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

Volume 2, Issue 11 April 1st., 1986

TI-LINES is the monthly newsletter of OXON TI USERS, and the form of presentation of any material remains the copyright of each individual author. OXON TI USERS reserves the right to submit any material to other newsletters for publication, with due credit to the author(s) concerned. Submissions are accepted on this understanding.

It is the responsibility of each author to ensure that no copyright infringement will occur by the publication of their articles in TI-LINES, and OXON TI USERS cannot be held responsible for any such infringements. Every effort is made to ensure that any information given in TI-LINES is correct.

TI-LINES is produced and published by:

Peter G. Q. Brooks

96 Banbury Road
OXFORD
OX2 6JT

Tel: Oxford 50822

Tel. (02357) 2360
WESSEX PRESS
9-11 Mill Street, Wantage, Oxon OX12 9AB.

Oxon residents are offered a subsidised subscription at £1.56 p.a., payable either as 12 second class postage stamps or by cheque.

THIS OFFER APPLIES UNTIL MAY 1986 ONLY

TI-LINES is available on Associate subscription to Users resident outside Oxfordshire, for £10 p.a. Back issues are £2 including post and packing. New subscriptions begin with Issue 1 of the current volume, up to and including the current issue, regardless of the number of issues elapsed.

Contributions should be submitted either on diskette in TI-Writer compatible files (DIS/VAR 80 or DIS/FIX 80), or in a form which is as legible as possible. Artwork should fit within an A4 area and should not contain colour. Very high contrast line drawings are preferred and these may be produced by arrangement with the publisher.

REMEMBER: RENEW YOUR SUBSCRIPTION SOON

TI-LINES V2.11 APRIL 1986

Index to articles Page

EDITORIAL

HISTORY REPEATS ITSELF

Once again, apologies are due (getting to be a bad habit, Brooks!). After I had put the March issue to bed, GRAHAM WOLSTENHOLME's amended diagram became available, so once again I stuck it in the centre of the newsletter without any explanation, mainly in the hope that I would get the updated diagram to those readers needing it as soon as possible. Subsequent copies of the February issue now carry a warning sticker on the original diagram notifying any potential constructor that the March issue contains the corrected diagram.

INBUILT ADVERTISING

I am highly impressed with BRUCE CARON's DM1000 V2.2 Disk Management Program to the extent that I have made use of the facility for sending control codes to the printer when printing catalogues, to inform any reader that the directory was provided by DM1000. Now my printed catalogues look like this:

```
CATALOGUED USING DM1000 V2.2

DM1000      Free   411 Used   309
Filename    Size Type/No.   7 P
----------  ---- ---------- - -
DMDOCPT1      52 DIS/VAR   80 P
DMDOCPT2      51 DIS/VAR   80 P
DMDOCPT3      93 DIS/VAR   80 P
DMDOCPT4      45 DIS/VAR   80 P
LOAD           9 PROGRAM      P
MGR1          33 PROGRAM      P
MGR2          24 PROGRAM      P
```

You could even send seasonal greetings (like Merry Christmas, Happy New Year and Drop Dead You Paunchy Bleeder What Do I Want A Valentine's Card From You For ? (ladies only)).

Magazine scanned 2022 by Stephen Shaw

THE FOURTH NATIONAL TI USERS MEETING

Uncle Clive and Auntie Audrey, with the extremely able assistance of PHIL MARSDEN in finding a scarce venue in LEEDS, are hosting the next National TI Users Meeting at the Old Dispensary, Institute For The Deaf, North Street, Leeds, on MAY 3RD. I understand that there will be a blanket entry fee of 50p per person, and I look forward to seeing as many OTIUsers as can make it. GORDON PITT and a crew from the WEST MIDLANDS TI USERS will be there, as will MARK LEE of MALTBY TI USERS. We hope to be able to demonstrate the light pen complete with machine code routines, and with any luck our own DAVE HEWITT will be there demonstrating his PIO interface.

COMMUNICATION IS THE THING

Don't forget, if you are thinking of getting a modem, give GORDON PITT a call on 0922 476373. He has recently finished evaluating the GEC DATACHAT modem, which has 1200/1200, 1200/75, and 75/1200 rates, software selectable. It makes use of the power supplied by the phone itself so doesn't need a separate mains socket. It comes in an ivory case to match the standard BT phone colour, and Gordon tells me that he found the instruction booklet clear and understandable.

You may remember me mentioning in an earlier issue that if Gordon can get enough people interested, GEC would be willing to come down in price. I am told that they will reduce the standard £89.95 by as much as £20 or more - but it depends upon the numbers who are interested. Don't forget that while TEII can only handle 300 or 110 baud, we have Public Domain software on disk which can provide access to the other speeds necessary to firkle around in Prestel for example (Viditel, and Fasterm).

I understand that the price will include free membership of MICROLINK, worth about £5.

I am planning to put together a 24hour Bulletin Board sometime this year if all goes well, so if you fancy getting into that, now's the time to get yourself organised.

ME - IN THE PRIVACY OF YOUR OWN HOME!

I am laying plans (no comments, please) for the production of a series of video tapes, and possibly also tape/slide presentations, to go alongside the Grand Booklets Scheme. Unlike the booklets, the tapes should be much easier to produce, but I would like to get some idea of what YOU, the potential consumer, think of the idea.

The price will probably be no more than £15 for the video tape, and the first will probably cover the learning of TI BASIC and an introduction to the skill of programming, as well as a brief rundown on the machine itself, some examples of jargon that might be useful to learn, and so on.

The eventual list will cover all the subjects destined to be covered by the booklets, including (I hope) a "servicing" video to help the electronically-minded to clean up, and repair, some of the simpler faults.

I am thinking of offering an aerial splitter for those who want one, so that they can run the video and the computer on the same TV, just by switching from the video channel to the computer channel, and thus try out everything that they see on the tape, as they see it.

Now this project will obviously require a great deal of work, and I am not keen to waste a large amount of time if there is no demand for such a service. I understand that a very large proportion of OTIUsers possess video players, so such a service would not be for the minority, as it might have been a few years ago.

I do need some feedback, however, in order to make sure that I am not chasing shadows. If you feel that you WOULD like, and could benefit from, the OTIU Video Show, then please let me know, either by phone or post, as soon as you can. Oh, just in case you are concerned about receiving video nasties through the post, although I will be on camera it won't be for very long, and you can always cover your eyes at the crucial moments.

The format, before I forget, will be VHS, and I intend to make use of a professional recording studio for the final edit, so the finished product should look as good as stuff produced by the Big Boys.

The Booklets are still on the cards, as is the tape/slide series which will cover the same material (but which is easier to produce), so there is still lots more coming up to help you get the best from your TI!

~~~~~~

WHICH SIDE ARE YOU ON ?

Once again Yours Truly has fouled up: I publicised the Small C and PRBase Public Domain programs as being single disks at £2.50 plus the usual post and packing of 65p.

Ahem. I forgot two very important things. Firstly, I didn't check the number of sectors used by either program. They both use far more than the 358 sectors available on a single-sided disk. Secondly, when I sent out copies to those who purchased them, I neglected to ask if they had the necessary double-sided drives in order to use either program.

This necessitates a slight change in pricing policy. If you have DOUBLE sided drives, the price is the same, and you receive the programs on a single disk (double-sided). If you have a SINGLE-sided drive, then the programs come on TWO disks, at a cost of £5.00 plus 65p post and packing.

I'm sorry about this - I should have been more awake.

Incidentally, if you have been thinking of upgrading your drives from single to double-sided, but without having to buy a replacement drive (and then try and sell the single-sided drive, a mammoth task in itself as I am finding out!), then I may be able to offer you a viable alternative.

For the price of £65, I will undertake to have your EXISTING drive upgraded to double-sided. My contact has about 200 second heads, as I understand it, and as the number of double-sided drives in stock is dwindling rapidly, it may soon be the only way for many of you to upgrade to double-sided drives.

If you have received Small C, or PRBase, from me and have only single-sided drives, then contact me immediately and I will provide the inaccessible files on a second disk for no further charge.

~~~~~~

NUTTY NOAH

No, this isn't a piece about the daft TV program for kids. It's a small insight into the "new" TI-compatible machine which has been in the process of being built for, oh, I forget exactly how long, but like the Booklets Project, it is much discussed and never actually seen.

The new machine, code-named NOAH, (a little pun based on the company's name: MY-ARC, geddit ?), has made a few rare appearances under rather peculiar circumstances. Recently, when its projected launch date was put back from late 1985 to early 1986 to mid-1986, it made a reluctant appearance at a meeting in Chicago (in November last year, to be exact), largely I am led to believe, as a result of intense pressure from the many 99 owners in the audience. Only a shell was available, as the company explained that they had blown a chip the previous day while working on it, and had been unable to replace it in time to present a working demo.

At the same venue, a purported demonstration of a new, hyper-fast Extended BASIC came unstuck when the Operating System for the hard disk containing the aforementioned BASIC was uncannily "forgotten".

This is like inviting motoring journalists to witness a test drive of the latest wonder vehicle from Leyland, only to forget the ignition key to the only production model available, and then be unable to show more than a chassis with no engine and no interior fittings.

I can't claim to have a fantastic amount of experience in the micro world, but in the nine years in which I have avidly absorbed nearly every event that ever took place (I'm talking now of non-TI events), it has always been the case that any company worth its salt has more than just one prototype machine, precisely to cater for the odd blowing of chips etc., in the process of development.

The fact that Myarc seem incapable of being entirely honest with their own domestic consumers bodes ill for we Europeans. The much-vaunted Noah micro with its large capacity, high speed, mouse, etc., etc., coupled with its supposed compatibility with the existing TI equipment, is but a shadow of its former self.

Compatibility with the existing PEB has been questioned, and certainly there is no intention of doing anything to make it compatible with the existing p-code system (the leading light of Myarc, Lou Phillips, is quoted as saying that p-code is dead, citing the fact that even its inventor has dropped it. I assume that this is a reference to the fact that the Regents at UCSD told the copyright holders within the University that their operation was against the University's Principles, for which reason UCSD PASCAL is no longer UCSD; I forget exactly who owns the rights at present - it may even be Readers Digest - because it has changed hands frequently recently).

In addition, the 80 column card has been dropped, and the IBM lookalike keyboard has spread its evil tentacles into the machine itself, to the extent that the most recent rumour is of a direct IBM clone, dropping most of the attempts to retain compatibility with existing equipment.

If Myarc intend dipping a toe into the IBM-ulation water they had better wear asbestos flippers, because that particular rivulet is set to become a raging, boiling torrent shortly, when the likes of Amstrad and Atari enter the fray with their cheapie IBM clones. Who's going to cough up maybe £400 for Myarc's baby when they may get a superior system for £250 elsewhere (and be able to pay for it by flogging their existing TI system) ?

On the other hand... The rumours could be totally wrong, Myarc could come up with exactly what they have been promising all these months, and I would dearly like to believe that they will.

The problem is, that even if they do, you can bet your cotton socks that the European market will get the least attention (just as it did when TI were in the business).

I am happy to say that some TI owners in the UK (and elsewhere) are not prepared to take anything lying down, and are girding up their loins even as I type. There are projects under way which will hopefully allow us all to cheaply enhance our systems, and get the kind of useful system which the 99s potentially had within them but which TI carefully denied us. It will take time, of course, because we are all part-timers, and not able to work with the kind of research and development funds and facilities possessed by the commercial bodies. We all know (and probably sneer at) the story of the Tortoise and the Hare; maybe this is one instance when its peculiar course and conclusion will be mirrored in Real Life...

~~~~~~

SERENDIPITY INDEED

Recently, GORDON PITT and I ventured into Solihull to deliver and install a twin half-height double-sided 40 track MPI (puff, puff) externally-powered and cased drive setup for a future (we hope) OTIUser.

In the process we discovered that his apparent full height, single-sided Toshiba drive had been double-sided all along, with the unit having been configured as TWO single-sided drives (one side is regarded as DSK1, and the other is DSK2, as is the case with Beeb drive systems, I believe).

I remember, long ago, when I published the Modulator Modification article by the first DAVE HEWITT, that Dave told me of his similar discovery, but he could not remember the exact details as it had happened some considerable time before.

Gordon and I must have appeared to be wearing stetsons and spurs (i.e., we looked like cowboys!) to the prospective benefactor of our ministrations, for it took us a fair amount of time and discussion before we managed to persuade him that snipping out a diode and removing a jumper lead would give him not just two but THREE double-sided drives, thus at a stroke increasing his disk capacity to six times its perceived original.

This may have had something to do with the fact that his disks were extremely valuable to him, each containing up to £35,000 worth of data, so it was with utter conviction and shaking hands that I performed the surgery.

Or at least part of the surgery. I won't go into the full story, heart-stopping though it is, but suffice it to say that as usual, the best laid plans of mice, men, and Pete Brooks oft come to naught. The surgery was carried out in two stages, and each time that the system did not function as it should have done according to Brooks' Theory, we could see that the victim ..er.. client suffered sudden increases in personal tension as a result.

I am happy to report that when I eventually carried out all the snips that I thought I needed to (and even Gordon was beginning to doubt that I knew what I was doing, I think!) the system powered up properly, with no awkward little red lights blinking on and off when they shouldn't have done.
~~~~~~

There is a distinct possibility that there may be other disk drive owners put there who are in a similar position (i.e., having twice the capacity without realising it!).

Akhter Instruments of Harlow (Harlowe ?) were the suppliers of the drive, so if you bought your drives from them, have a look inside. There are three things to look for. One is the obvious: a second drive head. You should be able to spot two sets of wires leading to two connectors on the drive's circuit board. The second is a jumper wire from the drive's circuit board edge connector leading to a series of vertical pins, where the third component, a diode, should be visible, connecting two of the pins.

Further, if you try addressing DSK2, the red light on your DSK1 drive should come on. It is actually addressing the other side of the disk in the drive, so you shouldn't have any problems with overwriting existing data on DSK1.

If you have any doubts, one way or the other, you can always contact me for some advice/offer to come and burn your system out for you.

And by the way, this is NOT an April Fool item...

~~~~~~

QUALITY UNCONTROLLED

My apologies for the quality of the photocopying in the last issue: the copier was playing up and there was nothing really that I could do about it, other than delay publication further, which I was loath to do.

As a result, I am repeating the printout of the Lady With The Amp Outside Her Ohm, courtesy, you may recall, of ED YORK of CIN-DAY USERS, hopefully this time with better results.

~~~~~~

ARE THE CAVALRY IN SIGHT ?

A recent copy of the February 1986 issue of NORTHWEST OHIO 99ER NEWS, sent by BILL SAGER with whom I have an exchange subscription, had a couple of items of interest to me (and therefore, hopefully, to you), one of which is perhaps of paramount importance.

Included in the issue's pages was a copy of a REPAIR CENTRE INVOICE from TI in Lubbock, which listed all the GROMS and other bits and bobs which go to make up TI modules. This suggests that if one of your more important modules goes AWOL (as did an Editor/Assembler for a member recently) then we may be able to save the day by ordering replacement chips from TI direct, rather than try to prise anything out of ECD over here (you want your Extended BASIC module repairing ? Certainly sir, for a little more than the price of a secondhand unit...)

The sheet of GROMS and prices is possibly courtesy of KENT SHEETS, whose name and address appeared at the bottom of the page, along with the firm assertion that the TI-99/4A is alive and well.

The only thing that made me wince a little was the fact that someone at TI didn't know how to spell "beige"...

~~~~~~
~~~~~~

RECOURSE TO REC CLAUSE PART II

You may recall that I mentioned last issue the possibility of sweeping a disk without using REC.

The simplest technique is to OPEN the file to the directory, read the first TWO entries and delete the second as long as it is not a null string, and then CLOSE the file.

Why two ? The first is the disk name and cannot be deleted. The second is the current first entry in the directory, as long as it is not a null string (which indicates the end of the entries).

However, it is so slow that listening to grass grow is an absorbing hobby in comparison.

Still, if you are desperate to learn everything you can about your machine, who am I to dissuade you ? Here's an example listing:

```
    100 OPEN #1:"DSK1.",INPUT ,INTERNAL,RELATIVE
    110 INPUT #1:A$
    120 INPUT #1:A$
    130 IF A$="" THEN 170
    140 DELETE "DSK1."&A$
    150 CLOSE #1
    160 GOTO 100
    170 END
```

~~~~~~
~~~~~~

FLOWER POWER et al IN TI FORTH

By GLENN DAVIS from the September 1985 issue of the MSP 99 Newsletter

This month I am presenting a tiny demonstration program to you. I had to find one that was small enough to publish, but which still did something interesting. The one I chose is a translation from an Apple "HCM one-liner" to TI Forth. The routine is one screen long, but needs two screens of support code to allow Forth to use the sine and cosine functions without resorting to floating point. (I did try using floating point, but it ran 29 times slower; and that's using my CODE definitions of the -FLOAT words!)

RICHARD LAUHEAD (another MSP member) added a few features to this routine too. Among them, that you must press CLEAR to get back to text mode after the final "flower" is drawn. By the way, you must have the -GRAPH2 routines loaded (or -64SUPPORT) to run the one-liner.

The Apple one-liner had a few "hidden" bugs in it, which we might as well discuss here. That author looped from 0 to 359 to generate the sine and cosine values. I do the same thing here. Do you see what the difference is ? In BASIC (and floating point on micros in general), the arguments are supposed to be in RADIANS. (A Radian is about 58 degrees of a circle).

This made the dots skip around so the "flowers" materialized instead of being drawn. I tried doing the same thing, but found Forth was so fast that the drawings weren't visible long enough to enjoy them!

Speed always seems to be inmportant to computer users. BASIC is always maligned for being less-than-fast. Forth is a very fast interpretive language but sometimes it is necessary to code things in machine language.

{In an earlier MSP 99 Newsletter, Glenn presented an alternative floating point word set, written in machine language, which I hope to be able to publish in either the May or June issue of TI-LINES. PB}

This month I present the "double number word set" in machine language.

Double numbers are 32-bit (4-byte) numbers. These words are intended to manipulate two single length numbers or one double-length number. Often when these words are used in a word at all, they are used frequently. Definitions are given in the TI Forth manual, but they are in Forth, not in machine language. Of course, those are slower. See a book on Forth or the TI Forth manual for additional descriptions of these words.

To enter these screens, type them in beginning with the initial comment line. DO NOT TYPE THE "SCR#n:" PART. THAT IS ONLY FOR YOUR REFERENCE. To use them LOAD the first screen in the series. For example, to use the flower power routine, you would load the sine and cosine functions and then the flower-power routine by typing -GRAPH2 90 LOAD 92 LOAD ONE-LINER.

```
SCR#90:

( transcendental functions                                      30JUN84 GED )
: TABLE ( data type of table                                       LJS 1982 )
  <BUILDS DOES> SWAP DUP + + @ ;
BASE->R DECIMAL
TABLE (SINES) ( Angle in degrees >=0 AND <=90 --- SINE * 10000)
 0 , 175 , 349 , 523 , 698 , 872 , 1045 , 1219 , 1392 , 1564 ,
1736 , 1908 , 2079 , 2250 , 2419 , 2588 , 2756 , 2924 , 3090 ,
3256 , 3420 , 3584 , 3746 , 3907 , 4067 , 4226 , 4384 , 4540 ,
4695 , 4848 , 5000 , 5150 , 5299 , 5446 , 5592 , 5736 , 5878 ,
6018 , 6157 , 6293 , 6428 , 6561 , 6691 , 6820 , 6947 , 7071 ,
7193 , 7313 , 7431 , 7547 , 7660 , 7771 , 7880 , 7986 , 8090 ,
8191 , 8290 , 8387 , 8480 , 8572 , 8660 , 8747 , 8829 , 8910 ,
8988 , 9063 , 9135 , 9205 , 9272 , 9336 , 9397 , 9455 , 9511 ,
9563 , 9613 , 9659 , 9703 , 9744 , 9781 , 9816 , 9848 , 9877 ,
9903 , 9926 , 9945 , 9962 , 9976 , 9986 , 9994 , 9998 , 10000 ,
R->BASE -->
```

```
SCR#91:

( trig functions                                                30JUN84 GED )
BASE->R DECIMAL
: SINES ( [N<361] --- sine * 10000 )
     DUP 270 > IF 360 SWAP - (SINES) MINUS ELSE DUP 180 >
     IF 180 - (SINES) MINUS ELSE DUP 90 > IF 180 SWAP -
     ENDIF (SINES) ENDIF ENDIF ;
: <SINE          ( n --- [n<360] )
    DUP 0< IF ( - ) 360 MOD 360 + ELSE ( + ) 360 MOD ENDIF ;
: SINE           ( n --- sine * 10000 ) <SINE SINES ;
: COSINE         ( n --- cosine * 10000 ) 90 + SINE ;
: TANGENT        ( n --- tang ) DUP SINE SWAP COSINE / ;
: COTAGENT       ( n --- cotan ) DUP COSINE SWAP SINE / ;
: SECANT         ( n --- secant) SINE 10000 SWAP / ;
: COSECANT       ( n --- cosec ) COSINE 10000 SWAP / ;
R->BASE
```

```
SCR#92:

( a translation from HCM "ONE-LINERS" vol 5 no 3               06JUL85 GED )
BASE->R HEX
: CLEAN    ( --- ) 2000 1800 0 VFILL ; DECIMAL
: SCALE    ( --- ) 10000 */ 110 / ;
: ?RETURN ( --- ) BEGIN ?TERMINAL UNTIL TEXT ;
: ONE-LINER ( --- )
   GRAPHICS2
11 1 DO CLEAN
        I 2 MOD MINUS 180 * 360 +
      0 DO    I J * COSINE DUP
              I COSINE SCALE 128 +
        SWAP I SINE SCALE MINUS 95 SWAP -
        DOT LOOP
     LOOP ?RETURN ;
R->BASE
```

```
SCR#93:

( Double number word set CODE definitions                    05JUL85 GED )
BASE->R DECIMAL 74 R->BASE CLOAD ;CODE
BASE->R HEX

CODE 2!     C039 , CC39 , CC39 , 045F ,
CODE 2@     C019 , C070 , C650 , 0649 , C641 , 045F ,
CODE 2DROP  8E79 , 045F ,
CODE 2DUP   0229 , -4 , CA69 , 6 , 2 , C669 , 4 , 045F ,
CODE 2OVER  0229 , -4 , CA69 , A , 2 , C669 , 8 , 045F ,
CODE 2SWAP  C029 , 2 , C059 , C669 , 4 , CA69 , 6 , 2 ,
            CA40 , 6 , CA41 , 4 , 045F ,
CODE 2ROT   C029 , A , C069 , 8 , C089 , 0222 , 6 , C892 , 4 ,
            0642 , 8242 , 16FB , CA40 , 2 , C641 , 045F ,

R->BASE -->
```

```
SCR#94:

( Double number word set cont. )
BASE->R DECIMAL
: 2CONSTANT <BUILDS , , DOES> 2@ ;
: 2VARIABLE <BUILDS 0. , , DOES> ;
: D-     DMINUS D+ ;
: D=     D- 0= SWAP 0= AND ;
: D0=    0. D= ;
: D<     D- SWAP DROP 0< ;
: DU<    ROT SWAP 2DUP U< IF 2DROP 2DROP
          ELSE = IF U< ELSE 2DROP 0 ENDIF ENDIF ;
: DMAX   2OVER 2OVER D- SWAP DROP 0<
          IF 2SWAP ENDIF 2DROP ;
: DMIN   2OVER 2OVER 2SWAP D- SWAP DROP 0<
         IF 2SWAP ENDIF 2DROP ;

R->BASE
```

BULLETIN BOARD

A few goodies on offer this month. I have half a dozen TI joysticks, brand new, going for £12 each inclusive of post and packing. First come, first served. If you can contact me by phone, that will get your order in as early as possible.

A few books for your delectation:

33 Programs For The TI-99/4A	£5	Dynamic Games For Your TI-99/4A	£3
Learning To Use The TI-99/4A	£3	Get More From The TI-99/4A	£3
4A User's Handbook	£5	Getting Started With The TI-99/4A	£4

I have a single sided, 40 track, full height, MPI drive at £35 looking for a PEB in which to sit, or I can provide a power supply and case (both new) for an additional £45.

I have a MUSIC MAKER module with manual at £10, and a PARSEC module for £4.

I have a TI-Writer module, with manual and folder, which I will upgrade with the Public Domain enhancements, going for £30.

I have a few TEII Protocol Manuals going for £2.50 including post and packing. Each comes in a slide binder, although the print quality is not that fantastic, I have to admit.

Ring Oxford (0865) 50822, best after 7 pm, and take into account my penchant for disappearing into the kitchen/bathroom/nearest major city to deliver bits of hardware!

WEST MIDLANDS TI USER GROUP has a TI Disk Controller card for sale. The asking price is £80. Contact GORDON PITT on 0922 476373.

CHRIS BAKER is selling an EDITOR/ASSEMBLER package in excellent condition. He is asking £40 inclusive of post and packing, and you can contact him on 0884 258272.

Another TI owner (not an OTIUser) is selling a complete system. Mr GILBERT has a console, a PEB, a disk controller card, a disk drive, an RS232 card, TWO 32K RAM cards, a speech synthesizer, Extended BASIC, TI-Writer package, a TI tape recorder, an MBX Expansion system (speech recognition/synthesis) with a Baseball game, and a number of cartridges - Soccer Supremo, Car Wars, Amazeing, Munchman, Video Chess, Connect 4, Indoor Soccer, and Parsec, - as well as a few cassette-based programs.

The asking price is £400 or nearest offer, and you can contact Mr Gilbert on 0733 222453.

(continued overleaf)

WANTED WANTED WANTED WANTED WANTED WANTED WANTED WANTED WANTED WANTED WANTED

DAVE CARR is looking for a PASCAL card. Contact him on 0229 89 346.

CHRIS BAKER (see above for phone number) is also looking for a PASCAL card.

SERVICES

WEST MIDLANDS TI USERS, alias GORDON PITT, can provide the following services to members of OXON TI USERS:

10 DISKXPRESS 48 TPI DOUBLE-SIDED DOUBLE DENSITY DISKS (includes library case)	£13.50 + p+p £1.50
10 MEMOREX 96 TPI DOUBLE-SIDED DOUBLE DENSITY DISKS (includes library case)	£18.00 + p+p £1.50

Allow up to 28 days (worst case) for delivery.

Lockable 80 disk size REXEL boxes	£18.00 + p+p £3.00
Tractor-feed paper, 2000 sheets fanfold, 80 gm (preferable to pick it up, either from Gordon or from me if that is more convenient. Ring well in advance)	£15.00
Power supply and case to house one full height or two half height drives, made to order (preferable to pick up, as above)	£45.00

Contact GORDON PITT on 0922 476373
        PETE BROOKS on 0865 50822

Incidentally, Gordon is responsible for instigating and co-ordinating the LIGHT PEN and the RGB/COMPOSITE VIDEO/MONITOR INTERFACE projects.

PRINT AT IN TI BASIC

By Peter Brooks Originally written for HOME COMPUTING WEEKLY

Lamentably, there is no PRINT AT facility on the standard Texas set-up; for those new to BASIC, it permits you to PRINT information (text, numbers, etc.) to any part of the screen without causing a 'scroll' upwards, and without disturbing any existing screen data elsewhere.

For example, PRINT AT(10,16): "SCORE:" will print the text "SCORE:", beginning at row 10, column 16. This command is virtually a necessity in any BASIC, so it is surprising that the designers chose to omit it. There is however a way of implementing it by using the graphics command HCHAR(), (or VCHAR()),but don't expect the data to be printed up quick as a flash.

The subprogram HCHAR() can place a 'character' (letter, digit, punctuation mark, graphic, etc.), and, optionally, repeat it, anywhere on the screen. In fact, HCHAR() and its two companions VCHAR() and GCHAR() can address a much greater screen area than can PRINT. HCHAR() can use the full 24 rows by 32 columns, while PRINT can only use columns 3 to 30 inclusive, and is restricted to operating on the 24th row. To imitate the PRINT AT example given above, you would need to PRINT TAB (16);"SCORE:"::::::::::, which will scroll everything and lose the first 14 lines.

What you can do is to create a subroutine which uses HCHAR() (and a few other functions and commands) to give a slow imitation of PRINT AT. Because this subroutine is (probably) going to be used quite often by your main program, it should be placed close to the beginning of the listing, where, for reasons too complex to go into here, it may be 'found' faster by the computer.

As the subroutine is going to be called upon to print different things at different rows and columns, we will need three 'variables' to hold the necessary data. In the example below I have used R for screen rows, C for columns, and D$ for the data to be printed. Because HCHAR() can only refer to caracters by their 'ASCII code', and because we will be using SEG$() to isolate and print the data a character at a time, a string variable must be used to hold the information to be printed out. A number must therefore be converted to a string using STR$() in order for it to be printed anywhere on the screen.

There is one further consideration: we may want to clear the line on which data is to appear, but not necessarily every time. We will therefore have two parts (and therefore two 'entries' to the subroutine): the first will clear the line (depending upon your requirements you may need to clear the entire 32 columns, or just from the column represented by C to the 32nd column, or even from column C for the length of the item to be printed; here I have chosen to clear from column C to column 32), and the second will perform the PRINT AT function. Here is the subroutine:

```
First line:   CALL HCHAR(R,C,32,33-C)
Second line:  FOR L=1 TO LEN(D$)
Third line:   CALL HCHAR(R,C+L-1,ASC(SEG$(D$,L 1)))
Fourth line:  NEXT L
Fifth line:   RETURN
```

Note that because C, L, R, and D$ are all used by the subroutine, you should not use them for other purposes in your main routine. After execution, R, C, and D$ will still contain the values they had before execution, and L will be equal to the length of D$ + 1. To use the subroutine, assign values to R, C, and D$, decide whether the line is to be cleared first, and then GOSUB to either First or Second line. Here is a short example routine:

```
100  CALL CLEAR
110  CALL SCREEN(8)
120  GOTO 180
130  CALL HCHAR(R,C,32,33-C)
140  FOR L=1 TO LEN(D$)
150  CALL HCHAR(R,C+L-1,ASC(SEG$(D$,L,1)))
160  NEXT L
170  RETURN
180  R=10
190  C=8
200  D$="MIMICKING PRINT AT"
210  GOSUB 140
220  CALL HCHAR(1,1,30,96)
230  R=2
240  C=14
250  D$=STR$("1984")
260  GOSUB 130
270  GOTO 270
```

Having entered and run this rather daft example, you will need to BREAK in order to stop the endless loop in line 270. Notice that you can print in any sequence; unlike ordinary PRINT, you can place data on screen first at the bottom, then at the top, then, say, in the middle. Because there is no scroll each time that you use the subroutine, nothing is disturbed.

Make sure that the data to be printed will not run off the edge of the screen, thereby causing a BAD ARGUMENT error, unless you modify the subroutine to cope with printing more than one line at a time. You should be able to see all the possibilities with the routine above: printing text, clearing before printing, and printing numbers. Remember that STR$() removes leading and trailing spaces as it turns a number into a string of digits.

Try changing the subroutine to print vertically, diagonally, backwards, with automatic centring, and so on, to gain familiarity with manipulating the screen.

LETTERS

PETER WALKER wrote to me recently concerning PRESTEL access:

ACCESSING PRESTEL FROM YOUR TI-99/4A

You need program B3 from the TI-EXCHANGE USER GROUP LIBRARY (or VIDITEL from the OTIU PUBLIC DOMAIN COLLECTION).

You will require 32K plus a disk system. The program originates from Holland and has been altered to give English instructions. It uses bit map mode to support both 40 column text and colour graphics. It uses proper split speed 1200/75 baud transmission, so you will need a modem which supports this speed (see Editorial on GORDON PITT and the GEC MODEM).

It does not support all UK PRESTEL features: noted so far as unsupported include flashing characters and conceal/reveal. The program does not have any capability of saving screen text to disk or printer.

HINTS WHEN USING PRESTEL

Keep your Identity Number (10 digits) and Password (4 characters) secret. Only quote your account number which is needed for addressing mailbox messages. ID and Password are not echoed to the screen for security. Ensure Alpha Lock is in the appropriate position if using an alpha Password.

Note that there are several PRESTEL computers and your Password can be different on each one, so if altering your Password you will need to do so on each machine.

(To change your Password go to page *920#). Account numbers are sometimes similar to your telephone number.

Never use the ENTER key while in PRESTEL. # is used to terminate input fields. If you suffer from line noise, you can key *00# to re-transmit a current screen.

TO SEND A MAILBOX MESSAGE

Select page *77#. Fill in MBX address field with the account number of the addressee, followed by #. A directory of registered MBX Users is on page *486#. Don't forget to register yourself (page *48644#). After your MBX number, you can type within the frame provided, using both upper and lower case and other ASCII characters. The cursor position is invisible which makes editing very difficult {I know, I know! PB}. However, the arrow keys (FCTN ESDX) do work but you have to wait for a few seconds while the screen is retransmitted up to the new cursor position. New text overstrikes the old.

To erase a message and revert to the beginning of the text field, type CTRL X. To revert to MBX address field, type CTRL 8. To complete the message, type # or ESCAPE J {For ESCAPE key, see later. PB}.

ESCAPE K aborts MBX and returns to main index. FCTN V (ASCII DEL) displays double vertical line characters.

SPECIAL MAILBOX EDITING FEATURES

You can use special ESCAPE sequences to change the colour of the text and background, select double height characters and use block graphics, but note that these attributes use screen space to set up, and are reset after each line. For example, to change the text from white to red, type ESCAPE A.

The ESCAPE character is sent on the TI-99 by pressing CTRL . (full stop).

Full details of these codes plus graphics characters can be found on page *777768# (*190600160# for graphics), or for MicroNet Users, pages *3237# and *32371# or I can give you more information.

If you are a member of MicroNet 800 (a closed User group for computer buffs) you can see examples of User-created graphics in "The Gallery", starting on page *323#. Pages beginning *2582201# are also worth a look.

PRESTEL ACCESS NUMBERS

There is usually a PRESTEL access number within local call range of you. (BT give you this number when you join). Page *5831# gives an index to other codes round the country, but note that these are often unavailable to callers from elsewhere. There is a London-based 300 baud service on 01 680 8245 which can be dialled nationwide using TEII or TE2 or similar emulator. This speed does not support graphics or colour but sends *s in their place.

LEAVING PRESTEL

Key *90# to leave PRESTEL, then switch off your modem. When quitting from program B3 (or VIDITEL) the normal character set is not reset, so the TI has to be switched off and on again (or use your Reset switch if you have one fitted), before running another program.

FURTHER HELP

I am willing to help any PRESTEL TI USERS via MBX 707873778 or by phone on 0707 873778.

PETER WALKER

DISK DRIVE PROBLEMS.

My first contact with computing came in 1983, when my wife presented me with a TI 99/4A, bought at Dixon's, being assured that availability was "Certain, as the Americans were using them in their schools" - this, it turned out was after Texas Instruments had announced their pulling out of the Home Computer market!

I, like most new owners, played about with it, managed to de-bug a program in the User's Handbook, and took programs from various computer magazines.

In 1985, about Easter, I decided that expansion must be the name of the game, as my wife said she would like a Word Processor, and I thought I could use the many File Handling facilities for work in my profession. I therefore began to accumulate "extras" such as Peripheral Expansion Box, TI 32K card, MYARC RS232 card, MYARC Disk Control card, and two MPI half/height DD/DS disk drives (I already had a TI cassette recorder, and an Amstrad 12" colour television as a cheap (and very good) monitor).

As the MYARC Disk Control card arrived without the Disk Manager disk (not the fault of Howard Greenberg, I must add) I asked Howard to supply a box of disks already formatted - thus not finding out that my set-up would not "format" properly!

When my next batch of disks were obtained, after receiving my Disk Manager disk from Howard, I began to wonder what had happened. One moment the system would format, the next not. Peter Brooks, who had supplied the drives, very kindly replaced one of the drives, without much difference, and then proceeded to embarrass me (I am sure, unintentionally (?)) by testing it and finding it to be working - later installing it for someone else with all success.

My next line of attack was to substitute the 32K card with a known good one - no use - then to try and get the MYARC Disk Control card checked by Howard - reported to be satisfactory. Another card was also loaned by Howard - again no use.

So to the Peripheral Expansion Box power supply.

My drives, MPI Type 502, need 12V±5% AND 5V±5%. The voltages available for the drive(s) are 12V on the purple wire, 5V on the blue, and 0V on the other two (orange and red). On my Box, they were 11.6V on the purple wire, and 4.7v on the blue (I was not able to measure current taken "on load") so contacted Parco - they referred me to TI's headquarters in Bedford.

After talking to one of their technicians, I sent my power supply board to them, and they said it tested out normally, being only 2% down. Further conversation obtained a replacement transformer also, which I fitted in hope of a complete cure. Guess what - now I had intermittent reading problems also, but was able to format a few disks after 10 attempts for 6 disks.

I felt the only thing to do at this stage was to get more information on the drives, but as MPI have pulled out of Great Britain, Peter was able to obtain little info., and I decided to telephone MPI in California. Guess what - MPI have stopped having anything to do with disk drives!!!

Now for a little good news - MPI have sold their disk drive division, including service responsibility, to T.R.W. in New Jersey, and kindly gave me the phone number. I rang them at 7:30 p.m. English time, and was told that they could supply a Development Manual which should give all relevant information, but that the Manuals had not yet arrived from California. They would despatch as soon as they arrived from California.

Later enquiries elicited the fact that they in fact despatched the documentation on 31st December.

The manual, when it arrived on February 6th, was for half-height drives all right, but for the "D" series drives - those with a Direct Drive motor.

Yet another phone call to T.R.W. established that they did in fact have a manual for the "Slimline" series, which was different! One of these was promised, this time Air-mail!

In the meantime, I heard of intermittent problems with the Flex Cable Card for the P.E.B., and rang TI in Bedford, with the result that on the 19th February I received a new Flex Cable Card to try. The disk drive that would not 'format' began doing so, but occasionally 'hung up', possibly due to reading/writing difficulties. At least some of the problem had been solved.

Surprisingly, after a short session of using the set-up, the reading difficulties became much less, usually managing to read successfully on a second attempt or after reading in a different program or file, then the difficult one.

The long-awaited Manual arrived on February 27th, and was found to be only an "Interim" manual - certain illustrations were missing, as were also the sections on "Troubleshooting" and "Optional Configuration". At least we now had a Technical Specification, even though it did not entirely agree with some of the information available in the U.K.

I understand that Peter Brooks now also has access to a similar Manual, so if information is needed about M.P.I. Slimline drives, help should be forthcoming.

For those who like to know such things, T.R.W.'s address and phone number are - T.R.W. Customer Service Dept., 20, Audrey Place, FAIRFIELD, New Jersey, 07006, U.S.A. (010-1-201-575-7110 EXT.4231).

I would like to thank both Howard Greenberg and Peter Brooks for help received with my drive problems, and hope that if anyone else has similar ones that I might be able to add my "twopenn'orth".

(Article written with TI-WRITER, and printed with the NLQ option on a STAR SG-10 Dot Matrix printer, and other equipment as described in the article.)

David Baines.

TI-WRITER PLUS

By PAUL MEADOWS from the September 1985 issue of TI Nova Scotia Newsletter

After a year of pounding away at the keyboard of my TI-99/4A using the very versatile TI-Writer program, I have finally managed to crack the biggest mystery the program presented.

Although most of my articles have been presented in 55 column format using compressed style print, I have dreamed of being able to force TI-Writer to print beyond its 80 column boundary in this print style. Since there is enough room for a hundred and thirty two characters on a line of 8.5 inch paper, why should I be limited to 80 columns ?

Well, we aren't limited to 80 columns after all! It was just the way we were going about printing that was holding us back. With a little thought, (a lot, really) a fresh approach and a lot of luck the right method would just fall out of the cartridge and away we would go.

Enough of this theorizing.... let's see the solution!

One or two precautions are warranted, first. If you are using tabulated information you will have to exercise extreme care; the FILL and ADJUST commands are essential to this process. Some experimentation with different form types might be a good idea, to get the feel of the technique.

Now, to the steps...

1. Prepare your file as you normally would with TI-Writer.

2. When you have finished typing in your text, go to line "0001" and enter the margins and commands, for example ".LM 10;RM 120;IN +5;FI;AD". This will set the left margin at a good starting point and the right margin will balance on the opposite side. The indent command, of course, is up to you. The Fill and Adjust are necessary. Watch out for any tabbed information!

3. Now, instead of SF for "save file", you use the PF or "print file" command preceded by the letter "C" and a space, which strips all control codes from the file. Along with the normal control codes, the TI-Writer tab line that was applied to the very last line of the file is stripped, thus taking away the mandatory 80 column indices.

4. Once the program has been "printed" to the disk using the above format, "C DSK1.FILE", you can exit the Editor and select the Formatter.

5. Select the Formatter, (option 2), and follow the prompts to print out the file.

Since the file no longer has the TI-Writer tab indices to look up, it will execute the internal margin and tab instructions.

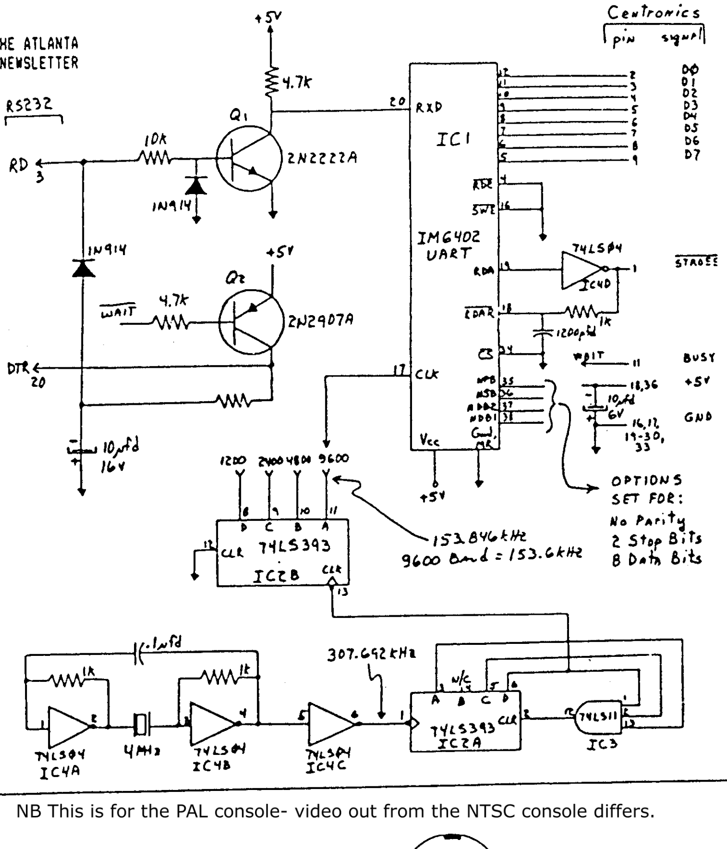
There you have it! Simple, isn't it ?

Please ensure you pass this information on to all TI-Writer users. It may be invaluable in some types of printing.

~~~~~~
~~~~~~

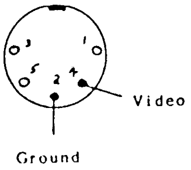
Here is a nifty project for the experimenter. Do you have a lot of software that is configured for RS232 yet you have a parallel printer? Well, this little gizmo will output on the serial port making the computer think it is talking to the RS232 and then on this external circuit all the data is converted to parallel for the parallel printer requirements. This circuit produces a CENTRONICS compatible interface. Good Luck.

-REPRINTED FROM THE ATLANTA 99/4A USER GROUP NEWSLETTER



NB This is for the PAL console- video out from the NTSC console differs.

-REPRINTED FROM SOFTEX, THE AUSTRALIA 99/4A NEWSLETTER



Video connector facing back of computer.

USING a BLACK and WHITE MONITOR.

The TI Video output is a color signal but an acceptable B&W monitor picture can be obtained by taking the 2 signals shown below to the monitor input.

Use shielded cable with shielding braid connected to the Ground pin.

Use what ever plug mates with monitor.

WATCHING A PROGRAM LOAD

By Peter Brooks. Originally written for HOME COMPUTING WEEKLY

One difficulty often encountered by Texas owners is that of loading a cassette-based TI BASIC program successfully. The volume may be set too high or too low, but you never find out until the middle of the loading or the very end, when an error message appears. It would have been nice if TI had provided a facility allowing you to vary the tape recorder volume level while the computer told you whether you'd got it right.

They didn't, but you can in fact monitor the initial stages of the loading of a cassette program in TI BASIC by using the USER-DEFINABLE CHARACTERS (UDC).

On the 99/4 these run from ASCII 96 to 159 inclusive, and on the 99/4A from 128 to 159.

The RE-DEFINABLE or standard characters (RDC) have their shape definitions stored in a ROM and when a TI BASIC program stops running, or when you go from the title screen into TI BASIC, these definitions are copied into an area of VDP RAM, where they are 'looked at' by the computer when it wants to define the shapes of the RDC, and you can alter the preset definitions using CALL CHAR(). Immediately following these in VDP RAM is another area into which the computer loads a program from tape or disk. It just so happens that this same area is used to hold the definitions of the UDC - when you use CALL CHAR() on these characters, the computer 'clears' space for the new definitions by 'shunting along' everything in this area of memory, so that there is no conflict over its use. If you don't use CALL CHAR() on the UDC, the computer has the use of this area of VDP RAM.

This means that if you place the UDC on the screen WITHOUT defining them with CALL CHAR(), you actually have a kind of 'window' on what the computer happens to be doing in that area - mostly it will be totally meaningless, but there are one or two occasions when it can be extremely useful.

If you begin OLDing a cassette program, for example, while the UDC are on the screen, you can watch the data coming in to the computer, for the initial few moments anyway. This in turn means that you can get a better idea about the volume level which you are trying, and, most importantly, about whether the machine is actually receiving any data at all.

On top of this, the computer is slow in responding to circumstances when it does not receive data, allowing you to quickly rewind the tape to the 'header tone', raise the volume slightly, and try again. A successful loading (initially) is signified by the UDC suddenly assuming random shapes. In the event that you are not fast enough in rewinding to try again, or if the volume level is either too high or fractionally too low, and data is missed, the computer will respond with the ERROR DETECTED IN DATA message, forcing you to start again. In this case, you will need to go back to the title screen in order to 'wipe clean' the UDCs, or they will retain their random shapes and you may not be able to 'see' whether the second attempt at loading is being met with success.

Select TI BASIC, and if you have a 99/4 enter this line:

```
        100  FOR A = 96 TO 159
```

If you have a 99/4A, enter instead:

```
        100  FOR A = 128 TO 159
```

and then for both models, enter these lines:

```
        110  PRINT CHR$(A);
        120  NEXT A
```

Don't forget the trailing semi-colon in line 110 - it's very important.

Now RUN the program, and when the computer reports 'DONE', begin the OLDing sequence with OLD CS1. The first time it is best to use a cassette program which you know will load successfully.

Once the header tone has passed, you should observe an 'army of insects' marching across the bottom line of your screen. If you don't, quickly rewind to the header tone, raise the volume level by one mark, and re-play. Provided you are quick enough (not necessarily Olympic standard) you can go on doing this until the insects start marching. If you subsequently get an ERROR DETECTED IN DATA message, that will most probably be caused by the volume being just a fraction too low. If you get the message straight away, the volume may already be too high, distorting the signal and causing the computer to miss data, in which case you need to reduce it and try again from scratch.

Incidentally, if you have a tone control, it is best to set the treble as high is possible, and paradoxically, the CHEAPER the tape recorder that you use, the more likely you are to have trouble-free OLDing and SAVEing.

The reason for this is that the more expensive tape recorders tend to have additional circuitry in them which is designed to remove unwanted hiss and other noise. To this circuitry your computer program/data sounds just like unwanted noise, and it is usually 'interfered with' in an attempt to reduce its level.

This is a rare instance of 'cheap and nasty' coming out on top...

~~~~~~
~~~~~~

LEARNING TI FORTH

By MIKE RICCIO. from the newsletter of PHILADELPHIA AREA TI-99/4A USER GROUP

Starting at square one

As you can see by the title, this column will be a tutorial for the programmer beginning TI FORTH. For those of you who have some experience with TI FORTH already, this may be going too slow for you, but bear with me. For those who have little or no experience with FORTH, I suggest you go out and buy an excellent book by Leo Brodie called "Starting Forth".

In order to use FORTH, you must first load it from disk, because FORTH is not like BASIC, which is built into the computer. Do the following in order to load FORTH:

1. Turn on all peripherals first, then the console and television (or monitor).

2. Insert the Editor/Assembler cartridge.

3. Press any key, then press 2 for "EDITOR/ASSEMBLER".

4. Press 3 for "LOAD AND RUN".

5. Insert your TI FORTH system disk, or the new one you made from the previous articles, into drive 1.

6. Type: DSK1.FORTH and press <ENTER>.

7. Wait until FORTH is finished loading and you see three columns of "loading options", the words TI FORTH, and the flashing cursor.

The oresence of the cursor means the same thing in FORTH as it does in BASIC; the computer is waiting for you to type something. If you happen to make a typing error BEFORE YOU PRESS <ENTER> , you can correct it by backspacing over it with the left arrow key (FCTN-S). This key works the same as in BASIC with one difference, it erases everything it passes over. Note: the right arrow does not function in TI FORTH. Also, as in BASIC, you must press the <ENTER> key when you're done typing to let the computer know that it is now its turn to do what you told it to.

The "loading options" at the top of the screen are commands that will load additional commands for specific purposes. Before entering any of these loading options, you must have your system disk in drive 1. Note: these extra commands take up memory, thus shortening the maximum length of your programs.

If you are familiar with TI Assembly Language, TI FORTH should also look very familiar to you. If you are an Extended Basic programmer, then some of FORTH's commands will be identical to their Extended Basic cousins. This is what I feel TI FORTH is, a cross between Extended Basic and Assembly Language.

Before actually learning how to program in FORTH, we must learn its structure. Unlike BASIC and Assembly Language, FORTH is a very structured language for the simple fact that it doesn't have a "GOTO" statement.

{There is far more, however, to being a "structured" language than not possessing the GOTO command. PB}

FORTH's basic structure is this: define small, useful tasks and give each a name. Then, group these smaller tasks together to form larger tasks, giving each one of these new tasks a name. Continue this until the whole process is grouped under one name.

As an example of this, let's picture a washing machine controlled by a computer. The ultimate command would be called WASHER. Here's how this might look in FORTH:

```
: WASHER   WASH SPIN RINSE SPIN ;
```

The colon (:) indicates the start of a new "definition". The first word after the colon is the name we want to give this new definition. The remaining words up until the semicolon (;) are what this new word is supposed to do. The semicolon marks the end of the new definition. Each of the above words have already been defined, so let's look at the definition of RINSE:

```
: RINSE  FILL AGITATE DRAIN ;
```

As you can see, the larger task (WASHER) has been broken down into smaller tasks, each of which is broken down even further:

```
: FILL  FAUCETS OPEN TILL-FULL FAUCETS CLOSE ;
```

If we were to continue tracing these words back, we would eventually find the basic FORTH vocabulary. As you can see, FORTH doesn't care whether you made the command, or if it was built in, FORTH treats them all the same.

One problem with FORTH is that you can't write the above program in that order. It must be done in the REVERSE order, with the small tasks FIRST, building up into larger ones. In other words, you can't put a word in a definition until you define its own function. Whatever you do, DON'T try to write programs from "the bottom up", because your mind thinks from "the top down". What you should do is write your programs on paper, designing them from the top down, but then later, enter them from the bottom up.

This all may seem perplexing at the moment, but I hope this example will clear things up a bit.

A real-life "bottom up" example (taken from Starting Forth) :

Imagine you're an office manager and you've just hired a new, eager assistant. On the first day, you teach the assistant the proper format for typing correspondence. (The assistant already knows how to type). By the end of the day, all you have to say is "Please type this".

On the second day, you explain the filing system. It takes all morning to explain where everything goes, but by the afternoon all you have to say is "Please file this".

By the end of the week, you can communicate in a kind of shorthand, where "Please send this letter" means "Type it, get me to sign it, photocopy it, file the copy, and mail the original". Both you and your assistant are free to carry out your business more pleasantly and efficiently.

FORTH lets you organize your own procedures and communicate them to the computer in just this way (except that you don't have to say "Please").

Well, now that you know how to correctly structure your programs, we can begin to learn FORTH's syntax, but I'm out of space, so it will have to wait until next time. Note: You can apply this structure to any program you write in any language (although it's hard in BASIC because it's oriented around line numbers). Until next month, live long and program!

{BASIC and its line numbers are no barrier to structured programming; rather it is the lack of logical and structured thinking which is the greater problem. PB}

~~~~~~

BEGINNING TI FORTH

By MIKE RICCIO, from the newsletter of PHILADELPHIA AREA TI-99/4A USER GROUP

Interactive FORTH

FORTH is called "interactive" because it executes (i.e. performs) your commands the instant you enter them.

Try it:

Press the <ENTER> key. FORTH will respond with "ok".

Pressing the <ENTER> key tells FORTH that you are done commanding it, and it's now FORTH's turn to execute these commands. On the other hand, "ok" is FORTH's way of telling you that it has completed what you told it to do. Because you didn't tell FORTH to do anything in the above example, FORTH readily did nothing and responded with "ok".

Commanding FORTH

FORTH is always ready to accept your commands when you see the cursor (the two exceptions are when you're in the EDIT mode or a program is accepting input; more on this later). Many commands can be entered at the same time as long as they are separated by at least one space.

Try it, type the following and press <ENTER> :

```
42 EMIT
```

The computer responds: "* ok".
~~~~~~

The FORTH word EMIT expects a number on the stack (the stack will be discussed in detail next time - for now let's just say that a number must be typed before the word). EMIT will print the character that has the ASCII code of the given number. (Refer to Appendix A in the TI FORTH manual). {Think of it as BASIC's PRINT CHR$(). PB}

Try it. How would the computer respond if you entered:

```
77 EMIT 73 EMIT 75 EMIT 69 EMIT
```

If you don't know the answer, then type it in and find out. The computer should respond "MIKE ok".

Your First Program

The colon, (:), followed by a space, will tell FORTH that you want to extend its vocabulary. The first word following the colon will be the name of this new command. The remaining words up to the semicolon (;) are the functions that this new command will perform. The "definition" MUST end in a semicolon.

Try it. Type the following and press <ENTER> :

```
: MIKE 77 EMIT 73 EMIT 75 EMIT 69 EMIT ;
```

From now on, when you type MIKE and press <ENTER>, the computer will respond "MIKE ok". Notice that this works exactly as if you had just typed in all those EMITs. Have fun and try it with your own name.

Programs Within Programs

Try this. Continue typing until the semicolon, then press <ENTER> :

```
: ME CR MIKE 32 EMIT 82 EMIT 73 EMIT 67 EMIT 67 EMIT 73 EMIT 79 EMIT ;
```

CR is another FORTH command which will move the printing to the beginning of the next line so it will look neater. {It stands for CARRIAGE RETURN, and TI-Writer users will be familiar with it. PB}

All of the EMITs should look familiar, but notice that I included the word MIKE which WE defined earlier. The point is that FORTH doesn't care whether you made the command or if it's one from the "kernel" (FORTH's original commands that are loaded when you power-up), FORTH treats them all the same. When you type ME and press <ENTER> , FORTH should respond on the next line with "MIKE RICCIO ok".

Next time we'll go into the realm of the dreaded STACK and Numbering Systems but until then, live long and program!

~~~~~~
~~~~~~

SORTING AND SEARCHING

Peter Brooks April

This issue we will tarry awhile (as they say) and examine the use of a single, two-dimensional array when applied to the sorting routine given last issue. First, though, there are three dumps for you to compare with the output from your version(s) of the routine from last time. It's all very well me giving you the routine to type in, but I need to show you the result(s) that you should expect to see!

```
        SORTING BY SURNAME
-----------------------------
0  BROOKS      OXFORD       P
1  BROOME      LINCOLN      A
2  BRUGGE      RHEIMS       O
3  BURNETT     LIVERPOOL    A
4  COOPER      OXFORD       A
5  CRUMB       NOTTINGHAM   A
6  FIELD       READING      A
7  FUTTOCK     CREDITON     A
8  GARDNER     MILWAUKEE    O
9  HERRESTHAL  HAMBURG      O
10 JAMIESON    WINDSOR      A
11 LEWIS       HULL         A
12 MANTLE      MORDEN       A
13 MARCHAM     DONCASTER    A
14 MASSIF      PARIS        O
15 MATHIS      LAS VEGAS    O
16 PAULI       CAPETOWN     O
17 ROLANDS     BIRMINGHAM   A
18 SALMON      MONTREAL     O
19 TRINDER     CANBERRA     O
```

```
     SORTING BY POSTAL TOWN
-----------------------------
0  ROLANDS     BIRMINGHAM   A
1  TRINDER     CANBERRA     O
2  PAULI       CAPETOWN     O
3  FUTTOCK     CREDITON     A
4  MARCHAM     DONCASTER    A
5  HERRESTHAL  HAMBURG      O
6  LEWIS       HULL         A
7  MATHIS      LAS VEGAS    O
8  BROOME      LINCOLN      A
9  BURNETT     LIVERPOOL    A
10 GARDNER     MILWAUKEE    O
11 SALMON      MONTREAL     O
12 MANTLE      MORDEN       A
13 CRUMB       NOTTINGHAM   A
14 BROOKS      OXFORD       P
15 COOPER      OXFORD       A
16 MASSIF      PARIS        O
17 FIELD       READING      A
18 BRUGGE      RHEIMS       O
19 JAMIESON    WINDSOR      A
```

```
SORTING BY MEMBERSHIP STATUS
----------------------------
0  BROOME      LINCOLN      A
1  FIELD       READING      A
2  COOPER      OXFORD       A
3  CRUMB       NOTTINGHAM   A
4  FUTTOCK     CREDITON     A
5  MANTLE      MORDEN       A
6  JAMIESON    WINDSOR      A
7  ROLANDS     BIRMINGHAM   A
8  BURNETT     LIVERPOOL    A
9  MARCHAM     DONCASTER    A
10 LEWIS       HULL         A
11 BRUGGE      RHEIMS       O
12 HERRESTHAL  HAMBURG      O
13 PAULI       CAPETOWN     O
14 MASSIF      PARIS        O
15 GARDNER     MILWAUKEE    O
16 MATHIS      LAS VEGAS    O
17 TRINDER     CANBERRA     O
18 SALMON      MONTREAL     O
19 BROOKS      OXFORD       P
```

If your results WERE different, let me know as soon as you can, together with the EXACT routine that you used (in case you made an error in typing, cough, cough).

Now to this month's main subject, at the end of which you may begin to wonder exactly what advantage there is to be gained by it!

The example routine given later should produce results which are similar to, but not identical with, the combined printout given above. Your task for this month is to figure out why the results might not be identical.

However, last issue's routine has gained a degree of complexity, brought about largely by the inclusion of additional options, and the use of a sneaky bit of maths derived from a project on which I have been working now for some considerable time (like since 1982, on and off!).

That project has bred a number of spin-off projects, all of which will, in the fullness of time, receive an airing in TI-LINES. You have been warned!

This particular spin-off is a technique for deriving an algebraic formula (or expression or equation) to give a specific series of numbers. In this instance, I wanted an equation which would take the numbers 0, 1, and 2, and calculate the results as 4, 15, and 28.

There are a number of ways of doing this (involving, for example, a series of Relational Expressions, of which more later in another series).

The method I chose involves the use of the little-known CALCULUS OF FINITE DIFFERENCES, and although I won't go into great detail here, it is a very useful technique to have under your belt when trying to work out the relationship between one series of numbers and another.

You may have been wondering why I should want to establish a relationship between 0, 1, and 2, and 4, 15, and 28. The answer is (reasonably) simple. The use of a two-dimensional array is helpful not only when sorting, but also when performing data INPUT (in this case READ) and OUTPUT (in this case PRINT). The 4, 15, and 28 refer to the TAB() values used in line 330 last issue; the 0, 1, and 2 refer to the elements which will contain Surname, Postal Town, and Membership Status in the routine to come shortly.

The PRINT routine from last issue changes from:

```
FOR P=0 TO 19
PRINT STR$(P);TAB(4);S$(P);TAB(15);T$(P);TAB(28);M$(P)
NEXT P
```

into:

```
FOR P=0 TO 19
PRINT STR$(P);
FOR S=0 TO 2
PRINT TAB(S*S+10*S+4);L$(S,P);
NEXT S
PRINT
NEXT P
```

which admittedly takes up more space, but which is now a general purpose routine capable of handling a large number of PRINT fields, using only one massive list: L$(,).

The comma in the brackets is used to indicate that it is a TWO-dimensional array.

One word of caution: the expression within the TAB() is accurate ONLY for values of S of 0, 1, and 2. That is, if you tried expanding the list to include another parameter - like SEX, for example - (alright, alright, settle down at the back!), and if you made S loop from 0 to 3, then apart from the fact that the display is too narrow to include anything else on the line, the use of S*S+10*S+4 will yield a TAB() position of 43 for an S value of 3, which leaves too large a gap between the STATUS field and what would be the proposed SEX field (calm down!).

No doubt this discussion is confusing you, in which case DON'T PANIC! You don't have to understand how the expression works in order to use it in this example. However, if you want to use it for your own purposes, well, yes, NOW you can panic.

One of the advantages of using a general approach here is that you can choose which key you want to use for the sort, and I have included a CALL KEY() scan to test for the pressing of keys 1, 2, or 3 to select one of the keys. (Don't forget, a sort key has nothing to do with a typewriter key!).

The routine presented here allows you to go back and re-sort according to a different key, without having to set up the array all over again.

The data INPUT section, using READ, is also slightly different, and although I could have coded it thus:

```
    FOR P=0 TO 19
    READ L$(0,P),L$(1,P),L$(2,P)
    NEXT P
```

I chose instead to keep it general, thus:

```
    FOR P=0 TO 19
    FOR S=0 TO 2
    READ L$(S,P)
    NEXT S
    NEXT P
```

If you were using a list with, say, 10 different parameters instead of 3, the original routine to READ last issue would have become horrendously long and complicated. The new READ routine would simply change the S loop from FOR S=0 TO 2 to FOR S=0 TO 9.

Now perhaps you can begin to see some of the advantages of the general approach compared with the specific one. (If the terms "general" and "specific" don't mean much to you, please let me know and I will explain them in another issue).

If you have typed in, and stored, the routine from last issue, retrieve it and delete all lines between 120 and 350 inclusive (unless you were smart and stored the DATA section separately!). Then you won't need to retype all the DATA again, only the active part of the program.

Here is the full listing:

```
   100 CALL CLEAR
   110 CALL SCREEN(8)
   120 DIM L$(2,19)
   130 FOR P=0 TO 19
   140 FOR S=0 TO 2
   150 READ L$(S,P)
   160 NEXT S
   170 NEXT P
   180 PRINT "PRESS:-","FOR:-":"-------","-----"::::
   190 PRINT " 1.","SURNAME"::" 2.","POSTAL TOWN"::" 3.","MEMBERSHIP"::::
   200 PRINT "SORT ON WHICH KEY"::"(1,2, OR 3) ?",
   210 CALL KEY(0,K,S)
   220 IF (K<49)+(K>51)THEN 210
   230 PRINT CHR$(K):::"SORTING...":::
   240 K=K-49
   250 F=0
   260 FOR P=19 TO 1 STEP -1
   270 IF L$(K,P)>=L$(K,P-1)THEN 340
   280 FOR S=0 TO 2
   290 G$=L$(S,P)
   300 L$(S,P)=L$(S,P-1)
   310 L$(S,P-1)=G$
   320 NEXT S
   330 F=1
   340 NEXT P
   350 IF F=1 THEN 250
   360 CALL CLEAR
   370 FOR P=0 TO 19
```

(continued overleaf)

```
 380 PRINT STR$(P);
 390 FOR S=0 TO 2
 400 PRINT TAB(S*S+10*S+4);L$(S,P);
 410 NEXT S
 420 PRINT
 430 NEXT P
 440 PRINT :::"PRESS <ENTER> TO CONTINUE";
 450 CALL KEY(0,K,S)
 460 IF K<>13 THEN 450
 470 CALL CLEAR
 480 GOTO 180
1000 DATA BROOKS,OXFORD,P
1001 DATA BROOME,LINCOLN,A
1002 DATA FIELD,READING,A
1003 DATA COOPER,OXFORD,A
1004 DATA BRUGGE,RHEIMS,O
1005 DATA HERRESTHAL,HAMBURG,O
1006 DATA PAULI,CAPETOWN,O
1007 DATA CRUMB,NOTTINGHAM,A
1008 DATA FUTTOCK,CREDITON,A
1009 DATA MANTLE,MORDEN,A
1010 DATA JAMIESON,WINDSOR,A
1011 DATA ROLANDS,BIRMINGHAM,A
1012 DATA BURNETT,LIVERPOOL,A
1013 DATA MASSIF,PARIS,O
1014 DATA MARCHAM,DONCASTER,A
1015 DATA LEWIS,HULL,A
1016 DATA GARDNER,MILWAUKEE,O
1017 DATA MATHIS,LAS VEGAS,O
1018 DATA TRINDER,CANBERRA,O
1019 DATA SALMON,MONTREAL,O
```

The subsections are as follows:

100 to 120: Initialise; clear the screen, set up the screen colour, and reserve space for the list (or array), called L$(,).

130 to 170: READ the DATA into the array L$(,).

180 to 240: Print the Options Menu, scan for keys 1, 2, and 3 (by ignoring anything pressed which has an ASCII code less than 49 (=1) or greater than 51 (=3)), echo any valid keypress to the screen, notify the User that the sort is under way, and finally subtract 49 from the ASCII code of the valid option chosen. Why 49 ? Well, L$(,) will be using L$(0,), L$(1,), and L$(2,), and K will be either 49, 50, or 51. Subtracting 49 from K means that K will also be either 0, 1, or 2, and doing it at this point removes the need for messy (and time- and space-consuming) equations later.

250 to 340: This is the heart of the program, most of which should be familiar from last issue.

250: Resets the flag.

260 to 340: Constitutes one pass of the sorting routine.

270: The all-important testing of adjacent entries.

280 to 330: The actual swap routine, still using G$ to temporarily hold the item being swapped, but this time in a FOR-NEXT loop. If there were 10 fields to be swapped in tandem, the only alteration would be to change line 280 to FOR S=0 TO 9. The flag is set to indicate that at least one swap has been performed.

350: Checks the flag after one complete pass to see if the sort is finally complete.

360 to 430: Printout routine for the sorted list.

440 to 480: Allows the User to reselect the key on which the sort is to be performed, but still keeps the current sorted list on the screen until ENTER is pressed, so that the User can examine it at his/her leisure.

1000 to 1019: DATA as before.

To stop the program, you will need to press CLEAR/BREAK (FCTN 4) - the option to finish by selecting from a menu was not included. Things are complex enough as they are.

THE RIPPLE SORT

Now that you have (hopefully) successfully entered and run the sorting routine presented here, it would be useful to examine another form of Bubble sort, known as the Ripple Sort. Don't look at me, I only report on these things.

Now, you already know that the Bubble sort starts from the bottom of the list and "bubbles" the out-of-place entries upwards to their rightful place. The Ripple sort, wait for it, works in exactly the same way, except that it starts from the TOP of the list and works its way DOWN. It's all true, I tell you.

All that needs changing, therefore, is line 260, to FOR P=1 TO 19. The effect is that instead of nudging "small" entries UP the list, the Ripple sort nudges "large" entries DOWN the list.

THE SHAKER SORT

There is a third, more esoteric, form of Bubble sort known as a SHAKER Sort. This combines a standard Bubble sort with a Ripple sort (yes, I do know that it's April, this is not a joke). The idea, as I understand it, is that having made one pass through a list using the standard Bubble sort, if a swap has been made (so the list might still be unsorted), then instead of commencing another pass from bottom to top, a pass is made using a Ripple sort.

If again a swap was made, the program would resort to another pass using the standard Bubble sort.

Why all the palaver ?

Well, if you stretch your memory back to an earlier issue of TI-LINES, when a standard Bubble sort on a simple list of single letters was dissected, you may recall that towards the end of the sort, repeated passes had to be made in order to bring the letter Z all the way down from the top of the list to its rightful place at the bottom. For every complete pass through the list, only one swap was made each time. If a Ripple sort had been used, only ONE pass would have been necessary, as Z would have been nudged downward after every swap. You see, there is occasionally method in the madness.

Alternatively, instead of alternating between standard Bubble and Ripple, you could modify the algorithm so that the Ripple was only brought into play at the most crucial moments - i.e., when only one swap is being made in each pass (indicating a "large" entry slowly on its way down the list).

The flag is the key section. As it stands, no matter how many swaps are made, F will only ever be a maximum of 1. If however you altered the way in which F was set, from F=1 to F=F+1, I think that you might see how the testing could be changed.

If F is 0, no problem, the list is sorted. If F is greater than 1, then a number of swaps have been made, and the sort is probably operating reasonably efficiently.

However, if F is exactly 1, then only one swap will have been made during the pass, and it just MIGHT be worthwhile running a pass through using a Ripple sort at that point.

The same criteria then apply to the Ripple sort. If, after a pass, F is 0, then no problem, the list is now sorted. If F is greater than 1, then the Ripple sort is probably operating reasonably efficiently. If F is exactly 1, that indicates that it just MIGHT be profitable to go back to using a standard Bubble sort, as there MAY be a "small" entry on its way to the top of the list.

OK, you say, where's the listing for the Shaker sort ?

Well,it's only of minor interest, like the Ripple sort, and hardly worth wasting too much space over, especially when compared to the piece de resistance (a bit of a resistor, or "she keeps saying 'No'"), known as a SHELL, SHELL-METZNER, or 'D' Sort (and probably a few other things as well).

I prefer 'D' sort - it's shorter to type - and like Tag sorting, the principle can be applied to the other forms of Sort which we will examine in the coming issues.

Next time, then, the 'D' sort, and also how to store the original positions of a list with the list itself so that the original order is not lost even after sorting.

Hold onto your socks, because the 'D' sort may be simple, but it is very fast indeed.

~~~~~~

Magazine scanned 2022 by Stephen Shaw
~~~~~~

CLOSE FILE

AS A MATTER OF FACT...

I had been going to reprint a list of CALL LOAD() values from one of the newsletters which I receive, until I took a closer look at some of the descriptions accompanying the values. It became impossible to distinguish between fact and belief, as many descriptions appeared to stem from the "poke a value in any old address and see what happens" approach. For example, "Produces 'mushy' keyboard with improper characters", or the even more inscrutable "searches disk", or "locks up computer", or even "screen goes wild". Equally uninformative (and therefore probably unreliable) is "random garbage" or "random characters on screen".

I hope that this doesn't sound too snide or supercilious, but it should be fairly obvious that such descriptions do not advance the User's practical knowledge of the system. Imagine a medical textbook that discussed the Nervous System by stating that a poke in the eye with a sharp stick elicited a shriek from the subject, followed by an assault on the investigator.

If LOADs or PEEKs are to be of any real value, then they must describe a known effect (like the LOAD which disables the QUIT key), or at the very least if they are based on "suck it and see" experimentation, they should clearly say so.

MAY BE LATE

The MAY issue (like this one, yawn) is going to be late. I will hold the publication until after the TI-EXCHANGE meeting on the 3rd., so that I can write it up in the usual roughage-based style.

AS YOU WERE...

It takes a fair degree of mental effort for me to remember that a number of OTIUsers have only recently acquired their machines and are new to almost all aspects of TI computing. I am therefore trying to include some articles which will be of primary interest to them, so if the elder statesmen amongst you spot the odd item which has seen the light of day before (maybe many times before!), you will now know why.

RAMDISK PROJECT

After having seen mentions of a 96K home-grown RAMDISK project in the excellent newsletter of the NEW HORIZONS and OH-MI-TI groups, I am wondering if OTIU doesn't have within it the capability of producing such a very useful item (especially for me with my total reliance on TI-Writer, which can be excrutiatingly slow at 3 in the morning!)...

Program 'til the cows come home...and then program 'til they go out again...

Peter Brooks