Queue ADT

```
interface Queue<T> {
   void enqueue(T value);
   T dequeue();
   T peek();
   boolean isEmpty();
   int size();
}
```

1) Implement a FIFO data structure using an array only
2) Implement a FIFO using a linked list only
3) Implement FIFO using LIFO only
4) Implement a circular queue
   Should contain an extra method called rotate() which rotates the front element to the back of the queue
5) Josephus problem

Deque ADT

```
interface Deque<T> {
   void addFirst(T value);
   void addLast(T value);
   T removeFirst();
   T removeLast();
   T first();
   T last();
   int size();
   boolean isEmpty();
}
```

6) Implement a double ended queue using an array
7) Cards in increasing order (Medium in Leetcode.com)
   You are given an integer array `deck`. There is a deck of cards where every card has a unique integer. The integer on the `i`th card is `deck[i]`.
   You can order the deck in any order you want.
   You will do the following steps repeatedly until all cards are revealed:

   1. Take the top card of the deck, reveal it, and take it out of the deck.
   2. If there are still cards in the deck then put the next top card of the deck at the bottom of the deck.
   3. If there are still unrevealed cards, go back to step 1. Otherwise, stop.

   Return *an ordering of the deck that would reveal the cards in increasing order*.

```
Input: deck = [17,13,11,2,3,5,7]

Output: [2,13,3,11,5,17,7]
```