IRC_JAVA_OOP_CC_1_Slot_1

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. **A Multiplication Game**

John and Michael play the game of multiplication by multiplying an integer **p** by one of the numbers 2 to 9. John always starts with **p = 1** and multiply it by 1, and passes the result to Michael. Then, Michael multiplies the number by 2 and sends the result to John, then John multiplies by 3 and so on. Before a game starts, they draw an integer **N** and the winner is the one who first reaches ***p ≥ n***.
Create a class that has two functions:
1) A function to perform the multiplication operation
2) The main()

Input Format

The input consists of the value of n.

Output Format

The output prints the n value and who won the game separated by a space.
Refer the sample output for formatting specifications.

Sample Input

```
10
```

Sample Output

```
10 Michael wins
```

Sample Input

```
100
```

Sample Output

```
100 John wins
```

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

Q2. **BO Classes**

We can use a BO class for computational purposes.
The Stall owners wanted to calculate the total cost of a particular ItemType for the given timeline. So add a feature in the application to calculate the total cost for the given timeline.
Create a class **ItemType** with the following attributes,

Attribute	Data Type
name	String
deposit	Double
costPerDay	Double

Add appropriate getter/setter, default and parameterized constructor.
public ItemType(String name, Double deposit, Double costPerDay).
Get the start date and end date (manipulate as Date object) from the stall owners to calculate rent for the particular ItemType.
Write a method **calculateCost** in **ItemTypeBO** class.

Method	Method Description
public Double calculateCost(Date start,Date end,ItemType typeIns)	returns a Double which corresponds to the total cost.

Create a driver class Main to test the above classes.
Note: Strictly adhere to the Object-Oriented Specifications given in the problem statement.
All class names, attribute names and method names should be the same as specified in the problem statement.
Display only 1 digit after decimal while displaying cost.
Input date format is **dd/MM/yyyy.**

Input Format

First line of the input consists of a string
Second and third line of the input consists of double.
Fourth and fifth line consists of starting date and the ending date.

Output Format

Refer sample output.

Sample Input

```
Morsh
1000.00
50.00
12/10/2018
```

Sample Output

```
Morsh
1000.0
50.0
100.0
```

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

Q3. Develop a class TelephoneIndex with two String objects as members. One should hold people's names and the other should hold their phone number. The class should have appropriate constructor, input, and display methods. Create an array of objects for TelephoneIndex and do the following:
a. Your program should ask the user to enter a name or the first few characters of a name to search for it in the array.
b. The program should display all of the names that match the user's input and their corresponding phone numbers.

Input Format

First-line has the number of records N in the Telephone Index. Following N*2 lines has the name and phone number one below the other as shown in The sample test case. The last line has the name(substring) to be found.

Output Format

The output displays the details of the matching records shown in the sample test case.

Sample Input

```
6
james
45464
jim
```

Sample Output

```
jim 66987
jill 454
```

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

Q4. Write a program to move all the special characters to the end of the string.
Create a class named **Demo** with the following method.
static String move(String str)
Create a **Main** class that extends **Demo** and call the method within it.

Input Format

Input to get the string S.

Output Format

Display the string after moving all the special characters to the end.

Sample Input

```
tenn34***hikm##!m9
```

Sample Output

```
tenn34hikmm9***##!
```

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

Q5. Write a java program to rotate a matrix n times in clockwise and anti-clockwise direction using inheritance.
Create a base class Clock.
From this extends a child class Clockwise and Anticlockwise.
Both these child classes should contain the method rotate().

Input Format

The first line contains the number of rows
The second line contains the number of columns
The following lines contain the matrix as input rows and columns with space as separator between each item in the matrix
Then the following line contains the number of rotations to be performed

Refer Sample input

Output Format

Print the matrix after performing the n number of clockwise and anticlockwise rotations

Refer sample output

Constraints

0 < Row <=10
0 < Column <=10
0 < Number Of Rotations <=10

Sample Input

```
3
3
1 2 3
4 5 6
```

Sample Output

```
Clockwise
4 1 2
7 5 3
8 9 6
```

Sample Input

```
4
4
1 2 3 4
5 6 7 8
```

Sample Output

```
Clockwise
13 9 5 1
14 7 11 2
15 6 10 3
```

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

Answer Key & Solution

Section 1 - Coding

Q1

Test Case

Input

```
3000
```

Output

```
3000 John wins
```

Weightage - 20

Input

```
5550
```

Output

```
5550 Michael wins
```

Weightage - 20

Input

```
40500
```

Output

```
40500 John wins
```

Weightage - 20

Input

```
750
```

Output

```
750 John wins
```

Weightage - 20

Input

```
200
```

Output

```
200 Michael wins
```

Weightage - 20

Sample Input

```
10
```

Sample Output

```
10 Michael wins
```

Sample Input

```
100
```

Sample Output

```
100 John wins
```

Solution

```
import java.io.*;
import java.util.*;
class multiplicationGame {
    public static void game(int n) {
        int sum=1,i=2,count=1;
        while(sum<n && i<=9) {
                sum *= i;
                if(sum <n) {
                i++;
                count++;
                }
                else {
                    break;
                }
    }
    if(count%2 !=0) {
        System.out.println(n+" Michael wins");
    }
    else {
        System.out.println(n+" John wins");
    }
    }
    public static void main (String [] args) {
        int n;
        Scanner sc = new Scanner(System.in);
        n= sc.nextInt();
        game(n);
    }
}
```

Q2

Test Case

Input

```
Morsh
1000.00
50.00
12/10/2018
```

Output

```
Morsh
1000.0
50.0
100.0
```

Weightage - 20

Input

```
Ankit
2000.00
35.00
08/08/2008
```

Output

```
Ankit
2000.0
35.0
350.0
```

Weightage - 20

Input

```
Sharma
8000.00
60.00
22/11/1997
```

Output

```
Sharma
8000.0
60.0
240.0
```

Weightage - 20

Input

Output

```
Williams
25000.00
70.00
01/01/2001
```

```
Williams
25000.0
70.0
2100.0
```

Weightage - 20

Input

Output

```
Lora
50000.00
80.00
14/02/2004
```

```
Lora
50000.0
80.0
560.0
```

Weightage - 20

Sample Input

Sample Output

```
Morsh
1000.00
50.00
12/10/2018
```

```
Morsh
1000.0
50.0
100.0
```

Solution

```
import java.io.*;
import java.text.DecimalFormat;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.*;
class ItemType {
    public String name;
    public double deposit;
    public double costPerDay;
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public double getDeposit() {
        return deposit;
    }
    public void setDeposit(double deposit) {
        this.deposit = deposit;
    }
    public double getCostPerDay() {
        return costPerDay;
    }
    public void setCostPerDay(double costPerDay) {
        this.costPerDay = costPerDay;
    }
    public ItemType()
    {
        this.name=null;
        this.deposit=0;
        this.costPerDay=0;
    }
    public ItemType(String name, Double deposit, Double costPerDay){
        this.name=name;
        this.deposit=deposit;
        this.costPerDay=costPerDay;
        System.out.println(this.name);
        System.out.println(this.deposit);
        System.out.println(this.costPerDay);
```

```
    }
}
class ItemTypeBO {
    public Double calculateCost(Date start,Date end,ItemType typeIns){
        long diff = (start.getTime()-end.getTime())/86400000;
        double result = diff*typeIns.costPerDay;
        return result;
    }
}
class Main {
    public static void main(String [] args) throws ParseException {
ItemType i = new ItemType();
Scanner sc = new Scanner(System.in);
DecimalFormat dd = new DecimalFormat("0.0");
i.name = sc.nextLine();
i.deposit = Double.parseDouble(sc.nextLine());
i.costPerDay = Double.parseDouble(sc.nextLine());
String date1 = sc.nextLine();
String date2 = sc.nextLine();
ItemType i1 = new ItemType(i.name,i.deposit,i.costPerDay);
Date start = new SimpleDateFormat("dd/MM/yyyy").parse(date1);
Date end = new SimpleDateFormat("dd/MM/yyyy").parse(date2);
ItemTypeBO iBO = new ItemTypeBO();
double result=iBO.calculateCost(start, end, i1);
System.out.println(dd.format(Math.abs(result)));


    }
}
```

Q3

Test Case

Input

```
10
king
787987
pinky
```

Output

```
pinky 987545
paul 897
plare 545565465
```

Weightage - 20

Input

```
5
plas
87987
plkokjd
```

Output

```
lfree 97879
```

Weightage - 20

Input

```
15
qwrqeqw
897879
pnottyty
```

Output

```
werwre 97865465
werwerwe 4654654
```

Weightage - 20

Input

```
7
ertet
78979745
fghfh
```

Output

```
qwrr 987654
qrerw 9787
```

Weightage - 20

Input

```
3
qwrew
8978979
qwnwn
```

Output

```
werwr 897987
```

Weightage - 20

Sample Input

```
6
james
45464
jim
```

Sample Output

```
jim 66987
jill 454
```

Solution

```
import java.util.Scanner;
class TelephoneIndex{
    String name, phone;

    TelephoneIndex(){
    }
    void getData(String Cname, String pno)
    {
       // System.out.println("set data");
        this.name = Cname;
        this.phone = pno;
    }
    void display(String Cname, String pno)
    {

        System.out.println(name + " " + phone);

    }
    void findData(String cname){
        if(name.startsWith(cname))
        {
                display(name,phone);
        }

    }

}
class Main{
    public static void main(String args[]){
        Scanner in = new Scanner(System.in);
        int N = in.nextInt();
        in.nextLine();
        TelephoneIndex[] ti = new TelephoneIndex[N];

        String contactName, phoneNum;
        for(int i =0;i<N;i++)
        {
            //System.out.println("get contactName");
            contactName = in.nextLine();
            //System.out.println("get phoneNum");
            phoneNum = in.nextLine();
            ti[i] = new TelephoneIndex();
            ti[i].getData(contactName, phoneNum);
            //System.out.println("output name phone" + ti[i].name + " " + ti[i].phone);
```

```
        }
        String findName = in.nextLine();
        for(int i =0;i<N;i++){
           // t = new TelephoneIndex();
            ti[i].findData(findName);
        }

    }
}
```

Q4

Test Case

Input

```
nu43/8()$#res
```

Output

```
nu438res/()$#
```

Weightage - 20

Input

```
#dear@67!ima
```

Output

```
dear67ima#@!
```

Weightage - 20

Input

```
jump*in#!2well
```

Output

```
jumpin2well*#!
```

Weightage - 20

Input

```
en(dure)
```

Output

```
endure()
```

Weightage - 20

Input

```
#inter#cal
```

Output

```
intercal##
```

Weightage - 20

Sample Input

```
tenn34***hikm##!m9
```

Sample Output

```
tenn34hikmm9***##!
```

Solution

```
import java.util.Scanner;
class Demo{
  static String move(String str)
    {
        int len = str.length();
       String regx = "[a-zA-Z0-9\\s+]";
       String res1 = "", res2 = "";
        for (int i = 0; i < len; i++) {
          char c = str.charAt(i);
        if (String.valueOf(c).matches(regx))
               res1 = res1 + c;
            else
               res2 = res2 + c;
        }
        return res1 + res2;
    } }
    class Main extends Demo{
  public static void main(String args[])
    {
        String str;
        Scanner in=new Scanner(System.in);
        str=in.nextLine();
        System.out.println(move(str));
    }
}
```

Q5

Test Case

Input

```
2
2
1 2
3 4
```

Output

```
Clockwise
4 3
2 1
Anti clockwise
```

Weightage - 10

Input

```
1
1
5
4
```

Output

```
Clockwise
5
Anti clockwise
5
```

Weightage - 10

Input

```
5
5
1 2 3 4 5
6 7 8 9 0
```

Output

```
Clockwise
12 11 6 1 2
10 14 13 22 3
20 15 33 7 4
```

Weightage - 20

Input

```
7
7
1 2 3 4 5 6 7
```

Output

```
Clockwise
23 12 11 1 2 3 4
33 34 24 13 22 33 5
```

Weightage - 20

Input

```
3
3
1 4 5
8 9 0
```

Output

```
Clockwise
0 7 2
5 9 3
4 1 8
```

Weightage - 20

Input

```
8
8
1 2 3 4 5 6 7 8
9 10 11 12 13 14 15 16
```

Output

```
Clockwise
41 33 25 17 9 1 2 3
49 50 42 34 26 18 10 4
57 51 45 44 43 35 11 5
```

Weightage - 20

Sample Input

```
3
3
1 2 3
4 5 6
```

Sample Output

```
Clockwise
4 1 2
7 5 3
8 9 6
```

Sample Input

```
4
4
1 2 3 4
5 6 7 8
```

Sample Output

```
Clockwise
13 9 5 1
14 7 11 2
15 6 10 3
```

Solution

```
import java.util.*;
class Main
{
    public static void main(String[] args) {
        Scanner s=new Scanner(System.in);
        int row=s.nextInt();
        int col=s.nextInt();
int [][]a=new int[row][col];
for(int i=0;i<row;i++)
{
    for(int j=0;j<col;j++)
    {
        a[i][j]=s.nextInt();
    }
}
int n=s.nextInt();
Clock cl=new Clock();
Clock cc=new Clockwise();
Clock ac=new Anticlockwise();
    int [][]b=new int[row][col];
for(int i=0;i<row;i++)
{
    for(int j=0;j<col;j++)
    {
        b[i][j]=a[i][j];
    }
}
Anticlockwise acw=new Anticlockwise();
for(int k=0;k<n;k++)
```

```
 cc.rotate(row,col,a,n);
 System.out.println("Clockwise ");
  for (int i = 0; i < row; i++)
    {
        for (int j = 0; j < col; j++)
        System.out.print( a[i][j] + " ");
        System.out.print("\n");
    }
 System.out.println("Anti clockwise ");
int r=row;
int c=col;
int k=n;
int f,K;
int l = 0;
int m = 0;
int Row = r-1;
int Col = c-1;
while(l < Row && m < Col)
{
       int rot = 2*(Row-l)+2*(Col-m);
       f = n%rot;
       for(int i=1;i<=f;i++)
          {
                ac.rotate(row,col,b,n);

                acw.rotation(l,m,Row,Col,b);
          }
          l++;
          m++;
          Row--;
          Col--;
      }
          for (int i = 0; i < row; i++)
          {
              for (int j = 0; j < col; j++)
              System.out.print( b[i][j] + " ");
              System.out.print("\n");
          }



    }
}
class Clock
{
    public Clock()
    {

    }
    void rotate(int row,int col,int a[][],int num)
    {

    }
}
class Clockwise extends Clock
{
    static int R;
   static int C;

    void rotate(int m,int n,int mat[][],int num)
    {
        R = m;
        C = n;

        int row = 0, col = 0;
```

```
int prev, curr;
while (row < m && col < n)
{
    if (row + 1 == m || col + 1 == n)
        break;
    prev = mat[row + 1][col];
    for (int i = col; i < n; i++)
    {
        curr = mat[row][i];
        mat[row][i] = prev;
        prev = curr;
    }
    row++;
    for (int i = row; i < m; i++)
    {
        curr = mat[i][n-1];
        mat[i][n-1] = prev;
        prev = curr;
    }
    n--;
    if (row < m)
    {
        for (int i = n-1; i >= col; i--)
        {
            curr = mat[m-1][i];
            mat[m-1][i] = prev;
            prev = curr;
        }
    }
    m--;
    if (col < n)
    {
        for (int i = m-1; i >= row; i--)
        {
            curr = mat[i][col];
            mat[i][col] = prev;
            prev = curr;
        }
    }
    col++;
  }


}
}
class Anticlockwise extends Clock
{
     void rotate(int m,int n,int mat[][],int num)
   {}
   void rotation(int l, int m, int Row, int Col,int mat[][])
{
       int si,sj,i,j,t,f;
       si = l;
       sj = m;
       t = mat[l][m];

       for(i=l+1;i<=Row;i++)
       {
              f = mat[i][m];
              mat[i][m] = t;
              t = f;
       }
       l++;
       for(i=m+1;i<=Col;i++)
       {
```

```
        f = mat[Row][i];
        mat[Row][i] = t;
        t = f;
}
m++;
if(l-1 < Row)
{
        for(i=Row-1;i>=l-1;i--)
        {
                f = mat[i][Col];
                mat[i][Col] = t;
                t = f;
        }
}
Col--;
if(m-1 < Col)
{
        for(i=Col;i>=m;i--)
        {
                f = mat[l-1][i];
                mat[l-1][i] = t;
                t = f;
        }
        }
        Row--;
        mat[si][sj] = t;
        return;
}

}
```