```java
import java.util.Scanner;

class CircularQueue {
    int[] queue;
    int front, rear, size, capacity;

    CircularQueue(int capacity) {
        this.capacity = capacity;
        queue = new int[capacity];
        front = 0;
        rear = -1;
        size = 0;
    }

    void enqueue(int item) {
        if (size >= capacity) {
            System.out.println("OVERFLOW");
            return;
        }
        rear = (rear + 1) % capacity;
        queue[rear] = item;
        size++;
    }

    int dequeue() {
        if (size <= 0) {
            System.out.println("UNDERFLOW");
            return Integer.MIN_VALUE;
        }
        int value = queue[front];
        front = (front + 1) % capacity;
        size--;
        return value;
    }

    int peek() {
        if (size <= 0) {
            System.out.println("UNDERFLOW");
            return Integer.MIN_VALUE;
        }
        return queue[front];
    }

    void display() {
        if (size <= 0) {
            System.out.println("UNDERFLOW");
            return;
        }

        System.out.print("queue > [ ");
        for (int i = 0; i < size; i++) {
            int index = (front + i) % capacity;
            System.out.printf("%d, ", queue[index]);
        }
        System.out.print(" ]\n");
        return;
    }
}

public class Queue {
    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);
        System.out.print("enter capacity > ");
        int capacity = sc.nextInt();
        CircularQueue queue = new CircularQueue(capacity);
```

```
        int choice;

        displayMenu();
        do {
            choice = sc.nextInt();
            sc.nextLine();

            switch (choice) {
                case 1:
                    System.out.print("enter item to enqueue > ");
                    int itemEnqueue = sc.nextInt();
                    queue.enqueue(itemEnqueue);
                    break;
                case 2:
                    int itemDequeue = queue.dequeue();
                    System.out.printf("deqeued item > %d\n", itemDequeue);
                    break;
                case 3:
                    int itemPeek = queue.peek();
                    System.out.printf("peeked item > %d\n", itemPeek);
                    break;
                case 4:
                    queue.display();
                    break;
                case 5:
                    displayMenu();
                    break;
                case 6:
                    System.out.println("exiting");
                    break;
                default:
                    System.out.println("invalid choicde");

            }
        } while (choice != 6);
        sc.close();
    }

    public static void displayMenu() {
        System.out.println("enter your choice:");
        System.out.println("1. enqueue");
        System.out.println("2. dequeue");
        System.out.println("3. peek");
        System.out.println("4. print queue");
        System.out.println("5. display menu");
        System.out.println("6. exit");
    }
}
```