COS132: 6th Practical: April 2023

Theme from Syllabus of the Week: **RECURSION**

Preliminaries:
Before you continue with this practical, **read** page → **https://en.wikipedia.org/wiki/Quicksort**

Overview:
In this practical we want to implement our own (slightly different) variation of Quicksort in C++
The **differences of our variation** (versus the standard algorithm) are as follows:

- Instead of "feeding" the array as a parameter into the functions we use a global array which all functions can "see" and access;
- We provide the function with an additional character parameter which shall indicate whether the global array shall be sorted from smallest to largest, or rather from largest to smallest.

Given is the following rudimentary C++ program:

```
#include <iostream>
using namespace std;

int globalArray[S];      // The actual size S will be entered by the Tutor !
  // The actual values in the array will also be initialized by the Tutor !
```

```
void my_own_Quicksort_Version(char direction, int lo, int hi)
{
    if (direction == '<')
    {
     // Insert here: Your code for quick-sorting from smallest to largest
    }
    else
    if (direction == '>')
    {
     // Insert here: Your code for quick-sorting from largest to smallest
    }
    else cout << " error ";
};
```

```
int main()
{
    // Tutor will insert code here to initialize the global array !

    int  l; // Tutor will give the initial value to this variable !
    int  h; // Tutor will give the initial value to this variable !
    char d; // Tutor will give the initial value to this variable !

    // Tutor will make this Test-Call to your own function of above:
    my_own_Quicksort_Version(d, l, h);

    // Tutor will use this code to inspect the final global array,
    // whereby Tutor will also replace size S by a proper number:
    int c = 0;
    while(c<S) { cout << globalArray[c]; c++ ; }

    return 0;
}
```

Please turn the page →

Your Task:
Complete the rudimentary C++ of above, such that it correctly implements the above-mentioned *recursive* functionality (similar to what is shown on the above-mentioned Wikipedia web-page).

Attention!

ASK the Lab-Tutors (in the weekly Colour-Labs) for **advice** before you submit any software which does not work.

NULL Marks will be given to any submission that violates any of the given Requirements**!**
As always, the submitted *file type must* be either *.**txt** or *.**cpp**

Deadline: Monday 24th of April!

Always remember:

In order to understand recursion
you have to understand recursion **:)**

And now: **HAPPY CODING** :)