

## HW 2: (Stacks and Queues)

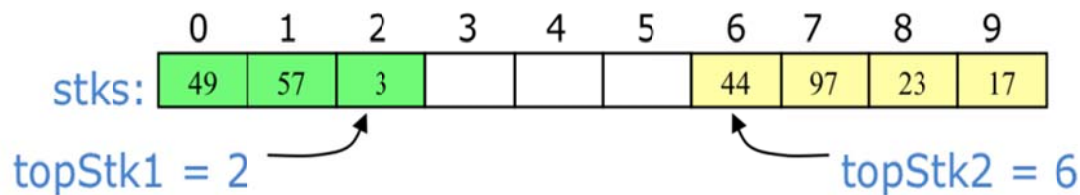
Q1) Create class *ArrayDeque* to implement double ended queue (Deque) (modify *ArrayQueue*). Deque has the following methods:

isEmpty(), size(), first(), last(), addFirst(), addLast(), removeFirst(), removeLast(). [10pts]

Test these methods in main method.

Q2) Create a java class called *DoubleStack* to represent a double stack data structure (see figure below). It is like two stacks shared the same array, one from left side and the other one from right side. However, this doesn't mean the array is divided equally between them. It depends on push and pop operations for each one of them. *DoubleStack* will have : isEmpty1, size1(), push1, top1, pop1, isEmpty2, size2(), push2, top2, and pop2.

Write a main() function to test all the above functionalities.



[10pts]

In the main, create an object of *DoubleStack*, test the methods.

Q3) Write a recursive method *searchStack(LinkedStack<E> S, E x)* which returns true if the stack *S* contains the element *x*, otherwise it return false. (You should not use any extra data structures).

[5pts]