Mariya Eggensperger
CST 370, Spring 2017
Dr. Feiling Jia
Design/Analysis
of Algorithms

Programming 2 Submission : Stacks

1. Download the sample code of a stack class (Stack class namely, Stack.h, Stack.cpp, Sample_Stack_tester.cpp) from the iLearn. Make a project with the three files.

Your assignment has two parts

a) Design an algorithm to sort a group of numbers in ascending order using two stacks. You can assume that the numbers are initially given to you in one of the stacks. You can assume that the stack data structure is available to you (i.e., you use can use the functions push, pop, top, isempty in the stack). Describe the steps of the algorithm in simple English (i.e., pseudo code).

Figure 1 Here is a simple implementation of a sorting algorithm in pseudocode. All English pseudocode is in comment.

```
void sort_stack_ascending(Stack &s1, Stack &s2) {
   // While stack 1 is not empty
   while(!s1.empty())
   {
      int temp_item = s1.top(); // Create a temporary integer to store the stack 1 top value
      s1.pop(); // Remove the top item from stack 1
      // While stack 2 is not empty and the temporary integer
      // item is less than the top item in stack two
      while(!s2.empty() && temp_item < s2.top())
      {
         s1.push(s2.top()); // Add the higher stack one item to the top of stack two
         s2.pop(); // And remove the value from stack two
      }
      s2.push(temp_item); // Add the swapped value to stack 2
   }

}
```

b) Implement the above algorithm using the sample stack class provided to you on iLearn. Note that you do not need to change Stack.h and Stack.cpp files. You only need to make changes to the Sample_Stack_tester.cpp (i.e., the driver) file. In this assignment, you should begin by pushing the numbers (1, 5, 3, -3, 4, 8, 10, -5) onto the first stack in the given order.

After your program executes one of your stacks should contain the numbers sorted in ascending order (i.e., the top of the stack has the highest value). This means that your sorted stack should look like this.

10
8
5
4
3
1
-3
-5

Pop elements out of the stack and display them on the screen. Note that the numbers will be displayed in descending order, as the highest value is present at the top of the stack.

For the example stack given above, the numbers will be displayed in the following way. 10, 8, 5, 4, 3, 1, -3, -5

Test the functioning of your program with two other sample inputs Sample Input 1: 1, 5 -4, 6, 2 Sample Input 1: -1, -4 -4, 6, 6, 9

Figure 2 As instructed by the homework specifications, the stacks program begin by pushing the numbers (1, 5, 3, -3, 4, 8, 10, -5) onto the first stack in the given order. As is evident from this image, the algorithm implemented within the program, **void** sort_stack_ascending(Stack & s1, Stack & s2); does function and sort all numbers in ascending order. The highest value is thus, placed at the top of the stack.

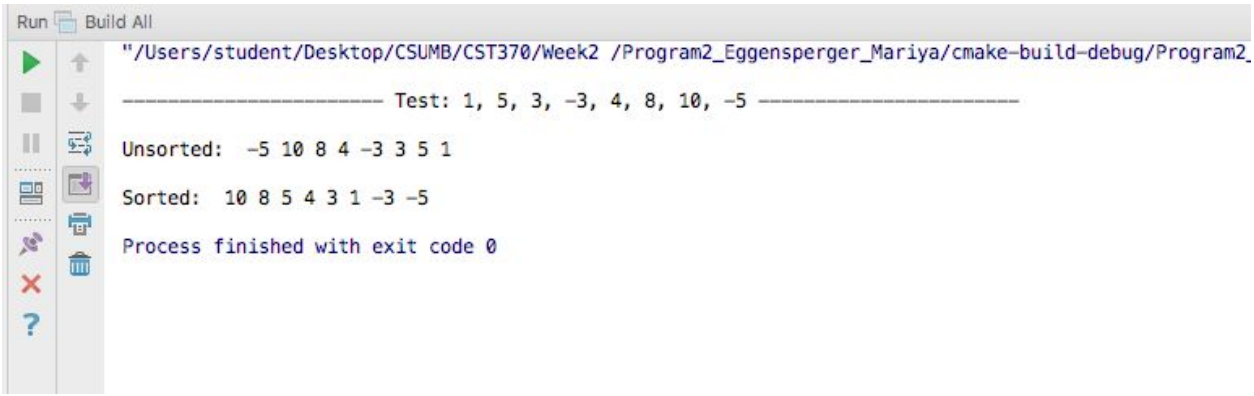

Figure 3 This is a second run to test the sort_stack_ascencing() function. The numbers 1, 5, -4, 6, 2 are pushed into stack 1 and sorted accordingly in stack 2 with the highest value at the top of the stack.

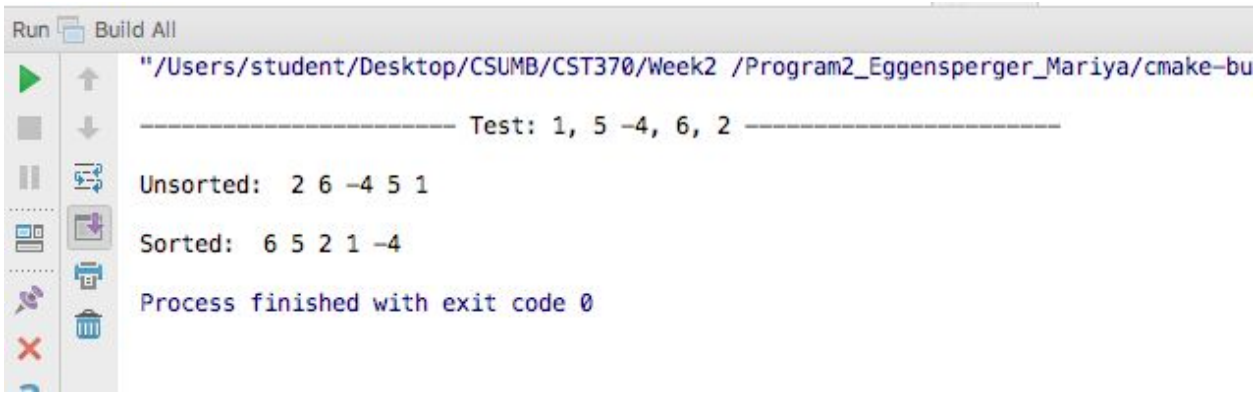

Figure 4 This is a third run to test the sort_stack_ascending() function. The numbers -1, -4, -4, 6, 6, 9 are pushed into stack 1 and accordingly sorted in ascending order in stack 2.