Due July 20 at midnight to eCampus

First Name Last Name UIN

User Name E-mail address

Please list all sources in the table below including web pages which you used to solve or implement the current homework. If you fail to cite sources you can get a lower number of points or even zero, read more: Aggie Honor System Office

Type of sources			
People			
Web pages (provide URL)			
Printed material			
Other Sources			

I certify that I have listed all the sources that I used to develop the solutions/codes to the submitted work.
"On my honor as an Aggie, I have neither given nor received any unauthorized help on this academic work."

Your Name Date

Programming Assignment 2 (140 points)

1. (30 pts) Implement a stack ADT using STL vector to support the operations: `push(), pop(), top(), empty().`
2. (10 pts) Test the operations of the stack class for correctness. What is the running time of each operation? Express it using the Big-O asymptotic notation.
3. Stack Application

   Use the implemented stack class as an auxiliary data structure to compute spans used in the financial analysis, e.g. to get the number of consecutive days when stack was growing.

Definition of the span:

Given a vector X, the span $S[i]$ of $X[i]$ is the maximum number of consecutive elements $X[j]$ immediately preceding $X[i]$ such that $X[j] \leq X[i]$

Spans have applications to financial analysis, e.g., stock at 52-week high

1. Example of computing the spans S of X.

X	6	3	4	5	2
S	1	1	2	3	1

2. (10 pts) Compute the spans based on the definition above:

```
Algorithm spans1(x)
Input: vector x of n integers
Output: vector s of spans of the vector x
for i = 0 to n-1 do
   j = 1
   while (j <= i ∧ x[i-j] <= x[i])
      j =  j + 1
   s[i] = j
return s
```

1. (10 pts) Test the algorithm above for correctness using at least three different input vectors.
2. (10 pts) What is the running time function of the algorithm above? The function should define the relation between the input size and the number of comparisons perform on elements of the vector. How can you classify the algorithms using the Big-O asymptotic notation and why?
3. (20 pts) Compute the spans using a stack as an auxiliary data structure storing (some) indexes of x.

   Algorithm spans2:

   (a) We scan the vector x from left to right
   (b) Let i be the current index
   (c) Pop indices from the stack until we find index j such that x[i] $<$ x[j] is true
   (d) Set s[i] $=$ i $-$ j
   (e) Push i onto the stack

4. (10 pts) Test the second algorithm (spans2) for correctness using at least three different input vectors. Your program should run correctly on TA's input.
5. (10 pts) What is the running time function of the second algorithm? The function should define the relation between the input size and the number of comparisons perform on elements of the vector. How can you classify the algorithms using the Big-O asymptotic notation and why? Compare the performance of both the algorithms.
6. (10 pts) Programming style, and program organization and design: naming, indentation, whitespace, comments, declaration, variables and constants, expressions and operators, line length, error handling and reporting, files organization. Please refer to the PPP-style document.

7. Instructions

    (a) Your files should be arranged as below

        i. Declaration of `My_stack` class in `My_stack.h`

        ii. Definition (implementation) of `My_stack` class in `My_stack.cpp`

        iii. Algorithm implementations and the `main()` function in the file `Application.cpp`

    (b) Compile your program by
    `g++ -std=c++11 *.cpp`
    or
    `make all`

    (c) Run your program by executing
    `./Main`

    (d) Testing code in `Application.cpp` and collect output data for your report.

8. Submission

    (a) Tar the files above and any extra files. Please create your tar file following the instructions.

    (b) "turnin" your tar file to eCampus no later than **July 2**0.

    (c) (20 points) Submit a hard copy of cover page, report (see below), `My_stack.h`, `My_stack.cpp` and `Application.cpp` in lab to your lab TA. Find a cover page .

        i. Typed report made preferably using "LyX/LaTeX"

            A. (2 pts) Program description; purpose of the assignment

            B. (4 pts) Data structures description

                - Theoretical definition
                - Real implementation
                - Analysis of the best and worst scenarios for computing spans.

        ii. (2 pts) Instructions to compile and run your program; input and output specifications

        iii. (2 pts) Logical exceptions (and bug descriptions)

        iv. (5 pts) C++ object oriented or generic programming features, C++11 features

        v. (5 pts) Testing results