVIDE 10 INTO R5, R6
       AI   R6,>30   ADD IMMEDIATE ANCII OFFSET >30 TO R6
       SLA  R6,8     SHIFT LEFT ARITHMETIC R6 8 BITS.
       MOVB R6,@ANS  MOVE MSBYTE R6 TO R1
*------------------------------*
* DISPLAY ON THE SCREEN AT ROW 12 COLUMN 15

       LI   R0,366   POSITION ON THE SCREEN IS 366
       LI   R1,ANS   LOAD R1 WITH THE ADDRESS OF ANS
       LI   R2,2     TWO BYTES TO WRITE
       BLWP @VMBW

       JMP  $        PREVENTS THE PROGRAM FORM ENDING SO YOU MAY SEE THE RESULT

*------------------------------*
* RETURN TO THE CALLING PROGRAM

       CLR  @>837C   CLEAR THE STATUS BYTE
       LWPI >83E0    LOAD GPL WORKSPACE REGISTERS
       B    @>0070   BRANCH TO THE CALLING PROGRAM

       END

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

GPL - GRAPHICS PROGRAMMING LANGUAGE...

The following is extracted from a 1979
TI Manual, in which it forms Chapter 1:

1.0 GRAPHICS PROGRAMMING LANGUAGE

The System software resident in the product consists of a monitor and a GPL (Graphics Programming Language) processor. It is the function of the monitor to insure that every time the system is turned on, a new cartridge is inserted, or an existing program terminates, that all memory and peripheral devices are initialized. The GPL processor is an interpreter optimized to execute GPL programs directly out of GROM. The GPL processor software is coded in TMS 9900 assembly Language. NOTE: See Appendix N.

1.1 OVERVIEW

GPL is a programming language specially developed by Texas Instruments to provide the best possible tradeoff of code compaction, execution speed, and ease of program development for the target computer system. The GPL instruction set facilitates development of programs which make use of the unique features of the system chip set. It is byte oriented, and instructions typically have one or two operands. The addressing scheme is such that most instructions can access either standard microprocessor RAM, GROM, or the video scratchpad RAM address space easily.

Most instruction operands can be either single or double byte values. The addressing modes are: immediate, direct, indirect, indexed, indexed indirect (with pre-indexing), and 'top of stack'. Source operands and destination addresses can be in the CPU, video RAM, or in GROM. Support for two stacks is available; a data stack and a subroutine return address stack (allowing arbitrary nesting of subroutines).

1.2 GPL INSTRUCTION SYNOPSIS

GPL has the following types of instructions:

*DATA TRANSFER
-single or double byte transfers
-block to block transfers
-formatted block transfers

*ARITHMETIC
-add, subtract, multiply, divide,
-negate, absolute value

*LOGICAL -and, or ,exclusive or, shifting
*CONDITION TESTS -arithmetic and logical tests
*BRANCHING -unconditional and conditional
*BIT MANIPULATION -set, reset, and test
*SUBROUTINING -call, return, parameter fetching
*STACK OPERATIONS -push and pop
*MISCELLANEOUS
-random number generation, keyboard scan,
-coincidence detection, pattern movement,
-sound control,
-TMS 9900 subroutine linking, I/O

1.3 GPL TIMING

The GPL interpreter contains an interrupt driven service routine which is tied to the video scan. Video symbols may be moved about the screen automatically; also sounds may be generated from a sequence table.

These are of the "set it and forget it" type of instructions which free up the control program to do concurrent decision and computational operations. The interrupt also controls a software real time clock.
Each system will have a clock byte
reserved in the console ROM at location
>000C to indicate the clock rate for
that system. Peripherals may read this
byte to adjust their timing interface
to the CPU's clock combinations in
different consoles. The high nybble
contains the integer frequency in
megahertz and the low nybble, the
fractional frequency.

1.4 GPL ASSEMBLER

The TI assembler for GPL (GPLASM) is written in a mixture of FORTRAN and assembly language and is currently available for installation on 990/10 DS minicomputers. The assembler provides standard features such as creation of a list file, cross reference tables, and error flagging. A set of macros is included to help structure GPL programs; these include statements such as: REPEAT ... UNITL and IF ... THEN ... ELSE. The output of the assembler is a 990 object module.

1.5 SOFTWARE MONITOR RECONFIGURATION

The monitor code is executed whenever a system restart is required. The system parameters and control values are initialzed to default values. A default character set is loaded into the video pattern generator, making it immediately available to GPL programs. This pattern set consists of 64 ASCII characters, including the upper-case alphabet, digits, arithmetic symbols, and punctuation symbols.

The monitor is also responsible for determining the existing system configuration. The power-up monitor must poll add-on I/O peripherals and the "SOLID STATE SOFTWARE CARTRIDGE" to determine which program to execute.

The Home Computer system has been designed to be flexible and expandable. Each plug-in ROM or GROM may contain power-up procedures. These power-up procedures will all be executed allowing for expansion of the power-up routines. A power-up routine may also be replaced by another.

1.6 FOREIGN LANGUAGE SCREENS

GPL code has been included in GROM 0 to allow a plug-in GROM to "translate" the main screen, the menu screen, and the cassette DSR messages to alternate languages. The main screen and the menu screen are "translated" after the screen has been formatted in English but while the screen is turned off (only the background color is visible on the screen).

At this time, the plug-in GROM is checked for a negative version number (byte 1 of the GROM). When a negative version is encountered, a GROM routine is called at >6010 for the main screen or >6013 for the menu screen. These locations should contain unconditional branches to the routines in the plug-in GROM that will rewrite the screen in the desired language. These routines may use all of the usual CPU RAM locations (>0 through >6F) and the full facilities of the monitor and interpreter. The routines should end with a RTN instruction.

==========

ADVENTURE SOLUTIONS... ONLY READ IF YOU ARE STUCK!

DISK FIRST, from INFOCOM:

ENCHANTER

Keep in mind you must eat and drink urged to do so by the game... but YOU ARE WHAT YOU DO !
Fork go NE Outside Shack. Go Inside shack. Open oven . Take bread and jug and lantern then frotz (light spell) the lantern. Go outside shack.
Go NE ,SE, NE; at Shady Brook fill jug with water . SW,SE,SW (The evil castle is to your east but you are not yet ready to go there you don't have the key!) SW, S. Outside Hovel read scroll then gnusto rezrov (open locks... your key into the castle!) N, N, S, S, NE, NE, E. At eastern Fork go E then E and you will be...
Outside Gate memorize nitfol and rezrov then rezrov gate then E You are in! A good place to save the game! Inside Gate go S,S to Tower .
At South Hall go down Dungeon open
door then go N Cell read walls then
remove block then E Secret Passage take
scroll then read stained scroll then
gnusto exex (make things go fast) go
W then S then U South Hall drop
lantern then go E Gallery look behind
lighted portrait, take black scroll
and candle, read black scroll,
gnusto ozmoo, (survive an unnatural
death), go W, take lantern-
go E, go E South Gate go S then SE to... The Beach nitfol turtle exex turtle, turtle, follow me, NW, N Tower go U Engine Room turtle, go SE, get scroll wave at turtle, take scroll, door;Tower (go W till you get to the south hall then go up to the bedroom get in bed sleep you will have a dream get up examine bedpost, press button, take gold scroll, read gold scroll, gnusto vaxum, memorize vaxum and rezrov and nitfol, go down, go N till you get to the inside gate and drop all Inside gate Go east until you are picked up by the goons and thrown into the cell Cell ozmoo self (don't do this earlier or it'll wear out), wait - you will be taken up to the sacrificial alter and a dagger will go into your heart - have no fear your spell will not only protect you - in addition you get a useful tool...the dagger! go down to the temple, go west until you reach all of your possessions then take all.
==========

NOW FOR CASSETTE OR DISK ADVENTURERS:
==========

* *
* LA TIBBS adventure cheat's... *
* *

Scott Adam's PIRATE ADVENTURE
courtesy of...
Ye Warrior

TAKE RUM, TAKE SNEAKERS, TAKE CRACKERS, CLIMB STAIRS, PULL BOOKCASE, ENTER PASSAGE, GO EAST, TAKE TORCH, TAKE BAG, SAY YOHO, SAY YOHO, OPEN BAG, GET MATCHES, DROP BAG, GO EAST, ENTER SHACK, DROP RUM (pirate SHOULD take rum and 'Scuttle off Shortling', although sometimes he doesn't. If that is the case, then pick the rum back up, drink some, then drop it again until the pirate takes it....),
GET CHEST, DROP CRACKERS, GO WEST, GO EAST, CLIMB HILL, DROP CHEST, LIGHT TORCH, DROP BOOK, ENTER CRACK, ENTER SHED, GET HAMMER, GO NORTH, ENTER CRACK, UNLIGHT TORCH, DROP TORCH, GET BOOK, GO DOWN, GO WEST, GO WEST, SAY YOHO, ENTER WINDOW, GO DOWN, GET NAILS, GET RUG, DROP RUG, GET KEYS, CLIMB STAIRS, ENTER PASSAGE, GO EAST, GET BOTTLE, SAY YOHO, SAY YOHO,
DROP BOOK, DROP NAILS, DROP HAMMER, DROP KEYS, DROP BOTTLE, GO EAST, GO EAST, CLIMB HILL, GET TORCH, LIGHT TORCH, ENTER CRACK, ENTER SHED, GET WINGS, GO NORTH, ENTER CRACK, UNLIGHT TORCH, GET CHEST, GO DOWN, GO WEST, GO WEST, DROP TORCH, GET KEYS, DROP CHEST,
UNLOCK CHEST, LOOK CHEST, GET PLANS, LOOK CHEST, GET MAP, DROP PLANS, DROP MAP, DROP KEYS, GET BOTTLE, ENTER LAGOON, GO NORTH, GET WATER, GET FISH, GO SOUTH, GO EAST, DROP WINGS, GET TORCH, GET KEYS, GO EAST, GO EAST, LIGHT TORCH, ENTER CAVE, GO DOWN, DROP FISH, DROP BOTTLE,
UNLOCK DOOR, ENTER HALL, GO EAST, DROP MATCHES, GET SHOVEL, GET SAILS, GET LUMBER, GO WEST, ENTER PIT, GO UP, GO WEST, GO WEST, GO WEST, UNLIGHT TORCH,DROP TORCH, DROP LUMBER, DROP SAILS, ENTER LAGOON, DIG ANCHOR (you must do this when the tide is OUT, otherwise you will drown. I suggest you wear your water wings just in case you enter the lagoon while the tide is in. If you happen to enter the pool when tide is in, just type WAIT several times.....),
GET ANCHOR, GO EAST, DROP ANCHOR,
BUILD BOAT, GET BOOK, SAY YOHO, ENTER WINDOW, ENTER PASSAGE, GO EAST, WAKE PIRATE, SAY YOHO, SAY YOHO, DROP BOOK, GET MAP, GO WEST, ENTER SHACK, GET CRACKERS, GET PARROT, GO WEST, GO WEST, ENTER SHIP, SET SAIL (this time the tide must be IN. So if it isn't, just type WAIT several times until the tide does come in.)
ENTER SHORE, GO SOUTH, GO EAST, ENTER MONASTARY, DROP PARROT, DROP CRACKERS, GET DOUBLOONS, GO WEST, GO 30, DIG, GET BOX, DROP MAP, GO WEST, GO NORTH, DIG, GET BOTTLES, DROP BOTTLES (again, the pirate MUST take the bottles...), GO SOUTH, WAKE PIRATE, GO NORTH, ENTER SHIP, SET SAIL, ENTER SHORE, DROP KEYS, DROP SHOVEL, GET BOOK, GET HAMMER, OPEN BOX, DROP BOX, GET STAMPS, SAY YOHO, ENTER WINDOW, GO DOWN, DROP STAMPS, DROP DOUBLOONS, SCORE AND YOU WIN!!!!
****

SOME ESSENTIAL DISK SOFTWARE

Peter Walker's personal view of disk software.

A member who recently renewed his subscription commented that as a relatively new "expanded" member, he would like to know what the classic "essential" software packages were that he should consider getting. I can sympathise, since the disk library catalogue is now so large and with commercial offerings still being generated, that one cannot always see the wood for the trees.

Unfortunately, what is 'essential' for one person may be useless to another. For example, you won't find me suggesting you get hold of games and adventure programs, since that is not my scene at all. What follows is therefore a highly personal view of what is 'essential' for disk owning members and largely follows the types of package used in business computing.

I will therefore consider Word Processors, Spreadsheets, Databases, Terminal Emulators, Graphics Editors, Desk Top Publishing, Disk Menu/Loaders, Disk Utilities and ExBas extensions. I should also mention that once you have disks you can also explore several new languages, such as Forth, C99, Lisp, Pilot and Assembler (though MiniMemory allows cassette owners to try the latter). However I will not be covering languages in this article.

WORD PROCESSORS

I have no hesitation in recommending the TI-Writer as the premier Word Processor for the TI99/4A. Several folks have improved on the original in recent years, notably the Funnelweb version (see below) and R.A. Green's TI Writer version 4.5. Both these have their own advantages and both have significantly speeded up some of the original functions.

SPREADSHEET

Again, the original TI offering, Microsoft's Multiplan (which requires a module as well) is hard to beat. Recently R.A. Green has issued a major rewrite which has significantly speeded up the processing and can be recommended. There is also an 80 column version of the original (slow) program available for Geneve and 80 column card owners.

DATABASES

I wrote extensively about Databases in issue 24. Suffice here that I recommend two of the best. The fairware PRBASE by William Warren has an excellent screen display and many search and select facilities. The other recommendation is Inscebot's TI-BASE, now at version 3.01. This is a programmable database, not for beginners, but is very similar to such classic PC packages as dBase.

TERMINAL EMULATORS

In this issue I have written about Terminal Emulators. Clearly, the two recommended fairware packages are Paul Charlton's Fast-Term and Charles Earl's Telco. If you don't have a modem, you might still want a terminal emulator for exchanging files between computers.

GRAPHICS EDITORS

In issue 22 there was a good history of the various graphics packages that have appeared over the years for our machine. The one that I recommend is Inscebot's "TI-Artist Plus!". This has inbuilt conversion routines for other popular graphics programs. My only criticism of TI-Artist is the lack of a background "chequerboard" grid for positioning items, as used in R.L. & C.P. Davis's GraphX.

DESK TOP PUBLISHING

There are a few packages around that claim to be DTPs for the TI99/4A and I can admit to having tried none of them. The package I have, Asgard's Page Pro 99, doesn't claim to be a DTP, but has as many features as you are likely to need. It can import both text (TI-Writer compatible) and pictures (including TI-Artist instances) and arrange them on the screen with two types of characters (large and small) which can be chosen from a wide variety of fonts. A line/edging character set is also available.

DISK MENU/LOADERS

With so much good software around, it is nice to have a program that features on a menu all the 'essential' software and your own favourites, plus have the ability to load programs of various formats. This in a nutshell is what "Funnelweb" is, by the McGoverns of Funnelweb Farm. This remarkable piece of fairware software has no competitors and can be thoroughly recommended. It comes with several useful programs, such as an improved TI-Writer Editor/Formatter, an Editor/Assembler, DM-1000 disk manager (see below) and a disk sector editor. Recent releases include a package called DiskReview, which is a combination program: part disk manager, part file reader and part program loader all rolled into one. 80 Column cards are also supported. The program to configure Funnelweb to your own needs is in itself a delightful piece of 'window' based programming.

DISK UTILITIES

Included in this section is a host of differing packages that disk owners will find useful. Again, a highly subjective selection based on what I have used and find useful.

Disk Managers

TI's module based Disk Manager is a bit limited. My own recommendation is Bruce Caron's fairware DM-1000. This allows multiple selection of files for copying, moving, renaming, changing protection or deleting. It can copy whole disks quickly, print a catalog, initialise and everything else you'd want a disk manager to do.

Sector Editors

Sometimes you may want to bypass the normal disk file access methods and look directly at what is on each individual disk sector and perhaps change some of the stored information. You need to know what you are doing, since these editors have the capability of destroying information or denying access to it from the normal disk access methods. There are many sector editors around. A modified version of TI's original Disk Patch is included with Funnelweb and is very effective. There are some other types around, such as Donald Thomson's Disk+Aid which features memory viewing also and a useful sector search routine.

Specialised Loaders

A drawback of Extended Basic is its appallingly slow loader for Assembly language programs and its inability, directly, to load Program Image files. There are a number of utilities around that help overcome these difficulties. EASLOADER from .. allows Program Image files to be loaded and run from ExBAs. From the same stable comes TextLoader which allows the execution of ExBas commands from a DV80 text file including such commands as CALL FILES(2) or NEW which traditionally can only be issued in command mode, rather than as part of a program.

Another method of loading assembler routines is to "poke" them into memory using a succession of CALL LOADs. ACE from ... allows a compiled DIS/FIX 80 program to be converted into a correct set of CALL LOADs. (This can be of great use to cassette owners who have memory expansion). Another approach to the same problem is to trick the Basic loader itself to load an assembler routine into the correct location. By combining the code with the Basic program, one can create so-called "hidden code". SYSTEX by Barry Boone is a utility that allows hidden code to be produced from any assembly routine.

Archiver

This fairware routine from Barry Boone allows data files to be stored on disk in a special format that is more space efficient than the normal TI file format. Also, data compression on the information itself is performed. Thus it is possible to condense files for more efficient distribution on disk or transmission over the telephone network or for archiving backup copies.

Neatlister

A routine from Danny Michael that lists programs such that multiple statement lines in ExBas are set out one per line for ease of reading. It can also list what variable names have been used and on what lines.

EXTENDED BASIC EXTENSIONS

This category contains a very mixed bag of routines all designed to add further sub-programs to those already in ExBas in order to allow control from a Basic program of areas of the TI99/4A that one would normally need to use assembly language to 'reach'. All these packages are indeed sets of assembly language routines packaged as CALL LINK() programs so that they can be used within your ExBas programs. I cannot here describe each of them and all the routines they contain, but the following should give an idea of what they do.

1. Display Enhancement Package (DEP). From Oaktree Software, this allows 4 pages of 40 column text mode to be used with instant switching between them. A range of very powerful DISPLAY and ACCEPT functions are provided with such features as alignment to left, right or decimal point; auto-input on filled field, function key breakout and more.

2. Enhanced Display Package (EDP) From Paragon Software. As the name suggests, this is very similar to the DEP. EDP is not as strong in its accept and display functions but has a powerful windowing facility which is worth getting EDP for alone.

3. Missing Link from Harry Wilhelm & Texaments. Similar in concept to DEP and EDP, this goes a step further by producing all the screen display effects in bit map mode. There is a free demo disk to wet your appetite.

4. STAR by Michael Riccio - A set of 53 CALL LINKs that allow loading of various character sets, sounds, cassette control, colours, VDP access, screen effects, sundry control routines, keyboard detection, character definitions, disk access, text mode and string handling.

5. JBM103 a French package of unknown origin that allows CALL LINK access to bit mode graphics, including saving and loading TI-Artist format pictures.

All the above have one thing in common: despite the fact that they comprise separate routines, the packages come on an all or nothing basis. One can see why the authors have done this, but it does mean that you have to load the whole package irrespective of how many routines you want to use. The problem is that the sheer size of some of these is such that it restricts the memory space available for your program or data in the main Basic program. It should also be mentioned that some extensions to ExBas are available in the various "extended" Extended Basics, such as those by Mechatronics, Triton and Myarc. These store the extra routines in GROM of course and so do not restrict your programming.

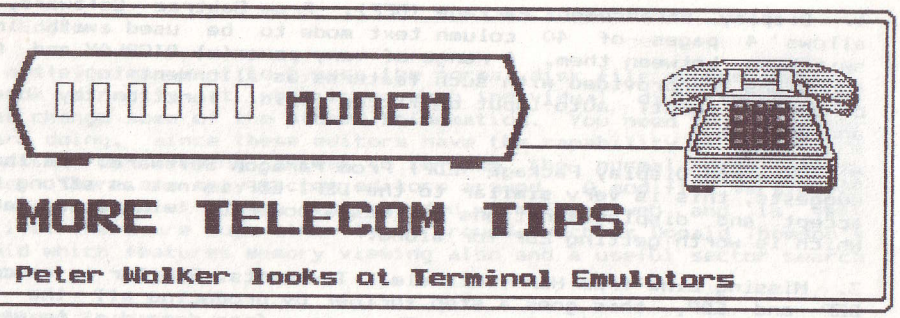
Well that's my list of 'essential' software. I realise there are plenty of omissions from the above list, such as disk catalogers, program development utilities, mailing list managers, file readers, label printers, printer set-up programs, banner printers, sorters/searchers, home budget/cheque book managers and all the sundry things people program on their machines, but it's horses for courses: I'll leave it to someone else to write about these!

Peter Walker

References: With the exception of Telco, all the fairware mentioned in this article is available from the Group's library and contact addresses for the authors are included in the documentation. The commercial software is available from:

Inscebot: PO Box 291610, Pt. Orange, FL 32029 USA
Asgard: PO Box 10306 Rockville, MD 20850 USA
Texaments: 53 Center St, Patchogue, NY 11772 USA

Stephen Shaw can give advice on ordering from the USA.



MORE TELECOM TIPS

Peter Walker looks at Terminal Emulators

"Just when you thought it was safe to return to the computer - Terminal Emulator II"

An old TI joke this, but nevertheless it points out that "Terminal Emulator" is a classic bit of computer jargon. This article will describe what they are and discuss what is available for the TI99/4A, concentrating on TI's "Terminal Emulator II" (TEII), Paul Charlton's "Fast-Term" and Charles Earl's "Telco".

Terminal Emulators are software packages that you traditionally use when accessing an interconnected, and usually remote, computer, often over the telephone network using modems. (See Telecom Tips No 1)

In the early days of computers two way communication with computers was usually performed via electromechanical teleprinters, which had long existed serving the needs of telecommunications systems such as telegrams and telex. My own first experience of a computer 'hands on' (as opposed to batch processing by computer operators) was using the sturdy Teletype Corporation ASR33 teletype. This was in 1968. The ASR33 worked at 10 characters per second, which corresponds to a transmission rate of 110 baud or bits per second. For the technically minded 110 = 10 x 11 bits (1 start bit, 7 information bits, 1 parity bit and 2 stop bits). See TI*MES 20 for description of RS232 and asynchronous transmission.

Later on, screen based terminals arrived, the infamous VDU: visual display unit, using a CRT (Cathode Ray Tube). With a screen one can achieve more effects than one can on a teleprinter with its continuous roll of paper. For example, the host computer can direct the terminal to erase the screen or position incoming text at a particular part of the screen, using specific 'control codes' it sends, ie codes which are not ordinary alphanumeric 'printing' characters. As time went by, these types of terminal evolved to the point where a very sophisticated set of control codes were developed to perform a range of special functions, such as getting the terminal to keep part of the screen fixed while the rest scrolls or directing text to an associated printer. In the reverse direction, control sequences are sent from the terminal for a wide range of special 'function keys' on the keyboard. Unfortunately, there is no one universal standard for terminals or these control codes. Some, such as DEC's VT100 protocol, gained wide acceptance by sheer market influence, while IBM's 3101 became, I believe, the de facto North American ANSI standard.

When micro-computers started to appear on the scene in the late 1970s, it soon became clear that linking these to larger mainframe computers would be of great interest and utility. The simplest way this could be done was to make the micro-computer attach to the computer just as a 'dumb' terminal would do. (Actually it's interesting to note that the term 'dumb' to describe terminals only emerged as micros took hold. In reality many 'dumb' terminals are quite complex and intelligent and often use microprocessor chips to drive their functions. But I digress!). In order that a micro can connect as a terminal would do to a computer, it needs a program that allows it to look like or 'emulate' a terminal, including all the special control codes for screen display etc. Such programs are called Terminal Emulators.

In their simplest form such programs must allow any key press on the keyboard to be immediately transmitted over the RS232 output port, while any incoming characters should be displayed on the screen, either scrolling down the screen as a simple ASCII terminal or teleprinter would do, or positioning text according to any control codes received. Most terminal emulator programs do more than this however. The following describes some commonly found features:

REVIEW (or 'window back') - this feature allows you to look again at text that has scrolled off the top of the screen.

SCREEN DUMP - allowing you to print the screen contents to a printer.

Before you can Review or Screen Dump, you usually have to ensure the screen is 'frozen' and no new text is received. To do this, the Terminal Emulator sends an X-OFF signal (ASCII 19) to the host requesting that no further info is sent. After screen dump or review is complete, X-ON (ASCII 17) is sent. This X-ON/OFF protocol, where available, is the normal means by which one computer tells another that it cannot receive more information. Its more normal use is to regulate the flow of information to ensure that any input buffers do not overflow.

PRINT SPOOLER - this allows the incoming text to be directed not only to the screen but to a printer also. Since the printer will almost certainly work slower than the incoming text, it is first stored in a memory buffer or 'spooler'. If the incoming text is sporadic, for example, a page being sent in response to a command, the printer will usually catch up and in so doing clear the buffer. Should the buffer however near exhaustion then the terminal sends X-OFF until the printer has cleared the buffer back to a lower level. A good emulation program should ideally strip out any control codes used for screen display since they might interact with the different control codes used for printer control.

LOG TO DISK - this allows a permanent record to be kept of the incoming text. The usual method is collect the text in a buffer until it is full. Then X-OFF is sent while the buffer contents are written to disk. It is usually possible to switch the log on and off so that only wanted text is stored. Log to disk allows a complete ASCII file sent from the host to be captured to disk.

FILE TRANSFER - This needs a little bit more explanation. We have seen above how Log to Disk allows the transfer of an ASCII text file from host to terminal. This is a file "Download". You may wish to send an ASCII text file to the host in one go: "Uploading". Left to its own devices, it is quite simple for the terminal to transmit the text file

from disk to the host at the maximum speed determined by the line transmission speed (eg 300baud, 1200baud or whatever). Such a rapid transmission could give the host problems. If the text is received faster than it can cope with in its input buffer, it may send X-OFF to the terminal to pause the transmission until it can cope again. However, in order that systems not employing X-ON/OFF can be sent to as well, other file upload arrangements may be provided in order to 'pace' the text transmission to what the host can cope with.

Such arrangements might include:

-Limiting the speed of character transmission
-Waiting between text lines (timed pause or manual pause)
-Waiting for a fixed prompt character from the host, eg ">"
-Waiting after a character is sent for the host to 'echo' it back

So much for ASCII downloads and uploads. The problem with straightforward file transfers is that, unlike manual typing, you may not notice any bit errors caused by line noise, since transfer is so fast. Because of this, most Terminal Emulators include a "Error-corrected" protocol for transmission of files between terminal and host. It has to be said that this is where the term "Terminal Emulator" loses its meaning, since such protocols are never found on 'dumb' terminals, but are more characteristic of host to host communications. Lets not quibble about terminology however: there are many other hurdles to overcome. The difficulty is that, like the terminal control codes mentioned above, there are many and varying methods of error-correcting file transfer protocols in use. These include XMODEM, YMODEM, COMPUSERVE, KERMIT, FIDO etc etc. And just to really confuse things, TI invented their own for the Terminal Emulator module, thus limiting its use to systems designed specifically for the TI99/4A. (This is "par for the course" however: the TEII module also sports its own unique set of control codes for screen & cursor control. You won't find these listed in the manual however, you need the full TEII Protocol Manual.)

I seem to have digressed again! Back to file transfers. Error correcting file transfer protocols all work on broadly similar principles. Data is collected into blocks and transmitted along with a trailing sequence of 'check-bits'. The receiving computer calculates what the check bits should be from the data received and compares them with those receieved. If all is OK, the receiving end signals that the block was received OK. Since another incoming block might already be being sent, a block sequence number is necessary to sort out what block is being acknowledged. If an error is detected then the transmitting computer is requested to retransmit the block, and often followed by a retransmission of any subsequent blocks to simplify the ordering of data at the incoming end. Some protocols however only allow a block to be transmitted if the previous has been positively acknowledged. If a set number of repeat transmissions fail, then the transfer session is aborted.

Two methods are used to generate check bits. The Checksum method sums the bits transmitted and sends its value (or some derivation of it) as the check bits. This is a more sophisticated, but essentially similar method to 'parity' used on individual characters. The alternative check bits method is Cyclic Redundancy Check or CRC for short. With this method a more complex algorithm then purely addition is used on the data block. The method is often called a cyclic polynomial and is too complex to describe in detail here. Suffice to say, it is better

at detecting multiple errors in the block that Checksum might miss.

When systems are transferring files using any of these protocols, no text appears on the screen. (As I said above, these protocols have nothing to do with strict terminal emulation). Instead the screen will indicate how many blocks have been transmitted, how many retransmitted and other progress information according to what protocol is being used.

That then is file transfer. Lets return to features one would find in a Terminal Emulation program.

The balance of features are dominated by configuration options used to set the terminal to a particular arrangement. These include:

TERMINAL OPTIONS - Speed of transmission, parity, local echo on/off, handling of CR/LF, terminal width. (Local echo on/off needs some explanation. It is often confusingly called half or full duplex, but has nothing to do with whether the modem transmission is working in half or full duplex (see TI*MES no 20). Normally the terminal does not display what you type since the host computer echoes back what you type. If it doesn't then half duplex is used where the local key strokes are displayed directly on the screen.)

SCREEN OPTIONS - Display width, screen colours.

MODEM OPTIONS - which RS232 port.

PRINTER OPTIONS -which port (PIO or RS232), speed & parity (if RS232).

There are likely to be many other features beyond those I have described here. Every Terminal Emulator has its individual features. TI's own TEII can actually speak what is on the screen for example.

Here is a brief review of the main features for the 3 best known terminal emulators: TE2, Fast-Term and Telco.

	TE2	Fast-Term	Telco
Emulates	TE2	ADM3A	ADM3A, ANSI, D410, VT100, VT52 HP2392
Review	Yes	Yes	Yes
Screen Dump	Yes	Yes	Yes
Print Spooler	No	Yes	Yes
Log to disk	No	Yes	Yes
File Transfers	TE2	ASCII TE2	ASCII Xmodem, TIBBS Xmodem, Ymodem Compuserve B
Speeds	110 300	110 300 600 1200 2400 4800 9600 19200	Any + also set speeds: 300 1200 2400 4800 9600
Terminal Widths	40/80	40/80	40/80
Screen Width	32/40	40	40/80 (80 col card req'd)

The last two items probably bear a little bit more explanation, since the difference between terminal and screen width may not be obvious to all. Many host systems assume they are communicating with an 80 column terminal. On the other hand, some allow special log-on options for different terminal widths. So if a 40 column option is available then the TI is quite OK. If the host assumes 80 column then the TI can store incoming lines as 80 columns and can then 'window across' (like TI-Writer) to view all the 80 columns on its 40 column screen. (Alternatively, the 80 columns can wrap round the screen using 2 lines.)

That then is Terminal Width. Screen Width is the actual width of the display screen. TE2 has both TI text and graphics mode, while Telco can use a true 80 column width with the Geneve or 80 column card. Both TE2 and Telco have additional features in case you TV monitor doesn't like displaying the full 40 column width.

As can be seen, both Fast-Term and Telco are very powerful programs. The differences between the two are many-fold. Fast-Term changes options by simple function or control key presses and the effect of the key press is written directly on the text screen. In Telco, key presses are noted either by changing status marks on the status line at the bottom of the screen or by bringing up a full screen menu for selection of more complex items. Telco is a much larger program and cannot fit on one SSSD disk. This doesn't prevent its use by SSSD owners however. Telco is so large that it uses the Program Overlay technique, just like Multiplan and TI-BASE do. In this technique, programs are left on the disk and pulled into RAM only when invoked. When the RAM space is needed by another program the oldest (usually) is erased. Since each of the Telco 'emulations' and file transfer types is a separate overlay program, you only need the ones you want to use on your disk.

Telco has a wide range of extra features compared to Fast-Term. Amongst these are a Dialler (Hayes compatible), Macros (for frequently sent strings eg IDs, passwords), a Conference mode (a split screen shows both sides of the text transmission), an Editor and as mentioned above, an 80 column display mode.

On the face of it, the TE2 module has little to offer compared to its main rivals. It can work without disks of course and can speak what it receives. However it does have some unique features specific to the TI99/4A which are briefly mentioned, but not explained, in the TE2 manual. It can receive data in both 40 column text mode and 32 column graphics mode. It can transfer to, and store in, the connected TI99/4A: character definitions, screen colour, colour sets and sound lists. It can make the connected TI99/4A display characters 128-255, play sound, speak words (with or without displaying them), speak allophones (speech components) and speak sentences by associating code numbers with words. It can also interrogate the status of the remote system. All this, and the TE2 file transfer protocol, is explained in the TE2 Protocol Manual. This volume is about as easy to understand as the Editor/Assembler Manual is on first reading: this is a polite way of saying that is is somewhat impenetrable! In a later article I hope to take you step by step through the special TE2 features.

In the meantime, try this: connect two TI99/4As, one with TE2 and one with an emulator that can send ASCII text. Send the following sequence to the TE2:

```
␛G■␛(! AiRtbAFiCqPBg(!J@iRtE@J_ovp@DP@␛)␛G■␛(" ␛)

␛ = ASCII 27  ■ = ASCII 127
```

If you don't have two systems, but do have a modem, then phone me and I will send the text string to you over the network. Alternatively, why not get together with other modem owning members to experiment.

It should make the TI99 ring a few bells!!

Finally, before ending this article. I should acknowledge the existence of several other terminal emulation programs for the TI99/4A. In my last article I described the Prestel emulation program. Additionally I have tried or heard about:

COMMTY - otherwise known as COM99 by David Ramsey: interesting in that this offers the split baud rate 1200/75 speed, not found on Fast-Term, Telco or TE2. However, it seems to me to possess a strange bug that makes the left hand column invisible, irrespective of how your monitor is set up. Please note that although it has 1200/75, it does not have the Prestel emulation.

TE3 - which is in our library. An incomplete and unfinished product designed to replace TE2.

TEX - originally in the old TI Exchange library under the title MODEM (program B1). It offers speeds 110-9600, review, logging (but in unfriendly DIS/VAR 128 form), TE2 (but not ASCII) file transfers.

TE1200 - I haven't used this package. It was written by John Koloen and distributed by Softmail. It was reviewed in Micropendium in August 1984.

If anyone knows of other terminal emulators, please let me know.

Peter Walker

RAMBLES

for TI*MES 30 / September 1990

Greetings and welcome to another Rambles, a collection of odds and sodds related to the TI99/4A. Your comments and requestes and queries are very welcome (as is an SAE if a direct reply is wanted!) please write to me at:
10 Alstone Road STOCKPORT Cheshire SK4 5AH
which is also the address of the disk library.

First an important update on ASGARD- a major publisher of software for the TI, who nevertheless managed to get all my four orders wrong and therefore carries a warning tag. In past issues I indicated you could obtain Asgard programs via Comprodine, alas this is no longer the case. You could try MICROpendium advertiser Jim Lesher, at 722 Huntley, DALLAS, TX, USA, 75214 (Telephone 214 821 9274), who seems to offer a huge stock of both current software and older modules, but I have not dealt with him so cannot speak for his service.

I received a letter from a Chuck Bower offering to supply modules, but my initial request for a quotation on an order has not been replied to. Reliable sources of supply do seem to be drying up!

==========

ANCIENT HISTORY

Our local library regularly sells off unwanted books and recently cleared their old computer books (including Townsends TI99/4A book). I picked up two product guides from 1982 and 1984.
I bought my first TI back in 1981, so the '82 book is a good record of what I had to choose from- the TI was just changing from 99/4 to 99/4a. Comparing the 1984 and 1982 books, only the ZX81 was still available two years later- pick any other computer in 81 or 82, and you would have been orphaned by 84!
Here was the scene in 82:
The TI99/4A was selling (for a day or so, the price started to crash downwards from here on!) at 399 pounds! Seen as competition, and producing a price reduction later, was the 4k VIC20 at 185 (22x23 char display!). Here are the more well know 1982 consoles in price order:

5k VIC20 (22x23 display) 185.00
8k OSI C1P with 8k basic, 250.00
12k Acorn ATOM with only intger basic and mono o/p. 250.00
16k Video Genie EG3003 with 12k basic and 64x16 display. 289.00
4k AIM 65 with a 20 x 1 (One) display! and 1k basic, 315.00
32k Acorn Proton (forerunner of BBC-B) 335.00
8k Commodore Pet 4008 with 8k basic. and mono monitor 378.00
32k ATARI 400 with rubber keyboard. 395.00
16k TI99/4A for 399.00
4k TANDY TRS80/1 incl tv and recorder. 430.00

By the time the 1984 book came out, the TI99/4A had been withdrawn, and the Dragon, which Boots had switched to selling, and which some TI owners rather too quickly switched to, had just been withdrawn as the liquidator was called in!

Of the 1984 computers, apart from the price drops and improved specs, it is interesting to look around the computer stores and see that some are still commercially supported, and indeed, much to Commodores chagrin, one the C64 is still earning large sums of money, six years on! Properly supported could the TI have kept going as the C64 has? You betcha! The 1984 computers...
A real curiosity to start- anyone remember the 4k Tandy MC10 at 70.00- it had no program editing facilities!!! If you wanted to alter a program after typing it in, you had to type it in again. Made debugging a chore!

48k Spectrum 130.00	20k CGL (Sord) M5 150.00
16k Colour Genie 168.00	48k Oric Atmos 170.00
48k Alphatronic 170.00	16k Spectravideo SV318 186.00
64k Commodore 64 199.00	32k Acorn Electron 199.00
48k Lynx 225.00	64k Amstrad CPC464 incl grn monitor 229.00
64k Dragon 225.00	64k Atari 800XL. 250.00

At this time you could have picked up a TI99/4A for around 50 or less.
There were others which did not last long enough to make it into either book, and of course others which were well outside this sort of price range.

==========

Update on the scan of George in last issue- and the cartoon scan in this issue, timed to coincide with a new series on BBC TV- Star Trek, The Next Generation.
The scans have been done by Ray Kazmer of Sunnyvale, Californaia, and the procedure now used is as follows:
Scan using IBM PC clone and clean up using FIRST PUBLISHER. Version 3 allows perfect ratio on the PC! Save in MAC format.
Transfer to TI99/4A using PC-TRANSFER, which despite a habit of scrambling pictures most of the time, is said to be faster than using a direct wire hook up with comms programs on both machines.
And once on the TI, the MAC format pics can be printed using PIX PRO and transferred to other formats such as TI Artist if size permits.

Did you ever want to make an Extended Basic program unlistable and unalterable but still playable and saveable? This is a bit of fun programming to see how to exploit the way our computer stores its programs!
Original programs from Tidewater Newsletter, Ken Woodcock- I took them from WORDPLAY June 1990:
Your computer stores all the basic program lines in one block, and then in a separate block, a line number table, which identifies where in memory each line number can be found- the line number table is in numeric order, the actual program lines are stored in the order you enter them, which may not be in numeric order!
The first byte of the Line Number Table tells the computer how long the line is in bytes- this is only used to list the program or to handle editting functions, when running a zero value byte terminates each program line and the LNT is only used to locate the start of the program line.
To make a program unlistable all you need do is change all the length bytes in the LNT! The following program will set all length bytes to zero! and requires XB and 32k.
First load your program. If it uses lines 1 and 2, resequence it!
Then MERGE in (or type in) the following code: (PTO)

```
1 CALL INIT :: CALL PEEK(-31
952,A,B,C,D):: SL=C*256+D-65
539 :: EL=A*256+B-65536 :: F
OR X=SL TO EL STEP -4 :: CAL
L PEEK(X,E,F,G,H):: ADD=G*25
6+H-65536 :: PRINT *256+F
2 CALL LOAD(ADD-1,0):: NEXT
X :: STOP :: !@P-
```

Now RUN the amended program. Now in command mode type in 1 [enter], 2 [enter] to remove the extra lines. Now save using a different file name if to disk or a different tape if to cassette!!!
And you can now RUN the program, no trouble- try it.
BUT... what happens if you try typing EDIT 100 or LIST?

What... you saved the amended program over your original? Not to worry you can get it back.
We could alter all the length bytes to the maximum possible, which would certainly allow the program to be LISTed again (change the value 0 in line 2 above to 255), but editting could still be problematical, so lets reset the length bytes to what they should be!
Looking for a zero byte is a start, but not the end, as a zero byte may also occur in certain cases in the program line. The simplest thing to do is to look for a zero byte, then look at the line number table to see if the value obtained is an actual start of a program line. So lets do it...
First load the corrupted program- you cannot list it to see if it uses lines 1 to 6, so RESequence it for safety! then MERGE in the following lines:

```
1 CALL INIT :: CALL PEEK(-31952,A,B,C,D):: SL=C*256+D-65539 ::
EL=A*256+B-65536 :: FOR X=SL TO EL STEP -4 :: CALL PEEK(X,E,F,
G,H):: ADD=G*256+H-65536 :: PRINT *256+F
2 I=1 :: CALL PEEK(ADD-1,V):: IF V THEN 6
3 CALL PEEK(ADD+I,V,W):: IF V THEN I=I+1 :: GOTO 3
4 FOR Y=SL TO EL STEP -4 :: CALL PEEK(Y,E,E,E,F):: IF E*256+F-
65536=ADD+I+2 OR =0 OR ADD+I>-3 THEN CALL LOAD(ADD-1,I+1):: GO
TO 6
5 NEXT Y :: I=I+1 :: GOTO 3
6 NEXT X :: STOP :: !@P-
```

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

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

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


```
* PROGRAM TO REENTER LINE
* LENGTH VALUES TO A BASIC PROG
* author wishes to be anon
* FROM TISHUG JULY 90
*
* RUNS ONLY FROM XB+32K
* load corrupted program into xb
* CALL INIT
* CALL LOAD("DSKn.FILENAME")
* CALL LINK("UNFIX")
*
       DEF  UNFIX
       AORG >2500
UNFIX  LWPI USRWS
       MOV  @>8330,R1
       MOV  @>8332,R2
       C    R1,R2
       JHE  FIN
*
       INC  R2
       S    R1,R2
       SRL  R2,1
       CI   R2,BUFMAX
       JGT  FIN
*
       SRL  R2,1
       MOV  R2,@BUFLEN
       INCT R1
       LI   R4,BUFF
*
LOOP   MOVB *R1+,*R4+
       MOVB *R1+,*R4
       DEC  R4
       DEC  *R4
       INCT R4
       INCT R1
       DEC  R2
       JNE  LOOP
       LI   R5,>FFE8
*
AGAIN  MOV  @BUFLEN,R3
       LI   R2,ZERO
       LI   R1,BUFF
       DECT R1
*
NEXT   INCT R1
       C    *R1,*R2
       JL   SKIP
       MOV  R1,R2
*
SKIP   DEC  R3
       JNE  NEXT
*
       MOV  *R2,R6
       MOV  *R1,*R2
       S    R6,R5
       DEC  R5
       SWPB R5
       MOVB R5,*R6
       MOV  R6,R5
       DEC  @BUFLEN
       JNE  AGAIN
*
FIN    LWPI >83E0
       CLR  R0
       MOVB R0,@>837C
       RT
*
USRWS  BSS  32
BUFLEN BSS  2
ZERO   DATA >0000
BUFMAX EQU  >1800
BUFF   BSS  BUFMAX
*
       END
```

A great deal faster! This machince code is for loading with ExBas ONLY! Why do we need to have the length right for editting? 'Cos if the length byte is 255 bytes, when you delete (or edit) the lines, the computer will delete 255 bytes, and if the line is shorter, you are going to lose other data, probably partial lines, which will be very sad!

Oh yes... if the corrupted program comes from elsewhere, after making it listable, take a look for any lines that look like our first two lines above, if they are there you need to remove them or they will make the program unlistable once more!

MODULE REVIEW: TRIS- Famous Game!
====== Asgard Software, 1989. Main author Jim Reis.

Who cannot know this game? The TI module is a perfectly good emulation, in which tetromino shapes fall, can be rotated and dropped, and must be formed into a solid horizontal line. You score by how quickly you line the shapes up and drop them into place. Game ends when the shapes stack up to the top due to your having left empty spaces in horizontal lines.
This is very similar to the Hulpke disk version, except that you may not amend the "next shape" (which is optionally displayed in the module version) and the module version benefits by automatically increasing in speed as play progresses- Hulpkes version plays constantly at the same level.
A good module for unexpanded owners who cannot play the Hulpke disk version, this is also a good module for all games players. Music and sound are credited to Ken Gilliland, Bruce Harrison, Jim Reiss. HOW GOOD IS IT? My wife -Cathy- has not played a module game before- she is hooked on this one, belting the living daylights out of a joystick and not infrequent *expletive deleted*s! My six year old doesn't get too high a score at the easiest level but still likes a go. I play it. Look- if you ever play games this is one for you. If you have never played a game on your TI, this is still one for you. Buy it. OK!
Recommended.
==========
GAME REVIEW- DISK- BALLOON WARS- 1985. ASGARD SOFTWARE
Written by John Morrison.
Quite an old game this one, I've seen it advertised but not reviewed. Its an ExBas game, but uses disk data files, so is not suitable for tape.
My disk was quite unusual, having a disk jacket a trifle floppier than the disk inside it. I was surprised when my drive read it (and copied it onto a more substantial disk!). The program is list protected but most disk owners know what to do about that!
Your task is to pilot a hot air balloon from screen left to right across several screens, using fuel as you go! There are enemy troops below with a nasty habit of firing at you- you may avoid their shots and you may bomb them. On some screens, after removing the enemy, you may land for repairs and restocking.
The documentation is in error- it is joystick down to release a bomb, and joystick up to drop a snadbag (not vice versa).
This program is not exactly sophisticated, and the graphics are very simple, but it is an interesting game which presents some simple strategic decisions to be made. If you like a game to can get into quickly, you'll like this one.
==========
UTILITY REVIEW- TI BASE VERSION THREE- DISK. TEXAMENTS. US$25+ $8 p and p.
We have already covered earlier versions, this one is fully compatible with earlier one but spreads it wings in many new directions!
Of particular interest is the ability to place command files into VDP memory- and save and load them as memory images- for really rapid operation, more so if you are using a ram disk for the actual data! There is also the availability of using this option to create macros- making up your own commands from the ones supplied.

The report generator is possibly misnamed, it is a database of command files, and while it can certainly be used to produce screen and printed formats, it is not limited to that use. The major limit would be the number of commands you can fit into the database, but with the macro facility already indicated above, this additional facility gives the programmer considerable power to hide TI Base behind a fast and friendly user interface- the user need never know TI base is in use, apart from the loading screen!
Inverse characters are available, but the documentation is incomplete. You can turn all text inverse with the command SET INVERSE ON but you may also mix normal and inverse text by leaving SET INVERSE OFF and inserting characters with ASCII codes 128 higher than usual- eg an inverse A is character 193 instead of 65. These characters are used on Epson printers as the italic set. You cannot enter a character 193 into a command file using TI Writer, you must use the TI Base editor, which has another new option, familiar to PC users as the ALT key: hold down CTRL and key in the three numbers of the character you want (eg 193) and that character will appear on screen.
(As the TI Base editor is limited to small command files, we have in the disk library a utility to merge such command files into longer ones).
Of course, as a program/language becomes more powerful it tends to become harder to use, and indeed, to become underused as few bother, but TI Base can be used without command files at all, for the early owner, leading upwards to a very professional database presentation for the more experience programmer who wishes to impress (very often its the experienced programmers who leave the stitches showing!).
If you want a database, this is the one! Can work with one SSSD disk drive, works better with two disk drives, and is happy with ram disks or hard drives.
==========
MUSIC TAPE REVIEW: THE FUNNIEST COMPUTER SONGS Various artists.
US$12 inc p&p from Vince Emery Productions
P O Box 460279, San Fransisco, CA, USA, 94146.

Not a mention of a TI99/4a on this but well worth having if you are a computer phreak, and essential if you are a computer pro! Most of the tracks on this tape are in the folk/blues tradition, and are well presented and recorded, making for fun listening. Tracks include...
HALs song... you need to have seen the film 2001 to appreciate this one. It ends with "I'm ready for my first leson"- remember now?
Please, Mr Compatibility... it helps to know a little about computers to appreciate the humour here, like the request to connect a printer using the parallel port via a co-ax cable. Dont understand that? Never mind there are other tracks...
I built a better model than the one at Data General- the music here is from Gilbert and Sullivan, "He is the very model of a modern major general"-was that Iolanthe?- a high tech patter song. Try it for yourself!
Stuck Here, Threes, and S-100 Bus. I'm a Mainframe, Baby- blues style.

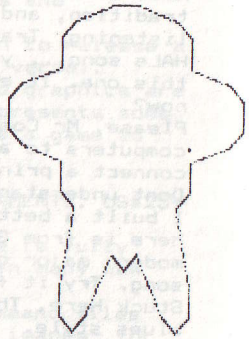
Uncle Ernies Used Computers Babbages Birthday Bargain Bash- a more modern patter song.
Mushrooms... you know, keep em in the dark!
Engineers Rap- this one is a bit like a thorn, being totally different to any other track, yes your actual computer rap!
Killer Byte Blues...more blues.
Do it yourself (this had Cathy-my wife-and I distinctly laughing. A different way to own a computer just like the ones IBM makes...
(Sit in the LOTUS position to hear this, make yourself WORDPERFECT and do not dBase yourself. Ready... then its 1-2-3...).
This is a good fun tape, recommended.
==========

... I received the above tape rather by accident,my initial contact with this supplier being for a STAR TREK (UNOFFICIAL) cassette at the same price, which has rather more American humour on it, of interest to serious trekkies but not nearly as good to a Brit listener as the Computer tape!
+++

In the last issue I indicated that I may be reviewing here a Fractal book by Clifford Pickover. Now... fractal programs are long running programs, this book has lots, and I am meantime buried in lots of good fractal ideas from the magazine FRACTAL REPORT, so this book review is held over, BUT if you are interested in fractals, this book is for you. One tiny error- the book has an extensive bibliography including lists of fractal / computer related magazines. Dr Pickover publishes one, and it is suggested the reader write to him for a free copy. Then there is a note to write to him c/o his AMERICAN publisher, whose address is NOT given. Fear not, I have his address, and if you buy the book and like it I can pass his address on to you! More next issue!
========

An issue of Rambles without fractal programs would be amiss, so overleaf are some Fractal programs for you. There is a program which gives an audio representation of bifurcation- see page 23 of TI*MES #27 for first Basic listing! (Under CHAOS!).

Then we have a program which draws the little character alongside --> take a look at him a work out a program to draw him! Can't do it? Look at the listing. Odd! Variables X & Y can take almost any value to start- high initial values mean you need a lower multiplier for A & B.



And finally, a program that draws wallpaper, composed entirely of circles- start with a SIDE of say 20 and gradually work up to say 2000! The pattern will recede and different detail will become apparent.

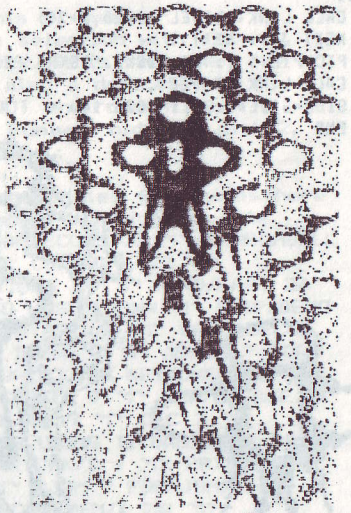
FRACTAL REPORT is ten pounds for 6 issues from Reeves Telecommunications Laboratories Ltd West Towan House,Porthtowan, TRURO, Cornwall, TR4 8AX - well worth having if these little progs intrigue you!

From an article BIFURCATED SOUNDS by Leon Heller in Issue 10 of FRACTAL REPORT. Based on a book by Becker and Dorfler regarding presentation of values generated by a Feigenbaum type system... what the heck, a DIFFERENT way of producing random sequences of tones instead of cycling round CALL SOUND(-100,110+RND*1000,4):

```
100 ON WARNING STOP
120 RANDOMIZE
140 CALL CLEAR :: PRINT "ANY KEY FOR NEW PATTERN": :
160 X=RND :: PRINT "X";X
180 R=3.2+RND :: PRINT "G FACT";R:"IF >3.57 PATTERN IS CHAOTIC"
200 FT=1000
220 FB=111
240 EF=360+230*RND
260 PRINT "EF";EF
280 D=70+RND*100
300 PRINT
320 X=R*X*(1-X):: N=EF*X :: IF N>550 OR N<56 THEN 140 ELSE CALL SOUND(-D,N*2,6*R
ND)
340 CALL KEY(0,A,B):: IF B=1 THEN 140 ELSE 320
360 END
```

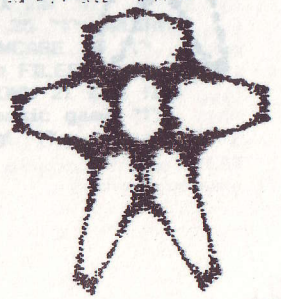
Also from Issue 10 of Fractal Report, by Tom Marlow, based on R Denaney, requires THE MISSING LINK as it stands but readily converted to any utility allowing bit map graphics...

```
100 CALL LINK("CLEAR")
110 ! H=240 :: V=180
120 X=8.56
130 Y=3.76
140 FOR L=1 TO 5000
150 NX=1-Y+ABS(X):: NY=X :: X=NX :: Y=NY
160 A=100+X*8-Y*8
170 B=70+X*8+Y*8
180 CALL LINK("PIXEL",A,B)
190 NEXT L
200 CALL LINK("PRINT",180,180,"END")
210 GOTO 210
```



or try

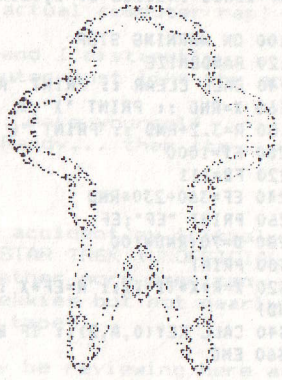
```
100 CALL LINK("CLEAR")
110 H=240 :: V=180
120 RANDOMIZE
130 REM
140 X=9*RND
150 Y=X+2*RND-RND*2
160 CALL LINK("PRINT",2,20,STR$(X)&"!"&STR$(Y))
170 FOR L=1 TO 3299
180 NX=1-Y+ABS(X):: NY=X :: X=NX :: Y=NY
190 A=100+X*7-Y*7
200 B=60+X*7+Y*7
210 CALL LINK("PIXEL",A,B)
220 NEXT L
230 CALL LINK("PRINT",180,180,"END")
240 RANDOMIZE
250 GOTO 140
```



more on next page.....

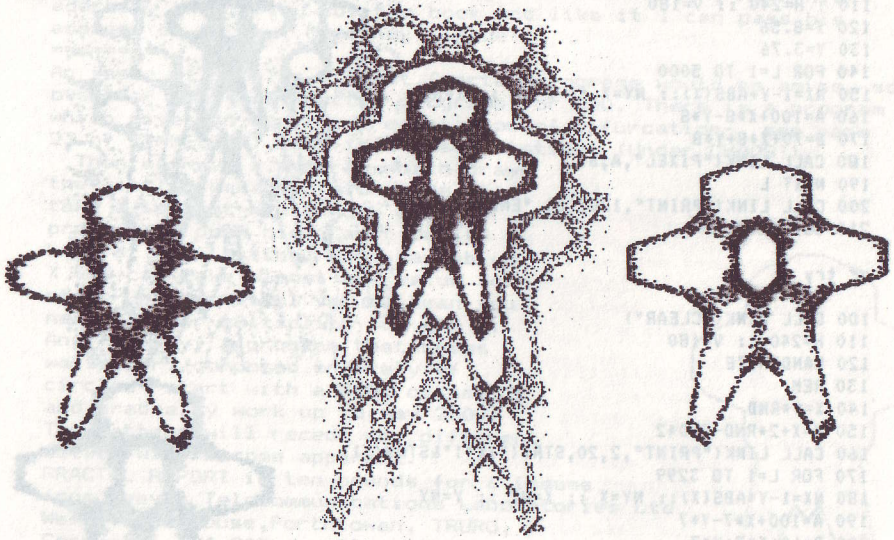
or

```
100 CALL LINK("CLEAR")
140 X=-.100000000001 ! needs the 0000001 bit!!!
150 Y=0
180 FOR L=1 TO 5299
190 NX=1-Y+ABS(X):: NY=X :: X=NX :: Y=NY
200 A=100+X*14-Y*14
210 B=60+X*14+Y*14
220 CALL LINK("PIXEL",A,B)
230 NEXT L
240 CALL LINK("PRINT",180,180,"END")
260 GOTO 260
```



And finally a little wallpaper...

```
100 REM CIRCLES
110 REM JE CONNETT/PWH MOON/S SHAW 1990
120 SIDE=20 ! anything from 10 to 2000 but
            ! avoid multiples of 150!!!
130 REM
140 CALL LINK("CLEAR")
150 FOR I=1 TO 150 :: FOR J=1 TO 150
160 X=I*SIDE/150 :: Y=J*SIDE/150 :: C=INT(X*X+Y*Y):: D=C/2 :: IF D-INT(D)>.1 THE
N 180
170 CALL LINK("PIXEL",I+20,J+20)
180 NEXT J :: NEXT I
190 PIC=PIC+1 :: A$="DSK2."&STR$(PIC)
200 CALL LINK("SAVEP",A$)
210 SIDE=SIDE*1.2 :: GOTO 140
220 END
```



DISK LIBRARY REPORT...

Write:
Stephen Shaw. 10 Alstone Road, STOCKPORT, Cheshire, SK4 5AH

Copying costs: You send blank disks plus one pound per side copied then add a flat one pound to cover packing/carriage.

>UTIL24 amongst other goodies has...: the PIO printer test from TI*MES which in a running XB program tests to see if the printer is connected and ready, without hanging the program up; a program to print PAGE PRO saved pages with a denser print; a GLOBAL DISK SEARCH utility which will search a specified disk of DV80 files looking for a specific word (or phrase) and when it finds it in a record, provides a print out of the complete record (or records- eg lines!) with file name and line number(s). And a turbo copy program for TI disk controllers-note the warning & do not use to initialise double sided disks! It TRACK copies rather than file or sector copies. And a utility to merge the 40 column command files produced by TI BASE- especially useful for Version 3- the 40 column editor only allows 50 line files but the database program can use longer ones. ALSO...
the Miami User Groups BOOT utility-this is another "environment" shell, giving you a menu selection with disk catalogue and text read/print capability. It is very very neat-this version is dated 12.2.89... also a dedicated videotape database is added. (The BOOT environment now resides in my DSK1 with T Shell while Funlweb hides in disk 2, giving a very neat system to work with).

>TI*MES PROGRAMS- short programs and utilities from times-here is the file list with TI*MES issue number following the file name: AUTOGRAPH ?"¦" BOMBER 29"¦" CHANGECURS 21 "¦" CHURCHBELL 12 "¦" CLEARALL 26 "¦" COL/COMB 16 "¦" CORNWIPE 22 "¦" DEF/NSUB 26 "¦" DEFAULT 26 "¦" EQUATIONS 24 "¦" FIND/LAST 26 "¦" FLASHDATA 23 "¦" FLIP 21 SEE ALSO UPSIDEDOWN "¦" FLIP/DEMO 21 "¦" FONTMAKER 16 "¦" FRACMY ? FOR MYARC XB "¦" GARBAGECOL 20 "¦" GET/KEY 25 "¦" HSCROLL 25 "¦" IKEDA ? "¦" JBMGR ? FOR JBM103 GRAPHICS UTILITY "¦" KALKULATOR 27 "¦" KEY/CHECK 14 "¦" KEYDISPLAY 17 "¦" LABELS 24 "¦" LET/SPRITE 7 "¦" LOW/UPCASE 16 "¦" NOISE 21 trick program! "¦" NUM/COLOR 19 "¦" OLC ? "¦" PRK/DV80 27 FOR PRK OR STATS MODULES USING TI BASIC "¦" PRTCHK + "¦" PRTCHK/1 + "¦" PRTCHK/A + "¦" PRTCHK/B + "¦" PRTCO all 25 "¦" PUTAT/1 25 "¦" READ-D/80 24 "¦" RJBM ? FOR JBM103 UTILITY "¦" RMXB ? FOR MYARC XB "¦" SCRNCDEM/X 25! "¦" SCRNCOLR/X 25! "¦" SL/CALCU 27 "¦" SPRITEMOVE 25 "¦" SQUIRMY 20 "¦" ST$REPLACE 25 "¦" STAR(MXB) ? "¦" TI/LOWCASE 25 "¦" TISAVECHAR 25 "¦" TIWRITER 21 CHANGES V2 DV80 FILES TO V1 COMPATIBLE "¦" TRAFFICCOP 16 game "¦" UPSIDEDOWN 21 see flip above "¦" VALCALLKEY 23 "¦" WONKAPILL 25 ti basic game "¦" XB/TRICK 25 note the name -list before running! "¦"

>TI*MES TEXT- three years of DV80 files from TI*MES in archived format (requires Bob Boones ARCHIVER to use-see Util 21). REQUIRES FIVE SSSD DISKS.

>FUNLWEB 4.3 - 40 COLUMN VERSION ! TWO DISKS BUT PARTLY ARCHIVED! to make it fit! Minor amendment to start up which may now be direct to UL or to DR, with space bar defeating this and going to main menu. The 80 column 4.3 has a major upgrade to DR not possible in 40 cols and comes as a single disk supplement to the 40 column version (eg 40 col=2 disks, 80 col=3 disks).

RLE28...Cairofont(lots of small pics to be clipped); Mrs Goggins, Hedgehog, Mermaid, Dogs, Pets, Old phone, Postman Pat.

TI RUNNER SCREENS A- in TI Artist format, the screens from the game, disk A contains screens 1 to 13. Shows exit ladders and different types of brickwork! May help you complete a screen you are stuck on.

TI RUNNER SCREENS B- screens 14 to 26 as above
TI RUNNER SCREENS C- screens 27 to 40 as above
TI RUNNER SCREENS D- screens 40 to 50 as above (some in colour)

MACFLIX DISKS- require you to own PIX PRO commercial program:

MACFLIX 21...Fievel01 and 02 (lovely mice); Make It Sew (Star Trek TNG); Shaw3 (you know who); TI
MACFLIX 22...Back to the Future 2; Country Code; and two Hedgehogs.

a new series also in Macflix format:
MACPAINT1...NAGEL1, NAGEL2, NAGEL3, NAGEL4 (girls)
MACPAINT2...Cavegirl, Communications Pin, Garfield, Little Men cartoon, Mickey Mouse, Picard (ST-TNG)
MACPAINT3...CIRCUS, HAN SOLO, ESCHER WATERFALL, WOOD DUCKS
MACPAINT4...SHAWS100,SHAWS400(slightly distorted),Unicorn, Yoda.
MACPAINT5...Five pics of George Shaw! plus a Manchester Theatre
MACPAINT6...BARTON ARCADE and TOWN HALL, MANCHESTER; GEORGE; STARTREK TRIO

==========
Archive section- the following disks are in archived format, and as appropriate replace double disk sets previously offered, or may be "new" adds-each now SSSD:
DEBUGGER- the 1984 Navarone debugger plus Source Code for Navarone Bugfixer. No docs but similar to TI Debug or SBUG.
DM1000 3.7+4.0-withdrawn versions for historic purposes- withdrawn as they are liable to corrupt files or disks from time to time.
FAST TERM- now offered with source code on one disk.
SYSTEM DISK LOADER-now on one disk.
TE3-now with source code on one disk.
==========

TI BASE section- databases for use with TI Base(required):
>TI*MES issues 1 to 26 (1000 entries) INDEX
>USER GROUP DATABASE (partly out of date-lots of duplications)
>NAMES INDEX [requires double sided disk-but only counts as one disk]-lots of addresses mainly American, associated with the TI, many well out of date!
>UK MAGAZINE INDEX-All issues of TI LINES, TIHCUG, TI USER, PARCO, TIdings, and EAR from May88 to Dec89
>MICROPENDIUM INDEX JAN89->
>UK INDEX 2 - TIMES from issue 27 and EAR from Jan90
==========
>UTIL20 has SAVEXT, a utility which recovers a crashed XB program! PROVIDED you keep the PEB switched on. Very neat.

>GAMES 22: starts off with another American Monopoly for 2-6 players, and an interesting Pinball construction set by John Behnke, POWERBALL, ALL in XB, and the most effective pinball in XB I've seen.

MA2 was accidentally wiped and it NOW contains a lovely picture of a bird, in 256 colours if you have a 9938VDP, and 424 lines long, so you need XHi (Geneve or TI+9938) or SmArtCopy (TI+9918).

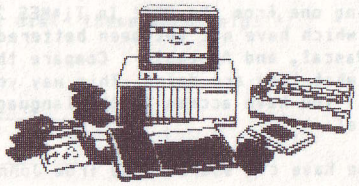
Best wishes!
Stephen.
****
COMPETITION from Doug Moller, no prizes just send in your code!
1. At various times the hands of a clock with trisect its face- there will be a 120 degree angle between the hands. Assume all hands start at 12 and all three hands move smoothly. When will the angles between the hands be exactly 120 degrees? OK harder one- this next clock has a ticking second hand, which moves only in one second jumps but the other hands move smoothly...

2. A journey of 300 miles is undertaken in a car rented at $3 per hour plus fuel at $2 per gallon. What is best (eg cheapest) speed? To be totally unreal, assume fuel consumption is 30mpg at 30mph, 20mpg at 60mph and 15mpg at 75mph. eg an increase in speed of 3mph means one gallon lasts 1 mile less.

This issue omits several regular contributions, as my annual holidays fell in the last two weeks before press date, and unfortunately, in the previous month to that my father was not too well in hospital, leaving me little time to put this together! Still we have the full 60 pages and perhaps back to normal content next time- not that anyone ever writes to me to say they want to see more XB tutorials or more Tigercub or more reviews or more... (or less!).

~stephen~

```
{
locate(1,1);
printf("7 to the power of %d=",power);
puts("          ");
locate(3,1);count=0;

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

      for(count=0;count<=800;count++)
         {puts(" ");}
      locate(3,1);count=0;
      }
   }
```



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

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

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

```
      DEF SEVAC

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



```
MULT  INC 2
      CLR 4
      LI 0,DIG    [DIGIT STORE]
C     INC 4
      MOV *0,5
      MPY 7,5
      A 3,6
      DIV 10,5
      MOV 5,3
      MOV 6,*0+
      C 4,1
      JNE C
      CI 3,0
      JEQ D
      MOV 3,*0
      CLR 3
      INC 1
D     B *11

TEST  CLR 4
      LI 0,DIG-2
E     CLR 8
F     AI 4,6
      C 4,1
      JGT RTN
      AI 0,12
      MOV *0,6
      C 6,7
      JNE F
G     DEC 4
      CI 4,0
      JEQ H
      DECT 0
      MOV *0,6
      C 6,7
      JEQ G
H     INC 4
      C 4,1
      JGT RTN
      INCT 0
      MOV *0,6
      CI 6,7
      JNE E
      INC 0
      CI 0,6
      JNE H
      B @PRINT
RTN   B *11

PRINT MOV 1,14
      MOV 2,6
      LI 0,>182
      LI 2,>E
      LI 3,TXT1
      BL @VMBW
      LI 0,>195
      LI 2,2
      LI 3,TXT2
      BL @VMBW
      LI 0,>193
I     CLR 5
      DIV 10,5
      MOV 6,1     N     B *11
      AI 1,>30
      BL @VSBW    TXT1  TEXT 'SEVEN TO POWER'
      DEC 0       TXT2  TEXT 'IS'
      MOV 5,6
      CI 6,0      DIG   BSS 400
      JNE I
                        END
      MOV 14,0
      AI 0,>1BF
      MOV 14,4
      INC 4
      LI 5,DIG
J     DEC 0
      DEC 4
      MOV *5+,1
      AI 1,>30
      BL @VSBW
      CI 4,1
      JNE J
      LI 0,>FFFF
      LI 1,>B
K     DEC 1
L     DEC 0
      CI 0,0
      JNE L
      CI 1,0
      JNE K
      CLR 0
      MOVB 0,@>837C
      LWPI >83E0
      B @>70

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

VMBW  SWPB 0
      MOVB 0,@>8C02
      SWPB 0
      MOVB 0,@>8C02
      CLR 4
M     MOV *3+,1
      MOVB 1,@>8C04
      INC 4
      C 15,2
      JEQ N
      SWPB 1
      MOVB 1,@>8C04
      INC 4
      C 4,2
      JNE M
      B *11
```

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

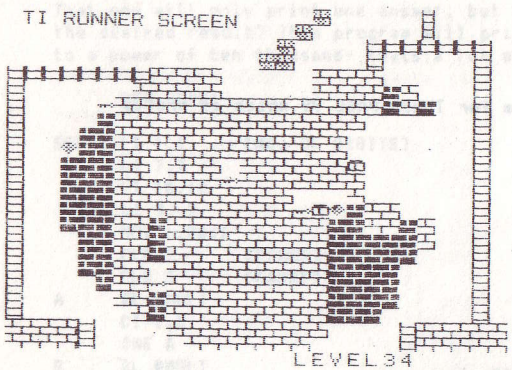
```
      DEF SEVHI

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

MULT  INC 2
      CLR 4
      LI 0,DIG    [DIGIT STORE]
C     INC 4
      MOV *0,5
      MPY 7,5
      A 3,6
      DIV 10,5
      MOV 5,3
      MOV 6,*0+
      C 4,1
      JNE C
      CI 3,0
      JEQ D
      MOV 3,*0
      CLR 3
      INC 1
D     B *11

TEST  CLR 4
      LI 0,DIG-2
E     CLR 8
F     AI 4,6
      C 4,1
      JGT RTN
      AI 0,12
      MOV *0,6
      C 6,7
      JNE F
G     DEC 4
      CI 4,0
      JEQ H
      DECT 0
      MOV *0,6
      C 6,7
      JEQ G
H     INC 4
      C 4,1
      JGT RTN
      INCT 0
      MOV *0,6
      CI 6,7
      JNE E
      INC 8
      CI 8,6
      JNE H
      MOV 11,8
      BL @PRINT
      MOV 8,11
      JMP E
RTN   B *11

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

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

VMBW  SWPB 0
      MOVB 0,@>8C02
      SWPB 0
      MOVB 0,@>8C02
      CLR 4
M     MOV *3+,1
      MOVB 1,@>8C04
      SWPB 1
      MOVB 1,@>8C04
      INC 4
      C 4,2
      JNE M
N     B *11

TXT   TEXT 'END OF RUN'

DIG   BSS 400

      END
```


==========

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

```
TXT2  TEXT 'IS'
PROGRAM SEVPAS;

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

PROCEDURE PRINT;
VAR Z:INTEGER;
BEGIN
  WRITELN("SEVEN TO THE POWER OF",POW," IS :");
  WRITELN;
  FOR Z:=MAXDIG DOWNTO 1 DO
  WRITE(SEG(CIS(DIG[Z]),2,1));
  WRITELN;
  WRITELN
END;

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

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

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

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

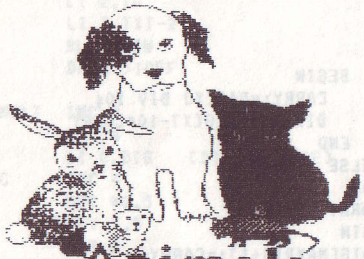
TI RUNNER SCREEN



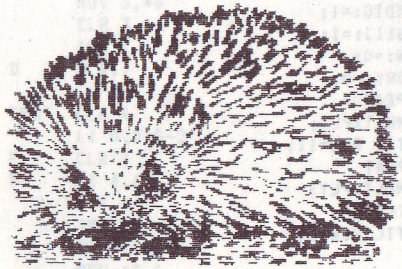
LEVEL 34

STAR TREK-
THE NEXT GENERATION
on BBC tv soon!

HUMANE RESEARCH TRUST
29 Bramhall Lane South,
STOCKPORT, Cheshire, SK7 2DN



Animals
are my
Friends



BRITISH HEDGEHOG
PRESERVATION SOCIETY
Knowbury House, Knowbury,
LUDLOW, Shropshire