

```
public class QueueAr{

    private Object[] theArray;
    private int currentSize;
    private int front;
    private int back;
    static final int DEFAULT_CAPACITY = 20;

    public boolean isEmpty()
    {return currentSize == 0;}
    public boolean isFull()
    {return currentSize == theArray.length;}
    public QueueAr(){
        this(DEFAULT_CAPACITY );
    }
    public QueueAr( int capacity ){
        theArray = new Object[capacity];
        makeEmpty();
    }
    public void makeEmpty(){
        currentSize = 0;
        front = 0;
        back = -1;
    }
    public void enqueue(Object x) {
        if(isFull())
            System.out.println("OVERFLOW!");
        back = increment( back );
        theArray[back ] = x;
        currentSize++;
    }
    private int increment(int x){
        if(++x == theArray.length )
            x=0;
        return x;
    }
    public Object dequeue(){
        if(isEmpty())
            return null;
        currentSize--;
        Object frontItem = theArray[ front ];
        theArray[front] = null;
        front = increment(front);
        return frontItem;
    }
}
```