Miderm-2: CSC241 (Data Structure and Algorithm)

Date: 11/02/2021

Special Instruction: Please read the instruction below:

1. Your answer will be graded based on correctness and clarity. Keep your answer precise and to the point. Do not provide irrelevant/unnecessary answers.
2. This is an open book/materials exam. However, discussion with anyone else is not allowed. You must provide a reference from the source you have taken help and should be able to explain your answer clearly to the instructor.
3. **Your answer must have to be typed. However, only for Question 5(b) you are allowed to draw a clear handwritten image, and attach it to your answer.**
4. **You need to upload your answer as a single file (pdf or Docx) by 07.59 PM, 11/02/2021. Don't submit multiple files or don't use any format other than pdf or Docx.**
5. If any question asks for justification of your answer/claim, you may receive a 0 if you merely provide an answer without justification.
6. No change or announcement on the D2L will be made (regarding questions) once it is published. If you think anything is wrong or not clear, state your assumption clearly, and then complete your answer.
7. All questions in a section carry equal weights unless stated otherwise.

Question 1 (12 Points): Let us have a stack with ten integer elements. You wrote a function *reversestack(Stack myStack)*, which reverses all the elements on the stack, i.e., 1st and 10th items swap their position, 2nd, and 9th items swap their position. Hence, we argue that myStack will no longer exhibit the LIFO characteristics once we call the *reversestack(Stack myStack) function*. Is it a valid claim? Why or why not.

Question 2 (10 Points): Consider the code below:

```
Stack<Integer> stackOne = new Stack<Integer>();
Stack<Integer> stackTwo = new Stack<Integer>();

for(int i =0; i<5; i++)
        stackOne.push(i);

stackOne.push(stackTwo.peek()); // Operation 1
stackTwo.push(stackOne.peek()); // Operation 2
stackTwo.push(stackOne.pop());  // Operation 3
stackOne.push(stackTwo.peek()); // Operation 4
```

Determine which of these operations are valid/invalid. You must justify your answer.

Note: While writing the answer, at first, clearly mention whether an operation is valid or invalid. Then describe the reasoning. Keep your reasoning straight and to the point. Do not provide irrelevant or unnecessary texts to enlarge the answer.

Question 3 (9 Points): What is the value returned when stack.peek() is called? Does it have any influence on modifying stack size? Explain your answer.

Question 4 (12 Points): Consider a singly lined list with integer data. Write the pseudocode or logic (not real code) to implement a method "*indexOf(int data)*" that returns the index of the first occurrence of the data. For example, if a linked list has the following items [45, 18, 65, 19, 20, 33, 65, 47, 65], then *indexOf(65)* will return 2.

Question 5 (18 Points): consider the following expression:

$$F = \frac{A}{B} + \frac{C-D}{A}$$

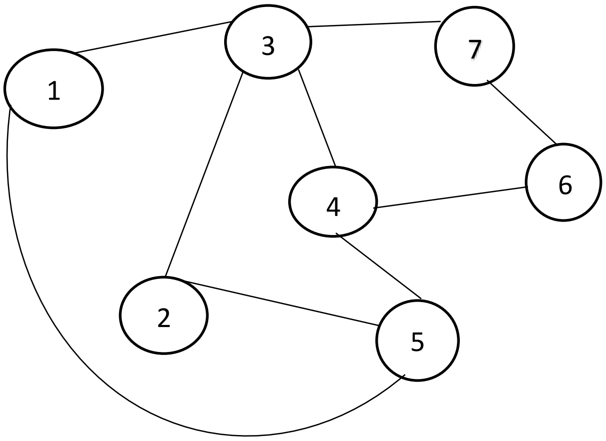
a) Express it by RPN/Postfix notation. **(5 points)**
b) Assume, A = D=10, B = 5, C= 30. Implement the RPN notation in a STACK and show final outcome. **To earn full credit you must show each step. (13 points)**

Question 7 (14 Points): Consider a undirected graph with n nodes, where n is an odd number and $n \geq 4$. Verify and justify the following statements:

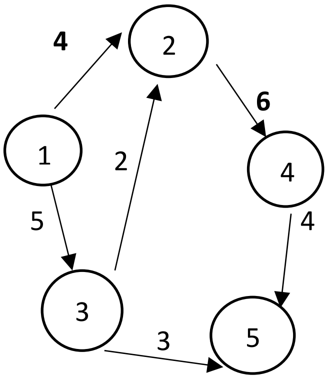
I. It is possible to create a graph where all these n nodes have even degree.
II. It is possible to create a graph where all these n nodes have odd degree.

You must clearly state whether you agree or disagree with these statements and then provide the reasoning.

Question 8 (25 Points):



(1)



(2)

1) Determine the classification of Graph (1) and Graph (2). **(6 points)**
2) Draw their Adjacency Matrix Representation. **(12 points)**
3) Can we reach to any node from any node in Graph (1) and Graph (2)? Why or Why not? **(7 points)**