Assignment # 3

TAKE A PRINT OF THIS ASSIGNMENT AND READ IT ONCE (preferably you should have the copy of this assignment in your pocket, while walking, and wandering around you should read one question at a time and keep thinking about the problem).

Good luck.

Problem 1

- Implement the class **MyStack** which we implemented in the class**.**
- Using **MyStack** Solve the **BalancedParenthesis** problem.
- Using MyStack Implement another class Decimal to B-base convertor. Which should have a static public function **ConvertDecimalToB** which takes as parameter a number N and a integer B (base) and converts the decimal number N into base B.

Problem 2

Describe how you could use a single array to implement three stacks.

Problem 3

How would you design a stack which, in addition to push and pop, also has a function min which returns the minimum element? Push, pop and min should all operate in O(1) time.

Problem 4

Imagine a (literal) stack of plates. If the stack gets too high, it might topple. Therefore, in real life, we would likely start a new stack when the previous stack exceeds some threshold. Implement a data structure SetOfStacks that mimics this.SetOfStacks should be composed of several stacks, and should create a new stack once the previous one exceeds capacity.SetOfStacks.push() and SetOfStacks.pop() should behave identically to a single stack (that is, pop() should return the same values as it would if there were just a single stack).

FOLLOW UP

Implement a function popAt(int index) which performs a pop operation on a specific sub-stack.

Problem 5

In the classic problem of the Towers of Hanoi, you have 3 rods and N disks of different sizes which can slide onto any tower. The puzzle starts with disks sorted in ascending order of size from top to bottom (e.g., each disk sits on top of an even larger one).You have the following constraints:

(A) Only one disk can be moved at a time.

(B) A disk is slid off the top of one rod onto the next rod.

(C) A disk can only be placed on top of a larger disk.

Write a program to move the disks from the first rod to the last using Stacks. **YOU SHOULDN'T use recursion now.**

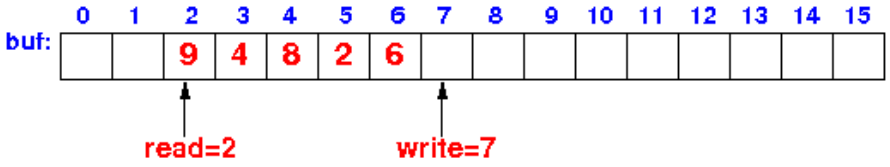
Problem 6

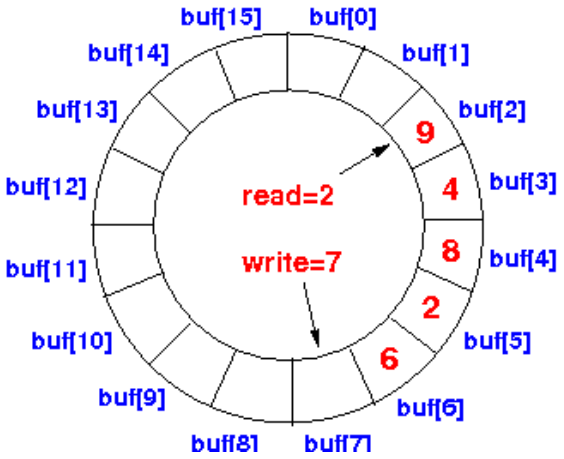
Write a program to sort a stack in ascending order. You should not make any assumptions about how the stack is implemented. The following are the only functions that should be used to write this program: **push | pop | peek | isEmpty.**

Problem 7

Implement the class QUEUE which should have ENQUEUE and DEQUEUE and ISEMPTY and ISFULL functions and internally it should an array in the background. Remember it must have all these operations in constant **O(1)** time just like in STACK, involving no loop at all. THE SIMILAR WAY OF STACK PUSH/POP will NOT work. **WHY?**

HINT: IMAGINE CIRCULARLY (CLOCKWISE), using modulus %???





Problem 8

Implement a **MyQueue** class which implements a queue using two stacks.

(This part is due by Saturday 1st April 2017)