Program : 4a

```
import java.util.LinkedList;

import java.util.Queue;


class StackUsingTwoQueues {

    Queue<Integer> q1 = new LinkedList<>();

    Queue<Integer> q2 = new LinkedList<>();

    void push(int x) {

        q2.add(x);

        while (!q1.isEmpty()) {

            q2.add(q1.poll());

        }

        Queue<Integer> temp = q1;

        q1 = q2;

        q2 = temp;

    }   int pop() {

        if (q1.isEmpty()) {

            System.out.println("Stack is Empty");

            return -1;

        }

        return q1.poll();

    }   int peek() {

        if (q1.isEmpty()) {

            System.out.println("Stack is Empty");

            return -1;

        }

        return q1.peek();

    }

    boolean isEmpty() {
        return q1.isEmpty();
    }

    public static void main(String[] args) {
        StackUsingTwoQueues stack = new StackUsingTwoQueues();
        stack.push(10);
        stack.push(20);
        stack.push(30);

        System.out.println("Top element: " + stack.peek());
        System.out.println("Popped: " + stack.pop());
        System.out.println("Top after pop: " + stack.peek());
        stack.push(40);
        System.out.println("Top after pushing 40: " + stack.peek());
    }
}
```

Ouput:

```
Top element: 30
Popped: 30
Top after pop: 20
Top after pushing 40: 40
```

Program :4b

```
import java.util.ArrayList;

class BagOfNumbers {
    private ArrayList<Integer> bag;

    public BagOfNumbers() {
        bag = new ArrayList<>();
    }

    public void add(int num) {
        bag.add(num);
    }

    public boolean remove(int num) {
        return bag.remove(Integer.valueOf(num));
    }

    public int size() {
        return bag.size();
    }

    public boolean isEmpty() {
        return bag.isEmpty();
    }

    public boolean contains(int num) {
        return bag.contains(num);
    }
    public void display() {
        System.out.println("Bag elements: " + bag);
    }

    public static void main(String[] args) {
        BagOfNumbers bag = new BagOfNumbers();
        bag.add(10);
        bag.add(20);
        bag.add(20);
        bag.add(30);

        bag.display();

        System.out.println("Removing 20: " + bag.remove(20));
        bag.display();

        System.out.println("Bag contains 30? " + bag.contains(30));
        System.out.println("Size of bag: " + bag.size());
        System.out.println("Is bag empty? " + bag.isEmpty());
    }
}
```

Output:

```
Bag elements: [10, 20, 20, 30]
Removing 20: true
```

Program :4c

```
import java.util.Scanner;

class DiskTower {
    static void arrangeDisks(int[] disks, int n) {
        int max = n;
        System.out.println("\nDisk Placement Process:");
        for (int i = 0; i < n; i++) {
            System.out.print("Day " + (i + 1) + " -> ");
            if (disks[i] == max) {
                while (max >= 1) {
                    boolean found = false;
                    for (int j = 0; j <= i; j++) {
                        if (disks[j] == max) {
                            System.out.print(max + " ");
                            max--;
                            found = true;
                            break;
                        }
                    }
                    if (!found) break;
                }
            }
            System.out.println();
        }
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter number of disks: ");
        int n = sc.nextInt();
        int[] disks = new int[n];

        System.out.println("Enter disk sizes (one per day):");
        for (int i = 0; i < n; i++) {
            disks[i] = sc.nextInt();
        }

        arrangeDisks(disks, n);
    }
}
```

Output:

```
Enter number of disks: 5
Enter disk sizes (one per day):
4
5
1
2
3

Disk Placement Process:
Day 1 ->
Day 2 -> 5 4
Day 3 ->
Day 4 ->
Day 5 -> 3 2 1
```