HW-2: CSC241 (Data Structure and Algorithm)

Due Date: 03/02/2023 (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 must need to be printed. Handwritten submissions will not be evaluated***. 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 (30 Points): Let we have a queue and a stack of size n, where $5 \leq n \leq 20$. Now, write a java code that will take n integer values from the user, and then reverse the queue. You need to use the stack to perform this operation and only following standard operations are allowed on queue:

- enqueue(x) : Add an item x to rear of queue.
- dequeue() : Remove an item from front of queue.
- empty() : Checks if a queue is empty or not

No built-in functions are allowed.

Sample Input 1: Number of elements: 6

Elements: 15 87 65 11 93 102

Sample Output 1: 102 93 11 65 87 15

You must test the following two test cases with your code and attach the output.

Sample Input 1: Number of elements: 10

Elements: 10, 20, 30, 40, 50, 60, 70, 80, 90, 100

Sample Output 1: ???

Sample Input 2: Number of elements: 9

Elements: 56, 20, 130, 49, 506, 26, 370, 84, 100

Sample Output 2: ???

Special instructions for Question 1 : Please upload all of the following to D2L separately (not as one zip file):

- A .pdf or .docx that contains your written answers. This file must show your **typed** code.
- Attach a screenshot of the output from your computer console/terminal.

Question 2 (12 Points): Here is a pseudocode for the algorithm which is supposed to determine whether a sequence of parentheses is balanced:

```
declare a character stack
   while ( more input is available)
   {
      read a character
      if ( the character is a '(' )
         push it on the stack
      else if ( the character is a ')' and the stack is not empty )
         pop a character off the stack
      else
         print "unbalanced" and exit
    }
    print "balanced"
```

What will be printed for the following three sequences?

- ())(()
- ((()()))
- ((())
- ((()())()()

Question 3 (10+20 = 30 Points): consider the following expression:

$$F = \frac{D^2 + A * C}{2 * C - B} + \frac{\frac{4C}{A} - \frac{2D}{3}}{A}$$

a) Express it by RPN/Postfix notation.

b) Assume, A = D=10, B = 5, C= 30. Implement the RPN notation in a STACK and show final outcome. **To earn full credit you must show each step.**

Question 4 (12 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 S_1 and S_2) using a single array**? In this case, if you call *push1(item)*, the item will be pushed to S_1, and if you call *push2(item)*, the item will be pushed to S_2. In the same manner, if you call *pop1()*, the top item from S_1 will be poped, and if you call *pop2()*, the top item from S_2 will be poped. If

- You think the above mentioned operation is **possible**, describe how it can be done and provide a pseudocode to implement it. Provide adequate comment in your pseudocode, but make sure they are relevant.
- You think the above mentioned operation is **not possible**, describe why it is not possible.

Question 5 (16 Points): Consider the following applications/scenarios. If you have the option to use STACK, QUEUE, Linked List and Graph, which one you should use for which application. Mention clearly at the beginning, and then explain your answer.

I. Your browser deletes the history past one month.
II. E-commerce websites : category -> subcategories -> products
III. Music player where you can play next or previous song.
IV. While booking bus/flights, you get list of available routes..

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. 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.*