Q1. Implement a Map in java which takes the input and print the list in sorted order based on key.

Input: 5- Rahul, 7 Lakshman, 1 Ram, 4 Krrish, 2 Lakshay

Output: {1=Ram, 2=Lakshay, 4=Krrish, 5=Rahul, 7=lakshman}

Program:

```
package Map;

import java.util.Scanner;
import java.util.TreeMap;

public class MapAss1 {
   public static void main(String[] args) {
      Scanner sc = new Scanner(System.in);
      System.out.println("Enter the number of elements of key-Pair: ");
      int n = sc.nextInt();

      TreeMap<Integer, String> tm = new TreeMap<>();

      for(int i = 0; i<n; i++){
         System.out.print("Enter the key "+i+" : ");
         int key = sc.nextInt();
         System.out.print("Enter the value "+i+" : ");
         String val = sc.next();
         tm.put(key, val);
      }

      System.out.println("The value of the map is: "+tm);
   }
}
```

Approach:

We would simply use TreeMap in java and put numbers as the key.

Q2. Implement a Map in java which takes the input and print the list in sorted order based on value.

Input: 5- Rahul, 7 Lakshman, 1 Ram, 4 Krrish, 2 Lakshay

Output: {Krish=4, Lakshay=2, Lakshman=7, Rahul=5, Ram=1}

Approach: We would simply use TreeMap in Java and will put Strings as key

Program:

```java
package Map;

import java.util.Scanner;
import java.util.TreeMap;

public class MapAss2 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the size for enter the key values: ");
        int n = sc.nextInt();

        // We would simply use TreeMap in Java and will put Strings as
key.
        TreeMap<String, Integer> treeMap = new TreeMap<>();

        for(int i = 0; i<n; i++){

            System.out.print("key: ");
            int key = sc.nextInt();

            System.out.print("Value: ");
            String val = sc.next();

            treeMap.put(val, key);
        }

        System.out.println(treeMap);
    }
}
```

Q3. Detect if an Array contains a duplicate element. At Most 1 duplicate would be there.

Input: 1,2,3,4

Output: No

```java
package Map;

import java.util.HashMap;
import java.util.Scanner;

public class MapAss3 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the number of elements of array: ");

        int n = sc.nextInt();

        int[] arr = new int[n];

        System.out.print("Enter the elements of array: ");
        for(int i=0;i<n;i++){
            arr[i] = sc.nextInt();
        }
```

```
        HashMap<Integer, Integer> tm = new HashMap<>();
        int answer = 0;
        for(int i=0;i<n;i++){
            if(tm.containsKey(arr[i])){
                System.out.println("Yes");
                answer =1;
                break;
            }
            tm.put(arr[i], 1);
        }

        if(answer == 0)
            System.out.println("No");

    }
}
```

Approach:

1. We would start traversing the array.
2. As me move ahead, we would keep adding the element in map.
3. If we found any element is already added in the map that means we have found our duplicate. 4. If no element is found then there is no duplicate.

Q4. Given an array nums of size n, return the majority element.

Input: 4,2,7,1,9

Output: 9

```
package Map;

import java.util.Scanner;
import java.util.TreeMap;

public class MapAss4 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the number of elements of array: ");

        int n = sc.nextInt();

        int[] arr = new int[n];

        System.out.print("Enter the elements of array: ");
        for(int i=0;i<n;i++){
            arr[i] = sc.nextInt();
        }

        TreeMap<Integer, Integer> tm = new TreeMap<>();
```

```
        for(int i=0;i<n;i++){
            tm.put(arr[i], 1);
        }

        System.out.println("Largest Element of Map is : " +
tm.lastEntry().getKey());
    }
}
```

Approach:

1. We would start traversing the array and will store each element into the TreeMap.

2. We would simply return the last key of map

Q5. Given two strings ransomNote and magazine, return true if ransomNote can be constructed by using the letters from magazine and false otherwise. Each letter in magazine can only be used once in ransomNote.

Input: ransomNote = "a", magazine = "b"

Output: false Input: ransomNote = "aa", magazine = "ab"

```
package Map;

import java.util.HashMap;
import java.util.Scanner;

public class MapAss5 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the RansomNote String: ");

        String s=sc.nextLine();

        System.out.print("Enter the Magazine String: ");

        String r=sc.nextLine();


        HashMap<Character, Integer> hm1 = new HashMap<>();

        for(int i=0;i<s.length(); i++ ){
            if(hm1.containsKey(s.charAt(i))) {
                hm1.put(s.charAt(i), hm1.get(s.charAt(i))+1);
            }
            else
                hm1.put(s.charAt(i), 1);
        }

        HashMap<Character, Integer> hm2 = new HashMap<>();
```

```
        for(int i=0;i<r.length(); i++ ){
            if(hm2.containsKey(r.charAt(i))) {
                hm2.put(r.charAt(i), hm2.get(r.charAt(i))+1);
            }
            else
                hm2.put(r.charAt(i), 1);
        }



        Integer answer = -1;
        for(Map.Entry<Character, Integer> e: hm1.entrySet()) {

            if(e.getValue() > hm2.get(e.getKey())){

                System.out.println("false");
                answer = 1;
                break;
            }
        }
        if(answer == -1)
            System.out.println("True");

    }
}
```

Approach:

1. We would store both words in two different maps one by one and update each character frequency.

2. Then we would start iterating over ransomNote map and would check that for each key in ransomNote map,the same key should be present in magazineMap and value > ransomNote map's value.

3. If we find any element not following this condition we would return false.

4. If we iterate the map completely the answer is Yes.