

## CIRCULAR DEQUE

```
class MyCircularDeque {  
    private int[] arr;  
    private int front;  
    private int rear;  
    private int size;  
    private int capacity;  
  
    public MyCircularDeque(int k) {  
        capacity = k;  
        arr = new int[k];  
        front = 0;  
        rear = -1;  
        size = 0;  
    }  
  
    public boolean insertFront(int value) {  
        if (isFull()) return false;  
  
        front = (front - 1 + capacity) % capacity;  
        arr[front] = value;  
  
        if (size == 0) rear = front;  
  
        size++;  
        return true;  
    }  
}
```

```
public boolean insertLast(int value) {  
    if (isFull()) return false;  
  
    rear = (rear + 1) % capacity;  
    arr[rear] = value;  
  
    if (size == 0) front = rear;  
  
    size++;  
    return true;  
}  
  
public boolean deleteFront() {  
    if (isEmpty()) return false;  
  
    front = (front + 1) % capacity;  
    size--;  
    return true;  
}  
  
public boolean deleteLast() {  
    if (isEmpty()) return false;  
  
    rear = (rear - 1 + capacity) % capacity;  
    size--;  
    return true;  
}  
  
public int getFront() {  
    if (isEmpty()) return -1;  
    return arr[front];  
}
```

```
public int getRear() {  
    if (isEmpty()) return -1;  
    return arr[rear];  
}  
  
public boolean isEmpty() {  
    return size == 0;  
}  
  
public boolean isFull() {  
    return size == capacity;  
}  
}
```