Started on	Saturday, 18 November 2023, 8:08 AM
State	Finished
Completed on	Saturday, 18 November 2023, 8:46 AM
Time taken	38 mins 21 secs
Marks	3.00/3.00
Grade	**15.00** out of 15.00 (**100**%)
Name	BALAJI S CSD

Question **1**

Correct

Mark 1.00 out of 1.00

Given a string s containing just the characters '(', ')', '{', '}', '[' and ']', determine if the input string is valid. An input string is valid if:

1. Open brackets must be closed by the same type of brackets.
2. Open brackets must be closed in the correct order.
3. Every close bracket has a corresponding open bracket of the same type.

Constraints:

- 1 <= s.length <= 10^4
- s consists of parentheses only '()[]{}'.

For example:

Input	Result
()	true
()[]	true

Answer: (penalty regime: 0 %)

```
import java.util.*;

public class Main{
    public static boolean checkBalanceString(String s){
        Stack<Character>stack = new Stack<Character>();
        int size =s.length();
        for(int i=0;i<size;i++){
            char c=s.charAt(i);
            if("({[".indexOf(c)>=0){
                stack.push(c);
            }
            else if("}])".indexOf(c)>=0){
                if(stack.isEmpty()){
                    return false;
                }
                char tmp = stack.pop();
                if("()-{}-[]".indexOf(""+tmp+c)<0){
                    return false;
                }
            }
        }
        return stack.isEmpty();
    }
    public static void main(String[]args){
        Scanner scan =new Scanner (System.in);
        String s=scan.nextLine();
        System.out.println(checkBalanceString(s));

    }
}
```

	Input	Expected	Got	
✔	()	true	true	✔
✔	()[]	true	true	✔
✔	(]	false	false	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

Question **2**

Correct

Mark 1.00 out of 1.00

Complete program to count frequency of each element of an array. Frequency of a particular element will be printed once.

Sample Test Cases

Test Case 1

Input

7
23
45
23
56
45
23
40

Output

23 occurs 3 times
45 occurs 2 times
56 occurs 1 times
40 occurs 1 times

Answer: (penalty regime: 0 %)

```
import java.util.*;
import java.util.Collections;
public class Daa
{
    public static void main(String args[])
    {
        Scanner sc=new Scanner(System.in);
        int n=sc.nextInt();
        ArrayList<Integer>list=new ArrayList<>();
        while(n!=0)
        {
            list.add(sc.nextInt());
            n--;
        }
        ArrayList<Integer> temp=new ArrayList<>();
        for(int i:list)
        {
            if(temp.contains(i))
               continue;
            else
            {
                temp.add(i);
                System.out.println(i+" occurs "+ Collections.frequency(list,i)+ " times ");
            }
        }
    }
}
```

	Input	Expected	Got	
✔	7 23 45 23 56 45 23 40	23 occurs 3 times 45 occurs 2 times 56 occurs 1 times 40 occurs 1 times	23 occurs 3 times 45 occurs 2 times 56 occurs 1 times 40 occurs 1 times	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

Question **3**

Correct

Mark 1.00 out of 1.00

Given a number, you are expected to find its two-digit "Reduced Subtracted Form (RSF)".

The "Reduced Subtracted Form (RSF)" of a number can be found by concatenating the difference between its adjacent digits.

To find the two-digit "Reduced Subtracted Form (RSF)", we need to continue this process till the resultant RSF is not a two digit.

For eg. If the number is 6928, its RSF can be found by concatenating the difference between (6 and 9), (9 and 2) and (2 and 8) as shown below:

difference between 6 and 9 is 3

difference between 9 and 2 is 7

difference between 2 and 8 is 6

So, the "Reduced Subtracted Form (RSF)" of 6928 = 376.

The resultant RSF (376) is not a two-digit, so we must continue finding its "Reduced Subtracted Form (RSF)".

difference between 3 and 7 is 4

difference between 7 and 6 is 1

So, the "Reduced Subtracted Form (RSF)" of 376 = 41.

The resultant number (41) is a two-digit, so we have reached the "two-digit Reduced Subtracted Form"

Therefore, the two-digit RSF of 6928 = 41.

Note1: input1 will always be >= 100.

Note2: Note that while concatenating the differences, we are expected to use the absolute values (non-negative).

Note3: The input values for all test cases in this program have been designed such that their two-digit RSF will definitely result a two-digit number.

Answer: (penalty regime: 0 %)

Reset answer

```
static String findTwoRSF(int input1) {
    String x=input1+"";
    char ch[]=x.toCharArray();
    String temp="";
    for (int i=0;i<ch.length-1;i++)
    {
        int diff=ch[i]-ch[i+1];
        diff=Math.abs(diff);
        temp+=diff;
    }
    String mass="";
    do
    {
        for(int i=0;i<temp.length()-1;i++)
        {
            int d=temp.charAt(i)-temp.charAt(i+1);
            d=Math.abs(d);
            mass=mass+d;
        }
    }
    while(mass.length()>2);
    return mass;
}

```

	Test	Expected	Got	
✔	findTwoRSF(5721)	34	34	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.