

Members wishing to submit helpline requests via email can use the email address helpline@quanta.org.uk or if you prefer to use traditional post, please send the helpline request to me via the address printed inside the front cover of the newsletter.

Obviously, we cannot guarantee to answer every query we receive, but we will do our best! Where we have been unable to answer the queries, we may print the help request as an open request in the newsletter to ask if any of the readers can come up with a solution. And, of course, if readers feel that they have a better solution than we came up with, or would like to correct any errors we make, please write to us!

Q. I need to find a routine to sort large amounts of whole numbers very quickly in BASIC. I have tried bubble sorts, shell sorts, quicksorts and so on but they all seem to be so slow in BASIC.

A. Try Steve Poole's "Tally Sort" routines. They were published in three issues of QL Today magazine some years ago. The original article was published in Volume 2 Issue 1 page 40 of QL Today, a very short and simple listing complete with a demonstration. At the time its simplicity deceived me into thinking it was an April Fool joke, as by coincidence it had arrived on April 1st and the name (Tally) was suspicious as I lived in a village called Talybont at the time. I should have known better! It was actually the fastest integer sorting routine I had seen running in BASIC.

The next installment was about a sort routine called Radix, which could be used to sort numbers or strings. Steve described that as an 'arborescent' sort routine (the name chosen because it was related to a sorting technique called 'tree sorting'. This article was in QL Today Volume 2 issue 3 page 6.

Then, in Volume 9 Issue 1 page 47 Steve and Bruno Coativy set about to produce a machine coded demonstration of the Tally sorting technique, which pushed up the sorting speed to about 200,000 numbers per second on his computer at the time!

But for most purposes the BASIC version will be fast enough, apparently able to sort up to 32766 numbers in 10 seconds on a Super Gold Card. I've asked Jochen Merz for permission to reproduce the original Tally Sort listing here:

```
100 REMark TALLY-sort by S. Poole v25 Mar 97
110 CLS : ct = 0 : n = 1000 : pr = 1 : REMark n maximum is 32766
120 DIM x(n),y(n) : dl = DATE
130 FOR f = 1 TO n
140   j = RND(n) : IF pr : PRINT j!!
150   x(j) = x(j) + 1
160 END FOR f
170 IF pr : PRINT \\
180 FOR f = 0 TO n
190   j = x(f)
200   IF j THEN
210     FOR k = 1 TO j
```

```

220         ct = ct + 1 : y(ct) = f
230         IF pr : PRINT f!!
240     END FOR k
250 END IF
260 END FOR f
270 PRINT\\DATE-d1

```

The listing sets up a list of random numbers in lines up to 160 in the array x(), then the Tally Sort itself takes place in lines 180 to 260, by building up a tally using the array y(). The simplicity is a little deceptive and it can be a bit hard to follow what's going on despite the short number of lines. But, believe me, it works, although it needs a fair bit of memory for the Tally array.

In fact, while preparing this article I happened to ask Steve if the technique could be adapted for use with strings in any way. I wasn't confident, as I knew that this routine relied on numeric values. Steve at first also thought it might not be possible, but I wondered if Steve would be as tenacious as ever in designing miraculous little BASIC listings in his usual style. Over the next few weeks we corresponded fairly continuously about this as Steve tackled this challenge with his usual vigour. In time, I hope he will present an article on it and if his new string routine is anywhere near as fast as his numeric sort routines, I think we could well be re-writing the book on sorting algorithms!

Q. What is the wiring for a QL with USA-style 9-pin D sockets?

A. I'm grateful to Dex on the QL Forum for the answer to this one. Although he sockets look like the equivalent serial port sockets on a PC, the wiring is not the same:

English QL (RJ connecotor): German/American (9 pin Canon):

```

6 ----- 5 (+12 V)
5 ----- 9 (CTS SER1, DTR SER2)
4 ----- 4 (DTR SER1, CTS SER2)
3 ----- 8 (RxD SER1, TxD SER2)
2 ----- 3 (TxD SER1, RxD SER2)
1 ----- 7 (GND)

```

Pins 1, 2 and 6 are not connected on the 9 pin Canon connector.

Q. I need to send an email with a very long web address in it. The last time I tried to send one to the mailing list, it got broken into two and people weren't able to click on the link. Is there a way around this?

A. One solution for emails can be to enclose the URL with < and > symbols, e.g. <URL> . This flags it as a URL or web address and so prevents it being chopped.

An alternative is to use a URL shortening service, such as www.tinyurl.com (there are several of them on the internet). TinyURL quote the example on their website of how to convert something fairly unreadable like http://www.amazon.com/Kindle-Wireless-Reading-Display-Globally/dp/B003FSUDM4/ref=amb_link_353259562_2?pf_rd_m=ATVPDKIK_X0DER&pf_rd_s=center-10&pf_rd_r=11EYKTN682A79T370AM3&pf_rd_t=201&pf_rd_p=1270985982&pf_rd_i=B002Y27P3M

into an equivalent URL like "http://tinyurl.com/KindleWireless" which is of course easier to read and cut and paste into an email!

Q. I have just started using QL2K and need to get a copy of unzip into it. I tried the "self extract" version of unzip, but couldn't get that to work. I then tried saving a copy of QL unzip to Windows and transferring it to the emulator, but now when I try to EXEC it, it just comes up with an error message with the words "bad parameter". I appreciate that sending QL programs to emulators in a zip file helps avoid the file headers being lost, but this is totally pointless unless I can get Unzip itself across to the emulator in a usable form – catch 22!

A. Indeed. What went wrong is the usual problem of Windows not understanding QL file headers and simply losing them. When you copied the Unzip program to Windows then to the emulator, it lost its file header, so the QL could not execute it. The rather cryptic "bad parameter" message in this case just means file header missing from the QL program.

Here is a short BASIC program to run on the emulator which will repair the damaged Unzip program. It is written for use with Unzip version 5.32. For other versions of Unzip you may need to change the value of "fl" in line 120 to the file length (in bytes) of the version of Unzip you are using (check it on your QL before you transfer it to Windows!). Basically, all the program does is load Unzip into the common heap memory, then uses an SEXEC command with suitable values to restore the lost file header.

```
100 REMark restore Unzip v5.32 executable file header
110 :
120 fl = 108460 : REMark length in bytes of this version
130 ds = 51270 : REMark dataspace in bytes of this version
140 :
150 REMark reserve some space to "fix" the file in memory
160 REMark use ALCHP in place of RESPR in next line if you have ALCHP
170 base = RESPR(108460) : REMark file length of Unzip
180 :
190 REMark change 'flp1_' to drive name required for loading UNZIP
200 LBYTES flp1_unzip,base
210 :
220 REMark change 'ram1_' to drive name required for saving UNZIP
230 SEXEC ram1_unzip,base,fl,ds
240 :
250 REMark if you used ALCHP in line 150 above,remove REMark in next line
260 REMark RECHP base
270 PRINT #0,'Program finished.'
```