1. Implement Stack using Arrays as well as Linked List
2. Implement Queue using Arrays as well as Linked List
3. Given an infix expression convert into postfix.
4. Given a postfix expression evaluate it.
5. Implement a function reverse() in your Queue Class.
6. Print the shortest path for rat in a maze problem.
7. Implement a Stack using Two Queues i.e your Stack class will have only two data members which are Queues and nothing else.
8. Implement a Queue using Two Stacks
9. Check for duplicate parenthesis in an expression e.g. ((a + b) + ((c+d))) has duplicate parenthesis.
10. The span si of a stock's price on a certain day i is the maximum number of consecutive days (up to the current day) the price of the stock has been less than or equal to its price on day i. Given input array with all stock prices return the spans. We can do this using an array in O(n^2) time but stack can help us do it in O(n) time. Implement the array approach if you can't find a solution using stack
11. Implement a class MinStack using the stack class we have already built. It should support O(1) push, O(1) pop and O(1) getMinimum() functions where getMinimum() returns the minimum element present in the stack. (Hint: You would need two stacks for doing this)
12. A deque is a data structure consisting of a list of items, on which the following operations are possible:

    1. push(x,d): Insert item x on the front end of deque d.
    2. pop(d): Remove the front item from deque d and return it.
    3. inject(x,d): Insert item x on the rear end of deque d.
    4. eject(d): Remove the rear item from deque d and return it.

    Write routines to support the deque that take constant time per operation