tidings

VOLUME 2, No. 4 AUGUST 1982

TIdingsTIdingsTIdingsTIdingsTIdingsTIdingsTIdingsTIdingsTIdingsTIdi
ingsTIdingsTIdingsTIdingsTIdingsTIdingsTIdingsTIdingsTIdingsTIding

TIHOME

tidings

VOLUME 2 , NUMBER 4 MORDEN SURREY 31st August 1982

TIHOME is not affiliated with Texas Instruments in any way and is supported only by its subscribers.

dingsTIdingsTIdingsTIdingsTIdingsTIdingsTIdingsTIdingsTIdingsTIdingsTIdingsTIdingsTI
ngsTIdingsTIdingsTIdingsTIdingsTIdingsTIdingsTIdingsTIdingsTIdingsTIdingsTIdi

Contents

EDITORIAL..EDITORIAL..EDITORIAL..EDITORIAL..EDITORIAL..EDITORIAL

First the good news, I have finally found someone to catalogue all the new programs for the software library. The bad news is that the job is not finished yet. So I am afraid that the new catalogue is not yet ready for distribution. However, as soon as it is ready it will be sent to you. I will not wait for the next newsletter to be ready if that means delaying the catalogue.

I attended the PCW show and found time to visit the TI stand. Their effort was most impressive. I think, though, that they learned quite a few lessons from the exhibition. I would have preferred to see the stand on the ground floor where there was more space, not up in the rabbit warren on the first floor. I kept wondering what would happen if some fool rang the fire bell. I was amazed to see how much the micro market has grown in the few years I have been involved with it. Home computers seem to be springing up like mushrooms. I saw the same thing happen with mini-computers, I also watched quite a few of them disappear within a very short time. It will be interesting to watch the path the micro market takes in the immediate future and see if there is a like pattern.

TIHOME is up to date with the pile of work that greeted us on our return from holiday, with one exception. A number of you are still waiting for listings. Do not panic, you haven't been forgotten. My first difficulty is that I am waiting for TI to send me some paper. Until then I can do nothing. The other problem is the length of time it takes to produce listings. The Thermal Printer is not exactly the fastest thing on four legs. I did hear a rumour that there might be a new printer coming out, but I shall have to wait and see. Those of you who have the TP will understand my problem when I say that the average order is for about 12 listings. In terms of time and paper 50p is really cheap. Perhaps too cheap! I am still working my way through an order for £45 of listings. God knows when I shall get it finished.

There is a possibility that your problems with the 99'er magazine could be solved soon. Certainly, more reliably and perhaps more cheaply. I can't say too much at the moment, but expect good news, soon.

There have been a great many requests for a childrens page in Tidings. I think this is a very good idea and so I am prepared to reserve a page for the use of children. By that, I mean I want to publish comments and, perhaps, small articles written <u>by</u> children for children. So, come on, all you young members and get the pens to paper. Don't be bashful, you know as much about this publishing business as I do. Believe me, I am not joking.

A word about letters. In the past, I have quoted odd snippets from members letters and many of you probably think that you said the same thing a lot better in one of your letters. In future I would prefer to publish letters where the writer has sent the letter specifically for publication. It would make my life a little easier and would mean that the letter writer knew, in advance, that the letter would be published. Could avoid some embarrassments! While I don't expect the feature to attain the status of a letter to the editor of the Times, it will be a letter to the editor of Tidings and that is of much more interest and importance to Texas Instruments.

And now, a cry from the heart. Jack McKillip carefully typed into his trusty 99 the program 'Mystery Words' from the 99'er magazine. He typed it in 'with the help of two pairs of reading specs (worn at the same time!) and a magnifying glass - God, that print is small'. When he ran the program he found it was a 'festering nest of bugs'. The first error was a BAD VALUE which he traced to line 4710 which appeared to read

```
   FOR MMM=1 TO 11
```

but 'under high magnification the digits of the number 11 don't look right. It turns out that this line should read'

```
   FOR MMM=1 TO M
```

The program then ran correctly until a BAD ARGUMENT in line 1200 turned up and this has our friend Jack stumped. So, you programming geniuses try this for size. The line reads as follows

```
CALL HCHAR(VAL(SEG$(D$,1,2)),VAL(SEG$(D$,3,2)),35)
```

I will publish any suggested solutions in the next issue of Tidings.

LETTERS TO THE EDITOR

In response to Pete's Soapbox, I thought I had better put pen to paper to avoid the ranks of the apathetic. I am fairly new to programming but after a little experience with ZX81, PET, ATOM and 380Z I think the TI gets top marks for ease of use, especially the handbooks and reference guide, following them is so simple. If only the modules were cheaper and ExBas was incorporated in.

For a bit of interest while waiting for a CALL KEY

```
100  DEF A=INT(5000*RND)+110
 .
800  CALL KEY(0,K,S)
810  CALL SOUND(-1000,A,0)
820  IF K=0 THEN [illegible]
830  GOTO 800
 .
```

Using more than one note gives very interesting patterns.

Ken Storry
Flitwick
Beds

From Arieh Yacobi

I have recently been making enquiries with some of the many mail order firms that advertise regularly in the American computer magazine "BYTE" for TI peripherals and software as Stateside prices seem much cheaper than the prices TI charge in this country. So far only one company, 47th. Street Photo Co. have replied and a copy of their price lists is attached. Although prices do seem lower, one has to bear in mind that postage and insurance (say about $25 for two modules) is on top and when the package arrives in this country duty at 5.3% plus VAT at 15% is charged on top. Nevertheless, this is still a good saving on what is usually charged in this country. However, one firm, The Computer Supermarket of Corby, Northamptonshire is offering discount prices on most TI lines - i.e. £75 for Extended Basic including VAT and postage - so perhaps it is worth trying them first as at these prices, it is hardly worth the trouble of importing.
Although it is an attractive machine with great potential, I often feel frustrated about its lack of speed - however, the accuracy of the machine helps make up for this drawback and I feel optimistic about the new Mini-Memory Module curing this very bad aspect of the TI 99/4A. Now if only Texas had seen fit to incorperate the functions of the M.M.M. into the machine originally, it might have won more friends.

Has anybody else noted some strange functions which are not mentioned in the handbook? One day whilst doing a rather complicated piece of programming (well, for me it was complicated!) I had to use the 'Insert' facility. I held down the Function and Insert keys and pressed the full stop (.) key, the symbol that needed inserting. Immediately the Master Screen came up and forty lines of programming were instantly erased. I was not amused.
One of the functions of the computer I have yet to get the hang of is the 'Control' key. Can anyone help? The only time I have generated symbols with it is when the computer has a program in its memory but is in the 'immediate mode' - i.e. not running. Some very pleasent designs can then be generated - but what can you do with them - and why do they only appear under these circumstances???? Personal answers to a new and bewildered programmer accepted! Otherwise, perhaps some wise elder of the TI system can give me the benefit of his knowledge in the next editition of TIdings.

Another remarkable experiance I had with the machine was after trying to modify the colors used in the 'YAHTZEE' program I obtained from the library. I noticed that despite the excellent quality of the program (see my comments on elsewhere) the numbering of the program lines seemed rather quirky. In my (usually) neat and tidy mind this did not seem to be right - so I tried Resequence the line numbers.The computer did not like this. It went all different colours (including some NOT mentioned in the TI handbook), displayed some rather nice designs and continued to repeat the whole process until I switched it off. The 'Break' and 'Quit' keys were of no effect, and nothing else I did could break the computer out of its awesome antics - WHAT HAPPENED??

Yours, Arieh Yacobi

Babbling Brooks

Introduction

Here we are again, and another change in format in both publication and BB, hopefully reflècting your wishes as expressed via the questionnaire. This issue sees the take-off of a couple of new subsections to BB: an introduction to BASIC and Programming, and a sort of Maths primer, which should help to dispel some of the difficulties which a large number of you have, it seems, been facing.

In the June issue, Stephen Shaw ran a post mortem on some of my feeble offerings, and I hope that you will follow his example now that he has set the ball rolling. And to add to the confusion, I can now tell you that there were one or two deliberate errors in last issue's main topic, to do with the algorithm, not the listing. The errors WERE deliberate, too.

Elsewhere there is a review of Version 110 Extended BASIC, and to add to that there is this little problem for you to solve (at least, those with EB): I ran a simple program which PEEKed the RAM in order to find out exactly where the program itself was situated. In order to make the program stand out, I made the first two lines REMs, filled with the letter A. I then spent 5 hours one evening examining every location I could, using a variety of different techniques - and finding the dreaded locations which PEEKing causes a total freeze-out or crash, probably because they represent the addresses which lead to the Extended BASIC module itself - and I couldn't find the program anywhere. I even tried filling most of the available RAM and then looking, and it didn't even look as though the RAM had been touched. Now I am aware that of the addresses I PEEKed some were simply repetitions of lower addresses: for example, I locked at addresses ranging from 32767 to -32768, but I did not have the 32K Expansion Memory, so really I was just looking (I assume) at the onboard 16K plus other areas; but that still doesn't explain why I couldn't find what I was looking for. Stephen Shaw's list of tokens in the last Tidings is going to be of some help, but I'm still stuck with explaining away the total absence of anything recognisable as my program. If anybody cares to hazard a guess, or has the answer (to life, the universe and everything, including TI's pricing policy), write to me at: 68, Kelburne Road, Cowley, Oxford OX4 3SH. I am currently trying to satisfy the Department of Transport that although I've been riding a motorcycle for 10 years I'm still not a complete hazard on the road, but in between riding down orange cones and Star Rider Instructors I'll be happy to answer any letters.

I've also been corresponding with a member of the Dutch User Group, who has come up with some ideas for speeding up the plotting subroutine given a while back, and who has managed to reduce the time for the Banthorpe 3D plot by almost ⅔! Although his algorithm is flawed, the basic premise is sound, and I have also used some of the ideas under EB (Extended BASIC) to speed things up just a little more.

His name is Bill van Kerkoerle and he has been a 99 owner for the past two years, only recently joining the Netherlands User Group. I hope to keep in regular contact both with him and with Paul Karis, the group's president, who, you may remember, was the one responsible for first bringing the additional CALLs available with the PRK and Statistics modules to light.

Bill's improvements are simple and straightforward. Although he originally used a method based on hexadecimal adding rather than substitution (wherein lay the flaw), it is the other changes which he made which speed things up considerably.

In the first place, Bill found that the image did not need to be so dense on the horizontal axis in order to produce an acceptable result. He therefore introduced a step of 2 into the control loop handling that axis. This simple change reduces the time taken by about half the original.

The second change made involves a principle which was presented in the Designs For Fun main topic last issue: that of producing mirror images. Bill reasoned that as the routine simply calculated the dot position for one side of the screen, and then just transposed the co-ordinates in order to make the plot on the opposite side of the screen, it might be faster to make the plot on one side of the screen only, and then at the end to run a loop checking the screen using GCHAR, and using the stored strings to produce a left-right reversal, and thus redefine further characters to make up the other side.

I changed the algorithm a little to make use of the fact that the screen row and column co-ordinates for a given character can be stored with the definition string for that character (except under Extended BASIC, but then that language can use CHARPAT), and thus remove the necessity for loops using GCHAR as a screen check. A further change I made was to restrict the range of characters which can be used for the first half of the plot to a maximum of 64 out of the 128 available, so that there are also 64 left to produce the second half. This also means that if there are insufficient characters available to produce the first half, then at least the gaps will be symmetrical!

I have enclosed a listing of the full program, including the plotting subroutine, elsewhere. Those members who are new to TIHOME and who are feeling intrepid or foolhardy enough might care to read issues 1 and 2 of Volume 2 of Tidings (back issues from Paul Dicks - check with him for prices).

An Introduction To BASIC

Although TI manuals have a reputation for being comprehensive and well-written, it would appear that many of you have been experiencing some difficulty in following the explanations given. It may be that this difficulty arises because the manuals assume some basic understanding on the part of the reader, or, more likely in my view, that the majority of readers are not used to reading what amount to reference works. There is a degree of skill involved in sifting and absorbing the small amount of information required from the mass presented; this problem has been highlighted recently [illegible] esponses given to the questionnaire, concerning the material involved in the main topics of Babbling Brooks. I have been told that some members are carefully putting aside BB until they have learnt enough about BASIC and programming in order to understand what the articles are about, which is the best way to handle information which you do not understand. My apologies for taking too much for granted; I've been doing the equivalent of expecting you to run before you can walk. At last we are getting the feedback which lets us know at what level to pitch the articles which we present.

In the light of this feedback I am starting a 'Beginners BASIC' which will hopefully fill in the gaps; perhaps presenting the same information in several different ways will enable many more of you to join the 'priesthood' of those who have begun to master the black art of programming.

And if you think that the biggest barrier to your understanding is my explanation, then write in and say so!

Back in the early days of computing, to program a computer you really did need to be something of a brainbox. The early machines were programmed in binary, by setting banks of switches; a far cry from today's sophisticated machines, some of which allow you to program verbally, by accepting speech input. Someone had a brainwave, and made life a little easier, by allowing programmers to enter their programs using either hexadecimal or octal code (base 16 or base 8) and simple keyboards. Later the so-called 'high-level' languages were developed, and are still being developed. These allowed programmers to enter what amount to English words, and one of the early languages was FORTRAN (Formula Translator), a scientific language, from which BASIC (Beginners All-purpose Symbolic Instruction Code) was developed in 1957 at Dartmouth College in the States. BASIC was apparently intended originally to help computer science undergraduates understand the principles of programming; a kind of Janet and John language which could be used to build up to the more complex languages used. But BASIC proved to be so popular, precisely because it is so easy to learn and use, that it is still the predominant language around today, despite the fact that much more pwerful languages have since been developed. In other words, the majority of people have not got any further than Janet and John in their reading, which is a source of irritation to all academics and some professionals. This is why you will see articles in the popular computing press which decry the amateur programmer and his BASIC, and why you won't stand much chance of getting a job in the computer industry if you only have BASIC under your belt.

To the average amateur, however, BASIC doesn't seem much like a simple language, until they've seen programs written in Assembly Language or Pascal, or even Forth.

One hurdle I found was that I couldn't grasp how the computer actually understood BASIC if, as the books all say, it only understands machine code (which can mean either binary, hexadecimal, octal, or even something called microcode!). Even now I have only an inkling, but it helps tremendously if you have a little understanding about how the computer sees things.

To begin with, from the moment we switch on our 99s, programs are running. There is a program which prepares and produces the screen display for example. There is one which continually checks the keyboard to see which keys are being pressed. There are others which actually execute the BASIC programs which you have written, and yet others which handle the flow of information between computer and peripherals.

Unfortunately, very few details are available about the internal workings of the 99s, so one can only speculate about what goes on inside. You can get an inkling of the volume of work being done by the machine if, after selecting TI BASIC, you turn up the volume until you can hear the cursor blinking on and off. If you then press a key like Shift for example, you can hear the tone change, as the computer responds. Enter a valid instruction like CALL CLEAR for example (and make sure it IS correct or you'll blow your socks off with a deafening error tone!), and you can hear the machine at work. So when you enter something (by pressing ENTER, or one of the other entry keys - shift E or shift X on the 4), the computer runs a little program which compares what you have entered with a list it has of things that it can understand; if it finds what you've entered doesn't appear in its list, it jumps to another program which deals with errors; otherwise it will jump to whatever routine or routines it needs to execute in order to carry out the commands which you have issued. When you enter a program, it runs a general check to see if what you are entering is allowed by the rules which have been laid down by the designer. Try entering 100 RUN and see if it is accepted. TI BASIC won't allow

RUN to be used as a statement in a program.

There are at least two aspects to writing any program. One is the language used - in this case BASIC, which will be looked at in greater detail later. The other is probably the most difficult thing to understand, and that is what it actually means to 'program'. I am handicapped by the fact that it has been five years since I first learned to program, and I find it very difficult to remember what it was like before then. I have yet to find the simplest way of getting the computer novice to realise that he or she has been programming for years without ever being conscious of it. If you've ever written a shopping list, you've written part of a program; every waking hour is filled with little programs which you have learned over the years; ones which make you get dressed BEFORE going to work, which make you open your mouth BEFORE shovelling food in, which make you open a door BEFORE trying to go through it. In theory, anyway. I've tried giving examples which make people think: my favourites are the smoking and shopping variety. You either write down the sequence of actions involved in lighting up, or in sending the kids to the shops for a tin of beans. The trouble is that while these illustrate the principles of programming, they have no relevance to BASIC, unless you happen to have a robot tacked onto your 99. The above examples are concerned more with the control of things, whereas BASIC is more concerned with processing data.

A better example then might be if I was to ask you to do some arithmetic for me. If I asked you to add four numbers together, how would you set about it ? You might ask for the first number, and write it down. You would then ask for the second number, and write that down under the first, and so on. Once you had the four numbers, you'd add them up, writing the answer underneath. You'd probably then tell me what it was.

There are three separate actions being performed here. The first is INPUT. The second is PROCESSING DATA. The third is OUTPUT. Things are not always so cut and dried, as you will come to find out; sometimes processing halts for further input, sometimes there is no output, sometimes no input as such. But generally programs can be divided up into those three sections. The sequence has to be right, too, before the program will work properly. You couldn't give me the answer BEFORE I'd given you the numbers, for example. The INPUT section is fairly obvious; it consists of you prompting me for the numbers, and me giving them to you. The PROCESSING is also fairly obvious - it consists of you doing the adding up; and the OUTPUT consists of you telling me what the answer is.

Not all processing involves arithmetic, however, at least not in the sense implied above. Suppose I had asked you to sort some words into alphabetical order instead.

Again there would be an INPUT section, where you prompted me for the words, perhaps writing them down on individual cards. The PROCESSING this time would involve shuffling the cards around according to a set of rules (the algorithm) until they were sorted. The OUTPUT might consist of you reading the cards in sequence to me, or storing them away somewhere without reading them out at all.

In theory, what you are supposed to do is to sit down and specify the problem that you want to solve, or the game that you want to play. You then write down as many of the steps involved as you can, elaborating where necessary. You go over this many times, trying to iron out errors, making sure that you've covered everything (which of course you never have), until you are satisfied that you have done all you need to. You can then begin to write the program in the language you have chosen.

What usually happens though is that you sit down at the console with a rough idea of what you want to do, and you begin writing, testing each little bit as you add it, until you have a program which works. Until you present it with something which it can't handle, in which case you now have to go right back to the beginning in order to find out what went wrong, and by now you've forgotten exactly why you did certain things, and you still keep getting INCORRECT STATEMENT IN 460, although 460 looks OK to you... At this point you wish you'd kept notes on what you have been doing, and the three scribbles on the back of an old beermat don't mean a thing three weeks later, and you might threw the whole thing over and take up fishing. Nice quiet hobby, fishing.

If you've been bitten by the programming bug though, you'll be back, having been struck by a bolt of wisdom on platform 3 in Bolton, breezing into the house, commandeering the telly, and three hours later you're looking out the perch pole again.

In time, if you are lucky, you will come to develop a programming style which suits you. If you are really lucky, it will be one which allows you to produce a program with the minimum of hair-tearing, but don't expect this stage to be reached for quite a long time, unless you are undergoing tuition. From an academic point of view, there is a right and a wrong way to program. From a commercial point of view, as Paul Dicks tells me, it doesn't matter how you program as long as the damned thing works!

This series will hopefully combine both examples of programming and explanations of BASIC so that the two proceed side by side. That's the intention, anyway.

A Re-Introduction To Mathematics

If some members are experiencing a little difficulty with the mathematical functions available on the 99s, this series may offer a little enlightenment. Many of the functions will be familiar to those who have occasion to use them at work, and to those who are still at, or who have just left, school. But for those who last locked at logarithms at school twenty years ago, and probably didn't understand them even then, some of the functions will be a closed book. Even after this series it may still be a closed book, but don't give up. If there is something which you haven't grasped, and would like explained in more detail, write in to the address given in Babbling Brooks, and if necessary, your anonymity will be preserved.

It seems that no matter where you start in this game, you always end up having to explain half a dozen simple things in order to explain another supposedly simple thing! That being the case, we'll start right at square one with the four basic mathematical operators: that way we shouldn't leave anybody behind.

In order to avoid confusion, computerists (good word, that) don't use the standard symbols for dividing and multiplying. Instead we use '/' for divide, and '*' for multiply. You may see why in some computer publications where the printers keep putting '-' instead of the division sign, and which is the function in YXXY ? Y*XY is much more legible. Alright, so I exaggerated, but it happens, very often.

There is, as usual, a set of jargon to go with the symbols. The ones we will use most frequently are those which describe the result of a particular operation. For example, the result of adding two numbers together is called the SUM, while the result of multiplying two numbers together is called the PRODUCT. The result of a division is known as a QUOTIENT and of subtraction is a DIFFERENCE. So the product of X and Y is the same as X*Y, and the quotient of A and R is A/R. There are a couple of other functions, known as the RECIPROCAL and INVOLUTION. The reciprocal of a number is the result of dividing 1 by that number. The reciprocal of 2 is therefore 0.5 or ½. Involution is slightly difficult if you've never come across it before. It involves 'raising' numbers to 'powers' of other numbers. For example, 3 raised to the power 3 is written 3^3, and is equivalent to 3*3*3. In other words, it is the number multiplied by itself as many times as is given by the number which appears to its top right. 5^2 is thus 5*5 : we usually say it five squared in this case; 7^3 is 7*7*7 or seven cubed; 8^4 is 8*8*8*8 and is said to be eight to the power four. It is a simple way of writing very large numbers so that they don't take up much space: compare 10^9 with 1,000,000,000 for example.

One of the things about writing numbers in this way is that certain mathematical functions become very easy. To multiply 7^2 by 7^7 for example doesn't mear that you have to write it out as 7*7 multiplied by 7*7*7*7*7*7*7. All you need to do if the first numbers are equal is to add the powers. $7^2 * 7^7$ is $7^{(2+7)} = 7^9$. And you can hold the number like that during a calculation until it needs to be worked out, instead of grappling with things like 49*823543. It applies to algebraic symbols as well: $A^5 * A^3$ is A^8. It is also possible to divide numbers as well. $7^6 / 7^4$ is 7^2, achieved by subtracting the second power from the first, but only if the two numbers which are being raised are the same. There are situations which will take a little more explaining later, like what is the result of $7^4 / 7^6$, which according to the rules set out would give 7^{-2}, only how do you calculate it ?

If we work with 10 and raise it to different powers, you may be able to see a pattern emerging. Look at the table below and see if you can spot it.

$$\begin{aligned}
10^5 &= 100,000\\
10^4 &= 10,000\\
10^3 &= 1,000\\
10^2 &= 100\\
10^1 &= 10\\
10^0 &= 1\\
10^{-1} &= 0.1\\
10^{-2} &= 0.01\\
10^{-3} &= 0.001
\end{aligned}$$

In each case, subtracting one from the power is equivalent to dividing by 10, which is the number to which we are raising different powers. In fact, the power,(called also the EXPONENT,)if negative, is the same as the reciprocal of the number to the same positive power. That is, 10^{-3} is the same as $1 / 10^3$, and 10^{-1} is the same as $1 / 10^1$. (You don't usually see 10 written as 10^1; we usually drop the 1). Going back to that 7^{-2}, it can be calculated as $1 / 7^2 = 1/49 = 0.02040816326$. (For you numerologists, that fraction has an interesting pattern in it as well. Can you see it ?)

If we were to write out a set of tables similar to that above, for every number between say 1 and 9, you would find that one result is always the same, and that is any number raised to the power of zero. The result is ALWAYS 1. In any equation then, if you come across A^0 in the course of calculation, you can replace it with 1. What would be the result of $A^x + B^x + C^x$ if x was zero ? (Anybody gets anything other than 3, see me afterwards!)

This particular function forms the basis of LOGARITHMS, which we will cover next time.

There are other things which can be done with powers. For example, dividing a power by 2 is equivalent to finding the square root of the number. (A square root of a number is a value which, when multiplied by itself, yields the number. That is, the square root of 4 is 2, as 2*2=4. The square root of 9 is 3, and 3*3=9. Put it another way, the square root of 2^2 is 2, of 3^2 is 3, of 4^4 is 4^2 or 16, and of 5^8 is 5^4. Work it out.) Dividing by 3 finds the cube root of a number. The cube root of 2^6 is 2^2. That is, the cube root of 64 is 4. 4*4*4=64. Divide by Y and you find the Yth root of a number. The fifth root of 7^{10} is 7^2. And so on.

By now you may have noticed something. What happens if you try to take the square root of 5^3 (125) ? According to the rules, dividing the power by 2 will give $5^{1.5}$. And how do we calculate that ? You could guess that the square root of 125 lies somewhere between 11 (11*11=121) and 12 (12*12=144), and it will be closer to 11. So far, we have worked with only integer values as exponents, but it is possible to have fractional values as well. That $5^{1.5}$ lies between 5^1(or 5), and 5^2(or 25). In fact, to solve the problem we will have to resort to logarithms. And those will be covered next time.

Three - Dimensional Noughts & Crosses (3DOXO)

Playing ordinary Noughts & Crosses doesn't demand a great deal of concentration or intelligence. My young nephew has been playing since he was 4 (although he cheats!). Programming a computer to play the game is a fairly hefty challenge, and it is usually one of the first games that the novice programmer finds himself caught up in.

There is a three-dimensional version of the game (available commercially as a stack of four plastic trays with different coloured pins, so that more than two players can engage at once) which is not only more stimulating to try and program, but is also more of a challenge to play, so that the interest generated can extend beyond the stage of programming alone. I find that generally I cease playing a game once I have got the program working satisfactorily unless there is more to be learned about the actual strategy of the game itself (and hence subsequently improvements to the program as well).

This BB main topic will probably extend over two or more issues as we explore both the principles of playing and of programming this game. The program will be written in TI BASIC in order to appeal to those without access to Extended Basic or PRE/STATS Basic, and should be available now from the library if you don't want to spend a week keying it all in.

There are a number of complex programming techniques which will be explained in detail, so treat the article as you would a reference work; keep coming back to it, and you will find that bit by bit it will provide all the answers to your questions (says he modestly!).

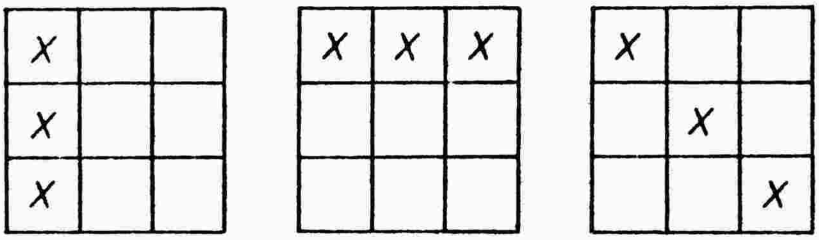
The Board(s)

You might expect 3DOXO to be simply a 3D version of the standard Noughts & Crosses board, but you'd be wrong. As we shall see, if a 3 x 3 x 3 'board' is used, the first player to move can control the game, either winning or forcing a draw, but never losing (unless he/she plays execrably!). The 4 x 4 x 4 board provides a more complex, less predictable game, although it reduces the category of move to either 'good' or 'bad', instead of giving a range of categories as 3 x 3 x 3 does; but I am jumping ahead.

The playing area in a 4 x 4 x 4 game comprises 4 boards consisting of 4 x 4 squares. Each board is termed a 'LEVEL', and each square on each board can be referenced by using its ROW and COLUMN number with its LEVEL. Because of display constraints, the final version of the program will not present the levels one on top of the other, as in some of the diagrams, but side by side. This orientation does not take long to become adjusted to.

There are 64 positions or squares on the 4 x 4 x 4 board (4 x 4 x 4 = 64), running from Row 1, Column 1, in Level 1, through to Row 4, Column 4, Level 4. There are also a number of what are called 'WINNING LINES'. These are groups of four positions which constitute winning lines: i.e., if you occupy, or your opponent occupies, all the positions in a given winning line, you, or he/she/a TI 99, will win.

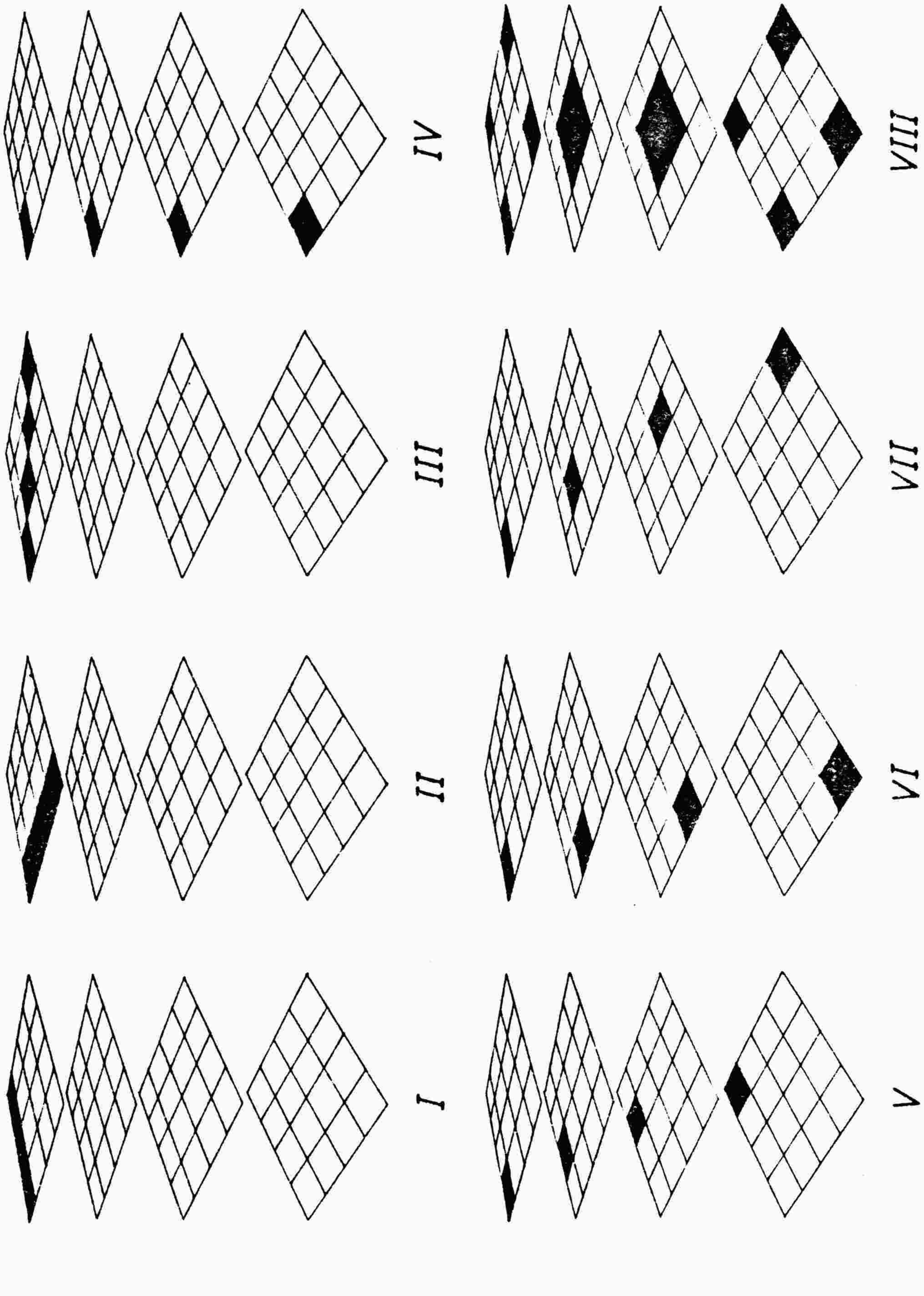
On a normal 3 x 3 2D Noughts & Crosses board, the winning lines are easy to visualise. They run vertically, horizontally, or diagonally. There are 3 vertical lines, 3 horizontal, and two diagonal, making a total of 8 winning lines. The diagram below shows the three types.



However, when the board is extended to three dimensions, the number of winning lines increases markedly. Not only are there the three types of winning line illustrated above, but there are others running down through the levels. (See Figure 1). To make drawing easier, the positions in each winning line have been filled in, rather than being represented by a nought or a cross. Illustrations I, II, and III in Figure 1 show the three straightforward winning lines we have already discussed. Illustration IV shows how a winning line can extend vertically through the levels; V and VI show how it can also extend through the levels while crossing rows or columns; and VII shows it extending from one upper corner diagonally down through to the opposing lower corner.

Now to the numbers of these winning lines. Of the type shown in I there are 4 per level, making 16. Of the type in II there are also 4 per level, making 16. Of the type in III there are 2 per level, making 8. When it comes to type IV, there are 16, one vertical line of four for every square in the top level. Type V are a little more difficult, but I hope that you can see that it is of the diagonal type, so that it runs in pairs across the levels (not through them) giving 4 pairs = 8. The same applies to type VI, which also is of the diagonal type, giving 4 pairs = 8. Type VII are a special case. No matter how many squares you expand your board to be, there are

Figure 1



always only four of this type in a 3D system. Adding up the totals then gives: Type I : 16; II : 16; III : 8; IV : 16; V : 8; VI : 8; VII : 4; with a grand total of 76 winning lines.

There are two equations which allow you to calculate the total number of positions, and the total number of winning lines, for a given 3D board of N x N x N squares. The total positions value is easy: N^3 (anyone who doesn't know what that means, see the new series on a Re-introduction to Mathematics). The total winning lines value is obtained from $3*N^2 + 6*N + 4$. Here are a few examples of the total position and total winning lines values:

N	Positions	Winning lines
3	27	49
4	64	76
5	125	109
6	216	148
7	343	193
8	512	244
9	729	301
10	1000	364

Note how the relationship between total positions and total winning lines changes. Up to N = 4, there are always more winning lines than positions. After that, the number of positions increases by a larger factor than the number of winning lines; in other words, boards with N = 5 or more will not necessarily make for a better game. (Incidentally, you can also extend the table in the other direction and find the values for N = 2 or even N = 1!).

If you don't want to play a game to the bitter end; i.e., when all the positions are filled; then a simple rule is to declare the game a draw when all the winning lines have been either filled or blocked, needless to say, without a win having occurred! There isn't much point continuing to play a game if there are no more possible winning lines.

The next question to ask is: Which are the best positions to move into ?

A Good Position

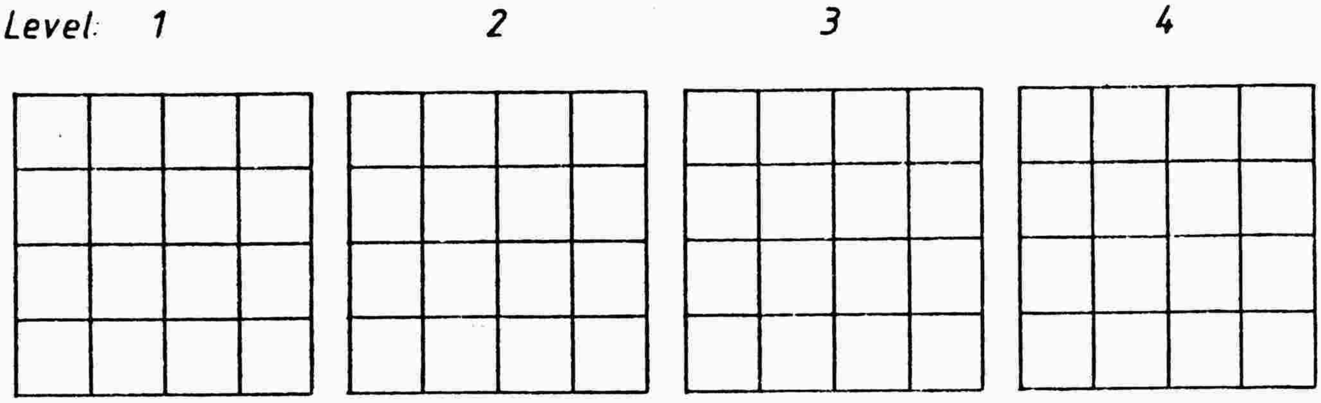
In a 2I Noughts & Crosses board of 3 x 3 squares, the best position into which to try and move first is the centre position. This one controls a large number of all the possible winning lines. You can get an idea of which positions are reasonably powerful by 'mapping out' each position in terms of its 'linkage' with other winning lines. Doing that for the 3 x 3 board gives:

3	2	3
2	4	2
3	2	3

The centre position is a component in 4 winning lines; both diagonals, one vertical, and one horizontal line, all run through the central square. I have coined the term 'linkage' to refer to this value ('componency' is probably the more strictly accurate). The four corner positions each have a linkage of 3 - one vertical, one horizontal, and one diagonal. The central square on each edge has only a linkage of 2. It appears, therefore, that if you were formulating a strategy for your own moves, you might come up with something like this: Move into the central position if it is empty; if not, then move into a corner position if one is empty; if not, then move into a middle edge position if one is empty; if not, then all the available positions have been filled, and the game is a draw.

Now that strategy, apart from applying to 2D Noughts & Crosses, is very, VERY far from being complete. As we delve further into the game, I think you will begin to see why.

Below is the layout as it will appear on screen of the 4 x 4 x 4 board which we will be using. See if you can evaluate all the positions and assign each one a linkage value. What do you notice about this board's range of linkage values ? Now you can see what item VIII shows! I'll give the answers in the next issue.



VERSION 110 EXTENDED BASIC : A REVIEW

In an earlier issue of Tidings I gave a cursory review of Version 100 Extended Basic. This latest version has corrected some of the bugs in the original, and appears to have some very powerful command variations which I don't remember seeing either in the early version, or for that matter, anywhere else. A little extra thought seems to have gone into this one, but alas it is still the old story of three steps forward, two steps back. I don't know what the current price for this module is, price cuts being showered upon us, but I think that £45 would be a serious suggestion, TI Invaders having been 'rationalised' in price. The trouble is, I suspect that it is selling (if at all) for £80!

However, my general conclusion is that it is worth having, although that doesn't stop me wishing forlornly that some of the commands had been a) provided with TI Basic, and b) made available without the necessity for that over-priced 32K dinosaur, the Expansion Memory. Can somebody please explain why 32K costs us £300, but costs most other owners £50 or less ?

The packaging for this module is different from the usual, necessitated by the size of the manual I suspect. This particular (American) unit came with two additions: a small booklet detailing such items as the differences between the 4 and 4A character sets and control codes and keys, details on the 'pre-scan' controls, even how to produce the second (4A) small upper-case character set for the 4 from the standard set. There was also a note on program files, which if I have understood it correctly, means that you cannot SAVE a program which is larger than about 12K. I hope to follow that one up. In addition, there were notes about the Expansion Memory, and the usual Errata - 1½ sides plus a separate sheet of changed error messages. The paperwork also included probably the best item in TI's documentation, the quick reference cards. I don't know whether this was planned, but the cards for my TI are white, while the cards for the module are grey, so I can't confuse the two. The first thing I do with these is to cut them up along the fold lines and encase them in a sealed plastic sheath, so that my sticky fingers don't tear or muck them up, and I can dip them in my coffee with impunity.

One of the first things I checked was the effect of Extended Basic (EB) on my thermal printer, as version 100 upset it considerably. To my surprise, not only did my TP like this new module, but a bug which I had been given to understand was based in the printer hardware, has disappeared; the one that turns ON...GOTO or ON...GOSUB and even occasionally IF...THEN...ELSE line numbers into gibberish, but only when they are listed to the printer. Well done, TI.

It was nice to see the auto-repeat on all keys (and for 4A owners it must be nice to have it back, as version 100 took it away from you!), and it doesn't take long to adjust to the new, black square, cursor. I still tend to tap the keys for most things rather than let the auto-repeat take the strain, as a) it has become almost instinctive, and b) for most purposes it is faster: there is a second delay before the repeat comes into play. I'm not knocking it, though!

On power-up, the EB apparently looks to see if you have a disk plugged in and switched on. If you have, it then looks for a program (written by you) called LOAD. If it finds it, it loads the program and runs it. This means that you can create your own title page with options etc., or as I believe one member does, you can deliver instructions about the programs which are present on that disk.

I think my biggest complaint has to be about the reliance of the manual on punctuation. In the format details on each command, a missed comma or a misplaced bracket can spell disaster. Some of the errata listed were of this type; it has made me particularly conscious of punctuation in TI's manuals, and as a result I have noticed things which I might otherwise have missed. I have two years of coping with the 99/4 under my belt, but a newcomer might not immediately realise the significance of commas, colons, etc., TI's note about EB really being for the experienced TI Basic programmer notwithstanding.

That apart, the handbook has a reasonable layout, and it is fairly easy to find details on a particular command or subprogram, even more so since the alphabetical indexing has been rationalised so that you can look for the CALLs under their subprogram name rather than thumbing through pages of CALLs.

For those contemplating an EB purchase a full list of the reserved words with a cursory explanation is given later. Some specific improvements (or lack of them) deserve special mention, though.

EB permits multiple statements on a line, each statement being separated from its predecessor by a double colon. This in itself, although making listed programs difficult to read, reduces space and increases execution speed. TI have gone further and allowed certain statements to become multiple-assignment; for example, if you want to set variables A, B, and YY to 7 you simply create a line with A,B,YY = 7 and that's something I have never seen before. A little insight into the internal operation of the 99s is given here: if you use I,A(I) = 5' for example, EB looks for a current value of I and assigns that element of the A array the value 5, and THEN assigns 5 to I. It seems to work from the inside out so to speak. If you wanted to assign I and A(5) the value 5, you'd have to write it thus: A(I),I = 5.

Some of the other statements are like that: CALL CHAR for example has a double dose, one of which caught me out for a while. If you specify a definition string for a character which consists of more than 16 hex digits (as in the example in EB of plotting, where two extra digits tacked onto the end recorded the screen location of the character), the next available character is selected and redefined with the remaining digits, and in this way you can specify a 64 digit definition string with a starting ASCII code, and define four consecutive characters with one command. As if that wasn't enough, although any digits in excess of 64 are ignored, you can stick a comma after the first definition sequence, and specify another ASCII code and another definition string, and so on! Use string variables to hold the definition strings, and you could redefine the entire alphabet with one command, if that is your wont.

A command is also provided, CHARPAT (no. CHARpat!), which allows you to obtain the full 16 digit definition string (with necessary trailing zeroes) of a given ASCII character. This is one that ought to have appeared in TI Basic.

A word about the pre-scan controls. A long program involves the system searching for some time until it has found all the variables and compiled a variable list, and also found all the CALLs and set aside memory to hold any parameters, and it often means a sizeable wait before a program begins running. With pre-scan controls, !@P+ and !@P- (pronounced shriek at P plus or minus) you can present the system with all the basic information it needs using a little trick so that the pre-scan can be switched off, thus saving time at the start of a program. As this item was covered in an additional booklet, I'm left wondering if any of the version 100 EBs have these controls available. There is a simple test. Type in and run the following program:

```
100  GOTO 110 :: !@P-
110  A$ = "THIS BASIC DOESN'T HAVE IT" :: PRINT A$ :: GOTO 110
```

If your EB does have these commands the program will not run; it will generate an error, because the system has been told that there are no variables, but it has encountered A$ in line 110, which it cannot understand, as with pre-scan off it cannot incorporate fresh variables into the variable list. These controls need

a little care in their use, but they are a useful addition to the range of instructi
If anyone finds that they have pre-scan controls but don't know how to use them, write to me at 68, Kelburne Road, Cowley, Oxford OX4 3SH, enclosing a stamped, self-addressed envelope, and I will send you details. Alternatively, write to TI direct.

The anomaly with SIZE noted in version 100 has been resolved - when using the A = A + 8 dodge against SIZE two different values were obtained, and the difference between them was much larger than normal error - so that standardly there are 13928 bytes of RAM available with no program present.

The IF..THEN..ELSE conditional branch statement has been enhanced so that not only can it be used with GOSUB as well as GOTO, but other commands may be used, subject to certain limitations (a FOR..TO..STEP..NEXT loop is not allowed, for example), and even multiple statements can be used between the THEN and the ELSE, which makes for quite an enhancement.

When it comes to I/O commands like SAVE, OLD, etc., there is the usual nasty disappointment as you read about all these juicy functions and then find that they are restricted to disk use. MERGE is one of these, and so is the ability to SAVE and OLD named programs. Only two likely additions to cassette I/O are given: one automatically begs criticism - the PROTECTED option. The blurb talks of protecting your valuable program from listing, editing, SAVEing, etc., and this is one that is causing some disturbance among the profes ionals, as each time someone stands up and says 'I can protect your programs from being copied', someone else works until they can overcome the protection. You might think this a little naughty, but the computer industry needs to know if its protection facilities can be violated, because if an 'honest' programmer can violate it, so can a 'dishonest' one.
For that reason I'm going to put my neck on the block and state that I broke the PROTECTION on a short cassette-based program with hardly any effort, using two different techniques. When I say hardly any effort, that's what I mean. Within 5 minutes of SAVEing the original under PROTECTED format, I had obtained either a full listing on screen without crashing, or a partial listing which could still have been used to deduce the overall structure. The first method is not very reproducible (that means I can't do it right every time), but the second is, and although I don't find it a matter for extreme concern, some of you might.

Either way, it just goes to show that you can virtually guarantee that anything anybody says can't be done, can - eventually. Please note, TI, I'm being a good little boy and not telling how I did it, only warning that it can be done, so fellow TIHOMErs don't get caught out.

The second addition, as far as cassettes are concerned, is the ability to OLD a program and have it RUN automatically once loaded. Typing RUN "CS1" for example will give you the CS1 OLDing instructions, and once loaded correctly, you get the usual DATA OK...PRESS ENTER, and when you press ENTER, the program runs. You can use this in programs themselves, so you could have successive chunks of program call each other in sequence, although I haven't tried this, but I suspect that variables will not be passed from one program to the next, so its use will probably be limited. Visions of vast Adventure programs recede into the distance...

I'm a little disappointed (again) in the 'machine code' instructions, CALLs PEEK, LOAD, LINK, and INIT, as only CALL PEEK works without the Expansion Memory, and although I have looked at all the locations possible, I have yet to find out whether PEEK <u>without</u> Expansion Memory can actually look at any program at all! It differs from the usual PEEK command found on other machines in that again you specify a start address, and then assign the contents of that address, and those of subsequent addresses if needed, to one or a series of variables: e.g., CALL PEEK(127, A, B, C, D) will assign the contents of address 127 to A, of 128 to B, of 129 to C, and of 130 to D. As the manual says, without the Expansion Memory CALL PEEK is of little use!

The CHARSET command resets all redefinable (note: NOT user-definable) characters to their default settings; it would have been nice if this could have been made more selective, enabling specific characters to be returned to default status.

The SPRITE facility is probably the best provision in EB, although Version 100 Sprites and their associated commands didn't always function as expected. You pay for your sprites by losing the ability to redefine ASCII characters 144 to 159, although you can redefine colours for the cursor and 'edge character' as well as for the reduced sets, but that doesn't really compensate for it. The sprites are reasonably versatile, and in fact can move with dizzying speed, and I found the provision of 28 to be more than sufficient. Another manufacturer's machine is rumoured to provide 256 of them; I can't think what for! The sprites are appearing on more and more machines now, but I think that TI can justifiably claim the distinction of being first. Commands like COINC can detect a coincidence within set limits if necessary, between any two sprites, or a sprite and a screen location (256 x 256 dot-positions, some of which lie off-screen, making it 192 x 256), or any sprite and any other sprite. It would have been nice if we could have had a form of ON COINC, so that instead of checking every so often for a coincidence (and probably missing it!), we could have had an interrupt caused by a coincidence, with associated servicing routine(s). Still, mustn't grumble. It is a very good function for games, and enables you to give that polished 'module' look to your Alien Mothers-in-law. (I must get round to writing that one soon!)

I don't know what useful purpose the DISTANCE subprogram serves as it stands at present, for it takes the row and column distances between two specified sprites, squares them, adds them together, and if the final value is greater than 32767, returns 32767 instead! Also the ERR subprogram: if the error severity (!) is ALWAYS 9, why include it ?

The ability to DISPLAY USING IMAGE is very clever, enabling all sorts of things to be done, including, as the manual example does, present standardised letters using word-processing-type controls. All we need now is the word-processor...

EB seems just as slow as it ever was when executing subroutine calls; I timed my A = A + 8 routine in TI Basic and EB, and EB was 1½ times slower. I wonder why ?

The LIST command has been enhanced, in that having begun listing, pressing any key will stop the process, and pressing any key will continue it, unless of course you press CLEAR (Shift C on the 4), in which case it won't.

As usual, VAL is still the same old cripple it was in TI Basic. Instead of being the inverse of STR$, the way it is on most of the other machines I've seen, so that you can obtain VAL on "C + COS(J) * L / 2.25 - ATN(1) * 4", it still limits you to string versions of valid numbers. Ah, well.

It's also a shame that you can't include SIZE in a program and warn users about the amounts of memory remaining (SIZE is used in commands like ACCEPT), although RUN can be used as a statement; or do I mean command ?

Two of the really good bits I have saved till last. The first is the Boolean algebra operators: AND, NOT, OR, and XOR. Not only can you use these in relational expressions (currently doubled in TI Basic by *, -1-, +, and -) BUT... provided the numbers being used lie between -32768 and +32767, you can assign the decimal result of the operation to a variable! So A = 8 OR 6 will set A to 14. So what, I hear some of you ask. 8 and 6 IS 14. Big deal. Not so, not so. What has happened is that the binary representations of the decimal numbers 6 and 8 have been ORed inclusively, and the binary result converted back into decimal. A = 8 OR 8 is thus not 16, but 8. The reason I have a touch of the ailing super-lunary talking birds, (over the moon, and sick as a parrot, Brian), is that this is precisely the function that my high-resolution plotting subroutine needed, and didn't have. I like it, anyway.

Alright, maybe you'll like the second good bit. You can create your own subprograms (CALLs to you and me), and if you have disks you can stick 'em onto one, and call

2021 note- Peter is presumably referring to SAVE DSK1.SUBS,MERGE - the only related item limited to disk use.

'em up at will. But not if you have cassettes, which is BORING. (Astute readers might by now have guessed that I do not possess a disk system - and won't, at these prices, - hence the soft-pedal around the disk commands: what I don't have, I can't test!)

With your own CALLs, you could create baffling procedures such as CALL HELP, or even CALL ARMPIT, although I doubt if many will use that one. On a slightly serious note, it might be an advantage if we began a library of user-defined CALLs which EB owners have developed for specific purposes, in order to try and a) prevent everyone from re-inventing the wheel, and b) create a little order among the confusion so that possibly a standard set of CALLs could be set up. It would improve the debugging of a program if CALL HORSE was a standardly-accepted way of filling the screen with carets. Horse, carets, get it ? Oh, never mind.

It's an idea anyway, and I would welcome some feedback. We might end up with a section like the Subset in Personal Computer World.

An interesting thing about the SUB facility is that although you can pass parameters from the main, or calling, routine to a subprogram, the variables used in it, if not parameters, are 'local' to that subprogram. That is to say, you can use variable BB in the main routine, and BB in the subprogram, but they are both distinct and separate variables with differing values. On top of this, you can, it seems, specify that a parameter that is passed to a subprogram can revert to its original value if altered by the subprogram.

The full list of commands, statements, and functions follows, with comments only where the item is either new, or an improved version of one available in TI Basic.

ABS

ACCEPT (AT) (VALIDATE) (BEEP) (ERASE ALL) (SIZE) more powerful version of INPUT, accepting data at any point on the screen, checking it for validity (UALPHA, DIGIT, NUMERIC, or STRING EXPRESSION), sounding a prompting tone, clearing the whole screen first, limiting the size of the input field, and either clearing it first, or leaving it undisturbed. Works with input of one variable only, and without an input prompt.

ASC

ATN

BREAK

BYE

CALL you can define your own subprograms with optional parameter list.

CALL CHAR changed so that strings in excess of 16 digits up to 64 digits may be used to redefine more than one character in a sequence.

2021 note: Like many users, Peter failed to note the ellipses ([...]) in the command heading in the manual which indicated for example you could now use CALL CHR$(42,"1234",47,"73581")- which was not possible in TI Basic.

CALL CHARPAT allows you to extract the current definition string for a specified character and assign it to a string variable.

CALL CHARSET causes the standard character set (ASCII 32 to 95 inclusive) to be reset to default definitions.

CHR$

CALL CLEAR

CLOSE (DELETE)

CALL COINC detects the coincidence within tolerance limits if specified of any two specified Sprites, or Sprite and screen location, or between any two Sprites. It may miss coincidences if either the sprite speed is very high, or it is called after a coincidence.

CALL COLOR can also specify a foreground colour for a specified Sprite (the background is always transparent, it seems).

CONTINUE, CON

COS

DATA

DEF

DELETE

CALL DELSPRITE (ALL) removes either a specified Sprite, or all sprites, from the screen.

DIM up to seven dimensions this time, instead of the usual three.

DISPLAY (AT) (BEEP) (ERASE ALL) (SIZE) a more powerful version of screen PRINT or DISPLAY, (see ACCEPT) with the additional use of USING and IMAGE which format the data to be output.

CALL DISTANCE gives the sum of the squares of the row and column (dot coordinates) distances, if less than 32767, between any two specified sprites, or a Sprite and a screen location.

END

EOF

CALL ERR returns the error code, error type, and optionally the error severity and line number, of the most recently uncleared error.

EXP

FOR TO STEP

CALL GCHAR

GOSUB

GOTO

CALL HCHAR

IF THEN ELSE improved so that statements in addition to line numbers (or instead of them) can be used, and is now nestable (IF...THEN...ELSE IF...THEN...ELSE...) but there are certain limitations on what may appear within it.

IMAGE specifies the format in which data is PRINTed or DISPLAYed when the USING clause is present. The line number of this statement is called by the appropriate command.

CALL INIT prepares the computer to load and run assembly language programs, but only if you've got the Expansion Memory.

INPUT

INT

CALL JOYST

CALL KEY

LEN

LET although it is rare to see LET in TI programs, or in many others for that matter, TI have increased the power by permitting a single statement to assign one value to several different variables.

CALL LINK Passes control to an assembly language subprogram, but only if you've got the Expansion Memory.

LINPUT (REC) different version of INPUT which doesn't edit what you type, so if you put in commas without quotes, or odd numbers of quotes, or leading or trailing spaces, it all goes in. Can also be used with files.

LIST enhanced so that pressing a key during listing suspends the process, giving you time to examine sections of it, when pressing a key again continues it. The 'escape' is BREAK (shift C, sometimes called CLEAR). Function 4 to you 4A owners.

CALL LOAD loads an assembly language subprogram, but all together now only if you've got the Expansion Memory.

CALL LOCATE repositions a given sprite on the screen.

LOG

CALL MAGNIFY sets the size and magnification of all the sprites.

MAX returns the larger of two specified numeric expressions.

MERGE merges lines from the specified device-filename (!) with lines already in the computer.

MIN returns the smaller of two specified numeric expressions.

CALL MOTION changes the row and column velocity of a given Sprite.

NEW

NEXT

NUMBER, NUM

OLD

ON BREAK (STOP) (NEXT) allows you to control what happens when a program BREAKs; you can even have it ignore the BREAK. The only trouble is that unless you've made other provisions, you have to SHIFT Q (Function =) to get out of the program, which means that you lose it as well.

ON ERROR determines the action if an error occurs. You can either STOP, or give a line number to which control can be transferred in the event of an error. See also RETURN.

ON GOSUB

ON GOTO

ON WARNING determines the action if a warning condition occurs, permitting you to either stop, or print a message, or ignore it and continue.

OPEN

OPTION BASE

CALL PATTERN changes the pattern of a given Sprite or Sprites to specified character value(s), which means that if you want to change shape rapidly, simply define a set of characters to the shapes you are likely to want, and swap the definition of the Sprite around.

CALL PEEK is virtually useless without shhhh...you know what.

PI an additional function previously obtained by 4 * ATN(1) in TI Basic.

POS

CALL POSITION Returns the dot row and column position of specified sprites.

PRINT (USING) enhanced PRINT capability by allowing formatting of both text and numbers.

RANDOMIZE

READ

REC

REM In addition, the exclamation mark, or tail mark, shout, shriek, or whatever, can also be used as a comment indicator. Note that on some machines that mark is used instead of the PRINT command.

RESEQUENCE, RES

RESTORE

RETURN Can be present with either a line number or NEXT when used in conjunction with ON ERROR. The line number specifies the point to which control should be passed after the error routine, while NEXT transfers control to the statement (not necessarily the line number) following the one which caused the error. If RETURN is used without anything following it, in conjunction with ON ERROR, the statement which caused the error is executed again.

RND

RPT$ will repeat a specified string a specified number of times. In some instances is worth its weight in gold.

RUN has been enhanced so that you can place it in programs, and also specify a device and in the case of disks, a program name, so that a form of chaining is possible.

SAVE (PROTECTED) (MERGE) enhanced so that you can give limited protection to your cassette-based software, and/or save routines for later merging with your disk-based software. Incidentally, Stephen Shaw tells me that using the MERGE option with disks, it is possible to resequence the program line index so that it is properly sorted - that's the index used internally by the machine to keep track of your program lines as they are entered.

CALL SAY this one comes from the Speech Editor Module and requires the Speech synthesizer for it to work. It will pronounce words taken from the vocabulary of around 373 words and phrases (includes a large number of duplicates though, like 2, two, to, and too are all the same sound). Version 100 was apparently very slow at saying things, but as my speech synthesizer is currently out on loan, I haven't checked on this; it is only of limited use, anyway. The Terminal Emulator II has it beaten hands down with regard to speech.

CALL SCREEN

SEG$

SGN

SIN

SIZE used either in Immediate mode, where it displays the amount of free space in bytes, and with the Expansion Memory will also give the size of Stack available, or see DISPLAY and ACCEPT.

CALL SOUND

CALL SPGET also from the Speech Editor, this command retrieves the string of speech codes which correspond to the word specified. It is possible to create a limited number of 'new' words by concatenating groups of speech codes, but you need a bit of care to use it gainfully.

CALL SPRITE allows you to create up to 28 sprites, specifying the ASCII characters to be used, the foreground colour, and the start screen location (by dots). You can also optionally set them moving as well.

SQR

STOP

STR$

SUB indicates the beginning of the subprogram name with an optional parameter list. The restriction with SUB is that it (or they) must be placed at the very end of a listing, and cannot be followed by anything other than other SUBs or REMarks, or END.

SUBEND indicates the end of a subprogram.

SUBEXIT allows control to be passed to the statement following the CALL, before SUBEND is encountered.

TAB

TAN

TRACE

UNBREAK

UNTRACE

VAL

CALL VCHAR

CALL VERSION   returns a value indicating the version of Basic being used. EB returns a value of 110, unless you have version 100.

!@P +/-   pre-scan controls. See text.

The manual also gives a number of appendices, listing illustrative programs; all commands, statements, and functions; ASCII codes; Musical tone frequencies; character sets; 'pattern-identifier conversion table' - how to define characters, in other words; colour codes; colour combinations; split console keyboard; character codes for split keyboard; mathematical functions - a very useful one, as it shows how you can obtain things like ASN, ACS, etc., using the standard functions, although I wish they'd been given as standard; a list of the speech vocabulary; a list of suffixes to be added to the speech codes; and a list of Error Messages. There is also a comprehensive index at the back.

SOAPBOX

Soapbox hasn't really been going long enough for you, the avid reader (!), to have submitted anything, so once again I'm going to get lathered up (SOAPbox, get it ?) about nothing in particular. Actually, what I'm going to do is to explain away some of the anomalies in Tidings (I hope), if I can.

For example, I'm in the process of putting the final touches to the latest BB which will go out with an August publication date, although I'm late - today's the 2nd of August, but Paul is on holiday so the newsletter will undoubtedly be late coming out. I've just received the June issue of Tidings and have been devouring it this morning during my coffee break. The equally voracious reader will have noticed certain contradictions throughout the newsletter. I refer in particular to a couple: firstly, I stated that the TEII could NOT speak listings. At the time of writing, to the best of my knowledge that was true; however, Mike O'Regan's friend, Les Ball, had contacted Stephen Fung who in turn had contacted me, to tell me that although the TEII manuals say nowt about it, the TEII CAN speak listings, BUT, and here's the catch, it CAN'T speak ALL of the listing. Any punctuation marks which correspond to 'pause' commands are not spoken (could be a problem in PRINT, INPUT, DISPLAY, etc.) and there may be other minor difficulties. Apparently all that TEII owners need to do is to LIST to the Speech Synthesiser as if it were a peripheral like the Thermal Printer. Under TEII, speech is output by opening a file to the synthesiser and then PRINTing to it. I do not have the exact commands to hand at present, but the format will be:

```
LIST "speech file instructions" : line numbers if required.
```

The second inconsistency arises over module prices. Robin Frowd (nice to hear from TI at last) in his page said that the new low-level language modules, Mini-Memory and Assembler/Editor, would cost around £90 (is that with or without VAT - it wasn't clear), while Stephen Shaw and I were quoting £115, and Stephen has also seen one for £104. (Interesting that £90 + 15% VAT = around £104.) When you come across such conflicting pieces of information, DON'T PANIC! (It seems I have heard that before somewhere...). Things will eventually sort themselves out;in the meantime, please bear with us.

As for the problems with my typewriter ribbon, all I can say is that photocopies made at my end, while not spectacular, were more readable than those in Tidings. If you look at the listings for that issue, which were made on the printer and which are usually crisp and clear, they, too, suffered from over-heavy contrast. I have put a fresh ribbon in the machine, but they seem to wear out very quickly. Mind you, I'm churning out something like 20,000 words an issue, so maybe that's not surprising. Either way, my apologies if my indistinct print made understanding my indistinct explanations even more difficult!

Another Soapbox special: a few days ago I received some unsolicited mail from a firm who sell computer cassettes. The envelope displayed a sticker with the words "CARE: RECORDED MATERIAL" only there was no recorded material, just a blank C10 cassette. Now I'm not averse to free gifts (I hope it IS free, otherwise they're going to come unstuck if they try and bill me for it), but only when I request them. The interesting thing here though was the way they had mis-spelled my address. From time to time, when I'm feeling really nasty and I'm filling in one of those order forms, I deliberately mis-spell my details so that I can trace any unsolicited mail back to source. These days, most companies take it upon themselves to pass your details around among each other, as they have found that the cost of posting things to you is usually more than paid for by the response. There are even companies who SPECIALISE in passing your details around for a price. Some ask you if you would mind them doing this, but only one so far has had the decency to ask me. The rest assume you don't mind, and some will not even remove your details when you demand that they do so.

It was considerably disconcerting therefore to discover that I had not made this particular mis-spelling, so it was done by the company. But where did they get my details ? Lo and behold, on the envelope which brought Tidings, was the SAME mis-spelled address! Now,I've met Paul Dicks, and I reckon he's not the type of bloke to go bandying other people's details about to commercial concerns. So who has been spreading OUR mailing list about ? Is it TI ? They supplied the first two copies of Tidings this year. Is it the printing firm with whom Paul has been arranging future publication of Tidings ?

Paul is now on holiday, so I can't discuss it with him, but I will once he returns. Unfortunately I can't include his comments with this article, but he will probably put something in the editorial. If YOU don't want your name and address shuttled around other companies' computer databases, let it be known in no uncertain terms.

On the subject of cassettes, I placed a joint order with a friend for some (well, he placed the order, I just handed over the folding stuff) with a firm called DATA-ASSETTE. That was some months ago and we are still waiting. If you decide to order such things, make sure that you specify a delivery date, and stipulate that unless you receive the items by that date, the contract (for that is what it is) will be broken, and you will expect your money back IN FULL. If the company play rough, take them to the SMALL CLAIMS COURT; it's usually fairly straightforward, if somewhat slow. Don't forget to let US know which company or companies cause you trouble:bad publicity ALWAYS helps!

SCIENCE WATCH

Nuclear Magnetic Resonance Spectroscopy

In an earlier issue of Tidings I listed a series of topics which I hoped to cover in the future. Among the more fascinating was Nuclear Magnetic Resonance (NMR) and its application to the field of medicine. To this end I recently paid a visit to Oxford University's Department of Biochemistry in the company of orthopaedic surgeon Mr Raymond Newman. Ray is conducting research with NMR into the biochemistry underlying those conditions in which the patient has muscle weakness (dystrophy).

The NMR lab is situated within the Radcliffe Infirmary, close to the centre of Oxford. In common with all laboratories in which research into the more sensational aspects of science is performed, the NMR suite is modest, quiet, and unobtrusive. Every day it is passed by countless numbers of people who have no idea of the advanced technology being used only a few yards away.

The outer room houses the usual paraphernalia found in research labs: filing cabinets, paper-littered desks, the essential kettle and cups, and in this case a Research Machines 380Z with twin 8" floppies and an impact dot-matrix printer. No 99-style colour graphics and text-to-speech here! The 380Z though is a baby compared with the set-up in the next room. On one side of a seven-foot high glass and aluminium safety partition stands a space-age console with twin monitors,(one grey, one green), twin disks (again, 8"), a colourful QWERTY flush-mounted keyboard (with separate numeric keypad, TI please note), and a Hewlett-Packard A4 flatbed plotter, also flush-mounted. On the other side of the partition lies what looks like a huge steel bobbin on its side - the 4.28T superconducting magnet. It is about four feet in diameter and is raised off the ground so that the bore running through the centre is at about waist height. There are a couple of ice-encrusted pipes emerging from the upper side of this over-sized cotton reel, but otherwise it looks as innocuous as any other piece of garden furniture. The pipes carry the Helium and Nitrogen gases away as they boil off from the -270°C and -196°C liquids inside.

On the occasion of my visit the unit were examining the arm of a patient with dystrophy who was lying supine on a trolley placed next to the spectrometer. Her arm was positioned completely within the magnet. It appears that the human body can act like a radio aerial, detecting stray signals which in this case would interfere with the very sensitive measurements being made. The arm therefore has to be 'isolated' from the rest of the body by wrapping an 'earth' consisting of a strip of ordinary kitchen foil coated with special conductive jelly (usually used to improve electrode contact during ECGs) around the upper part of the arm.

Within the bore of the magnet, the arm is restrained gently but firmly so that it cannot move freely and interfere with the scanning. An aluminium former is fitted, and the interesting thing here is that although aluminium is not magnetic, when it is fitted it has to be firmly pushed through the magnetic field, which causes eddy currents to be set up within the metal, causing it to act like a magnet. Once it stops moving, the eddy currents cease, and it stops behaving like a magnet.

A small coil of two or three turns of wire is used as the detector. It is placed on the arm over a particular muscle (in this case the flexor digitorum superficialis - somewhere in the forearm). The size, shape, and number of turns of the coil determine the volume that is examined. It is 'tuned' after the patient's arm is inserted, to home in on certain emissions from the nuclei in the muscle; in this case the researchers were interested in both hydrogen and phosphorus - from which the study takes its name: Phosphorus NMR.

Now we come to the sticky bit: explaining the principles of operation. The last issue of Tidings briefly touched on Superconductivity, and it rears its ugly head here as well. If you remember, it is a quality associated, at present, with very low temperatures, whereby resistance to the passage of electrical current becomes zero. This particular phenomenon is used here to enable magnets with very strong fields to be built; in fact they are so strong that special care has to be taken not to take certain objects near them - watches for example, and floppy disks can be erased just by sitting around five feet away, as Ray found to his consternation a little while back!

Having got our immensely powerful magnet, then what ? Well, if you remember your balmy schooldays spent in the idyllic surroundings of the chemistry lab, you probably won't remember the atomic structure of molecules. Suffice it to say that the nucleus spins, wobbling a little on its axis within the magnet as it does so. If one starts exciting the structure by causing it to absorb energy at the same frequency, the wobble increases until the atom flips over. Each nucleus has a tiny magnetic field associated with it, and when the atom flips over, so does its field. Either you can measure the amount of energy given out by the atom as it eventually flips back, or you can measure the time that it takes to flip back - the 'relaxation time' - a matter of tiny fractions of a second.

There are a large number of different chemicals in the human body, so what the researchers do is to decide on a particular one. They then tune the coil appropriately so that they are only detecting that particular chemical nucleus. The readings taken subsequently are processed by the computer over a number of scans, the results being averaged, to produce a spectrum which indicates the concentration of the particular compound. This concentration takes a short time to measure, and represents a considerable advance over what are known as 'invasive' procedures, not least because the muscle under study can be measured both at work and at rest. The whole operation takes about thirty to forty minutes from beginning to end, and all the patient has to do is lie there and doze. There is no pain, and the only thing one is left with is a sticky arm!

The particular application described here does not process the results to provide an image of the tissue under examination, but other research groups are involved in that side of things (Nottingham, Aberdeen, and London), and there are promising signs for the future.

A follow-up to the items on superconductivity has come from Scientific American. It seems that researchers in America have discovered that certain organic crystals can 'superconduct', and they are currently engineering them with a view to obtaining superconductivity at room temperature. Look out for some startling advances when that happens, not least of which will be the realisation of Star Trek's total body chemistry analyser - in your own hand!

I would like to thank Ray Newman both for the opportunity of seeing this application and for his constructive criticism during the preparation of this article. I would also like to thank the staff of the NMR lab for allowing me to watch them at work.

References: Mr R. J. Newman, BSc, MBChB, FRCS, FRCS Ed.
Electronics & Computing Monthly, Vol. 2, Issue 4, p71 (1982)
Scientific American, Vol. 247, No. 1, pp 50 - 59, Bechgaard and Jerome, "Organic Superconductors"

Postscript

In an earlier issue of Random Dots, you may remember Mike O'Regan commenting on the number of characters sets which the 99s appeared to possess. Although the H/ and V/ CHAR commands will allow you to specify ASCII codes up to 32767, in fact the 99s will reduce whatever code you specify if it exceeds 255. If you specify 256 as the code, you will in fact get 0; 257 will give you 1, 258 gives 2, and so on, in multiples of 256. You can check this by calling CALL HCHAR(1, 1, 256, 768), and then CALL GCHAR(1, 1, H). When you print H you will find it is 0.

A number of members, in addition to yours truly, have discovered that you can get more than 4 lines to a statement in a program using TI BASIC. For example, if you enter the line number and then, without adding an intervening space, set up your program line, and enter it, when you LIST that line it will (provided you filled all the available positions) have been extended to occupy 5 lines, which can, under most circumstances, be added to, and may even allow a sixth line to be created. Alternatively, just editing an existing line often allows you to run onto a fifth program line.

The reason for this is most probably the 'tokenisation' of many of the reserved words in the line (see Stephen Shaw's very helpful items on tokens in the last issue) so that the internal use of memory is actually less than the screen would lead you to believe.

Tidings has now been going for long enough that the most useful hints and tips could stand being repeated all together. I've just run through from issue 1 and found loads, and I'm now debating whether to produce a small booklet with them all in, or to put them into a bumper issue of BB in a couple of months time. Watch this space...

A little question which someone might like to answer: if 99/4 owners can use ENTER, shift E, and shift X to enter data during INPUT, can 99/4A owners use ENTER, function E, and function X, to the same effect ?

That just about wraps it up for this issue. If you have any comments, they will be much appreciated, whatever your age or sex, and all will be treated seriously.

Pete Brooks

Speedy 3D Plotting - after Bill van Kerkoerle

A little explanation to help you along:

The subroutine begins at 120 and ends at 320. The REMs which appear in earlier versions have been removed to save space. The initialisation is split into two parts: line 100, which dimensions the array (see Guidelines For Software Contributors, last issue), and lines 330 to 390 which set up the screen format and colour and create the constants. Note that HR$ is specific to this program only. I came across information similar to that given by Stephen Shaw about X*X being faster than X∧2 and have changed lines accordingly.

The variables R and C are used to pass the plot coordinates to the subroutine.

The subroutine variable list is: B, C, H, I, P, R, S, X, Y, B$, H$, I$, Z$, C$(), and of those, S, B$, H$, Z$, and C$() should not be modified by the main routine(s), although C$() may be under certain circumstances.

Observant plotting-addicts may notice that HR$ is similar to the mirror-image reversal string used in Designs For Fun.

Lines 610 onwards perform the mirror-imaging upon C$(). Refer to Issues 1 and 2 of Volume 2 of Tidings for further explanation.

```
100 DIM C$(128)
110 GOTO 330
120 Y=INT(R/8+.875)
130 X=INT(C/8+.875)
140 CALL GCHAR(Y,X,H)
150 IF H>31 THEN 220
160 IF S=95 THEN 320
170 S=S+1
180 C$(S-31)=Z$&CHR$(Y)&CHR$(X)
190 CALL CHAR(S,Z$)
200 CALL HCHAR(Y,X,S)
210 H=S
220 H=H-31
230 B=C-X*8+8
240 P=2*R-16*Y+16+(B<5)
250 IF B<5 THEN 270
260 B=B-4
270 I$=SEG$(B$,POS(H$,SEG$(C$(H)
,P,1),1),4)
280 I$=SEG$(I$,1,B-1)&"1"&SEG$(I
$,B+1,4-B)
290 I=POS(B$,I$,1)
300 C$(H)=SEG$(C$(H),1,P-1)&SEG$
(H$,I,1)&SEG$(C$(H),P+1,16-P)
310 CALL CHAR(H+31,C$(H))
320 RETURN
330 CALL SCREEN(8)
340 S=31
350 CALL HCHAR(1,1,S,768)
360 B$="0000.0001.0010.0011.0100
.0101.0110.0111.1000.1001.1010.1
011.1100.1101.1110.1111"
370 H$="0....1....2....3....4...
.5....6....7....8....9....A....B
....C....D....E....F"
380 HR$="0....8....4....C....2..
..A....6....E....1....9....5....
D....3....B....7....F"
390 Z$="0000000000000000"
400 H1=192
410 V=104
420 X1=H1/2
430 X2=X1*X1
440 Y1=V/2
450 Y2=V/4
460 FOR X5=0 TO X1 STEP 2
470 X4=X5*X5
480 M=-Y1
490 A=SQR(X2-X4)
500 FOR I1=-A TO A STEP V/10
510 R1=SQR(X4+I1*I1)/X1
520 F=(R1-1)*SIN(R1*12)
530 R=INT(I1/5+F*Y2)
540 IF R<=M THEN 590
550 M=R
560 R=Y1-R
570 C=X1-X5+32
580 GOSUB 120
590 NEXT I1
600 NEXT X5
610 H=S-31
620 FOR K=1 TO H
630 D$=C$(K)
640 Y=ASC(SEG$(D$,17,1))
650 X=33-ASC(SEG$(D$,18,1))
660 FOR L=2 TO 16 STEP 2
670 I$=SEG$(D$,L,1)
680 J$=SEG$(D$,L-1,1)
690 I$=SEG$(HR$,POS(H$,I$,1),1)
700 J$=SEG$(HR$,POS(H$,J$,1),1)
710 C$(K+H)=C$(K+H)&I$&J$
720 NEXT L
730 CALL CHAR(K+S,C$(K+H))
740 CALL HCHAR(Y,X,K+S)
750 NEXT K
760 GOTO 760
```

TI NEWS

Well, I think I have recovered from the rigors, feet-ache and reduction in my word store as a result of the PCW Show held at the Barbican Centre on 9th-12th September. The Exhibition was a resounding success and I understand that approximately six times more people came to this years show as last year, and I am convinced I talked to each individual visitor. I met a number of the members of TIHOME during the Exhibition including those well known contributors to TIHOME, Peter Brooks, Mike O'Regan and Stephen Shaw, as well as a gentleman by the name of Paul Dicks.

The reaction of the visitors to the Texas Instruments stand was very good and we had to get one of our London dealers to sell off the stand due to the demand. We had many compliments about the TI stand and at many times during the Exhibition they were queing four deep to get at the computers. Consequently we will be at the show next year, but with a much bigger stand.

For us, one of the biggest bonuses from the Exhibition was the many contacts we made with 3rd Party Software companies who were eager to write and market software for the TI-99/4A. All these contacts will be followed up and you can look forward to many additional sources to purchase software of all types. It was surprising how many of these companies had not contacted us because they felt awkward in talking to a very large multinational corporation. Well we really are friendly people and want to talk to anyone who has really good ideas and even better software packages.

I have had some compliments recently about availability of TI Software. Its the old story about the local stockist not having the particular package you require. We are currently working on a Nationwide mail order set up to allow people to order any currently marketed TI Software Package from a source which stocks the whole range.

I had intended to put in this particular edition a section on the answers to your problems, compliments etc., which were highlighted in the last issue of TIHOME. Unfortunately your normally very well organised chairman forgot to send me a copy. He has promised to rectify this so I will try to answer those questions in the next exciting edition. I am aware of one problem, however, concerning Terminal Emulator II and Text to Speech. We have been asked if the computer will "speak" a program listing, it will and the input to achieve this is;

```
LIST "SPEECH"
```

Remember a space is need between LIST and the open quote marks. This particular command is an example of how subtle we at Texas Instruments can be - we did not put it in the manual.

Minimemory, Logo, Editor/Assembler and the Peripheral Expansion system will all be available this month, the TI-99/4A and Software are all now freely available and next month comes some new games like Alpiner, Othello, and Chisholm, as well as UCSD PASCAL.

Sales of the TI-99/4A are going very well and extra distribution is being added all the time.

Finally I know that some of you have been having problems getting copies of the 99'er Magazine, well I think we have got it cracked now so watch this space next issue!

ROBIN FROWD
Product Manager
TEXAS INSTRUMENTS LTD

Handling PRK files with Enhanced BASIC

by Franc Grootjen and Paul Karis
Blauwgras 2
3902 AA VEENENDAAL -
The Netherlands

Owners of the Personal Record Keeping (PRK-) module will have discovered the many advantages of this useful preprogrammed file handling system. Some may have even heard about an interaction with TI BASIC to format an INPUT or PRINT anywhere on the screen with CALL A and CALL D respectively, aptly christened Enhanced BASIC (See 99-er volume 1, no 4 page 72)

Few will have heard of the further features of Enhanced BASIC enabling the user:

a. to change any characteristic of once chosen pages without losing all the entered data.
b. to perform mutations upon PRK-data which are not possible with the PRK options.
c. to use two independant memory systems side by side and in conjunction.
d. to perform string operations on PRK-data.
e. to save long arrays quickly without problems.

I am driving at the use of the statements:
CALL P
CALL L
CALL S
CALL G
CALL H

CALL P(p)

In order to prepare or partition the RAM memory space in two sections the statement CALL P(p) has to be entered in the immediate mode as a first action after the TI BASIC READY prompt from the computer and must be followed by NEW. In this article we will refer to MEM1 (= normal RAM) and MEM2 (=PRK file memory). In the position of p in CALL P(p) a number (constant) must be used (an integer between -4 and 13821) which points to the spot in the RAM stack where OLD will start loading instead of what otherwise would have been the starting address reserved for OLD. The memory space below this point is the thus created MEM2 (see figure 1) and consists of p bytes.

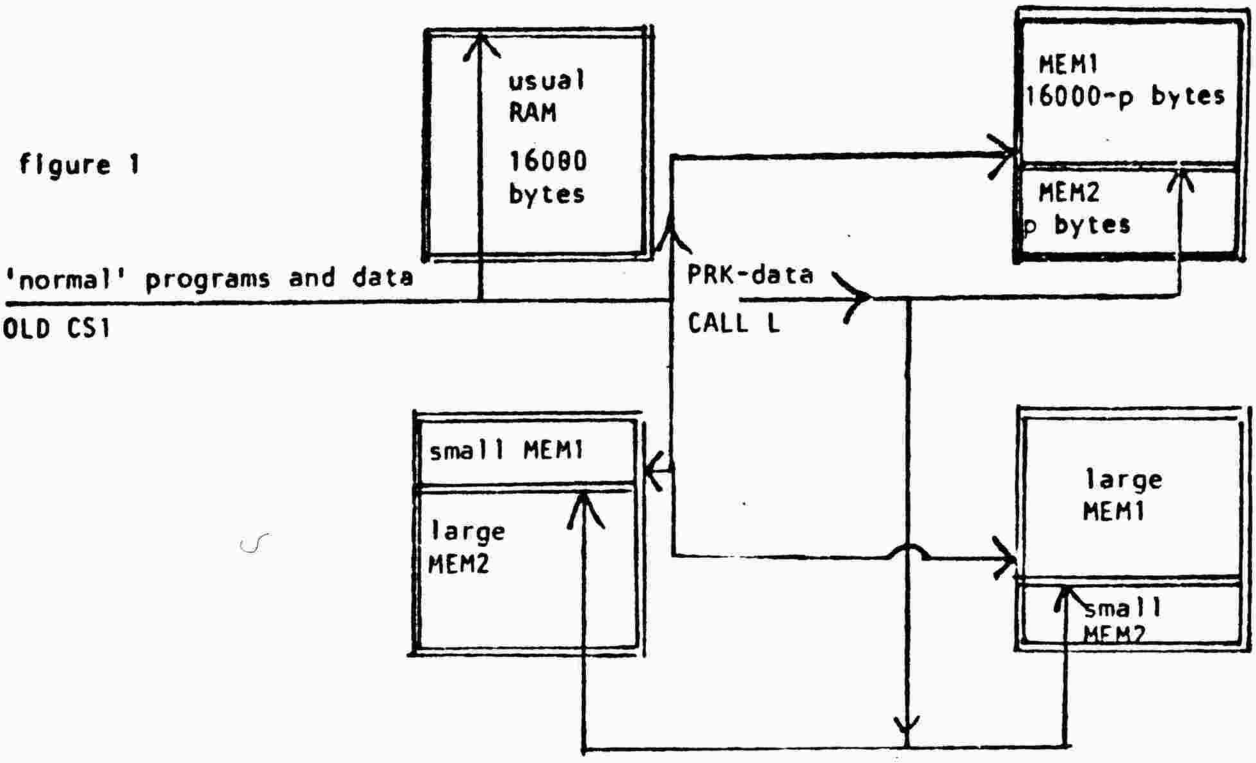


figure 1

figure 2a

figure 2b

If you need a large MEM2 (e.g. because a large PRK file has to be loaded) then p must be in the higher range. (figure 2a) This obviously at the same time restricts the size of MEM1.

If on the other hand you have a large TI BASIC program or need to load a lot of 'normal' data, but still you want to interact with a PRK file as well, then you are best off with a large MEM1 and a small MEM2 (figure 2b). Then in the CALL P(p) statement you assign a low value to p.

If you are still with us then it will dawn upon you that this beautiful mechanism enables you to freely choose the ratio of MEM1 against MEM2 to suit any particular application. MEM1 is cleaned and loaded with programs and data in the usual way with NEW, OLD, SAVE, INPUT and PRINT without affecting MEM2 in any way. PRK-data are loaded in MEM2 with the statement:

CALL L("CS1",C) resulting in the usual cassette prompts which will now read data previously generated by the PRK module in its normal mode.

An error message: ERROR DETECTED IN DATA probably means that MEM2 is not large enough to contain the PRK-data and a new CALL P(p) with a higher p must be entered (and followed by NEW). The quickest and surest way to remove the previous CALL P is to switch the console off and on again.

MEM1 can be used:
a. normally as if MEM2 had not been prepared
b. In the direct mode interacting with the loaded PRK-data
c. to load a program in TI BASIC that interacts with PRK-data or even loads or saves PRK-data when this program is RUN, as program lines with CALL L and CALL S (you guessed it, S for save) are permitted.

The syntax and parameters of CALL L and CALL S are:

CALL L("CS1",C)   and   CALL S("CS1",C)

C is a control variable that gets a value 0 if loading or saving has been successful. When an error has been detected the value of C in not equal to 0.

Instead of "CS1" also "DSK1" or "RS232" may be used for similar interaction with these devices.

There are only 2 statements that can be used in the TI BASIC (or rather Enhanced BASIC, a term coined by Peter Brooks of the British Users group) dominating MEM1 that cross the borderline between MEM1 and MEM2. These two statements are:

CALL G   and   CALL H

The CALL G (G for getput) statement has a write mode and a read mode. In the write mode it enables the user to change PRK-data, which might also have been changed using the normal procedures of the PRK module (change a page and mathematical transformations).

But the values can also be taken from variables which in turn get their values from the whole bag of tricks of TI BASIC including string handling (which otherwise lacks in the module). Needless to say this dramatically enhances the versatile use of the PRK-data, the more since the saved data (CALL S) can be reloaded in the module mode, permitting their further use 'module fashion'.

The syntax and parameters of CALL G are:

Read mode: CALL G (1,PAGE,ITEM,MIS,VALUE)

Write mode:CALL G (0,PAGE,ITEM,VALUE)

MIS is a control variable: if data found then MIS=0. When no data found MIS=1.

Note 1: If a string variable is taken for VALUE (e.g. VALUE$) and the ITEM type was chosen as 'characters', then strings can be handled
Note 2: CALL G(2, PAGE,ITEM,VALUE) writes 'nothing' (=missing data) in the assigned place. The value of VALUE has no meaning.
Note 3: Although not mentioned each time, these techniques also work with the Statistics module.

CALL H effects the header, i.e. those choices that the PRK does not permit you to alter, once you have okayed the file structure.

Changing the names of FILES and ITEMS is a cinch.
Changing the type or width if ITEMS is also easy, but the results are usually not the ones aimed for, because all data are written sequentially as beads glued on their string. Nevertheless it is possible to change any entry for the file structure maintaining the right data in their correct place, but this requires routing through a special TI BASIC program. This will be discussed in a future article. For those of you too impatient for this feature we recommend you buy the SSS Command module 'Personal Record Generator'. Apart from then disposing of several features described in this article, you also overcome the frustration of having spent days and days entering data in a PRK programme and then coming to the conclusion that a small change in file structure is necessary forcing you to start from scratch.

The syntax and parameters of CALL H are:

Read mode: CALL H(1,INFO,ITEM,VALUE)

Write mode: CALL H(0,INFO,ITEM,VALUE)

INFO	Stands for
1	File name (maximum of 9 characters)
2	Day of the month
3	Month
4	Year
5 *	Number of items per page
6 *	Number of pages
7 *	Length of header in bytes
8 *	Length of each page in bytes
9	Item name (maximum of 9 characters)
10	Type of item (1=char; 2=integers; 3= decimals; 4= scientific)
11	Width of item
12	Number of decimal places
13 *	Amount of memory required for this item in bytes
14 *	Position of item in page

* = this value is automatically calculated and placed by PRK mode

The ITEM value is of no importance for those INFO values between 0 and 9

Setting up the header (i.e. file structure) is easiest in the module mode (Let's begin-create a file), but if you insist on doing this from TI BASIC start with INFO 9 to 12 before going on to INFO 1-4.

We intend to give examples of the practical use of all this knowledge in future articles covering such subjects as an easier way to save arrays and word processing in PRK-files.

Although Texas Intruments are obviously aware of these possibilities they have not considered them perfect enough to document them and 'sell' them officially. The authors declare that although inside information from TI made this article possible, that in no way can Texas Instruments be held responsible for its contents, nor are the authors who present these data in good faith but without any warranty whatsoever.

Still we consider knowledge of ENHANCED BASIC important for a complete enjoyment of your TI99/4(A).

And that is what counts !

ENHANCED BASIC for efficient loading/saving of arrays

by Franc Grootjen and Paul Karis
Blauwgras 2
3902 AA VEENENDAAL
The Netherlands

This article will only make sense to those who have familiarised themselves with ENHANCED BASIC the term Pete Brooks of the British users group invented for the extra's provided by CALL A/ CALL D/ CALL P/ CALL L/ CALL S/ CALL G/ CALL H in combination with a Personal Record Keeping or Statistics Module.

Imagine having completed 150 observations with values between 0-100 and that you want to save these values on tape . You don't want to use the PRK or STAT mode as the standard possibilities of these modules are too restricted for your intentions. TI BASIC allows us to OPEN and PRINT :# n and so writing data on tape, but this method has several drawbacks;

- saving and loading consume a lot of time and tape. Frequently the pilot tone alone, marking the start of each record, takes longer than the actual time required for data transmission. Waiting time is wasted time.
- the compacting of data that sometimes partly solves the first problem in itself is complicated . (E.g. the string "2/2.3/5/6.7/3.4/1.3/3/34/123/4.5" can be written as one variable but used to transmit 10 numerical values)
- there is no built-in data check possibility like we have for program saving.
- when an error occurs the program BREAKs and cannot be CONtinued, and the data loaded cannot be used and must be loaded again. The same holds true if a small alteration needs to be made in the program after the data are loaded.

ENHANCED BASIC does away with all these problems.

Back to our example: Let us presume that the 150 observations are held in the array O(I). At this point we would like to remind you that the use of ENHANCED BASIC requires you to prepare your computer with CALL P before typing in or loading the following subroutines or programmes containing these subroutines.

```
10000 REM SAVE SUBR
10010 CALL H(0,5,1,1)
10020 CALL H(0,10,1,2)
10030 CALL H(0,11,1,2)
10040 FOR I=1 TO 150
10050 CALL G(0,I,1,O(I))
10060 NEXT I
10070 CALL S("CS1",Y)
10080 REM END OF SUBR
10090 RETURN
```

```
10100 REM LOAD SUBR
10110 CALL L("CS1",Y)
10120 FOR I=1 TO 150
10130 CALL G(1,I,1,MIS,O(I))
10140 NEXT I
10150 REM END OF SUBR
10160 RETURN
```

When your main program is at a point that you want to save the values of array O it only takes a line GOSUB 10000 to do so. In the same way GOSUB 10100 will fill the array with the values played from the tape.

Detailed explanation:

10010 tells the computer that one item will be used per page.
10020 determines this item to be of type 2 (=integer)
10030 reveals that you do not intend to have values beyond 99. Width is 2.
10040 to 10060 stores all 150 values of the elements of array O in MEM2.
In PRK language: item 1 on page I gets a value of O(I)
10070 saves the array as a PRK file from MEM2. CHECK TAPE (Y/N) is asked.

10110 loads PRK file from tape in MEM2
10120 to 10140 is a loop that fills array O with the values in MEM2

We recommend you study this example closely and you will discover other possibilities that are meaningful to you. Change lines 10020 and 10030 to suit your data.
Use nested loops for multi-dimensional arrays. Use your head as in computing:
The sky is the limit.

2021 note: The left hand edge printing went beyond the physical paper. A new white left edge added 2021.

AB & C Computers

MAIL ORDER for TI-99'ers

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

TI INVADERS	£14.74	VIDEO GAMES 1	£18.22	MIND CHALLENGES	£11.09
A-MAZE-ING	£18.22	CONNECT FOUR	£18.22	SOCCOR	£18.22
TOMBSTONE CITY	£18.22	WUMPUS	£18.22	ZERO ZAP	£18.22
ATTACK	£18.22	BLASTO	£18.22	HUSTLE	£18.22
YAHTZEE	£18.22	CARS WARS	£21.70	BLACKJACK/POKER	£18.22
MUNCHMAN	£21.70	CHESS	£29.35	ADVENTURE & PIRATE%	£29.35
NUMBER MAGIC	£11.09	PRE-SCHOOL FUN	£12.83	BEGINNING GRAMMAR	£14.74
HANGMAN	£18.22	MULTIPLICATION	£21.70	ADD/SUBTRACT 1	£21.70
EARLY READING	£21.70	MUSIC MAKER	£21.70	ADD/SUBTRACT 2	£21.70
HOME BUDGET	£18.22	PERSONAL RECORDS	£36.74	PERSONAL REPORTS	£36.74
SPEECH EDITOR	£14.74	STATISTICS	£36.74	TERMINAL EMULATOR	£36.74
EXTENDED BASIC	£66.09	TI LOGO	£113.00	MINI MEMORY %	£66.09
EDITOR/ASSEMBLER*	£66.09	TEXT FORMATTER*	£43.44		
* REQUIRES DISKS		% MODULE/CASSETTE			

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

ADVENTURE LAND	£18.22	VOODOO CASTLE	£18.22	MISSION IMPOSSIBLE	£18.22
THE COUNT	£18.22	STRANGE ODYSSEY	£18.22	MYSTERY FUN HOUSE	£18.22
GHOST TOWN	£18.22	GOLDEN VOYAGE	£18.22	SAVAGE ISLAND 1+2	£21.70
PYRAMID OF DOOM	£18.22	OLDIES/GOODIES 1	£7.18	OLDIES/GOODIES 2	£7.18
PROGRAM AIDS 1	£7.18	PERSONAL FINANCE	£7.18	MARKET SIMULATION	£7.18
EXTENDED BASIC CSE	£7.18	TEACHYOURSELF BASIC	£7.18		

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

TEXT FORMATTER*	£43.44	INVENTORY	£43.44	MAILING LIST	£43.44
INVOICING	£43.44	PROGRAMMING AIDS 2	£14.74	PROGRAMMING AIDS 3	£14.74
MATHS LIBRARY	£21.70	ELECT/ENG LIBRARY	£21.70	STRUCT/ENG LIBRARY	£21.70
PASCAL EDITOR	£43.44	PASCAL COMPILER	£69.52	PASCAL LINKER	£58.70

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

GAMES TAPE 1	£1.50	GAMES TAPE 2	£3.45	GAMES TAPE 3	£4.30
O's + X's,MASTERMIND	£5.20	HAUNTED HOUSE	£5.20	MAZE/TOWERS HANOI	£5.20
SIM/O's + x's	£4.30	FRUIT MACHINE	£5.20	ROULETTE/CRAPS	£4.30
PLANET LANDER/LIFE	£4.30	WUMPUS	£4.30	BLACKJACK	£5.20
LUNER LANDER	£4.30	FOUR IN A ROW	£5.20	TEN PIN BOWLS	£7.50

TEXAS TI99/4A HARDWARE

TI99/4A COMPUTER	£170.00	SPEECH SYNTHESISER	£75.00	EXPANSION SYSTEM	£135.00
DISC CONTROLLER	£115.00	DISC DRIVE 5 inch	£215.00	RS232 CARD	£95.00
32K RAM CARD	£165.00	P - CODE CARD	£145.00	MATRIX PRINTER	£325.00
CASSETTE CABLE	£7.95	JOYSTICKS PAIR	£16.95	TI BASIC BOOK	£11.25

WE HOLD A WIDE RANGE OF BOOKS AND COMPUTING ACCESSORIES, PLEASE SEND A SAE FOR LISTS

TELEPHONE ORDERS FOR PURCHASES AGAINST ACCESS ACCOUNTS WILL NORMALLY BE DESPATCHED THE SAME DAY. CHEQUE WITH ORDER PURCHASES ARE DESPATCHED AFTER CHEQUES ARE CLEARED PLEASE HELP US TO HELP YOU WITH YOU SOFTWARE REQUIREMENTS, SOME SOFTWARE IS IN SHORT SUPPLY, SOME TAKES A WHILE TO OBTAIN. AN EARLY ORDER WITH 20% DEPOSIT WILL ENSURE YOU GET WHAT YOU WANT WHEN YOU WANT IT. PLEASE PHONE US IF YOU HAVE A PROBLEM WITH TI99/4A

DONT FORGET TO ORDER EARLY FOR YOUR XMAS REQUIREMENTS.

A. B. & C. COMPUTERS,
11 BROCKSTONE ROAD,
ST AUSTELL, CORNWALL.
Tel: St Austell (0726) 64463
Tel: Newquay (06373) 6886
Personal callers welcome

Terms — Please add VAT at 15%. Cheque P.O. with order. Carriage Computers' Printers £5.00. Post and Packing other items 60p. All hardware checked by service engineer.

24 HOUR ANSAPHONE FOR ACCESS ORDERS

random dots

BOOKS & MAGAZINES

If you haven't already noticed, CURRYS MICRO C are now selling "Introduction to TI Basic" and 99er Magazine, along with a good range of software, (at least they are in Nottingham!!).

PROGRAM LENGTH

I came to the 99/4 (like many others I suppose) via a succession of programmable calculators, which had made me very careful about the length of my programs due to the very limited amount of memory available on these devices. Then, suddenly, I had 16K to play with and my programs used RAM as if it was limitless. Then I learned that 16K was not much RAM after all and once more I started looking for memory saving devices in my programs. However, that's not the end of the saga, as I then went on to consider that, apart from tidiness, there is nothing to be gained from using less than 16K, unless your program is a DATA HANDLING one where maximum memory is required for the data. This is quite different to my SHARP PC 1211 Pocket Computer, which has non-volatile memory and so has the facility for holding several programs at the same time in memory, again making memory conservation almost essential. I suppose that all you lucky (and wealthy!) DISK DRIVE owners out there will disagree (and I only know of 3 in the whole Group).

DELIBEERATE? MISTAKE

I would like to thank the 273 members who wrote to me pointing out a bug in the little routine listed in the last RANDOM DOTS. For the handful who remain, and presumably have run out of ink, (or just plain missed it), here is the correction.

* Line 200 should have been:

```
200 PRINT "CALL COLOR(GROUP"&","
&STR$(FOREGROUND)&","&STR$(BACKG
ROUND)&")"
```

ARITHMAGRAPHS

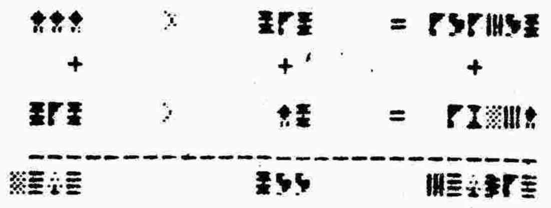
Here are another two puzzles using the same codes as some people apparently had trouble with the first one.

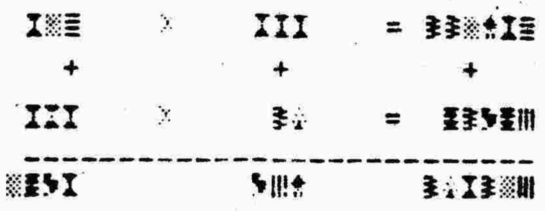
ARITHMAGRAPHS

DEVISED BY MIKE O'REGAN

THESE SYMBOLS REPRESENT NUMBERS

TRY TO FIGURE THEM OUT!





LANGUAGES

I have just read the "LOGO Story" (Seymour Papert's "Mindstorms") and it really set me thinking that perhaps TI have missed the best opportunity ever available to a Micro Computer maker. Why didn't they produce the 99/4 with 48K RAM from the outset, running LOGO as its RESIDENT LANGUAGE? Instead of that, what have we really got! LOGO available as an extra, if we are prepared to pay the exorbitant price for the RAM Expansion which is required, plus the cost of the LOGO Module. Meanwhile, APPLE are already jumping on the LOGO bandwagon, with the BBC and, no doubt, Sinclair (who already have the cheap 48K RAM available in the SPECTRUM) waiting in the wings. If BBC price comparisons of current software, compared to Texas, are anything to go by, then their version of LOGO will probably cost about £30 and once again we will have missed the boat!

STANDARD CHAR CODES

For those of you without Extended Basic, here is a list of the Standard Character Hexcodes which you may find useful.

33	0010101010001000	!
34	0028280000000000	"
35	00287C28287C2800	#
36	0038543018543800	$
37	00444C1820644400	%
38	0020502054483400	&
39	0008102000000000	'
40	0008101010100800	(
41	0020101010102000	)
42	0044287C28440000	*
43	0010107C10100000	+
44	0000000030102000	,
45	0000007C00000000	-
46	0000000000303000	.
47	0004081020408000	/
48	0038444444443800	0
49	0010301010103800	1
50	0038440810207C00	2
51	0038441804443800	3
52	00081828487C0800	4
53	0078407804443800	5
54	0038407844443800	6
55	007C040810202000	7
56	0038443844443800	8
57	0038444C3C047800	9
58	0030300030300000	:
59	0030300030102000	;
60	0000040810080400	<
61	0000007C007C0000	=
62	0000402010204000	>
63	0038440810001000	?
64	0038445458403C00	@
65	003844447C444400	A
66	0078447844447800	B
67	0038444040443800	C
68	0078444444447800	D
69	007C407840407C00	E
70	007C407840404000	F
71	003844404C443800	G
72	0044447C44444400	H
73	0038101010103800	I
74	0004040404443800	J
75	0048506050484400	K
76	0040404040407C00	L
77	00446C5444444400	M
78	004464545444C4400	N
79	007C444444447C00	O
80	0078444478404000	P
81	00384444544C3C00	Q
82	0078444478484400	R
83	0038443008443800	S
84	007C101010101000	T
85	0044444444443800	U
86	0044444444281000	V
87	0044444454542800	W
88	0044281010284400	X
89	0044442810101000	Y
90	007C081020407C00	Z
91	001C101010101C00	[
92	0080402010080400	\
93	0070101010107000	]
94	0010284400000000	^
95	0000000000007C00	_

USER GROUPS etc.

Would you believe it! there are no less that five TI Home Computer User Groups in GERMANY!! and Groups in Belgium, Sweden and Denmark (plus the group in Holland who we already know). I am writing to the German groups to establish contact (if they are interested) because I can speak/write German. Any volunteers to contact the other nationalities? While on the subject of User Groups, I think it a good idea for local groups to set up, within the framework of TIHOME, so if anyone in the Nottingham/Derby area is interested they can contact my by 'phone or Long Eaton 5482 or write to:

130 Stapleford Lane
Beeston
Nottingham NG9 6GB

UNCONNECTED DOTS

My Music Maker Module hesitates at regular intervals on Playback, when all three voices are extensively used in Traditional Mode. Has anyone else had the same trouble??

I like Pete Brooks' "Patterns" Program (have YOU tried it yet?). I can sit watching the changes for minutes on end. Just one thing - Input of 9 makes it "go high".

My T.P. is playing silly-socs again - missing out a line of dots (or had you already noticed). On these occasions I find that the "head cleaning" routine doesn't work and I have to dismantle the printer to clean the matrix-carriers. Going to the PCW Show on the 9th September - see you hanging around the TI Stand drooling over the goodies we can't afford.

HARDWARE COST

I very much endorse what Peter Brooks said in the last TIdings about the cost of hardware for the 99/4. It's not so much that we paid through the nose for our computers in the first place (many moons ago), but when we continue to be ripped off when it comes to buying peripheral hardware it begins to hurt.

As Pete says, in the USA the 99/4 is known as a low-cost system, and the prices over there bear little relationship to the same items over here. I did notice that Robin Frowd didn't mention the PRICE of the TIBOX (oops, sorry Peripheral Expansion Box) Is it just accidental that he didn't mention that the said TIBOX will NOT ACCEPT ANY OF THOSE pretty little COMMAND MODULES which you have been collecting, at no small expense!!!

Pity the first TI Newsletter couldn't have told us that the Mickey Mouse Wired Remote Controlllers (Joysticks to you), have at last been replaced by an improved?? model. Not before time, as my family have just about worn-out the E, S, D, bnd X buttons on my console with all-night sessions of TI Invaders, the Attack, and the latest craze Munch Man. When are we going to see Modules for Wordstar & Mailmerge (complete with 80 columns!),Supercalc etc. Where are the details on that nice-looking (full-width?) PRINTER, as illustrated in the latest glossy handouts for the 99/4

Mike O'Reg

by Les Watson

BUYING A COMPUTER FOR THE FIRST TIME.

The choice between a new washing machine and a computer wasn't easy.The day I made the Impulse buy that I had been thinking about for weeks was a Saturday when my wife decided to stay at home.

Walking through Dixons on the Saturday afternoon I did what Ibelieved was impossible until that moment.I saw what looked like a genuine bargain.It was a T.I.99/4A being offered at £199.95.(This was on June the second.)

Carefully removing some of the protective layer of dust I quickly assured myself that there was no obvious physical damage to the machine before I bought it.Half way out of the shop thinking about how I could explain this sudden purchase to my wife I had a brainwave if I purchased a game for my daughter it would be two against one which meant certain victory.I returned to purchase T.I. Invaders at £19.95 which turned out to be an excellent game as well as fulfilling my purpose.

Fifteen minutes later my daughter and I were sampling the delights of the nearest thing I've seen to the arcade version of Space Invaders.

Whilst my daughter played I read the manuals,when I played my daughter complained that she wasn't playing.Eight hours later we decided that it was time to call it a day and we went to bed.

The fight for computing time continued the following day from abott seven thirty a.m..We have now come to an arrangement ie:-When my daughter doesn't want the computer I can have it.

Several days later when my daughter had realised that typing 'What do I do next'did not provide the answer to all known problems she asked me why I could not write 'Pacman'.

I explained that it was purely because I did not have the time and immediately sent off to T.I.HOME. for some copies of programs of various types.

The thought of writing ' Pacman' intrigued me so I started to do this never having written a games program in my life before. Atari have no need to worry yet.My version of 'Pacman' is so slow that he dies of starvation.

The exercise was nevertheless instructive since I had to find out how to create the illusion of movement,how to create a maze,and how to live without Peek and Poke.

I have omitted to relate my experiences of trying to buy a cassette cable from Dixons and Rumbelows.The salesman at Dixons new immediately that they didn't have one and that it would be at least thre weeks before they could get one.

The salesman at Rumbelows gave me a blank stare when I first made my request and even when I showed him what a T.I.99 was he was not very helpful.It was fairly obvious that he could not understand why on earth I should want to connect a tape recorder to a computer.(Full marks T.I. for product training.)

Not all my experiences of trying to buy software and hardware have been bad ,just most of them.

The shining light amongst all the companies I have dealt with or tried to deal with must surely be 'Computer Supermarket' in Corby Northants.Having been convinced by Dixons and Rumbelows that there had to be a better way I got out my latest edition of P.C.W. and called Computer Supermarket.To my surprise the answer to my enquiry about a cassette cable was an immediate 'Yes we have them in stock if you give me your credit card number we will ship it today'.To my complete amazement it arrived eighteen hours later.

Equiped with a cassette cable I was able to start developing my own programs and to start doing lengthy conversions of other peoples programs from magazines and books.

To date I have converted programs written for PET BBC and the ubiquitous ZX81.Without doubt the most successful conversions have been adventure type games which generally don't require Peek,Poke,Fast Graphics etc.

Anyone interested in copies of these games please contact me by post enclosing a cassette and a stamped addressed envelope plus 20 pence to cover photocopying costs.If the programs do not please I'm sorry but you are not likely to find any cheaper.Anyone who thinks I am trying to make a profit need not apply.

Although the emphasis so far has been on the use of the T.I.99 for games programs this is not where my main interest lies.

My interest are predominantly scientific with a strong bias towards Physics.I would very much appreciate hearing from others with similar interests.

Before June 2 I was really unaware of any other language but FORTRAN I am now a convert to BASIC for a number of reasons but in the main because it is simpler and because of its string handling capability.

I am trying to purchase Extended Basic and the Minimemory modules in order to be able to write more complex programs.So far I have been unable to obtain either.This must surely be a marketing mans nightmare,everyone would buy it if it was available.(Robin Frowd please note if I was your boss you would be unemployed next month)

Let us hope that before our beloved machine is swamped by the VIC 20,BBC model A&B,ZX81,SPECTRUM,DRAGON,ATARI 400/800,VIC64?? that T.I. reduce the price of there software and make sure that it is available.

-LES WATSON

71 BADDOW HALL CRESCENT
CHELMSFORD
ESSEX

TI HOME
AUGUST 1982

RAMBLES

AUSTRALIA

Not many days after a phone call from Canada, arranging for me to distribute some Canadian programs in the U.K., I have received a phone call from even more distant shores - Perth - in Australia.

Travelling a distance of about 2 x 10^8 metres (return trip) even at the speed of light there is an appreciable echo, a bit disconcerting. (That distance is via Satellite by the way!). The following news was passed on:

Extended Basic Version 110 - variations:

a. !@P+ & - does not have a separate coding (see my request in April issue for details of condensed format for this new pair of commands). It is stored just like that, in four bytes.

b. Everything I've written regarding memory locations for your program is for Version 100. If you use Version 110 the first 24 bytes are occupied with something else. Therefore if you use CALL LOAD to amend a program, when using 110 you must subtract 24 from the memory location - eg a program line starting at location -200 in V.100 will start at-224 in V.110.
This need not prevent portability of programs- use CALL VERSION as follows:

```
NNNN. CALL VERSION(VN) :: VN=(VN-100)*2.4
```

Then when using call load, use the form:

```
CALL LOAD( LOC-VN, CODE)
```

where LOC is the location when using version 100.

Think about it- if using V.110, then VN will be 24
if using V.100, then VN will be nil.

The minus sign has been used (-VN) because program locations are in the negative side of memory, and to increase a negative number, you take more away.

c. Another unpublished difference - my BenchMark timings for ExBas in June issue were for Version 100. You should find Version 110 behaving as 'speeded-up' v.100. This is because v.110 automatically sets memory location -31878 to the number of sprites you are using, without the need for expansion memory!
So you lucky v.110 owners have a faster version - your programs will on average run 30% faster in ExBas than in TI BASIC.
If you have v.110 - dont bother putting in the 'speed-up' line - you dont need it. (If you have v.100 you need the Expansion RAM and need to use:

```
CALL INIT :: CALL LOAD(-31878,0).     )
```

You think we have it tough? Down under their peripherals are supplied for 110 volt operation- and they use 250 v. They have to use a separate transformer.

The users club there is still a little new- just two newsletters out so far -but they seem to have some very competant members- one has built the speech synthesiser INTO the console (I don't know where!) - with luck the details may be forwarded to Paul in due course. Acknowledgement- the April article on disks came from Australia.

Thats all from the Antipodes.

Now for some GOSSIP - styled for Mike O'Reagan in easily digested format:

BOOTS had a full page Ad in PUNCH recently- just for the VIC-20. No mention of the 99/4A. Presumably Commodore went halves on the cost of the ad.

+++++++

NORTON SOFTWARE:
The prices given in the last issue are cancelled. New prices effective immediately are:
Graphics-£9 Music-£7
Attack Man, Car Rally-£7
Star Trek-£5 all rest £6.

My address is:
10 Alstone Road, STOCKPORT,
Cheshire, SK4 5AH. (U.K.)

An SAE is always appreciated -funds are not limitless alas.

Dealers don't seem too interested in software for the 99/4A. So keep your eyes on the ads in TIdings!

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

Speaking of Commodore- they have adjusted their American Ad for the VIC-20, to correct all those lies they were telling about the 99/4A

+++++++

Taking a week off work recently I was impressed by the number of times the phone rang. If you need to ring me, please do so after 7pm, and not on Wednesdays. I prefer letters, they give me time to think!

++++++

Idea from Australia- CONTEST:
Write a SHORT program to demonstrate the 99/4A- graphics &/or sound -a game or just a demo.
Target: TI BASIC: 25 lines
Extended Basic:10 lines.

If you really try you'll be amazed at what you can do.
PRIZE: Instant fame in TIdings.

===========

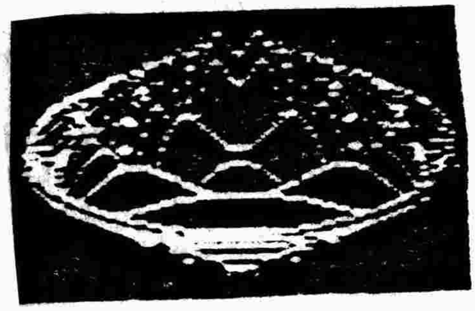
REQUEST

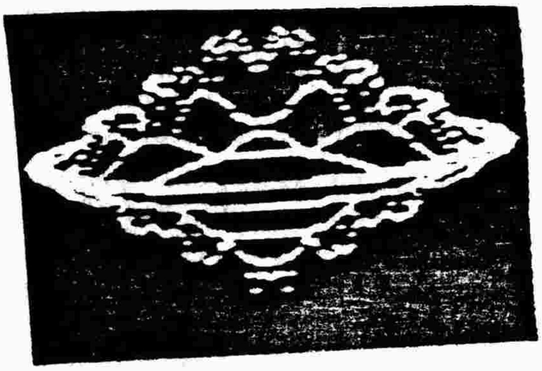
TI HOME TIdings welcomes news and articles from ANYONE (...... that means YOU). Do drop a line. Editorial address is usually on the bottom of the last page.

99er Magazine Issue 5 - for those of you not yet subscribers-

Contained some VERY good action games in Extended Basic, and a rather easy version of Reversi (Othello).
Contained a detailed review of the mis-named 'MINI-MEMORY MODULE' -not just 4k of RAM but a whole lot more. With this module you may write and run programs in Assembly Code with only the console and a tape recorder -no need for a disk system or expansion ram.
You can access every byte in the machine, including switching to a 40 character screen width, and can define 768 different characters at once (that does however take up 12k of memory...). This magic module SOUNDS very exciting - I have one on order from the States, & will report in due course.
NB: Owners of the old 99/4: You have a different VDP chip & can't define 768 characters- sorry. The demo program supplied will not therefore work on a 99/4-its a hi-res graphics demo using 768 chars.

Have you tried the hi-res graphics of TI HOME Feb & April issues? If you haven't, the following photos are the result of one of the February programs (hope they Xerox OK) :





BRICKBAT:

Computing Today take full copyright on programs they publish. They also 'unequivocably' refuse to consider republication. Don't bother asking.

====

Creative Computing:

New ownership from now- now owned by ZIFF DAVIS a very large US publisher - well known for its Science Fiction Magazines over here. With the new owner the regular 'OK to REPRINT' consent has vanished.

+++++++

I now understand Robin Frowd is sending Paul full details of the Extended Basic now on sale- version 110. Very few of us have v.100, but the news will be very welcome! (cf.Australian news earlier).

****

99'er magazine is US$43 for 6 issues, by airmail from:
P O BOX 5537, Eugene, OR, USA, 97405

Strongly recommended.

BOUQUET

MicroComputer Printout take full copyright -but they will consider republication AND ensure a fair deal for the contributer. Nice.

=====

By the time you read this the TI price reductions on modules & other software should have filtered down to your dealer.
The new prices are quite nice - the Editor/ Assembler package is good value at double the price.

P-CODE

The new peripheral box which TI are bringing out will be able to run UCSD P System programs & languages.

Pascal is an example.

The 'p' stands for 'pseudo-code' - the p-system was developed to be used on any micro via a special interpeter for each machine, thus greatly increasing portability of programs.

(In Holland work is going on on a similar pseudo-code to permit exchange of Basic programs between machines).

This system permits the free exchange of software , and can be run on any new machine.(Provided the manufacturer has an interpreter).

Developed by the University of California, the system is at present owned by SofTech. There are estimated to be 100,000 systems, two thirds Apple. This compares to 300,000 C/PM systems- a single processor based language which runs slowly on other processors.

The following manufacturers use the p-system- Philips, TI, IBM, Apple &c.

In addition to the Pascal compiler, other languages supported are Fortran 77, Basic, APL, Lisp, and Modula-2. Possibly TI may introduce these if there is any demand.

At present TI have in the pipeline-

p-Code Card for use with the peripheral box.	$195
On Disk:	
Pascal Compiler	$100
p-System Assembler/Linker	$ 80
p-System Editor/Filer Utilities	$ 65

Prices are typical US estimated retail.(Change to a £ sign for UK.?)

TI are using the most recent Version IV UCSD P System.
'PILOT' is forthcoming in p-system code.

The p-card will enable you to RUN Pascal programs,but you need the Compiler to write them. Oh - you will also need the Memory Expansion, and if you want to write Pascal then you will of course need a Disk Drive.(& Controller).

Now-

IF you are interested in p-code, it has its own tiny users group in this country - about 150 members!

Mark Woodman,Dept of Maths,Open University,Walton Hall,
Milton Keynes.

IF you are quite bewildered by the whole thing, you don't need to understand it to run programs written in it!

The new availability of Assembly language, and shortly the p-system will mean that (in one or two years....?) programs available for the TI system will be of a much improved quality.

(Sourcew: Personal Computer World, 99'er Magazine)

PRINTERS:

If you want a better printer than the Thermal Printer, the standard printers used in the States are by EPSON . In addition to the RS232 interface you also need EPSON interface 8141 or 8145. However,the new TI RS232,for use with the peripheral box,has a parallel output and will connect directly with say an MX-80.

The connecting details are fiddly- buy through a dealer who is prepared to assist you in setting the system up!

ASSEMBLY LANGUAGE:

By the time you read this one or two members should have either the Editor/Assembler or the Mini-Memory Modules, both of which permit Assembly Language programs to be written.

If you have a) Mini Memory or
b) Extended Basic AND Expansion Memory

then you can RUN programs in Assembler (4k with a, 8k with b).

It is now possible to provide routines and utilities for use by programs written in Extended Basic using CALL LINK - eg CALL LINK(HIRES).

There will inevitably be more members able to RUN assembly language, and wanting fast routines, than there will be members able capable & willing to write the necessary routines.

Why not write in and suggest general purpose routines that you would like to run faster than they do in Basic ? Is there anything you want your 99/4A to do which you cannot do in Basic? We have heard of a US program which is very long-taking about 4 mins to load FROM DISK using the standard TI operating system. The supplier sells it with an Assembly Language LOAD routine which does it all in 20 secs.

(NB: To run assembly language routines in Extended Basic, you also require the expandion memory, which gives you, in rough figures, 24k for BASIC programs, 14k for Stack, and 8k for assembly language programs/routines).

SPACE SAVER

In Extended Basic you have TWO rems - you can use REM or !
If you use REM you have to leave a space after REM- and that takes up a byte of memory.
If you use ! your rem may follow immediately:no space is required.
So- save-a-byte ; Use !

```
100 !SAVE A BYTE
```

The new TI Publicity Brochure for the 99/4A has a picture of an anonymous tractor feed printer, but describes in the text the usual Thermal Printer.
If you haven't identified it yet, the printer pictured is the EPSON MX-80 !!! Could TI be going to stick their name on it? At the same price as the straight Epson? And for which RS232 interface will it be- the old serial one or the new parallel (Centronics) one ?

Correction: If you are a new member & are using this sales brochure (white cover) please delete MPI Ltd from p.14 -they no longer handle the 99'er Mag. TI may have made other arrangements by the time this reaches you.

.........................+..................................-.....

MICROCOMPUTERS: Dealers,Manufacturers & Price Fixing:

From 'Computer Dealer' July 1982:

APPLE'S threat to axe any dealer with a poor sales record has been carried out. The casualty toll is 231.

The company's official dealer list now carries 351, against the previous 582. And it could be trimmed still further.

DEALERS keen to cash in on the boom in sales of the BBC micro have been given a point-blank "No" from manufacturers Acorn to their plea to be able to sell the machines at a profit.

And Acorn say there is no shortage of dealers willing to sell the machine at the mail order price and make their profit on add-ons.

APPLE pledged this month to protect its dealers' profit margins and do everything in its power to bring an end to discounting.

The promise came from Apple Inc's vice-president and European general manager Tom Lawrence.

And the campaign is already well under way, he told *Computer Dealer*.

Said Lawrence: "We are working pretty hard right across Europe to stamp this out, and with some success.

The operation of Resale Price Maintenance is illegal in respect of consumer sales.
It is illegal for a manufactuer to prevent,by any means, a retailer selling goods to a CONSUMER at a price below that fixed by the manufacturer.
If a manufactuer says to a dealer 'Sell my computer for less than £180 and I will not sell you any more' - it is illegal IF IT APPLIES TO CONSUMER SALES.

If a dealer sells other than to end-consumers eg to industry, business, schools, colleges, he can be forced to sell in excess of a minimum price.

Under such circumstances he may well wish NOT to sell at a lower price to end-consumers.

The dealer is in a cleft stick. He may wish to and be able to sell at a much lower price to consumers, and could perhaps help the manufacturer sell many more computers, but he is -indirectly- prevented from doing so.

Why not drop a line to TI-UK, TI-Dallas, your MP, the BBC etc, and let them know what you think of such operations.
Do you think that it is in a) the dealers interest?
b) the manufactuers interest?
c) the publics interest?

My reply is 'NO' to all three categories. NB: I have no evidence to suggest TI operate in this way.

INTERNATIONAL PRICE DIFFERENCES:

Some years ago cameras were very much cheaper abroad than in the UK.
At present you can buy a car abroad and make a worthwhile saving- though it is a lot of trouble.

There is some evidence that car price differentials are slowly falling, and UK Camera prices have dropped a very great deal in the last decade, as more people forsook guarantees in favor of buying abroad,cheaper.

TI blame the long (and expensive) supply line of their products on the price differential (about 30% on some software) - which has recently dropped. TI shipments do NOT come direct to the UK- TI have operations in many European countries. Should we pay for an inefficient supply line? Is there further room for price reduction- especially on the peripherals?

These are topics for open discussion.

CASSETTE UPDATE:
A list of suitable recorders appeared in Vol.2 Issue 1 (Feb 82).

The main problem seems to be that some recorders are very sensitive as to the setting of the volume control - presumably because the output lacks sufficient treble. DATA is even harder to read than programs because it seems to affect AVL more.

The solution is to buy a recorder with a treble control & set it high.

In addition to the recorders previously listed
(I changed to a Lloyd V182 from Comet & found a BIG improvement)
I am advised by Dr N Rabindran of Warrington that the following recorders are also very easy to use, and care very little what the volume is set to:

BOOTS CR325 (also marked 'Automatic CRo2')
MARANTZ C190.

NB: The Commodore computers use data signals - cassettes for and by Commodore are NOT suitable for the 99/4A which requires high quality audio tapes. W H SMITH's tapes are OK, as are the tapes advertised in the computing press by DATA-ASSETTE, who also appear to be selling the V182, but not called that!

GOODIES TO COME:

News reaches me of some exciting programs for the 99/4A, as programmers just begin to use its power, for instance- would you like

Program A: which follows in the footsteps of TI's Programming Aids 3, and uses the technicalities of the Merge function to enable you to use the computer to shorten programs?
This program has several options, which you can use, for instance, to remove REMS and tail statements,
to use multiple statement lines to the maximum depth
to shorten all variable names (and provide a cross reference listing!) as much as possible.
Can speed up execution time 20%as well as saving Disk space!

or perhaps you may prefer:

Program B: A selection of utilities written in Assembly Code to enhance your displays - for instance with part of the screen scrolling, reverse scrolling, 40 character screen width etc.

I'm trying to obtain copies for myself and will give further news if & when they are to hand. I will also try to obtain consent to sell them in the UK- watch out for the ad in TI HOME.

These two will give you some idea of what is possible. The bad news is they require peripherals like the Disk System or the Expansion Memory!

99/4A and COMPUTER AND VIDEO GAMES:

You will have seen me mention in these columns the magazine 'Computer and Video Games'.

Now, TI HOME has several members, but the Good Editor of the above magazine tells me that NONE of you have sent him a program to publish for the 99/4A.

That's why you haven't seen any listings.

It is an excellent magazine, and several of the programs listed are fairly easy to translate for the 99/4A.

I am informed that the 99/4A will be included in the issues dated Oct,Nov and Dec 1982. These are available in the middle of Sept,Oct and Nov 82!

The magazine pays £10 for first publication and sends you a free issue too. You may specifically reserve all other rights.

If you want to see programs for the 99/4A published you know what to do now.

SEND THEM to C&VG mag. At the very least write to the Good Editor and thank him for his interest.

Unless he receives a good response to the Oct-Dec issues he will probably not give the 99/4A another glance.

It's up to YOU (stop snoring at the back!).

I have receifed a free copy of a new magazine 'Personal Computing Today' - I'd like to say something nice about it. Sorry.

I have the impression that the content of the general UK mags is dropping.
This could be because I compare them to 99'er Magazine, or it could be because they are written by so few people (some names occur in more than one mag) or it could be insensitive aggressive publishers, or perhaps they cannot stand the competition from the many specialist magazines?

I have so far seen specialist magazines for the Apple, Vic, BBC, ZX81. It will be interesting to see how they last. Microcomputer Printout started as a 'PET-only' magazine but is now quite general.

Extract: Editor/Assembler Manual:

"The extended operation instruction (XOP) is available in some TI-99/4A computers.
The only way to determine if your computer supports this instruction is to try it."

That is definitive info!

((Dont use XOP if you write assembly programs. OK?))

ELECTRONICS & MUSIC MAKER July 82

Has a review of TI's Music Maker Module.
Their biggest beef was about a feature only to be found on the very few PAL 99/4's sold.
The problem is not found on 99/4A's. (No sound through tv).

SCIENTIFIC AMERICAN July 82

Has an article on 3d puzzles such as the Cube. There is an honorable mention of the 99/4 program 'Quadcube' -mentioned before by me.

I heve now received from NY: Editor/Assembler (v.1.2)
Mini-Memory Module
Personal Report Generator

The first two cost £65* each, the third £37 - inclusive of airmail post & insurance, Customs Duty & Clearance Fee, and V.A.T.

The Editor/Assembler package is IMPRESSIVE - it weighs over 2Kg. You receive the module, two disks, and a BIG strong loose leaf binder with 470 'A4'-size pages, a container for four modules. and a page to hold four disks (of course, you make back up copies and use them, not the original disks).

The 'manual' is detailed, but not at all easy! Difficult is a good word.

The Mini Memory Module comes with a cassette and two SMALL manuals. The cassette has on it a line-by-line assembler, and a demo-program which ONLY works on the 99/4A - it uses graphics mode 4, which is not available on the older 99/4. You really need the big manual to use it properly to write Assembly programs.

To WRITE Assembly programs I recommend the Editor/Assembler package, although you CAN write using the Mini-Memory.
If you wish to have access to Assembly programs, for fast action games, extended utilities, then the Mini-Memory is probably a good buy.
NB: With the Memory Expansion, you can load 8k Assembly Code with the Extended Basic module, or nearly 32k code with the Editor/ Assembler or nearly 36k of code with the Mini-Memory.
The 'Loader' in Extended Basic differs from that in these two new modules in various respects .

I have been able to 'CALL LOAD' Assembly programs, in TI BASIC, with the two new modules inserted. My Version 100 Extended Basic returns 'Unrecognised Character' when CALL LOAD is attempted.

The Mini-Memory module can LOAD Assembly from Disk, but only save to tape.

If you have the Memory Expansion and wish to LOAD the Mini-Memory permanently from Disk, for the program to stay in storage after switch off,you must either switch off the Memory,or 'instruct' the computer it isn't there,using CALL LOAD's. The Manual tells you how.

Memory usage- a small assembly code program, published in 99'er, served as my test program-
The source code (what you enter) took up 8 sectors of disk.
The object code (what you RUN) took up 7 sectors.
Condensed Object Code took up 5 sectors.
'Program Format' object code took up just 2 sectors.

OC and COC can be run with the Mini Memory, PF requires the E/As.

SIGNIFICANT ERROR: The Editor/Assembler,as an example,includes disk version of 'Tombstone City' and the TC Manual suggests you can run the disk with Extended Basic. No Way -not with version 100 anyway! It may work using CALL LOAD with VERSION 110,but never by using the normal LOAD which is suggested!

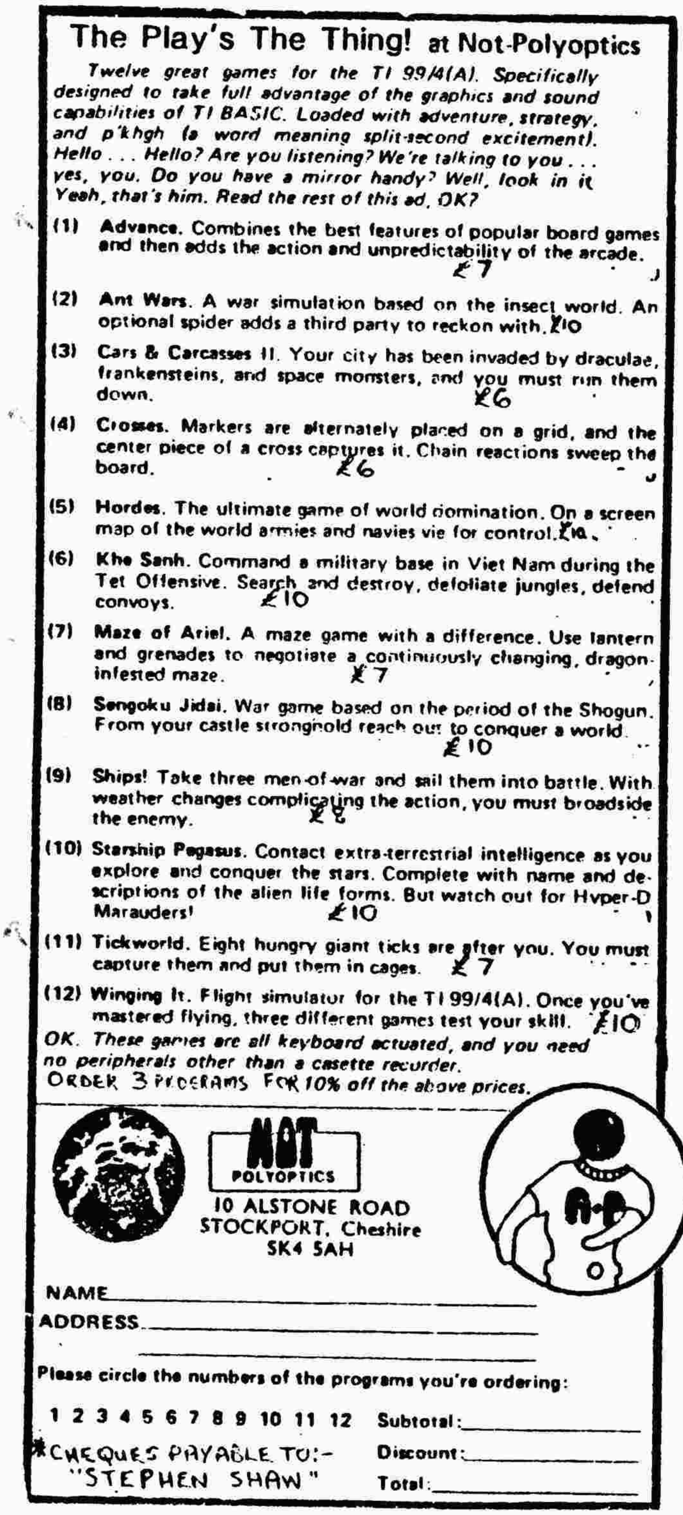
Both these modules have their strong points,and are not really 'either/or' , but rather supplement each other.

NOT POLYOPTICS have advised me they plan to miss Extended Basic, and develop their next list of programs for the Mini-Memory - I'll let you know what comes next! OAK TREE SOFTWARE are producing Assembly routines claimed to run with ExBas -I'll inform you when I have been able to try them out!

* U.K. price ABOUT £90. Watch for discounts.

PAID ADVERT
Programs from NOT POLYOPTICS :

NB: 2, & 9
are NOT at present in stock.



The Play's The Thing! at Not-Polyoptics

Twelve great games for the TI 99/4(A). Specifically designed to take full advantage of the graphics and sound capabilities of TI BASIC. Loaded with adventure, strategy, and p'khgh (a word meaning split-second excitement). Hello . . . Hello? Are you listening? We're talking to you . . . yes, you. Do you have a mirror handy? Well, look in it Yeah, that's him. Read the rest of this ad, OK?

(1) **Advance.** Combines the best features of popular board games and then adds the action and unpredictability of the arcade. £7

(2) **Ant Wars.** A war simulation based on the insect world. An optional spider adds a third party to reckon with. £10

(3) **Cars & Carcasses II.** Your city has been invaded by draculae, frankensteins, and space monsters, and you must run them down. £6

(4) **Crosses.** Markers are alternately placed on a grid, and the center piece of a cross captures it. Chain reactions sweep the board. £6

(5) **Hordes.** The ultimate game of world domination. On a screen map of the world armies and navies vie for control. £10

(6) **Khe Sanh.** Command a military base in Viet Nam during the Tet Offensive. Search and destroy, defoliate jungles, defend convoys. £10

(7) **Maze of Ariel.** A maze game with a difference. Use lantern and grenades to negotiate a continuously changing, dragon-infested maze. £7

(8) **Sengoku Jidai.** War game based on the period of the Shogun. From your castle stronghold reach out to conquer a world. £10

(9) **Ships!** Take three men-of-war and sail them into battle. With weather changes complicating the action, you must broadside the enemy. £8

(10) **Starship Pegasus.** Contact extra-terrestrial intelligence as you explore and conquer the stars. Complete with name and descriptions of the alien life forms. But watch out for Hyper-D Marauders! £10

(11) **Tickworld.** Eight hungry giant ticks are after you. You must capture them and put them in cages. £7

(12) **Winging It.** Flight simulator for the TI 99/4(A). Once you've mastered flying, three different games test your skill. £10

OK. These games are all keyboard actuated, and you need no peripherals other than a casette recorder.

ORDER 3 PROGRAMS FOR *10% off the above prices.*

NOT
POLYOPTICS
10 ALSTONE ROAD
STOCKPORT, Cheshire
SK4 5AH

NAME__________________________________
ADDRESS_______________________________
____

Please circle the numbers of the programs you're ordering:

1 2 3 4 5 6 7 8 9 10 11 12   Subtotal:________________

*CHEQUES PAYABLE TO:-   Discount:________________
"STEPHEN SHAW"   Total:________________

Please send 50p for large software catalog (refundable on first order).

(Over 45 programs! listed).

OVERSEAS: ADD £3 per order

Paid ads in this issue cover programs from NOT POLYOPTICS and NORTON SOFTWARE.

As I have a small commercial interest I'll not be reviewing them - but I will be sending the games to C&VG Mag for their review.

As at the time of posting: 9th Aug 82 the supply position was:

NORTON SOFTWARE:
All EXTENDED BASIC programs in stock.
NOT POLYOPTICS:
Everything available EXCEPT:
Ships, Ant Wars, (sorry!)
'Winging It' is superb.

.. : ... ...

Personal Report Generator- this module supplements the Personal Record Keeping and Stats Modules.

With it, you may:

a) Delete files
b) Add files (at end of page)
c) Print out in a very controlled format. Of great use with 80col printers as a 'standard' letter writer. CAN be used with TP. (But NOT TP.U)

And speaking of printers- the Ed/As module can LIST your program for you- assuming an 80 char printer! On the TP it is unreadable. You can however list the SOURCE code separately, which looks OK on the TP.

REMINDER: The Ed/As package can ONLY be used with Expansion Memory AND a Disk System. The Mini-Memory requires ONLY the Console.

* * * * * * * * * *

If you have requested Norton programs in TI BASIC, a public apology- news was given on the basis of a telephone call - and when the package at last arrif ed, it came covered in PostOffice tape, with a note that it had been damaged in transit. No TI BASIC programs. I have requested replacement & MAY have them when you read this!

More gossip:

I have taken a trial sub to the monthly newsletter of the International Home Computer Users Association (Santa Fe). Trial sub is US$18.50 for 6 months.
Full membership which adds a biweekly Bulletin - all by airmail- is US$80 per year.
I'll report the results later, if you want the address:

ICA, PO Box 371,
Rancho Santa Fe,
California, USA,
92067

'COMPETITION'- Can anyone suggest a practical use for two 5¼" floppy disks which the postman has carefully folded in half before throwing through the letterbox? Prize:Two bent floppies!

NB: PLEASE -if posting a floppy- give it some protection huh!

Vic-20 in BOOTS is now £10 less than the 99/4A. Wow.

One bent floppy contained a chained extended basic program of stellar conquest. Sorry I cant report on it for obvious reasons. It is the first offering I've had where the chaining facility has been used (permits programs of unlimited length with console & disk drive only).
I've asked for a replacement & will report further.

Is there anyone out there would like to review a program,for publication in TI HOME? I have a couple of programs for review...
Free copy to first 5 letters- if you dont receive a reply,you've missed it! Why not send 50p for a catalogue...(thats called an editorial plug....).

The editorial content of Creative Computing is not so good with the new publisher

New publisher for Personal Computing World,same editorial team.

Rumour of changes at MicroComputer Printout too.

News that TI (were) at the PCW show at the Barbican - for release of the peripheral box, mini memory etc

Sorry this bit is blotchy- new ribbon in the typing machine.

BOUQUET: Special award for fast service to Action Computer Supplies of FREEPOST, Wembley, Middlesex, HAO 1BR.
One box of ten WABASH SS/SD 40 track disks ordered by phone at 11.48 delivered by first post next day, at a price VERY much lower than TI DISKS.

WABASH are new to this country- they have reinforcing rings, and have a good reputation in the States. (2yr guarantee)

I now have an Extended Basic program called SPACE RESCUE which is the best simulation of an arcade game I've seen.
The use of sprites is brilliant.
I am now trying to obtain resale rights- keep in touch!

The US software producers would like to know what programs UK owners want to buy. If you let me know what YOU want, I'll let them know.

Four branches of Micro-C are experimentally stocking some of my cassette software for the 99/4A in TI BASIC.
(Manchester,Birmingham,
Leicester,Southampton)

TI HOME Aug 82

SAVING DATA TO CASSETTE FILES:

Saving DATA to cassette files is fairly easy in TI BASIC, but there are one or two things to watch out for:

1. When you open a file and print to it, a blank header is first loaded (to pass over a possible leader tape)
2. When you wish to INPUT from tape, the computer will expect the header to be there!

THEREFORE: Always start your tape files at the beginning of the tape - this ensures the DATA is just where the computer expects it to be.

3. You can only open INPUT and OUTPUT files to tape.

4. Until you become used to file handling, stick to INTERNAL file types. The syntax of display files is a little more complex.

5. The OPEN statement for your data INPUT must be identical the the OPEN statement used for the OUTPUT - only OUTPUT becomes INPUT.

6. The following is suitable to open the files:

```
10    OPEN # 1:"CS1",INTERNAL, OUTPUT, FIXED

40    INPUT "YOUR NAME?":N$
50    PRINT # 1:N$
60    INPUT "YOUR TOWN?":T$
70    PRINT # 1:T$
80    PRINT "YOU ARE ";N$;" OF ";T$
90    CLOSE  # 1
```

```
10    OPEN # 1:"CS1",INTERNAL, INPUT,  FIXED

40    INPUT # 1:N$
50    INPUT # 1:T$


80    PRINT "YOU ARE ";N$;" OF ";T$
90    CLOSE  # 1
```

Need to rewind to open the saved file....

MEMO: ALWAYS USE 'INTERNAL' & 'FIXED'
ALWAYS REWIND TAPE AS INSTRUCTED

You may PRINT & INPUT more than one variable at a time, with each separated by COMMAS (,), provided the total data length does not exceed 64 bytes (each number takes up 9 bytes, each string one more than its length)

You CAN specify longer data fields- by using a number after FIXED e.g.: FIXED 128 or FIXED 192

Example: PRINT # 1: A,B,M$

Then your INPUT must also be: INPUT # 1: A,B,M$

The number of variables- and type - must agree.

This information should enable you to save at least SOME Data. From this start you should be able to progress by experimentation and rereading your Users Guide.

TI HOME AUG 82

TI99/4 vs 99/4A VERSION 100 vs 110 DIFFERENCES

There are several variations on the TI system which you must know about if you wish your programs to run on every combination.

Firstly, the different consoles:

The most important difference is in using the SPLIT KEYBOARD. In particular, you must not use keys returning values of 13,16,or 17. The two consoles use different keys for these codes.
If you use X & M, returning value 0, when testing for this value you should not directly equate with zero, as the 99/4A will return False. You should use,for instance:

```
If B+1 = 1 THEN ...
```

The A model has a different VDP chip. If you are developing an assembly language routine or program which addresses this chip you should test-run on a 99/4 if available, or indicate that your program/routine may not work with a 99/4.
As an example - with the A you may define 768 characters. This is NOT possible on the 99/4.

The old 99/4 only has available CALL KEY 0,1,and 2. The other CALLs will not cause a crash, but they will have NO effect.

The old 99/4 does not have a lower case set - the characters are blanks. If you use a 99/4 do not assume any blank characters- if you use an A,do not assume lower case will be printed.
For use on either console,you may wish to add the following,if your A program uses lower case (and if it is not vital to the program):

```
100 PRINT "PRESS ENTER"
110 CALL KEY(2,A,B)
120 CALL KEY(0,C,D)
130 IF (D=0)*(B=0) THEN 110
140 CALL KEY(2,A,B)
150 IF A<>13 THEN 1000
160 FOR A= 65 TO 90
170 CALL CHAR(A+32,CHR$(A))
180 NEXT A
```

CALL KEY(2...) is repeated in case ENTER is pressed when the computer is between 110 & 120!

In Extended Basic you may use:

```
.....
170 CALL CHARPAT(I,A$)
172 B$="0000"&SEG$(A$,1,4)&
 SEG$(A$,7,4)&SEG$(A$,13,4)
174 CALL CHAR(I+32,B$)
```

This will give the 99/4 the small letters.

VERSION 100 / VERSION 110 EXTENDED BASIC:

Only one major problem- the area the program is in is very slightly different - 24 bytes!
If you use CALL LOAD to amend a program line, you must use:

```
100 CALL VERSION(VN) :: VN=(VN-100)*2.4
```

For Version 100, VN=0 but for Version 110 VN=24
If the program is located at -1000 in Version 100 it will be at -1024 in 110:
If we start with a Vn 100 address we convert it to a Vn 110 address:
LOC=LOC-VN

Follow this guidance and there should be no problems!

TI HOME   Aug 82

PERSONAL RECORD KEEPING MODULE : Report Format.

Back in April I advised that when the PRK module asks you to specify your printing device, instead of just 'TP' you could enter 'TP.U.S.E' to save paper.

For many months, in my ignorance, I have been bemoaning the **awkward** way in which it printed out -in report format- data lines over 32 characters long.
For example- see right.
This was printed when 1,4,6,7 were requested- in one printout.

```
0       1
- ---------------
1 SAMPLE PAGE
2 SAMPLE PAGE
3 SAMPLE PAGE

0       4          6 7
- ---------------  - -
1 ONE              1 1
2 TWO              2 2
3 THREE            3 3
```

I have now found out how to print out the same data as:

Either:

```
0       1             2
3       4          5     6 7
- --------------- ------------
- --------------- --------- - -
1 SAMPLE PAGE
  ONE                    1 1
2 SAMPLE PAGE
  TWO                    2 2
3 SAMPLE PAGE
  THREE                  3 3
```

Or:

```
0       1               2
3       4         6 7
- --------------- -------------
- --------------- - -
1 SAMPLE PAGE
  ONE             1 1
2 SAMPLE PAGE
  TWO             2 2
3 SAMPLE PAGE
  THREE           3 3
```

These printouts involve telling the Module that the printer width is 64 and inserting blank buffer fields to put the data in the right place.

Here are the field specifications- for all 3 printouts -but different fields are used in each example. (See the numbers in the heading over the printouts).

```
         FILE STRUCTURE
-------------------------------

   ITEM          TYPE  WIDTH  DEC

 1 LINE 1A       CHAR   15     0
 2 BUFFER A      CHAR   13     0
 3 BUFFER B      CHAR    1     0
 4 LINE 2        CHAR   15     0
 5 BUFFER C      CHAR    9     0
 6 ITEM 1        CHAR    1     0
 7 ITEM 2        CHAR    1     0
```

You need to know:

1. Every data field printed is preceded by a space.

2. The printer sees each data field as a fixed length string.

3. If there is insufficient line for the printer to print the FULL data field on a line, that data field will be printed to start at the beginning of the next line (always beginning with that blank space!).

I hope this helps you make better use of your PRK module - I keep finding new uses for it,and this formatting capability is a real help.

Best wishes.

Stephen

CALL LOAD

or how an extended basic programme can alter its own lines....

VERSION A
suggested from an article
in 99er Mag

```
100 GOTO 140
110 PRINT "!!!!!!!!!!!!!"
120 END
130 STOP
140 CALL INIT ! POKE TEST
150 FOR B=-8 TO -24 STEP -1
160 READ Z
170 CALL LOAD(B,Z)
180 NEXT B
190 DATA 69,84,69,76,80,77,79,67
,32,84,83,69,84,13,199,156,17
200 GOTO 110
1000 FOR Z=0 TO -28 STEP -1 :: C
ALL PEEK(Z,A):: PRINT Z;A;CHR$(A
);:: NEXT Z
1001 END
```

(Use RUN 1000)

[2021: XB Vn110 see below]

VERSION B
suggested by comparison
to CALL PEEK

```
100 GOTO 140
110 PRINT "!!!!!!!!!!!!!"
120 END
130 STOP
140 CALL INIT ! POKE TEST B
150 CALL LOAD(-23,156,199,13,84,
69,83,84,32,67,79,77,80,76,69,64
,69)
160 REM NOTE IN REVERSE ORDER
TO PROG A
170 GOTO 110
180 END
```

This version an Original SJS creation!

NOTE: SWITCH ON EXPANSION MEMORY THEN CONSOLE. SELECT EXTENDED BASIC then key in your selected program EXACTLY as shown above. If you need to edit or amend a line, enter NEW and start again!

RUN the program then LIST it - notice any difference?

Prog B is neater as it allows you to LOAD a line in normal order.

NB: The last data item in Prog A is the number of characters in the line (including command chrs and including the number 17 itself. It is not necessary but is included to remind you that writing lines this way you may not alter the length of the line!!! If it starts 17 long it should remain 17 long!

Now, use these techniques in a program to use:
  DIM(n)   GOTO n   RUN DSK1 N$

Try activating/deactivating a line by amending the first command character to & from REM (try command char 131)

**** **NB: If you have VERSION 110, for listing A, in line 150, use 'FOR B=-32 TO -48 STEP -1, for listing B, line 150, use CALL LOAD(-47,156,199 &c...).**

With this type of procedure you can now place a 'HISCORE' for your games into your programs - and SAVE with the proigram instead of using a separate data file.
Don't forget to also put in your program a "re-set" feature!

ADVANCED PROGRAMMING: GRAPHICS MODE THREE
(Multicolour Graphics)

The programs which follow will only run if you have:
- a) Extended Basic
- b) Expansion Memory.

The 'Call Key' is important- when you press a key, the value you have loaded into CPU RAM address -31788 is transferred to VDP Register 1.
This SIMULATES Graphics Mode 3 - to make full use you must use Assembly Language, not Extended Basic.

In Mode 3, you have each character position divided into 4 small squares. Every small square can be any one of the 16 colours available.

As the final list illustrates, the Mode 3 Cc' Table is in the same area of memory as Mode 1 Character Dc.inition Table- which permits us to use those small squares in Extended Basic.

16	15
14	13
12	11
10	9
8	7
6	5
4	3
2	1

Where
1= Rightmost
defining position
& 16 = Leftmost

If you place a vertical column of 4 characters on the screen, giving a 2x8 grid of small squares, you can manipulate the small squares by redefining the character forming the column.

The VALUE defines the color (1 to F),the position in the definition string defines which of the 16 small squares is to be that colour.

The next project is to write a program in ExBas which can make intelligent use of this. NB: Note what ha)pens when you scroll.
NB: It is wise NOT to use CALL SOUND.
Keep pressing keys to return to normal afterwards. If the screen appears to lock out in a funny pattern,enter a letter and press ENTER -that should clear it. If all else fails,switch off.

Have fun with these:

```
100 PRINT "ONE MOMENT:REDEFINING
 A FEW CHARACTERS....."
110 B$="1234567890ABCDEF12341234
000123123"
120 FOR CH=32 TO 136
130 FOR C=1 TO 16
140 CH$=CH$&SEG$(B$,32*RND+1,1)
150 NEXT C
160 CALL CHAR(CH,CH$):: CH$=""
170 NEXT CH
180 CALL SOUND(500,440,0) ←before
190 CALL INIT                LOAD
200 CALL LOAD(-31788,204)
210 CALL CLEAR
220 CALL KEY(0,A,B)
230 IF B<1 THEN 220
240 CALL HCHAR(1,1,32+FLAG,760)
250 FOR CH=32 TO 95
260 CALL CHARPAT(CH,B$)
270 CALL CHAR(32+FLAG,B$)
280 FOR DL=1 TO 200 :: NEXT DL
290 NEXT CH
300 FLAG=FLAG+1
310 IF FLAG<100 THEN 240
```

```
100 CALL CLEAR
110 CALL INIT
120 CALL LOAD(-31788,204)
130 CALL KEY(0,A,B):: IF B<1 THE
N 130
140 B$="1234567890000ABCDEEEFF11
114448888FF123"
150 FOR T=32 TO 100
160 CALL CHAR(T,SEG$(B$,RND*20+1
,16))
170 NEXT T
180 FOR Z=1 TO 30
190 CALL HCHAR(23*RND+1,31*RND+1
,70*RND+30,4*RND+1)
200 NEXT Z
210 IF FLAG=2 THEN 150
220 FOR Z=1 TO 700
230 CALL HCHAR(22*RND+2,30*RND+2
,65*RND+32)
240 NEXT Z
250 FLAG=2 :: GOTO 150
260 END
```

```
100 CALL INIT
110 CALL LOAD(-31788,204)
120 CALL KEY(0,A,B)
130 IF B<1 THEN 120
140 CALL HCHAR(1,1,45,200)
150 FOR Z=1 TO 57
160 FOR X=1 TO 14
170 PRINT CHR$(Z+30);
180 NEXT X
190 FOR DL=1 TO 500 :: NEXT DL
200 NEXT Z
210 CALL LOAD(-31788,224)
220 CALL KEY(0,A,B)      ↖Normal
230 IF B<1 THEN 220        Value
240 END
```

```
40 CALL CLEAR
100 A$="00000000000000110000000
0000000000000000000000"
110 FOR A=32 TO 60 STEP 3 :: CAL
L CHAR(A,A$):: NEXT A
120 CALL INIT
130 CALL LOAD(-31788,204)
140 CALL KEY(0,A,B)
150 IF B<1 THEN 140
155 CALL HCHAR(1,1,32,770)
160 FOR I=1 TO 16
170 CALL CHAR(32,SEG$(A$,I,16))
180 FOR DL=1 TO 3E3 :: NEXT DL
190 NEXT I
200 END
```

CALL SOUND

CALL SOUND

Having read your manual and seen just how high a frequency you may specify with CALL SOUND, did you wonder why TI claim a music range of only four octaves?
Because when you go over that, the accuracy of the frequency suffers.

Using the notes D and D sharp, and starting from the D above the A above high C-

Intended Frequency: Theoretical Output Frequency:

1175.......1177
1245.......1243

---pretty close. Now, the same notes, going up an octave at a time:

2350.....2330
2490.....2486

4700.....4661
4980.....5085

9400.....9322
9960....10169

18800.....18643
19920.....18643

} no error in my typing here

Notice those last two - a bit out.

This is because the frequencies played are all denominators of a constant, and as the frequency rises, the number of playable frequencies drops.

This will affect the Call Sound routine I gave for playing sounds below 110Hz.(TI Home April 82).
For bass sounds the error is quite low, but as the frequency of the thfrd sound generator rises, so does the error! In practice you should not try to play a note over 300 Hz using this formula.

If you dont have TI HOME April 82, here is a little list for you to enter and try:

```
100 INPUT FREQ
110 CALL SOUND(400,FREQ,5)
120 CALL SOUND(400,30000,30,3000
0,30,FREQ*13.58,30,-4,2)
130 CALL KEY(0,A,B)
140 IF B=0 THEN 110
150 GOTO 100
```

NB: The computer plays a note for a duration of N sixtieths of a second- not in milliseconds. The minimum duration is thus 17ms -in theory. In practice, it takes time to evaluate the instructions!(abt.40ms).

It appears that the multiplier used gives an o/p approximate to the FREQuency.
The frequency of the third generator sets the 'shift rate' of a 15 stage divider network. With feedback. Thats what the editor/assembler manual says anyway.

TI HOME Alphabetical index to VOLUME ONE
(produced with Personal Record Keeping & Personal Report Modules)

A) **ISSUE #**

VOLUME ONE

B) **ISSUE #**

TI HOME

Alphabetical index to VOLUME 2.   Up to Issue 3 -June 82

VOLUME TWO

INDEX- TI HOME - VOLUME TWO continued

)PPRESSSTOPPRESSSTOPPRESSSTOPPRESSSTOPPRESSSTOPPRESSSTOPPRESSSTOPPRESSSTOPPRESSSTOPPR
)PPRESSSTOPPRESSSTOPPRESSSTOPPRESSSTOPPRESSSTOPPRESSSTOPPRESSSTOPPRESSSTOPPRESSSTOPPR

To encourage TIHOME members to submit articles & programs to the commercial micro press, TEXAS INSTRUMENTS will give you a module of your choice up to a price of £50 for each article or program published from now till 31st December 1982. Send proof of publication in the form of a photocopy of the magazine page containing your entry to R. FROWD, Texas Instruments, Manton Lane, Bedford, Beds.

After the finishing date Robin Frowd and Paul Dicks will decide which is the best of the articles and the winner will receive a language module of their choice, i.e. Assembler/Logo/MiniMemory/Ex Basic/Speech Synth.

This offer does not apply to employees of T.I. or TIHOME. Nor does Tidings count as a commercial publication.

)PPRESSSTOPPRESSSTOPPRESSSTOPPRESSSTOPPRESSSTOPPRESSSTOPPRESSSTOPPRESSSTOPPRESSSTOPPR

If you are willing to have your name and address published in Tidings and to be the focus of a regional group of TIHOME members who can meet and discuss programs, problems, discoveries, etc, please send me a postcard and I will publish your details. It would be good if we could form groups in all the major areas of the country. Perhaps after a few meetings each group could submit a few notes to Tidings so that everybody can keep in touch with what is going on in other areas of the country. A group doesn't have to be a formal organisation, rather a meeting of friends with a common interest. You can get together to channel common requests or complaints through TIHOME to T.I. I am sure they will be delighted to receive input from regional groups as well as from TIHOME as an organisation and members as individuals. So, get with it, if you work it right it will only cost you a few pots of tea.

)PPRESSSTOPPRESSSTOPPRESSSTOPPRESSSTOPPRESSSTOPPRESSSTOPPRESSSTOPPRESSSTOPPRESSSTOPPR
)PPRESSSTOPPRESSSTOPPRESSSTOPPRESSSTOPPRESSSTOPPRESSSTOPPRESSSTOPPRESSSTOPPRESSSTOPPR

CLOSE ROUTINE.

Well, that's the end of the first Tidings in the new format. I hope you think it was a good idea. I expect you will write to me and tell me if you think it wasn't.

One problem I have come across is that there is one TV on sale in this country that you cannot use with the 99/4A. It is called the Thompson and is sold by Boots the Chemists. This is not a failing on the part of TI but is because the TV in question is not actually English. It is a converted French SECAM model and just is not compatible with the PAL modulator that goes with the TI. So, you have been warned.

Membership continues to rise and now stands at about 400. I won't say that the higher the membership the more notice TI takes of us, but I'm sure it has some affect. As the membership rises I am having to introduce strict office routines or else I will loose track of what I have done and what still needs to be done. All members have a membership number already and pretty soon I shall let you into the secret. If you quoted it on your letters it would certainly help when I get letters with almost no address, signed 'Yours Harry'. While I appreciate Harry's letter, it would be useful to know who he is!

Well, I think I will leave it there, partly because it is 11.53 p.m. and partly because the newsletter is already late, and if I don't get it out to the printer it will be October before I know it.

I am off to TI for the day soon to have a long chat with our friends. If there is anything of interest (and I am sure there will be) I will be reporting on the meeting in the next issue of Tidings.

```
So PRINT "END MESSAGE"
   CLOSE#4
   STOP
```

God bless and
happy computing

Paul

Published by:-
TIHOME, 157 Bishopsford Road, Morden, Surrey.   Reg. No. 2778359
Printed by:-
THE SECRETARY BIRDS, 19 Coombe Road, New Malden, Surrey.