1. A Double Ended Queue (Deque and pronounced *deck*) is a generalized type of the Queue data structure that allows insert and delete at both ends.

    a. What are the main operations on deque.
        insertFront(): Adds an item at the front of Deque.
        insertRear(): Adds an item at the rear of Deque.
        deleteFront(): Deletes an item from front of Deque.
        deleteRear(): Deletes an item from rear of Deque.

    b. What are the other operations that should be checked (according to the operation in part a)
        getFront(): Gets the front item from queue.
        getRear(): Gets the last item from queue.
        isEmpty(): Checks whether Deque is empty or not.
        isFull(): Checks whether Deque is full or not.

    c. Implement it using: Circular array, Doubly Linked List (write the algorithm and code using c++)

2. Solve the following question in brief words/sentences:

    1. A linear list of elements in which deletion can be done from one end (front) and insertion can take place only at the other end (rear) is known as _____ while its modified version that allows deletion and insertion at both ends is known as _____.
    2. Queue is implemented using an array Queuearray. The "rear" is the current element pointer and "data" is the new item to be added. For adding a new element int the queue, we use _____
    3. If implemented using an array of size MAX_SIZE, a normal queue gets full when _____
    4. T/F: To delete an element from Queue, Front is incremented by one.
    5. In the linked list implementation of a queue, where is a new element inserted?
    6. The front and rear pointers of a queue are monitored in a linked list implementation. During insertion into a NONEMPTY queue, which of these pointers would change?
        a. only front pointer
        b. only the rear pointer
        c. both front and rear
        d. neither front nor rear
    7. a Queue of *n* elements are to be reversed using another queue. The number of "enqueue" and "dequeue" operations required to do so is:
        a. n
        b. 2n
        c. 4n
        d. This task cannot be accomplished.
    8. Can We implement Queue data structure using a stack data structure?

3. Implement all the codes for the lecture.