Circular Queue

Implement the MyCircularQueue class:

- MyCircularQueue(k) Initializes the object with the size of the queue to be k.
- int Front() Gets the front item from the queue. If the queue is empty, return -1.
- int Rear() Gets the last item from the queue. If the queue is empty, return -1.
- boolean enQueue(int value) Inserts an element into the circular queue. Return true if the operation is successful.
- boolean deQueue() Deletes an element from the circular queue. Return true if the operation is successful.
- boolean isEmpty() Checks whether the circular queue is empty or not.
- boolean isFull() Checks whether the circular queue is full or not.

```
class MyCircularQueue {
    int* arr;
    int size, capacity;
    int front, rear;
public:
    MyCircularQueue(int k) : capacity(k), size(0), front(0), rear(0), arr(nullptr) {
        arr = new int[capacity];
    }

    bool enQueue(int value) {
        if (size < capacity) {
            if (rear == capacity) {
                rear = rear % capacity;
            }
            arr[rear++] = value;
            size++;
            return true;
        }

        return false;
    }

    bool deQueue() {
        if (size > 0) {
            if (front == capacity - 1)
                front = 0;
            else
                front++;

            size--;
            return true;
        }
        return false;
    }


    int Front() {
        if (size)
            return arr[front];
        return -1;
    }

    int Rear() {
        if (size)
            return arr[rear - 1];
        return -1;
    }

    bool isEmpty() {
        return size == 0;
    }

    bool isFull() {
        return size == capacity;
    }
};
```