

=====

T I - L I N E S

=====

Volume 2, Issue 5

October 1st., 1985

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

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

TI-LINES is produced and published by:

* Peter G. O. Brooks *
* 96 Banbury Road *
* OXFORD *
* OX2 6JT *
* Tel: Oxford 50822 *
* *

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

THIS OFFER APPLIES UNTIL MAY 1986 ONLY

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

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

I n d e x t o a r t i c l e sP a g e

EDITORIAL.....	3
Tabletop Static ? Cellular Data. Reading It Right. Belgian Pen Pal. Crabs ? Excuses, Excuses! When The Worms Really Are Bad For You. Welcome Mat. Old Rope For Money.	
CONTACTS.....	8
One More For Your Address Books	
LETTERS.....	9
SYD MICHEL Debugs MiniMemory	
ENHANCED BASIC.....	10
Last But One Article In The Present Series (I Think)	
DIS/VAR WARS.....	12
Some More Lunacy, Here And On Page 22	
AID FOR DEF USERS.....	13
Originally Written For HOME COMPUTING WEEKLY, And Revamped To Give Food For Thought To The Up And Comers	
MSP 99 USER GROUP.....	15
Fascinating Article For Forth Buffs From MIKE KABALA On How To Make Music With Forth	
RICHARD OWEN.....	19
You'll Flip When You Try Richard's Disk Modification...	
SORTING AND SEARCHING.....	20
All About Flags And The First Taste Of A Bubble	
BULLETIN BOARD LIST.....	23
Boards In The DENVER Area Courtesy Of ROCKY MOUNTAIN 99ERS	
OTTAWA TI-99/4A USERS GROUP.....	25
LLOYD GALENZOSKI With Some Helpful Information	
BULLETIN BOARD.....	26
Selling Up	
CLOSE FILE.....	27
Shifty Modules. Unhappy Dumping Grounds ? Work, Work, Work!	

INSIST ON FORMICA ?

In the course of scouring the popular computing rags for snippets to impart to you, dear reader, I chanced upon a letter from a BRIAN HAMER of FORMICA LTD who wrote to PRACTICAL COMPUTING with regard to problems caused by ELECTROSTATIC DISCHARGE (ESD). It seems that static electricity is thought to be a problem in America, and Mr Hamer says that his company has a solution to the problem BUT... it seems that ESD is not thought to be a problem in the UK - it seems that a TOMORROW'S WORLD program highlighted ESD and its effects on chips but gave no solutions, and nobody in the UK suffers with ESD and it has never been brought to public attention before.

I wonder how it is possible for TI not to have been asked about its problems with ESD ? I know of a number of occasions that ESD has caused sufficient concern in TI circles for articles on the subject to have been published in newsletters like the late TIDINGS. I shall write in due course (i.e. when I can find time!) to Mr Hamer, but doesn't it strike you as odd that all of a sudden companies are discovering something about which we were aware five, six years ago ?

~~~~~  
FROM CELL TO CELL  
-----

As part of the presence of OTIU at the coming BIRMINGHAM show, we were trying to put on a demonstration of a Bulletin Board access via CELLNET phone and modems, but alas BT and SECURICDR, who run the largest cellular radio network in the UK, were only interested in attending (and sponsoring us!) if we could provide guaranteed orders for equipment and if several thousand people were likely to attend (i.e. 20,000). We had hoped to avoid the difficulties involved in having BT put in a line especially for us by using a cellular radio, thus enabling us to set up anywhere in the hall. In fact, it has turned out that Digbeth Civic Centre is often used by journalists reporting on meetings held there (like those of the National Front!) so there are already a number of lines available - we have only to get BT (at eighteen quid a throw) to connect us up and we will be able to run our demonstrations after all. GORDON PITT, OTIUser and Organiser of the WEST MIDLANDS TI USER GROUP, will be handling the details, and I believe that he has hopes of accessing not just ALAN DAVEY's 4ABC but also MICRONET if he can arrange it.

Anyway, I'm wandering off the point. The point is that cellular radio has been experiencing problems in transmitting data using modems. If you are in motion at the time, it seems that your data cannot be transmitted reliably. However, Transam Microsystems Ltd are galloping to the rescue with a special modem which should overcome the four phenomena described by BEN KNOX in his August column in PRACTICAL COMPUTING: Rayleigh fading, Hand-off, Base station to cellular phone communication (occurs in the gaps between speech), and Co-channel interference.

I won't go over these in detail (unless anyone writes in and threatens to beat me with one of my own jellies) but suffice it to say that reception is rendered virtually impossible under those conditions. The new modem from Transam uses "error correction techniques based on a positively-acknowledged data block with CRC checking" - CRC is Cyclic Redundancy Checking, used on our TI disk system - and is smart enough to switch from 1200 baud to 300 baud when the conditions

require it. It also has a special "intelligent" mode which it uses when you are on the move; of course you have to have a similarly intelligent Transam modem at the other end when this happens...

At a price of £350 this makes it a bit pricey for us, but if you've already spent £2000 on a cellphone what the hell...

~~~~~  

THE END OF TYPING AS WE KNOW IT

If you've got £400 to hand and an RS232 interface, the OMNI-READER is probably a nifty bit of kit for you to invest in - should you want to automatically read moderate quantities of text into your micro, that is. It can be set to recognise Courier 10 and 12 pitch, Letter Gothic 12, and Prestige Elite 12, and the manufacturers, OBERON INTERNATIONAL LTD, claim a maximum speed of 150 words per minute. Funnily enough, I actually could make use of such an item: some of the material for TI-LINES would fit into the narrow range suitable for Omni-Reader and save me (or Jenny Keane) some finger exercises.

~~~~~  
-----  
COME FORTH AND COMMUNICATE  
-----

RALF LAUWERS, 18, Forth buff, and son of RENE LAUWERS, OTIUser and contact with one of the Dutch groups, has indicated that he would be happy to correspond with any OTIUser (or relative) on the subject of Forth, Life, The Universe, and probably Everything. If you are interested in extending a hand across the Channel, you can get further details from me.

~~~~~  

WHEN THE CRABS BITE...

A recent issue of SCIENTIFIC AMERICAN revealed yet another creature lurking within the depths of some computer terminals. You may remember my past reports on Worms (see elsewhere), Dwarves, and the like, comprising the protagonists in CORE WAR. Well, A. K. DEWDNEY's Computer Recreations has thrown up CRABS this time. It seems that he visited the AT and T BELL Labs where at times the distinction between work and play can become very blurred indeed. Having developed a "smart" terminal with windows, iconography, and mice, its designers felt the urge to produce a program which broke the rules of operation of the terminal. In the grey area surrounding the windows they created little blobs which moved around at random, through a program called MEASLES. An operator would be called, or lured, away from a terminal and on their return would find that it had developed measles. Operators soon found that they could confine the activities of these measles by slapping a window over them, so the designers modified the program and made the measles flash on and off to try and annoy the operators. The new program was called ANGRY MEASLES.

The operators were not annoyed however, so the designers went one step further, and violated one of the basic rules of the terminal's operation, which was that no window contents could interfere with each other. ANGRY MEASLES became HUNGRY MEASLES, and they began to eat into any window and munch the contents. (This was achieved by having the measles "pave over" the window with the grey background). Someone commented that the measles were now behaving more like crabs, and so the program was modified still further with a Crab icon (what we know as a User-definable graphic character!). The CRABS program was run on the terminal of any unsuspecting operator, which needless to say resulted in a lot of angry operators!

They fought back with SQUISHCRABS, a program which looked for the CRABS program in the terminal's memory, erased it, and then turned any crab on the screen into a flattened blob.

However, a modification to the terminal firmware rendered SQUISHCRABS inactive, and the only way to erase the crabs was to switch the terminal off. This had the effect of breaking contact with the host computer, requiring the operator to log on again, and losing all the work in the process. Not a very friendly thing to do to a colleague!

The pro-Crab faction have since moderated themselves a little, and now restrict themselves to relatively harmless pranks like running programs which leave a variety of bird and animal tracks across the screen, rather like those seen after a fall of snow.

Now, it occurs to me that the interrupt facility on the 99s together with our redefinable character set makes the CRABS program feasible for us too. It would have to be included with another game, with a routine to check for the coded entry (like Shift 8 3 8 on certain modules, or FCTN X ten times on the Disk Manager module), but I have no doubt that it could provide lots of fun for someone - if not for the player, then for the observer.

DON'T PANIC!

Looking at the September Editorial, the casual observer might be forgiven for not noticing the faint tone of alarm which underlay the later items. Publication had been repeatedly held up, first due to technical problems, then due to technical problems, and later due to technical problems. At one point I lost over three weeks' worth of typing on one disk, the bizarre circumstances of which are the subject of this little item. During one phase of the operation to recover the files I managed to snatch only two hours sleep in three days - after all, I do still have a 9 to 5 job to keep me occupied. The 600+ miles that I mentioned in the last issue was trumped again and again, and ended with what is a marathon for me: over 1,500 miles travel in two weeks and I STILL didn't get everything done that I needed to!

Eventually, after a great deal of sweating and frowning of brows, I did manage to recover those files for which I had no backup. Before anyone leaps down my throat, I do normally make a backup copy of most files; in this particular case I was hurrying to try and get September's TI-LINES out while the month was still called September (dismal failure Brooks!), so short cuts had to be taken. In addition, I was reduced to a single disk system, which, as those of you with such systems will attest (go on, then, attest!), effectively discourages the copying of disks with large numbers of files, especially a double sided disk. At 3 in the morning, the last thing you want to do is start swapping disks in and out of a drive!

Anyway, that aside, my disk problems began when I tried to add a file of about 15 sectors of DIS/VAR 80 text to a collection of about 37 files of sizes varying from 2 to 76 sectors. TI-Writer threw up an I/O error indicating that it had been unable to SaveFile, so I exited TI-Writer (having no need to save the file elsewhere as it was itself a copy) and ran Disk Manager 2 through its non-destructive test. Three sectors were highlighted as being unreadable, so I decided to transfer the contents to another, hopefully more reliable, disk.

That's where my problems really began. All of a sudden, I couldn't make any backups of anything at all. I tried cataloguing. Disk Manager 2 fell over the first file. I ran the non-destructive test again. A list as long as my arm emerged. Don't Panic, I thought. Get Disk Manager 1 and see what it has to say.

Those of you who surrendered their early Disk Managers for a Disk Manager 2 (DM2) might like to know that DM1 can do things which DM2 refuses point blank to do. The most useful, from my point of view, is its inability to fall over. It trips, staggers, but never falls. If you give DM2 a directory containing one duff sector, it will catalogue up to that sector but not beyond. DM1 on the other hand will issue an error message and stop, but will continue with PROC'D.

DM1 revealed that only one of the directory sectors was at fault, so I inserted Navarone's Disk Fixer module to see what else I could learn.

Now, it might appear that these events took place contiguously - that is, one immediately followed another. Not so. In between, the phone rang continuously, (well, almost), I had to eat occasionally, I still had post to sort out, and I clocked up over 750 miles in that week alone - quite a feat for someone who normally only travels 100 miles in the same period.

Eventually I got Disk Fixer to dump out sectors 0 to 33 on my printer. If you have followed earlier descriptions of these sectors in TI-LINES, you will know that sector 0 contains the disk "header", sector 1 is the pointer list for files in alphanumeric order, and 2 to 33 contain the headers for each file together with pointers to their whereabouts.

I'd not really thought about it until then, but the 99s can have up to 127 named files in a single directory, and I'd been told long ago that the TI Disk File Management System simply stuck as many file headers as it could on sectors 2 to 33. Not so, as the result of a chance conversation with RENE LAUWERS proved (see elsewhere for a chance to correspond with his 18 year old son RALF).

Once sector 33 has been used to store a file header, any subsequent directory requirements are taken from the first free sector found beyond 33. I had 37 files, the early one quite large, so the later files had their headers stored from about sector 400 onwards. A few of the increasing numbers of duff sectors occurred around sector 165, the res from 459 to 490, but a couple of the suddenly-inaccessible files existed in their entirety, complete with headers, between 400 and 459. Disk Fixer showed them to be totally intact, sector 1 had the necessary pointers showing the correct locations, yet I could not load these files into TI-Writer, could not back them up using either Disk Manager, and could not even OPEN a file in console or Extended BASIC.

At this point I began to wonder if sleep hadn't taken hold and I had been dreaming all of this. Files which are totally inaccessible to everything bar Disk Fixer, and which are completely intact? It didn't make sense.

I decided to try and copy what files I could by loading them from the duff disk into TI-Writer and saving them out to a backup disk. By this time the TI-Writer Enhancement files, which I keep with every TI-LINES disk, had succumbed to the Dreaded Lurgy. Just sitting around, the disk was losing sector after sector - no exposure to direct sunlight or to any extreme temperatures, no electro-magnetic fields - so I had to work fairly quickly.

Eventually I managed to narrow the losses down to two files, one of which I discovered was already in existence elsewhere, albeit in its unedited form.

That left just one file lost. Phew! (as they say). I still hadn't been able to understand why this last file wasn't loadable (the intact file discussed above), and then it dawned on me. Supposing the access to files which didn't have headers in sectors 2 to 33 was by means of a brute force search? That is, instead of using sector 1's pointers to guide the search, suppose that DM1, DM2, TI-Writer, and TIDOS generally (TI's Disk Operating System) simply looked at every sector after 33 until it found a sector headed with the required file name and then perhaps ran some kind of validation process on it?

This would mean that any access could never get past sector 165 - the first duff sector - and would thus never encounter the intact files at sector 400 onwards.

The missing final file - which could be catalogued by DM1 but not copied by it / was 16 sectors long including the header, and quite by chance there was another 16 sector file with the same characteristics lying between the missing one and sector 459 whose header lay within sectors 2 to 33 (DAVID BROWN's work, as it happens; David, you're a life-saver!).

All I did then was, using Disk Fixer, to successively read in the "inaccessible" file, sector by sector, and write it out over David's file sectors (already copied safely). David's file then contained all of the missing file, albeit now with a different file name, and because the header was in 2 to 33 there was no need for TIDOS to go searching through the disk, falling over sector 165 in the process.

I powered up TI-Writer using the backup disk, crossed my fingers, loaded the faulty disk into the drive (difficult thing to do with crossed fingers) and bingo! loaded the missing file without a hitch.

A quick prayer to the gods of the Power Supply to Not Cause A Power Cut In Their Infinite Wisdom I Shall Sacrifice A Virgin Just As Soon As One Comes My Way, and I saved the file under its original name onto my bright shiny new highly trustworthy kiss kiss pat pat backup disk. By 3.30 am that morning I was happily plonking my guitar and making the woodlice wonder if the end of the world was nigh.

That's one of the nice things about working with the 99s - the feeling of utter elation you get when you manage to overcome yet another obstacle placed in your path by its designers. Whoopee!

~~~~~

A NASTY DOSE OF THE WORMS

I have been told of a case in which the dreaded Worm, mentioned in an earlier TI-LINES, and designed to help make mainframe systems more efficient, has been used to destroy someone's disk files. It seems that an innocent individual downloaded some material from a Bulletin Board into his micro - possibly an IBM PC or similar animal. Over a period of time he began to be aware that he was losing large quantities of data from his disks but did not discover the truth until it was too late.

Somewhere in the material which he had received from the Bulletin Board was a Worm, and it had invaded his Disk Management System (presumably held on a disk as is done with the latest MYARC Manager, rather than in firmware). The result was that each time he used the Manager to format or process a disk in some way, the Worm copied itself onto that disk and then began chomping whenever the opportunity arose.

This is an example of not just a nasty prank as is the case with the CRABS item discussed earlier, but of a criminal act. I have read stories about threats made in Italy by individuals who suggest that they would like to release worms into the Italian computer network which links Universities (something like the JANET system in the UK). If they ever did so - and there is the possibility that they may have done so already, using a software "time bomb" to control its release, - then a massive amount of valuable and important knowledge and information would be lost. This would be an act of terrorism of the most sophisticated kind, but there is a precedent. Once the power of the Worm was recognised in America, reports appeared in the popular computing press to the

effect that the military had taken an active interest with a view to leaking material to the Russians in the form of contaminated software. The intention is that when the Iron Curtain countries eventually develop their own computer systems to a more sophisticated level, (currently about ten years behind the West), any attempt to use the computers for aggressive purposes could be blocked by the military without the need for physical deployment of any kind.

The mind boggles.

~~~~~  
HELLO, HELLO  
-----

It has been a while since I had occasion to say Hello to anyone new, so it gives me pleasure to welcome ROY GARDINER, DEREK O'BRIEN, RICHARD CREDITON-HUGHES, D. B. ALLEN, DAVID BROWN (another one!), B. J. BADGER, and NEIL LAWSON. Through the auspices of GORDON PITT and the WEST MIDLANDS TI USER GROUP I can also say welcome to R. JONES and D. MAIDMENT. I have realised recently that because I have not sent out any questionnaires with volume 2, we have no real opportunity for Contacts to make themselves known. Accordingly, I will be sending out a flier with one issue (not sure which yet) on which you may make yourselves known to all and sundry. There is a fairly detailed equipment/software check list, which, as usual, you do NOT have to fill in if you don't want to.

~~~~~  
RIP-OFF

I recall STEPHEN SHAW telling me of a firm in the States selling the TI-Writer Enhancements, which are Public Domain, for some \$35. I understand that he wrote to the people concerned, but that they do not appear to have altered their pricing structure.

In a similar vein I came across an Editorial in a recent newsletter which waxed lyrical about some of the software offerings made available to members. For a mere \$14.95 you could purchase a short piece of machine code which, wait for it, unprotected Extended BASIC programs. I wonder what happened to CALL LOAD() ?

~~~~~  
C O N T A C T S  
-----

IVAN M. NIBUR

48 DITTON ROAD, SURBITON, SURREY KT6 6RB Until 31/12/85  
Tel: 01 390 2019 but not after 10 pm please.

-----  
L E T T E R S  
-----

SYD MICHEL wrote to me to advise me of a bug in the MINIMEMORY software, and provided the answer to a question which has been niggling away at the back of my mind ever since I bought the module (more than two years ago, so the battery must need changing - no, not my mind's, the MiniMemory's!).

I had observed that there didn't seem to be any noticeable difference between NEW and OLD when entering or re-entering the MiniMemory (MMM), but just put this down to my lack of experience/knowledge. Syd has received many enquiries over the last couple of years regarding this difference, and here is his response:

"On page 5 of the Line-by-line Assembler handbook, the following information is given:

- B. If you want to create a new assembly language program, type NEW and press ENTER. The program clears the symbol table and waits for your first program line entry.
- C. If you want to continue writing a program that you began previously, type OLD and press ENTER. The old symbol table is retained, and the screen displays the next memory location, ready for you to continue your program.

That is what the difference SHOULD be, but good ol' TI cocked it up again, for the Line-by-line Assembler treats OLD and NEW as...NEW!!

As a result, when you try to carry on writing a program the assembler doesn't recognise any LABEL you REFER to, because it has overwritten the symbol table. You have to try and re-EQUate all of your labels - if you can remember all of their addresses! Nine times out of ten you will have to start all over again and re-enter your entire program from scratch.

By popular request I have had a look at the code (Aaaagh! What a mess of nested subroutines and indexed memory addresses!) and eventually I came up with the bugs:

Use EASY BUG (once you have loaded the assembler) to alter the following memory locations' contents:

| ADDRESS | OLD CONTENTS | CHANGE TO |
|---------|--------------|-----------|
| >71A9   | >3E          | >8E       |
| >71AF   | >3E          | >8E       |
| >7229   | >3E          | >8E       |

Having made these changes, DON'T FORGET: SAVE THE CHANGED PROGRAM. The Line-by-line assembler should now behave itself (fingers crossed)."

Thanks, Syd, your efforts are much appreciated.

~~~~~

Enhanced Basic in word processing

The term Enhanced Basic is used to indicate the extra statements that are available in addition to TI Basic when the Personal Record Keeping (PRK) or Statistic module is present in the GROM-port.

The exact mechanism of these extra commands with two separate memories (MEM1 and MEM2) is explained in a previous article.

The word processing features of your TI 99/4(A) are greatly improved by Enhanced Basic and it permits you to load a PRK file and use the data in this file in your TI-Basic programme.

An example may show more than many words, therefore allow me to introduce the chess club Excelsior.

A sample members PRK file is illustrated in appendix 1. This needs to be created in the module mode, saved and then loaded when the main programme (appendix 2) is RUN.

As example letters we have taken:

- I an invitation to a general meeting (all members)
- II an invitation to play (selection on team no)
- III an request to pay the (balance of the) membership fee (selection on fee)

Key this example in, study the programme closely, RUN it and your BASIC knowledge will be greatly ENHANCED.

Paul Karis & Franc Grootjen
Blauwgras 2
3902 AA VEENENDAAL

```

100 REM EXCELSIOR PRO
110 REM DEMONSTRATION OF ENHANCED BASIC
  IN WORD PROCESSING
120 REM COPYRIGHT PAUL KARIS + FRANC BR
  OUTJEN, BLAUWGRAS 2, VEENENDAAL, THE NET
  HERLANDS.
130 CALL CLEAR
140 CALL D(1,5,2B,"PRESS",5,1,2B,">1< GE
  NERAL MEETING",7,1,2B,">2< INVITATION",9
  ,1,2B,">3< FINANCIAL LETTER")
150 CALL D(11,1,2B,">4< .....",24,
  1,2B,"MEM FILL MEM 2 (PRK-DATA)")
160 CALL KEY(O,K,S)
170 IF K=77 THEN 950
180 IF (K<49)+(K>51) THEN 160
190 INPUT "DATE \ ":DATE$
200 REM GET FILENAME
210 CALL H(1,1,1,N$)
220 IF N$="" THEN "EXCELSIOR" THEN 960
230 REM GET # OF MEMBERS
240 CALL H(1,6,1,N)
250 ON K-48 GOTO 260,380,620
260 REM LETTER GENERAL MEETING
270 GOSUB 910
280 FOR I=1 TO N
290 REM GOSUB LETTER HEAD PLUS ADDRESS
300 GOSUB 790
310 PRINT #1:"WE SHALL CONVENE FOR AN AN
  NUAL GENERAL":MEETING ON SATURDAY 25TH
  OF AUGUST 1984:;
320 PRINT #1:"WE SHALL DISCUSS":TAB(5);"
  1) INCREASE IN MEMBERSHIP FEE":TAB(5);"2
  ) CHOOSING OUR TEAM LEADERS"
330 PRINT #1:TAB(5);"3) NO SMOKING SECTI
  ON":;
340 PRINT #1:"WE ARE COUNTING ON YOUR PR
  ESENCE":;"YOURS TRULY":;"PAUL KARIS":;"(SE
  CRETARY)":;
350 CLOSE #1

```

```

360 NEXT I
370 GOTO 130
380 REM INVITATION
390 CALL CLEAR
400 CALL D(1,1,2B,"PLAYING DATE ?")
410 CALL A(1,16,2B,C,PLAYDATE$)
420 CALL D(3,1,2B,"TEAM ?")
430 CALL A(3,16,2B,C,TEAM,1,4)
440 CALL D(5,1,2B,"AT ?")
450 CALL A(5,16,2B,C,PLACE$)
460 CALL D(7,1,2B,"OPPONENT ?")
470 CALL A(7,16,2B,C,OPPONENT$)
480 GOSUB 910
490 REM PICKUP TEAMLEADER
500 ON TEAM GOSUB 1010,1040,1070,1100
510 REM ACTUAL LETTER
520 FOR I=1 TO N
530 CALL G(1,1,5,MIS,TEST)
540 IF TEAM<TEST THEN 600
550 GOSUB 790
560 PRINT #1:"YOU ARE CORDIALLY INVITED
  TO PLAY ON ":PLAYDATE$;" AT ":PLACE$;" A
  GAINST":OPPONENT$
570 PRINT #1:;"PLEASE TELEPHONE YOUR TE
  AM LEADER":LEADER$;" TELEPHONE NUMBER:
  ";TEL;" IF YOU CANNOT COME."
580 PRINT #1:;"GREETINGS":;"FRANC GROO
  TJEN":;"EXTERNAL COORDINATOR"
590 CLOSE #1
600 NEXT I
610 GOTO 130
620 REM DUNNING LETTER
630 GOSUB 910
640 FOR I=1 TO N
650 REM GET PAID UP AMOUNT
660 CALL G(1,1,6,MIS,PAID)
670 IF PAID=100 THEN 770

```

```

600 REM ACTUAL LETTER
690 GOSUB 790
700 PRINT #1:"IT HAS COME TO MY ATTENTION
N THAT YOU": "HAVE NOT YET FULFILLED YOUR
FINANCIAL"
710 PRINT #1:"OBLIGATIONS TO OUR CLUB":;
"AS YOU KNOW THE ANNUAL MEMBERSHIP FEE I
S"
720 PRINT #1:"$100 AND SINCE MY BOOKS SH
OW THAT YOU": "HAVE TO DATE PAID $":PAID;
" THE BALANCE"
730 PRINT #1:"OF $":100-PAID;" IS STILL
OUTSTANDING AND": "OVERDUE":;
740 PRINT #1:"PLEASE PAY $":110-PAID;" (
BALANCE + $10 EXTRA": "CHARGES) WITHIN 14
DAYS INTO OUR ACCOUNT":;
750 PRINT #1:"YOURS TRULY":; "G.RABBER":
"(CONTROLLER)"
760 CLOSE #1
770 NEXT I
780 GOTO 130
790 REM SUBROUTINE ENTRY FOR LETTERHEAD
800 REM GET MEMBER, STREET/NO, ZIP/TOWN, S
EX
810 CALL G(1,1,1,MIS,A$)
820 CALL G(1,1,2,MIS,S$)
830 CALL G(1,1,3,MIS,T$)
840 CALL G(1,1,4,MIS,F$)
850 OPEN #1:"RS232.BA=9600.DA=8"
860 AH$="MR"

```

```

870 IF F$="M" THEN @90
880 AH$=AH$+"S"
890 PRINT #1:TAB(14); "CHESSCLUB 'EXCELSIOR
OR'":TAB(14); "BROOKLYN":TAB(14); "DATE :
":DATE$: "A$":S$:T$: "DEAR ":AH$: " ":F$: "
"
900 RETURN
910 REM SUBR SCREEN MESSAGE
920 CALL CLEAR
930 PRINT "PRINTING"
940 RETURN
950 REM LOAD PRK-FILE
960 CALL L("CS1",Y)
970 IF Y=0 THEN 990
980 GOTO 130
990 PRINT "* ERROR"
1000 GOTO 960
1010 LEADER$=" MR.B. FISCHER"
1020 TEL=99444
1030 RETURN
1040 LEADER$="MR.P.SECOND"
1050 TEL=12345
1060 RETURN
1070 LEADER$="MR.A. THIRD"
1080 TEL=98765
1090 RETURN
1100 LEADER$="MR.B. BOTTOM"
1110 TEL=34521
1120 RETURN

```

Remarks:

1. Prepare your memory partition in this example with CALL P(10792) as the TI BASIC program is about 3} K in size. Don't forget to follow up your CALL P(p) with NEW.
2. Your previously saved main program is loaded in the normal way with OLD CS1. When it is RUN, choosing 'M' from the menu allows you to enter the previously recorded PRK-data. (In this case the sample file of club members)
3. Note that the variables N and N\$ are automatically assigned values from the permanent memory, also after a BREAK, reRUN, SAVE or alteration in the program lines without the PRK-data having to be reloaded.
4. Study the CALL H and CALL G statements of this example and trace their functions.
5. This example program happens to take data from CS1 and output is directed towards an RS232-printer with the width set at 40 characters. Obviously one must alter the program to suit other peripherals if so required.

Voorbeeld 1 CHESSCLUB 'EXCELSIOR'
BROOKLYN

DATE :10 DECEMBER 1982

B.L. KING
QUEENSSTR. 14
1313 NEW YORK

DEAR MR B.L. KING,
WE SHALL CONVENE FOR AN ANNUAL GENERAL
MEETING ON SATURDAY 25TH OF AUGUST 1984

WE SHALL DISCUSS

- 1) INCREASE IN MEMBERSHIP FEE
- 2) CHOOSING OUR TEAM LEADERS
- 3) NO SMOKING SECTION

WE ARE COUNTING ON YOUR PRESENCE

YOURS TRULY
PAUL KARIS
(SECRETARY)

Voorbeeld 2 DATE :11 DECEMBER 1982

P. LAYER
BLACKALLEY 1212
2348 WASHINGTON

DEAR MR P. LAYER,
IT HAS COME TO MY ATTENTION THAT YOU
HAVE NOT YET FULFILLED YOUR FINANCIAL
OBLIGATIONS TO OUR CLUB

AS YOU KNOW THE ANNUAL MEMBERSHIP FEE IS
\$100 AND SINCE MY BOOKS SHOW THAT YOU
HAVE TO DATE PAID \$ 75 THE BALANCE
OF \$ 25 IS STILL OUTSTANDING AND
OVERDUE

PLEASE PAY \$ 35 (BALANCE + \$10 EXTRA
CHARGES) WITHIN 14 DAYS INTO OUR ACCOUNT

YOURS TRULY

G.RABBER
(CONTROLLER)

Voorbeeld 3 CHESSCLUB 'EXCELSIOR'
BROOKLYN

DATE :12 DECEMBER 1982

B.L. KING
QUEENSSTR. 14
1313 NEW YORK

DEAR MR B.L. KING,
YOU ARE CORDIALLY INVITED TO PLAY ON
26 DEC. 1982 AT THE SQUARES AGAINST
THE CHESSMEN

PLEASE TELEPHONE YOUR TEAM LEADER
MR.B. BOTTON, TELEPHONE NUMBER: 34521
IF YOU CANNOT COME.

GREETINGS

FRANK GROOTJEN
EXTERNAL COORDINATOR

FILE STRUCTURE

FILE: EXCELSIOR

DATE: 10/20/82

TITLE: 6 SAMPLEMEMBERS

ITEM	TYPE	WIDTH	DEC	0	1	2	3	4	5	6
1 MEMBER	CHAR	15	0	1 R.L. KING	QUEENSSTR. 14	1313 NEW YORK	M	4	100.00	
2 STREET/NO	CHAR	15	0	2 P. LAYER	BLACKALLEY 1212	2348 WASHINGTON	M	2	75.00	
3 ZIP/TOWN	CHAR	15	0	3 L.L. ANGEL	HEAVENSTR. 1	1290 L.A.	M	1	100.00	
4 SEX (M/F)	CHAR	1	0	4 L.O. SER	DUMP 12	1212 NEW YORK	F	3	13.24	
5 TEAM	INT	1	0	5 O.P.O. NENT	PANWPLACE 12	5678 LUBROCK	M	1	100.00	
6 PAID	DEC	8	2	6 BLACKJACK	CLOUD 9	234 ACROPOLIS	F	4	.01	

FICTION

DIS / VAR WARS

A tale of two CTs, adapted from two original stories in such a way as to make them both totally unrecognisable.

PAGE 2

Syn spoke firmly to her, assuring her that the insect, a long-standing friend, could not possibly have acted in such an ill-mannered way towards someone so obviously aristocratic as she. The blob felt a little mollified, but could not quite bring herself to relinquish her complaint of indecent assault.

Syn affirmed the insect's reliability, saying that he had known Herbert for most of his life, and had never known him to act improperly towards anyone.

It is a strange thing, but invariably the pronunciation of an individual's name in the native tongue has a phonetic equivalent in other languages. Thus it was that although the furry insect's name, translated into English, was indeed Herbert, phonetically, in his own language, it had a strong similarity to Pooper-Scooper. Interestingly, Herbert came from a society which believed implicitly in re-incarnation, and Herbert's next incarnation - although he did not know it - was to be as a polythene bag used in conjunction with the terrestrial pooper-scooper in downtown New York. Additionally, the direct translation of "pooper-scooper" in Herbert's local dialect sounded a lot like the American phrase "Tricky Dicky", and was often used on Herbert's home planet to describe the dubious activities of secondhand speedster salesmen (as in "Would You Buy A Secondhand Speedster From This Pooper-Scooper?").

I mention this in passing simply to acquaint the reader with some of the coincidences which can, and frequently do, occur in the known Universe.

For example, the name of the orange blob of gelatine in her own language translates as Samantha; however, phonetically it sounds very similar to the phrase "Watch 'er, She's A Man-Eater". The phonetic interpretation of the word "Samantha" in her own language is a little difficult to translate directly, but involves the acquisition of items from a chain of stores not unlike our own Mothercare chain.

But I digress. The reception was swiftly drawing to a close, the guests clambering over each other in their haste to reach their speedsters, or the nearest Warp or Hype terminals. A certain degree of panic was to be expected: intelligence sources had revealed that the Mint Imperial Guards (often called MIGs but never mistaken for aircraft) of the dreaded REMpire were closing in fast; having repaired their damaged bit maps using a special CB Hash Coding System. Many said that the initials "CB", a closely-guarded REMpire secret, stood for Corned Beef, and it was true that part of the secret lay in the way that the vegetables were tossed into the frying pan, but there the similarity ended. Or so confidential sources within the REMpire's Central Processing Unit claimed.

(continued on p.22)

SOME AID FOR DEF USERS

Peter Brooks

Originally written for HOME COMPUTING WEEKLY

Some owners have experienced a little difficulty with DEF, the User-definable function facility. The most common question appears to be: "When should we use it, and how?". This small article won't do justice to the function, but hopefully it will begin to lift the veil of mystery which seems to surround it.

In TI BASIC you are provided with "functions" such as ASC(), CHR\$(), EXP(), VAL(), LEN(), SEG\$(), POS(), and so on. These enable you to manipulate or examine numbers and strings of text. ASC() for example gives you the ASCII code of a string character, while CHR\$(), its "opposite number", gives you the string character for a specified ASCII code. Thus ASC("A") is 65, and CHR\$(65), is "A".

DEF allows you to create your own functions, sometimes quite complex ones, which can then be used within a program to make it either a little easier to write or to understand. DEF is usually quite a slow function, and in some cases it is best replaced by a subroutine, but it can on occasion save considerable space in a program line, and therefore mean less typing!

TI BASIC provides you with the function INT(), which will obtain a copy of the INTEGER part of a number. Generally, the integer part of a number is everything to the left of the decimal point, while the FRACTIONAL part is the decimal point and everything to the right of it. There are exceptions to this (e.g. scientific notation) but for this article they shouldn't matter. You are not provided with the FRAC() function, which would give you the fractional part of a number, but you can implement it with DEF.

If we have a number: 123.456 - stored in a variable called X, then INT(X) will give 123. The fraction, .456, can be obtained by subtracting the integer, 123, from X thus:

$X - \text{INT}(X)$

To save us continually writing $X - \text{INT}(X)$ or in fact any other complex expression: $((A + B) / 3 - C / D) - \text{INT}(((A + B) / 3 - C / D))$ for example - to get the fractional part of the result, we could instruct the computer using DEF to do it for us. If we write:

140 DEF FRAC(X) = X - INT(X)

then although we have used X in the DEF line, the 99s take this to mean "anything". You can put what you like in the brackets:

140 DEF FRAC(A) = A - INT(A) / 1000 + A / 17

and so on. You don't HAVE to use a name (like FRAC) which has anything to do with what the function is; in the example immediately above, FRAC(A) quite obviously does NOT provide a fraction function.

The result of the operation of your function can be used like the existing functions: it can be tested with IF...THEN, it can be assigned to a variable (Z = FRAC(P / Q)), it can be used with other functions (Z\$ = STR\$(FRAC(R + B) + .15)), and so on.

Our DEF is a little limited in that you can only have one parameter within the brackets of the definition: you can't have DEF BAR(X,Y) = (X - Y) / 2 for example, but you CAN make use of other parameters:

```
140 DEF A(B) = B / C + B / D
```

Here the computer will regard "B" as anything, but will search through the list of variables being used by the program until it locates C and D, and will use the current values which they represent. If either C or D are not used in the program (which means that you have made a boo-boo!) the computer will place the variable name in its list as usual, and use zero as the value - just in case you wondered what would happen.

Suppose you needed to use logarithms base 10 in a program (TI BASIC gives you logs base e), you could create your own log 10 function, and an anti-log 10 function as well:

```
DEF BLOG(X) = LOG(X) / LOG(10)
DEF EXB(X) = 10^X      (N.B. The symbol between the 10 and X is the
                       caret)
```

To obtain the log base 10 of a number (X) from logs to the base e, you simply divide the log e of X by the log e of 10; to anti-log base 10 you simply raise 10 to the Xth power. If all this talk of logs has your eyes glazing over, don't worry, you can use functions like these without really needing to know what logarithms are - as long as you get the equations right, that is.

To obtain logs to any base, you divide the log to the base e of X by the log to the base e of the required base, which is very badly explained, but this is what the DEF statement is:

```
DEF ALOG(X) = LOG(X) / LOG(B)
DEF EXA(X) = B^X      (N.B. don't forget the caret)
```

In this instance, you will need to assign a value for the base to variable B before using the DEFINED function. To use log to the base 25 just set B = 25, and then use ALOG() or EXA() just as you would LOG() or EXP(). I've chosen the names of the new functions arbitrarily - you might have better mnemonics in mind.

One of the nicer things about the TI system is that if you use DEF in a program and then stop the program, all your DEFINED functions are still active and can be used in "IMMEDIATE MODE". You can therefore RUN a short program which consists of just DEF statements, and then test their function "by hand" as it were.

Write a short program to create the functions discussed here, and then test them out. If you have any problems, send them to me and where the problem is a common one I will present it in a future article; otherwise I will respond personally.

~~~~~

-----  
SOUNDING OFF IN TI FORTH  
-----

by MIKE KABALA from the newsletter of the MSP 99 USER GROUP

I'm sure that by now you're probably tired of hearing about how Texas Instruments designed a really fantastic machine, but did such a poor job of marketing it that nobody knew about it until it was too late. That appears to be the case with the sound chip as well. There are a lot more possibilities for this little gem than you are given access to in BASIC. (Or EXTENDED BASIC, for that matter.) There's a whole chapter on sound in the Editor Assembler manual that describes these capabilities.

According to this manual, there are two different ways to create sound. The first involves creating a "sound table" in RAM containing information on frequency, attenuation, and duration for each sound you want to produce. The other method is to feed the information directly to the sound processor at address )8400 (hexadecimal).

I chose the second method for two reasons: By poking data directly into the sound chip, you have the ability to change one of the voices (there are three, plus one noise channel) while the other voices just go on doing what they were doing. Also, you have the ability to change the attenuation while a note is playing. This allows you to construct an envelope for each sound as I will demonstrate later in this article.

Wait a minute! This article was supposed to be about creating sounds in FORTH, wasn't it? So why am I spending so much time talking about assembly language? Well, it appears that the authors of TI-FORTH forgot about sound when they wrote the manual. This would lead most people to believe that TI FORTH does not support sound generation (as I found out when I attended the November MSP 99 Users' Group meeting where I was drafted to write this article and to do a presentation at the December Assembly Language meeting). This is definitely not the case. Remember TI FORTH gives a programmer access to ALL machine resources. That includes address )8400.

MAKING NOISE  
-----

All you need to do is store the right values at address )8400. For example, to start a frequency of 110 Hz with an attenuation of 16 Db on voice #1, just type:

```
8900 8400 ! 3F00 8400 ! 9800 8400 !
```

and to turn it off, type:

```
9F00 8400 !
```

Simple, isn't it?

Okay, okay! So it's not so simple. The only thing that makes it hard is the fact that you have to remember all of those meaningless numbers. It will become a lot simpler if you just define a word to do the remembering for you. The following word will take a frequency value and a voice number and automatically set the proper frequency on that note:

HEX

```
: CREATE-NOTE
  ( freq voice -- voice )
  SWAP OVER 2000 * 6000 + SWAP DUP
  10 / 100 * SWAP ROT SWAP OF AND
  100 * + 8400 ! 8400 ! ;
```

DECIMAL

Now all you have to do is type 1017 1 CREATE-NOTE and you will have set the frequency on voice #1 to 110 Hz. Still not simple enough? Try this. Just define another word called AO (meaning the note A in octave 0) like this:

```
: AO 1017 ;
```

Then you just have to type AO 1 CREATE-NOTE to do the same thing. (CAUTION: This word actually redefines a hexadecimal number. Be sure you're through with hexadecimal before you define this word.)

Attenuation can be treated in a similar manner. The following two words will allow you to turn a note on and off:

HEX

```
: NOTE-ON ( voice -- )
  2000 * 7000 + 800 + 8400 ! ;

: NOTE-OFF ( voice -- )
  2000 * 7000 + F00 + 8400 ! ;
```

DECIMAL

Now, typing 1 NOTE-ON will turn voice #1 on, and typing 1 NOTE-OFF will turn it off.

IT'S ALL IN THE TIMING

---

The only thing that remains is to define words to waste time. That can be done with loops. I defined a word called MILLISECONDS to keep track of most of this.

```
: MILLISECONDS ( n -- )
  10 / 155 * 0 DO LOOP ;
```

Place a value on the stack corresponding to the number of milliseconds you wish to delay, for example:

```
A0 1 CREATE-NOTE NOTE-ON
50 MILLISECONDS 1 NOTE-OFF
```

will play the note we defined earlier for a duration of approximately 50 milliseconds.

Notice that you don't necessarily have to turn the note off after the time delay. Let's define a few more notes for this next example.

```
: C1 855 ;
: E1 679 ;
: G1 571 ;
```

Now try the following. (You may wish to define it as a word and then execute it.)

```
C1 1 CREATE-NOTE NOTE-ON
50 MILLISECONDS
E1 2 CREATE-NOTE NOTE-ON
50 MILLISECONDS
G1 3 CREATE-NOTE NOTE-ON
50 MILLISECONDS
1 NOTE-OFF 2 NOTE-OFF 3 NOTE-OFF
```

Notice that the first two notes continue playing as the next note is switched on. This provides a great deal of versatility in creating sounds. (You just have to remember to turn all the notes off when you're done with them!)

#### ON THE ATTACK

-----

Now comes the fun part -- envelope generation! You can create much more interesting sounds if you play with the attenuation a little bit. The way we have it set up now, a note is either on or off with no variations in between. Natural sounds aren't quite so simplistic, however. The sound that a note makes at its beginning is usually very different from the sound it makes a few milliseconds later. That's what enables us to tell the difference between a flute and a piano.

While the TI Home Computer can't reproduce the sounds of these instruments exactly, we can come closer by changing the attenuation (volume) of a note very rapidly at the very beginning and the very end. To do this, I defined some variables to hold my attenuation values.

```
O VARIABLE ATT-LEN
O VARIABLE ATTACK 18 ALLOT
```

Then I defined a word to make it easier to store these attenuation values away.

```
: STORE-ATTACK ( n1 ... n1 n -- )
  DUP ATT-LEN ! DUP O DO SWAP
  100 * OVER 1 - 1 - 2 *
  ATTACK + ! LOOP DROP ;
```

To store an attack, place the attenuation values (numbers between 0 and 15) on the stack followed by the number of attenuation values in the list and then call STORE-ATTACK. For example,

```
12 8 4 6 7 8 6 STORE-ATTACK
```

will produce a piano-like attack. Similarly,

```
13 11 9 8 4 STORE-ATTACK
```

will produce a more organ-like sound.

Now we may re-define NOTE-ON to produce an attack on each note corresponding to the stored values. (These values may be changed at any time simply by calling STORE-ATTACK with the desired values on the stack.)

```
: NOTE-ON ( voice -- )
  2000 * 7000 + ATTACK ATT-LEN @
  0 DD OVER OVER SWAP 1 2 * + @ +
  8400 ! 5 MILLISECONDS LOOP
  DROP DROP ;
```

Now, if you type NOTE-ON, you will have the attack added to the start of the note.

#### IN CONCLUSION

-----

Well, I could go on, but the remainder of what I've done has been mostly in the way of embellishments to the stuff I've already mentioned and it wouldn't be much fun if I told you everything. You can probably think up better things on your own anyway. I will make one little suggestion, though. If you make the names of your words a lot shorter than what I've used here, you will be able to fit your sounds on much fewer screens. (You won't have to type in as much either).

I've found it easiest to define a song as a single word called PLAY. Then, after you've loaded your song, you just have to type PLAY to hear it. I also define an empty word called SONG at the beginning of the first screen of a song definition. That way, if I want to load a different song, I just type FORGET SONG before loading the new song and my old song is painlessly removed from memory.

I hope you have as much fun with this as I have. If you come up with any really good ideas, please consider writing an article or a letter to the editor and sharing it with the rest of us. In the meantime, have fun with FORTH sound!

~~~~~

S O R T I N G A N D S E A R C H I N G

P e t e r B r o o k s

O c t o b e r 1 9 8 5

THE FLAG

Flags in BASIC are usually (but not always) numeric variables. Sometimes, to hammer the point home, the variable's name is actually FLAG.

To "set" a flag, you might give it a value of 1, to "unset" or "clear" it, you might give it a value of 0.

FLAG = 1 SET the flag

FLAG = 0 UNSET or CLEAR the flag

I say "might", because YOU choose what to do, whatever suits your purpose, frame of mind, shoe size, or underarm deodorant.

You might (if you didn't have both oars in the water) choose this:

FLAG# = "ICKY POO" SET the flag

FLAG# = "WIZARD PRANG" UNSET or CLEAR the flag

Or even:

SAMANTHA = 256 SET the flag

SAMANTHA = 999 UNSET or CLEAR the flag

Testing a flag to see if it is SET or UNSET can be really, really difficult:

IF FLAG = 1 THEN aaa ELSE bbb

which translates as:

IF the FLAG is SET, THEN jump to line aaa, ELSE it must be UNSET so jump to line bbb.

The ELSE isn't even necessary if it would mean passing on to the next line in the program anyway!

If you have trouble visualising (i.e. creating a little mental picture, and if you liked FLAG# then the emphasis must be on the word "mental"!) the SET and UNSET operations of the flag, think of it in football terms, where if the linesman's flag is UP, the ball went out of play at that point, and if DOWN, well, the captain of the protesting team probably trains his lads to leap up and down every time the ball veers away from the centre spot.

So now you know what a FLAG is, how you might SET and UNSET it, and how you can test it. FLAGS can also be "multi-state", which means that a single flag can indicate the occurrence (or not) of several events, by taking one of several unique values.

Back to Bubbling. Here is a fully-fledged Bubble Sort routine, complete with FLAG, which works on array A\$(). A\$() has ten entries (elements) numbered 0 to 9 (and is designed to sort the list of letters discussed in the previous article).

```
1000 FLAG = 0
1010 FOR POINTER = 9 TO 1 STEP -1
1020 IF A$(POINTER) > A$(POINTER - 1) THEN 1070
1030 G$ = A$(POINTER)
1040 A$(POINTER) = A$(POINTER - 1)
1050 A$(POINTER - 1) = G$
1060 FLAG = 1
1070 NEXT POINTER
1080 IF FLAG = 1 THEN 1000
```

This may have thrown you in at the deep end a little, but it is mainly so that you have something tangible to examine as we discuss it. It has the following components:

a flag: FLAG

a value of 0 means it is UNSET or C.....

a value of 1 means it is SET (which will show that a SWAP has occurred)

a FOR-NEXT loop

The "control variable" is called POINTER because it will "point" to each entry (element) in the list (array) called A\$().

It counts down from 9 to 1 in steps of -1. This corresponds to looking from the BOTTOM of the list (array) to JUST BELOW THE TOP.

Why not to the top? Because you will always be comparing each entry with the one ABOVE it, so the highest entry (element) you will examine ("point to") will be the one immediately below the top. The top is at A\$(0), so you only want the pointer to go as far as A\$(1).

two IF-THEN "test and branch"

The first IF-THEN compares one entry (A\$(POINTER)) with the entry above it (A\$(POINTER - 1)). In English, this test translates as:

"If the entry you are examining is greater than, or equal to, the entry ABOVE it, then leave them where they are".

Why "or equal to"? Simply because it is pointless swapping two entries around if they are identical. (Yes, I know that there are no identical entries in the example list - don't pick holes!).

The only time you need to swap entries is if the lower of the two should lie above the upper one.

The second IF-THEN simply tests the FLAG to see if it has been set as a result of the use of the SWAP routine (see later). If the FLAG variable has a value of 1 then a swap took place at some point during the pass, and it is possible that further swaps need to be made. If the FLAG has been set, the test causes a jump back to an earlier point in the routine where the flag is unset (by being assigned a value of 0) before launching into another pass through the list (array).

a "SWAP" routine

Computers aren't anywhere near as clever as we are (open to argument), so the simple process of swapping the positions of two entries becomes a tactical nightmare unless you "think computer". You can't just say:

```
A$(POINTER) = A$(POINTER - 1)
A$(POINTER - 1) = A$(POINTER)
```

as the first statement wipes out the entry in A\$(POINTER) and replaces it with a copy of the entry in A\$(POINTER - 1). Both entries would then be the same. Twit.

You have to involve another variable as a temporary store for one of the two entries. I chose G\$ because I'm funny that way.

The SWAP routine thus wriggles about like this:

```
G$ = A$(POINTER)           copy lower entry into G$
A$(POINTER) = A$(POINTER - 1) copy upper entry into lower position
A$(POINTER - 1) = G$       copy G$ (which is currently the former
                             lower entry) into the position of the
                             former upper entry
```

The SWAP routine then accomplishes one final thing: it sets the FLAG to indicate that at least one swap took place during the pass.

The SWAP routine might be in use 20 times in one pass; the FLAG will be set 20 times, but its value will only ever be a maximum of 1.

the end of the loop

i.e., change the value of the POINTER so that it "moves up" one entry.

Isolated like this, the Bubble sort routine is of little use. It needs two more components: a section which fills the A\$() array with entries, and a section which prints the list once it has been sorted. You don't HAVE to print the list of course - in a "real" application you might not be in the least interested in seeing the sorted list; you might store it on disk/cassette, and/or begin searching for specific items (something we'll come to later in the series).

Next issue, we'll look at a complete working program.

~~~~~

(continued from p.12)

### FICTION

Either way, the vicinity was not a place to be found by the guards, and the guests were departing rapidly, leaving the task of explaining the marquee and its victual remnants to the Squire, a staunch character whom, it was asserted by many, held the Key to the Universe. Equally, many claimed that honour for a lesser-known individual called Yale, but as events were to show, the true holder was a piscean inhabitant of the watery planet Fission, who went by the name of Chubb. At that time, Chubb was a recluse, having been hounded from his home solar system after an episode following his invention of the Integrated Circuit. Public Relations experts decided that his use of the word "chip" to describe the invention (which resulted in the phrase Fission Chips) was anti-social and betrayed a warped sense of humour, so he was exiled to a distant part of the galaxy for his pains. He had fled to one solar system after another until finally he had become embroiled in the politics of the REMpire, which wanted to fry him publicly for his behaviour.

But I digress. Again.

~~~~~

**USA AREA CODE 303
IS COLORADO**

AMERICAN

BULLETIN BOARD LIST

COLORADO

**Tel Nos are
for USA
callers.**

Courtesy of ROCKY MOUNTAIN 99ERS "TIC TALK" May 1985

AD-COMM TBBS 300/1200 BAUD
PHONE: (303) 755-3888
HOURS: 24 HOURS A DAY
SYSOP: ??
SPONSOR: ??
COMMENTS: ADVERTISING COMMERCIAL
MESSAGES, SIG SECTIONS

APPLE PI 6BBS
PHONE: (303) 469-7541
HOURS: 24 HOURS A DAY
SYSOP: STEVEN MARTIN
SPONSOR: APPLE PI CLUB OF DENVER
COMMENTS:

THE ASYLUM
PHONE: (303) 989-5215
HOURS: 24 HOURS A DAY
SYSOP: SYS/DOC
SPONSOR: SYS/DOC
COMMENTS: RUNS ON A COMMODORE 64

ATRARI BBS
PHONE: (303) 758-6233
HOURS: 6.00 PM - 10.00 PM MON-SUN
SYSOP: STEVE
SPONSOR: VIDED WORLD
COMMENTS:

AURORA-NET 300/1200 BAUD
PHONE: (303) 343-8401
HOURS: 24 HOURS A DAY
SYSOP: KEN SCROGAN
SPONSOR: THE COMPUTER-LINE
COMMENTS:

THE BAHAI BBS
PHONE: (303) 449-3306
HOURS: 24 HOURS A DAY
SYSOP: FRANK HAENDEL
SPONSOR: FRANK HAENDEL
COMMENTS:

THE CONTROLLER BOARD 300/1200 BAUD
PHONE: (303) 698-9423
HOURS: 24 HOURS A DAY
SYSOP: MIKE HASTINGS
SPONSOR: MIKE HASTINGS
COMMENTS:

GBBS 11
PHONE: (303) 693-1064
HOURS: (SEE COMMENTS)
SYSOP: GREG SCHAEFER
SPONSOR: GREG SCHAEFER
COMMENTS: WEEKENDS: 24 HOURS
WEEKDAYS: 3.00-7.00 AM

INTERCOMEX
PHONE: (303) 367-1935
HOURS: 24 HOURS A DAY
SYSOP: BERNARD SAULTER
SPONSOR: BERNARD SAULTER
COMMENTS:

LAKEWOOD RCPM/BBS
PHONE: (303) 985-1108
HOURS: 24 HOURS A DAY
SYSOP: GARY SHAFFSTALL
SPONSOR: INT'L SOFTWARE SERVICES
COMMENTS: LARGE CATALOG OF CP/M 2.2
PUBLIC DOMAIN SOFTWARE

ODYSSEY 1
PHONE: (303) 691-0169
HOURS: 24 HOURS A DAY
SYSOP: ??
SPONSOR: ODYSSEY MAGAZINE
COMMENTS: X-RATED BOARD
FEE REQUIRED

OFF THE WALL
PHONE: (303) 443-3367
HOURS: 24 HOURS A DAY
SYSOP: ROB POLAND
SPONSOR: MICRO SQUIRE
COMMENTS:

PARAPSYCHOLOGY INFO NET
PHONE: (303) 722-6210
HOURS: 24 HOURS A DAY
SYSOP: BILL TEDDER
SPONSOR: ??
COMMENTS: BBS DEVOTED TO EXPLORING
VARIOUS PSYCHIC PHENOMENA

PHOTO TECH BOARD
PHONE: (303) 781-1079
HOURS: 24 HOURS A DAY
SYSOP: JIM GERRITZ
SPONSOR: ??
COMMENTS: FORUM FOR CAMERA TECHS TO
EXCHANGE IDEAS AND INFO

**Times are not specified but are
probably local Colorado time**

CORPORATE HQ TBBS
 PHONE: (303) 297-9127
 HOURS: 24 HOURS A DAY
 SYSOP: BILL GEORGE
 SPONSOR: BILL GEORGE
 COMMENTS: VARIOUS TRS-80 DOWNLOADS

PINECLIFFE RCP/M
 PHONE: (303) 642-3034
 HOURS: 24 HOURS A DAY
 SYSOP: CRAIG BAKER
 SPONSOR: ??
 COMMENTS:

DATA BANK
 PHONE: (303) 0167
 HOURS: 24 HOURS A DAY
 SYSOP: HARRY BOLLS
 SPONSOR: ??
 COMMENTS: DOWNLOADS MESSAGES
 IBM WITH RBBS

THE POLICE STATION
 PHONE: (303) 373-1417
 HOURS: 24 HOURS A DAY
 SYSOP: MR SOFTWARE
 SPONSOR: MR SOFTWARE
 COMMENTS:

DENHUG BBS
 PHONE: (303) 423-3224
 HOURS: 24 HOURS A DAY
 SYSOP: LARRY MEANS
 SPONSOR: HEATHKIT ELEC CENTRE
 COMMENTS: NON-MEMBER SIGN-ON
 IS "PUBLIC"

RCPM/RBBS
 PHONE: (303) 499-9169
 HOURS: (SEE COMMENTS)
 SYSOP: JACK RILEY
 SPONSOR: JACK RILEY
 COMMENTS: WEEKDAYS 7.30 - 11.30 PM
 WEEKDAYS 12.00 - 9.00 PM

DENNY'S HOUSE
 PHONE: (303) 751-2546
 HOURS: ??
 SYSOP: DENNIS DYER
 SPONSOR: DENNIS DYER
 COMMENTS:

STAR BOARD
 PHONE: (303) 455-3113
 HOURS: 24 HOURS A DAY
 SYSOP: MARK JOHNSON
 SPONSOR: MARK JOHNSON
 COMMENTS: HAS A SUB-BOARD FOR TI-99

FIDO-NET
 PHONE: (303) 973-9338
 HOURS: 24 HOURS A DAY
 SYSOP: OSCAR BARLOW
 SPONSOR: ??
 COMMENTS: IBM PC: DOWNLOADS
 AND MESSAGES

TBBS #1 300/1200 BAUD
 PHONE: (303) 741-4071
 HOURS: 24 HOURS A DAY
 SYSOP: CLIFFORD WINDSLOW
 SPONSOR: APPARAT, INC
 COMMENTS: NEWDOS/80 SUPPORT
 PROGRAM DOWNLOADS

FIDO-NET #334 300/1200 BAUD
 PHONE: (303) 534-5456
 HOURS: 24 HOURS A DAY
 SYSOP: MIKE PORTUGAL
 SPONSOR: MICRO BROKER
 COMMENTS: MESSAGES AND DOWNLOADS

TBBS #2 (UFO-NET)
 PHONE: (303) 278-4244
 HOURS: 24 HOURS A DAY
 SYSOP: BOB DRATCH
 SPONSOR: BOB DRATCH
 COMMENTS:

FIDO-NET SANYO
 PHONE: (303) 431-0051
 HOURS: 24 HOURS A DAY
 SYSOP: ??
 SPONSOR: ??
 COMMENTS: SANYO USERS BBS

TBBS HEADQUARTERS 300/1200 BAUD
 PHONE: (303) 690-4566
 HOURS: 24 HOURS A DAY
 SYSOP: DAVE EBERT
 SPONSOR: EBERT PERSONA COMPUTERS
 COMMENTS: SHOWCASE BBS FOR EBERT'S
 TBBS SOFTWARE SYSTEM

BBBS EXPERIENCE SOFTWARE
 PHONE: (303) 798-0792
 HOURS: SEE COMMENTS
 SYSOP: ??
 SPONSOR: EXPERIENCE SOFTWARE
 COMMENTS: MON-FRI: 6.00 PM - 10.00 AM
 WEEKENDS: 24 HOURS

THE UNKNOWN BBS
 PHONE: (303) 988-8155
 HOURS: 24 HOURS A DAY
 SYSOP: HENRY BIRDSEYE
 SPONSOR: HENRY BIRDSEYE
 COMMENTS: ALL JOKE BBS
 DENVER AREA BBS LIST

BBSing

by LLOYD GALENZOSKI from the newsletter of the OTTAWA TI-99/4 USERS GROUP

One of my favourite passtimes is BBSing (note the extra "B"). At the present time, I maintain regular contact with ten systems, including our own.

Calls to distant boards are always made between midnight and 8 a.m., as these are the most economical hours.

Pterm is my favorite emulator program. Although, there are times when I need to switch to TE II, so that I can download programs. Switching while on line is a simple process and allows effective use of the most efficient features of each of these emulators.

The average amount of time, on line, is about fifteen minutes. At about twenty-five cents per minute, that means a typical call would cost about \$3.75. I probably make two calls per week, which means the weekly cost is about \$7.50. Considering the news, information and pleasure I get out of it, I consider this to be a bargain. Once you get to know your way around, you can get an awful lot of information in fifteen minutes. In addition to software and expanded knowledge of the system, it gives me something to talk about with my TI friends.

The following is an extract of some of the information downloaded from the Bull Board, located in the vicinity of Washington D.C. This will give you an idea of the kinds of things you can learn and enjoy yourself at the same time.

24K OF DATA STORAGE

If you need to start with quite a bit of data or would like to change programs, but save the data after you press FCTN QUIT then you can set up the 24K of High-Memory in the PEB as a single data file called "EXPMEM2". You open this file just as you would a disk file with one exception - you must PRECEDE the OPEN statement with a CALL LOAD to the location -24574 as follows:

```
For INT/VAR files - 24
For DIS/VAR files - 16
For INT/FIX files - 8
For DIS/FIX files - 0
```

Here's an example:

If you want to open up the Expansion Memory for Display, Variable 80 files this is what you would do:

```
100 CALL INIT
110 CALL LOAD(-24574,16)
120 OPEN #1:"EXPMEM2",RELATIVE,UPDATE,DISPLAY,VARIABLE 80
```

Then continue as you normally would.

If you want to store both data and assembly language routines at the same time do this:

KNOW YOUR SHIFTS

Some of you may be aware of the Shift 8 3 8 facility on some TI-INVADERS and MUNCHMAN modules, which allows you access to the programmers' Test Mode. IAN MARTIN suggested that I try Shift 8 3 5 on a Munchman module and see what happened. I did, and although it isn't staggering, it shows that there is yet more on some modules than we have had leaked to us!

~~~~~

MODULAR DEPROGRAMMING  
-----

Information received from yet another source confirms the story about TI's wanton destruction of modules - the train containing the material was seen leaving Lubbock bound for the special waste dumping site - and one has to wonder about the mental processes which went into such a decision. The circuit boards alone would have been eminently saleable to hardware buffs looking to beef up their Editor/Assemblers, or to put specialist software into firmware form. TI, you missed a golden opportunity to not only sell those boards to us, but also replacement Ed/As chips when we fouled up trying to be smart and beef up our Ed/As modules! Lots of goodwill could have been made - oh, I forgot, TI spit on goodwill, as do Hewlett-Packard, IBM, etc., etc., in fact just about every "major" computer firm in the West. Come to think of it, aren't they the guys suffering burnt fingers as a result of their arrogant attitude to the small micro market? One pundit recently predicted that TI would not even figure in the computer sphere in the next decade, and I have a sinking feeling that he could be right.

To date, TI's policies on Consumer Relations seem to lurch from one extreme to another. They release Forth and its source code, yet withhold and then destroy not only valuable modules but also the entire documentation relating to the Operating Systems, Monitors, etc., of all the 99s, as well as deliberately restricting access to the GPL manual, an act of gross irresponsibility in itself.

Are we dealing with a group of co-ordinated, motivated, adult human beings, or just an over-indulged, pouting, spoilt little brat?

~~~~~

A PERSONAL OVERFLOWETHING CUP SITUATION

So much work has appeared in my In-Tray, thanks largely to the efforts of JENNY KEANE, that I have managed to overflow into the November issue of TI-LINES, which, SHOCK! HORROR! might even hit the streets on time...

~~~~~

Good Programming,

Peter Brooks

This page is blank