IRC_JAVA_CODING CONTEST 3

Test Summary

- No. of Sections: 1
- No. of Questions: 5
- Total Duration: 60 min

Section 1 - Coding

Section Summary

- No. of Questions: 5
- Duration: 60 min

Additional Instructions:

None

Q1. Write a program to validate an IP address(IPv4) with the help of Regular Expressions.

The IP address is a string in the form “A.B.C.D”, where the value of A, B, C, and D may range from 0 to 255. Leading zeros are allowed. The length of A, B, C, or D can’t be greater than 3.

Input Format

A string in the first line

Output Format

Valid IP address or not along with IP address as shown in the sample output.

Sample Input

```
000.12.12.034
```

Sample Output

```
IP address 000.12.12.034 is Valid
```

Sample Input

```
000.12.234.23.23
```

Sample Output

```
IP address 000.12.234.23.23 is Invalid
```

Sample Input

```
I.67.nt.3an
```

Sample Output

```
IP address I.67.nt.3an is Invalid
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q2. **Problem statement:**
Refer to the sample outputs and write a regular expression to represent the indian mobile numbers with +91 or 0.

Input Format

Mobile number input as string

Output Format

Print whether the mobile number is valid or invalid along with a mobile number.
Refer sample output.

Sample Input

```
+91-7123456789
```

Sample Output

```
+91-7123456789 : Valid Number
```

Sample Input

```
08123456789
```

Sample Output

```
08123456789 : Valid Number
```

Sample Input

```
9876543210
```

Sample Output

```
9876543210 : Valid Number
```

Sample Input

```
02123456789
```

Sample Output

```
02123456789 : Invalid Number
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q3. Given a string, the task is to check if the string contains any alphabet ‘g’ followed by one or more ‘e’s in it. If present then display the string otherwise, print No match.

Input Format

String input in the first line

Output Format

Display the matched output or No match.

Sample Input

```
archana
```

Sample Output

```
No match
```

Sample Input

```
higeeram
```

Sample Output

```
higeeram
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q4. **Problem statement:**
Write a Java program to compute the difference between two dates (Hours, minutes, seconds).ie., get two date inputs from the user and calculate the difference.

Input Format

The input consists of ten integers- year, month, date, min, hrs of the first and second date.

Output Format

The output prints the difference between the two dates. Refer to the sample input and output for the formatting specifications.

Sample Input

```
2016
9
6
0
```

Sample Output

```
Difference is 1 Hours, 60 Minutes, 3600 Seconds
```

Sample Input

```
2020
4
21
0
```

Sample Output

```
Difference is 8760 Hours, 525600 Minutes, 31536000 Secon
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q5. **Problem statement:**
Write a java program to add 1 week to the date, add 1 month to the date, add 1 year to the date , and add 10 years to the given date.

Input Format

Input consists of 3 integers representing the year, month, and date

Output Format

The output prints the expected information. Refer to the sample input and output for formatting specifications.

Sample Input

```
2022
3
27
```

Sample Output

```
Given date : 2022-03-27
Next week : 2022-04-03
Next month : 2022-04-27
Next year : 2023-03-27
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Answer Key & Solution

Section 1 - Coding

Q1

Test Case

Input

```
000.12.12.255
```

Output

```
IP address 000.12.12.255 is Valid
```

Weightage - 25

Input

```
000.12.12.257
```

Output

```
IP address 000.12.12.257 is Invalid
```

Weightage - 25

Input

```
000.12.12
```

Output

```
IP address 000.12.12 is Invalid
```

Weightage - 25

Input

```
126.1.1.02
```

Output

```
IP address 126.1.1.02 is Valid
```

Weightage - 25

Sample Input

```
000.12.12.034
```

Sample Output

```
IP address 000.12.12.034 is Valid
```

Sample Input

```
000.12.234.23.23
```

Sample Output

```
IP address 000.12.234.23.23 is Invalid
```

Sample Input

```
I.67.nt.3an
```

Sample Output

```
IP address I.67.nt.3an is Invalid
```

Solution

```
 import java.util.regex.*;
import java.util.*;
  class IPAddressValidation {
    public static boolean isValidIPAddress(String ip)
    {


        String zeroTo255
            = "(\\d{1,2}|(0|1)\\"
              + "d{2}|2[0-4]\\d|25[0-5])";

        String regex
            = zeroTo255 + "\\."
              + zeroTo255 + "\\."
              + zeroTo255 + "\\."
              + zeroTo255;


        Pattern p = Pattern.compile(regex);


        if (ip == null) {
            return false;
        }

        Matcher m = p.matcher(ip);

        return m.matches();
    }


    public static void main(String args[])
    {
        Scanner sc = new Scanner(System.in);

        String str1 = sc.nextLine();
        System.out.print("IP address " + str1 +" is ");
        if(isValidIPAddress(str1))
        System.out.print("Valid");
        else
        System.out.print("Invalid");



    }
}
```

Q2 **Test Case**

Input

```
+917123456789
```

Output

```
+917123456789 : Valid Number
```

Weightage - 25

Input

```
+91- 9876543210
```

Output

```
+91- 9876543210 : Valid Number
```

Weightage - 25

Input

Output

```
+9198765410
```

```
+9198765410 : Invalid Number
```

Weightage - 25

Input

```
+9876543210
```

Output

```
+9876543210 : Invalid Number
```

Weightage - 25

Sample Input

```
+91-7123456789
```

Sample Output

```
+91-7123456789 : Valid Number
```

Sample Input

```
08123456789
```

Sample Output

```
08123456789 : Valid Number
```

Sample Input

```
9876543210
```

Sample Output

```
9876543210 : Valid Number
```

Sample Input

```
02123456789
```

Sample Output

```
02123456789 : Invalid Number
```

Solution

```
 import java.util.regex.*;
import java.util.*;
class Main
{
    public static void main(String[] args)
    {
        String indiaRegex = "^(?:(?:\\+|0{0,2})91(\\s*[\\-]\\s*)?|[0]?)?[789]\\d{9}$";
        Pattern p = Pattern.compile(indiaRegex);
         Scanner  s = new  Scanner(System.in);
         String   str;
         str= s.nextLine();
         Matcher m=p.matcher(str);
         if(m.matches())
         {
            System.out.println(str+" : "+"Valid Number");
         }
         else
         {
            System.out.println(str+" : "+"Invalid Number");
         }
    }
}
//Pattern p=Pattern.compile("[7-9][0-9]{10}");
```

Q3

Test Case

Input

```
anitha
```

Output

```
No match
```

Weightage - 25

Input

```
whether
```

Output

```
No match
```

Weightage - 25

Input

```
higeeks
```

Output

```
higeeks
```

Weightage - 25

Input

```
regexregex
```

Output

```
regexregex
```

Weightage - 25

Sample Input

```
archana
```

Sample Output

```
No match
```

Sample Input

```
higeeram
```

Sample Output

```
higeeram
```

Solution

```
 import re
l=[]
t=1
for i in range(t):
    def check():
        string =input()
        regex = re.compile("ge+\w*")
        match_object = regex.findall(string)
        if len(match_object) != 0:
            for word in match_object:
                l.append(string)
        else :
            l.append("No match")
        check()

    for i in l:
        print(i)
```

Q4

Test Case

Input

```
2016
9
6
0
```

Output

```
Difference is 0 Hours, 0 Minutes, 0 Seconds
```

Weightage - 25

Input

```
2021
4
21
0
```

Output

```
Difference is 8760 Hours, 525600 Minutes, 31536000 S
```

Weightage - 25

Input

```
1996
8
28
0
```

Output

```
Difference is 744 Hours, 44640 Minutes, 2678400 Seco
```

Weightage - 25

Input

```
1992
12
11
5
```

Output

```
Difference is 0 Hours, 3 Minutes, 180 Seconds
```

Weightage - 25

Sample Input

```
2016
9
6
0
```

Sample Output

```
Difference is 1 Hours, 60 Minutes, 3600 Seconds
```

Sample Input

```
2020
4
21
0
```

Sample Output

```
Difference is 8760 Hours, 525600 Minutes, 31536000 S
```

Solution

```
 import java.time.*;
import java.util.*;

class Main
{
   public static void main(String[] args)
    {
        Scanner  s= new Scanner(System.in);
        int year=s.nextInt();
        int month=s.nextInt();
        int date=s.nextInt();
        int min=s.nextInt();
        int hrs=s.nextInt();
        int year1=s.nextInt();
        int month1=s.nextInt();
        int date1=s.nextInt();
        int min1=s.nextInt();
        int hrs1=s.nextInt();
        LocalDateTime dateTime = LocalDateTime.of(year, month, date, min, hrs);
        LocalDateTime dateTime2 = LocalDateTime.of(year1, month1, date1, min1, hrs1);
        // LocalDateTime dateTime2 = LocalDateTime.now();
       // int diffInNano = java.time.Duration.between(dateTime, dateTime2).getNano();
        long diffInSeconds = java.time.Duration.between(dateTime, dateTime2).getSeconds();
        //long diffInMilli = java.time.Duration.between(dateTime, dateTime2).toMillis();
        long diffInMinutes = java.time.Duration.between(dateTime, dateTime2).toMinutes();
        long diffInHours = java.time.Duration.between(dateTime, dateTime2).toHours();
        System.out.printf("Difference is %d Hours, %d Minutes, %d Seconds",  diffInHours, diffInMinutes,  diffInSeconds );

  }
}
```

Q5

Test Case

Input

```
2021
2
14
```

Output

```
Given date : 2021-02-14
Next week : 2021-02-21
Next month : 2021-03-14
Next year : 2022-02-14
```

Weightage - 25

Input

```
2000
2
16
```

Output

```
Given date : 2000-02-16
Next week : 2000-02-23
Next month : 2000-03-16
Next year : 2001-02-16
```

Weightage - 25

Input

```
1983
5
8
```

Output

```
Given date : 1983-05-08
Next week : 1983-05-15
Next month : 1983-06-08
Next year : 1984-05-08
```

Weightage - 25

Input

```
1919
9
12
```

Output

```
Given date : 1919-09-12
Next week : 1919-09-19
Next month : 1919-10-12
Next year : 1920-09-12
```

Weightage - 25

Sample Input

```
2022
3
27
```

Sample Output

```
Given date : 2022-03-27
Next week : 2022-04-03
Next month : 2022-04-27
Next year : 2023-03-27
```

Solution

```
 import java.time.LocalDate;
import java.time.temporal.ChronoUnit;
import java.util.*;

class Main
{

   public static void main(String args[])
   {
       Main m = new Main();
        m.testChromoUnits();
   }

   public void testChromoUnits()
   {
      //Get the current date
        Scanner  s= new Scanner(System.in);
        int year=s.nextInt();
        int month=s.nextInt();
        int date=s.nextInt();
        // int min=s.nextInt();
        // int hrs=s.nextInt();

        // LocalDateTime dateTime = LocalDateTime.of(year, month, date, min, hrs);
      LocalDate today = LocalDate.of(year, month, date);
      System.out.println("Given date : " + today);

      //add 1 week to the current date
      LocalDate nextWeek = today.plus(1, ChronoUnit.WEEKS);
      System.out.println("Next week : " + nextWeek);

      //add 1 month to the current date
      LocalDate nextMonth = today.plus(1, ChronoUnit.MONTHS);
      System.out.println("Next month : " + nextMonth);

      //add 1 year to the current date
      LocalDate nextYear = today.plus(1, ChronoUnit.YEARS);
      System.out.println("Next year : " + nextYear);

      //add 10 years to the current date
      LocalDate nextDecade = today.plus(1, ChronoUnit.DECADES);
      System.out.println("Date after ten years : " + nextDecade);
   }
}
```