HW-2: CSC241\_Sec2 (Data Structure and Algorithm)

Due Date: 10/28/2021 (11.59:59 PM)

**Special Instruction**

Your HW will be graded based on correctness and clarity. Keep you answer precise and to the point. If any question ask for justification of your answer/claim, you may receive a 0 if you merely provide an answer without justification. Your answer can be printed or handwritten. If handwritten, make sure your scanned version is legible. All sub questions carry equal weights unless specified otherwise. Finally, please check the HW rules at the end.

**What to do:** Reach to me ASAP if you have any confusion and/or have any emergency that may deter you to submit HW on time. Never hesitate to ask me if any of the previously discussed topics is unclear and you need some more discussion.

**What not to do:**

1. Ask to verify your solution
2. Ask to debug/analyze your code

-------------------------------------------------------------------------------------------------------------------------

**Question 1 (40 Points):** In the class, we have seen an array can be used to implement a stack. Now, let’s tweak the problem a little. **Can you implement two different stack (say S1 and S2) using a single array**? In this case, if you call *push1(item)*, the item will be pushed to S1, and if you call *push2(item)*, the item will be pushed to S2. In the same manner, if you call *pop1()*, the top item from S1 will be poped, and if you call *pop2()*, the top item from S2 will be poped.

* The above-mentioned operation is possible when you divide the array in to 2 equal parts and then implement the 2 stacks in the two halves of the stack. The first half will contain the S1 and the second half will contain S2
  + Pseudo Code:
    - First Stack(left of array) will be 0 to index/2-1 and the second stack will be index/-1 to last index.
    - So if there is a array with the size of 8 then the First Stack will have 4 indexes and the Second stack will have the latter 4 indexes.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0 |  |  | (8/2)-1 |  |  |  | Last |

First Stack Second Stack

**Question 2 (20 Points):** Consider the following RPN expression: 8 2 2 ^ / 3 3 \* + 3 1 \* -

* Evaluate its value.
  + 8 4 / 33 \* + 3 1 \* -
  + 2 3 3 \* + 3 1 \* -
  + 2 9 + 3 1 \* -
  + 11 3 1 \* -
  + 11 3-
  + 8
* Convert it to infix notation.

Here, ^ is the exponential operator.

**Question 3 (10+30=40 points):** Consider the following Infix mode equation:

* Express it by RPN/Postfix notation.
  + 7 3 + 2 \* 10 + 2 3 + 3 \* /
* Implement the RPN notation in a STACK. **To earn full credit you must show each step**.
  + Answer = 2

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  | + | \* |  |  |
| + | \* | + | 3 | 3 | / |  |
| 3 | 2 | 10 | 2 | 5 | 15 |  |
| 7 | 10 | 20 | 30 | 30 | 30 | 2 |

**Special Instruction:** Please follow the below mentioned instructions:

* + Your answer must have to be typed.
  + On the D2L submit a single pdf or docx file that contains your answer. ***Please don’t upload zip file.***

**Rules for ALL HW:**

1. If any programming problem is given, the code must be written by yourself. DO NOT copy code from anywhere else.

2. You can discuss the problem sets and study together in group, but when it comes to formulating/writing solutions you must work alone independently; i.e., you should be able to explain your answer clearly to anyone else (including the TA and the instructor). Note that this says discuss in group — copying homework solutions from another student, from the Internet, solution sets of friends who have taken this course or one similar to it previously, or other sources will be considered **cheating** and referred to the university. *At the beginning of each submission, you should explicitly list the people you worked with.*