

```

import java.util.Arrays;

class ArrayQueue {

    private final int[] items;
    private int rear;
    private int front;
    private int count;

    public ArrayQueue(int capacity) {
        items = new int[capacity];
    }

    public void enqueue(int item) {
        if (isFull()) {
            throw new IllegalStateException();
        }
        items[rear] = item;
        rear++;
        count++;
        //System.out.println("rear = " + rear);
    }

    public int dequeue() {
        if (isEmpty()) {
            throw new IllegalStateException();
        }

        var item = items[front];
        items[front] = 0;
        front++;
        count--;
        return item;
    }

    public void addFromFront(int item) {

    }

    public int removeFromRear()
    {

    }

    public int peek() {
        if (isEmpty()) {
            throw new IllegalStateException();
        }

        return items[front];
    }
}

```

```

    public boolean isEmpty() {
        return count == 0;
    }

    public boolean isFull() {
        return count == items.length;
    }

    @Override
    public String toString() {
        return Arrays.toString(items);
    }

    public int getRear() {
        return rear;
    }

    public int getFront() {
        return front;
    }
}

public class DoubleEndedQueueUsingArray {

    public static void main(String[] args) {
        ArrayQueue queue = new ArrayQueue(5);
        queue.enqueue(10);
        queue.enqueue(20);
        queue.enqueue(30);
        System.out.println(queue);
        System.out.println("Front = " + queue.getFront());

        System.out.println("rear = " + queue.getRear());

        System.out.println("Deleted item = " + queue.dequeue());

        System.out.println(queue);
        System.out.println("Front = " + queue.getFront());

        System.out.println("rear = " + queue.getRear());

        System.out.println("Deleted item = " + queue.dequeue());

        System.out.println(queue);

        System.out.println("Front = " + queue.getFront());

        System.out.println("rear = " + queue.getRear());

        queue.enqueue(40);
        System.out.println(queue);
        System.out.println("Front = " + queue.getFront());

        System.out.println("rear = " + queue.getRear());
        queue.enqueue(50);
        System.out.println(queue);
    }
}

```

```
System.out.println("Front = " + queue.getFront());

System.out.println("rear = " + queue.getRear());

//queue.enqueue(60);
queue.addFromFront(60);
System.out.println(queue);
System.out.println("Front = " + queue.getFront());

System.out.println("rear = " + queue.getRear());
System.out.println("Deleted item = " + queue.removeFromRear());
System.out.println(queue);
System.out.println("Front = " + queue.getFront());

System.out.println("rear = " + queue.getRear());

}

}
```