1. Find the house numbers between which Noddy can build the largest house

In the city of Toy land, there are N houses. Noddy is looking for a piece of land in the city to build his house. He wants to buy the land where he can build the largest possible house. All the houses in the city lie in a straight line and all of them are given a house number and position of the house from the entry point in the city. Noddy wants to find the house numbers between which he can build the largest house.

Input: The input to the function/method consists of two arguments

- **numOf House, an integer representing the number of houses.**
- **houseList, a list where each element of the list is a list of integers representing the house number and its position respectively.**

Constraints

2 < numOfHouse < 106

1 <houseList[i][0] <numOfHouse

0 < houseList[i][1] < 106

0 < I < numOfHouse

Note: No two houses will have the same position. In case of multiple such answers, return the one with the least distance from the reference point Zero.

Example:
Input:
numOfHouse = 5
houseList = [[3, 7],[1, 9],[2, 0],[5, 15],[4, 30]]
Output: [4, 5]

Explanation:
The largest land area with size 15 is available between the 4 and 5 numbered houses. So the output contains these house number s in ascending order.

Number of houses: 5
Input: [3,7], [1,9], [2,0], [5,15], [4,30]

Step 1: Get the House_Numbers in an array

[0]	[1]	[2]	[3]	[4]
3	1	2	5	4

Step 2: Get the position_of_the_houses in an array

[0]	[1]	[2]	[3]	[4]
7	9	0	15	30

Step 3: Copy the position of the houses in another array so that the original array will not be altered

[0]	[1]	[2]	[3]	[4]
7	9	0	15	30

Step 4: Sort the copied array of position of the houses

[0]	[1]	[2]	[3]	[4]
0	7	9	15	30

Step 5: The difference between the adjacent elements is calculated. The two elements are selected based on the highest difference.

Here the two elements selected are 15 and 30 as the difference between them is maximum.

Step 6: The 2 selected elements' index is obtained for the position_of_the_houses array.

Here, '15' is in 3rd index and '30' is in 4th index.

Step 7: The two house number are obtained by getting the elements from the house_number array matching the indexes Here, '5' is in house_array[3] and '4' is in house_array[4].

The house numbers should be printed in ascending order.

```
import java.io.*;
import java.util.Scanner;
import java.util.*;
public class Main
{
public static void main (String[] args)
{
Scanner sc = new Scanner(System.in);
int num;
int temp;
int x1=0;
int x2=0;
int position1=0;
int position2=0;
int maxi=0;
System.out.println("Enter the number of
houses:");
num = sc.nextInt();
int house_number[] = new int[num];
int position[] = new int[num];
int copy_position[] = new int[num];
System.out.println("Enter the house number
and position of the house:");
for(int i=0; i<num; i++)
{
house_number[i] = sc.nextInt();
}
System.out.println("Input:");
```

```
for(int i=0; i<num; i++)
{
System.out.print("[" + house_number[i] + "," +
position[i] + "]");
}
System.out.println("");
for(int i=0; i<num; i++)
{
copy_position[i] = position[i];
}
Arrays.sort(copy_position);
for(int i=0; i<num-1;i++){
temp = copy_position[i+1] - copy_position[i];
if(temp>maxi)
{
maxi = temp;
x1 = copy_position[i];
x2 = copy_position[i+1];
}
}
for(int i=0; i<num; i++) //The elements position
are found in the main position array
{
if(x1 == position[i])
{
position1 = i;
}
else if (x2 == position[i])
{
```

```
position2 = i;
}
}
if(house_number[position1]>house_number[po
sition2]) //The house number is displayed which
is matched by the position obtained
{
System.out.println("Result: [" +
house_number[position2] + "," +
house_number[position1] + "]");
}
else
{
System.out.println("Result: [" +
house_number[position1] + ","+
house_number[position2] + "]");
}
}
}
```

OUTPUT

```
Enter the number of houses:5
Enter the house number and position of the
house:
3 7
1 9
2 0
5 15
4 30
```

Input:
[3,7] [1,9] [2,0] [5,15] [4,30]
Result:[4,5]
Problem Statement –

2. A chocolate factory is packing chocolates into the packets. The chocolate packets here represent an array of N number of integer values. The task is to find the empty packets(0) of chocolate and push it to the end of the conveyor belt(array).

Example 1 :

Input:

N=7

arr = [4 5 0 1 9 0 5].

Output:

4 5 1 9 5 0 0

There are 3 empty packets in the given set. These 3 empty packets represented as O should be pushed towards the end of the array

Input :

7 – Value of N

[4,5,0,1,0,0,5] – Element of arr[O] to arr[N-1],While input each element is separated by newline

Output:

4 5 1 9 5 0 0

Example 2:

Input:

6 — Value of N.

[6,0,1,8,0,2] – Element of arr[0] to arr[N-1], While input each element is separated by newline

Output:

6 1 8 2 0 0

```
import java.util.*;
class Main
{
    public static void main(String[] args)
    {
            Scanner sc=new Scanner(System.in);
            int n=sc.nextInt();
            int arr[]=new int[n];
            for(int i=0;i< n;i++)
                  arr[i]=sc.nextInt();
            int count=0;
            for(int i=0;i< n;i++) // 4 5 0 1 9 0 5
```

```
            if(arr[i]!=0) //arr[3]!=0
                arr[count++]=arr[i]; //arr[2]=1 count =2
        for(int i=count;i < n;i++)
            arr[i]=0;
        for(int i=0;i< n;i++)
            System.out.print(arr[i]+" ");
    }
}
```

Problem Statement –

3. Joseph is learning digital logic subject which will be for his next semester. He usually tries to solve unit assignment problems before the lecture. Today he got one tricky question. The problem statement is “A positive integer has been given as an input. Convert decimal value to binary representation. Toggle all bits of it after the most significant bit including the most significant bit. Print the positive integer value after toggling all bits”.

Constrains-

1<=N<=100

Example 1:

Input :

10 -> Integer

Output :

5 -> result- Integer

Explanation:

Binary representation of 10 is 1010. After toggling the bits(1010), will get 0101 which represents “5”. Hence output will print “5”.

```
import java.util.*;
class Main
{
  public static void main(String[] args)
  {
        Scanner sc=new Scanner(System.in);
        int no=sc.nextInt();
         String bin="";

         while(no!=0)
         {
               bin=(no&1)+bin;
               no=no>>1;
         }
          bin=bin.replaceAll("1","2");
          bin=bin.replaceAll("0","1");
          bin=bin.replaceAll("2","0");
          int res=Integer.parseInt(bin,2);
         System.out.println(res);
   }
```

}

4. Jack is always excited about sunday. It is favourite day, when he gets to play all day. And goes to cycling with his friends.

So every time when the months starts he counts the number of sundays he will get to enjoy. Considering the month can start with any day, be it Sunday, Monday…. Or so on.

Count the number of Sunday jack will get within n number of days.

Example 1: Input

mon-> input String denoting the start of the month.

13 -> input integer denoting the number of days from the start of the month.

Output :

2 -> number of days within 13 days.

Explanation:

The month start with mon(Monday). So the upcoming sunday will arrive in next 6 days. And then next Sunday in next 7 days and so on.

Now total number of days are 13. It means 6 days to first sunday and then remaining 7 days will end up in another sunday. Total 2 sundays may fall within 13 days.

```
import java.util.*;
class Main
{
public static void main(String[] args)
{
        Scanner sc=new Scanner(System.in);
        String str=sc.next(); //mon
        int n=sc.nextInt(); //13
String
arr[]={"mon","tue,","wed","thu","fri","sat","sun"};
        int i=0;
        for(i=0;i< arr.length;i++)
            if(arr[i].equals(str))
                break;
        int res=1;
        int rem=6-i; //rem = 6
        n=n-rem;    //n= 13-6 = 7
        if(n >0){
          res+=n/7;
        System.out.println(res);     }
else{ System.out.println("No sunday");}
} }
```

5. Airport security officials have confiscated several item of the passengers at the security check point. All the items have been dumped into a huge box (array). Each item possesses a certain amount of risk[0,1,2]. Here, the risk severity of the items represent an array[] of N number of integer values. The task here is to sort the items based on their levels of risk in the array. The risk values range from 0 to 2.

Example :

Input :

7 -> Value of N

[1,0,2,0,1,0,2]-> Element of arr[0] to arr[N-1], while input each element is separated by new line.

Output :

0 0 0 1 1 2 2 -> Element after sorting based on risk severity

Example 2:

input : 10 -> Value of N

[2,1,0,2,1,0,0,1,2,0] -> Element of arr[0] to arr[N-1], while input each element is separated by a new line.

Output :

0 0 0 0 1 1 1 2 2 2 ->Elements after sorting based on risk severity.

Explanation:

In the above example, the input is an array of size N consisting of only 0's, 1's and 2s. The output is a sorted array from 0 to 2 based on risk severity.

```
import java.util.*;
class Main
{
    public static void main(String[] args)
    {
        Scanner sc=new Scanner(System.in);
         int n=sc.nextInt();
         int arr[]=new int[n];
         for(int i=0;i< n;i++)
                arr[i]=sc.nextInt();
          int countZero=0,countOne=0,countTwo=0;
          for(int i=0;i< n;i++)
          {
                if(arr[i]==0)
                      countZero++;
                else if(arr[i]==1)
                      countOne++;
                else if(arr[i]==2)
                      countTwo++;
```

```
            }
            int j =0;
            while(countZero >0)
             {
                        arr[j++]=0;
                        countZero--;
             }
             while(countOne >0)
             {
                        arr[j++]=1;
                       countOne--;
              }

              while(countTwo >0)
              {
                       arr[j++]=2;
                       countTwo--;
               }

              for(int i=0;i < n;i++)
                    System.out.print(arr[i]+" ");
    }
}
```

Given an integer array Arr of size N the task is to find the count of elements whose value is greater than all of its prior elements.

Note : 1st element of the array should be considered in the count of the result.

For example,

N=5

Arr[]={7,4,8,2,9}

As 7 is the first element, it will consider in the result.

8 and 9 are also the elements that are greater than all of its previous elements.

Since total of 3 elements is present in the array that meets the condition.

Hence the output = 3.

Example 1:

Input

5 -> Value of N, represents size of Arr

7-> Value of Arr[0]

4 -> Value of Arr[1]

8-> Value of Arr[2]

2-> Value of Arr[3]

9-> Value of Arr[4]

Output :

3

Example 2:

5 -> Value of N, represents size of Arr

3 -> Value of Arr[0]

4 -> Value of Arr[1]

5 -> Value of Arr[2]

8 -> Value of Arr[3]

9 -> Value of Arr[4]

Output :

5

Constraints

1<=N<=20

1<=Arr[i]<=10000

```
import java.util.*;
class Main
```

```
{
  public static void main(String[] args)
  {
        Scanner sc=new Scanner(System.in);
        int n=sc.nextInt();
        int arr[]=new int[n];
        for(int i=0;i< n;i++)
              arr[i]=sc.nextInt();
        int max=Integer.MIN_VALUE;

        int count=0;
        for(int i=0;i< n;i++)
        {
            if(arr[i]>max)

            {
                  max=arr[i];
                  count++;
             }
        }
        System.out.println(count);
}
}
```

6. A supermarket maintains a pricing format for all its products. A value N is printed on each product. When the scanner reads the value N on the item, the product of all the digits in the value N is the price of the item. The task here is to design the

software such that given the code of any item N the product (multiplication) of all the digits of value should be computed(price).

Example 1:

Input :

5244 -> Value of N

Output :
160 -> Price

Explanation:

From the input above

Product of the digits 5,2,4,4

5*2*4*4= 160

Hence, output is 160.

```
import java.util.*;
class Main
{
    public static void main(String[] args)
    {
        Scanner sc=new Scanner(System.in);
        int n=sc.nextInt();
        int res=1;
        while(n>0)
```

```
        {
            res=res*(n%10);
            n=n/10;
         }
          System.out.println(res);
   }
}
```

7. A furnishing company is manufacturing a new collection of curtains. The curtains are of two colors aqua(a) and black (b). The curtains color is represented as a string(str) consisting of a's and b's of length N. Then, they are packed (substring) into L number of curtains in each box. The box with the maximum number of 'aqua' (a) color curtains is labeled. The task here is to find the number of 'aqua' color curtains in the labeled box.

Note :

If 'L' is not a multiple of N, the remaining number of curtains should be considered as a substring too. In simple words, after dividing the curtains in sets of 'L', any curtains left will be another set(refer example 1)

Example 1:

Input :

bbbaaababa -> Value of str

3 -> Value of L

Output:

3 -> Maximum number of a’s

Explanation:

From the input given above.

Dividing the string into sets of 3 characters each

Set 1: {b,b,b}

Set 2: {a,a,a}

Set 3: {b,a,b}

Set 4: {a} -> leftover characters also as taken as another set

Among all the sets, Set 2 has more number of a’s. The number of a’s in set 2 is 3.

Hence, the output is 3.

Example 2:

Input :

abbbaabbb -> Value of str

5 -> Value of L

Output:

2 -> Maximum number of a’s

Explanation:

From the input given above,

Dividing the string into sets of 5 characters each.

Set 1: {a,b,b,b,b}

Set 2: {a,a,b,b,b}

Among both the sets, set 2 has more number of a’s. The number of a’s in set 2 is 2.

Hence, the output is 2.

Constraints:

1<=L<=10

1<=N<=50

The input format for testing

The candidate has to write the code to accept two inputs separated by a new line.

First input- Accept string that contains character a and b only

Second input- Accept value for N(Positive integer number)

The output format for testing

The output should be a positive integer number of print the message(if any) given in the problem statement.(Check the output in Example 1, Example 2).

```
import java.util.*;
class Main
{
	public static void main(String[] args)
	{
		Scanner sc=new Scanner(System.in);
		String str=sc.next();
		int n=sc.nextInt();
		int max=0,count=0;
		for(int i=0;i< str.length();i++)
		{
			if(i%n==0)
			{
			   max=Math.max(count,max);
			   count=0;
			}
			if(str.charAt(i)=='a')
			    count++;
		 }
		 max=Math.max(count,max);
```

```
            System.out.println(max);
        }
}
```

8. An international round table conference will be held in india. Presidents from all over the world representing their respective countries will be attending the conference. The task is to find the possible number of ways(P) to make the N members sit around the circular table such that.

The president and prime minister of India will always sit next to each other.

Example 1:

Input :

4 -> Value of N(No. of members)

Output :

12 -> Possible ways of seating the members

Explanation:

2 members should always be next to each other.

So, 2 members can be in 2!ways

Rest of the members can be arranged in (4-1)! ways.(1 is subtracted because the previously

selected two members will be considered as single members now).

So total possible ways 4 members can be seated around the circular table 2*6= 12.

Hence, output is 12.

Example 2:

Input:

10 -> Value of N(No. of members)

Output :

725760 -> Possible ways of seating the members

Explanation:

2 members should always be next to each other.

So, 2 members can be in 2! ways

Rest of the members can be arranged in (10-1)! Ways. (1 is subtracted because the previously selected two members will be considered as a single member now).

So, total possible ways 10 members can be seated around a round table is

2*362880 = 725760 ways.

Hence, output is 725760.

The input format for testing

The candidate has to write the code to accept one input

First input – Accept value of number of N(Positive integer number)

The output format for testing

The output should be a positive integer number or print the message(if any) given in the problem statement(Check the output in example 1, example2)

Constraints :

2<=N<=50

```
import java.math.BigInteger;
import java.util.*;
class Main
{
      public static BigInteger fact(int number)
    {
        BigInteger res= BigInteger.ONE;
        for (int i = number; i > 0; i--)
            res = res.multiply(BigInteger.valueOf(i));
        return res;
    }
```

```
    public static void main(String[] args)
    {
        Scanner sc=new Scanner(System.in);
        int n=sc.nextInt();
        BigInteger res=fact(n-1);

System.out.println(res.multiply(BigInteger.valueOf(2
)));
    }
}
```

Problem Statement

9. An intelligence agency has received reports about some threats. The reports consist of numbers in a mysterious method. There is a number “N” and another number “R”. Those numbers are studied thoroughly and it is concluded that all digits of the number ‘N’ are summed up and this action is performed ‘R’ number of times. The resultant is also a single digit that is yet to be deciphered. The task here is to find the single-digit sum of the given number ‘N’ by repeating the action ‘R’ number of times.

If the value of ‘R’ is 0, print the output as ‘0’.

Example 1:

Input :

99 -> Value of N

3 -> Value of R

Output :

9 -> Possible ways to fill the cistern.

Explanation:

Here, the number N=99

1. Sum of the digits N: 9+9 = 18
2. Repeat step 2 ‘R’ times i.e. 3 tims
   (9+9)+(9+9)+(9+9) = 18+18+18 =54
3. Add digits of 54 as we need a single digit 5+4

Hence , the output is 9.

Example 2:

Input :

1234   -> Value of N

2   -> Value of R

Output :

2 -> Possible ways to fill the cistern

Explanation:

Here, the number N=1234

1. Sum of the digits of N: 1+2+3+4 =10
2. Repeat step 2 ‘R’ times i.e. 2 times
   (1+2+3+4)+(1+2+3+4)= 10+10=20
3. Add digits of 20 as we need a single digit.
   2+0=2

Hence, the output is 2.

Constraints:

0<N<=1000

0<=R<=50

The Input format for testing

The candidate has to write the code to accept 2 input(s)

First input- Accept value for N (positive integer number)

Second input: Accept value for R(Positive integer number)

The output format for testing

The output should be a positive integer number or print the message (if any) given in the problem statement. (Check the output in Example 1, Example 2).

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

```
class Main
{
	public static int sumOfDigits(int n)
	{
		int sum=0;
		while(n>0)
		{
			sum+=n%10;
			n=n/10;
		}
	   return sum;
	 }
	 public static void main(String []args)
	 {
		Scanner sc=new Scanner(System.in);
		int n=sc.nextInt();
		int r=sc.nextInt();
		if(r==0)
			System.out.println("0");
		else
		{
				int res=sumOfDigits(n)*r;
				int sum=0;
				while(true)
				{
					while(res>0)
				{
					sum=sum+res%10;
					res=res/10;
```

```
                    }
                    if((sum/10)==0)
                      break;
                    else
                      res=sum;
                }
                System.out.println(sum);
            }
    }}
```

Problem Statement

10. Particulate matters are the biggest contributors to Delhi pollution. The main reason behind the increase in the concentration of PMs include vehicle emission by applying Odd Even concept for all types of vehicles. The vehicles with the odd last digit in the registration number will be allowed on roads on odd dates and those with even last digit will on even dates.

Given an integer array a[], contains the last digit of the registration number of N vehicles traveling on date D(a positive integer). The task is to calculate the total fine collected by the traffic police department from the vehicles violating the rules.

Note : For violating the rule, vehicles would be fined as X Rs.

Example 1:

Input :

4 -> Value of N

{5,2,3,7} -> a[], Elements a[0] to a[N-1], during input each element is separated by a new line

12 -> Value of D, i.e. date

200 -> Value of x i.e. fine

Output :

600 -> total fine collected

Explanation:

Date D=12 means , only an even number of vehicles are allowed.

Find will be collected from 5,3 and 7 with an amount of 200 each.

Hence, the output = 600.

Example 2:

Input :

5 -> Value of N

{2,5,1,6,8} -> a[], elements a[0] to a[N-1], during input each element is separated by new line

3 -> Value of D i.e. date

300 -> Value of X i.e. fine

Output :

900 -> total fine collected

Explanation:

Date D=3 means only odd number vehicles with are allowed.

Find will be collected from 2,6 and 8 with an amount of 300 each.

Hence, the output = 900

Constraints:

- 0<N<=100
- 1<=a[i]<=9
- 1<=D <=30
- 100<=x<=5000

The input format for testing

The candidate has to write the code to accept 4 input(s).

First input – Accept for N(Positive integer) values (a[]), where each value is separated by a new line.

Third input – Accept value for D(Positive integer)

Fourth input – Accept value for X(Positive integer)

The output format for testing

The output should be a positive integer number (Check the output in Example 1, Example e) if no fine is collected then print ”0”.

```
import java.util.*;
class Main
{
   public static void main (String[]args)
   {
      Scanner sc = new Scanner (System.in);
      int n = sc.nextInt ();
      int arr[] = new int[n];
      for (int i = 0; i < n; i++)
          arr[i] = sc.nextInt ();
      int d = sc.nextInt ();
      int x = sc.nextInt ();
      int countEven = 0, countOdd = 0;
      for (int i = 0; i < n; i++)
      {
      if (arr[i] % 2 == 0)
          countEven++;
```

```
        else
            countOdd++;
        }
        if (d % 2 != 0)
        {
        if (countEven == 0)
            System.out.println ("0");
        else
            System.out.println (countEven * x);
        }
        else
        {
        if (countOdd == 0)
            System.out.println ("0");
        else
            System.out.println (countOdd * x);
        }
    }
}
```

11. **Problem Statement –** Write a program to calculate the fuel consumption of your truck.The program should ask the user to enter the quantity of diesel to fill up the tank and the distance covered till the tank goes dry.Calculate the fuel consumption and display it in the format (liters per 100 kilometers).

Convert the same result to the U.S. style of miles per gallon and display the result. If the quantity or distance is zero or negative display ” is an Invalid Input”.

[**Note:** The US approach of fuel consumption calculation (distance / fuel) is the inverse of the European approach (fuel / distance). Also note that 1 kilometer is 0.6214 miles, and 1 liter is 0.2642 gallons.]

The result should be with two decimal place.To get two decimal place refer the below-mentioned print statement :

float cost=670.23;

System.out.printf(“You need a sum of Rs.%.2f to cover the trip”,cost);

Sample Input 1:

- Enter the no of liters to fill the tank
  20
- Enter the distance covered
  150

Sample Output 1:

- Liters/100KM
  13.33

- Miles/gallons
  17.64

Explanation:

- For 150 KM fuel consumption is 20 liters,
- Then for 100 KM fuel consumption would be (20/150)*100=13.33,
- Distance is given in KM, we have to convert it to miles (150*0.6214)=93.21,
- Fuel consumption is given in liters, we have to convert it to gallons (20*0.2642)=5.284,
- Then find (miles/gallons)=(93.21/5.284)=17.64

Sample Input 2:

- Enter the no of liters to fill the tank
  -5

Sample Output 2:

- -5 is an Invalid Input

Sample Input 3:

- Enter the no of liters to fill the tank
  25

- Enter the distance covered
  -21

Sample Output 3:

- -21 is an Invalid Input

```
import java.util.*;
import java.text.*;
class Main{
public static void main (String[] args) {
DecimalFormat df2 =new DecimalFormat("0.00");
Scanner sc= new Scanner (System.in);

System.out.println("Enter the no of liters to fill the
tank");
int ltt =sc.nextInt();
double lt= (ltt*1.00);

if(ltt<1){
System.out.println(ltt+" is an Invalid Input");
System.exit(0);
}
System.out.println("Enter the distance covered");
int diss =sc.nextInt();
double dis= (diss*1.00);

if(diss<1){
System.out.println(diss+" is an Invalid Input");
System.exit(0);
}

double hundered = ((lt/dis)*100);
System.out.println("Liters/100KM");
System.out.println(df2.format(hundered));
```

```
double miles = (dis*0.6214);
double gallons =(lt*0.2642);
double mg = miles/gallons;
System.out.println("Miles/gallons");
System.out.println(df2.format(mg));
}
}
```

12. **Problem Statement –** Vohra went to a movie with his friends in a Wave theatre and during break time he bought pizzas, puffs and cool drinks. Consider the following prices :

- Rs.100/pizza
- Rs.20/puffs
- Rs.10/cooldrink

Generate a bill for What Vohra has bought.

Sample Input 1:

- Enter the no of pizzas bought:10
- Enter the no of puffs bought:12
- Enter the no of cool drinks bought:5

Sample Output 1:

Bill Details

- No of pizzas:10
- No of puffs:12

- No of cooldrinks:5
- Total price=1290

ENJOY THE SHOW!!!

```
import java.util.Scanner;
public class Main
{
    public static void main (String[]args)
    {
        int totalprice;
        Scanner sc = new Scanner (System.in);
System.out.print ("Enter the no of pizzas bought:");
int pizza = sc.nextInt ();

System.out.print ("Enter the no of puffs bought:");
int puffs = sc.nextInt ();

System.out.print ("Enter the no of cool drinks
bought:");
int coolDrinks = sc.nextInt ();

int pizzaa = Math.abs (pizza) * 100;
int puffss = Math.abs (puffs) * 20;
int coolDrinkss = Math.abs (coolDrinks) * 10;

System.out.println ("Bill Details");
System.out.println ("No of pizzas:" + pizza);
System.out.println ("No of puffs:" + puffs);
```

```
System.out.println ("No of cooldrinks:" +
coolDrinks);

totalprice = pizzaa + puffss + coolDrinkss;
System.out.println ("Total price=" + totalprice);
System.out.println ("ENJOY THE SHOW!!!");
}
}
```

13. **Problem Statement –** Ritik wants a magic board, which displays a character for a corresponding number for his science project. Help him to develop such an application.
For example when the digits 65,66,67,68 are entered, the alphabet ABCD are to be displayed.
[Assume the number of inputs should be always 4]

Sample Input 1:

- Enter the digits:
  65
  66
  67
  68

Sample Output 1:

65-A
66-B
67-C
68-D

Sample Input 2:

- Enter the digits:
  115
  116
  101
  112

Sample Output 2:

115-s
116-t
101-e
112-p

```
import java.util.Scanner;
public class Main
{
public static void main (String args[])
{
Scanner in = new Scanner (System.in);
System.out.println ("Enter the digits: ");

int a = in.nextInt ();
int b = in.nextInt ();
int c = in.nextInt ();
int d = in.nextInt ();

char q = (char) a;
char w = (char) b;
char e = (char) c;
```

```
char r = (char) d;

System.out.println ();
System.out.print (a);
System.out.println ("-" + q);

System.out.print (b);
System.out.println ("-" + w);

System.out.print (c);
System.out.println ("-" + e);

System.out.print (d);
System.out.println ("-" + r);
}
}
```

14. **Problem Statement –** FOE college wants to recognize the department which has succeeded in getting the maximum number of placements for this academic year. The departments that have participated in the recruitment drive are CSE,ECE, MECH. Help the college find the department getting maximum placements. Check for all the possible output given in the sample snapshot

Note : If any input is negative, the output should be “Input is Invalid”. If all department has equal

number of placements, the output should be “None of the department has got the highest placement”.

Sample Input 1:

- Enter the no of students placed in CSE:90
- Enter the no of students placed in ECE:45
- Enter the no of students placed in MECH:70

Sample Output 1:

- Highest placement

CSE

Sample Input 2:

- Enter the no of students placed in CSE:55
- Enter the no of students placed in ECE:85
- Enter the no of students placed in MECH:85

Sample Output 2:

- Highest placement

ECE

MECH

Sample Input 3:

- Enter the no of students placed in CSE:0
- Enter the no of students placed in ECE:0
- Enter the no of students placed in MECH:0

Sample Output 3:

- None of the department has got the highest placement

Sample Input 4:

- Enter the no of students placed in CSE:10
- Enter the no of students placed in ECE:-50
- Enter the no of students placed in MECH:40

Sample Output 4:

- Input is Invalid

```
import java.util.Scanner;
public class Main
{
    public static void main(String[] args)
    {
        // Initialize Scanner object
        Scanner sc = new Scanner(System.in);
        // Take user input
        System.out.print("Enter the no. of students
placed in CSE: ");
        int cse = sc.nextInt();
System.out.print("Enter the no. of students placed
in ECE: ");
int ece = sc.nextInt();

System.out.print("Enter the no. of students placed
in MECH: ");
int mech = sc.nextInt();
```

```
sc.close();
// If any integer is negative, print message and exit

if (cse < 0 || ece < 0 || mech < 0)
{
System.out.println("Input is Invalid");
}
// If all values are equal, print message and exit
else
{
if (cse == ece && ece == mech && mech==cse)
{
System.out.println("None of the department has got
the highest placement");
}
//System.out.println("Highest Placement:");
// First, check if any two values are equal and
greater than the third
else if (cse == ece && cse > mech)
{
System.out.println("Highest Placement:");
System.out.println("CSE");
System.out.println("ECE");
}
else if (cse == mech && cse > ece)
{
System.out.println("Highest Placement:");
System.out.println("CSE");
```

```
System.out.println("MECH");
}
else if (ece == mech && ece > cse)
{
System.out.println("Highest Placement:");
System.out.println("ECE");
System.out.println("MECH");
}
// Now, if we reached here, all values are distinct
// Check if one value is greater than both
else if (cse > ece && cse > mech)
{
System.out.println("Highest Placement:");
System.out.println("CSE");
}
else if (ece > mech)
{
System.out.println("Highest Placement:");
System.out.println("ECE");
}
else
{
System.out.println("Highest Placement:");
System.out.println("MECH");
}
}
}

}
```

15. **Problem Statement –** In a theater, there is a discount scheme announced where one gets a 10% discount on the total cost of tickets when there is a bulk booking of more than 20 tickets, and a discount of 2% on the total cost of tickets if a special coupon card is submitted. Develop a program to find the total cost as per the scheme. The cost of the k class ticket is Rs.75 and q class is Rs.150. Refreshments can also be opted by paying an additional of Rs. 50 per member.

Hint: k and q and You have to book minimum of 5 tickets and maximum of 40 at a time. If fails display "Minimum of 5 and Maximum of 40 Tickets". If circle is given a value other than 'k' or 'q' the output should be "Invalid Input".

The ticket cost should be printed exactly to two decimal places.

Sample Input 1:

- Enter the no of ticket:35
- Do you want refreshment:y
- Do you have coupon code:y
- Enter the circle:k

Sample Output 1:

- Ticket cost:4065.25

Sample Input 2:

- Enter the no of ticket:1

Sample Output 2:

- Minimum of 5 and Maximum of 40 Tickets

```
import java.util.Scanner;
import java.text.DecimalFormat;
public class Main
{
    public static void main (String[]args)
    {
        int noTicket;
        double total = 0, cost;
        String ref, co, circle;
        Scanner s = new Scanner (System.in);
System.out.println ("Enter the no of ticket:");
noTicket = s.nextInt ();

if (noTicket < 5 ll noTicket > 40)
{
System.out.println ("Minimum of 5 and Maximum of
40 tickets");
System.exit (0);
}

System.out.println ("Do you want refreshment:");
ref = s.next ();
```

```
System.out.println ("Do you have coupon code:");
co = s.next ();

System.out.println ("Enter the circle:");
circle = s.next ();

if (circle.charAt (0) == 'k')
cost = 75 * noTicket;
else if (circle.charAt (0) == 'q')
cost = 150 * noTicket;
else
{
System.out.println ("Invalid Input");
return;
}
total = cost;

if (noTicket > 20)
cost = cost - ((0.1) * cost);
total = cost;

if (co.charAt (0) == 'y')
total = cost - ((0.02) * cost);

if (ref.charAt (0) == 'y')
total += (noTicket * 50);

System.out.format ("Ticket cost:%.2f", total);
}
```

}

16. **Problem Statement –** Rhea Pandey's teacher has asked her to prepare well for the lesson on seasons. When her teacher tells a month, she needs to say the season corresponding to that month. Write a program to solve the above task.

- Spring – March to May,
- Summer – June to August,
- Autumn – September to November and,
- Winter – December to February.

Month should be in the range 1 to 12. If not the output should be "Invalid month".

Sample Input 1:

- Enter the month:11

Sample Output 1:

- Season:Autumn

Sample Input 2:

- Enter the month:13

Sample Output 2:

- Invalid month

```
import java.util.Scanner;
public class Main
{
public static void main(String args[])
```

```
{
System.out.print("Enter the month:");
Scanner s = new Scanner(System.in);
int entry = s.nextInt();

switch (entry)
{
case 12:
case 1:
case 2:
System.out.println("Season:Winter");
break;
case 3:
case 4:
case 5:
System.out.println("Season:Spring");
break;
case 6:
case 7:
case 8:
System.out.println("Season:Summer");
break;
case 9:
case 10:
case 11:
System.out.println("Season:Autumn");
break;
default:
System.out.println("Invalid month");
```

```
}
}
}
```

17. **Problem Statement –** To speed up his composition of generating unpredictable rhythms, Blue Bandit wants the list of prime numbers available in a range of numbers.Can you help him out?

Write a java program to print all prime numbers in the interval [a,b] (a and b, both inclusive).

Note

- Input 1 should be lesser than Input 2. Both the inputs should be positive.
- Range must always be greater than zero.
- If any of the condition mentioned above fails, then display “Provide valid input”
- Use a minimum of one for loop and one while loop

Sample Input 1:

2

15

Sample Output 1:

2 3 5 7 11 13

Sample Input 2:

8

5

Sample Output 2:

- Provide valid input

```
import java.util.*;
public class Main
{
public static void main (String[] args) {

Scanner sc=new Scanner(System.in);
int a=sc.nextInt();
int b=sc.nextInt();
int flag;

if(a<=0 || b<=0 || a>=b)
System.out.println("Provide valid input");
else
{
Inner:
while(a<=b)
{
if(a==2)
System.out.print(a+" ");
else if(a==1)
```

```
{
a++;
continue;
}
else
{
flag=0;
outer:
for(int i=2;i<=a/2;i++)
{
if(a%i==0)
{
flag=1;
break outer;
}

}

if(flag==0)
System.out.print(a+" ");
}
a++;
}
}
}
}
```

18. **Problem Statement –** Goutam and Tanul plays by telling numbers. Goutam says a number to Tanul. Tanul should first reverse the number and check if it is same as the original. If yes, Tanul should say “Palindrome”. If not, he should say “Not a Palindrome”. If the number is negative, print “Invalid Input”. Help Tanul by writing a program.

Sample Input 1 :

21212

Sample Output 1 :

Palindrome

Sample Input 2 :

6186

Sample Output 2 :

Not a Palindrome

```
import java.util.Scanner;
public class Main
{
public static void main(String args[])
{
Scanner in = new Scanner(System.in);
int n = in.nextInt();
int sum = 0, r;
```

```
int temp = n;

if(n>-1)
{
while(n>0)
{
r = n % 10;
sum = (sum*10)+r;
n = n/10;
}
if(temp==sum)
System.out.println("Palindrome");
else
System.out.println("Not a Palindrome");
}
else
{
System.out.println("Invalid Input");
}
}
}
```

19. XYZ Technologies is in the process of increment the salary of the employees. This increment is done based on their salary and their performance appraisal rating.

1. If the appraisal rating is between 1 and 3, the increment is 10% of the salary.

2. If the appraisal rating is between 3.1 and 4, the increment is 25% of the salary.
3. If the appraisal rating is between 4.1 and 5, the increment is 30% of the salary.

Help them to do this, by writing a program that displays the incremented salary. Write a class "IncrementCalculation.java" and write the main method in it.

Note : If either the salary is 0 or negative (or) if the appraisal rating is not in the range 1 to 5 (inclusive), then the output should be "Invalid Input".

Sample Input 1 :

- Enter the salary

8000

- Enter the Performance appraisal rating

3

Sample Output 1 :

8800

Sample Input 2 :

- Enter the salary

7500

- Enter the Performance appraisal rating

4.3

Sample Output 2 :

9750

Sample Input 3 :

- Enter the salary

-5000

- Enter the Performance appraisal rating

6

Sample Output 3 :

- Invalid Input

```
import java.util.*;
class Main{

public static void main (String[] args) {

Scanner sc = new Scanner (System.in);

System.out.println("Enter the salary");

int salary = sc.nextInt();

System.out.println("Enter the Performance
appraisal rating");
```

```
float rating = sc.nextFloat();

if(salary<1||rating<1.0||rating>5.0){

System.out.println("Invalid Input");

System.exit(0);

}

else if(rating>=1&&rating<=3){

salary=salary+(int)(0.1*salary);

System.out.println(salary);

}

else if(rating>3&&rating<=4){

salary=salary+(int)(0.25*salary);

System.out.println(salary);

}

else /*if(rating>4&&rating<=5)*/{

salary=salary+(int)(0.3*salary);
```

```
System.out.println(salary);

}

}

}
```

20. **Problem Statement –** Chaman planned to choose a four digit lucky number for his car. His lucky numbers are 3,5 and 7. Help him find the number, whose sum is divisible by 3 or 5 or 7. Provide a valid car number, Fails to provide a valid input then display that number is not a valid car number.

Note : The input other than 4 digit positive number[includes negative and 0] is considered as invalid.

Refer the samples, to read and display the data.

Sample Input 1:

- Enter the car no:1234

Sample Output 1:

- Lucky Number

Sample Input 2:

- Enter the car no:1214

Sample Output 2:

- Sorry its not my lucky number

Sample Input 3:

- Enter the car no:14

Sample Output 3:

- 14 is not a valid car number

```
import java.util.*;
class Main{
public static void main(String[] args){

int sum=0;

Scanner sc= new Scanner (System.in);

System.out.print("Enter the car no:");

int carNum = sc.nextInt();

if(carNum<1000 ll carNum>9999){

System.out.println(carNum+" is not a valid car
number");

}

else{
```

```
while(carNum!=0){

int l=carNum%10;

sum=sum+l;

carNum=carNum/10;

}

if(sum%3==0||sum%5==0||sum%7==0){

System.out.println("Lucky Number");

}

else{

System.out.println("Sorry its not my lucky number");

}

}

}

}
```

21. Problem Statement –

IIHM institution is offering a variety of courses to students. Students have a facility to check whether a particular course is available in the institution. Write a program to help the institution accomplish this task. If the number is less than or equal to zero display “Invalid Range”.

Assume maximum number of courses is 20.

Sample Input 1:

- Enter no of course:

5

- Enter course names:

Java

Oracle

C++

Mysql

Dotnet

- Enter the course to be searched:

C++

Sample Output 1:

C++ course is available

Sample Input 2:

- Enter no of course:

3

- Enter course names:

Java

Oracle

Dotnet

- Enter the course to be searched:

C++

Sample Output 2:

C++ course is not available

Sample Input 3:

- Enter no of course:

0

Sample Output 3:

Invalid Range

```
import java.util.*;
class Main
{
```

```
public static void main(String[] args)

{

int n=0,flag=0;

String courseSearch;

Scanner sc = new Scanner (System.in);

System.out.println("Enter no of course:");

n= sc.nextInt();

if(n<=0||n>20)

{

System.out.println("Invalid Range");

System.exit(0);

}

System.out.println("Enter course names:");

String[] course = new String[n];
```

```
sc.nextLine();

for (int i = 0; i < course.length; i++)

{

course[i] = sc.nextLine();

}

System.out.println("Enter the course to be
searched:");

courseSearch=sc.nextLine();

for (int i = 0; i < course.length; i++)

{

if(course[i].equals(courseSearch))

{

flag=1;

}

}
```

```
if(flag==1)

{

System.out.println(courseSearch+" course is
available");

}

else

{

System.out.println(courseSearch+" course is not
available");

}

}

}
```

22. **Problem Statement –** Mayuri buys “N” no of products from a shop. The shop offers a different percentage of discount on each item. She wants to know the item that has the minimum discount offer, so that she can avoid buying that and save money.
[**Input Format**: The first input refers to the no of items; the second input is the item name, price and

discount percentage separated by comma(,)]
Assume the minimum discount offer is in the form of Integer.

Note: There can be more than one product with a minimum discount.

Sample Input 1:

4

mobile,10000,20

shoe,5000,10

watch,6000,15

laptop,35000,5

Sample Output 1:

shoe

Explanation: The discount on the mobile is 2000, the discount on the shoe is 500, the discount on the watch is 900 and the discount on the laptop is 1750. So the discount on the shoe is the minimum.

Sample Input 2:

4

Mobile,5000,10

shoe,5000,10

WATCH,5000,10

Laptop,5000,10

Sample Output 2:

Mobile

shoe

WATCH

Laptop

```
import java.util.*;
import java.io.*;
public class Main{

public static void main (String[] args) throws
IOException{

BufferedReader br=new BufferedReader(new
InputStreamReader(System.in));

Scanner sc=new Scanner(System.in);

int num = Integer.parseInt(br.readLine());
```

```
int itemPrice[] = new int[num];

int itemDis[] = new int[num];

String itemName[] = new String[num];

//String[] values;

float dis[] = new float[num];

String[] input = new String[num];

for(int i=0;i<num;i++)

{

String s[]= br.readLine().split(",");

itemName[i] =s[0];

//System.out.println(itemName[i]);

itemPrice[i]=Integer.parseInt(s[1]);

// System.out.println(itemPrice[i]);

itemDis[i]=Integer.parseInt(s[2]);

// System.out.println(itemDis[i]);
```

```
//float  x = itemDis[i]

dis[i]=(float)((itemDis[i]*itemPrice[i])/100);

// System.out.println(dis[i]);

}

int idx[]=new int[num];

int j=0;

float min= Float.MAX_VALUE;

for(int i=0;i<num;i++){

if(dis[i]<=min)

{

min=dis[i];

idx[j++]=i;

//System.out.println(min);

}
```

```
}

for(int i=0;i<j;i++){

System.out.println(itemName[idx[i]]);

//System.out.println(idx[i]);

}

}

}
```

23. **Problem Statement –** Raj wants to know the maximum marks scored by him in each semester. The mark should be between 0 to 100 ,if goes beyond the range display “You have entered invalid mark.”

Sample Input 1:

- Enter no of semester:

3

- Enter no of subjects in 1 semester:

3

- Enter no of subjects in 2 semester:

4

- Enter no of subjects in 3 semester:

2

- Marks obtained in semester 1:

50
60
70

- Marks obtained in semester 2:

90
98
76
67

- Marks obtained in semester 3:

89
76

Sample Output 1:

- Maximum mark in 1 semester:70
- Maximum mark in 2 semester:98
- Maximum mark in 3 semester:89

Sample Input 2:

- Enter no of semester:

3

- Enter no of subjects in 1 semester:

3

- Enter no of subjects in 2 semester:

4

- Enter no of subjects in 3 semester:

2

- Marks obtained in semester 1:

55
67
98

- Marks obtained in semester 2:

67
-98

Sample Output 2:

You have entered invalid mark.

```
import java.util.*;
import java.lang.*;

import java.io.*;

class Main

{
```

```
public static void main (String[] args) throws
java.lang.Exception

{

Scanner sc = new Scanner (System.in);

System.out.println("Enter no of semester:");

int sems = sc.nextInt();

boolean incorrect = false;

int arr[] = new int [sems];

for(int i=0;i<sems;i++)

{

System.out.println("Enter no of subjects in "+(i+1)+"
semester:");

arr[i]=sc.nextInt();

}

int maxMarks[] = new int[sems];

for(int i=0;i<sems;i++)
```

```
{

System.out.println("Marks obtained in semester "+
(i+1)+":");

int max = sc.nextInt();

if(max<0||max>100)

{

System.out.println("You have entered invalid
mark.");

System.exit(0);

}

for(int j=1;j<arr[i];j++)

{

int marks=sc.nextInt();

if(marks<0||marks>100)

{
```

```
System.out.println("You have entered invalid
mark.");

System.exit(0);

}

if(max<marks)

max=marks;

}

maxMarks[i]= max;

}

for(int i=0;i<sems;i++)

{

System.out.println("Maximum mark in "+(i+1)+"
semester:"+maxMarks[i]);

}

}

}
```

24. **Problem Statement –** Bela teaches her daughter to find the factors of a given number. When she provides a number to her daughter, she should tell the factors of that number. Help her to do this, by writing a program. Write a class FindFactor.java and write the main method in it.

Note :

- If the input provided is negative, ignore the sign and provide the output. If the input is zero
- If the input is zero the output should be “No Factors”.

Sample Input 1 :

54

Sample Output 1 :

1, 2, 3, 6, 9, 18, 27, 54

Sample Input 2 :

-1869

Sample Output 2 :

1, 3, 7, 21, 89, 267, 623, 1869

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

```
public class Main{

public static void main(String[] args){

int i;

Scanner sc = new Scanner (System.in);

int num = sc.nextInt();

if(num==0){

System.out.println("No Factors");

}

else if(num>0){

for(i=1;i<num;i++){

if(num%i==0){

System.out.print(i+", ");

}

}

System.out.println(num);
```

```
}
else{
num=num*-1;
for(i=1;i<num;i++){
if(num%i==0){
System.out.print(i+", ");
}
}
System.out.println(num);
}
}
}
```