Data Compression

One simple way to reduce the size of a file is to identify and replace repeated sequences. For example, `''she sells sea shells on the sea shore''` can be 'factored' as `''she sells (se)a (she)(lls )on t(he s)(ea s)hore''`, provided we have a good way to refer to previous sections of the file. An obvious way to encode this is to add bits to differentiate between raw characters and copies, where a copy is a pointer to a previous location, together with the number of chars to copy. Our example would then be:

`(0,'s')(0,'h'),(0,'e'),(0,' '),(0,'s'),(0,'e'),(0,'l'),(0,'l'),(0,'s'),(0,' '),(1,-5,2),(0,'a'),(0,' '),(1,-13,3),…`

For this exercise, we define the compressed format as:

A 0 bit followed by 8 bits represents a single byte (=9 bits total).

A 1 bit is followed by 16 bits that store how many bytes ago we should start copying from (relative to the current position), followed by 6 bits that store the number of bytes to copy (=23 bits total)

There is no benefit to copying only 2 chars, since (0,char1),(0,char2) only costs 18 bits whereas (1,offset,2) costs 23. Therefore 6 bits can represent lengths from 3-66 instead of 0-63. This means we will really compress the example to `''she sells sea (she)(lls )on t(he s)(ea s)hore''`, saving 2 bytes over the original.

We would like you to write both a compressor and decompressor for binary files. Furthermore, we want something that runs as fast as possible, since the input could be very large. You can use any language you like. Feel free to ask if you have any questions. Good luck!