M.Sc. in High-Performance Computing
5613 - C programming
Assignment 2

Marina Krstic Marinkovic
(mmarina@maths.tcd.ie)
School of Mathematics, TCD

Rules

To submit, make a single tar-ball with all your code and a pdf of any written part you want to include. Submit this via `msc.tchpc.tcd.ie` or via email to `mmarina@maths.tcd.ie` by the end of **Tuesday October 25th**. Marks will be given for the efficiency of your implementation. Late submissions without prior arrangement or a valid explanation will result in reduced marks.

Question

1. (a) Write a `C` program that reads a string from the first line of an input text file `inputfile.txt` and prints to an output file `outfile.txt` sorted characters from the read-in string.

   Example of an input file `inputfile.txt`:

   ```
   the quick brown fox jumps over the lazy dog
   cozy lummox gives smart squid who asks for job pen
   ```

   Your output should contain only the printout of the sorted characters (including the empty spaces) of the **first line** of the file and ignore the rest. In order to show the blank spaces and possibly other punctuation characters that might be elements of your string, print the result of the sort between the signs '<' and '>'. For the above example, `output.txt` would read:

   ```
   <          abcdeeefghhijklmnoooopqrrsttuuvwxyz>
   ```

   (b) Extend your code written in (a) so that it contains functions for **two different sorting algorithms of your choice**, e.g.

   ```
   void bubblesort(char string[],int length)
   ```

and

```
void quicksort(char string[],int low, int high)
```

which would again sort the characters in a read in string up to the end of the first line of the input file. Measure the time of the execution of both functions on several examples.

Your program should write the result of the two sorting algorithms in two different files (`output1.txt` and `output2.txt`), where the first line of each file should contain the sorted elements of the read-in string and the second line should contain the result of the timing in the following format `"Time spent for <name_of_function>: X.YZe-02 sec"`.

In the above example, the file `output1.txt` would read:

```
<        abcdeeefghhijklmnoooopqrrsttuuvwxyz>
Time spent for bubblesort: 1.20e-05 sec
```

and the file `output2.txt` would read:

```
<        abcdeeefghhijklmnoooopqrrsttuuvwxyz>
Time spent for quicksort: 1.00e-05 sec
```

N.B. Timing will depend on the type of CPU on the machine where your program is run, as well as on the current CPU usage. Therefore, by repeating the test, you will probably not get the exactly same timing.