

TIHOME PRICE LIST.

TIHOME MEMBERSHIP (English)	£12.00
TIHOME MEMBERSHIP (Foreign)	£18.00
Back copies of Tidings	£ 1.50
Programs (supply your own tape) per program	£ 1.00
Listing of programs	£ 1.00
Instructions on the use of the PRK module	£ .50
Pin out/in diagrams	£ .50
Errors in Extended Basic	£ .50
Book : Beginners Basic	£ 2.53

HINTS ON THE USE OF CASSETTE RECORDERS WITH THE T I 99/4A.

1. Check the distance between the cassette and the T.V. The longer the distance the better.
2. Stick to short tapes. Nothing longer than C60. Better stick to program tapes from Smiths.
3. Fast forward and rewind occasionally. Tapes get stuck if you only keep playing the same piece again and again.
4. Check the plug in the back of the 99/4A. Put a matchbox under it to stop the weight pulling down.
5. If you run on batteries -- don't. Always work on the mains.
6. We have had trouble with Ferguson.
7. We have used a Marantz C190 and have never had any trouble.
8. Cassettes with a manual record level are better.
9. Recorders with a manual tone control are better. Computers like treble not bass.
10. Check your 3.4mm plugs. You may find that the T I ones are not the correct length.
11. Check your Alpha Lock key is down. The cassette recorder does not recognise lower case characters.
12. CS1 is the only cassette that will read and write programs.
13. If your motor will not work, check the possibility that the polarity needs to be reversed.
14. Check that you have good connections by pushing the plug in the back of the console. See 4 above.
15. Don't forget to clean your record heads. Dirty heads make dirty data. Remember GIGO. Garbage IN Garbage OUT.

Any more problems
ring me :

Paul

01.640.7503

29th November, 1982,

Mr. M.C. England,
4 Pinetree Close,
Bernton,
Northwich,
Cheshire.

Dear Mr. Dicks,

In TI Home you said that it would be better to address Texas Instruments through the magazine rather than direct contact, OK then, I hope you will be able to publish this letter in the next issue.

Dear Mr. Frowd,

This weekend (27.12.82) my son and I decided to visit the Northern Computer fair at Belle View, Manchester.

We set off early with great expectations to Manchester and after a tortuous journey through the outskirts and an hour long queue we were finally inside.

On purchasing an Exhibition guide we scanned the contents for Texas Suppliers. At last we found one (which turned out to be the only one out of nearly 100 exhibiting stalls). This was Home and Business Computers of Manchester. The "Write up" promised "Home and Business Computers will have on show the TI99/4A with full speech capability. You can see and hear the machine in use" - "General education and special education software is available and includes packages for games, personal management" - "The TI home computer with speech synthesiser can be seen teaching children from four years old how to program using TI logo".

Great we thought, speech synthesis 32K RAM, peripheral box and even possibly disc drive demonstration, mini memory etc. here we come. This is the reason we came, demonstrations, queries and even a possible purchase.

After fighting my way through the mass of people in front of the stall (My God! I thought, all these people taking an interest in the Texas). I reach the demonstration desk, where was the Texas now?

On hearing the stand I had heard a loud voice, audible over the general noise. "One plus so and so equals etc." Good I thought at least Texas is making itself heard. I looked around. The voice had been coming from a vic computer a few stands away.

At first there was no sign of the Texas on the stand. A vic - 20 was nightdriving with a group of amazed spectators looking on. An Atari was blasting away at a space game - but where was the Texas? I looked again, and there tucked away behind the Atari was the Texas. The non-existent speech synthesiser must have got a sore throat. I looked again, "Oh dear", the Texas must be ill it was tucked up in bed - no I looked again and it was sitting there - silent - unworking and unloved.

"What about the Texas?" I asked the stall holder, "What about it" he said, "Well have you Munchman?" I enquired, "Sorry sold out" he replied, "Now Madam about the vic-20, we are offering etc. A bit non-plussed we walked on. About five stands

Continued

..... Continued

-2-

further down, we spied a Texas working - or at least with the Texas logo on the screen. "My God" I thought again there's an epidemic of them. Two of them in a space of 50 yards. "Do you sell Texas software" I enquired, "Sorry no, we are selling the stand it's on" was the reply, "The Texas is on loan."

Well at least he had the thing working. We walked around for about an hour hopefully to spot another Texas amongst the dozens of vics, dragons, acorn's , BBC etc. but alas no.

After this time I thought I would tackle stall 410 again. The people were still going and aching at the Vic and Atari. The Texas however was still tucked up in bed. "Have you Space Invaders for the Texas" I enquired, "Sorry sold out" was the reply. (Actually we have it but my idea was to at least get it on the screen and show the other poor people how good Texas Graphics are compared with the rest). "Well what about"? :- and I went through the list of Texas module he had on display for sale. By the end of it the poor old Texas in bed had turned a deathly green and started to sweat. "Sorry, sold out" was the reply. The Texas stopped sweating "Well how about Winging it?" I said, "Sorry sold out". "Star Trek", "Sorry sold out", By this time the colour had returned to the Texas and it had turned over and returned to it's slumbers. A faint sigh coming from the non existant synthesiser. In a last valliant attempt I asked. "How are sales of Texas going", "Very well was the reply". "I could sell HUNDREDS if I could get my hands on them". Now madam about this vic-20. We departed from the stand and do you know on turning round a little further on, catching a fleeting glimpse of the disappearing Texas it wasn't in bed at all but standing in the shadow of the Atari by its side and for all the world I could have sworn that above the hum of voices in the hall I could hear faint sobs coming from the non existant speech synthesiser, perhaps it was missing the four year olds.

The story however has a happy yet sad ending, happy for us, sad for Texas.

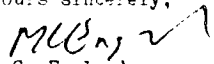
We had left my wife and daughter in Altrincham shopping (better sense than us).

We originally bought our Texas from Boots. They have now stopped selling it in preference to the Dragon, Vic and ZX81. However, they still had some Texas comand modules left and we bought a chess module for £20.00 and Tombstone city for £12.50, not bad, and they still had some left. So to any of 'TI Home' Readers - get a long to your local Boots and see what is left - you might be lucky.

Now come on Texas, I don't know what you and your merry men are thinking Robin but for goodness sake do something.

If a shop can sell in his words 'Hundreds' and can't get any what's the matter? Someone somewhere wants a swift kick in the derriere. Why is it at the show - an ideal opportunity before Christmas - that the place was inundated with every other computer but no Texas. You have missed the Christmas boat - for goodness sake don't miss the next, - or I and others will be the processes of a system which does not exist anymore. I could go on about why is it the best or one of the best selling in the States, why don't you advertise on the T.V. etc. but I am sure you take my point. If you need another sales executive down there at Texas I am available - at a price - but available. I have already persuaded three people on the merits of a Texas system and they have bought one - thats three up on you. Don't let them - and the rest of us down.

Yours sincerely,


M.C. England.

Random Lines

by
Mike O Regan

NEW FORMAT

As there have been a few complaints about the legibility of RANDOM DOTS after it has been photo-copied TWICE I am trying a number of experiments.

Number 1 is to try the whole column on the WORDS SMALLEST WORD-PROCESSOR. This is written on a SHARP PC 1500 and CE 150 Plotter/Printer, which, in case there is someone out there who hasn't seen the write-ups, uses a miniature ball-point pen to write each separate character in addition to drawing high-resolution graphics! The system is capable of printing in NINE DIFFERENT SIZES in all four directions. This is written in the smallest size and the title is in SIZE 6. It can also print in green, red or blue, by rotating the pens.

The program, which I have called PICOWRITE, runs in about 200 bytes. It prints up to 36 characters per line (8 more than my JJ program). The SHARP BASIC allows all characters, except double quotes to be ENTERed. The length of acceptable string may also be dimensional (default is 16 chars) up to 88 chars per string, which is a valuable feature.

As the PC 1500 has only 2K of RAM, I have had to reduce the number of features of this program to the bare essentials, which are:

1. WRITE..text - up to 25 lines before either printout or DUMP to tape. The machine allows upper and lower case (with caps lock, if required).
 2. READ..back the text with line: numbered to facilitate EDITing.
 3. EDIT..is limited to whole line replacement, by calling up the required line by number. Note that attempts to WRITE lines which exceed 36 chars results in the line being shortened to the nearest whole word, with an audible warning and display of the shortened line.
 4. PRINT..finished text, with double line spacing for legibility.
 5. SAVE..to tape, which is a very simple (one line) procedure.
 6. LOAD..from tape - just as simple.
- 4.8. the really neat feature of this system is that any procedure may be interrupted at any time (by the BREAK key) without losing either program or data. This allows, for instance, text to be interspersed with titles in a bigger type size.

SPACE LASER BUGS

If you keyed in the nice little game SPACE LASERS (from Computer & Video Games-Nov) you may have been a little miffed when you found it had one or two bugs. There was also a little matter of no running instructions. I wonder how the author (GH Program?) managed to get a listing on what appears to be a SINCLEAR ZX PRINTER. Anyway, for those who didnt spot the bugs, the following may be of some help. After the following lines:
210 K# should be BRASS etc.
103A CX-CX+INT(RND#3)+1
105B CX-CX-INT(RND#3)+1
Insert the following lines:-
925 IF P0# 2 THEN 388
926 P0-P0+3
1015 IF P0# 32 THEN 1020
1016 P0-P0-3

These changes will enable the game to be played, but still dont explain the rules. However, it should be a nice little exercise to sort out and enter your own rules!

WEARING SPECS

Following up STEPHEN SHAWs advice about VDU viewing distances and wearing specs, when I got some rather stronger reading specs a few months ago, I found that they were not suited to viewing the screen, so I have kept my previous pair especially for the job. With these I can see every dot on the screen, which is not possible with my new ones.

ExBas SPEED

I cannot understand why the idea is around that ExBas is slow. It may run various BENCHTESTS at a slower speed than II BASIC, but I have found that many programs written in II BASIC run quite a bit faster in ExBas, especially such functions as CALL HCHAR/UCHAR. Also the facility of being able to by-pass the PRE-SCAN makes programs take-off that much faster. However, I have found the CALL SAY statements, especially when dealing with numbers or single alpha characters to be quite a bit slower than the SPEECH EDITOR Module.

BARGAIN BASEMENT

Where is the current trend of price slashing going to end? Surely the continuing fall in the price of the 99/4a cannot do much for its image, especially among those who have paid quite a bit more within a matter of weeks. For a short period it was possible to buy at CURRYs for £199, less the £50 Texas cash refund, less CURRYs own offer of £50 worth of software. Now I see a firm advertising in PCW to supply the 99/4a for £135, less the £1 money-back offer. Thats just £85!!!

TI SOFTWARE OFFER

Thanks to ROBIN FRODGS kind offer of SOFTWARE for PUBLISHED PROGRAMS I now have two new modules to add to my growing collection. (2311 at the last count - can anyone top that?). As a matter of interest there is only ONE AFA (99/4a), with the other 22 evenly divided between LIA and ACH (prefix for 99/47) which originated in HOLLAND.

I am a bit concerned by the new trend to produce modules which will not operate on the 99/4. (the new Word-Processor, for instance). Also worrying is the trend towards cheaper CARDS (for TIBOX. Will these ultimately oust the PROM modules altogether?

On the other hand, I have two modules which WILL NOT RUN ON THE 99/4a. Not only will they not RUN, they are not even recognised by the 4a console as being plugged-in!

99/4 v. 99/4a

As a long-time 99/4 owner, I am naturally interested in comparing it with the newer machine. So far my experiences with the 4a have done little to persuade me that it is a vastly superior machine to its older brother. The much-vaunted keyboard is not all it is cracked up to be, in my opinion. It seems that not much thought has been given to the SHIFT and FUNCTION keys. I realise that the 4a has quite a few more functions, but the character allocation between SHIFT and FUNCTION leaves much to be desired.

Also, the actual spacing of the keys seems to be more cramped. The slip-in label above the top line is no real substitute for the full overlays which are one of the better features of the 99/4, even compared with other makes. For instance, I am currently doing some research into alternatives to the QWERTY keyboard layout. This makes an overlay, (or keycaps) essential, and it would be difficult to fit caps to the 4a keys. Another drawback to the 4a is the quality of the SCREEN display. I have yet to see a display which even approaches the MISC display which I get from my 99/4 & SKANTIC combination (and that includes the displays on the TI Stand at the PCW SHOW!)

MORE ON PRINTERS

With the demise of the TI Thermal Printer and its replacement being (as usual) too expensive, I have been looking around to see if there are any reasonable alternatives. The tiny printer which has produced this column has a B16 BROTHER, available from TANDY for £149 (sorry about the pound sign). This can probably be attached to the 99/4 through an RS 232 interface (incidentally, does anyone know if the new TI Printer needs a separate interface?). What I don't know is how to access the plotting routines through TI BASIC.

POCKET COMPUTERS

Now that I have finished this column, I am quite impressed by the facility of producing it on a Pocket Computer. Some may say that to call the SHARP PC 1500 a POCKET computer is stretching more than the imagination, as, when it is attached to its printer it would take a rather large poachers packet to accommodate it! My PC 1500 is just the last of a longish line of programmable pocket devices, which started with the humble SINCLAIR ENTERPRISE. This was followed by a TI 50, a number of CASIOs, two SHARP PC 1211s (one of which I still have) and two HEWLETT-PACKARD 41Cs. However, the PC 1500 is the first one capable of any form of Word-Processing. For those of you who have not yet tried a Word-Processor, and do all your writing either long-hand or on a conventional typewriter, the advantages of a PORTABLE word-processor may not be immediately obvious. It is quite something not only to be able to write anywhere, independent of mains power, but I also find that using a word-processor aids the flow of thought in a way which is hard to describe if you have not experienced it. Anyway, it is quite fascinating to see the tiny ball-point pen chuntering away at a rate of knots. Quite apart from using the system for writing, I find I am using it in the office more and more (I am a local government officer). In my department we get a large

number of ad-hoc data handling jobs which have traditionally been done manually, largely because such jobs were thought to be too trivial and/or one off. Some of these jobs have been very time-consuming and, if they involved repeated calculations, were subject to many errors, either in the manuscript or typewritten copying. Although I have repeatedly shown that a pocket computer is a valuable tool for such jobs, I still have great difficulty trying to persuade my colleagues that we should be taking more advantage of the NEW TECHNOLOGY. I Personal computing is a very SOLITARY OCCUPATION!!

GEMINI MARKETING

I have just had a letter from (member?) NJ SEWARD from Cornwall. He tells me that he ordered three programs from GEMINI, who advertise extensively that they stock 99/4a software. You must have seen CHOPIN - LES ETUDES etc. Mr? SEWARD tells me that not only did he NOT RECEIVE the software he ordered, but several phone calls to GEMINI have established the fact that they have NO SOFTWARE for the 99/4 IN STOCK. I have just phoned GEMINI, who said that they haven't got 99/4a programs in stock YET!!! but would have some NEXT WEEK!! I seem to have heard that selection from CHOPIN before.

Dear PCT,

First the buttering up. Congratulations on your new magazine. It is an excellent informative one. Second the 'battering'. I must take sides with your writer J. Davies in your September issue in defence of the Texas TI 99/4A. When I saw the article on the Texas v Atari, I thought "at last a magazine that is going to take notice of the much maligned Texas". Not so however. Why is it that you and other magazines do not come down on the side of the Texas model. I have had it described to me as a "Rolls Royce" among computers, which I agree.

In your article comparing the 99/4A and the Atari you do not mention that Atari is made from and looks like a cheap bit of plastic, and as far as the buzz when they keys are touched - more like a very inaudible squeak of a dying mouse. The Texas however is very sturdily built, is extremely pleasing to the eye, has proper keys and is easy to edit.

Also lets have mentioned that in total the price of nearly £300 for the Atari against the Texas £200-£225 including tape recorder. A price difference of some £70-£80 and please correct your fact file at the back, the price of the Texas should read £199.

The Texas programs are not difficult to load from a tape recorder but are very simple and easy.

Lastly how about being the first magazine to print a Texas programme, there is a dearth of them. It would ensure a regular purchase. I hope you can print this letter in its entirety to give people a chance to buy a good computer which I am reliably informed is now selling like hot cakes.

Yours faithfully,
M. C. England.

Editor's reply: So the battle for the Texas continues! It is very difficult when writing a review not to offend somebody, however carefully you consider the arguments. I will admit that the Texas has been much maligned by the computer press and a lot of this ill feeling stems from the first couple of abortive launches of the computer during which it received quite a pasting. I would disagree with your comments on Atari, but everybody is entitled to opinions.

The review couldn't have been too anti-Texas because I have received a letter from a reader who, after reading the review, went out and bought a Texas. Suffice to say that whatever the good or bad points of a machine there will always be somebody somewhere who thinks it is the best.

Salleh Chemat,
87, Chancel Row, Hyde Park Flats,
Sheffield S2 5LD, S. Yorkshire.

ADVERTISEMENT

- | | | |
|----|--|---------|
| 1. | TI EXTENDED BASIC (Brand new) | £60. 00 |
| 2. | TI INVADERS (also new) | £12. 00 |
| 3. | Cable lead to cassette recorder | £ 6. 00 |
| 4. | Book: Programming Basic for the
TI computer by Herbert D. Peckham | £ 8. 00 |

All 4 items for £84. 00
add 1. 50 for postage and packing
£85. 50

Reason for selling - badly need the money. Cheque made payable to S. B. CHE'MAT.

ADVERTISEMENT

11, Coldblow Crescent,
Bexley, Kent DA5 2DR

FOR SALE

3 BACK ISSUES OF 99' ER MAGAZINE.
VOLUME NO.1 NUMBERS 2, 3, 4, 5.

£2.95 each plus postage.

Immaculate Condition.

CRAYFORD 521429

Dear Paul,

Thank you very much for the back copies of 'Tidings'. As a new member of TI HOME who has read the last five issues 'at a sitting'. I feel obliged to put pen to paper and say a few words of thanks.

It is obvious that 'tidings' exists due mainly to the considerable effort of a small band of contributors who naturally wish to see their ranks added to and need feedback to the effect that their efforts are worthwhile and appreciated.

I'm afraid I'm not likely to be able to contribute anything of great value but will try a scribble to see what comes out of it. I most certainly can however say a hearty thank you and echo many of the sentiments which I've recently read on TI's pricing policy and availability. The latter has been my major hobby horse. I've been trying in vain for eighteen months to find an 'Editor/Assembler' and this week have been simultaneously informed by TI that 'there should be no problem with availability' while told by every supplier I've phoned (U.K. wide) that it is 'impossible to obtain from TI'.

The main reason I purchased a '99' two years ago, in preference to one of the many more competitively priced units, was that I use TI kit in my office/daily work. This allows me the luxury of access at weekends and holiday periods to an 810 printer. Another of TI's 'hiding light under bushels' was painfully experienced. Much trial and error was needed to build up the knowledge on such essentials as 'form feed' (Print \neq x : CHR \$ (12)) or line feed (Print \neq x : CHR \$ (13)) or even the appropriate OPEN statement to enable a 99 to print 132 character lines in place of the default 80, and many more. The RS 232 booklet was quaintly silent on such matters. I appreciate that TI would not have envisaged that many 99 users would have access to an 810 but surely the RS 232 manual should contain some reference to the use of a printer and appropriate examples of TI Basic commands and statements!

This reticence is consistent with TI's marketing policy which remains silent on the 99's potential for use as a small business machine. I have found it to be an excellent tool for this purpose despite a complete lack of commercial software. As an example I currently run a Vehicle Repairs and Maintenance Report for a 150 vehicle fleet on a single disk which contains the program (insufficiently User Friendly to be much use to anyone else I fear) and twenty two items of data per vehicle. The report produce 18 'branch' reports with 'branch' totals and calculates m.p.g. and pence p. mile per vehicle and average per branch.

Before acquiring my 99 and embarking on the learning curve I was quoted £2500 by a software house to write such a programme - sobering thought, eh!

May I please have copies of volume one issues 1 - 6. I know these were not as large as volume two and may be cheaper but as I only sent £1. each for those received (sub divided by 12 oops) I'm also sending £1. each for volume one and happily looking forward to the postman's delivery.

Keep up the good work

Yours gratefully

Wynne

F.W. GRIFFITHS

Babbling Brooks

By the time you read this it will probably be 1983, so a belated Merry Christmas and Happy New Year to all! You probably won't be in a fit state to appreciate all the fine things in this issue of Tidings for a good couple of weeks (alcoholic poisoning, you know) so I will have a fortnight's breathing space before getting back down to 'hacking' - writing. In fact I spend so much time writing for Tidings these days that I have hardly any time left for writing original programs (my excuse for not having submitted anything to the popular mags!). Part-way through scribing this issue of BB I managed to lay my sticky fingers on a reel of photoprocess ribbon, and I hope that it will improve the legibility of BB when it is reduction-photocopied. The original doesn't look that good, though, so I have my doubts.

A lot has been happening in the last two months - TI dropping their prices and giving that £50-back scheme a try, for example - much of which was left out of last issue's BB because of lack of time to prepare anything. Some of what you read this issue therefore actually happened as long ago as September 1982!

1983 looks set to be the year that prices fall lower than anyone dreamed they could. The Japanese are waiting in the wings with goodies at prices which will pin your ears back - how about a full plotter for around £90 ? - and if that makes other micro-manufacturers look to their laurels that is no bad thing. I continually harp on about the fact that Britain is the high-priced dumping ground for many companies; with any luck (and if there is any justice) that should change in '83. Look out for cheap disk drives - and I don't just mean Uncle Clive's - with very high density storage, and for video disks (see elsewhere), as these are due to be launched shortly with the Read/Write versions waiting in the wings (probably with the Japanese!). Why am I so over-the-moon about video disks ? Because TI woke up and produced a controller which can be used with one - the Pioneer system - and wisely chose a laser-based system to go with. Phillips and Pioneer are the people to watch in the coming years in this field.

Britain is apparently now the 'most computer-aware' country in the world, with more personal computers per capita than even the Americans. Without wishing to thump the tub, it's about time someone recognised that our potential lies with the Future, and got off their backsides and did something concrete about it. Members of TIHOME and other like groups have done it; now all we need is for the politicians and senior management to wake up and do likewise. If the decision-takers don't follow us, there's

a very real risk that the projected 'Golden Age' will never be reached.

On to more interesting things. Thankyou for your letters - if you haven't had an answer yet then I probably haven't received it! - and because no-one specifically said that their comments were for Soapbox I haven't quoted from any of them. Some have said that I can quote them on certain subjects, and a short piece on what they have to say about Tidings, Software, TI, and Prices, appears elsewhere (Open Quotes).

A word about the deliberate 'errors' which I have included in the past. The general idea was to toss in a bit of nonsense amongst the correct information (some might say it was the other way about) just to see if anyone actually read and understood enough to point out the nonsense. As it happens, the accidental errors were picked up, but not the deliberate ones ! Deliberate errors therefore are out. They were probably too contrived anyway, but they are a common occurrence in certain forms of Maths, where the author of a thesis on what might be termed 'riddles' will throw in a rogue statement or two, just to see if those propounding solutions REALLY understand the problems.

Last issue I dished out some quirks with regard to REMless comments and the use of "." for TP output. I have since tried "." on a 4A with no success, so it looks as though it may be restricted to 4s, and maybe not all of those either. In the meantime, personal communication with STEPHEN SHAW (Stockport and 'Raables' author) and PETER PHILLIPS (Glasgow) has elicited so much specific information with regard to the REMless comments that I have reduced both their results and my own researches to a set of general guidelines about how and where comments can most usefully be employed. Those appear elsewhere (Basic Comments). In the meantime, another bug has reared its head: the failure of POS to function correctly under certain circumstances. It occurs when the length of 'string 1' (in which you are looking for a possible occurrence of 'string 2', starting at position P) is in excess of 127 characters. If the length of string 1 is 127 or less, POS functions normally. If it is exactly 128, then a value of zero is always returned, regardless of starting position (P), and if it is 129 or over, the starting value will have to be set to 128 if the first occurrence lies after the 127th character. This only came to light when I was attempting some initial input routines to create a 'header' and 'pointer' string for a simple data management system. I used a minor trick with INPUT - specifying a null input prompt - which gives you a full 4 x 28 character input (usually you either lose space to any input prompt, or if you don't specify a prompt you lose space to the "?" which is inserted) i.e.: INPUT "" : 15 - and found that subsequent decoding wouldn't work (the total string length was usually around 200 characters, achieved through successive INPUTs and concatenation) when using POS to search for an asterisk field separator. The simple answer is: don't use POS on strings whose content is more than 127 characters.

Stephen has also pointed out an 'inexactitude' in my review of Extended BASIC V110 in issue 4: using ON BREAK you do not necessarily get stuck with an unBREAKable program, requiring a general reset (FCTN = , or Sift Q) to get unstuck. If your routine has a halt for input (ACCEPT or INPUT) then you can BREAK in the normal way.

Stephen has also been giving me some answers to my queries over CALL PEEK and searching for program listings. Since buying the Mini-Memory module some of that information has been underlined, with the result that the clouds of uncertainty have largely blown away, to be replaced by just mists of uncertainty.

It appears that CALL PEEK and CALL LOAD can only address CPU RAM; that is, memory which is directly addressable by the central processor. The additional memory provided by modules like Mini-memory and by peripherals like the Expansion Memory are also CPU RAM. The console only possesses a mere 256 bytes (1K) of RAM in the standard system; your programs and the system's own variables are stored in VDP RAM, under the control of the video display processor. In order to execute your program therefore the CPU has to address the VDP's RAM as though it were a peripheral, like the Thermal Printer, through the CRU (Communications Register Unit - I have also heard it called a Control Register Unit). The CPU uses just two bytes of its 256 in order to carry this out. Further details are not yet available.

Hearsay USA

To find out about the origin of this item, read the section on computer magazines.

As Sol Libes describes his article in BYTE (Oct 82) as News and Speculation, I take no responsibility for the accuracy of the following items. I haven't quoted verbatim either, in order not to enter the tricky field of copyright infringement!

1. The Americans have had 'Read-Only' disks for a number of years now; they are in the process of being released over here. Some systems are mechanical (they require a stylus to be in contact with the surface of the disk) and as such are subject to very rapid wear and tear. One system's manufacturer's have said that after 30 plays, their disk is worn out. Other systems do not involve a physical contact with the disk surface - they are described as 'optical' disks - and use a combination of laser technology and 'pits' burned into the surface of a plastic-coated disk to store enormous quantities of information. However, they are still 'Read-Only' - you cannot record on them yourselves. Yet. For this reason, video-cassette systems have not been unduly threatened (although early systems here were also 'Read-Only') but Sol's information is that the 'Read/Write' disk is on its way here. Next year,

a 'Write-Once' system is expected (no prices) which will make use of special erasable optical disks to store some 10,000 Mb (megabytes - 1024K). These disks will operate a little like PROMs (programmable Read-Only memories), chips which can be programmed, but which, if you make one boo-boo, cannot be re-written. You have to use a fresh one. Even if this sounds like nothing to get excited about to you, be advised: it is.

2. The Americans call them 'Flat Panel'. We call them 'Flat Screen'. Whatever you call them, the first of a new range of TVs are on their way. Uncle Clive Sinclair is due to launch his first flat-screen TV soon (should have been this Autumn, but now may not be until March '83) for around £50. It was to have been the subject of a special Science Watch, only, ahem, I lost my references. However, Uncle Clive was only just ahead of the pack. Other manufacturers have been experimenting with different systems; Uncle Clive's is a true CRT (cathode ray tube, like in the usual TV), while some of the competition is based on different principles. For example, 'Tomorrow's World' have already shown two systems, one involving lasers and scanning mirrors, the other involving gas discharge tubes (an enormous screen designed for mass audiences, not the home). Others include LCD (liquid crystal diode) panels, electroluminescent screens, and 'gas plasma' which may/may not be similar to gas discharge. IEM and yes good old TI have demonstrated large flat panel displays, but they are not expected to include them in micro-computer systems in the near future. Some companies are, though; Sharp have a 320 x 240 pixel electroluminescence (!) display with bubble memory and built-in modem with of course a computer for about \$2800 to \$3200 (perhaps a little expensive for the home computer end?). Siemens claim to have a 2 1/4" thick 28 x 80 character plasma display - word processing on the train, maybe? Even as I write, Epson, makers of printers (I think the MX80 is currently putting on TI clothes), have launched a neat little package with proper QWERTY keyboard, built-in printer and micro-cassette, and a 4 x 20 'window' screen of LCD pixels, all for about £400. I paid more than double that for my system, and that was without printer or cassette! Admittedly the facilities cannot compare with current, CRT-based, systems, but it's a first step in the right direction. You have to look ahead, to when the 'child' has come of age, to see the potential. Uncle Clive has been working with ICL on a flat-screen terminal for the mini environment (look at the advantages: low power requirement, smaller, lighter, and probably easier to maintain - you can carry them about a lot more readily) which is rumoured to be colour as well, and to top it all, Clive is also launching another, business, micro next year, which will apparently feature a full-sized flat screen with built-in Microdrives (see also below) for about £350. I can see an awful lot of home buyers going for that one!

3. Remember Josephson Junctions? Science Watch issue 3? Well, IEM have just started pilot production on a facility for manufacturing such devices. They may be due to be used in ultra-fast IEM machines in about 5 years time.

4. Even floppy disks are coming in for a little redesigning: the 5¼" systems are being challenged by new, higher density, 3" versions. Often called micro-floppies, these may yet be themselves challenged by video (and audio!) disk systems. One company, Adek, claim their micro-floppies can hold around 500K in double-sided, double-density format. Compare that with the TI single-sided, single-density of 92K or so. Adek are apparently asking \$899 for a twin-microdrive system (offering 1 Mb), but even as I write, Sony are launching their micro-drive, a 3½" system with single-sided, single-density capacity of about 219K. Double-density is about 438K. The cost for the double-density system would be £235, but you have to add to that the cost of a controller to suit the particular system with which the drive would be used. That could still compare very well with the current cost of TI drives and controller; however, for me the price is still not competitive enough. Sinclair's microdrives may only offer 100K and may have an extremely long access time (3 secs) probably because they are not true disks, but the price is far more realistic. If anyone doubts that, look at it this way. Suppose the cheapest car you could buy was £10,000, but your finances would only allow you to spend £3,500 maximum, and you heard that a perfectly acceptable car was going to be produced for £2,500. Would you scrimp and save for three years to buy one of the £10,000 machines? Or would you rush out and buy the cheaper model and make do with the minor deficiencies? If you'd scrimp and save, then you're the type of customer who was made for firms who charge unjustifiably-high prices. At present, for most micros the cost of buying a printer, disk drives and controller, or some of the other more esoteric peripherals, is far higher for each individual item than the cost of the computer itself. If you accept the car comparison, then a trailer for your car would cost you between 2 and 5 times as much as the car. Sinclair is probably the only manufacturer who is making both his computers AND his peripherals cheap. Cue some crazy manufacturer pricing his micro at £3000 so that his printer (£400) and disks (£700) ARE cheaper than his micro!

5. The floppy disk saga gets taken a step further. Syquest Technology have produced a 3.9", 6.38 Mb hard disk which is removable. If you know something about disks, then there are two very interesting facts there! Not only that, but two drives can be mounted in the space occupied by one 5¼" drive.

6. It seems that TI are expected to introduce shortly an 8 bit portable micro, with a 1 x 40 character display for about \$250, and a compatible 4 colour printer will also be available. (I've just been reading about Sharp's latest brain-children, one of which is a colour printer for £85 to go with a new, 64K micro for £175). I have no further details on TI's machine.

7. A quick post-script to the flat-screen info: Sony have a gas plasma display with

1024 x 524 pixels (compare that with our 192 x 256), aiming to market a full product incorporating it next year ('83). Apple, too, are reportedly working on a flat-screen-based micro.

8. Finally, one of my favourite topics. The Ford Motor Company are said to be working on a 'cockpit' computer with display to assist in the control of functions in Ford vehicles. The computer will provide a navigation system tied to a communications satellite, displaying a map with the vehicle's location pin-pointed on it. You may have seen a recent Tomorrow's World item on a Japanese system which involves a monitor and transparent overlays, with the computer providing a 'track' of where you have been. This kind of thing is what I have been predicting for the last three years (each year I get the date wrong!) - it's nice to see the seeds beginning to grow!

In issue 4, readers might remember my comments in the Extended BASIC review with regard to being able to RUN one program from another, and how 'visions of vast Adventure programs' had receded into the distance, as all variable would be cleared, thus data could not be passed from one program to another. Stephen tells me that it could be possible to transfer some data in this simple fashion. You code the data you want to pass - i.e., current level in the Adventure, items currently carried, etc., - so that it can be passed off as string(s) of hexadecimal digits. Thus 12F7 might represent Level 1, Room 2, carrying 15 items, and 7 lives remaining. You then use this string to redefine one (or more) of the user-definable characters. When the program has been loaded, the data can be retrieved using CALL CHARPAT, and decoded.

I can see only one minor flaw with this, although I haven't yet tested it out. Back in the early days I presented a 'loading monitor' which enabled you to monitor the OLDing of a program from CS1 by watching the user-definable characters being redefined by the incoming program... As I haven't had time to test this out, does anyone fancy trying their hand? You'll need Extended BASIC and either cassettes and/or disks. Stephen had tried a simple program (2-3 lines) using disks and it had worked, but what about a 5K program, and does the presence of a disk system alter the storage locations for any incoming program? All offerings gratefully received (and acknowledged).

Other News

Netherlands TI-owner Bill van Kerkoerle (he sped up the 3D plotter - issue 4) has written to me, offering to provide articles (in Dutch, I think, which I would translate before publishing) so sometime early in the New Year we may have another contributor,

another viewpoint. I have an article already which I will attempt to translate and then submit to Bill for critical appraisal of my handiwork. If I manage to make a fair job of it (translating a foreign language into your own tongue is usually easier than the other way about), it could open the door for many more Dutch contributions; at present we are limited to those provided by English-speaking contributors who have taken the time (and not inconsiderable effort) to translate their article(s) - Paul Karis has a small advantage in that he is involved with the running of a Language School - and it must be said that the command of the English language shown by most Europeans is far greater than our command of their language(s). English is a compulsory second language in just about every European country, and in many other countries around the world.

Dave Hewitt of Hoddesdon wrote to me in mid-September with some queries which I have passed on to TI for their comments (they may have presented something in the last issue). One of Dave's proposals was that the 99s might be used in conjunction with a video recorder to produce professionally-scrolling titles and graphics for home productions - perhaps superimposing them on existing images - or using the recorder to save pleasing patterns produced by the Designs program (issue 3). Any takers? In addition, has anyone managed to obtain 'direct' video+sound - i.e., without the PAL modulator - on their video recorders? If you have any thoughts on the matter, send them in, and I will pass them on to Dave and publish them in the next convenient issue of Tidings.

BASIC Comments

The general conclusion that I have to draw from the information I have about these 'REMless comments' is that the 'quoted' variety (see later) are probably the only ones worth using, and the best place to use them is when transferring control from one part of a program to another. The 'unquoted' variety, of which there are two sub-species, are so memory-consuming, and so awkward to implement, that they are just worth noting and nothing more. Nevertheless, researching into this particular phenomenon has been very interesting, and has given an insight into certain of the computer's operations.

There are basically two kinds of non-REM comment in TI BASIC: those which can be placed within quotes directly after the last destination line number in a jump command, and those which must conform to the requirements for numeric variables. This second category can be sub-divided into two further groups: those which can employ words coinciding with 'Reserved Words', and those which can't.

I have classified these as QUOTED (Q) and UNQUOTED, VARIABLE formats 1 and 2 (U1, U2). Stephen's information is that Q types use far less memory than U1/U2, and in fact can in some circumstances use less memory than equivalent REMs. Q types appear to be usable

only after destination line numbers in jump commands - specifically after the LAST line number in a statement. U1/U2 types can also be placed in similar positions - subject to their own requirements - and can be used elsewhere; for example in TRACE, UNTRACE, BREAK, UNBREAK, RESTORE with line number, END, STOP, and even in some CALLs.

Q types simply require the comment to be placed within quotes, thus:

```
1000 GOTO 500"JUMP TO RE-ENTRY POINT"
```

whereas U1/U2 types are slightly different:

```
1000 GOTO 500+JUMP+TO+RE-ENTRY+POINT
```

In this second case, JUMP, TO, RE, ENTRY, and POINT are all incorporated into the variable list - this can be simply checked by entering such a line (doesn't have to be syntactically correct - you don't need line 500) and when the 'program' crashes, typing

```
PRINT JUMP(0)
```

whereupon NAME CONFLICT will appear, because the use of JUMP(0) implies a numeric array, and JUMP already exists in memory as a 'simple' variable. Note that RE-ENTRY is stored as TWO variables: this is because the two words are separated by a valid mathematical operator: "-". Another reason for avoiding U1/U2 types is this: the use of the division sign (/) will lead to error or warning messages as a result of the system evaluating division of zero by zero (NUMBER TOO BIG). While this is not a problem with GOTO, it could be in certain circumstances:

```
CALL CHAR(98+U1/U2-EXAMPLE,"")
```

Additionally, a form of comment may be added to the end of the definition string in CALL CHAR statements, being either tacked onto the end of a (minimum) 16 hex digit string (which then makes it incompatible with Extended BASIC) in an 'explicit' string i.e.:

```
CALL CHAR(98, "0102040801020408FOR EXAMPLE LIKE THIS")
```

or concatenated with a similar limitation thus:

```
CALL CHAR(98, A$ & "OR LIKE THIS")
```

Note that A\$ MUST contain at least 16 hex digits for this to work without crashing. You may perhaps see now why Q types are favoured!

The BASIC interpreter appears to have a conflicting set of operational rules: for example, Q type comments can be added after destination line numbers because the interpreter does not search further than the line number given - it may assume that because the syntax allows nothing further, then there is no point searching further - yet no comments can be added after the closing bracket of the argument list for a CALL - where no additional data can be used anyway - because the interpreter examines anything after the bracket and denounces it as an error.

I say 'appears' to have a conflicting set of rules, because I suspect that the BASIC interpreter (in TI BASIC) may in fact be geared to use a facility currently only associated with Extended BASIC - multi-statement lines. With multi-statement lines, there is little point searching after a GOTO instruction for further commands to execute, as anything following such a command (on the same statement line) will never be executed, unless the destination line number forms part of a conditional expression like IF...THEN and the additional commands follow THEN but not ELSE. (This doesn't explain why you can put a comment after the ELSE line number, as if multi-statements were supported, further commands could exist after ELSE.)

After other commands, though, a multi-statement facility would allow further commands to be executed:

```
CALL CLEAR :: A = 10 :: GOSUB 1000
```

Of further interest is the fact that you can prepare and list (but not RUN) Extended BASIC programs complete with double colon statement separators (see also Mike O'Regan's Random Dots in issue 3) using TI BASIC. Before you begin licking your lips, however, such programs are coded differently from TRUE Extended BASIC multi-statement programs. I SAVED such a TI BASIC-created program, and then OLDED it back into Extended BASIC. The double colons were treated like PRINT separators (in Extended BASIC this means placing a space between them - i.e., PRINT ::::: becomes PRINT : : : ;) and thus the above example becomes:

```
CALL CLEAR : : A = 10 : : GOSUB 1000
```

and won't RUN under Extended BASIC either!

Open Quotes

Here are a few quotes taken from letters I have received.

"I have given up getting 99er because of very late deliveries from MPI and demands for extra cash...Tidings is now the only source of information on the TI 99/4....You can quote me on TI pricing policy and their abysmal distribution to dealers...I have just bought a chess module and find that all the key controls are given only for the 4A. It wouldn't have cost much to give those for the 4 also. The packaging too is not as good or as convenient as it used to be....

Peter M. Phillips, Glasgow.

Now on to the rest of Babbling Brooks' usual garbage dump. The BASIC and Maths articles are continuing, as is the Three Dimensional Noughts & Crosses (you can't have everything!) and Science Watch looks at a high-speed train system for the USA. No-one's been moved sufficiently to write to Soapbox, so I'm going to jump on the Predictions bandwagon and sound off about Changing Times. (I would probably have done that anyway!).

If you fancy looking ahead with Tidings, Babbling Brooks has plans for some new items for 1983. If you have anything you want to say about anything under the Sun then now is the time to get writing. I want to begin a forum for information exchange on different peripherals - Speech, for example. Has anybody been playing with SAI and SPGET and done something interesting? Write in if you have. Can you make your 99 sing with TEII? Do you fancy making music with Mike O'Regan or going Rambling with Stephen Shaw? Have you organised a local 99 group, and if so, what have you been doing? Do you think that you can do a better job of introducing BASIC to beginners? We are all ears (funny looking lot, eh?) and eager for your contribution. At the end of 1983 I want to be able to sit back and relax and read someone else's brain dump in Tidings. Blow the dust off your thinking caps and sharpen your typewriter, especially you girls! We know you're out there...

If anybody lives in the Oxford area, well, never mind, drop me a line anyway, and the same goes for anybody else (don't forget to sign your anonymous letters): send it all to:

Pete Brooks, c/o 68 Kelburne Road, Cowley, Oxford OX4 3SH.

If you don't have access to a typewriter but you'd like to contribute, send me a neatly-written contribution and I will type it out for you. If you'd like any help with an article get in touch and we'll see what can be done. We're only after you for your minds...

REVIEW : MINI-MEMORY

Recently I took delivery of another module from Science Studio (Oxford) - run by one of our members, Chris Poole, - the much-vaunted Mini-Memory. I was actually in the shop when the delivery was made, so I know how long the module sat around on the shelves; i.e., not at all. Why should I want to know that, the uninitiated might ask. Because the module has a battery on board which keeps your precious data stored on it even after you have switched the console off and pulled the module out of the machine, that's why. A flat battery means no retention of data, and the battery is not 'guaranteed' to hold its charge for a definite period, so don't entrust the details of your Swiss account to it !

The module came in one of the new-style boxes, together with a cassette in a protective plastic wallet, documentation for the 'Line-by-line Assembler' - of which more later, - the Mini-memory manual with a sheet detailing one correction, and a large chunk of yuk yellow plastic to hold everything in place.

Avid readers of this garb..er..section of Tidings might have noticed a certain animosity expressed by me towards TI's pricing, and although I was given a very reasonable price by Science Studio, it was still a good £20 above what I would consider the MAXIMUM price for this item. My reasons for buying the module are purely mercenary: I hope to develop a number of programs which would otherwise not be possible, and sell them for pots of money. If I can find the time. If I can find my brain. As it happens, the module has left me with that vaguely dissatisfied feeling which you get when you lose 50p and find 49½p. Allow me to give you my verbatim impressions (or are they ad hoc impressions ?) covering the 24 hours since I picked the module up.

First impression: Drool, drool, machine code within my grasp at last, heh, heh, heh!

Second impression: Utter confusion as I try to take everything in at once.

Third impression: Irritation as I start to uncover all the types (printing errors) which look as though they have been custom-designed to trap the unwary. Or thick, as in my case. Further irritation as I find the little things it can do and the bigger things it can't. Exhaustion at 2.30 a.m., with concomitant typing errors (on my part) adding to the generally blue atmosphere. Final capitulation to sleep at 3 a.m.

Up with the lark next morning, hung-over and bleary-eyed, but now with an inkling of the intellectual challenge ahead coupled with not a little anticipation.

Sound good so far ?

Two things have immediately forced themselves to the front of my excuse for a brain: a decent plotting facility (I have a 99/4, not a 4A - you lucky people!) and perhaps a disassembler (a what ? the non-machine code buffs might exclaim. Well it sort of, you know, undoes what an assembler does...cue verbose article on Assembler...). I managed to write one (on and off in two years, mostly off) for another 16 bitter from Digico (in octal yet!) which could disassemble paper tape input into 64 recognised instructions, all in less than 590 words, with output to a teletype, and I think that there are about 70 - odd instructions which can be implemented on the TMS9900 system (in 10 formats, compared with only 5 for the Digico), so...

Returning once more to plain English, I am beginning to see what a drawback the old 4s might be, especially as I understand that there are no plans to offer 4 owners a plug-in replacement VDP to match the 4As. (Author demonstrates peeved expression to sheet of A4.) I have managed to obtain a sketchy idea of what lies where in memory, although I haven't yet worked out where the locations for redefining the user-definable characters are; any attempt so far has trespassed on an area of memory obviously used as a store for loops under execution or something equally esoteric, as any program manages to end before it should. If you get my drift.

A number of things seem geared to the TI animal called Expmem (Ramus Dearnus Americanus Dinosaurus) - cue re-run of peeved expression - but all in all I'd say it was better than a kick in the teeth, mind you, almost anything's better than a kick in the teeth...

In addition, you would appear to need the manual from the Assembler/Editor command module box - manuals without accompanying GROMs have been notoriously elusive in the past, as some members have found when trying to find out whether Extended BASIC was the right one for them - as well as being experienced in programming in assembler already, which seems a bit of a raw deal. Mind you, other manufacturers do that: they publish opcodes and mnemonics for their respective processors without telling you how to use them - you're supposed to have found that out for yourself.

As usual, some functions are restricted to cassettes and others to disks, which is BORING, and there is no facility for hard copy (i.e., printing), but it might be possible to tweak the Line-by-line Assembler which is supplied on cassette with the module. There is also a program called Lines on the cassette which is a demo for the high speed, high resolution full-screen plotting facility CHLY available to 4A owners. I believe I saw a version of this running during the PCW show recently and it looked quite good.

The module has two facilities: Easy Bug, and Mini-Memory. Easy Bug allows you to look at or modify RAM contents (both VIP and CFU) and to execute machine code routines. It

also allows you to look at and modify CRU bits - a rather hazy area which has to do with Input/Output (I/O) operations. There is also a facility for loading from and saving to CS1 only. If you want to load a machine code program into the module from cassette, you do it using L in Easy Bug. You can specify start and finish addresses for saving the program (it will load back into those same locations), but as the manual points out, if you want to make sure you don't lose anything, it's probably better to save the whole 4K.

If you want to load the LINES or the Line-By-Line Assembler (LBLA) programs from a cassette (I wonder how you'd copy it to disk ?) you have to load it using Easy Bug. If you want to use LBLA, you appear to have to exit Easy Bug and return to the title page, selecting Mini-Memory (MM) from the subsequent menu. This presents you with 3 further options: LOAD AND RUN; RUN; and RE-INITIALISE. LOAD AND RUN works, yes, you've guessed it, not with cassette storage, but with disks. Quote from the handbook:-

"The LOAD AND RUN option allows you to load and execute assembly language programs developed with the EDITOR/ASSEMBLER package and stored on diskette." Grrrrr!!

The RUN option allows you to either run a machine code routine, like LINES, specifying it by name, or to use the LBLA, which is probably the most useful item. Where would MM be without this cassette-based afterthought ? Depending upon whether you want to begin writing a new machine code routine or continue with writing one already stored, so you specify NEW or OLD. The difference between the two is that NEW clears the symbol table (a sort of variable list) first. There is a 9 page memory available which can hold what you type in the LBLA and you can scroll at speed backwards and forwards through it. Once you exit the LBLA all the 'english-like' information disappears though, so you shouldn't rely on the LBLA alone for writing the routines. There is no facility for hard copy either, so pencil and paper are essentials.

Having written your program, you cannot run it under LBLA unless you give it a name first (you can't call it up from TI BASIC either until you name it), which is a little fiddly and one would have thought that the LBLA should have been capable of doing that automatically for you.

The LBLA allows you to enter your machine code in the form of MNEMONICS: i.e., english-like versions of the instruction codes. For example, LI RC, >C200 means load the following value (hex C20C, or decimal/denary 512) into register RC. Much more intelligible than C20C C20C. The LBLA will translate the semi-english commands into '9900' opcodes for you as you enter each line, checking labels and incorporating them into a table,

checking commands and syntax, even converting decimal/denary values into hex. That's a rather confusing bit: the addresses are presented to you in hex, but the default number base for entry is denary, with hex entry requiring a special TI symbol ">" to indicate hex. Thus the above example could also have been written LI RO, 512.

- There are a number of facilities which are implemented, enabling you to look at the symbol table, change current address, enter DATA, and TEXT, and so on. Fairly versatile, I think, and moderately-well documented in the separate LBLA manual.

Where TI have really come up trumps (makes a change!) is that, unlike every other home and personal computer manufacturer I know, they have made available the addresses of routines (utilities) and details of their entry requirements to perform a whole host of very complex things, from reading/writing to the screen to evaluating trig and log functions on floating point numbers, which is VERY useful and most unusual. Other micro producers tend to keep very tight-lipped about what their routines do and where they reside in ROM. You usually have to wait until some third party provides you with such information in the form of an independantly-produced publication - more often than not a skimpy, over-priced book.

On top of this, a few extra subprograms have been provided for use in TI BASIC programs; these are INIT, LOAD, LINK, PEEK, PEEKV, FOKEV, and CHARPAT. I have listed these at the end with brief details of what they do. Some of them have already been covered in a previous review of Extended BASIC in Tidings. PEEK and LOAD work like PEEK and FOKE on CPU RAM while PEEKV and FOKEV do the same for VDP RAM. I'm still not too clear on the exact distinction between VDP and CPU RAM's, but it would appear that the CPU can only directly address its own RAM, and has to address VDP RAM as though it were one of the peripherals, using the CRU, and TI BASIC programs are stored in VDP RAM hence the slow operation. If anybody knows that just ain't so, I'm all ears.

You can SAVE and CLD TI BASIC programs and data to NM (response is very fast compared with cassettes - I don't have disks yet so I can't make a comparison there) with about 4K capacity so large files are out. This is probably the most powerful aspect of the module, in that you can store data on the module in small quantities and access it anywhere that you happen to have a 99 console and TV. You don't need cassettes and leads, you don't need disks. (To do anything 'serious', of course, you DO need those things!)

Back to English again. If you're contemplating making the quantum leap from BASIC to 9900 Assembler Language, then be prepared for a great deal of mental gymnastics. This subject is not really for the faint-hearted - but don't let that put you off. I would strongly recommend that BEFORE you buy you should try and get a copy of the dreaded

Assembler/Editor manual if at all possible (you may end up buying THAT module instead). Otherwise this module will be of little use to you. Its manual DOESN'T give you a course in machine code programming, it doesn't even provide you with a list of opcodes and mnemonics. I am sorely tempted to try and present an introduction to 9900 Assembler early next year (1983), but it would need someone to take over a section of Tidings to free me to do it (hint, hint). It currently takes about a month to prepare a BB article, and about two weeks to write it, which leaves me two weeks to catch my breath before plunging in again.

I did contrive a short machine code routine to write 'Hello Mother' to the screen a few times (resulting in 'Hello Motherrrrrrrrrrrrrrrrrrrrr...') just for fun, but while it functioned fine on my NTSC 4, it wouldn't run without crashing on the PAL 4 belonging to the Rowe family in Oxford. Cue deluge of error sheets for future MM manuals ?

The third option, RE-INITIALISE, allows you to clear the module's RAM of TI BASIC data or program(s) and optionally any assembler language programs as well. The odd thing is that, elsewhere, the manual says you cannot mix TI BASIC and assembler language together in the module's RAM. You apparently need the Expansion RAM (q.v.) to aid you in that endeavour. (Interesting how one module seems to require £375 of other modules and peripherals in order to even partly utilise some of its capabilities!).

All in all I suppose the module has something for everybody, with the greatest possibilities opening up for those prepared to invest the greatest amounts of folding stuff. Perhaps the best market for it is for those who want to extend their language experience without dropping BASIC; mixing BASIC and Assembler is a recognisable step on the road to becoming a fully-fledged machine code programmer and to a better understanding of the working of any computer.

Here are brief details of those extra CALLs:

INIT : initialises CPU memory for Assembly Language routines. Erases all programs and data from the MM. There are no parameters.

LCAD() : loads 'assembly language object files' - machine code programs - into CPU memory, or can act like FCKE on other machines. Parameters are either a string expression (object-filename) e.g., "DSK1:LEPC" or address(es) and value(s). For example, CALL LCAD(32000,1,C,7,6,"",16381,7,221) is the same as this series of FCKEs:

FCKE 32000,1 :: FCKE 32001,C :: FCKE 32002,7 :: FCKE 32003,6 :: FCKE 16381,7
:: FCKE 16382,221

- LINK() : passes control and optionally a list of parameters from TI BASIC to a machine code program. Requires a 'program-name' e.g. LINES or FLOTXY, which you must have previously entered in a 'REF/DEF' table on board the module, and an optional list of parameters - string or numeric variables or expressions. If a variable is passed and altered by the machine code its value in the calling program is also altered. Values resulting from expressions however cannot be passed back from machine code to calling program. Entire arrays can be passed by name with one command, which is very powerful. In addition, you can specify that any variable passing data should not be altered by the machine code program.
- PEEK() : reads the contents of specified address(es) in CPU memory into specified variable(s). For example, CALL PEEK(3000, A, B, C, "", 7196, D) is the same as:
A = PEEK(3000) :: B = PEEK(3001) :: C = PEEK(3002) :: D = PEEK(7196)
- PEEKV() : does the same for VDP memory.
- POKEV() : acts like PEEK on other machines for VDP memory; i.e., the opposite of PEEKV. It is similar to part of the multi-faceted LCAD subprogram. If you are unfamiliar with PEEKing and POKing, try this: memory on computers is organised like one gigantic street. Each memory location is like a house on that street: it has a unique address. Each house has a number of occupants: in this case, there is provision for 8 occupants at a time. Sometimes they are all away, sometimes they are all present. These occupants are a little odd: each one has an age, and never grows any older. The first one is 1, the second is 2, the third is 4, the fourth is 8, fifth 16, sixth 32, seventh 64 and the eighth is 128. Whenever we attempt to decide who is at home we count not heads, but ages, which lets us see exactly who's in and who's out. When we PEEK an address, we rip the roof off and tot up ages. Depending on who's in, we'll get a value of between 0 and 255 (1+2+4+8+16+32+64+128). Being precocious, we occasionally like to change who's there and who isn't, by POKing a fresh group of eight occupants to a given address. POK 16384, 128 replaces all at no. 16384 with one 128 year old, and boots everybody else out. How's that for a bedtime story ?
- CHARPAT(): returns the 16 digit hexstring currently defining the character(s) specified by ASCII code. The opposite of CALL CHAR.

POPULAR MICRO MAGAZINES : A PERSONAL VIEW

I currently buy about 10 micro mags a month, so I thought I'd pass on a few highly personal comments to you, dear reader, in case it helps you make a choice between two equally gaudy (and equally expensive) front covers. The list is not arranged in any particular order, and does not include any of the 'trade' rags, which I don't get.

Personal Computing Today Monthly 65p

My newsagent put me on to this one - he thinks I'm interested in computers - so I've been getting it regularly (no comments). It has had a few programs and letters from our members (notably Mike O'Regan, who, together with Stephen Shaw has been opening the hitherto closed doors of some of the more popular mags). It is mostly a software mag, with a few articles on programming technique and hints and tips. If you're into doing conversions of other folks' software, this is quite a good one, but beware of the usual copyright pitfalls. It hasn't been around to really establish a name for itself yet (beginning its career with no less than THREE different names in one issue!) but if the reference section is anything to go by, it still needs to pull its socks up. I noticed that the reference sheet on the TI stated that there were no languages other than TI BASIC available, so I wrote to them, but they haven't done anything about it. Maybe they're waiting for the 'horse's mouth' version, in which case TI why aren't you supporting your product ?

Popular Computing Weekly Weekly 35p

This cheeky publication scores with me because it has such a short lead time compared with the monthlies (about 2-3 weeks compared with 3-4 months) which gives it an edge when it comes to latest developments. At 35p a week it ain't cheap, but it recently underwent a face-lift and should now offer a little better value for money. It has the usual reviews and odd bits of software for some of the most popular micros and a very irritating habit of sticking in totally incomprehensible strip-cartoons. They just got rid of one by popular demand and promptly replaced it with another. You can't win.

Computer & Video Games Monthly 75p

This mag came out dedicated mostly to arcade games and how to play them well, but it came in for a lot of criticism for its habit of obscuring those listings it did publish by overprinting with colour pictures - not very clever. Stephen Shaw has had a

for my buying both of these. They have a very cheeky item called CHIF CHAT which sails so close to the wind that you'd expect them to be immersed in lawsuits as a result. The general tone of the mag is very informal, but not so loose that it offends. Apart from all that, they published the first letter I ever had published anywhere, which commands a certain loyalty!

Electronics & Computing Monthly Monthly 70p

This mag has oodles of technical information and projects, and as I stand in almost religious awe of anyone who can put two components together and pick up BBC2, this particular publication is the only electronics-biassed one I get. It has relegated its software to mostly ZX81 programs produced by youngsters, but it is interesting for me also because it covers topics outside the immediate computer field (like Flat Screen TVs, for example).

MicroComputer Printout Monthly 95p

Probably the dearest mag on the market, this one began life devoted almost exclusively to Commodore products (Pets, etc.) but has since become a very professional all-rounder. It has informative articles on database management (not the mini-computer kind) and on things like how a disk system works. Its software is invariably accompanied by very detailed articles, but it does have one flaw. There is this contributor called 'Tommy', and if I see him write about how useless a function 'FCS' is again, I'm going to take time out to correct his misapprehension.

Computing Today Monthly 75p

Back in the early days (i.e., my early days, circa 1978) I used to get CT, but stopped because it wasn't very good value for money. In the intervening years they have much improved its content, and I recently started taking it again. They take the trouble to try and standardise on listing formats, and to tell you which characters do what.

Computer Answers ?Bimonthly 75p

This is the mag which has supplanted C&VG, and if subsequent issues are all as packed with information as the first (realistically not a likely prospect) this will be a very good buy indeed, rivalling the extinct Computer Age in quality and content. Alas, it seems that the road to success these days is paved with a high adverts-to-editorial content with games software. I hope it survives.

There are other mags which I pick up from time to time out of interest. For example, I used to buy Which Computer? until I realised that they scarcely seemed to acknowledge the existence of micros under £3000. Microdecision, monthly at 80p, is another which I have bought - the November issue had an interview with Man-watcher Dr Desmond Morris (whose son has a 4A and to whom I have issued an invitation through the retailer to join our august company, but without response) - but it is largely a business, rather than hobbyist, mag. Which Micro & Software Review is also a business rag, but I have picked it up on odd occasions in order to have an article as reference. It's bimonthly, price 85p, but I seem to remember that it is going monthly shortly.

I have recently picked up two American mags: Popular Computing, monthly at £1.85, which had a fascinating article on high resolution graphics; and Byte, often quoted as the 'bible' of micros in the States - monthly, priced also £1.85. For that sum, though, Byte offers you a massive 530 pages (compared with about 250 from Personal Computer World) although I'm not sure of the advert-to-editorial ratio. It had a number of articles of interest to me - industrial robotics, speech synthesis as two examples, but the real treasure was a pot-pourri of items from one Sol Libes. His section is called Bytelines and is described as News and Speculation about Personal Computing. I am passing on some of his pieces elsewhere in Babbling Brooks. I just might consider getting this one regularly... His article is presented as 4 columns of about 4-5 words wide on each page for a number of pages. Critics of Mike O'Regan's format take note!

The only other publication I take regularly is Science Digest - a sort of American Tomorrow's World but much more sensible (and often more up-to-date too!). That's a monthly - at £1.50 it too ain't cheap - but it has a wealth of items covering a massive range of scientific subjects as well as articles with a distinctly artistic flavour (meaning of the Arts, not of Art. They ran a series of 'windows' into the social practices of other cultures, some of which were a little eye-opening. Did you know that in the States the practice of cooking and eating ^{the} human placenta is on the increase ? Yuk!).

I also read Scientific American (resident in the medical library at my hospital) and occasionally Nature, and World Medicine, but these last two are unlikely to be of any interest to you, dear reader!

So now you know where much of the material for Babbling Brooks comes from, and why my turnaround for letters is often so slow - I'm busy reading!!

Beginner's BASIC

In the last issue, we briefly looked at the Immediate Mode; LET; Variables - Numeric and String; PRINT and PRINT Separators - and the use of PRINT ::::: to scroll the screen contents; Line Numbers and Statements; LIST; RUN; and RES(EQUENCE).

Before continuing, I think it might be a good idea to depart from BASIC for a while, and look at the functions on the keyboard - like Editing - so that either we can clear up any blind spots which you may have had, or we can avoid any blind spots in the future as we progress to other functions.

Almost immediately we run into difficulties (at least, I do). There are at least three different versions of the Texas at large in TIHOME, which is why some of you have been experisncing diffioulities when it comes to running software from the library, or when trying to run routines presented in this Babble from me.

The three versions that I have come across are i) the early NTSC 99/4, which produces American TV signals and needs a specially-adapted monitor in order to work. It has some facilitiss which other 99s lack, and until recently I thought that these were of only minor importance, until I came across a program in the disk-based library which uses not CS1 but CS2 to both write and read data; ii) the later, slightly-modified, PAL 99/4, which uses a modulator to enable the owner to make use of his domestic TV; and iii) the latest 99/4A, which also has a modulator and can be used with domestic TVs.

The principle difference is of course the keyboard. 4A owners have the kind of keyboard which just about every machine should have (note that I say 'kind of' - the way that the unit is implemented on the 4A leaves a great deal to be desired!) a so-called 'proper' QWERTY keyboard. However, this new keyboard possesses keys and functions which are not available on the 99/4s, which makes life difficult for me as I don't possess a 4A! Not only do 4As possess extra keys, their range of valid variable names is also greater than on the 4s, and at the time of writing I have been told by Paul that it is possible, using the CONTROL key, to make use of 'single key' entry of reserved words, something like the kind of thing available on the Sinclair machines. More details should be available long before you get to read this!

There are other differences - there is an automatic repsat function on 4A keys which is lacking on the others, and I understand that the SHIFT key doesn't have a keycode as it does on the 4s, which mucks up one of the programs I submitted to the library!

In addition, there is the FUNCTION key, which has taken on many of the functions previously performed by Shift - including, to my utter bewilderment, the selection of 'normal' QWERTY characters like quotes and the question mark!

Editing

In TI BASIC, there are two occasions on which you can 'edit' - i.e., insert, delete, re-write - a line of instructions (bearing in mind that a line can often mean 3 or 4 screen lines!). The first is in the so-called IMMEDIATE mode (the one in which you write programs and enter them) and the second occurs during the execution of an INPUT instruction while a program is running (we'll deal with INPUT later in this article).

The simple editing functions allow you to move left or right over (without deleting) a line or lines, to erase the whole line (which could mean 4 screen lines), to delete characters one at a time, or to insert characters one at a time into an existing line. These functions are available in both Immediate and Run modes. Slightly different functions, like Shift (or Function) E and X can also be used in both modes: they have slightly different effects depending upon the situation in which they are used. For example, while executing an INPUT instruction or while typing instructions in Immediate mode, these two 'Up' and 'Down' keys can substitute for ENTER. If you have a program already entered, then typing an existing line number and pressing Up or Down will call that statement onto the screen, and it can be edited. (Typing 'edit' and the line number has the same effect). If you subsequently press Up or Down, then any numbered statement lying either above or below the existing one will be called into the display ready to be edited. To get out of this 'mode', you simply press Enter.

Here are the equivalent editing keys on both 4s and 4As:

<u>Key</u>	<u>99/4</u>	<u>99/4A</u>	<u>Effect</u>
CLEAR/BREAK	Shift C	FCTN 4	Ignore entry/Interrupt program
DELETE	Shift F	FCTN 1	Delete character lying under cursor
INSERT	Shift G	FCTN 2	Insert character lying under cursor
ERASE	Shift T	FCTN 3	Delete entire line
LEFT arrow	Shift S	FCTN S	Move to left over line
RIGHT arrow	Shift D	FCTN D	Move to right over line
UP arrow	Shift E	FCTN E	Enter and move to lower line no. if editing
DOWN arrow	Shift X	FCTN X	Enter and move to higher line no. if editing

On the 4s there is a standard overlay which contains details of the editing keys, while on 4As there is a strip which is inserted in the guides above the keyboard.

To try and demonstrate all of the main editing functions (except Enter and its alter egos), switch your machine on and select TI BASIC. As usual, if anything is to be entered at the keyboard it will appear 'indented' on these pages; you type in a line from this page, check it, and then press Enter (unless told otherwise). Because we are going to be using the editing functions, you will need a way of distinguishing between data to be typed in, and instructions about which editing keys to press. What I will do is to use a double oblique (//) to separate data from editing keys, and hope that no confusion arises.

```
FOR EXAMPLE //Shift T (FCTN 3)
```

That shows the Erase function.

```
MOTHER //use Shift S (FCTN S) until the cursor is as far left as it will go, then  
press Shift G (FCTN 2)
```

```
PRINT "HELLO //follow "HELLO with a space. The line should now read PRINT "HELLO  
MOTHER . Now press Shift D (FCTN D) until the cursor lies to the  
right of the R in MOTHER.
```

```
" : "HELLO SON" ::::: //now use Shift S (FCTN S) to move to the left until the  
cursor lies over the colon after ...SON" . Press Shift  
F (FCTN 1) until all the colons have been deleted.
```

The line should now look like: PRINT "HELLO MOTHER" : "HELLO SON"

That has shown the effects of INSERT, DELETE, LEFT, RIGHT, and ERASE. Now press Shift C (FCTN 4) and notice that the line moves up the screen, but is not executed - the 99 has ignored the instruction! You can fill the screen with text this way, none of which will be acted upon by the computer - should you feel so inclined.

Right, back to BASIC. So far, we've looked at ways of storing and retrieving data, and of getting the computer to do it automatically if we store the storing/retrieving instructions (i.e., write a program!). Remember that the PRINT instruction doesn't alter the contents of the variables that it places on the screen - it just copies them out.

There is a parallel command called DISPLAY, which is a limited form of PRINT; DISPLAY can only put information on the screen, while PRINT can be added to, allowing it to print information not only to the screen, but also to cassettes (filing), a printer, a disk system (filing), speech synthesiser, and so on.

The way that PRINT is implemented on the 99s makes the screen behave like a 'glass teletype' - that is, you can't type over information already placed on the screen, and the printing always begins at the bottom of the screen, scrolling up any current information, line by line, as the bottom line is filled. There are ways round this (equivalent to the PRINT AT command found on other machines), and we will look at them later in the series.

PRINT is not restricted to printing out numeric and string variables; it can print the solution to an equation, whether using numeric variables, or numbers themselves:

```
PRINT 2*3 + 4*5
```

or:-

```
A = 12.3
```

```
B = 32.1
```

```
PRINT A*B + A - B + 7
```

and so on. In the example program which was given last issue, the variables were assigned values (numbers or words were stored in them) actually in the program. Using that system, if you wanted to use different numbers or words, you'd need to stop the program and alter some of the lines. This is not very convenient, especially if you wanted to change the values in an equation and then calculate the result!

This is where INPUT comes in. INPUT tells the computer to stop and wait until ENTER, or UP or DOWN (discussed previously) is pressed (and as we shall see, sometimes that isn't always true!) before continuing with a program. When the computer stops to wait for an entry, it behaves a little like it does in the Immediate mode: the cursor appears, flashing on and off, you have around 4 screen lines in which to type your entry, and the editing keys allow you to alter what you type in, should you make a mistake (although Shift C (FCTN 4) functions in this case to interrupt the program and return to Immediate mode). When you use INPUT, you can specify which variable is to be assigned whatever you type in; if you used INPUT A\$, then once you'd typed your entry and pressed Enter (or UP or Down), the computer would have stored what you'd typed in A\$, and then gone on to the next instruction to be carried out. If you typed nothing, then a 'null string'

would have been 'stored' in $A\$_$ - which means that $A\$_$ would contain nothing at all - a good way of 'clearing' $A\$_$ or wiping it clean, or emptying it. There are one or two things to watch out for here: if you use any quotes (") in your typing, you must make sure that they match up - typing "BERT would cause an error, because you need another quote - "BERT" before the computer will accept what you have typed. This is all part of something called INPUT VALIDATION - some of which can be carried out by the computer automatically, as we shall see, and some of which has to be done by you in the programs that you write. Also, if you use a comma (,) in your entry, be very careful to begin and end the WHOLE entry with quotes. The comma has a special meaning during INPUT; it is possible to enter information to be assigned to more than one variable in an INPUT, and the comma acts as an instruction to the computer to store anything lying before a comma into the first variable, and to store anything lying after it (and before any other commas) in a second variable. For example, INPUT $A\$_,B\$_$ will require you to separate your entries into two parts, separated by a comma, thus: BERT,WEEDON , where BERT will be stored in $A\$_$, and WEEDON will be stored in $B\$_$. This form of INPUT can save a little space as it replaces INPUT $A\$_$ as one instruction and INPUT $B\$_$ as another.

So far we've looked at entry of strings (words); when it comes to numbers, the rules are a little different. If the computer came across the instruction INPUT A in its program, it would stop, as usual, and wait for you to make an entry and press Enter. However, if you make no entry, but just press Enter, the computer will come back with an error message, and will present the INPUT again. Where entering numbers is concerned, the computer needs something to be typed (and it is fussy over just what you can type) before it will carry on. For entry of numbers, there is the same restriction over the use of commas as with entering words, but there is the added restriction that only certain keys may be pressed, and in a certain order. Those keys are: the '.', +, -, the digits 0 to 9, and the E. The E stands for 'times 10 to the power of' and is explained in the Maths article in V2.6 Tidings. Valid entries are, for example:

15 -20.87 +.6 -.004 -0.45 1.23 1. 1E6 -2E3 4E17 4E-17 -5E-26 1.23E24

and so on. Unacceptable formats would be:

-1-1 ..23 1.2. E4.5 EE-.9 and so on. The manual gives full details.

On many machines you can enter numeric variables and/or expressions (equations) and the system will evaluate them and assign the results to whatever variables are specified in the INPUT. Not so the 99s, which are very restrictive about what they will and will not accept.

INPUT actually is a very powerful function, and it has other properties which we will

look at in more detail. At a simple level, then, let us write a short routine to work out the reciprocal of a given number (that means the result of dividing 1 by that given number - see the Maths articles!). What will we need? Well, obviously an INPUT statement first, to assign the entry to a variable. Then we'd need to calculate the reciprocal and print it out - as we can get PRINT to do both things for us, we'll use just that statement.

```
1 INPUT N
2 PRINT 1 / N
```

RUN

When the cursor appears, type 1000 and press Enter. The computer will respond with .001 and DONE. It can get a bit boring typing RUN every time, so let us introduce a useful (and much-maligned) command: GO TO (or GOTO). This command tells the computer to go to a particular line number and continue processing from there; to save us having to type RUN every time, we could have the program loop round to statement number 1 each time it has printed the answer. Add another statement, then:

```
3 GOTO 1
```

This will keep the program running for ever (unless we make a spectacular mistake!) until Shift C (FCTN 4) is pressed (BREAK).

Now if you are the curious type, you may have wondered why a question mark appears when the INPUT is executed. This is called a 'prompt' and is provided free of charge by the computer (helpful, isn't it?) to remind you that you are about to enter some data.

The 'prompt' however, can be very helpful indeed, especially if you can expand it yourself to give more information. Surprise, surprise, you can! Retype statement 1:

```
1 INPUT "ENTER A NUMBER" : N
```

Note that a colon must be present between your prompt and the variable to which the entry will be assigned. In addition, the prompt can be another variable, but it must be in string format, but we will cover that later in the series.

RUN

This time the computer prompts you in more detail; you may find that the cursor lies a little too close to the prompt for comfort - you might need to enter a space in the

AB & C Computers

MAIL
ORDER
SPECIALISTS

for
TI-99'ers

GOOD TI-DINGS FOR XMAS AND NEW YEAR TO ALL READERS

EXCLUSIVE
SUPER SAVERS
FOR TI-HOME

REMINDE YOUR
FRIENDS ABOUT
THE £50 HANDBACK
ON TI 99/4A CONSOLE
(OFFER CLOSES 31 JAN
83)

BIG SAVINGS ON
PERIPHERAL BOX OPTIONS

COMPARE
OUR
PRICES!



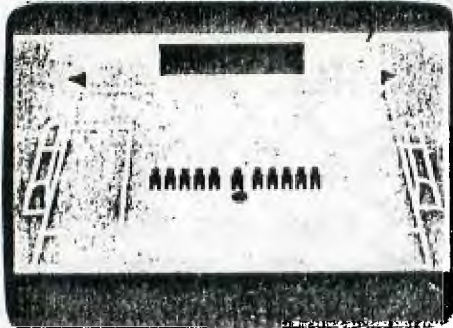
A. B. & C. COMPUTERS
11 BROCKSTONE ROAD
ST. AUSTELL, CORNWALL
Telephone St. Austell 64463

AB & C Computers

MAIL
ORDER
SPECIALISTS

for
TI-99'ers

TEXAS ORIGINATED SOFTWARE MODULES- TI99/4 and TI99/4A

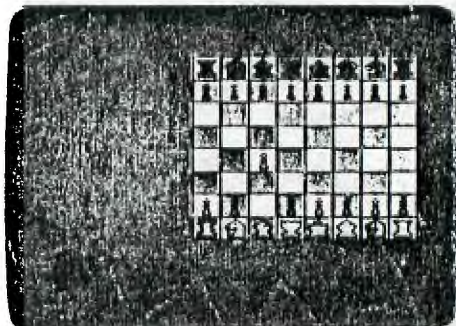


TI INVADERS	£16.95(A)
A-MAZE-ING	£20.95(A)
TOMBSTONE CITY	£20.95(A)
ATTACK	£20.95(A)
YAHTZEE	£20.95(A)
MUNCHMAN	£24.95(A)
NUMBER MAGIC	£12.75(A)
HANGMAN	£20.95(A)
EARLY READING	£24.95(A)
HOME BUDGET	£20.95(B)
SPEECH EDITOR	£16.95(A)
EXTENDED BASIC	£69.00(C)
EDITOR/ASSEMBLER*	£75.00(C)
TEXT FORMATTER*	£49.95(B)
ADVENTURE & PIRATE%	£33.75(B)



Every item of software
comes complete
with full operational instructions.

MIND CHALLENGES	£12.75(A)
SOCCOR	£20.95(A)
ZERO ZAP	£20.95(A)
HUSTLE	£20.95(A)
BLACKJACK/POKER	£33.75(B)
BEGINNING GRAMMAR	£16.95(A)
ADD/SUBTRACT 1	£24.95(A)
ADD/SUBTRACT 2	£24.95(A)
PERSONAL REPORTS	£42.25(B)
TERMINAL EMULATOR 2	£42.25(A)
MINI MEMORY %	£75.00(C)



VIDEO GAMES 1	£20.95(A)
CONNECT FOUR	£20.95(A)
WUMPUS	£20.95(A)
BLASTO	£20.95(A)
CARS WARS	£24.95(A)
CHESS	£33.75(A)
PRE-SCHOOL FUN	£14.75(A)
MULTIPLICATION	£24.95(A)
MUSIC MAKER	£24.95(A)
PERSONAL RECORDS	£42.25(B)
STATISTICS	£42.25(B)
TI-LOGO	£69.00(C)

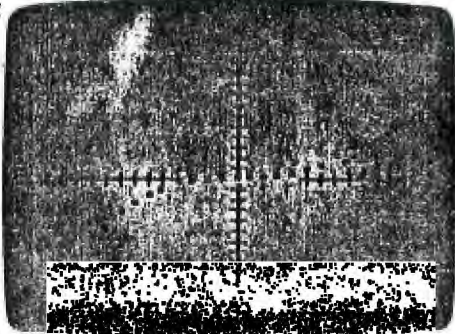
*REQUIRES DISCS

%MODULE & CASSETTE

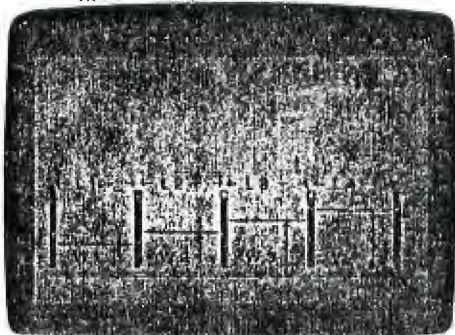
VAT INCLUDED IN ALL PRICES

TEXAS ORIGINATED SOFTWARE CASSETTES - TI99/4 and TI99/4A

ADVENTURE LAND%	£20.95(A)	VOODOO CASTLE%	£20.95(A)
THE COUNT%	£20.95(A)	PYRAMID OF DOOM%	£20.95(A)
GHOST TOWN%	£20.95(A)	OLDIES/GOODIES 2	£ 8.25(A)
STRANGE ODYSSEY%	£20.95(A)		
GOLDEN VOYAGE%	£20.95(A)		
MISSION IMPOSSIBLE%	£20.95(A)		
MYSTERY FUN HOUSE%	£20.95(A)		
SAVAGE ISLAND 1+2%	£24.95(B)		
%REQUIRE ADVENT'/PIRATE MOD'			
PROGRAMMING AIDS 1	£ 8.25(B)		
TEACHYOURSELF BASIC	£ 8.25(B)		
---EXTENDED BASIC	£ 8.25(B)		
PERSONAL FINANCE	£ 8.25(A)		
MARKET SIMULATION	£ 8.25(A)		
OLDIES/GOODIES 1	£ 8.25(A)		



TEXAS ORIGINATED SOFTWARE DISKETTES - TI99/4 and TI99/4A



INVENTORY	£49.95(B)
PROGRAMMING AIDS 2	£16.95(A)
PROGRAMMING AIDS 3	£16.95(A)
TEXT FORMATTER	£49.95(B)
INVOICING	£49.95(B)
PASCAL EDITOR	£49.95(B)
PASCAL LINKER	£67.50(B)
PASCAL COMPILER	£79.95(B)
MATHS LIBRARY	£24.95(A)
ELECT/ENG LIBRARY	£24.95(A)
STRUCT/ENG LIBRARY	£24.95(A)
MAILING LIST	£49.95(B)

OTHER SOURCES CASSETTE SOFTWARE (BASIC) - TI99/4 and TI99/4A

GAMES TAPE 1	£ 1.75(A)	GAMES TAPE 2	£ 3.95(A)
GAMES TAPE 3	£ 4.95(A)	O's & X's/MASTERMIND	£ 5.95(A)
HAUNTED HOUSE	£ 5.95(A)	MAZE/TOWERS OF HANOI	£ 5.95(A)
FRUIT MACHINE	£ 5.95(A)	ROULETTE/CRAPS	£ 4.95(A)
PLANET LANDER/LIFE	£ 4.95(A)	FOUR IN A ROW	£ 5.95(A)
WUMPU	£ 4.95(A)	TEN PIN BOWLING	£ 7.95(A)
BLACKJACK	£ 5.95(A)	NIM & O'S + X'S	£ 4.95(A)

VAT INCLUDED IN ALL PRICES

TEXAS TI99/4A HARDWARE (**£50 HANDBACK TI99/4A**)

TI99/4A COMPUTER	£198.95(D)	SPEECH SYNTHESISER	£46.58(E)
PERIPHERAL BOX	£137.75(F)	DISC CONTROLLER	£127.35(E)
DISC DRIVE (INT)	£238.95(F)	DISC DRIVE (EXT)	£317.00(F)
RS232 CARD	£103.49(E)	32K RAM CARD	£119.39(E)
PASCAL P CODE CARD	£159.22(E)	MATRIX PRINTER	£363.78(D)
CASSETTE CABLE	£9.15(A)	JOYSTICKS PAIR	£19.49(C)

PERIPHERAL EXPANSION BOX SUPER VALUE COMBINATIONS *****

OPTION 1: EXPANSION BOX, 32K RAM	£249.00(D)
OPTION 2: EXPANSION BOX, RS232 CARD.....	£235.00(D)
OPTION 3: EXPANSION BOX, 32K RAM, RS232 CARD.....	£345.00(D)
OPTION 4: EXPANSION BOX, RS232 CARD, 32K RAM CARD MATRIX PRINTER.....	£695.00(G)
OPTION 5: EXPANSION BOX, RS232 INTERFACE CARD.... MATRIX PRINTER.....	£580.00(G)
OPTION 6: EXPANSION BOX, 32K RAM, LOGO MODULE....	£315.00(D)
OPTION 7: EXPANSION BOX, 32K RAM, DISC CONTROLLER INTERNAL DISC DRIVE.....	£599.00(G)
OPTION 8: EXPANSION BOX, 32K RAM, DISC CONTROLLER INTERNAL DISC, PASCAL P CODE CARD.....	£745.00(G)
OPTION 9: EXPANSION BOX, 32K RAM, DISC CONTROLLER INTERNAL DISC, EDITOR/ASSEMBLER MODULE.	£675.00(G)

ALL PRICES INCLUDE VAT

****CERTAIN ITEMS ARE IN SHORT SUPPLY AND ARE OFFERED SUBJECT**
**TO AVAILABILITY. ORDERS WITH 20% DEPOSIT ARE TAKEN FOR **
THESE ITEMS. IN ORDER TO SECURE THE SPECIAL TI HOME PRICE
**OFFERED TO YOU, PLEASE MENTION TI HOME WHEN YOU ORDER. **
WE HOLD A WIDE RANGE OF BOOKS AND ACCESSORIES, SAE FOR LISTS.**

*******TERMS CHEQUE/CASH/ACCESS PLEASE WITH ORDER*******

CARRIAGE CODE:

(A)= 60p. (B)= 80p. (C)= £1.20. (D)= £5.00. (E)= £1.40.
(F)= £2.50. (G)= £6.50. PLEASE ADD CARRIAGE TO ORDER VALUE.

CHRISTMAS ORDERS
ARE BEING TAKEN.
£50 HANDBACK TI99
CLOSES 31 JAN 83.

A. B. & C. COMPUTERS,
11 BROCKSTONE ROAD,
ST AUSTELL, CORNWALL.
Tel: St Austell (0726) 84483
Tel: Newquay (06373) 6886
Personal callers welcome

Terms Cheque
P.O. with order. Carriage Computers
Printers £5.00 Post and Packing other
Items. All hardware checked by service
engineer.
24 HOUR
ANSAPHONE FOR
ACCESS ORDERS

prompt string to space things out a little. Try entering a few numbers to see how the computer reacts; throw in a few incorrect entries to see what happens.

As an example of how 'friendly' the computer can be, provided the programmer makes it so, try this short program:

```
1 INPUT "PLEASE ENTER YOUR FIRST NAME " : N$
2 PRINT "THANKYOU "; N$
3 INPUT "NOW PLEASE ENTER YOUR AGE IN YEARS " : Y
4 PRINT "THANKYOU," : N$;" NOW I KNOW WHO YOU ARE AND HOW OLD YOU ARE"
```

When you run this short routine, notice how stupid the computer is; it can't tell if you enter somebody else's name, or if you lie about your age. This is a small example of how computers are only as intelligent (and as capable of taking over the world) as their programmer is!

The above routine doesn't make any use of Y, the age variable. Use your imagination to write some more lines to make the computer respond with "YOU ARE";Y;"YEARS OLD", or to print "HELLO, 305 YEAR OLD SPLDGE" if you enter SPLDGE and 305 as your INPUTs.

Where would you place a GOTO instruction to make the program continually loop back to line 1? Try it and see if you have understood the instructions properly. What happens if you alter line 3 to make Y into Y\$, and then alter every other occurrence of Y into Y\$? Remember the difference between the way in which numbers and words are printed out?

The screen must be fairly cluttered by now, and it would be nice to be able to wipe it clean every so often. You'd think that there would be a 'wipe' command, and you'd be right. On the 99s it is CALL CLEAR.

```
CALL CLEAR
```

You can also incorporate it in programs - it needs a line to itself - and in this case we'd really like to insert it before line 1. If you remember, RES can be used to make room for more lines, so use RES to allow you to enter 1 CALL CLEAR - I leave it up to you to decide how that should be done. The manual and V2.5 of Tidings show you how.

Next issue we'll look at further INPUT VALIDATION using an IF command.

A Re-Introduction To Mathematics

Continuing the logarithmic brain-storm...

And beginning with an apology. Specifically designed to confuse you, the line:

```
" PRINT (5 ^ 2) ^ 3
```

is the same as $5^2 * 5^2 * 5^2$. " was included by accident. It formed part of an early explanation which was unfortunately not crossed out in the original, and I failed to pick it up when proof-reading prior to submitting to Tidings. Oops!

Anyway, on with the strain. I threatened last time to involve you in the intricacies of NAPIERIAN logarithms, so here we are. Funnily enough, these logarithms may just be what you need to underline some of the points about BRIGGSIAN logs which we have already looked at. Instead of being based on something simple like 10 (as Briggsian logs are), Napierian logs (\log_e or \ln normally, but on the 99s is written as LOG) are based on the number 2.718281828 for reasons best known to John Napier or someone who knows more about him than I do. Napierian logs are the ones most commonly found on calculators which boast logs, and on most computers, which seems a little daft, as we shall see, as the conversion from one to another doesn't take a great deal of extra work. It is very likely that Napierian logs can be derived from a peculiar function in maths known as a series; for example, one series can be used to find the SINE of an angle in radians (!) and surprise, surprise, guess what: most computers which have a SINE function use the dreaded radians. A future episode in this thrilling saga will be devoted to reminding you about SINES and the like, and will delve a little into the use of series to generate such functions. The mysterious radians (and their relationship to degrees, plus another measurement: gradians) will be explained in that future article!

Back to Napier. To see the difference between Napierian and Briggsian logs, look at the following table, where the logs for a range of numbers are given:

<u>Number</u>	<u>Briggs</u>	<u>Napier</u>
.0001	-4	-9.210340372
.001	-3	-6.907755279
.01	-2	-4.605170186
.1	-1	-2.302585093

<u>Number</u>	<u>Briggs</u>	<u>Napier</u>
1	0	0
10	1	2.302585093
100	2	4.605170186
1000	3	6.907755279
10000	4	9.210340372

If you cast your minds back (or better still, look at a copy of) issue 5 explained that the anti-log (in other words, the table in which you find out what number is represented by a particular logarithm) of a log could be found by raising the 'base' to the power of the log. Thus the anti-log, base 10, of 1.5 is $10^{1.5}$. In the case of Napierian logs, the base is 2.718281828, so that the anti-log, base 2.718281828, of say, 1.5, is $2.718281828^{1.5}$. As the base is such an unwieldy number to keep writing down, and as mathematicians are essentially a lazy bunch (as are scientists generally), as standard the letter 'e' is used to represent 2.7182818...yawn...(it might help you to understand that, if you think of the BASIC statement: LET E = 2.718281828; on the other hand it just might help you to understand the BASIC statement instead!) - hence, \log_e for Napierian logs and \log_{10} or just log, for Briggsian (except on the 99s of course. Here LOG means \log_e). This laziness helps when writing anti-logs: anti-log base e of 1.5 is $e^{1.5}$, and that is provided on the 99s in the shape of the EXP function (i.e., EXP(1.5)). To remind you, anti-logs base 10 are written on the 99s as 10^{\wedge} number; e.g., $10^{\wedge}1.5$.

I hope that you can see why I chose to deal with Briggsian logs first; explaining that $100 * 100$ is 10000 using Briggsian logs is so much easier than using Napierian: i.e., using Briggsian, $= 2 + 2 = 4$, and $10^4 = 10000$, whereas with Napierian, $4.605170186 + 4.605170186 = 9.210340372$, and $2.718281828^{9.210340372}$ is 10000 puff, puff. (Look at the earlier table for comparison). Of course, if you choose your numbers carefully, even Napierian can look easy:

<u>Number</u>	<u>Briggs</u>	<u>Napier</u>
.0183156389	-1.737177928	-4
.0497870684	-1.302883446	-3
.1353352832	-0.8685889638	-2
.3678794412	-0.4342944819	-1
1	0	0
2.718281828	0.4342944819	1
7.389056099	0.8685889638	2
20.0853692	1.302883446	3
54.59815003	1.737177928	4

If you are confused, follow the explanations through slowly and carefully. You will undoubtedly need issues 4 and 5, as they set the ball rolling in this direction and unless you have those to refer back to, you won't make much headway.

Going back to those two tables, if you look at the Briggs column in the first table, and the Napier column in the second, you will see that they are all integers, running from -4 to +4. What is obvious in the first table is that as the Briggs value increases by 1 each time, the number in the Number column increases by a factor of 10 (the base). I.e., adding one to the logarithm is equivalent to multiplying by 10. For example, $-4 + 1 = -3$, and $.0001 * 10 = .001$; or $1 + 1 = 2$, and $10 * 10 = 100$. What is not so obvious in the second table is that as the Napier column value increases by 1, so the number in the Number column increases by a factor of 2.718281828: for example, as the Napier value rises from 2 to 3, so 7.389056099 is multiplied by 2.718281828 to become 20.08553692 (in each example, the multiplying factor is the base).

It might be worth underlining one or two points here:

1. $X * X = X^2$

$X * X * X = X^3$

2. $1 / X = X^{-1}$

$1 / X^2 = X^{-2}$

3. Any value raised to the power of 1 stays the same value: $X^1 = X$

4. Any value raised to the power 0 is 1: $X^0 = 1$

5. Any value, V, raised to the power P, thus: V^P , is the equivalent of the anti-logarithm, base V, of P. Thus if the base = 15, then the anti-logarithm, base 15, of 5, = 15^5 . If the base = 7.22, then anti-log base 7.22 of 3.5 = $7.22^{3.5}$.

6. Whether you use Briggsian or Napierian logs, the results are always the same, give or take a slight difference due to internal calculation errors on the machine used. For example, the simple equation: $2.5 * 3.5 = 8.75$. Using Briggs, this is $\log(2.5) + \log(3.5)$, which is then anti-logged by raising 10 to the power of that sum - i.e., $0.3979400087 + 0.5440680444 = 0.942008053$. Then, $10^{0.942008053}$ is 8.75. Now with Napierian logs: $0.9162907319 + 1.252762968 = 2.1690537$, and $e^{2.1690537}$ (remember 'e' ?) is also 8.75. The internal errors mentioned above are caused by circumstances like this: divide 1 by 3. The answer will be 0.3333333... or 0.3333 recurring (which means that you could go on and on with the 3s). If you multiplied 0.33333 recurring back up to 1 by multiplying by 3, the answer would be 0.999999... or 0.9999 recurring, and not 1. In binary, the numbering system which all computers use internally, there are similar areas

in which errors occur. To try and overcome this fault in the maths system, there is a custom that 'rounding up' is performed on fractional numbers like 0.99999 recurring: the convention is that all fractions ending in 5 or more are rounded up to the next unit, while 4 and below are rounded down to the previous unit. In english, then:

5.4 rounds down to become 5; 0.999 rounds up to become 1. You work from the far right to the left: thus 1.23456789 successively becomes 1.2345679 as the 9 is rounded up, and the 8 then becomes a 9, 1.234568 as the 9 is again rounded up, and the 7 becomes an 8, 1.23457 as the 8 is rounded up and the 6 becomes a 7, 1.2346 as the 7 is rounded up, and the 5 becomes a 6, 1.235 as the 6 is rounded up and the 4 becomes a 5, 1.24 as the 5 is rounded up and the 3 becomes a 4, then 1.2, as the 4 is rounded DOWN, not affecting the 2, and eventually 1, as the 2 is rounded down, not affecting the 1. So eventually 1.23456789 rounds up to 1, if you were rounding to just the integer. Had the number been something like 1.467 then the result would have been 2. Can you see why ?

Back to making the points again:

7. You can only have logs of positive, non-zero, numbers, and you can also only have bases which are positive and non-zero. You cannot have a base of 1, either. You can have bases which lie either side of 1: 0.5 or 1.2, but never exactly 1. Here's why.

If you look closely (again) at the tables, you may begin to see a relationship between the Briggsian log of a number and the Napierian log of that same number. There are two ways of describing the relationship: either Briggsian logs are always Napierian logs divided by 2.302585093, or Napierian logs are always Briggsian logs divided by 0.43429-44819. (The first number is the reciprocal of the second, in case you were wondering!)

You might also be interested to learn that 2.302585093 happens to be $\text{LOG}(10)$ - i.e., the Napierian log of the base of Briggsian logs. There is therefore an equation which will cover the conversion of Napierian logs to Briggsian logs; it is:

$$\text{Briggsian log}(X) = \text{Napierian log}(X) / \text{Napierian log}(10)$$

If you have issue 5 to hand, examine the definition of BLOG, the specially-created Briggian log function. (Remember that Napierian logs are called LOG on the 99s). This equation is very similar to the 'general' equation which can be used to give a logarithm to any base (except those listed earlier: negative values, and 0 or 1).

$$\text{Anybase log}(X) = \text{Napierian log}(X) / \text{Napierian log}(\text{base})$$

The important thing to look at here is the divisor: Napierian $\log(\text{base})$. I have already shown (I hope) that you cannot have logarithms of 0 or negative numbers. Note that you always get 0 if you take the log, to any base, of 1. If you specify a base of 1, therefore, the above part of the equation, Napierian $\log(\text{base})$, will evaluate to 0, which means that you will end up dividing by 0.

Technically, the result of dividing any number by 0 is regarded as being 'undefined' - in other words, nobody really knows the answer for certain. It is likely to be infinity; but as that is an abstract value and cannot be proved to exist using standard maths, it is something which we will not concern ourselves with further.

Switch on your computer, and select TI BASIC - where we will experiment a little. To remind you, where something is to be keyed into the computer, I will indent the line on the page. Having checked that you have copied the line OK, press ENTER to get the computer to respond. Begin by putting in our BLOG definition from last issue; if you recall, we put it in as a DEF statement using a line number so that a form of 'program' was produced. This 'program' can then be RUN, and the BLOG function is then temporarily incorporated into the list of functions available on the computer. Note that if you NEW, BYE, or reset the computer to the title page by whatever means, you will wipe out the BLOG function.

```
100 DEF BLOG(X) = LOG(X) / LOG(10)
```

```
RUN
```

The new function should now be ready to use in the so-called IMMEDIATE mode (so-called because the computer will act immediately on whatever you give it to do, instead of storing instructions for later execution). Check with:

```
PRINT BLOG(1000)
```

The answer should be 3. If you get anything else, you must have made a mistake when keying in the definition. If you are ABSOLUTELY sure you didn't make a mistake, then panic because there is something drastically wrong with your machine !

Now let's give the computer the equivalent anti-log function. For LOG it is EXP on the 99s; we'll call ours EXB for BLOG anti-log.

```
110 DEF EXB(X) = 10^X
```

```
RUN
```


Check that it works OK with:

```
PRINT BLOG(1000), EXB(3)
```

You should get 3 and 1000. There is another way that the EXB function could have been written: $EKP(X * LOG(10))$. You might try that and see if it is true.

Now let us try and include a logarithm function which will permit us to obtain the logarithm to any base of a number, and its associated anti-logarithm. We have a slight problem in that TI BASIC will not let you specify two parameters in the DEF statement, but we can get round that with a little thought. Add the following two lines to our 'program':

```
120 DEF ALOG(X) = LOG(X) / LOG(B)
```

```
130 DEF EXA(X) = B ^ X
```

```
RUN
```

In 120, the definition will expect a variable, B, to be present, which will already have been assigned the base of the logarithm which we want. Therefore,

```
B = 10
```

```
PRINT ALOG(100)
```

Will first assign 10 to B (making the base 10, and equal to our BLOG function therefore) so that the subsequent PRINT instruction will give 2. Try:

```
B = 20
```

```
PRINT ALOG(400)
```

What is the answer, and why ?

I have also given the anti-log, EXA, so that once B has been assigned a base, you can experiment with ALOG and EXA to obtain logs and anti-logs to any base. Try using a range of bases (don't forget to assign the value to B before using ALOG or EXA), and also try using the bases I have said cannot be used: i.e., 0, 1, and a negative number. With a base of 1, you should get WARNING NUMBER TOO BIG whenever you use ALOG, but because

of the way in which EXA has been implemented, you won't get such an error message when using a base of 1 there, as 1 raised to any power is always 1. Had the function been implemented as $EXP(I * LOG(B))$ then again no error message would be produced. Can you see why ?

If you are having problems with understanding how the DEF function has been used to create these extra functions, a future article in the Beginners' BASIC series should help to unravel some of the mystery, unless you find that the explanation given by TI in the handbook is sufficient.

I have only been able to gloss over most of the more intricate facets of logarithms - like what are their practical applications - because it is very unlikely that you will find yourselves using them on a daily basis; however, if you do have any queries, please don't hesitate to write in to the address given elsewhere, and I will do my best to confuse you still further!

Next issue I will deal with a technique for finding the base of a logarithm, given that no-one writes in with insuperable difficulties over issue 5's attempts at lucidity.

THREE - DIMENSIONAL NOUGHTS & CROSSES

The artificially-intelligent story on artificial intelligence continues...

The latter part of the last episode was written in some haste, in order that the explanation should not occupy too many pages. Having dumped the cursory explanation of the existence of Winning Lines on you, in this issue I will attempt to expand on the subject. I'm trying to make sure that the whole series DOESN'T last 5 years - for this section is thrown open to you, dear reader (both of you), for you to present an explanation of one of your programs, or of the theory behind a game or serious application. After this short series on Three-dimensional Noughts & Crosses (3DOXO) is finished (hopefully next issue, when the final points will be explained, and a full listing given, together with outline explanation) I have no immediate plans to follow it up with Othello or whatever, although something may appear between now and Feb 83. The space will be yours, if you choose to fill it.

More On Winning Lines And The Linesum Array

The Winning Line Coordinate Sequence Array (WLCS array) is one which holds the 76 coordinate sequences (on a scale 1 - 64, for each of the four positions which comprise a 'winning line'), each coordinate being expressed as a PAIR of digits (01 - 64), so that each winning line, in coordinate terms, is 8 digits (4 pairs) long. The sequences of 4 are 'ordered' in two directions:

- A. Laterally, in that the first digit pairs in any group of 4 are those with the highest Linkage Values (see last two issues) - except where all the linkage values are the same - for example, look at Table 4 (i), where the first element of the array is 23420461. Decoding this in pairs of digits gives 23, 42, 04, and 61. If you look at p31 of Tidings V2.5, there are two corresponding 'maps' of the playing boards. The lower map shows the positions, the upper shows the linkage value for each position. Position 23 (underneath 19) has a linkage value of 7, as does 42, 04, and 61. This winning line sequence is one of the four 7777 types which run diagonally across and through the boards. As no one position has a higher linkage value than its companions in this type, the position sequence (23, 42, 04, 61) is ordered in a way LEAST likely to be spotted by a human player looking for patterns in the computer's choice of moves. If the sequence was 04, 23, 42, 61, the human opponent might readily notice the pattern as it built up. (The psychology of this game is something else!)

The same rule of thumb is applied to element 76 of Table 4 (i): 18191720, (or 17, 18, 19, 20 if you hadn't spotted the sequence) where the linkage value sequence is 4444.

B. Longitudinally, in that there are three main types of laterally-ordered linkage value sequences:- 7777, 7744, and 4444, and the longitudinal ordering places the four which are 7777 first in the array. The next 48 (which are 7744) follow them in no particular order, and the 24 which are type 4444 bring up the rear. Tidings V2.4 gives the formula $3*N^2 + 6*N + 4$ as calculating the total number of possible winning lines for a three-dimensional $N*N*N$ board. In this case $N = 4$, giving $48 + 24 + 4$; there are $3*N^2$ of the 7744 type, $6*N$ of the 4444, and four of the 7777 type. For $N = 5$ the pattern changes, becoming far more complex. We need not concern ourselves with that here.

The more pattern-conscious among you might have spotted the fact that each of the WLCSs can be generated (in a pre-scrambled sequence) from an initial value and an increment. It is faster though to have the sequence all ready for decoding, rather than have to calculate it each time. Those with greater experience of programming might also recognise that the coordinate sequences can be stored in a more compressed manner using ASCII codes (0 - 255, whereas we only need a range of 1-76 maximum) and they WOULD have been, only TI BASIC doesn't have the facility for POKEing the requisite values into place - but that's another story! I'm working on a tool using the Minimemory to correct that waste of space, altering the program so that it can run in less RAM and still use TI BASIC and no modules or memory expansion.

The Linesum Array is an array which parallels the WLCS array - the element numbers correspond, and the Linesum array holds the totals of the move values contained in a winning line sequence, generated during a game. That mouthful just means that element 1 of the Linesum array holds the sum of the move values for positions 04, 23, 42, and 61 (Tidings V2.5 gave details about move values - a value of 1 for the computer's moves, and of 4 for the opponent's moves). If the opponent had won a game by occupying the positions 29, 30, 31, and 32 (i.e., the scrambled winning line sequence recorded in element 74 of the WLCS array - see Table 4 (i)), then element 74 of the Linesum array would contain the value 16 - i.e., four opponent's moves = $4 \times 4 = 16$. Unfortunately, the principle behind this is very simple, but extremely complex, and difficult to fully explain in only a few words. We haven't yet come to the most difficult part - the Update Pointer Array - but we will do (you have to take the rough with the er.. rough).

So, to try and clarify things: what happens when a human player makes a move? Well, first the move has to be 'validated' - the computer searches the boards for the position, and then checks to see if it is empty. If it is, the opponent's move is accepted.

It then has to search through all the possible winning line sequences to find those which involve that position - there will be either 4 or 7 of them to find (see Tidings V2.5) and then add the move value (in this case 4 for the opponent) to the corresponding elements of the Linesum array.

For example, if I move into position row 1, column 1, level 1, (i.e., on scale 1 - 64, is 01), then the computer will add 4 to the contents of Linesum array elements 4,(22, 43, 01, 64), 29,(01, 16, 06, 11), 35,(01, 52, 18, 35), 37,(01, 61, 21, 41), 43,(01, 04, 02, 03), 48,(01, 13, 05, 09) and 51,(01, 49, 17, 33) i.e., 7 elements. (Look at elements 4, 29, 35, 37, 43, 48, and 51 in Table 4 (i).)

The computer then has to decide how to respond. It begins by selecting values from the strategy sequence (see Tidings V2.5) one at a time, and then running through all 76 elements of the Linesum array, comparing the contents of each element with the strategy value, until perhaps it finds a match. (The strategy sequence is arranged so that if no match is found for any strategy value, the game is automatically drawn, as all the winning lines have been blocked without a win occurring!). Let's say, for the sake of argument, that it found no match at all for strategy values 16 and 3, but did find a match for strategy value 12 with element 1 of the Linesum array. Now, a value of 12 means that the winning line of four positions contains 3 moves by me, and the fourth position is empty (for the reasoning behind this you will need to read V2.5 Tidings). This is a dangerous position for the computer, as it means that unless it blocks me, I can move into the empty position on my next move and win. The computer obviously needs to occupy that empty position. But where is it? The WLCS array holds the answer. As the match occurred with element 1 of the Linesum array, then we need to look in element 1 of the WLCS array, where the four positions corresponding to that line are stored. The sequence held in WLCS is 23, 42, 04, and 61 (see Table 4 (i)). All the computer has to do is to decide which is the empty position. It does this by decoding the 23420461 into pairs of digits, and then using these to address the corresponding screen locations, until it finds the empty position, into which it then moves, thus successfully blocking my attempted win.

The ordered sequence of positions in each element of the WLCS array performs another function: it dictates which positions the computer looks at first, and thus which position is eventually moved into. If a match had been found which required the computer to choose between 3 empty positions, it would in fact choose the FIRST empty position that its searches unearthed. If the sequence type was 7744 for example, with the occupied position being a 4-type, the computer would begin decoding and searching from the first position in the sequence - which is a type-7 here - and which is empty and thus occupiable; then the computer would respond with a strong type-7 move.

What takes the time in this initial version is the searching through all 76 winning line sequences in order to find the 4 or 7 which have the same position in them as the one which the opponent has chosen (this searching also has to be done when the computer has made its move!), and then incrementing the parallel Linesum array elements.

It would be much quicker (by a factor of 40 or more) if we could go DIRECTLY to the 4 or 7 elements of the Linesum array and increment them by the requisite value. This would require a series of 'pointers', answering the question: 'if I move into position 9, which Linesum array elements do I need to increment?'. The list of pointers (4 for the 4-type, 7 for the 7-type) could be held in an array, each element address being the same as the board position for which it is acting as a pointer list. Thus there would be 64 elements, each containing a sequence of digit pairs (so that no confusion can arise - at least for the computer!); Table 4 (ii) gives such an array, called an Update Pointer array or just Update array. If you look at element 1 in that array, you will see an enormous number: 04293537434851. Decode it, in pairs, and you get 04, 29, 35, 37, 43, 48, 51. Now where have you seen that before? Look at the top of the previous page and work down until you find the list of Linesum elements to be incremented as a result of moving into position 1 on the boards.

Although the Update array looks very complex, it is in fact generated from the WICS array itself. In the development program I wrote for this section (and which is even more involved than this one!), a simple loop ran through from the first digit pair in the first element of the WICS array through to the last digit pair in the last element. It used the digit pair as the address for the Update array, and the WICS element address as the number to be incorporated in the list of pointers for that element of the Update array.

It might help to make things a little clearer if I run through the same sequence of operations as the development routine did.

Beginning with the first decoded digit pair in WICS array element 1, we get 23. So, Update array element 23 has 01 (the coded version of 1) incorporated in its list of pointers. If you look at the 23rd element of Table 4 (ii), you will see that the first digit pair is indeed 01. Back to Table 4 (i) again: the second digit pair in element 1 is 42. Look at element 42 in Table 4 (ii) - and you will find 01 appears there. And so it goes on. The next pair gives 04, and element 4 of Table 4 (ii) has 01 coded into its list. Look at the very last digit pair in element 76 of Table 4 (i): it is 20. Look at element 20 in Table 4 (ii), and at the end of the list of pointers is 76.

Next issue should see the end of this article; you can heave a sigh of relief!!

Element	Contents	(i)
1	23420461	39 13492537
2	27384316	40 16522840
3	26391352	41 49525051
4	224300164	42 61546263
5	22261930	43 01040203
6	38423446	44 13161415
7	23271931	45 49615357
8	39423547	46 52645660
9	22232124	47 04160812
10	38392740	48 01130509
11	26272528	49 16643248
12	42434144	50 13612945
13	22380654	51 01491733
14	27431159	52 04522036
15	26421058	53 40245608
16	23390755	54 44286012
17	22390556	55 47316315
18	26430960	56 46306214
19	23380853	57 41255709
20	27421257	58 37215305
21	22420262	59 34185002
22	23430363	60 35185103
23	26381450	61 55545653
24	27391551	62 59586057
25	22271732	63 59556351
26	38433348	64 58546250
27	23262029	65 07060805
28	39423645	66 11101209
29	01160611	67 11071503
30	49645459	68 10061402
31	04130710	69 40443648
32	52615558	70 47464845
33	04491934	71 41374533
34	16613146	72 34353336
35	01521835	73 24282032
36	13643047	74 31303229
37	01612141	75 25212917
38	04642444	76 18191720

TABLE 4

Element	Contents	(ii)
1	0. 33537434851	33 26517172
2	21435968	34 06335972
3	22436067	35 08356072
4	01313338434753	36 28526972
5	17485865	37 10395871
6	13296568	38 02061013192329
7	16316567	39 03081016172428
8	19475365	40 10405369
9	18485766	41 12375771
10	15316668	42 01061215202128
11	14296667	43 04081214182226
12	20475466	44 12385462
13	03313639444950	45 28507071
14	23445668	46 06345670
15	24445567	47 08365570
16	02293440444749	48 26496970
17	25517576	49 02303329414551
18	05355976	50 23415964
19	07336076	51 24415063
20	27527376	52 03623540414652
21	09375975	53 19455361
22	04050913172125	54 13306164
23	01070916192227	55 16326163
24	09385373	56 17465361
25	11395775	57 20455762
26	03051115182327	58 15326264
27	02071114202425	59 14306260
28	11405473	60 18465962
29	27507475	61 01223437424550
30	05365674	62 21425664
31	07345574	63 26425563
32	25497374	64 04303638424649

Soapbox

This glance into the future is in two parts: this time a look at what tomorrow's new machine ought to have as standard, and next time a highly-personal view of the proposed Golden Age and a possible path to it.

Changing Times : Part I

It's not all that long ago that micro-enthusiasts fell over themselves to get a look at the latest machines possessing a whopping 8K or so of User RAM, usually costing a good £400. Some of them had to load their operating systems (language etc.,) from tape or disk, and colour ? Well, everybody knew that colour would cost an arm and a leg, and as for sound, well...

How far back do you think that is ? 10 years ? In fact, it's only about 4. The micro market has changed dramatically over the last few years, with predictions about the market direction and rate of hardware change being beaten again and again. Predictions about the social effects and the speed of implementation of some facilities have gone the other way, as fewer and fewer new applications are actually made use of. You might know the kind of thing: "We have the technology NOW to do so-and-so, but, political and economic considerations....".

When I bought my NTSC 99/4 (with its own specially-modified colour TV), it cost £695 for the console and £300 for the TV. If Binatone are true to their word, a 16K, colour and sound, micro for domestic use should be available in January for about £50. As the price of memory continually drops, and its speed increases, we are seeing 8 bit machines offering fast, 128K systems for very low prices. Looking at what is coming up soon, how might we specify the functions of say the next TI computer (ignoring the items in the pipeline now) ?

For a start, the processor. The 99s are 16 bit, and that seems to be a reasonable type to continue with, although most other 16 bitters give you far more directly-addressable memory than the TMS 9900 does. What about secondary processors ? The ones that handle display, input/output, and so on ? No great change there, I suspect. Memory ? Judging by current trends, 128K would seem to be the minimum (i.e., 128K User RAM, not total RAM/ROM). Colour ? Yes, and more than is currently offered. Perhaps 1024 colours, and 128 shades of grey ? Sound ? Built-in speech synthesis would appear to be a likelihood, with use of a dedicated sound synthesis chip to enable you to outperform Ultravox. External storage ? Currently this is either to tape or disk, with the expense of disk pushing most owners to tape. Perhaps built-in micro-floppies with 1 Mb density, or even

CMOS RAM packs (maybe the 128K+ User RAM would be CMOS ?). There is a tentative move in this direction. Other peripherals: the keyboard, with full QWERTY layout (PROPERLY laid out!), separate numeric keypad and separate cursor controller - probably a stick or ball control - in fact the whole keyboard separated from the main console, and not necessarily by cables!; colour printer/plotter, able to reproduce faithfully the image on the screen, or to produce bar-code listings; a flat-screen display unit, with built-in light pen OR touch-sensitive screen; a BUILT-IN RS232C (or later improvement) with modem, for networking; perhaps even a small robotic arm (or leg) - your unit could then cling to, or hop round after, you as you moved about.

The whole thing might come as a single package, for about £500. Included with the hardware would be a number of software items. For a start, compiled AND interpreted BASIC, Forth, Pascal, Assembler, Disassembler, perhaps LOGO, LISP, and other Artificial Intelligence languages. Perhaps also a Program Generator, for those who want the fun of electronic filing without the brainstrain of writing suites to implement it! On top of that, a firmware (permanently-resident in ROM) word processor. After all, most households have a typewriter; why neglect that area in the home, when it is one of the areas getting attention in the office ? The editing facilities already offered on most machines for program editing are a large step towards word processing..

You never know, the current bumbling experiments with 3D TV just might find their way into the micro world in 12 months time...

Science Watch

This issue the subject brings together many of the principles which have been presented in broad outline in previous issues. The subject is called PLANETTRAN, and it has been around since about 1954. Technical problems kept it on the drawing board until just recently, when micro-chip technology appeared to be able to solve most of the problems.

Planetran already exists in a less-ambitious form both in this country and in Japan. We haven't yet coined a term for it here, other than 'Maglev', which really describes the principle behind its operation. You can induce electromagnetic repulsion between a coil and a magnet if a sufficiently-strong magnetic field can be produced in the coil. To date this is achieved by making use of the phenomenon of superconductivity, where cooling the coil to around -270°C produces a situation where the electrical resistance to the passage of current drops to zero. Recent discoveries are leading to the 'engineering' of such superconductors so that they will operate at much higher temperatures, reducing the need for bulky coolant apparatus.

In Japan they have taken the principle of magnetic levitation and put^{it}_A to use in a

research project which has produced a 325 mph train, a speed more than double our fastest rail-riding best.

The Americans, however, who take anyone's beating them as a personal challenge, are looking to put everybody else in the shade with a 3,000 to 6,000 mph underground system, running from East to West coast. The plans involve depressurising the tunnel to reduce wind friction, and placing micro-electronic sensors at regular intervals to sense any undue swaying in the vehicle, which would then be corrected by relay stations situated at intervals along the tunnel. It seems that the sway problem has been the one holding the system back: in 1957 there was no technology small enough and cheap enough to do the job. By 1969, though, when the first micro-electronic components began to be made, the solution to the problem began to take shape.

The proponent of this system is one Dr Robert M Salter of the Rand Corporation, and it appears that many of his colleagues share his belief. It is no longer in the realm of Science Fiction. The system would operate with an acceleration of about 0.3G, so that passengers would, with the assistance of specially-engineered, gimballed, seats, feel only about 5% heavier. Fail-safe mechanisms are proposed, to allow emergency stopping, and there would be life-support systems aboard, as well as emergency exits in the tunnel walls. Mind you, with a depressurised tunnel, it wouldn't be a good idea to hang about for long!

Magnetic propulsion offers many advantages over other forms of travel. Traditional railway systems are reaching the end of their usefulness; they are slow, expensive, largely polluting, and by their nature are prone to mechanical failure due to excessive wear. Air travel is also expensive and polluting, as well as wasteful of energy, and often the daft situation arises where more time is spent waiting at either end of a journey than is actually spent flying between two places!

Perhaps the thought of travelling at 3,000 mph underground doesn't appeal to you, but then, the thought of travelling at 25 mph didn't appeal to our Victorian forebears, and the idea of travelling through air at 500 mph would have given them heart-failure.

Don't expect this type of transport to be available next year however. The projected appearance is apparently in about 15 - 20 years' time...

Postscript

Volume 2 Issue 5 of Tidings plummeted through the letterbox while I was still producing this Babble, and both it and issue 4 contain so many things worth commenting on that I'm spoilt for choice! Shining above everything else though are two articles which I have been waiting eagerly for for ages: those by Messrs Harding and Hutton. If nobody beats a path to their doors there is something badly wrong with the world. I have seen Gary's program and it is first rate; I am trying to persuade him to write a series on Assembly Language Programming for the New Year: watch this space...

If TI have their eyes open while reading Tidings, then Eric Hutton's article should send them reaching for their Re-thinking Caps. I know that one of the companies which have advertised in Tidings are cutting prices down to size, but even they would be hard put to match £94 for a 32K RAM expansion (and just think: produced on a large scale, such an expansion could begin to approach the more realistic prices which other folks pay for their memory add-ons!).

Some of the points raised in this Babble have either been raised in issue 5 or even partly-answered; I don't have the time or the energy to retype BB to take this into account!

For those who like to wear their finger-ends down, I have included a program listing, which, if you are tenacious enough, will give you the fastest plot in the west when you run it. If you like more than one of everything, then add the modification given at the bottom of the listing. The resulting 3D plot should show you just what you can do given decent graphics-handling and 3 hours of typing! I'm working on a version of my early plotters which will do all the work for me (similar to Stephen Shaw's Save-A-Sketch) but using the Minimemory module.

Finally, if you don't have the time or confidence to put yourself up as a contact for TI owners in your area, maybe you'd prefer to act as a kind of tutor for those members who are still struggling up the slippery path to programming proficiency? Why not offer your help either through Tidings or through Paul Dicks? You might find that although YOU think you're a beginner, the REAL beginners would give their eye teeth for the experience you have gained already. Why not pass on some of your experience; a word to the wise to a beginner could save them weeks of effort, and you will always learn a thing or two yourself! Being a 'contact' for your area just might turn out to be the best thing you ever did...

Pete Brooks

```

120 REPTORE
110 DATA 000000FF0000FFFF,9,16,0
00000000FF00FF,8,16,FF000000001
FE000,7,16,0000000000071950,6,16
120 DATA 100003E01F30C000,5,16,0
000007F80804E0,4,16,000000FF000
0FF3F,9,15,0000000000FF00FF,8,15
130 DATA FEFF0000000000FF0,7,15,0
480E31F3640092,5,15,00000007F9
10340,4,15,0408100020401080,6,15
140 DATA 000000FF00FFFF00,9,14,0
7126080FEFF00FF,8,14,00011E60800
00000,7,14,01500200A40008FF,6,14
150 DATA 7E7E45A912C42089,5,14,0
000000000707EF9,4,14,000000FF00F
FFF00,9,13,800000E1FF1F00FF,8,13
160 DATA 1E080000030C1950,7,13,4
10A02238404FE03,6,13,229049000248
01300,5,13,00000000000003FC,4,13
170 DATA 0000F80700FF0300,9,12,4
08106FFFF0000FF,8,12,18204083040
81020,7,12,90052008E21E6186,6,12
180 DATA 3FD2A84510A00A40,5,12,0
000FF00FF1F0000,9,11,80D0FF017F8
000FF,8,11,8224488011224408,7,11
190 DATA F0E97EBF59A24C11,6,11,0
007FE8DF6F8E3DC,5,11,0000FFF83F0
00000,9,10,F43FFF0000FF00FF,8,10
200 DATA 9A6570CA84D14AA0,7,10,1
7AD0F5E953A2974,6,10,0000033FE95
5A843,5,10,00FCFB0F00000000,9,9
210 DATA FF001FE000FF00FF,8,9,0C
31C20CF106F9FF,7,9,D224C91264881
1E2,6,9,00000000077DFAAD,5,9
220 DATA FCF703000000000,9,8,00
FF00FF00FF00FF,8,8,DOFEE9F8FD3FC
1FF,7,8,FCF7F8FFC09FE0FF,6,8
230 DATA 0000000000010FFF,5,8,07
0000000000000,9,7,9FE0FE3DCFFF
F7F,8,7,55FFFFBFF3DEE778,7,7
240 DATA 3FFDF000000000,6,7,00
00000000000005,5,7,FFFFFFFFF1F0
300,8,6,FFFFFFFFFFFFEFFF,7,6
250 DATA 00011F7FF000000,6,6,3F
1F0F0300000000,8,5,7FFFFFFF0000
F7F,7,5,0000000003071F3F,6,5
260 DATA 0000000100000000,7,4,00
0000FF0000FFFF,9,17,0000000000FF
00FF,8,17,FF010000000F80700,7,17
270 DATA 000000000E01806,6,17,0

```

```

8100007F8801300,5,17,000000FE010
10204,4,17,000000FF0800FFFC,9,18
280 DATA 0000000000FF0,9,18,0
FFF00000000000000F,7,18,206547F8000
20001,5,18,00000000FE91002,4,18
290 DATA 2010080004020001,6,18,0
00000FF00FFFF00,9,19,E01806017FF
F00FF,8,19,0080780601000000,7,19
300 DATA 800A4000250010FF,6,19,7
E7EA29548230411,5,19,00000000000E
07E9F,4,19,000000FF00FFF0,9,20
310 DATA 01000037FF000FF,8,20,7
8060100C0300806,7,20,825040C4212
07FC0,6,20,4409282002401C800,5,20
320 DATA 000000000000C03F,4,20,0
0001FE000FFC000,9,21,928160FFFF0
000FF,8,21,180402C120100804,7,21
330 DATA 09A0041047788661,6,21,F
C4B15A208055002,5,21,0000FF00FFF
80000,9,22,010BFF80FE0100FF,8,22
340 DATA 412412018844210,7,22,0
F977EFD9A453288,6,22,0E07FBD6F1
FC738,5,22,0000FF1FFC000000,9,23
350 DATA 2FFCFF0000FF00FF,8,23,5
9A60E532585E205,7,23,E885F07AA95
C942E,6,23,0000C0FC97AD502,5,23
360 DATA 003FDFF000000000,9,24,F
F00F80700FF00FF,8,24,308C43308F6
09FFF,7,24,4B24934826118847,6,24
370 DATA 0000000E08E5F85,5,24,3
FFEC00000000000,9,25,00FF00FF00F
F00FF,8,25,0B7F976FBFFC83FF,7,25
380 DATA 3FEF1FFF03FD07FF,6,25,0
00000000080F8FF,5,25,E00000000000
00000,9,26,F907FFC73BFFFFFFE,8,26
390 DATA FFFF00000000000,7,26,F
CFFF8FFFFFFF000,6,26,0000000000
000A0,5,26,FFFFFFFFFFFF8C000,8,27
400 DATA FFFFFFFFFF7FFFF,7,27,0
080F8FE7FFFFFFF,6,27,FCF8F0C0000
00000,8,28,FFFFFFFFFFFFE,7,28
410 DATA 000000000E0F8FC,5,28,0
000008000000000,7,29
420 CALL CLEAR
430 CALL SCREEN(8)
440 CALL HCHAR(1,1,31,768)
450 FOR I=32 TO 153
460 READ A$,R,C
470 CALL CHAR(I,A$,R,C)
480 CALL HCHAR(I+6,C,I)
490 NEXT I
500 GOTO 500

```

```

470 CALL CHAR(I,A$,R,C)
480 CALL HCHAR(I+6,C,I)
490 NEXT I
500 GOTO 500

```

The Fastest 3D Plot in the West!

TI NEWS

Let me start by apologising for missing the last exciting edition of TIDINGS, here at TI everything is go, with the considerable increase in demand for the TI-99/4A we are all very busy.

I shall start this newsletter by answering some of the points that arose in October's edition of TIHOME. Firstly Mr Dicks (I always need to be polite to him!) referred to GROM serial numbers in his editorial. Sorry Paul but you have it wrong, LTA and ATA refer to the site of manufacture e.g. L(TA) is Lubbock and A(TA) is Almelo in Holland. Further TI Invaders was released long after the introduction of the TI-99/4A, and anyway users should not see any differences between the 4 and 4A on this game. The only difference that very few of you will see applies to those of you who own a NTSC TI-99/4A, since you will have a longer screen size.

Under Dorse Dialect by Jeff Freeman, there was a section relating to TI Servicing. This story of woe prompted me to see sorrowful Sid of the Service Section, deep in the bowels of TI. Sid said "Oh dear, let me check on this immediately". True to his word Sid reported back. "Well", he said, "it would appear that we received Mr Freeman's computer at 9.24am 19th August and despatched it 27th August 2.19pm". At this stage Sorrowful Sid broke down with remorse "I am so sorry, we had the computer for 8 days and not 5, but Ted the technician's aged aunt (the one with the money) was taken ill, and Ted had to be inattendance." I managed, with difficulty, to calm Sid down by telling him that I am sure that the nice Mr Freeman would understand. As for sending his computer back covered in Foam Chips (SIC), that was a mistake and our apologies for it.

Now onto new products. In January we will be launching a new module, called PARSEC, it is a high speed arcade type game and I believe it is the best we have produced to date. PARSEC is similar to Defender as sold for ATARI, although I think the graphics are better. Judging by the reaction of the people here in the office once seen you cannot put it down. PARSEC is also the first game to include speech, not your standard boring male voice either, but a sultry females voice. It is worth pointing out that the Speech Synthesizer is not required to run PARSEC, but if used will make the game even more enjoyable.

.....Cont/d

I am pleased to see some new contributors to TIDINGS, but I feel sure that there are out there some budding authors still to put in an appearance. I am sure that TIHOME would be pleased to here from some of the less experienced computer users. Also, it is good that four brave lads have come forward as local contacts, anyone in their areas, get in contact with them, and I am sure there must be a few more out there who are willing to start local clubs etc. Get in contact with TIHOME, it is to your advantage.

I understand from Paul Dicks that membership is booming, sales of the TI-99/4A have increased dramatically and we are now sold out for the rest of the year!

Finally my congratulations to Paul on the new style of TIDINGS, I think it is excellent.

Have a merry Christmas and a happy New Year.

ROBIN FROWD
Texas Instruments Ltd

RAMBLES

by
Stephen Shaw

Greetings and welcome to Rambles. A little shorter than usual this issue - the 'press date' is shorter, as we catch up on the issue date, and not many of you have been asking me questions!

There is a lot of good news elsewhere in TIDINGS so it remains for me to give a little assistance to those of you who have just bought your first computer and are meeting with those frustrating problems we have all met with.

TAPE RECORDERS:

I recently had a phone call from a TI owner who had tried an awful lot of recorders - and none would work. The problem was the REMOTE CONTROL operation. With many recorders sold in the UK it is necessary to reverse the polarity of the remote lead - and TI include with the cassette cable two small adaptors for this purpose (one for CS1 and one for CS2). Use one of these for trouble free operation - do not do as a second TI owner did and discard them!

There are differences between recorders in the ease with which you can load tapes recorded by other people - almost all will allow you to load your own programs with greater or lesser difficulty!

In my own experience the BOOTS CR325 is outstanding in its performance with the 99/4A, while the cheaper LLOYD VL82 does a very good job too (but is built a little less strongly!).

ALPHA/CAPS KEY: At the left of the keyboard is a key which switches from small letters to large letters - in the down position the computer will print the larger letters. This seems to be causing a little problem with several owners.

When you are keying in or loading a program, you should keep this key in the DOWN position. It is usually essential to keep it DOWN while RUNNING a program, but there is one important exception - see next page....

JOYSTICKS: When the program uses JOYSTICKS you must keep the ALPHA/CAPS key in the UP position - otherwise they do not work well! We have heard of several owners returning joysticks to their dealers as 'not working' when all that has been wrong is the position of the ALPHA/CAP key!

RUNNING THIRD PARTY PROGRAMS:

With the best tape recorders there is no problem, but with the intermediate tape recorders - probably the majority held by 99/4A owners - you will find you will have to increase the volume setting to load a third party program. The volume range at which the program will load may also be reduced.

For those few 3rd party programs which come with DATA FILES (the tape runs and stops several times to load them) there is also a problem of one or more files being missed - this usually gives an error message such as BAD VALUE - which is not quickly associated with loading problems! This difficulty usually means the volume is a tiny bit too low.

PROGRAM DEBUGGING

While receiving a lot of comments on the programs in Computer & Video Games, some have experienced no problems while for others the programs just will not run!

Bear in mind the earlier note about the ALPHA/CAPS key! It is VERY easy to make an error in keying in a program - and spotting a missing comma or bracket may not be so easy. Much harder to spot are letter O's used instead of figure zero's or letter I's instead of number 1's.

The 99/4A has some good features for helping to spot errors - but when the error is many lines away from the line producing the error message, you have to understand the program and the computer!

A recent problem reported to me involved a program which when RUN gave the error message:

DATA ERROR IN xxx

The owner found xxx to be in a FOR-NEXT loop, so adjusted the TO NNN number downwards, and found the program ran OK - but an error occurred elsewhere- BAD ARGUMENT in.....

The SYNTAX of this new line was OK - so what was going wrong?

See next page for next enthralling episode (can you guess?)

Whenever an error message is generated, the program is halted - but all the variables retain their values until you edit a program line.

By looking at these variables you can work out what went wrong. In this case, the FOR-NEXT loop which had been adjusted was loading DATA into an array $R\cancel{A}(n)$, and the BAD ARGUMENT line was using the function VAL ($R\cancel{A}(n)$).

First step: type in PRINT $R\cancel{A}(n)$ - the result was a single line scroll - eg $R\cancel{A}(n)$ was a nul string, resulting in the bad argument (You cannot VAL("")).

The next step was to look at all the data items, and see to what variable the last data item appeared to be READ to - in this case $R\cancel{A}(15)$ - which was NUL.

Now we had to trace the SIZE of the problem- by stepping back a few DATA items and telling the computer to print the variable we thought should have been loaded with that string - this time there was a result - but not the expected one: The computer printed the PREVIOUS data item - which told us that ONE comma had been omitted in one of the DATA lines - so back to do a careful check of that section.

If the DATA section had been very large we could have checked 'anticipated' values for a series of variables, to see when we had the anticipated answer - and when the answer was wrong.

(This does require a careful check, both of data items, and of READ statements, which are usually in a FOR-NEXT loop!).

So - if you have a problem with a program, as well as checking every comma in every line, the computer CAN help you to find your mistake. In the above example the action taken at first (assuming a magazine print error) only served to obscure the real problem - a DATA ERROR should lead you immediately to check your DATA lines, using the above procedure.

Magazines make some mistakes of course - but the chances are that its your typing! As you develop a knowledge of your computer you WILL learn how to spot and correct some magazine printing errors (there aren't THAT many!).

Offer: If a program bug really stumps you, send me a cassette of what you have, together with return postage (don't send the cassette case!) and I'll have a look at it for you. If not a 99'er prog, please send the original listing. May take a while to reply. It is really impossible to debug a program by letter, and not very easy by phone!

BACK TO THE GOSSIP: 99/4A's made in ITALY -may help keep the price down!

Lots and lots of serious errors reported in the Editor/Assembler Manual - please may we have a list of what you have TI?

Thorne*EMI in an advanced stage of preparing to launch programs for the 99/4A - we know of at least one program they have bought.

99'er magazine now MONTHLY - lovely mag- do get it. 99'er reports it IS possible to upgrade a 99/4 to the same graphics as the 99/4A - how about it TI?

SCREEN D 1

If you wish to save a screenfull of information or graphics, how do you do it?
 You could use CALL GCHAR and then save the variables to a tape file- but it is both slow and uses an awful lot of tape -and you cannot (normally) verify tape data files.
 If your screen contains redefined characters you may not have enough tape to save the definitions (assuming the use of CALL CHARPAT - Ext Basic).

So how ?

Using a disk system for the data files is a lot faster - but still not especially fast.

You could write a program which saves the data TO the program (using either mini-memory or ext.bas with 32k ram.) This is certainly faster, but all those CALL CHAR lines take up a great deal of memory. It is OK if you are not redefining the characters- then all you have is 24 'PRINT' lines to overwrite.

You can do wonders with Assembly language of course, and such a utility could fit into the Mini-Memory - but the MMM has BASIC utilities which come in very handy! Below you will find an example- the program on the LEFT saves the screen to the MMM, the program on the right reads it out - and in between you can power down and remove the MMM.

Tape storage is possible using the 'L' and 'S' options of EASYBUG on the MMM - the files CAN be verified and load quite quickly. These examples also save definitions. Can you follow these programs?:-

LOAD

(FOR MMM)

DISPLAY:

LOAD MMMM

```

1000 CALL INIT
1010 FOR T=0 TO 750 STEP 14
1020 CALL PEEK( T,A1,A2,A3,A4,A5
1030 A7,A8,A9,B1,B2,B3,B4,B5,B6
1040 B9,C1,C2,C3,C4,C5,C6
1050 CALL LOAD( T+30000,A1,A2,A3,
A4,A5,A6,A7,A8,A9,B1,B2,B3,B4,B5
B6,B7,B8,B9,C1,C2,C3,C4,C5,C6)
1060 NEXT T
1070 FOR T=1024 TO 1041 STEP 20
1080 CALL PEEK( T,A1,A2,A3,A4,A5
1090 A7,A8,A9,B1,B2,B3,B4,B5,B6
1100 B9,C1,C2)
1110 CALL LOAD( T+30000,A1,A2,A3,
A4,A5,A6,A7,A8,A9,B1,B2,B3,B4,B5
B6,B7,B8,B9,C1,C2)
1120 NEXT T
1130 END
    
```

```

100 CALL CLEAR
110 FOR T=1024 TO 2041 STEP 20
120 CALL PEEK( T+30000,A1,A2,A3,A4
A5,A6,A7,A8,A9,B1,B2,B3,B4,B5,
B6,B7,B8,B9,C1,C2)
130 CALL LOAD( T,A1,A2,A3,A4,A5,
A6,A7,A8,A9,B1,B2,B3,B4,B5,B6,B7
B8,B9,C1,C2)
140 NEXT T
150 FOR T=0 TO 750 STEP 14
160 CALL PEEK( T+30000,A1,A2,A3,A4
A5,A6,A7,A8,A9,B1,B2,B3,B4,B5,
B6,B7,B8,B9,C1,C2,C3,C4,C5,C6)
170 CALL LOAD( T,A1,A2,A3,A4,A5,
A6,A7,A8,A9,B1,B2,B3,B4,B5,B6,B7
B8,B9,C1,C2,C3,C4,C5,C6)
180 NEXT T
190 GOTO 190
    
```

You turned the page rather quickly then!

Those listings are not as bad as they look-

CALL PEEKV reads values from VDP RAM- that is where the SCREEN content and character definitions are kept - and in TI BASIC, also your program.

CALL PEEK (no V) - reads values from CPU RAM - the console on its own has only 255 bytes (actually in the CPU Chip). In this program it is used to look at CPU RAM in the Mini Memory Module.

CALL POKEV places values into VDP RAM while **CALL LOAD** places values into CPU RAM.

The format is **CALL PEEK(MEMORY LOCATION, VALUE, VALUE, VALUE..etc)**

The first value is that in the memory location, the second value is that in the next highest memory location and so on.

Although not used here, you can also split the commands by using a nul string eg

CALL PEEK(LOCATION ONE, VALUE, VALUE, "" , LOCATION TWO, VALUE)

So lets have a look at those two listings (and see what is wrong with them!):

LOAD: CALL INIT clears the CPU RAM and ensures you don't trip over anything already there.

Lines 1010 to 1040 look at the screen contents, details of which are in VDP RAM at locations 0 to 767 (32x24) with 0 at top left and 767 at bottom right. The values stored in these locations are the ASCII CODE plus 96 (eg A is 161).

I have used several locations in each **CALL** for greater speed.

The **CALL LOAD** places these values into the MM CPU RAM - the actual address used is not important so long as the address used is actually IN the MM ! (locations 28672 to 32767 - ALL numbers used here are decimal). Having placed the values into MM they will stay there even when you switch off - until you initialise or replace them.

They can be saved to tape- with verify function.

So- we have read the screen and placed it into MM.

Now lines 1050 to 1080 read the character definitions from VDP RAM and place them into MM.

Character definitions for ASCII 32 to 96 are held in memory locations 1024 to 1535. Each character definition takes up 8 bytes. We can infer that the 'lower case' characters are initially derived as they are not defined. When you define a character over 96, the definition uses a higher memory location, from 1536 up. The highest memory location used in character definitions is thus 2040.

Now the information is in MM it can be recalled to the screen by reversing the process - reading MM and placing the values in VDP RAM * - continued...

The DISPLAY program does just this - moving values from one location to the other, to recreate the screen.

I have first transferred the character definitions and then placed the characters onto the screen.

NOTE A- In this example if you have used characters 97 & 98 you may notice something odd. In normal use memory location 1536 up is used for the VALUE STACK (eg variables), and if you define extra characters with CALL CHAR the value stack is automatically moved up to make room for the extra definitions.

In this case we have only moved the definitions to this area of memory - we have not instructed the computer that the value stack starts at a higher address - so it carries on using 1536 up - messing up our definitions for Chars.97 & 98.

You could get round this by forcing the stack higher by using:

```
FOR T = 97 TO 159
  CALL CHAR(T, "O")
NEXT T
```

before you move the data from MMM to VDP.

NOTE B- If you do not have redefined characters you need only transfer VDP addresses 0 to 750.

If characters ARE defined you need only transfer the data for the redefined characters - ASCII 32 is in 1024 to 1031 etc up.

NOTE C- The programs given take 25 seconds to recreate the screen, which is the same as using CALL CHAR for every character and then using 24 'print' statements. This way is much easier to save redefined characters.

NOTE D- Once in MMM the data can be quickly saved to cassette using the EASY BUG commands 'S' and 'L' for locations (hex) 7000 to 7FFF. You will be able to use the tape-verify option, and not much tape is used.

This example is given to show you something the MMM is capable of - it is not the best solution to every problem!

→ Here is another program to try
with the MMM - can you do this any faster in any other way? (Use a stopwatch!).

SHORT ONE-
(For MMM)

```
100 CALL CLEAR
110 FOR T=0 TO 768 STEP 33
120 CALL POKEV(T,126)
130 NEXT T
140 GOTO 140
```

If you have a problem, let me know - enclose an SAE for a reply and allow plenty of time for a reply! The most persistent or more interesting problems will get a mention in RAMBLES at some future date.

NB: No daytime callers or phone calls - not in! If a phone call is urgent please ring after 7.30p.m. (not Wednesdays). I do not keep a stock of programs for sale, so cannot sell you any if you call - mail order only!

Address: 10 Alstone Road, STOCKPORT, Cheshire, SK4 5AH

(PLEASE use the Post Code- letters are subject to great delay if you dont. Also -PLEASE let me have your name and address in CAPITALS - I've been unable to read some recent post.).

Don't forget - the mail gets pretty slow around Christmas.

TIHOMÉ is always happy to receive new contributors!

PRODUCT REVIEW

I have recently purchased an INMAC 'suspension copyholder' -this is like an Anglepoise lamp, which clamps to your desk or table, but instead of a lamp there is an adjustable platen to hold your copy (eg magazine).

I have found it of very great use in entering programs & data, as the magazine etc can be placed just where you want it.

Available by mail order at £29 plus carriage plus vat it is not cheap - but very useful if you do a lot of entering.

INMAC are at Runcorn and keep giving their catalogue away with various computer mags.

CRIME: Watch out for modules offered for sale without manuals. Watch out for modules offered for sale with a piece of chain attached.

Sad to say, after the theft of hand controllers at PCW, Computer Supermarket have suffered the loss of an Extended Basic module - which was chained to the display stand!

Now, as the module is rarer than gold, they cannot demonstrate my programs. If you are in the North West and want to see some of the programs I offer call in to Computer Supermarket.

TI: There was once a large multinational company with a profit figure larger than you can imagine, who invented lots of good things. They made a computer, overpriced it, and then to encourage sales offered a rebate. Pity they didn't make sure there were adequate supplies first. Result- frustrated buyers (potential) and dealers. No EXBas, No printer - hummmmm...

NEWS HOT FROM AMERICA:

The Following news is from private sources and is not confirmed with TI:

NEW TI COMPUTER next year based on 99000 CPU, 'compatible with 99/4A peripherals and software', and with features comparable to the IBM PC - initial launch date June 83 (but note TI usually run very very late on their launch dates, and it may take longer to cross the Water - if it does.).

PRICE will of course be a bit higher - but not more than double the 99/4A price.

Believe it - or don't - but it is possible.

News of 75000 99/4A sales in the States in just ONE MONTH!

FAST EXTENDED BASIC: The extra speed of the language is only apparent in running a reasonable size program, and does not show in BENCHTESTS - of particular note is the speed of line transfers, and the much faster screen handling. Note that it should be a tiny bit faster with the 32k RAM.

Could someone with the 32k AND Version 110 please resolve this by running a fairly lengthy program and checking the times in TI BASIC, in EXTENDED BASIC 110, and same with 32k RAM.

EX-BAS:-

More news from America -of a variation in firmware which is causing problems in programs with Sprites moving upwards or downwards- ExBas version 110 contains a new subroutine which keeps track of the number of sprites in use.

Normally when a Sprite moves downwards off screen it is still considered 'active' and movement from bottom to top is smooth- even though you cannot see the sprite for some of the time.

It is apparent that in some versions of ExBas 110 this is not the case- the 'off screen' section is no longer considered as an active area, and sprites 'suddenly' appear, just as they do at screen left.

This has been causing problems in programs where:

- i) A sprite is started from 'off screen' - it will actually start in a different location than intended
- ii) Sprites move vertically with different rates - there can be timing differences in the program execution.

So- if you need a fully transportable program you can no longer start a sprite in the invisible off-screen area, nor can you rely on vertical timing.

Apparently only SOME Vn110 modules behave this way. Please would TI inform us if this applies to EARLY releases or to the CURRENT release.

Vn 110 only is affected. Vn 100 doesn't keep count of sprites.

Special Thanks to Mike O'Regan for the Joystick Cure in last TIHOME.

Anyone with Graphics talent out there- how about a LOGO for TIHOME, for TIDINGS and for a T*Shirt (who can afford sweat shirts?).

Warm welcome GARY - PLEASE would you write lots about Assembly language for us BASIC programmers? NB: A disassembler in the USA is listed at US\$95.....

NO news from TI in last TIHOME.... too busy looking for stock probably...

With so much discussion of TI's prices, here are some comparisons- of discount prices in the USA and in the UK.

Prices advertised in the USA DO NOT include Sales Tax - Prices quoted in UK ads usually INCLUDE V.A.T.

TI have to pay CUSTOMS DUTY on their imports, and there is also the cost of actually lying the products in.

Consider: In terms of engineering the TI product cannot be compared with any other. True their firmware is rather poor.... but the product is made both to last, and to perform well.

Compare the TV display of a DRAGON to a 99/4A.

Take a good hard look at a 'cheap' 32k expansion and TI's 32k expansion - methods of construction are quite different.


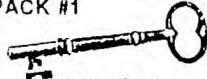
I'm not defending TI's strange pricing policy on modules - but the hardware prices at least are not the rip-off they once were.

Item	Discounted U.S. Price		Final US Price in £ at 1.65	Discounted UK Price	
	Basic \$	+6% Tax \$		Basic £	+15% Tax £
CONSOLE	300-	318	192.00	173-	199-
Rebate (TI)	100-	106	64-	43-	50-
Free Software	—	—	—	43-	50-
FINAL CONSOLE	\$200-	\$212	£128-	£86-	£99-
Speech Synth	\$125-	\$133	£81	43-00	50-00
BOX	196-	208	126	130-	150-
32K Card	236-	250	152	130-	150-
RS 232	138-	146	89	99-	114-
Ext Basic	80-	85	£52	£66-	£75-00
Pers. Rec. Kptg	41-	44	27	38-	44-00
TI Invaders	35-	\$37	£22	£15-	17-25
Car Wars	35-	37	22	22-	25-00

SOME items are cheaper here than in the States!

But the ExBas and PRK are still too high in price!

The following programs are available by MAIL ORDER ONLY from:
 Stephen Shaw, 10 Alstone Road, STOCKPORT, Cheshire, SK4 5AH.
 Prices include p&p and are valid to 31.3.83. UK Orders only--sorry.

PS SOFTWARE	
NEW	
DOUBLE PACK #1	
 OCTAL 1 Can you save your system from the insidious Octtug Spydys?	 Keys of the CASTLE An adventurous, graphic game of fast action and quick thinking.
16K Extended BASIC tape Both for £9	
PS PESTEROIDS	(Extended BASIC) - Has features like changing characters and special options that put it ahead of the Space-Rock Pack! on tape £9
SPACE RESCUE 2.0	(Extended BASIC) - Evade the Xyolian Hordes and rescue the power pods in PS SOFTWARE's most popular game. on tape. £9

Heuterware PRESENTS

BLUEGRASS SWEEPSTAKES

1st - ROCKY

2nd - INVICTUS

3rd - OLD PAINT



"You pays your money and you takes your chances" Turn your \$1000 into a million in this 9 race game for 1 to 8 players. Go for the "big bucks" in the Sweepstakes Race Excellent graphics make this an enduring family favorite (great for parties, too). (BASIC only)

BLUEGRASS SWEEPSTAKES... **£6** p.p.d.
 (T.I. BASIC)

NEW:

IN TI BASIC
 from Kuhl Software

FISH

£6.00

Simple lo-res animation. You vs your computer in a fishing contest - 3 types of fish, 5 baits, 5 locations. After selecting your choice of combination you just watch your rod and press 'B' when the fish nibbles the bait. Suitable for younger owners, not unsuitable for older ones.

Good programming. Good demo prog.

Mr. & Mrs. S. Shaw
 10 Alstone Road
 STOCKPORT
 Cheshire SK4 5AH

Coming Soon: At last!
 Programs from OAKTREE SOFTWARE
 Reported as On Their Way...
 GROSSUMS- numeric Scrabble
 SMASH- Program compactor
 and later on
 DISPLAY ENHANCEMENT package.

Possibly available Jan 83?

PRODUCT NEWS:

STARSHIP PEGASUS: A revised version has been issued from 12.11.82, with improved chances of survival.

WINGING IT: A revised version has been issued since 11.11.82 with an improved display for Game 1, and slightly easier weather.

If you have Vnl and would like the new versions, return your tape with £2. If not bought direct please state where purchased.

An extra 'Hints' sheet has also come in for WINGING IT (11.11.82)- just send an SAE if you'd like it but not version 2.

ALL OF THE PROGRAMS ON THIS PAGE ARE AVAILABLE BY MAIL ORDER

ONLY from: Stephen Shaw
10 Alstone Road
STOCKPORT
Cheshire
SK4 5AH

A large illustrated
catalogue with over
90 items of software
is yours for just
22p in stamps.

(Please USE the post code!) (Your address in CAPITALS please!)

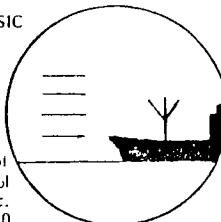
LOTS of new programs since last TIDINGS -

Here are some of the more

interesting new ones:

Pewterware PRESENTS
UP PERISCOPE

in EXTENDED BASIC
Depth - 60
Speed - 6
Course - 320
Bearing - 310
Range - 3,000



You are the skipper of the USS Porpoise on patrol in the Western Pacific. "Final Bearing Mark - 310 degrees." "Final Range Mark - 3,000 yards." "Fire One!" "Fire Two!" You watch through the periscope just long enough to see two torpedo wakes converge on a 10,000 ton tanker and send it to 'Davy Jones Locker.' "Take 'er deep; right full rudder; rig for depth charge." The destroyer escort has already started looking for you. Written by a 20 year submarine Vet.

UP PERISCOPE . . . £9.- postpaid
(Extended BASIC Only)

PEWTERWARE

Also available (in BASIC):
Bluegrass Sweepstakes
Challenge Poker
Decathlon
Meth Wits
Each £6 postpaid

ADVENTURE PROGRAMS in
EXTENDED BASIC - both
require 32k EXPANSION RAM:
Both by Sam Pincus from
Data Systems:

MAD SCIENTIST - rescue the daughter
before dad blows the (large) house
up. A simple adventure suitable
for beginners - on tape £8-00

STONEVILLE MANOR- A more advanced
adventure with lots of things you
have to do - you can't just wander
around with this one! A big
program so DISK ONLY for £10-00

From MIKE O'REGAN in EXTENDED BASIC
we have a WORD PROCESSOR for the
THERMAL PRINTER (NOT the dot-matrix
printer) On tape for £8-00

and with Dave Hamilton a simple
but original and entertaining
game PEN THE PIG - trap the pig in
one of the pens, as he bounces off
(sometimes through...) your walls.
With invisible wall option!
Recommended for family fun and
only £4-00 on tape. (T.I. BASIC)

*** IMPORTANT**

Prices given in last TIHOME were
stated as valid to 30.1.83.
I regret that due to continuing
falls in exchange rates those
prices will now only be valid
to 31st December 1982.

Members of TIHOME may order at those
prices to 30.12.82, although I
shall be increasing catalogue
prices before then. Sorry!

I have had a telephone call to say
programs from FFF SOFTWARE are on
way - may be in next issue.

PRICES ABOVE VALID TO 31.3.82
Supply to UK only.

- CATALOGUE: Send one of:
- i) 9x6" S&B
 - ii) 22p in stamps
(covers cost of envelope too!)
 - iii) 50p, which is refundable on
first order.

Please mention TI HOME!!!

Prices will be held at stated
level unless there is a major
fluctuation in exchange rates
(e.g. £ goes below 1.60).

TIPS

from

U.S.A.

International
99/4
Users-Group



Hidden Statements In Extended Basic

Recently we have found some interesting things when using the Extended Basic with the 99/4A console. For instance, try this:

```
10 FOR A=1 TO 100
20 PRINT A
30 NEXT A
```

Let's count the keystrokes ...39. Now type LIST. What we get is an exact echo of the program we typed in above. Now let's try something a little different. Type in the following: (For the purpose of the indication of when to hold down the control key and type a letter, we will use the following symbol: "@". This would mean if you see @C, you would hold down the control key while depressing the letter C on the keyboard.)

```
10 @L (enter)
20 @: (enter)
30 @V (enter)
```

Now type LIST. Your screen should read:

```
10 FOR
20 PRINT
30 NEXT
```

Now let's take this series a step further. This time let's use our old friend the REM character (!) and type the following: (Don't forget when you see the @ you must depress the control key as well as the key indicated.)

```
10 ! @L=100 (enter)
20 ! @: A (enter)
30 ! @V A (enter)
```

Now type LIST. As you can see, you now have our original program on your screen preceded with (!), the REM statement. If we were to remove the (!) REM statement either manually or by using a program to do so (this can be done with

*"TI informs us
that this was not
their original intention
for these functions."*

a disk system only), and type RUN, the program would execute in the same way that our original program did.

Although TI informs us that this was not their original intention for these functions, and that they do not recommend this type of programming method, we have used it in several programs and find it to work quite well.

For your convenience we have listed all of the hidden statements and the keys to make them operate.

Try working with this yourself. If you find any other uses we missed, please let us know.

1 (TO)	2 (STEP)	3 (,)	4 (:)
5 (;)	6 (!)	7 (!)	8 (OPTION)
9 (OPEN)	0 (THEN)	= (CALL)	Q (UNTRACE)
W (READ)	E (GO)	R (INPUT)	T (RESTORE)
Y (DELETE)	U (RANDOMIZE)	I (DEF)	O (UNBREAK)
P (TRACE)	\ (AND)	A (ELSE)	S (DATA)
D (IF)	F (GOTO)	G (GOSUB)	H (RETURN)
J (DIM)	K (END)	L (FOR)	: (PRINT)
K (REM)	X (STOP)	C (!)	V (NEXT)
B (:)	N (BREAK)	M (LET)	> (ON)

CLOSE ROUTINE.

I am delighted to report that many more members have offered to become county contacts.

The list, at the moment, is thus:-

NOTTINGHAMSHIRE

Mike O'Regan, 130 Stapleford Lane, Toton, Beeston, Notts, NG9 6GB
Tel:- 06.076-5482

OXFORDSHIRE

Peter Brooks, 68 Kelburne Road, Cowley, Oxford, OX4 3SH.

STRATHCLYDE

Peter Phillips, 16 Lomond Road, Bearsden, Glasgow, G61 1BA.

LANCASHIRE

Gerry Howard, Briarcross Cottage, 72 Watling Street, Assetside, Bury,
BL8 32W

LINCOLNSHIRE

B & A O'Reilly, 38 Regent Ave, Birchwood, Lincoln, LN6 0PQ.

SOUTH LONDON

Peter Wallis, 8 Abbey Grove, Abbey Wood, London, SE2 9EX.

Co DURHAM

I G Haszeldine, 59 Hundens Lane, Darlington, Co Durham, DL1 1JJ.

EAST LONDON

Alec Stanley, 4 Victoria Court, Victoria Road, S. Woodford, London,
Tel:- 01.530-6248 E18 1LH.

ESSEX

B E Bailey, Haldane Mill, Althorne, Essex, CM3 6BX.
Tel:- 06.217-40189

NORTH LONDON

Arieh Yacobi, 56 Winchester Road, Kenton, Harrow, Middx, HA3 9PE.

DORSET

Jeff Freeman, Cutlers, Abbey Road, Sherborne, Dorset, DT9 3LE.

DYFED

Tim Cairnes, Caedy Crib, Talybont, Nr Aberystwith, Dyfed, SY24 5EZ

KENT

G Tapp, 48 Mangravet Ave, Maidstone, Kent, ME15 9BG

WEST MIDLANDS

N Clemons, 45 Seagrave Road, Coventry, W Midlands

HEREFORD & WORCESTER

M A Morris, 32 Eastwood Drive, Kidderminster, Worcs, DY10 3AW.

GLOUCESTER

R Wright, 8 Linden Road, Gloucester, Glos, GL1 5HD.

EAST SUSSEX

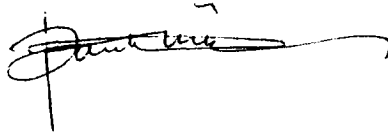
C Scally, 40 Barrhill Avenue, Patcham, Brighton, E Sussex.
Tel:- 02.735-03268

If you have any problems, ring your local county contact; or ring me on 01.640-7503 after 7 p.m. in the evening. I will try to help you.

Best wishes and God bless

Happy Christmas and a Good New Year

Paul Dicks

A handwritten signature in black ink, appearing to read 'Paul Dicks', with a long horizontal flourish extending to the right.

Published:- TIHOME, 157 Bishopsford Road, Morden, Surrey.

Printed:- SECRETARY BIRDS, 19 Coombe Road, New Malden, Surrey.