

Volume 3, Issue 4

September 1st., 1986

A PUBLICATION OF THE INTERNATIONAL TI-USER GROUP



Formerly OXON TI USERS

R PUBLICATION
OF THE
INTERNATIONAL
TI USER GROUP



PETER G. Q. BROOKS 96 BANBURY ROAD OXFORD OX2 6JT

OXFORD 510822

Index to articles	Page
BALDIE'S BURBLE	3
ADVERTISEMENT	7
C99 Release 2 from STEPHEN SHAW	_
CLARION CALL	8
Help Wanted - Apply Within	_
JUST PLAIN LOGO	9
Ol' Baldie attempts a graphic description	
WORD PROCESSING	12
The continuing saga from JD ANN COPELAND	
THE DREADED MAP	
Drawn by Baldie's own inimitable foot, and probably totally unhelpful, X doesn't mark the spot, and there is no pot of golat the end of the rainbow - but there might be a Workshop	ld
LETTERS	18
From JOHN ROE, a booboo in DEOOO4 (Chess), and from NIGEL CLEN a brief tutorial on how to get bewildered in one easy lesson	10NS
ADVENTURE HELP	21
Once again SCOTT and JO ANN COPELAND provide welcome relief to all those struggling with the transvestite bear on the wet trampoline with only a woolly sock and a spanner to build a rope bridge up to the chateau perched on the clothes line)
PACKET SWITCHING	25
NEVILLE BOSWORTH discusses PSS from British Telecom	
CONTACTS	27
Two more stalwarts who have placed themselves in the window	
BULLETIN BOARD	27
Roll up, roll up, get your luvverly bargains 'ere!	
99/4A DISK PERIPHERAL SOFTWARE OVERVIEW	28
COLIN HINSON continues his in-depth look at the disk system	
THE TIHOME SOFTWARE COLLECTION	32
The new pricing structure for disks full of programs	
THE NATIONAL TI USERS WORKSHOP IS TO BE HELD ON SATURDAY, OCTOBE 9.30 AM TO 7 PM.	
IF YOU HAVEN'T ALREADY PAID, IT WILL BE \$2 PER PERSON ON THE DOO	JR .

INTERNATIONAL TI-LINES: A PUBLICATION OF THE INTERNATIONAL TI USER GROUP

BALDIE'S BURBLE

July and August seem to have been fraught with more than their fair share of difficulties — or at least I have. Two consecutive disks full of current IT editorial, articles, and other support files have gone on holiday, never to return. Luckily I had backup files located at various strategic points along the way, but for some unknown reason my files have been going on the blink in a big way. It doesn't appear to be a drive fault, as one disk managed to make itself unreadable on many sectors just by sitting in a disk storage box, away from nasties like excessive heat, light and radiation... hmmm... blame Chernobyl ???

Anyway, lots of things going on as usual. After having seen an advert for full-blown, OFFICIAL flippies (yes, those things which I warned folks about using) made by DISKING (and you couldn't really miss the advert, adorned as it was with the slogan GO BANANAS and pictures of bright yellow disk boxes), I ordered 20 disks and have been putting them through their paces.

They are claimed to be compatible with double-sided, double density, 40 or 80 track drives, and at a very reasonable price. I ran one disk for over three hours on a comprehensive DM2 double-sided test with no errors at all, so hopefully they will prove reliable.

The nice thing about them (apart from all the sturdy packaging, the robust plastic storage/library box, and the daft badge with GO BANANAS on it) is that they permit me to retain a single price structure for just about all my software. In the recent past, I have had to charge owners of single-sided drives twice as much for Public Domain software as I did owners of double-sided drives, simply because the latter could get more on one disk than the former.

These "official" flippies change that, permitting me to use them either as double-sided, or as two single-sided but-as-one-flippy, reducing both my costs and yours at a stroke. Or something like that.

I have been nursing the sneaking feeling that all the adverse publicity concerning flippies (the home-made variety) was put about by the disk manufacturers as they saw their sales sliding; however, it just might be that Disking saw an opportunity: if drive owners were going to be daft enough to make flippies and use them regardless of the dire consequences well, who were Disking to get in their way? Why not make "real" flippies and milk the market for all it is worth?

Either way, it makes life easier for me (and my conscience), and if you have any objections to receiving material on flippies, well, I will just have to force myself to charge you twice as much as everyone else. OK ?

This month (August) saw the arrival of the first batch of RAMDISK boards from New Horizon Computers Ltd., and by the time you read this the lot should have been built. Thanks to considerable effort on the part of

GORDON PITT during a period of bereavement, the Hitachi HM6264LP15 chips which form the basis of both the MATCHBOX EXPANSION RAM and this latest project, the NEW HORIZONS RAMDISK, should be available at a very low price indeed. If you have been considering building your own 32K RAM following the "Westralian Instruments" design, then you should be able to take advantage of an extremely good offer.

The final prices for the New Horizons RAMDISKs have now been finalised:

For the DOUBLE-SIDED equivalent RAMDISK, £170 inclusive.

£160 as a kit

For the SINGLE-SIDED equivalent RAMDISK, £150 inclusive

For the BARE BOARD ALONE, with manuals and system disks: £40 inclusive.

The HM6264LP15 are available at £2.75 per chip, with an overall post and packing of £1.50 (unless you collect them!).

If you have any queries about the above, please contact either GORDON on 0922 476373, or 01' Baldie on 0865 510822.

Thanks also to considerable wheedling on Gordon's part, GEC have kindly given him, on extended loan, two modems for "laboratory testing", and he has kindly passed one onto me in order to support the up-and-coming ITUG TIBBS. I already have almost all the additional system set aside to run the BBS, and I recently ordered some BBS Operating Software from RALPH FOWLER based on a recommendation by STEPHEN SHAW. Problems with one of the disks have necessitated re-recording (and the attendant postal delay as the disk trundles back and forth across The Pond) so it will be a while yet before initial trials can begin.

The first runs will be on my domestic line during cheap rate (and most probably after 11 pm) at 1200, with Gordon providing major assistance in testing. Once the initial bumbling phase is over (i.e., my bumbling!), modemised subscribers will be welcome to help out by acting as guinea pigs (or should that be 105p pigs...). Watch this space!

Ol' Baldie has also been busy on the etymological front, inventing new words to add to the English language. You probably noted the use of one of them in the last Burble when I talked about JONATHAN TYLER and his numeric keypad on page 5. I invented the special word "comlpeted", but I have yet to devise a meaning for it. Perhaps as in "Ol' Baldie's a comlpete berk "?

Now, peek over the page for details of the TI Users Workshop...

YES. THIS IS IT!

THE NATIONAL TI USERS WORKSHOP

ORGANISED BY THE WEST MIDLANDS TI USER GROUP

AND SUPPORTED BY ITUG

Well, almost. The response to the idea of the Workshop has been barely lukewarm, which was expected, but you have to live in hope, don't you?

However, enough enthusiastic people have committed themselves to putting in an appearance, and grateful thanks for their support. It will be more like a large club meeting than a full-blown Workshop, but in many respects that is better for all concerned. In large numbers people do tend to become faceless, and the pressure can be such that anyone who is actively involved in answering questions can feel as though their grey matter is being put through a blender (I speak from experience...).

The details are as follows:

*

DATE: SATURDAY OCTOBER 4TH 9.30 AM TO 7 PM

VENUE: SNEYD ("SNEED") COMMUNITY SCHOOL (a rough map for car drivers is included elsewhere).

CATERING: Insufficient numbers to warrant hot meals being available, but you should be able to purchase sandwiches and coffee to keep yourself on the go. If you require sandwiches, contact GORDON PITT on 0922 476373 and tell him, so that provisions can be made. If you can't reach Gordon ring me (Ol' Baldie).

If you haven't sent in the appropriate cheques by the published date (the end of August) you can still attend, but you will have to pay at the door: £2 per person. This seems only fair to those who were prepared to provide their support by sending in cheques even before they knew the date of the Workshop!

In addition to the facilities notified in the last issue, we will also be able to present MIKE GODDARD and his self-built expansion system, and DAVE HEWITT with his self-built expansion system with disk drives.

The nearest station is WOLVERHAMPTON, and there is a bus service between a road near the Community School and the bus station, which is near the Rail Station. The bus number is 560 and is apparently known as the "Bloxwich Bus". You ask the driver to tell you when the CRAB LANE stop is reached (or the stop AFTER the one at "the square at New Invention"). The cost is 32p one way. Crab Lane is right at the bottom of the map, and it is a few minutes walk from the Sneyd Community School (which is signposted).

Unfortunately, although the School is within spitting distance of the M6 and there is a junction nearby (10a), there is no connection (the link is to the M54!) and junction 11 is the nearest. Take the WILLENHALL (WALSALL) turning off the roundabout, and follow the road (WARSTONES ROAD) looking for the fourth exit on the left which is SNEYD LANE. This is a small crossroads, and don't be confused by the fact that Warstones Road becomes BURSNIPS ROAD on the way — it'll all come out in the wash.

About 800 yards along Sneyd Lane is a signpost for the School on the left, pointing to VERNON WAY.

Find a spot to park the car (there should be plenty of space), and walk round to RECEPTION, where (depending on your arrival time) you will be greeted by GORDON PITT (or someone), OR... you may have to ask at the office, when you will hopefully be guided by a native runner...

PLEASE REMEMBER: if you are not on the list of "we paid £1.50 in August" you will have to pay £2 PER PERSON at the door.

If you come by bus from the station (should you be so intrepid), you should head for RECEPTION as above. I am advised that the bus driver may smile happily upon you if you tender the EXACT fare so he doesn't have to search around for change!

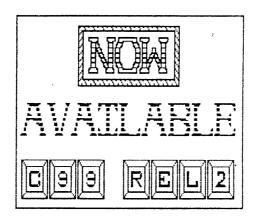
Elsewhere you will find CLARION CALL, which I have gone ahead and published despite having received a helpful letter from NIGEL CLEMONS on the subject of my C trials (sea trials, geddit? Oh, never mind...), on the grounds that its general principle still applies.

I have stuck in a shortie on LOGO II, not, I hasten to add, as anything even remotely resembling an Expert, but just to try and awaken a few interesting souls with a view to coercing them into providing a little of their wisdom for us all to share.

SCOTT and JO ANN COPELAND are proving themselves to be valuable assets to ITUG, as the columns on Adventure Help and Word Processing attest. A number of callers have mentioned their work, and although I am no Adventurer, I find myself reading the column (yes, I know I'm supposed to when I proofread it!) Just because I find the clues intriguing and well-presented. Five stars, I think?

Scott is also producing a Forth Tutorial, which has yet to put in an appearance because of my own failure to find time to proofread the first episode. Every time I sit down to perform my allotted task, something else crops up needing immediate attention, so my apologies to Scott and to any Forth first-timers — I will have it out as soon as I can (no naughty comments, please).

This issue is going out at least a week late, in order to follow the end of August deadline, and is of necessity a bit garbled. I hope to get back on a more even keel by the next issue, but as that will be out at the same time as the Workshop, I'm making no promises!



C99 REL2 is NOW AVAILABLE
On TWO DISKS from
Stephen Shaw
cone Road, STOCKPORT, Cheshire

10 Alstone Road, STOCKPORT, Cheshire, SK4 5AH (Usual Disk Library cost)

This latest release of c99 from Clint Pulley is now fully compatible with FUNLWRITER, and memory image files generated from source code compiled by this release WILL run from FUNLWRITER.

C Functions included:

CONV;C String (-) int conversion

functions

FCOPY;C Simple file copy program
RANDOM;C Random number generators
SOUNDS;C Simple beep and honk fcns
BIT... Bitmap graphics functions

and documentation

FLOAT... Floating point functions

and documentation

SPEECH... Speech function and documentation

STRINGFNS Library of string

manipulation functions.

AND MANY OTHERS.

C99REL2A includes version 2.0 of the c99 compiler, console and file I/O libraries and functions, the RUNOFF text formatter, and documentation.

This release includes the standard formatted output functions in object form. printf, fprintf and sprintf perform as in standard C and support c, d, o, s, u and x conversions. A REF directive (cf manual) must be included for each function used. At load time, PRINTF must be loaded if any of these functions are used. If sprintf is used, SPRINTF must be loaded too. If fprintf is used, both FPRINTF and CFIO must be loaded. If you know of anyone who has received an earlier version of c99, please feel free to provide them with this version. CAUTION: Object files produced by earlier versions of the c99 compiler are not compatible with c99 v2.0. You should re-compile and re-assemble all functions and libraries to ensure compatibility and to take advantage of the shorter, faster code generated

CLARION CALL

CLARION CALL is an occasional feature in which anguished souls like me can issue a plea for those with not necessarily mastery, but at least familiarity with certain packages, to please pass on just a morsel of their experience.

A classic example of the need for such morsels is provided by the untutored efforts of Yours Baldly to initiate himself into the mystical rites of C99 (and hopefully later PILOT 99).

I made some extremely elementary (and I mean ELEMENTARY!) mistakes which I could have avoided if someone with even a week's more experience of C99 had paved the way for me to compile, assemble, load, and run a childishly simple first attempt at C programming.

In the not-too-distant past I have bemoaned the fact that I had long since forgotten what it is like "not to understand" — in other words, I can't remember what it is like to be a "beginner".

This applies largely to BASIC, but my recent tussles with C99 have brought it all back to me with a vengeance. I can safely say that I know EXACTLY what it is like to be a beginner...BUT... not a beginner at BASIC, only one at C99! (This lets me off the hook as far as any foul-up in BEGINNING BASIC goes. Doesn't it ?)

I am sure that there are a large number of us (yes, US) who would dearly like to get our grubby little paws on things like C and PILOT (and Forth for those who like wrestling with RPN - myself included), but who are almost put off by our failure to even manage a tiny little program in any language other than BASIC.

It doesn't take much, but there is one very important factor: whatever we get shown or told during the very early stages MUST be accurate. The few things we learn early on tend to stick with us for ever, so that if we get pointed in the wrong direction at the start we're going to take a very long time to get back on the right course.

So, if you reckon that you can stretch a helping hand down to First-Timers, you DON'T have to be Expert, you DO have to be Accurate, and you will have to have the patience of the proverbial saint (or Traffic Warden).

There may well be those among you who read this and think: all the twit has got to do is read the manual/documentation! It's all in there!

It may well be. That doesn't mean to say that it will be immediately comprehensible to all, and it doesn't mean that everyone will interpret every sentence in exactly the same way.

So how about someone lending us a little of his/her experience and writing a page or two (literally) on C99 and/or PILOT 99 ?

All contributions gratefully received ...

JUST PLAIN LOGO

Ever since a kind person lent me LOGO II (without manuals) I've been dying to try something out.

Many moons ago, when I still had hair on top and none on my ugly mug, with the aid of a soft pencil, a chequer-patterned Formica-topped kitchen table, and a complete lack of awareness on my parents' part (not through deliberate deception on my part - they were Just not around when I put pencil to table-top, and I always erased my efforts!), I played around with a set of rules for producing a graphic image.

Later, (much later), I produced a painfully-slow and not very representative TI BASIC program to do a similar thing and published it in my Babble in the late TIHOME's Tidings (back copies of which can STILL be bought from me, by the way).

The other evening, after a marathon session on the phone (I picked it up to answer a call around 7 pm, and finally managed to lay it back on the table at 11.30 pm after several more), I decided to apply my limited language skills (see elsewhere, about C99) to LDGO II, bereft of manuals though I was.

With the aid of an August 1984 issue of Home Computer Magazine (also still available from me – I must put together an index of the useful topics covered in that particular issue and publish it in IT), plus vague memories of an article produced by BETTY LUMLEY in Tidings, coupled (puff, puff) with a recent re-exposure to LOGO when attempting to demonstrate it to recent subscriber ALAN NEWMAN and family... where was I ? Oh, yes – I began trying to uncover some of the commands.

I knew already TELL TURTLE, RT, LT, FD, BK, PENUP, PENDOWN, how to define and edit a procedure, and a growing understanding of REPEAT [].

After much trial and error, I worked out variables (:N), assignment (MAKE "C :C+1), how to clear the screen (CS), and how to SAVE procedures to disk (it took a great deal longer to uncover RECALL!!).

I now had the basic essentials in order to produce a working program, and I present my efforts here NOT as an expert, nor even as an introductory guide to getting started with LOGO II, but simply because no-one else has bothered to submit ANYTHING in LOGO to me.

Maybe the more adept will point out my errors, and the more confident might be induced to submit something better (as in "Good God! Even I can do better than that bald twit!").

Principles of the procedure

The procedure is very simple, is subject to endless variation, and can produce figures so large and/or complex that LOGO II frequently runs out

of ink (can somebody PLEASE design a version for the 4A using Bit-map mode ???), which annoys the pants off me.

All one does is this:

- 1. Pick a starting point
- 2. Move forward one unit (of distance)
- 3. Turn through V degrees
- 4. Move forward two units
- 5. Turn through V degrees
- 6. Move forward three units

Move forward NNN units Turn through V degrees Go to (2)

My original approach was to use 45 degrees for V, and to use the length or diagonal of a chequer-square on the table-top as one distance unit.

Because turtle graphics give a pseudo-high-resolution drawing facility, I used one pixel to represent one distance unit.

Initially I did each drawing on the screen manually - that is, I typed in directly FD 1 RT 45 FD 2 RT 45 FD 3 RT 45 FD 4 RT 45 and then pressed ENTER, and then did it all over again. The images began to resemble those which I had created on the table-top some 20 years ago, so I resolved to try larger and larger maximum values to be used with FD.

This in turn necessitated the use of variables and REPEAT [], and to cut an interminably-long story short, here is the finished procedure, giving opportunities to vary both the maximum distance unit and the repetitive turning angle. (There are several points which arise from this, which are discussed over the page).

TO I :N :A CS REPEAT :N [MAKE "C O REPEAT :N [MAKE "C :C+1 FD :C RT :A]] END

Select the turtle screen with TELL TURTLE. The procedure is activated by typing I maximum-unit angle and then pressing ENTER:

I 11 45

The above example means that the procedure will produce segments of lengths 1 to 11 before repeating, each segment being turned 45 degrees to the right with respect to its predecessor.

I leave the procedure's fine detail unexplained, for anyone with the determination to request clarification, (or to wipe the floor with me

by showing a better way!).

I don't have a Load Interrupt Switch, nor the software to convert the screen images into hard copy, and most of them are beyond my ability to draw them (the most I managed on the table-top was an :N of 13 with an :A of 45 degrees), so unless something useful crops up before this goes to press (unlikely), the uninitiated LOGOless reader will remain so, until such time as I can (a) produce output using a computer-controlled turtle robot, or (b) find some software to convert the graphics (query tile definitions ???).

Discussion

Almost all the variations are cyclic. (Which ones aren't, and why, can be left to the intrepid LOGO Adventurer - or next issue...).

In some cases, the turtle begins to cover the same tracks again, in others it finishes the procedure without having completed the figure.

You can either amend the first REPEAT :N to REPEAT :N*2, or discover the relationship between :N, :A, and completion (or over- or under-completion), and let us all know (or me at any rate).

In some cases, you may be surprised (and disappointed) at how simple some of the images are.

To begin with, try using an :A of 45 degrees, and vary :N from say 2 to about 25. In many cases the dratted turtle will run out of ink, which is a pain.

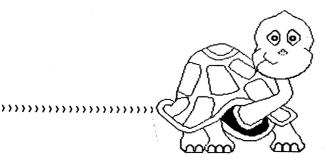
Then try altering the angle used - remember that angles over a certain value have the effect of turning to the LEFT instead of the RIGHT (what is the "certain value"?).

Finally, one other thing you can examine is a single component, expanded until it runs out of ink. Try I 1000 87; or I 1000 88; or: I 1000 89; I 1000 145; I 1000 155; I 1000 166; I 1000 170; I 1000 179.

There are many others.

You might even modify the inner REPEAT [] to incorporate another FD and an LT section at some point (otherwise you'd just get a wobbly, but almost straight line!) to add kinks and possibly investigate another avenue.

Any positive feedback which this may generate will be eagerly received!!



MORD PROCESSING

By JO ANN COPELAND

Well, for those of us who were new to the TI Writer/Funlwriter we've come a long way, but as usual, there's a lot more to come! Cringe, grovel, hide, load up an Adventure...

We'll start off this 'tutorial' with a very 'moving' experience - okay, let's clean our minds up and get to some serious business...

We will be investigating the Lines Commands this month. If you load up your TI Writer/Funlwriter you would see the usual prompt...

Edit, Tabs, Files, Lines, SearcH, RecoverEdit etc.

After loading up any practise document (Command Mode, LoadFile, etc.), you would enter command mode, type L (Lines) and press ENTER to display the Lines prompt. This should look like:

Move, Copy, Delete or Show lines

First, we'll look at Move. If you have a practise document loaded, and would like to move a line or a few lines, pressing M for Move and then ENTER will display another prompt:

MOVE start line, stop line, after line:

Okay, looking at your practise document, write down the line number of the first line you wish to move. If this is one line, one line number will suffice. Then, write down the line number that you wish to stop at. If you have more than one line you wish to move, it would be something like - 34 48 (start line and stop line). Then, select the line number that you wish to place these lines AFTER. So, if you have line numbers 34 to 48 and you wish to move them after line 10 you would type the following after going to command mode:

MOVE start line, stop line, after line: 34 48 10 (Press ENTER)

Depending on the length of lines being moved, the computer will take a certain amount of time (depending) and then you will automati- cally see the changes made on the screen. This saves you re-typing any documents because you may have typed a paragraph in an area different from where you really wanted it to go, etc. The good thing about the TI Writer/Funlwriter is that it takes you step by step on each command and if you read as you go, following the prompts and questions, you really shouldn't go wrong. Alas, I am here to disprove that theory!

Okay, let's go on to Copying lines with the Copy Command. Almost the same in theory: you go into Command Mode, press C for Copy and then press ENTER. The display comes up:

COPY start line, stop line, after line:

So, let's say you were typing a document that required more than one line of the same thing. (Huh?) For example, my last example of this was making a graph enclosed by the (*) sign. I typed my first couple of lines:

0059	***********	+** **
0060	***	***
0061	***	***

Now, I want to extend the second and third line without having to actually type it. So, I go to command mode, type C and ENTER. I choose my first line to copy (60) and the stop line (61). I wish them to be placed after line (61). Now, not everyone's line numbers will be the same, but mine show up with these line numbers. Okay, after pressing ENTER I would see the screen show up with:

0059	*********	*************
0060	***	***
0061	***	***
0062	***	***
0063	***	***

If I wanted to continue copying these lines, I could now enter command mode, C (ENTER), type start line 60, stop line 63, after line 63, then four more lines would print up on the screen. So, any line(s) you wish to copy, without typing over and over again, can be accomplished with the Copy Lines command. Lost yet?

Remember, prompts give you the questions in order, and if followed, will take you through step by step.

Delete Lines works pretty much the same way. Enter command mode, Type D and ENTER. The prompt will ask you: DELETE enter start line and stop line. Just type in the line number you wish to start deleting, the Stop line where you wish to stop deleting, and hit ENTER. The computer will take a few minutes, depending on the amount of lines being

deleted, and then will show on screen your finished product.

Last but not least, is Show Line. Entering Command Mode, L (Enter), S (enter), will show a prompt asking: Show LINE, enter line number. If you enter any line number and press ENTER the line will be shown to you at the top of the screen with the cursor at the beginning of the line.

That's it for the Lines Command. Easy, huh? Okay, now let's discuss Loading Parts of a File. Previously, you learned how to load an entire file. Now, we can also load part of a file with the LoadF command. Here again, a start line number, stop line number, and a filename are required. As usual, separate the start, stop line numbers by a space (in each case). Now, if you have something on screen — it will be lost and replaced with the new line numbers being pulled up. If you wish to keep your text already on screen, you should then use the Merge Command discussed earlier. However, if you just want part of a text pulled up, here's how we do it:

Command mode, LoadFile (LF), ENTER. Type in the following: Start line number (say 15), (SPACE), stop line number, (say 45), (SPACE) and DSK?.FILENAME. Then press ENTER. You will shortly see on screen that part of the file you requested.

To confuse you more, instead of merging two complete files, you can also merge your on-screen file with a part of another file. Enter Command Mode, LoadFile, and then the following: Line Number of current text after which you want to merge your new file lines. Then, line number of first line of part to be merged, then line number of last line of part to be merged, and then the File Name of the file to be merged. UGH! Let's check that out...

Of course, it will all be very confusing at first. But writing down line numbers, and practising these options will eventually lead you to doing them without looking at what you are doing. Practise makes perfect! Of course, in my case, I still look at all prompts and still write down line numbers! And I often save a file before I do anything in case I wipe myself out! So practise these for a bit, and you will see in a while you have it down pat!

For further confusion, loading full files, parts of files, etc., can also give way to saving parts of a file. Perhaps you typed a text, and after finishing it and printing it, etc., decided you wanted to save a

particular part of text for future reference or whatever.

Rather than save a whole file you don't need, just save the particular lines you really want. For this, we use the SaveFile (SF) command.

Again, you will need the line number of the first (start) line and the last (stop) line number, along with the File Name. Check whether you wish to save these under your current Save File Name or another.

If you saved your whole original file under DSK?.LETTER, and then save only a few lines under the same name, the original LETTER file will be overwritten to show the new file saved.

Okay, let's go. Enter Command Mode, and choose the prompt Savef (SF). On prompt, type in the start line number (SPACE), stop line number (SPACE), and then the File Name under which you wish to save the lines.

Then press ENTER. For example: Start Line Number (55) (SPACE) Stop Line Number (88) (SPACE) DSK?.LETTER1 (ENTER). On screen you will see those lines remaining and they would have been saved onto Disk. Thus, you have only saved part of the text you originally had on-screen.

Now, after you have saved, practised, etc., you may want to take those files off your disk. DeleteF (DF) can do it for you. Enter Command Mode, choose DF, enter the File Name of the file you wish to delete and hit ENTER. So, Command Mode, DF, DSK?.LETTER1. The file will be deleted from the disk.

The last of this column will be Printing a File from the Text Editor. If you have a text on screen and wish to print it out for proofing without leaving text mode, enter Command Mode, PF, ENTER, and the prompt will show: PRINT FILE, enter devicename. Mine being PIO, I type that in and press ENTER. My text will go to the printer and I can proof it before doing anything else.

So, it seems we have quite a bit to keep us busy and to practise on.

Right now, I am going to proof this, and then step by step verify everything to be sure I (a) said everything correctly, (b) that it works. (c) know what I'm talking about.

We haven't too much more to go into, but Fixed Mode, Text Formatter, and the dreaded Mailing List will be discussed in the future. Until later then, happy word processing. Don't get discouraged if something doesn't work at first — just keep practising until you get it right, and remember to follow the prompt lines — they are there to help you. (And me!)

So keep at it, and until later, have fun. Will see you next issue - if you're unlucky!



word Brend for S SKETCH - MAP MARSTONE SO JCTN. 11 M6 g Bheatigus HILTON BY CAR HOBINGLY EROPE LANE BLOXWICH TI USERS WOEKSHOP B SNEYD COMMUNITY SCHOOL UPPER SNEW VBRHOH WAY HEATH CAUTION: ROADS WITH THIS NAME!

LETTERS

26.8.86.

Dear Peter.

Thanks for the Demo disc recently received.

There are a couple of boobs in the chess program (DE0004). (Now then, don't all rush to look at it). The programmer has defined the symbols for King and Queen the wrong way round. The conventional symbols used in chess diagrams all over the world have the King with a flat topped (or slightly domed) crown with a little cross on top, whereas the Queen has a spiky crown. In the program these are the other way round.

I have amended the program to correct these errors. It was not easy, as the program is quite complicated. My first thought of finding the character numbers for the symbols (4 for each) and swapping them round didn't work and had all sorts of funny effects. The only way to do it is to exchange the program lines defining the characters for the Kings and Queens. The following pairs of lines have to be exchanged:-

White pieces: 1470 <---> 1670 1490 <---> 1690 1510 <---> 1710 1530 <---> 1730

Black pieces: 2810 <---> 3010 2830 <---> 3030 2850 <---> 3050 2870 <---> 3070

The easiest way to do this is to load the program in ExBas and then use the REDO facility to change the numbers round. Then save it under a different name in case you've made a mistake and want to get back to the original. Reset and select TI Basic and then run. You should now find the pieces correctly depicted.

This is a program that cries out for conversion to ExBas so that sprites can be used for the pieces along with the other advantages from using ExBas. Unfortunately, the starting position requires 32 sprites and ExBas will only give you 28. There are some ways round this but it does mean some departure from the simplicity which the use of sprites would otherwise give you.

Another idea I have in mind when I have time to get around to it, is to use Minimem with 32 sprites and putting the moves of a game into records of a data file to be stored in the Minimem's 4K RAM. Many games could be used by this method, saving to or loading from tape using the Easy Bug. This is a more portable arrangement which I could take to the club to use for demonstrations.

Yours sincerely,

John Roe.

NIGEL CLEMONS wrote to me with a few words of advice concerning my C problems (C sickness ?):

Small C and Baldie's Burble

The manual supplied with CLINT PULLEY's C99 is a little confusing for the Beginner, as Peter is finding out. To use PRINTF, etc., the manual says that a REF directive (see manual) must be included for each function used. Well, what IS a REF directive?

The I/O FUNCTION DECLARATORS (i.e., REF directive!) make use of a small integer value, called a FILE DESCRIPTOR (fd). For a complete description of I/O functions see chapter 17 of The Small C Handbook by J. E. Hendrix (isn't he the one who used to play the guitar with his teeth? PB)

Following is a list of these functions including PRINTF at the start of the file STDIO:

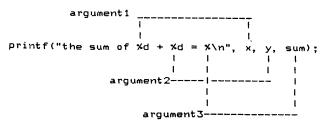
#asm

REF FOPEN, FCLOSE, GETC, PUTC, FGETS, FPUTS, FREAD REF FWRITE, FSEEK, REWIND, FDELET, FERRC, FEOF REF PRINTF #endasm

Certain symbols are used both for file descriptors and for values returned by these functions. They are defined in the file STDID, which should be included in every program. Following is a list of these symbols:

```
#define stdio
             -1 /* fd for standard input file */
#define YES
              1 /*
                                           */
              O /*
O /* value of a null character */
#define NO
#define NULL
                                           */
#define EOL
             10 /* end-of-line return value */
#define FF
             12 /* clear screen and home cursor return value */
            8 /* cursor backspace return value */
#define BS
```

Formatted I/O, the printf function { printf("control string", argument1, argument2,...) }. Where the control string can be ordinary text characters, such as "hello world", or conversion characters to specify what and how data should be printed. The conversion character informs printf how to convert the corresponding argument into a character string for output. As follows:



```
Following is a small printf program with all the terminating letters b, c, d, o, s, u, and \mathbf{x}_{\bullet}
```

```
#include dsk1.stdio
main()
{
    printf("\n", "Formatted Display with printf\n");
    printf("octal %o,decimal %d\n",127,127);
    printf("%u=%x\n",-1,-1);
    printf("%d% interest\n",10);
    printf("(%6d)\n",55);
    printf("(%-6d)\n",123);
    printf("The letter is %c.\n",'A');
    printf("Call me %s.\n", "NIGEL");
    printf("Call me %8.3s.\n", "NIGEL");
}
```

I hope that the above clears up some of the problems on using printf in Small C. $\label{eq:small}$

ROVENTURE HELP

By SCOTT and JO ANN COPELAND

For those Adventurers who require help - a question is often worded "Where is the last treasure in the Adventure ____". Well, there is no real last treasure. Treasures are not always found in any particular order, and ending an Adventure does not always count on finding treasures in a certain order. So, whenever asking for help from a friend, please state what treasures you have found. That way, it can be narrowed down to which treasure you do not have. Simplifies matters for everyone!

Also, many thanks to those who have written in requesting advice on a particular Adventure. Your questions help in the column, and often give rise to new thoughts for providing hints! As I am not a 'name-dropper' so to speak, if you write in please include a statement that says you do not mind if your name is used within a column. Again, thanks for your very kind letters, and also for your support!

One question posed was how you knew what a prompt word actually meant. For example, in Scott Adams' series, you may use three or four (sometimes five) letter words in your prompt, such as BREA MIRR. Now, does BREA mean BREAk or BREAth or what? All I can figure out is — it means whatever the author meant it to. APL language set—up, etc., only requires that the first allowed number of letters be typed in, so if the author wanted BREA to mean BREAK, that's what it is. Unfortunately it can be confusing for those of us trying to accomplish one train of thought, and using one word find it means something else. This can make the Adventure challenging, or curse—provoking, depending.

Either way, keep on Adventuring, and keep up the questions, and I hope I may help as many people as I can! Call it repayment for all the times I required help! Now, on to Adventure Help!

MYSTERY FUN HOUSE

GHOST-TOWN

Number of Messages: 100
Number of Objects: 77
Number of Verbs: 104
Number of Nouns: 118
Number of Locations: 38
You can carry 7 objects

Starting Location: 1
Number of Treasures: 13
Location for Treasures: 15
Length of Nouns/Verbs: 4

TREASURES (13): Cashbox, Silver Spurs, Oriental Go Board, \$200.00, Golden Derringer, Gold Coin, Turquoise Necklace, Sacred Tom-Tom, Gold Nugget, Pelts, Silver Bullet, Silver Cup, Bag Gold Dust.

VERBS:

AUTO SAVE SHOOK SCREW BURN WITH KILL BREEP PASS FILL	GO QUIT INVENTORY EXAMINE WHISTIE IGNITE USE CLAP SMASH TAPE CONNECT WAIT SHOE	WALK GET DROP DESCribe SING LIGHt DIG APPLaud BUST WEAR SPLICE CLEAR KNOCK	RUN TAKE PUT CHECK MIX OPEN SHOVe1 READ JUMP PUTOn REPAir COVEr TOUCH	CLIMB PICK THROW HELP MAKE LOCK HIT UNLIGHT LEAP FIND MOVE EMPTY FEEL	ENTEr GRAB LEAVE SCORE SMEL1 CLOSE RING KILL CUT LOCATE PUSH KISS	MOUNT COLLect REMOVE SAY TASTE SHUT BEAT LISTEN SLICE SEEK PRESS DUMP SPUR	RIDE LIFT TOSS YELL EAT LOAD PLAY HEAR SHAKE CROSS TAP DANCE EXTINGUISH
---	--	--	---	---	--	--	---

NOUNS:

ANY AROUND POWDer MAGNet KEYS WIRE BRUSH GIDY DOOR BAG SHOVel GO NUGGet SHACK DUST	NORTH STAB INVENTORY BELL MAP STORE MINE GITY WINDOW OFFICE BOX BOARD KEG TRAI1 HAMMER	SOUTH MANUTE MATCHES GHOST CANDIE JAIL ENTRANCE COIN SPURS PAINT CASH PATH NAIL MOUNTain TEEPee	EAST PILE ON FIGURE LIGHT ROOF BED TOWN HAT SLEEP ROAD HELLO FUSE FLOOR TOM-tom	WEST HILL SAFE HAND MUSIC SIGN HOTE1 HOLE STRINGS SNAKE FORK HI TOPPEr PLANK HOW	UP SULFa MIRROR FEET BULLet BARBER ROOM SALOON TAPE RATTIER VAIN FIELd LOBBy PELTs COUNt	DOWN GRAVE COMPASS HEAD GUN SAGE GIDDy STAL1 ROLL WORM PIANO CRYStal RAVIne CUP NECKlace	GAME GUNPowder HORSe KEY DERRinger TUMBleweed GITT CHARCOal WALL SLIMe \$200 GOLD RIDGe TELEgraph
--	--	---	---	--	--	--	---

Who in their right mind, except for those needing a place to sleep and 3 meals a day, would want to actually get in to a Jail? An Adventurer would, of course! In this case it's more easier said than done. Having no outside latch and finding it locked you could climb through the window. And then finding you can't do that, what else? In this case the answer lies within the horse shoe and the compass. Examine them both and see what you can do.

subject. He ain't going nowhere unless you fix him up first! Try examining 'Ole Paint, and seeing what he needs, provide him with the means...

SAVAGE ISLAND PART I

Getting together the things you need (log, vines, knife) you can find your way to building a raft, and eventually setting sail. Since you really aren't such a good carpenter, stock some spare parts for repair. I have a feeling you'll need them. A password later required can be found on the Atoll, and using the password in the second cave helps you solve your Adventure. Be sure to sail to the Atoll after the cannon does its thing.

The force field in this Adventure is nothing compared to Part II. But this one helps you the best - carrying an item with you through it offers you the chance to actually see what you are doing!

GOLDEN VOYAGE

Took me a while (and some help) to realize this Adventure had more than one island to sail to. And not only that, you find yourself going back and forth to each one! Fortunately, you have a well-established ship! You'll also need some further supplies - provided for you if you can find them! On one island, try digging, and if at first you don't succeed, try, try again.

A hidden passageway awaits you - besides an overgrown Cyclops! To find the passageway, and then a route, be devout in the altar.

ENCHANTER

Spells, spells and more spells. Some you will put into your book, and some you save for a better use at a later date! The Melbor and Ozmoo spells help you survive - literally. A trip to the altar in the Temple might gain you points when using these! Plus another object!

Most often, to admire great works of art, you need light on the situation to see what you can see. In this case, no light also helps you see what you can see! Yes, it makes sense...

HITCHIKER'S GUIDE TO THE GALAXY

I'm a true masochist! I walked through this Adventure twice within 8 months! Now I can truly order an "I got the Babel Fish T-Shirt"! Don't forget - you'll need your towel, gown, satchel, and junk mail. In this case, junk mail actually pays off!

When aboard the Heart of Gold and mapping out the ship - do not be dismayed when you give a direction and the prompt says "you really

don't want to do that!" Do it anyway - you may have to convince the computer of it, but do it anyway! Points await!

PLANETFALL

Trot and double trot! Complete this Adventure the correct way and Ensign Blather gets his just rewards for all those demerits he gave you! The combination to the locked door with the dial is in the lab coat — use it if you need it. To open the padlock is another idea. Finding the key is a must!

Your emergency provisions just might run out, depending on your appetite. There are other provisions found in the right place. A tin can of spam is available, but for the more tasteful stuff, remember the canteen is octagonally shaped, and so is the dispenser in the kitchen. Give you any ideas?

SORCERER

The second series to Enchanter, what an Adventure! There's a great Fun Park in this game - with roller coaster rides and a flume log boat ride. The descriptions in this Adventure made you feel you were really alive and going through it! Enjoy the games and rides, and especially enjoy the Arcade! You just might win a prize!

After scoring 200 points out of 400, I found I had to restart (what's new) to do something more at the beginning. Remember, thoroughly check out the store room. There's more than one thing there in store (sic) for you!

As usual, questions are very much welcome. Please address your questions to:

Scott or Jo Ann Copeland P. O. Box 5927 RAF Lakenheath Brandon Suffolk IP27 9PN

Also, please write and let me know if you want the format changed or not. For example, should I give complete clues on one Adventure per article, or continue as is? And do more people want the Scott Adams' Series, or do you want more on Infocom Adventures? Your comments and thoughts are appreciated — and will probably help us all. So write in to the address above and let us know!

Until next issue then, Happy Adventuring and keep those cards and letters coming in!

PACKET SWITCHING

By NEVILLE BOSWORTH

The Packet Switchstream (PSS) network is run by British Telecom and is designed to provide fast error free data exchange between computers at low cost.

Although originally designed for exchange of data around the country or world by big business Users, it can also be used by the home computer enthusiast. Although the system is based on the very high quality lines of the Public Data Network which are not so prone as the public telephone circuits to data corruption, the home User can still access the network on his home phone by connecting in through the local exchange.

Thus a member of this network can access all parts of the country for the same call charge, and can access computers in around forty other countries in the world at very little cost.

The basic principle of PSS is that of Time Division Multiplexing where the multiplexer interleaves packets of information from each of the low speed Users onto a single high speed data channel.

If one User is sending more information than another User, and at a higher rate, then the system will devote more time to him than the other.

Each of the packets of information has a header attached so that the system will always be able to keep track of where the data is going and so it can never get lost in the system. It is here that the cost saving is produced in that if you were to call for example the Source in the USA using a normal phone line, and you were connected for one hour then you would spend most of that hour reading the contents of the screen and would pay in excess of thirty pounds for the privilege.

However, if you used the PSS network you would be sharing a line with many other Users and would only pay for the time during which data transmission was taking place: i.e., you would not pay for the time taken to peruse the information on your screen.

To access the network you do not need any special equipment or software as the Packets are assembled and addressed by the system itself. You can also access the system using all of the commonly used transmission speeds: i.e., 300, 1200, 1200/75.

The following is an example of how to access the Source in the USA:

1-Dial your local PSS exchange

2-Obtain access to a User port

(continued overleaf)

3-Type in your national User identity.

4-At the ADD prompt you would type a 9 for international service followed by 3110 which is the code of the network on which the Source is found, followed by the code for the Source which is 30100162.

All together this would read as A9311030100162.

5-You will now be told that connection has been made and you can then interrogate the database as normal.

Costs of the PSS network are approximately as follows:

A one-off connection charge of 25 pounds.

Charge to connect via local exchange (PAD) = 6.25 per quarter.

Charge per kilosegment of transmitted data (64K) = 15p during cheap rate period.

Duration charge/hour (cheap rate) = 90p

Cheap rate periods are as for the public telephone network. Phone charges are shown on your bill as normal while PSS charges are shown on a separate bill.

If you would like to read the PSS guide using your modem then use your TE2 module set to 300bps, PA=N, DUPLEX = FULL then phone 01-300 2333.

On connection you should press ENTER twice, and when asked for service type PAD, then press ENTER twice. At the prompt PAD type CALL 40; for USER ID and NAME/EST type HELP and then type GUIDE. (There may be a short delay after CONNECTED appears).

If you then decide that PSS is for you, phone 01-920 0661 or write to:

British Telecommunications PLC Customer Service Group, GO7 Lutyens House, 1-6 Finsbury Circus, LONDON EC2M 7LY

They will be pleased to send you an enormous information pack to enable you to decide.

In the next article, Neville will present some basic facts about modems for those of you who may have some questions which need answering.

CONTACTS

ALAN NEWMAN

64 DAKLEY ROAD, CHINNOR, DXON DX9 4HX

0844 51202

MARK PLAYLE

104 TUDOR WAY, SELE FARM, HERTFORD, HERTS SG14 2DS

0992 57507

BULLETIN BOARD

WANTED / 4 SALE / WANTED / 4 SALE / WANTED / 4 SALE / WANTED / 4 SALE /

VIC VERE has a console, Extended BASIC, cassette leads, single ATARI joystick with adaptor, numerous cassette-based utilities and games, as well as issues of User group newsletters (TI*MES, I understand), for sale at \$70 or nearest offer. Vic tells me that he has kept the console and Extended BASIC in very good condition, and you can contact him after 5 pm most evenings on 0926 39653

RICHARD SIERAKOWSKI has some more disks for sale. The price is £12 for ten disks, plus £1 post and packing. If you want a plastic library case add £1.50. I believe that these are still the XIDEX QUAD DENSITY disks which Richard offered back in July. Contact him on 0672-54975.

Richard is also still the agent for TATUNG monitors — if you want a good display at a good price, then get in touch with him.

I have recently received DM1000 V3.3, so if you have an earlier issue and you want to play the upgrade game, contact me on $0865\ 510822$.

I still have the PIO upgrade chips (£10), and the upgrade project to enable standalone disk controllers to access double-sided drives is just in its final stages – watch out for it at the Workshop!



T.I. 99/4A DISK PERIPHERAL OVERVIEW (Continued)

By COLIN HINSON

So far in the previous parts of this article, we have dealt with the floppy disk layout and the methods of accessing the disk itself.

When accessing the disk from a high level (i.e. using file access as opposed to sector access), current information about the disk and file(s) etc are held in VDP RAM so as to avoid having to re-access the relevant sectors on the disk each time a sector is read from a file. In addition to this data space is also required to buffer the data being read from the disk. The memory used for this is at the high end of VDP RAM and is permanently allocated by the power up routine within the DSR ROM of the disk controller, and as a consequence cannot be used by application programs, although its size can be changed by a (GPL) utility routine.

The allocated VDP memory is roughly sub-divided into three categories:

- 1. Drive Control information
- 2. File allocation information
- 3. Data buffering.

Each of these is discussed below.

DRIVE CONTROL INFORMATION

In order to control the Disk hardware, the software needs to know the current status of each drive before it can access it. All this information is readily available (some through checking the actual current status of the drive directly).

The power up routines take care of the FD1771 register intialisation.

FILE ALLOCATION INFORMATION

File allocation information is held in the File Control Blocks (FCBs), each open file having an FCB associated with it.

The information contained in the FCB is identical to the File Descriptor Record (FDR) data held on the disk for each file, with the addition of 6 bytes of dynamic information about each file. These six bytes are stored in front of the FDR information (i.e. the FDR starts at FDB byte 6). As the length of the FDR is 256 bytes and there is a data buffer for each file of 256 bytes, the total length of an FDB is therefore 256 + 256 + 6 = 518 bytes.

The format of the 6 byte extension of the FDR is:

```
Bytes -6 and -5 = Current logical Record offset on Level 2
Bytes -4 and -3 = Physical Record location of the FDR
Byte -2 = Logical Record offset
Byte -1 = Drive ID
```

1:

The meanings of these entries are as follows:

Drive ID - Contains the Drive number on which the associated file resides. If the highest bit of this byte is set, then the current data block has been modified and it will have to be written back to the drive before the file is closed or a new data block is accessed.

Logical Record offset - Contains the offset of the next logical record in the current physical record. If during a READ operation this points to a byte entry of >FF then this indicates an end of record for the current physical record.

Note that this entry is used only for variable length records. For fixed length records, the actual AU and the position within that AU is computed before each I/O operation, and therefore the logical record offset byte is irrelevant.

During WRITE operations, this offset points to the first free byte in the physical record. If the next logical record would leave less than one byte in the current record, a byte count of >FF will be written, and the logical record will be located in the next physical record. The first logical record in a physical record can never cause the physical record to overflow as the maximum logical record length is 254 and the physical record length is 256.

Physical location of the FDR - Points to the physical sector on the disk where the FDR resides for the associated file, and is used when it is necessary to re-write the FDR to the disk. It is maintained on read only accesses even though it is not required.

Current Logical Record Offset on Level 2 - Contains the physical record offset of the most recently processed physical record and is independent of READ or WRITE operations.

Always contains the logical offset for Level 2 of the datablock which is currently in memory.

It should be noted that this system causes fixed length sequential files to be accessed as relative access files on Level 2.

In order to buffer the data to and from the disk, a 256 byte buffer is maintained for each OPEN file. The buffer is located immediately above the FCB memory area.

One of the VDP RAM buffers is permanently assigned for processing VIBs (Volume Information Blocks - see previous parts of this series). If more than one drive is used in WRITE mode, then the bit maps are moved in and out of this area as required. This buffer is accessed for each access to the disk VIB.

Every Level 3 WRITE operation to a file will eventually be passed on to Level 2 as a physical sector WRITE. To keep the number of disk accesses to a minimum, a flag (the MS bit of the Drive ID) to indicate that the current data buffer has been modified. The data buffer is only written to the disk if the next physical record access involves another physical record. If the file is closed then the last data buffer is written onto the disk if required (this is the reason why the ends of files go missing if you forget to close them!).

VDP MEMORY LAYOUT

The memory layout is outlined in the diagram (such as it is!) below. This block of memory is reserved by the power up routine in the Disk DSR ROM. The size of the area from then on depends upon the number of files which are allowed to be open at the same time, and is initially set to 3. This number can be varied between one and 16 by calling a subprogram (through CALL FILES from BASIC). Each extra file will of course take up 518 bytes.

As for each peripheral, the disk peripheral identifies its reserved area through its CRU address (unique for each peripheral). The area is 'validated' by an >AA byte, followed by the address of the previous top of memory. As the disk peripheral has the highest priority on power up, this entry will always point to the actual top of memory. The disk system does not use this however, and so will work equally well on other CRU locations.

The first entry after the CRU ID contains the number of files for which the area is reserved, and directly determines the length of the reserved area. After this entry come the areas reserved for the FCBs and the associated buffers for each file. In order to simplify(?) the buffer allocation, buffers are not allocated on demand, but as soon as a file is opened. The FCB and buffer are associated with the file for its entire 'open' life.

3FFF	Name Compare Buffer	-+ 	++
: 3FF5	1	1 /	Current track 3
3FF4	Volume	1 /	Current track 2
	~ Information Block	~ !	Current track 1
3EF5	 B106k		Current Disk No!
3EF4	•	·+ / /·	* ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~
3EEF	Additional Info.	!/ /	
3EEE	 Disk Status Info.		l Data Buffer N
3EEB	l	1//	
3EEA	•	1 /	File
3DEB	VDP Stack space	1 /	Control Block N
3DEA	 	† ! +	·+
•	l Data and ~ File buffer	\ \ \ \ \ \ \	
3DEB-N*518	Space		1
3DEA-N*518	Maximum No. of files(N)	+ ! !\ !	Data Buffer
3DE9-N*518	CRU ID	+ \	number 1
3DE8-N*518	Next Buffer ()3FFF)	+ \	File
3DE6-N*518	Validity code ()AA)	+ \ \	Control Block #1
3DE5-N*518		+ \1 +	

The VDP Stack area is used to simulate a stack based machine with the TMS 9900, giving the programmer the advantage of being able to use the multilevel stack oriented CALL/RETURN system, rather than the single level BL system used by the 9900 series processors. The stack can of course also be used to PUSH and POP registers and data to and from it.

The disk status information area is used to save the current track numbers of the (3) drives, and the most recently accessed drive number.

The additional information area is no longer used (by the 99/4A).

The VIB buffer is described above.

At the top of memory, an 11-byte buffer is reserved which is used for name comparison. Every high level entry point automatically saves the drive number and the 10 character file name in this entry. If less than 10 characters are available, the buffer is padded with spaces.

Next time: DSR subprograms

THE TIHOME SOFTWARE COLLECTION

Pricing for TSC disks has been reorganised, although the full Collection is still offered at £35 inclusive.

Check the TSC Catalogue (available free to ITUG subscribers on request) to find out what the TSC entries stand for. Make sure that you indicate clearly what your choice is, specifying the name of the disk/s you want.

GAMES:	TSC ENTRIES:	PRICING:
TSC_DSK_A TSC_DSK_B TSC_DSK_C TSC_DSK_D TSC_DSK_E TSC_DSK_F EDUCATIONAL:	GA0023 - GA0032 GA0033 - GA0043	The number of programs on a disk can vary between about 7 and 11, dependent upon the sizes of the programs in terms of sectors used. The exception is the DEMONSTRATION disk, which alone is offered at £2.95 inclusive of post and packing
TSC_DSK_G TSC_DSK_H CONSTRATION:	ED0001 - ED0009 ED0010 - ED0017	The rest are priced as follows: £ 3.95 for ANY 1 DISK £ 6.90 for ANY 2 DISKS (SAVE £ 1)
TSC_DSK_I	DE0001 - DE0006	£ 9.85 for ANY 3 DISKS (SAVE £ 2) £12.80 for ANY 4 DISKS (SAVE £ 3) £14.75 for ANY 5 DISKS (SAVE £ 5) £16.70 for ANY 6 DISKS (SAVE £ 7)
TSC_DSK_L	MU0001 - MU0009 MU0010 - MU0017 MU0018 - MU0026 MU0	£18.65 for ANY 7 DISKS (SAVE £ 9) £20.60 for ANY 8 DISKS (SAVE £11) £22.55 for ANY 9 DISKS (SAVE £13) £24.50 for ANY 10 DISKS (SAVE £15) £26.45 for ANY 11 DISKS (SAVE £17)
UTILITIES:	i	£28.40 for ANY 12 DISKS (SAVE £19) £30.35 for ANY 13 DISKS (SAVE £21)
TSC_DSK_M TSC_DSK_N TSC_DSK_D TSC_DSK_P	UT0023 - UT0033 ! UT0034 - UT0044 !	£32.30 for ANY 14 DISKS (SAVE £23) £34.25 for ANY 15 DISKS (SAVE £25) £35.00 ENTIRE (SAVE £27.20)

All programs are recorded on Single-sided disks.

You may elect to be supplied with software on Double-sided disks, when you should deduct £1 from the prices shown above.

If you prefer, you may send in your own disks, in which case deduct 50p for each disk from the prices shown above.

For example, 8 disks supplied by you would work out at £20.60 - (8 x 50p = £4) = £16.60 nett.

All the above prices are inclusive of post and packing.

The pricing for programs recorded on cassette remains $m{\mathcal{L}1}$ per program, with an overall charge of 65p for post and packing.

PLEASE MAKE ALL CHEQUES PAYABLE TO "PETER BROOKS"