

5.2 Sorting a stack (graded assignment)

Due 7 Oct 2019 by 23:59 **Points** 10 **Submitting** an external tool

Write a program that uses recursion to sort the elements on a stack, without using any other containers like vectors, arrays, or even other stacks. For this purpose, write a function **void sortStack(stack<int>& s)** that works as follows:

- If the number of elements on the stack is smaller than 2, return. (Any stack with less than 2 elements is sorted by definition.)
- If not, store the top element of the stack in a variable and remove it from the stack.
- Repeat the following steps until done:
 - Recursively sort the remaining stack.
 - If the separately stored element is smaller than the current top element of the stack, put the stored element on the stack and you are done.
 - If not, exchange the current top element and the separately stored element and continue.

For this assignment, you are using the container class **stack**, just like you have been using **vector** before. From **stack**, you need the functions **push**, **pop**, **top**, **size**. You can find its description on the [c++ reference site](http://www.cplusplus.com/reference/stack/stack/) [\(http://www.cplusplus.com/reference/stack/stack/\)](http://www.cplusplus.com/reference/stack/stack/).

Your program shall read a sequence of whole numbers from a file. (see zyBook Chapter 7.5) The name of the file shall be taken from the command line parameters of your program. You can learn how **main()** reads parameters on [cplusplus.com](http://www.cplusplus.com/articles/DEN36Up4/).

[\(http://www.cplusplus.com/articles/DEN36Up4/\)](http://www.cplusplus.com/articles/DEN36Up4/) (If needed, you can convert a command line parameter to a string like this: **string fileName = argv[1]**)

If no file name is given, or if the file does not exist, your program shall print an error message. If the file can be opened, the program should read the numbers from the file into the stack, sort the stack, and then print the contents of the sorted stack. (It is OK to destroy the stack while printing its contents.) Correct runs of your program look as follows:

```
./stack-sort
an error occured: no input file name given

./stack-sort stack-foo
an error occured: could not open input file stack-foo

./stack-sort stack-numbers
-3 -1 0 2 7 99
```

You do not need to check for proper file contents. The first non-numerical item in the file, or the end of the file, will end the reading of numbers. Your program is **not** expected to print an error message in case of a non-numerical item in the file.

For your own testing purposes, create one or more small files containing whole numbers in random order.

This tool needs to be loaded in a new browser window

Load 5.2 Sorting a stack (graded assignment) in a new window