IRC_SKCT_Java2_COD_Classes&Objects

Test Summary

- No. of Sections: 1
- No. of Questions: 15
- Total Duration: 120 min

Section 1 - Coding

Section Summary

- No. of Questions: 15
- Duration: 120 min

Additional Instructions:

None

Q1. Java program to find the count of all digits of a number using class.
In this program, we will read a positive integer number and then calculate the count of all digits using a class.

Input Format

The input consists of a number.

Output Format

The output prints the count of all digits in the number.

Sample Input

```
12345
```

Sample Output

```
Count of all digits: 5
```

Sample Input

```
22
```

Sample Output

```
Count of all digits: 2
```

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

Q2. Write a program to find the number of occurrences of a character in a string. Create a constructor with two parameters, pass the value from the main method to the constructor, and perform the mentioned task in the constructor and display it.

Input Format

Input to get a string in the first line and a character in the second line.

Output Format

Output the number of occurrences of a character in a string.

Sample Input

```
utter
t
```

Sample Output

```
2
```

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

Q3. Write a program to check whether the given character is vowel or consonant.
Create two methods namely main method and alph. Create an object in the main method and access the alph method, that performs the above operation.

Input Format

Input to get a character.

Output Format

The output prints whether the character is a vowel or consonant.
Display the output as shown in the sample output.

Sample Input

```
j
```

Sample Output

```
j :consonant
```

Sample Input

```
e
```

Sample Output

```
e :vowel
```

Sample Input

```
5
```

Sample Output

```
5 :consonant
```

Time Limit: 100 ms Memory Limit: 256 kb Code Size: 1024 kb

Q4. Create class money with two attributes:
int rupee
int paisa
Include getters, setters, and constructors.
Create the main class and initialize the values for the data members
Get two amounts and print their sum.

Input Format

The input consists of two amounts.
Rupee and Paisa are separated by a space.

Output Format

The output prints the total sum.
Refer sample input and output for formatting specifications.

Sample Input

```
50 85
42 65
```

Sample Output

```
93.50
```

Sample Input

```
254 45
845 20
```

Sample Output

```
1099.65
```

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

Q5. **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 $p \geq n$.
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

Sample Output

10	10 Michael wins

Sample Input **Sample Output**

100	100 John wins

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

Q6. **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 startDate, endItemType 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 **Sample Output**

Morsh 1000.00 50.00 12/10/2018	Morsh 1000.0 50.0 100.0

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

Q7. Sunrise Basket founder has decided to organize a fun event at your college. The event coordinator has announced a coding contest for creating the application for the Contest. The Best application would be used for the fair and the developer gets a cash prize. You are a well-versed and aspiring Programmer in your college. Many programmers have enrolled themselves for the contest and you are one of them. Every contestant is provided with a Schema diagram of the Fair. Get yourself acquainted with Schema and brace yourself for the challenge!!!.

As a part of this, the Application requires a user prompt to create a new Item type. Hence create an **ItemType** class with the following private attributes.

1. **name** (String)
2. **deposit**(double)
3. **costPerDay**(double)

Include appropriate Getters and Setters for the class and also include a method "**void display()**" to display the output shown in the sample output.
The main class is implemented already to get input from the user and display. Write the suitable code complete ItemType class.

Input Format

Name of Item in the first line.
Deposit in the second line.
Cost per day in the third line.

Output Format

Display the details as shown in the sample output

Sample Input

```
Fan
5000
300
```

Sample Output

```
Name : Fan
Deposit Amount : 5000.0
Cost per day : 300.0
```

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

Q8. Create a class named **Address** with the following member variables and methods

1. street as String
2. city as String
3. pincode as integer
4. country as String
5. displayAddress() to display all the details.

Create a main class named **AddressMain** to include the Main method.

In the main method, obtain the details of the Address by creating an object for the Address class and assign the values to the attributes. Call the method displayAddress() in the Main class to display the values.

Note:
Use the same class names, attribute names, and method names
Implement suitable getters and setters

Input Format

The first line of the input contains the street name
The second line of the input contains the city name
The third line of the input contains Pincode
The fourth line of the input contains the country name

Output Format

Print the street name in the first line
Print the city name second line
Print the Pincode in the third line
Print the country name in the fourth line

Sample Input

```
13,Rockfort Street
Chennai
654035
India
```

Sample Output

```
Street: 13,Rockfort Street
City: Chennai
Pincode: 654035
Country: India
```

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

Q9. Write a program to display the day of a week.
Note: Create a constructor and perform the above task,the object in main method should pass the value to the constructor.

Input Format

Input to get an integer N.

Output Format

Display the output as shown in the sample output.

Constraints

N ≤ 7

Sample Input

```
7
```

Sample Output

```
Saturday
```

Sample Input

```
0
```

Sample Output

```
Weekend
```

Sample Input

```
9
```

Sample Output

```
Invalid
```

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

Q10. Create a class with two methods one to read the elements of an array and the other to find all pairs of elements in an array whose sum is equal to a specified number.

Input Format

The first line of the input consists of the value of n.
Next input is the array elements.
The last input is the sum value.

Output Format

The output prints the pair whose sum is equal to a specified number.

Sample Input

```
5
1 2 3 4 5
8
```

Sample Output

```
3 5
4 4
5 3
```

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

Q11. Create two classes a Box class and a Main class, create an object for the Box class in the Main class and calculate the volume of box.

Input Format

Input to get width,height and depth separated by single space.

Output Format

Display the volume of the box.
If inputs <= 0 then print "Invalid".

Constraints

Inputs (double type).

Sample Input

```
7.2 8.0 1.1
```

Sample Output

```
63.36000000000001
```

Sample Input

```
2.2 1.1 3
```

Sample Output

```
7.260000000000002
```

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

Q12. A group of ‘n’ candidates have applied for faculty recruitment. Their Name, qualification, experience and gender are to be stored in a class “Recruitment”. Write a program to sort the objects based on their experience and display their details.

Input Format

First line specifies the number of employees "n"

In the following lines Name, qualification, gender and experience of the faculty will be given for "n" employees

Output Format

Print the details of the faculty in the sorted order of their experience

Sample Input

Sample Output

```
2
ram
Be cse
male
```

```
pravin
Be ece
3
nam
```

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

Q13. **Big Bash Event**

The fair has an event called Big Bash event. It is conducted to increase the business of the stalls. It gives a discount on the particular bills and the constraint is not told to the audience attending the fair. Create a program to check whether a bill is eligible for the BigBash event or not. The eligibility is calculated on the basis of the purchased date. If month in the purchased date is even, then the bill is eligible for the event. If the purchased month is odd, then it is not eligible for the event. If the bill is eligible for the event, then the discount is given. The discount percentage should be the purchased month number.
Example:
If purchased date is --> 12-10-2017 [dd/MM/yyyy format]
The purchased month is 10, so Peter is eligible for the event and discount of 10% should be given to the user.
If the purchased amount is 100, then the discount amount is 10. So, the total amount is 100-10= 90.
If purchased date is --> 12-01-2018 [dd/MM/yyyy format]
The purchased month is 01, so Peter is not eligible for the event.
Create a class **Event** with the following methods,

Method Name	Description
static int checkEventAvailable(Date date)	This method takes the date of purchase and check for the month. If the month is even it should return the date value, else return 0.
static Double getAmountWithDiscount(Double amount, Date date)	This method takes the amount of purchase and the date of purchase as parameters and calculate the final amount after discount and return the discounted amount as Double.

Create a driver class **Main** to test the above class.
[Note: Strictly adhere to the object oriented specifications given as a part of the problem statement.Use the same class names, attribute names and method names]

Input Format

The first line of the input is the purchased date.
The second line of input is an Integer which corresponds to the purchase amount.

Output Format

The output consists of discounted amount if he is eligible for the event, else display "Not Eligible for BIGBASH event".
Refer sample output for formatting specifications.

Sample Input

```
12/12/2017
100
```

Sample Output

```
88.0
```

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

Q14. Create a class **NumberConverter** with required methods to convert between four major number systems (Decimal, Binary, Octal, and Hexadecimal).

Create a **Main** class and call a suitable method using **NumberConverter** object. Get the source and destination number system as a single character from the user along with the number in the main class. Call a suitable method in **NumberConverter** class to convert**.**

Note : **D** for Decimal, **B** for Binary, **O** for Octal, and **H** for Hexadecimal.

Input Format

Number System Code(From)
Number System Code(To)
Number

Output Format

Print the result after conversion

Constraints

Only 4 codes for Number system

Sample Input

```
D
B
23
```

Sample Output

```
10111
```

Sample Input

```
H
O
27
```

Sample Output

```
47
```

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

Q15. 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

Answer Key & Solution

Section 1 - Coding

Q1

Test Case

Input

```
2147483647
```

Output

```
Count of all digits: 10
```

Weightage - 15

Input

```
222222888
```

Output

```
Count of all digits: 9
```

Weightage - 15

Input

```
77777777
```

Output

```
Count of all digits: 8
```

Weightage - 15

Input

```
1234567
```

Output

```
Count of all digits: 7
```

Weightage - 15

Input

```
654321
```

Output

```
Count of all digits: 6
```

Weightage - 15

Input

```
3214
```

Output

```
Count of all digits: 4
```

Weightage - 15

Input

```
12345
```

Output

```
Count of all digits: 5
```

Weightage - 10

Sample Input

```
12345
```

Sample Output

```
Count of all digits: 5
```

Sample Input

```
22
```

Sample Output

```
Count of all digits: 2
```

Solution

```
import java.util.*;
class Main {
    public static void main(String[] args) {
        int count = 0, num;
        Scanner s = new Scanner(System.in);
        num = s.nextInt();
        while(num != 0)
        {
            num /= 10;
            ++count;
        }
        System.out.println("Count of all digits: " + count);
    }
}
```

Q2

Test Case

Input

```
nuts
s
```

Output

```
1
```

Weightage - 20

Input

```
range
u
```

Output

```
0
```

Weightage - 30

Input

```
kettle
e
```

Output

```
2
```

Weightage - 20

Input

```
runner
n
```

Output

```
2
```

Weightage - 20

Input

```
meets
e
```

Output

```
2
```

Weightage - 10

Sample Input

```
utter
t
```

Sample Output

```
2
```

Solution

```
 import java.util.Scanner;
class Main{
Main(String s,char c){
int res = 0;
for (int i=0; i<s.length(); i++) {
if (s.charAt(i) == c)
res++;
}
 System.out.print(res);

}
public static void main(String args[])
{

String str;

Scanner in=new Scanner(System.in);

str=in.nextLine();

char c;

c=in.next().charAt(0);

Main obj=new Main(str,c);

}
}
```

Q3

Test Case

Input

```
k
```

Output

```
k :consonant
```

Weightage - 15

Input

```
o
```

Output

```
o :vowel
```

Weightage - 15

Input

```
l
```

Output

```
l :consonant
```

Weightage - 15

Input

```
a
```

Output

```
a :vowel
```

Weightage - 15

Input

```
7
```

Output

```
7 :consonant
```

Weightage - 25

Input

```
q
```

Output

```
q :consonant
```

Weightage - 15

Sample Input

```
j
```

Sample Output

```
j :consonant
```

Sample Input

```
e
```

Sample Output

```
e :vowel
```

Sample Input

```
5
```

Sample Output

```
5 :consonant
```

Solution

```
import java.util.Scanner;
class Main {
    void alph(char ch){
if(ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u' )
            System.out.println(ch + " :vowel");
        else
            System.out.println(ch + " :consonant");
}

public static void main(String[] args) {
        char ch ;
        Scanner in=new Scanner(System.in);
        ch=in.next().charAt(0);
        Main obj=new Main();
        obj.alph(ch);
    }
}
```

Q4

Test Case

Input

```
854  96
2486  96
```

Output

```
3341.92
```

Weightage - 25

Input

```
8642  25
6428  60
```

Output

```
15070.85
```

Weightage - 25

Input

```
753  65
854  80
```

Output

```
1608.45
```

Weightage - 25

Input

```
8564  25
24687  20
```

Output

```
33251.45
```

Weightage - 25

Sample Input

```
50  85
42  65
```

Sample Output

```
93.50
```

Sample Input

```
254 45
845 20
```

Sample Output

```
1099.65
```

Solution

```
import java.io.*;
import java.util.*;
class money {
    int rupee;
    int paisa;
    public void setRupee(int r) {
        this.rupee = r;
    }
    public void setPaisa(int p) {
        this.paisa = p;
    }
    public int getRupee() {
        return rupee;
    }
    public int getPaisa() {
        return paisa;
    }
}
class Main {
    public static void main(String [] args) {
        Scanner sc = new Scanner(System.in);
        money [] m = new money[2];
        int i;
        for(i=0;i<2;i++) {
            m[i] = new money();
            m[i].setRupee(sc.nextInt());
            m[i].setPaisa(sc.nextInt());
        }
        int r,p;
        r = m[0].getRupee()+m[1].getRupee();
        p = m[0].getPaisa()+m[1].getPaisa();
        if(p>99) {
            r +=1;
            p = p-100;
        }
        System.out.println(r+"."+p);
    }
}
```

Q5

Test Case

Input

```
3000
```

Output

```
3000 John wins
```

Weightage - 20

Input

Output

5550	5550 Michael wins

Weightage - 20

Input	Output
40500	40500 John wins

Weightage - 20

Input	Output
750	750 John wins

Weightage - 20

Input	Output
200	200 Michael wins

Weightage - 20

Sample Input	Sample Output
10	10 Michael wins

Sample Input	Sample Output
100	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);
    }
}
```

Q6

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

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

Output

```
Williams
25000.0
70.0
2100.0
```

Weightage - 20

Input

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

Output

```
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)));


    }
}
```

Q7

Test Case

Input

```
Car
5000
250
```

Output

```
Name : Car
Deposit Amount : 5000.0
Cost per day : 250.0
```

Weightage - 20

Input

```
Light
0
0
```

Output

```
Name : Light
Deposit Amount : 0.0
Cost per day : 0.0
```

Weightage - 20

Input

```
Banu
45000
334
```

Output

```
Name : Banu
Deposit Amount : 45000.0
Cost per day : 334.0
```

Weightage - 20

Input

```
ASDFGHJKLQWERTY
3456789
9876543
```

Output

```
Name : ASDFGHJKLQWERTY
Deposit Amount : 3456789.0
Cost per day : 9876543.0
```

Weightage - 20

Input

```
1234567
456789
3456
```

Output

```
Name : 1234567
Deposit Amount : 456789.0
Cost per day : 3456.0
```

Weightage - 20

Sample Input

```
Fan
5000
300
```

Sample Output

```
Name : Fan
Deposit Amount : 5000.0
Cost per day : 300.0
```

Solution

Header

```
import java.util.*;



class ItemType{
    String name;
    double deposit;
    double costPerDay;

    public void setName(String name){
        this.name=name;
    }

    public void setDeposit(double deposit){
        this.deposit=deposit;
    }

    public void setCostPerDay(double costPerDay){
        this.costPerDay=costPerDay;
    }

    public void display(){
        System.out.println("Name : "+this.name);
        System.out.println("Deposit Amount : "+this.deposit);
        System.out.println("Cost per day : "+this.costPerDay);
    }

}
```

Footer

```
class Main{
    public static void main(String args[]){

        Scanner sc=new Scanner(System.in);
        ItemType obj1= new ItemType();
        String name = sc.nextLine();
        double deposit =sc.nextDouble();
        double costPerDay =sc.nextDouble();

        obj1.setName(name);
        obj1.setDeposit(deposit);
        obj1.setCostPerDay(costPerDay);
        obj1.display();
    }
}
```

Q8

Test Case

Input	Output
Main road Coimbatore 638401 India	Street: Main road City: Coimbatore Pincode: 638401 Country: India

Weightage - 20

Input

```
2nd Road
Goa
538401
India
```

Output

```
Street: 2nd Road
City: Goa
Pincode: 538401
Country: India
```

Weightage - 20

Input

```
5th Cut
Mad
5401
USA
```

Output

```
Street: 5th Cut
City: Mad
Pincode: 5401
Country: USA
```

Weightage - 20

Input

```
AAAAA
BBBBB
11111
CCCCC
```

Output

```
Street: AAAAA
City: BBBBB
Pincode: 11111
Country: CCCCC
```

Weightage - 20

Input

```
2nd street
Mumbai
123
India
```

Output

```
Street: 2nd street
City: Mumbai
Pincode: 123
Country: India
```

Weightage - 20

Sample Input

```
13,Rockfort Street
Chennai
654035
India
```

Sample Output

```
Street: 13,Rockfort Street
City: Chennai
Pincode: 654035
Country: India
```

Solution

Q9

Test Case

Input

```
2
```

Output

```
Monday
```

Weightage - 15

Input

```
4
```

Output

```
Wednesday
```

Weightage - 15

Input

```
6
```

Output

```
Friday
```

Weightage - 15

Input

```
3
```

Output

```
Tuesday
```

Weightage - 15

Input

```
1
```

Output

```
Sunday
```

Weightage - 15

Input

```
8
```

Output

```
Invalid
```

Weightage - 25

Sample Input

```
7
```

Sample Output

```
Saturday
```

Sample Input

```
0
```

Sample Output

```
Weekend
```

Sample Input

```
9
```

Sample Output

```
Invalid
```

Solution

```
import java.util.Scanner;
class Main
{
    Main(int day){
        if(day <= 7){
```

```
switch(day)
{
    case 1:
        System.out.print("Sunday");
        break;
    case 2:
        System.out.print("Monday");
        break;
    case 3:
        System.out.print("Tuesday");
        break;
    case 4:
        System.out.print("Wednesday");
        break;
    case 5:
        System.out.print("Thursday");
        break;
    case 6:
        System.out.print("Friday");
        break;
    case 7:
        System.out.print("Saturday");
        break;
       default:
            System.out.print("Weekend");
            break;
    }}
    else{
        System.out.print("Invalid");
    }

}

public static void main(String s[])
{
    int d;
    Scanner in=new Scanner(System.in);
    d=in.nextInt();
    Main obj=new Main(d);

}
}
```

Q10

Test Case

Input

```
10
12 23 45 56 78 89 14 25 36 47
72
```

Output

```
25 47
36 36
47 25
```

Weightage - 25

Input

```
12
10 20 54 78 36 59 30 40 55 60 88 70
115
```

Output

```
55 60
60 55
```

Weightage - 25

Input

```
5
123 456 789 147 258
603
```

Output

```
456 147
147 456
```

Weightage - 25

Input

```
8
10 20 30 40 50 60 70 80
110
```

Output

```
30 80
40 70
50 60
60 50
```

Weightage - 25

Sample Input

```
5
1 2 3 4 5
8
```

Sample Output

```
3 5
4 4
5 3
```

Solution

```
import java.io.*;
import java.util.*;
class main {
    public static  void printArray(int arr[],int n,int sum) {
        int i,j;
        for(i=0;i<n;i++) {
            for(j=0;j<n;j++) {
                if(arr[i]+arr[j] == sum) {
                    System.out.print(arr[i]+" "+arr[j]);
                    System.out.println();
                }
            }
        }
    }
    public static void main(String [] args) {
        int n,i;
        Scanner sc = new Scanner(System.in);
        n = sc.nextInt();
        int arr[] = new int[n];
        for(i=0;i<n;i++) {
            arr[i] = sc.nextInt();
        }
        int sum = sc.nextInt();
        printArray(arr,n,sum);
    }
}
```

Q11

Test Case

Input

```
3 2 4
```

Output

```
24.0
```

Weightage - 10

Input

```
7.7 8.8 5.3
```

Output

```
359.12800000000004
```

Weightage - 20

Input

```
3 4 5
```

Output

```
60.0
```

Weightage - 20

Input

```
12.1 20.2 17.4
```

Output

```
4252.907999999999
```

Weightage - 20

Input

```
-4.5 -2 -3
```

Output

```
Invalid
```

Weightage - 30

Sample Input

```
7.2 8.0 1.1
```

Sample Output

```
63.36000000000001
```

Sample Input

```
2.2 1.1 3
```

Sample Output

```
7.260000000000002
```

Solution

```
import java.util.Scanner;
class Box {
double width;
double height;
double depth;

}
class Main {
public static void main(String args[]) {
Box mybox = new Box();
double vol;
```

```
Scanner sc=new Scanner(System.in);
mybox.width=sc.nextDouble() ;
mybox.height=sc.nextDouble() ;
mybox.depth=sc.nextDouble() ;
if(mybox.width>0 && mybox.height>0 && mybox.depth>0){
vol = mybox.width * mybox.height * mybox.depth;
System.out.print(vol);
}else{
    System.out.print("Invalid");
}
}
}
```

Q12

Test Case

Input

```
3
ram
Be cse
male
```

Output

```
muzam
Be mechanical
5
pravin
```

Weightage - 10

Input

```
5
ram
Be cse
male
```

Output

```
surya
Be cse
5
ram
```

Weightage - 20

Input

```
7
ram
Be cse
male
```

Output

```
priya
Be It
7
surya
```

Weightage - 25

Input

```
10
ram
Be cse
male
```

Output

```
Imran
MCA
12
aadhi
```

Weightage - 30

Input

```
7
ram
Be cse
male
```

Output

```
ram
Be cse
3
priya
```

Weightage - 15

Sample Input

```
2
ram
```

Sample Output

```
pravin
Be ece
```

Be cse | 3

Solution

```
import java.io.*;
import java.util.*;


class Recruitment implements Comparable<Recruitment>
{
    public String name, qualification, gender;
    public int experiance;

    public int compareTo(Recruitment m)
    {
        return m.experiance - this.experiance;
    }


    public Recruitment(String nm, String qua, String gender, int exp)
    {
        this.name = nm;
        this.experiance = exp;
        this.qualification = qua;
        this.gender = gender;
    }


    public String getName() {
        return name;
    }
}


class Main
{
    public static void main(String[] args)
    {
        ArrayList<Recruitment> emp_list = new ArrayList<Recruitment>();
        Scanner in = new Scanner(System.in);

        int num_of_emp;
num_of_emp = in.nextInt();


for (int i=0;i<num_of_emp;i++) {
    int exp;
    String name, qua, gender;

    name = in.nextLine();
    qua = in.nextLine();
    gender = in.nextLine();
    in.nextLine();
    exp = in.nextInt();

    emp_list.add(new Recruitment(name, qua, gender, exp));
}

Collections.sort(emp_list);

for (Recruitment each: emp_list)
{
    System.out.print(each.name);
```

```
            System.out.println(each.qualification);
            System.out.println(each.gender);
            System.out.println(each.experiance);
        }
    }
}
```

Q13

Test Case

Input

```
10/11/2008
50.00
```

Output

```
Not Eligible for BIGBASH event
```

Weightage - 20

Input

```
08/06/2005
1000.00
```

Output

```
992.0
```

Weightage - 20

Input

```
02/07/1996
2500.00
```

Output

```
Not Eligible for BIGBASH event
```

Weightage - 20

Input

```
14/09/2000
150.00
```

Output

```
Not Eligible for BIGBASH event
```

Weightage - 20

Input

```
29/06/2006
750.00
```

Output

```
721.0
```

Weightage - 20

Sample Input

```
12/12/2017
100
```

Sample Output

```
88.0
```

Solution

```
 import java.io.*;
import java.text.SimpleDateFormat;
import java.text.DecimalFormat;
import java.text.ParseException;
import java.util.*;
 class Event {
    public static int checkEventAvailable(Date start){
        Calendar c= Calendar.getInstance();
        c.setTime(start);
        if((c.get(Calendar.MONTH)+1)%2 == 0)
        {
            return c.get(Calendar.DATE);
        }
        else {
            return 0;
        }
}
public double getAmountWithDiscount(double amount, int dis) {
    double result = amount-dis;
    return result;
    }
    }
class Main {
 public static void main(String[] args) throws ParseException{
     Scanner sc = new Scanner(System.in);
     DecimalFormat dd = new DecimalFormat("0.0");
     double amount;
     Calendar c= Calendar.getInstance();
        String date1 = sc.nextLine();
        amount = Double.parseDouble(sc.nextLine());
        Event e = new Event();
        Date start = new SimpleDateFormat("dd/MM/yyyy").parse(date1);
        c.setTime(start);
        int dis = e.checkEventAvailable(start);
        if(dis == c.get(Calendar.DATE)) {
            double finalAmount = e.getAmountWithDiscount(amount,dis);
            System.out.println(dd.format(finalAmount));
        }
        else if(dis == 0) {
            System.out.println("Not Eligible for BIGBASH event");
        }
 }
}
```

Q14

Test Case

Input	Output
D B 39	100111

Weightage - 10

Input	Output
D O 39	47

Weightage - 10

Input

```
D
H
39
```

Output

```
27
```

Weightage - 10

Input

```
B
O
111
```

Output

```
7
```

Weightage - 10

Input

```
B
H
1111
```

Output

```
f
```

Weightage - 10

Input

```
B
D
1111
```

Output

```
15
```

Weightage - 10

Input

```
H
O
76
```

Output

```
166
```

Weightage - 10

Input

```
H
B
20
```

Output

```
100000
```

Weightage - 10

Input

```
O
B
45
```

Output

```
100101
```

Weightage - 10

Input

```
O
D
56
```

Output

```
46
```

Weightage - 10

Sample Input

```
D
B
23
```

Sample Output

```
10111
```

Sample Input

```
H
O
27
```

Sample Output

```
47
```

Solution

```
import java.util.Scanner;

class NumberConverter {
    public String converter(String number,int sBase,int dBase)
    {
         return Integer.toString(Integer.parseInt(number, sBase),dBase);
    }



}

class Mainclass{
    public static void main (String args[]) {
        NumberConverter number = new NumberConverter();
        Scanner myObj =new Scanner(System.in);
        char sBase = myObj.nextLine().charAt(0);
        char dBase = myObj.nextLine().charAt(0);
        String input =myObj.nextLine();

if((sBase == 'B') || (sBase =='b'))
    sBase =2;
if((dBase == 'B') || (dBase =='b'))
    dBase =2;

if((sBase == 'D') || (sBase =='D'))
    sBase =10;
if((dBase == 'D') || (dBase =='D'))
    dBase =10;

if((sBase == 'O') || (sBase =='o'))
    sBase =8;
if((dBase == 'O') || (dBase =='o'))
    dBase =8;

if((sBase == 'H') || (sBase =='h'))
    sBase =16;
if((dBase == 'H') || (dBase =='h'))
    dBase =16;
```

```
        System.out.println(number.converter(input,sBase,dBase));
    }
}
```

Q15

Test Case

Input

```
10
king
787987
pinky
```

Output

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

Weightage - 20

Input

```
5
plas
87987
plkokid
```

Output

```
lfree 97879
```

Weightage - 20

Input

```
15
qwrqeqw
897879
nnettvtv
```

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);
        }

    }
}
```