**PSNA College of Engineering and Technology Department of CSE**

**Date: 19.8.2023**

**Stable and Unstable Numbers:**

Five numbers are available with the kids.

These numbers are either stable or unstable.

A number is stable if each digit occur the same number of times.i.e, the frequency of each digit in the number is the same. For e.g. 2277,4004,11,23,583835,1010 are examples for stable numbers.

Similarly, a number is unstable if the frequency of each digit in the number is NOT same. For e.g. 221,4314,101,233,58135,101 are examples of unstable numbers.

The output can be:

**Output=Maximum of all stable numbers + Minimum of all stable numbers**

Assuming that the five numbers are passed to a function as input1, input2, input3, input4, and input5, complete the function to find and return the output.

For example:

If input1=12, input2=1313, input3=122, input4=678, and input5=898,

Stable numbers are 12,1313 and 678

Unstable numbers are 122, and 898

So the output should be = Maximum of all stable numbers + Minimum of all stable numbers

= 1313+12=1325

Case a): output = Maximum of all stable numbers + Minimum of all stable numbers

Case b): output = Minimum of all stable numbers + Maximum of all unstable numbers

**Program:**

import java.util.\*;

public class Main

{

  public static void main(String[] args)

  {

    Scanner sc=new Scanner(System.in);

    //**getting the number of elements**

    int n=sc.nextInt();

    //**creating two lists to store the ordered and unordered lists**

    List<Integer>stable=new ArrayList<>();

    List<Integer>unstable=new ArrayList<>();

    for(int i=0;i<n;i++)

    {

      //**getting the number**

      int num=sc.nextInt();

      int tmp=num;

      //**defining a map to store the occurences of each digit in a number and a set to check**

      Set<Integer>set=new HashSet<>();

      Map<Integer,Integer>map=new HashMap<>();

      //**traversing all the numbers**

      while(tmp>0)

      {

        int rem=tmp%10;

        tmp/=10;

        map.put(rem,map.getOrDefault(rem,0)+1);

      }

      set.addAll(map.values());

      if(set.size()==1)

        stable.add(num);

      else

        unstable.add(num);

    }

    System.out.println("Maximum of all stable numbers + Minimum of all stable numbers = "+(Collections.max(stable)+Collections.min(stable)));

    System.out.println("Minimum of all stable numbers + Maximum of all unstable numbers = "+(Collections.min(stable)+Collections.max(unstable)));

  }

}

**Sample I/O:**

**Input:**

5

12

122

678

898

1313

**Output:**

Maximum of all stable numbers + Minimum of all stable numbers = 1325

Minimum of all stable numbers + Maximum of all unstable numbers = 910