Calculate Number of Days

Ram was weak in mathematics. One day his Maths teacher asked him to find out the number of days he lived on the earth and that was his homework. Since he was weak in mathematics he doesn't know how to calculate the number of days between two date. Guide him by a program that find number of days between the given two given dates.

[Note:] Consider singular and plurals while printing. First date can be earlier than second date and vice versa.
[Constarints :] If any one of the date is not valid, then print “Invalid Date” else print number of days between two dates(Inclusive).

Input Format :
Read day1,month1,year1 and day2,month2,year2 line by line from the user
Output Format :
Print No of days between two dates (Inclusive).

Sample Input/Output Example 1:
Enter Date1
Enter day
1
Enter month
1
Enter year
2016
Enter Date2
Enter day
31
Enter month
12
Enter year
2016
366 days

Sample Input/Output Example 2:

Enter Date1
Enter day
32
Enter month
3
Enter year
2016
Enter Date2
Enter day
21
Enter month
1
Enter year
2017
Invalid Date

Sample Input/Output Example 3:
Enter Date1
Enter day
3
Enter month
3
Enter year
2016
Enter Date2
Enter day
3
Enter month
3
Enter year
2016
1 day

```
using System;

class Pg{

    public static void Main(){

        Console.WriteLine("Enter Date1");

         Console.WriteLine("Enter day");
```

```
        int d =int.Parse(Console.ReadLine());
         Console.WriteLine("Enter month");
        int m =int.Parse(Console.ReadLine());
        Console.WriteLine("Enter year");
        int y =int.Parse(Console.ReadLine());
       Console.WriteLine("Enter Date2");
         Console.WriteLine("Enter day");
        int d2 =int.Parse(Console.ReadLine());
         Console.WriteLine("Enter month");
        int m2 =int.Parse(Console.ReadLine());
        Console.WriteLine("Enter year");
         int y2 =int.Parse(Console.ReadLine());
          if(m2>12||d2>31||m>12||d>31){
            Console.WriteLine("Invalid Date");
            System.Environment.Exit(0);
        }
           DateTime dt1 = new DateTime(y,m,d);
         DateTime dt2 = new DateTime(y2,m2,d2);
      TimeSpan ts=(dt2-dt1);
      if(ts.Days>0)
Console.WriteLine("{0} days",ts.Days+1);
      else
      Console.WriteLine("{0} day",ts.Days+1);
    }
```

}

Rajee's Tasting Experience

Rajee attended a tasting party hosted by a restaurant. The dishes were arranged on a number of tables randomly such that the table arrangement resembled a matrix. The participants were allowed to taste all the dishes that they can reach from their table. Assuming that a Rajee can reach all the tables surrounding her, help her find a table from where she can taste the maximum number of dishes.

Input Format:
The first line conists of the number of rows(m) of the matrix arrangement.
The second line consists of the number of columns(n) of the matrix arrangement
The next m*n lines consists of the number of dishes in each table.

Output Format:
The output consists of the coordinates for the optimal table where she could taste the maximum number of dishes.

[All elements in Bold corresponds to Input and the rest corresponds to the Output]

Sample Output 1:

Enter the number of rows and columns:
3
3
Enter the matrix elements
1
2
3
4
5
6
7
8
9
Result: x:2 y:2 45

Sample Output 2:

Enter the number of rows and columns:
5
5
Enter the matrix elements

47
15
40
32
30
41
4
30
6
48
15
38
44
20
9
9
3
5
30
15
13
44
2
41
30
Result: x:2 y:2 274

```
using System;
class Pg{
    public static void Main(){
        Console.WriteLine("Enter the number of rows and columns:");
        int m=int.Parse(Console.ReadLine());
         int n=int.Parse(Console.ReadLine());
         int [,]arr = new int[m+2,n+2];
         Console.WriteLine("Enter the matrix elements");
         for(int i=1;i<=m;i++){
```

```
            for(int j=1;j<=n;j++){
                arr[i,j]=int.Parse(Console.ReadLine());
            }
        }
        int max=0,sum=0;
        int x=0,y=0;
        for(int i=1;i<=m;i++){

            for(int j=1;j<=n;j++){

                sum=arr[i-1,j+1]+arr[i,j+1]+arr[i+1,j+1]+arr[i-
1,j]+arr[i,j]+arr[i+1,j]+arr[i-1,j-1]+arr[i,j-1]+arr[i+1,j-1];
                if(sum>max){
                    max=sum;
                    x=i;
                    y=j;
                }
                sum=0;
            }
        }
        Console.WriteLine("Result: x:{0} y:{1} {2}",x,y,max);
    }
}
```

Flickering Bulbs

Consider there are n bulbs in a room. Due to voltage fluctuation the bulbs start flickering. Each bulb will flicker at different intervals. Write a program to find the first coincidence that all the bulbs will flicker together.

[Note:] There will be only positive integers. Use singular and plurals while printing.

Input Format:
The first line consists of n number of bulbs
Then n lines consists of n elements which denotes the ith second that each bulb flicker

Output Format:
The ouput will be the coincided second at all bulb will flicker

Sample Input/Output 1:

Enter number of bulbs
3
Enter the flicker second of bulb 1 :
5
Enter the flicker second of bulb 2 :
8
Enter the flicker second of bulb 3 :
7
From now, all bulbs will flicker after 280 seconds

Sample Input/Output 2:

Enter number of bulbs
3
Enter the flicker second of bulb 1 :
1
Enter the flicker second of bulb 2 :
1
Enter the flicker second of bulb 3 :
1
From now, all bulbs will flicker after 1 second

```
using System;
class Pg{
    public static void Main(){
        Console.WriteLine("Enter number of bulbs");
        int n = int.Parse(Console.ReadLine());
        int []arr = new int[n];
        for(int i=0;i<n;i++){
          Console.WriteLine("Enter the flicker second of bulb {0} :",i+1);
        int f1 = int.Parse(Console.ReadLine());
        arr[i]=f1;
        }
        int count=0;
        int i1=0;
        for(int i=1; ;i++){
            count=0;
            for(int j=0;j<n;j++){
```

```
            if(i%arr[j]==0)count++;
        }
        if(count==n){
            i1=i;
            break;
        }
    }
    if(i1>1)
    Console.WriteLine("From now, all bulbs will flicker after {0}
seconds",i1);
    else
      Console.WriteLine("From now, all bulbs will flicker after {0}
second",i1);
  }
}
```

Roman Numbers

Few people in Rome know only roman letters for expressing the numbers. They are planning to visit England for their business establishment. Their problem is with the number expression. They need an application which converts the roman string to numbers.
Write a prigram to build that appication.

Input:
An String is given which is the roman number.

Output:

The number has to be printed.

Note:

The Input Number range is 1 to 89.
(I - 1, V – 5, X – 10, L – 50)

Sample Input1:
VIII
Sample Output 1:
8

Sample input 2:
XXXV
Sample Output 2:
35

```
using System;

class Pg{

    public static void Main(){

        string a =Console.ReadLine();

        int v=0,i=0,x=0,l=0;

        for(int j=0;j<a.Length;j++){

            if(a[j]=='V'){

               if(j>0&&a[j-1]=='I')

               v=v+3;

               else

                v=v+5;

            }

            else if(a[j]=='I'){

                i=i+1;

            }

             else if(a[j]=='X'){
```

```
            if( j>0 &&a[j-1]=='I' )
           x=x+8;
           else
            x=x+10;
         }
          else if(a[j]=='L'){
            l=l+50;
         }
      }
      int sum=v+i+x+l;
      Console.WriteLine(sum);
   }
}
```

Arrays

Sahana and her friend Kodis are given two arrarys of integers of same length. Sahana is required to match her array elements with Kodis's array. Help her find the number of times the increment or decrement that she has to do to make the array elements same. She can either increment her array elements by 3 or decrement them by 2 as many times as she needs to finish her task. Help her count the minimum number of steps in which she can complete the given task.

Input:

The first line consists of the size (n) of the arrays
The next n lines consists of the n integers of the first array. (Sahana's array)
The next n lines consists of the n integers of the second array. (Kodi's array)

Output:

The output consists of a single line which contains the number of times the increment or decrement has taken place.

Sample Output 1:
Enter the size of the arrays:
3
Enter the elements of array 1
4
3
8
Enter the elements of array 2
5
9
12
Result: 7

Sample Output 2:
Enter the size of the arrays:
1
Enter the elements of array 1
5
Enter the elements of array 2
-3
Result: 4

Explanation:
Consider the first sample input.
Need to change to array {4,3,8} to {5,9,12}

To change 4 to 5, we need to be increment (by 3) once and decrement (by 2) once. No. of operations is 2.
To change 3 to 9, we need to be increment (by 3) twice. No. of operations is 2.
To change 8 to 12, we need to be increment (by 3)twice and decrement (by 2) once. No. of operations is 3.

Minimum number of operations required to convert Sahana's array to Kodi's array is 2+2+3 = 7.

In the above sample output 2, we have 5 at index 0 in array 1 and -3 at same index 0 in array 2. Since the difference between the two is 8, the value 5 is decremented by 2, 4 times to get the value of -3.

```
using System;
class Pg{
    public static void Main(){
        Console.WriteLine("Enter the size of the arrays:");
        int n=int.Parse(Console.ReadLine());
        int []arr1 = new int[n];
         int []arr2 = new int[n];
          Console.WriteLine("Enter the elements of array 1");
          for(int i=0;i<n;i++){
              arr1[i]=int.Parse(Console.ReadLine());
          }
         Console.WriteLine("Enter the elements of array 2");
          for(int i=0;i<n;i++){
              arr2[i]=int.Parse(Console.ReadLine());
          }
          int count =0;
          for(int i=0;i<n;i++){
              while(arr1[i]!=arr2[i]){
                  if(arr1[i]<arr2[i]){
                      arr1[i]=arr1[i]+3;
                      count++;
                  }
                  else{
                      arr1[i]=arr1[i]-2;
```

```
                count++;
            }
        }
    }
    Console.WriteLine("Result: {0}",count);
  }
}
```

Shopkeeper Problem

There are n chocolates in a box. Shopkeeper decides to distribute those chocolates to 3 children. You have to find out the number of possible combinations that the chocolates can be distributed . Note that these combinations must be unique. It is also possible that a child may not receive any chocolates. Help the shopkeeper by writing a program to distribute the n number of chocolates.

[**Note :** Number of chocolates is a positive integer]

If n <=10, print all the possible combinations, else print the total number of combinations.

Input Format :
The first line consists of n number of choclates

Output Format :
All the possible combinations printed as [A,B,C]
The total number of unique combinations

Sample Input 1:
5

Sample Output 1:
[0,0,5]
[0,1,4]
[0,2,3]

[1,1,3]
[1,2,2]
5

Sample Input 2:
25

Sample Output 2:
65

```
using System;
using System.Linq;
using System.Collections.Generic;
class Pg{
    public static void Main(){

        List<string> li = new List<string>();
        int n=int.Parse(Console.ReadLine());
        for(int i=0;i<=n;i++){
            for(int j=0;j<=n;j++){
                for(int k=0;k<=n;k++){
                    if(i+j+k==n){
                        string []arr = new string[3];
                        arr[0]=i.ToString();
                        arr[1]=j.ToString();
                        arr[2]=k.ToString();
                        Array.Sort(arr);
                          string aa=arr[0]+","+arr[1]+","+arr[2];
```

```
                    // li.Add(aa);
                    if(li.Contains(aa)){
                        continue;
                    }
                    else{
                        li.Add(aa);
                        if(n<=10){
                            Console.WriteLine("[{0}]",aa);
                        }
                    }

                }
                }
            }
        }
        Console.WriteLine(li.Count);
    }
}
```

Reader's Paradox

A writer received a job to write a sentence which when reversed also produced some words which can be read the same.
Write a program to find the number of such words.

Input Format:
Input is a string that corresponds to the writer's sentence.

Output Format:
Output is the number of words that can be read as the original word even after reversing.
Sample Input 1:
tit for tat
Sample Output 1:
2

Sample Input 2:
Honda civic is a racecar
Sample Output2:
3

```
using System;

using System.Text.RegularExpressions;

class Pg{

    public static void Main(){

        string a11 = Console.ReadLine();

        string a=Regex.Replace(a11,"  "," ");

        string[]a1 = a.Split(' ');

        string []a2 = new string[a1.Length];

        for(int i=0;i<a1.Length;i++){

            char [] s=a1[i].ToCharArray();

            Array.Reverse(s);

            string str = new string(s);

            a2[i]=str;

        }

        int count=0;

        for(int i=0;i<a1.Length;i++){
```

```
            if(a1[i]==a2[i])count++;
        }
        Console.WriteLine(count);
    }
}
```

King's Triplet

There is a kingdom in brimton where the king wants to find intelligent people who can come with a solution. The king's lucky numbers are 3, 5, 10. He calls anyone and gives a number. If that person answers the number of ways in which the number can be formed using those three numbers. Only addition is allowed. Write a program to do so.

For example :

consider n is 13. There are two ways that this number can be formed.
one way is 13 = 10 + 3.
other way is 13 = 5 + 5 + 3
The output is 2.

Input:

Input is an integer that corresponds to the number, n.

Output:

Output is an integer that corresponds to number of ways in which the number can be formed using 3, 5 and 10.

Note:

Assume the given input number will have atleast one way that the number can be formed using 3, 5, 10.

Sample Input 1:

13

Sample Output 1:

2

Sample Input 2:
25
Sample Output 2:
5

```
using System;

class pg{

    public static void Main(){

        int count=0;

         int n = int.Parse(Console.ReadLine());

         int x=n/3;int y=n/5;int z=n/10;

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

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

                 for(int k=0;k<=z;k++){

                     if(3*i+5*j+10*k==n)count++;

                 }

             }

         }

         Console.WriteLine(count);

    }

}
```

Series

Write a program to print the following series
9 7 11 29 111 549 3287...
Input Format:
The input consists of an integer.

Output Format:
Refer the sample input and output.

Sample Input:
5
Sample Output:
9 7 11 29 111

```
using System;

class Pg{

   public static void Main(){

   int a=9;

   Console.Write(a);

   int n = int.Parse(Console.ReadLine());

   for(int i=1;i<n;i++){

        a= (a*i-(i+1));

       Console.Write(" "+a);

   }

}

}
```

In a parking lot, the vehicles are parked in a certain order. The order to be maintained is, a car is followed by a bike, then car again and so on. A car cannot be parked followed by a car. While taking out the vehicles from the parking lot, it will be taken in order as bike followed by car, then the bike and so on. A car cannot be taken out following a car as the vehicles are parked in a congested manner. Assume that, a lot will contain only 2 vehicles, i.e. a car and a bike. There are various parking lots in the field.

Input Format:
The First line of the input consists of an integer corresponding to the number of vehicles.

The next N line of the input consists of an integer corresponding to the order of the vehicle(car and bike).
Output Format:
The N line of input consists of an integer corresponding to the order of the vehicle(bike and car) .
[All Text in Bold correspondig to the input remaings are output]
Sample Input 1:
Enter total number of elements:
6
2
8
4
5
7
3
Sample Output 1:
8
2
5
4
3
7
Sample Input 2:
Enter total number of elements:
5
2
5
7
4
9
Sample Output 2:
5
2
4
7
9

```
using System;

class Pg{

    public static void Main(){
```

```
        Console.WriteLine("Enter total number of elements:");
        int n = int.Parse(Console.ReadLine());
        int []arr = new int[n];
        for(int i=0;i<n;i++){
            arr[i]=int.Parse(Console.ReadLine());
        }
        for(int i=1;i<n;i=i+2){
            Console.WriteLine(arr[i]);
            Console.WriteLine(arr[i-1]);

        }if(n%2!=0)  Console.WriteLine(arr[n-1]);
    }
}
```

Hit the Table

For millin birthday, his friends are planning for a game. He will be standing in a position, eyes
tied with a black cloth, Assume that your position is 0 and your standing between walls(X and Y).
According to the user Input, the position of X and Y will change. When a bell rings,
he has to go a certain step forward and then certain steps backwards according to the organisers decision(user input).
The game ends when millin hits the front or back table, we have to find how many steps he takes to hit the table.
Help them to find the total steps count and which wall he is going to hit.

Input Format:
The first two lines inputs consists of an integer X(x>0) and
Y(y<0)corresponding to the distance between table and Millin's.
The Second two lines of the inputs consists of a positive integer A and B corresponding to forward and backward steps.

Output Format:
The first of output consists of the total number of footsteps. If X value less than 0 or Y value greater than 0 print "Invalid Input".
Sample input 1:
8
-7
5
3
Sample output 1:
20
Sample input 2:
5
5
3
1
Sample output 2:
Invalid Input

```
using System;

class Pg{

    public static void Main(){

        int x=int.Parse(Console.ReadLine());

        int y=int.Parse(Console.ReadLine());

        int a=int.Parse(Console.ReadLine());

        int b=int.Parse(Console.ReadLine());

        if(y>0){

            Console.WriteLine("Invalid Input");

            System.Environment.Exit(0);

        }

        int c=0,count =0;

        while(c<x || c>y){

            c=c+a;
```

```
            if(c>=x){
                count=count+(a-(c-x));
                break;
            }
            else{
                count=count+a;
            }

            c=c-b;
            if(c<=y){
                count = count+(b-(y-c));
                break;
            }
            else{
                count =count+b;
            }
        }
        Console.WriteLine(count);
    }
}
```

Tricky Game

Prasanth wants to play a game as he is very fond of arithmetics. He sets the game in a different manner. Letters A through Z represents 0 to 9 i.e starting from A to J represents 0 to 9 and letters after J again represents from 0 to 9 and so on. Prasanth asks the children to tell their names and he wants to remove the

characters representing the digit 0 and wants to arrange the letters in ascending order and to find the sum.
But the children are confused to find such sum. So please help the children to win the game by writing a program.

Input Format:
Input is a String that corresponds to a name.
Note: Input is Case - InSensitive.

Output format:
Output is a String that corresponds to the name after removing the characters representing 0, followed by the sum of each character.

Sample Input 1:
venkatesh
Sample Output 1:
ventesh -36

Sample Input 2:
Srinivasan
Sample Output 2:
Srinivsn -4

```
using System;

class Pg{

    public static void Main(){

        int sum=0;

        string a1 = Console.ReadLine();

        string a=a1.ToUpper();

        for(int i=0;i<a.Length;i++){

            char x= a[i];

            int ans = (x-'A')%10;

            if(ans!=0){

                sum=sum+ans;
```

```
            Console.Write(a1[i]);
          }


      }
      Console.Write("-{0}",sum);
   }
}
```

Dice Game

Each player will be allowed to throw a dice 3 times. The points for each player will be calculated as follows :

1) If the first two values thrown are different, the points scored is equal to the difference of the 2 values.

2) If the difference is a negative number consider it as a positive number and multiply it by the 3rd number.

3) If the values are thrown are the same, add the 3 values and square root them.

4) If the first two values are same, add all the values.

Write a program to calculate the points scored by a player.

Input and Output Format:

Input consists of 3 integers. The valid range of inputs is from 1 to 6.

Output the points scored. In case of invalid inputs, print “Invalid Input”.

Refer sample input and output for formatting specifications.

Sample Input 1:

5
6
4

Sample Output 1:

4

Sample Input 2:

5
5
5

Sample Output 2:

3.87

```
using System;
class Pg{
    public static void Main(){
        int check=0;
        int a =int.Parse(Console.ReadLine());
          int b =int.Parse(Console.ReadLine());
            int c =int.Parse(Console.ReadLine());
            if((a<1 || a>6)||(b<1 || b>6)||(c<1 || c>6)){

                Console.WriteLine("Invalid Input");
                System.Environment.Exit(0);
            }
            int ans=0;
            if(a-b<0){
                int d=Math.Abs(a-b);
                ans=d*c;

            }
            else if(a==b && b==c){
                double an=Math.Sqrt(a+b+c);
                 Console.WriteLine("{0:0.00}",an);
                 check=1;
            }
            else if(a==b){
```

```
            ans =a+b+c;
        }
        else if(a!=b){
            ans =a-b;
        }
    if(check==0)
        Console.WriteLine("{0}",ans);
  }
}
```

Pattern

Write a program to print the given pattern.

Input Format:

The input consists of a single integer n.

Output Format:

Print “Invalid Input” if n is not a positive integer.

Note: 0 is not a positive number.

Refer sample input and output for further

Sample Input 1:

5

Sample Output 1:

```
      1
    3 3 3
  3 5 5 5 3
 5 5 7 7 7 5 5
5 7 7 9 9 9 7 7 5
```

Sample Input 2:

-2

Sample Output 2:

Invalid Input

Strapped Compartments

Rohit was creating a train game. He was running out of time to deliver the goods to the client. So, he fixed the train compartments randomly in ascending order.After fixing a few of the compartments, he found that the compartments were not in proper order. So he decided to keep the remaining compartments as 0. And later he thought of adding those missing compartments instead of that 0's.

Kindly help Rohit in finding the compartment numbers which are missing from the train.

Input Format:
The First line of the input consists of an integer N(N>=1 and N<=15)corresponding to the size of the array.
The next N line of the inputs consists of an integer corresponding to the values in the array.
Note: The values in the array should be less than N(given by the user).
Output Format:
Refer the sample input and output.

Sample Input 1:
5
1 2 4 5 0
Sample Output 1:
1 2 4 5 3
Sample input 2:
6
2 3 5 6 0 0
Sample Output 2:
2 3 5 6 1 4

```
using System;

using System.Collections.Generic;

class Pg{

    public static void Main(){

        int check=1;

        List<int> li = new List<int>();
```

```
int n = int.Parse(Console.ReadLine());
int []bar = new int[n];
for(int i=0;i<n;i++){
   bar[i]=i+1;
}
int []arr= new int[n];
for(int i=0;i<n;i++){
   arr[i]=int.Parse(Console.ReadLine());
}
for(int i=0;i<n;i++){
   check=1;
      for(int j=0;j<n;j++){
         if(bar[i]==arr[j]){
            check=0;
            break;
         }
      }
         if(check==1){
            li.Add(bar[i]);
         }
      }
   for(int i=0;i<li.Count;i++){
      for(int j=0;j<arr.Length;j++){
         if(arr[j]==0){
```

```
                arr[j]=li[i];
                break;
            }
        }
    }
    for(int i=0;i<arr.Length;i++){
        Console.WriteLine(arr[i]);
    }
  }
}
```

Hitman

Hitman's job is to assassinate selected people. Brian is one such hitman, he is assigned a job to kill an important person in the city. For killing hitman uses a high precision sniper rifle. But the rifle is not in one piece and has to be assembled on the particular day of his assignment.

The rifle parts are placed in a building opposite to the place of kill which is 16 storey. Our hitman will be given a four digit number each of its digits should be converted into its four-digit binary equivalent and the ones represent the storey where the rifle parts are placed and zeros represent there is nothing. the position of the ones will be the storeys he finds the rifle parts to be assembled.

Kindly, help our hitman to decode the number by writing a program.

Input Format:

The input consists of an integer x(x>=1000 && x<=9999).

Output Format:

The output consists of the list of integers separated by space.

Sample Input 1:

4567

Sample Output 1:

16 15 14 11 10 8 6 2

Sample Input 2:

5366

Sample Output 2:
15 14 11 10 8 7 4 2

```
using System;
using System.Collections.Generic;
class Pg{
    public static void Main(){
        List<int> li = new List<int>();
        string a=Console.ReadLine();
        string []bin = new string[a.Length];
        for(int i=0;i<a.Length;i++){
            int y=Convert.ToInt32(a[i])-48;
            bin[i]=Convert.ToString(y,2).PadLeft(4,'0');

        }
        string str=null;
        for(int i=0;i<bin.Length;i++){
            str=str+bin[i];
        }
        for(int i=0;i<str.Length;i++){
            if(str[i]=='1'){
                li.Add(i+1);
            }
        }
        li.Reverse();
```

```
        foreach(int item in li){
            Console.Write(item+" ");
        }
    }
}
```

Perfect Order

Mr David is a Ukranian ex-intelligence agent, a weapons expert and the most wanted arms dealer in Asia. He recently got a deal with the Russian mafia leader Mr Barkawi for large amount of weapons and he wants this consignment to be shipped from Ukraine to Russian by road, As being an ex-intelligent agent he had a plan to trick the Interpol by shipping the consignment by moving, at least one of the weapons injected vehicle(odd-numbered) in between the normal vehicle(even-numbered) vehicle so that no one can get to track his vehicle.
Help Mr David to check whether his consignment will reach Mr Barkawi or get caught by the Interpol.

Input Format:
The input consists of an integer corresponding to the vehicle order.

Output Format:
The output is either "Perfect Order" or "Imperfect Order" corresponding to the vehicle order.

Sample Input 1:
6585368
Sample output 1:
Perfect Order

Sample Input 2:
12356
Sample output 2:
Imperfect Order

```
using System;

class Pg{
```

```
    public static void Main(){
        int check=0;
        string n = Console.ReadLine();
        int [] arr=new int[n.Length];
       for(int i=0;i<n.Length;i++){
           arr[i] = Convert.ToInt32(n[i]);
       }

       for(int i=1;i<n.Length-1;i++){
           if(arr[i]%2!=0 && arr[i-1]%2==0 && arr[i+1]%2==0){
               check=1;
               break;
           }
       }
       if(check==1){
           Console.WriteLine("Perfect Order");
       }
           else
            Console.WriteLine("Imperfect Order");

    }
}
```

GAME PLAY

Raj and Ram are playing a game.Since Raj was good in programming, Ram challenged him in a game.In this game, a set of numbers will be given. The given numbers will be replaced by the count of numbers on the right side of the array which has more number of factors than the given number. Replace all the numbers with the count of numbers which has more factors than the given number.Help Raj to do this program.

[Hint:factors of 2 is 1,2]

Input Format:
The first line of the input consists of an integer corresponding to the size of the array.
The next N lines of the input consists of an integer corresponding to the values in the array.
Output Format:
Refer sample input and output.

Sample Input 1:
5
1
2
3
4
5
Sample Output 1:
4 1 1 0 0

Sample Input 2:
7
5
6
8
9
10
6
3
Sample Output 2:
5 0 0 2 0 0 0

using System;

```
using System.Collections.Generic;
class Pg{
    public static void Main(){
        List<int> li = new List<int>();
        int n=int.Parse(Console.ReadLine());
        int []arr = new int[n];
        for(int i=0;i<n;i++){
            arr[i]=int.Parse(Console.ReadLine());
        }
        int count=0;
        for(int i=0;i<n;i++){
            count=0;
            for(int j=1;j<=arr[i];j++){
                if(arr[i]%j==0){
                    count++;
                }
            }
            li.Add(count);
        }
        int cc=0;
        for(int i=0;i<li.Count;i++){
            cc=0;
            for(int j=i+1;j<li.Count;j++){
                if(li[i]<li[j])cc++;
```

```
            }
            Console.Write(cc+" ");
        }
    }
}
```

FLIP GAME

Ramya and Harsha are playing a game. In the game, Harsha will give two numbers to Ramya. She has to find out how many bits will be flipped from first number to form the second number. Help Ramya in finding the number of bits to be flipped.

Input Format:
Input consists of 2 integers.

Output Format:
Output is an integer which consist of how many digit count to be flipped.

Sample Input 1:
15
0
Sample Output 1:
4

Sample Input 2:
12
12
Sample Output 2:
0

```
using System;
class Pg{
 public static void Main(){
```

```
    int count=0;
    int a =int.Parse(Console.ReadLine());
     int b =int.Parse(Console.ReadLine());
     string a1=null,b1=null;
     int padl=0;
     if(a>b){
         a1=Convert.ToString(a,2);
         padl=a1.Length;
         b1=Convert.ToString(b,2).PadLeft(padl,'0');
     }
     else{
         b1=Convert.ToString(b,2);
         padl=b1.Length;
          a1=Convert.ToString(a,2).PadLeft(padl,'0');
     }

     for(int i=0;i<a1.Length;i++){

            if(a1[i]!=b1[i])count++;
         }
     Console.WriteLine(count);
 }
}
```

CHILD NAME

It is considered lucky to have the name of the person whose name is a combination of their Mother's name and Father's name. Given name of the father, mother and the child. Can you help to finding out whether the child name is lucky or not.chid name is lucky if it is the combination of father's name and mother's name otherwise not lucky.

Input Format:
Input consists of 3 strings. First string is the Father's name. Second string is the Mother's name. Third string is the Child's name.

Output Format:
Output consists of a string indicating whether the child's name is "Lucky" or "Not Lucky".

Sample Input 1:
abc
def
adbecf

Sample Output 1:
Lucky

Sample Input 2:
rrr
ttt
rrttrh

Sample Output 2:
Not Lucky

```
using System;

class Pg{

    public static void Main(){

        string m=Console.ReadLine();

         string f=Console.ReadLine();
```

```
    string nr =m+f;
    string c=Console.ReadLine();
     char []ra = new char[nr.Length];

     for(int i=0;i<nr.Length;i++){
        ra[i]=nr[i];
     }
     int count=0;
     for(int i=0;i<c.Length;i++){
        for(int j=0;j<ra.Length;j++){
            if(c[i]==ra[j]){
                count++;
                ra[j]='0';
                break;
            }
        }
     }
     if(count==c.Length){
        Console.WriteLine("Lucky");
     }
     else{
         Console.WriteLine("Not Lucky");
     }
}
```

}

BMI Calculation

Meyyappan is a foodie . He eats a lot and does not take care of his health . Given his height(X) and weight(Y) as inputs , find his BMI and categorize him into one of the
following : " Underweight " , " Normal " , " Overweight " , " Obesse " .

Note:
1)Underweight - BMI less than 18.
2)Normal - BMI ranges from 18 to 24.
3)Overweight - BMI renges from 24 to 30.
4)Obesse - BMI more than 30.

Input Format:
First input is an integer that corresponds to height (X).
Second input is an integer that corresponds to weight (Y).

Output Format:
Output is a string ,Underweight/Normal/Overweight/Obesse.

Sample Input 1:
175
80

Sample Output 1:
Overweight

Sample Input 2:
170
98

Sample Output 2:
Obesse

using System;

```
class Pg{
    public static void Main(){
        float height=int.Parse(Console.ReadLine());
        float hr=height/100;
        int w = int.Parse(Console.ReadLine());
        double bmi =w/(Math.Pow(hr,2));
       if(bmi<=18){
           Console.WriteLine("Underweight");
       }
       else if(bmi>18 && bmi<=24 ){
           Console.WriteLine("Normal");
       }
          else if(bmi>=24 && bmi<=30 ){
           Console.WriteLine("Overweight");
       }
         else if(bmi>30 ){
           Console.WriteLine("Obesse");
       }

    }
}
```

Unique Prime

Raju is very interested in mathematics,he always likes to try out new things with

the numbers . One day he came up with idea to check for the unique prime numbers . Unique prime are two prime numbers whose absolute difference is exactly 6 . Given a range from X to Y,Write a program to find the count of numbers which form a unique prime.For example (5,11) and (7,13) are Unique prime since difference between them is 6.

Input format:
First input is an integer that corresponds to the X value.
Second input is an integer that corresponds to the Y value.

Output format:
Output is an integer that corresponds to the count of unique prime numbers.

Sample Input 1:
1
50

Sample Output 1:
9

Sample Input 2:
10
100

Sample Output 2:
13

```
using System;
using System.Collections.Generic;
class Pg{
    public static void Main(){
        int check=0;
        List<int> li = new List<int>();
        int a = int.Parse(Console.ReadLine());
```

```
    int b = int.Parse(Console.ReadLine());
    for(int i=a;i<=b;i++){
        check=0;
        for(int j=2;j<i;j++){
            if(i%j==0){
                check=1;
                break;
            }
        }
        if(check==0 && i!=1){
            li.Add(i);
            //Console.Write(i+" ");
        }
    }
    int count=0;
    for(int i=0;i<li.Count;i++){
        for(int j=i+1;j<li.Count;j++){
            if(Math.Abs(li[i]-li[j])==6){
                count++;
            }
        }
    }
    Console.WriteLine(count);
}
```

}

Lucky Number

Mathews a strong believer in numerology believes that he would surely win the lottery if he gets a lucky number . A lucky number according to him is lucky only if sum of first half of numbers is equal to the second half.Given an integer X as an input,Write a program to determine whether the given number X is a lucky number or not.

Note:
The given number X will always consists of even number of digits.

Input format:
First input is an integer that corresponds to X value.

Output format:
Output is a string,"Yes" if its a lucky number else "No".

Sample Input 1:
1230

Sample Output 1:
Yes

Sample Input 2:
441720

Sample Output 2:
Yes

```
using System;

class Pg{

    public static void Main(){
```

```
        string a = Console.ReadLine();
        int n=a.Length;
        int sum1=0,sum2=0;
        for(int i=0;i<n/2;i++){
           int n1=Convert.ToInt32(a[i])-48;
           sum1=sum1+n1;
        }
        for(int i=n/2;i<n;i++){
            int n2=Convert.ToInt32(a[i])-48;
           sum2=sum2+n2;
        }
        if(sum1==sum2)
        {
           Console.WriteLine("Yes");
        }
        else
         Console.WriteLine("No");
    }
}
```

Increasing or Decreasing

Varun a math's Lover.He loves Strange things on a number.Once he framed a question that a number is whether in increasing or decreasing order. So that a number will be in a order of increasing when the each digit preceeds as before digit as well as reverse concept in decreasing. A number will be neither

increasing nor decreasing if single digit or same digits. (Eg: 5 or 55 or 657).

Input Format :

The first line consists of a number

Output Format :

The first line prints the result as Increasing or Decreasing or Neither.

Sample Input/Output 1:

5

Neither

Sample Input/Output 2:

123

Increasing

Sample Input/Output 3:

876542

Decreasing

`

```
using System;
class Pg{
    public static void Main(){
        string a = Console.ReadLine();
        int n=a.Length;
         int c=0,c1=0;
         int chi=0,chd=0;
        if(n==1){
```

```
        Console.WriteLine("Neither");
    }
    else{
        for(int i=0;i<n-1;i++){
            if(Convert.ToInt32(a[i])-48 < Convert.ToInt32(a[i+1])-48){
                c++;
                chi=1;


            }


            else if(Convert.ToInt32(a[i])-48 > Convert.ToInt32(a[i+1])-48
&&chi==0){
                c1++;
                chd=1;
            }
        }
        if( chi==1 && c==n-1){
            Console.WriteLine("Increasing");
        }
        else if(c1==n-1 && chd==1){
            Console.WriteLine("Decreasing");
        }
        else{
             Console.WriteLine("Neither");
        }
```

```
    }
  }
}
```

Unique and Distinct Elements

A college planned two Toor trips for students in a class.There are two teams in which one plans for Goa trip and another plans for Ooty trip. There will be two list which contains rollnumber of students, In that first list contains rollnumber of goa trip students and second list contains rollnumber of Ooty trips students.There is a problem that students gives their rollnumber in both Toor trips as well as duplicate entry in same trip.Help the organiser to find out the rollnumber of students that enrolled in two lists as well as the rollnumber of students that enrolled in only one trip in ascending order.

[Note :]Remove duplicates

Input Format:

The first line contains N1 number of students
The N1 line consists of rollnumber of students for Goa trip
The next line contains N2 number of students
The next N2 line consists of rollnumber of students for Ooty trip

Output Format:

The firts line consists of common rollnumbers present in both trips (Sorted order)
The second line consists of unique rollnumbers present in both trips (Sorted order)

Sample Input/Output 1:

Enter N1
4
Enter roll numbers
10

20
20
30
Enter N2
3
Enter roll numbers
30
10
15
Common rollnumbers : [10 30]
Distinct rollnumbers : [15 20]

Sample Input/Output 2:

Enter N1
2
Enter roll numbers
15
15
Enter N2
2
Enter roll numbers
16
16
Common rollnumbers : []
Distinct rollnumbers : [15 16]

Sample Input/Output 3:

Enter N1
3
Enter roll numbers
12
12
12
Enter N2
2
Enter roll numbers

12
12
Common rollnumbers : [12]
Distinct rollnumbers : []

```
using System;
using System.Linq;
using System.Collections.Generic;
class Pg{
    public static void Main(){
        Console.WriteLine("Enter N1");
        int n1=int.Parse(Console.ReadLine());
         Console.WriteLine("Enter roll numbers");
        List<int> li1 = new List<int>();
          List<int> li2 = new List<int>();
        for(int i=0;i<n1;i++){
            li1.Add(Convert.ToInt32(Console.ReadLine()));
        }
         Console.WriteLine("Enter N2");
        int n2=int.Parse(Console.ReadLine());
         Console.WriteLine("Enter roll numbers");
          for(int i=0;i<n2;i++){
            li2.Add(Convert.ToInt32(Console.ReadLine()));
        }
```

```
    List<int> lc=  li1.Intersect(li2).ToList();
    List<int> ld= li2.Except(li1).Concat(li1.Except(li2)).ToList();
    ld.Sort();
     Console.Write("Common rollnumbers : [");
    foreach(int item in lc){
       Console.Write(item+" ");
    }
    Console.Write("]");

     Console.Write("\nDistinct rollnumbers : [");
    foreach(int item in ld){
       Console.Write(item+" ");
    }
    Console.Write("]");
   }
}
```

Find Days in a month

Varun was weak at calendars.Help him to find out the number of days in a month of a year.

Input Format :

The first line consists of year
The second line consists of month

Output Format :

The output will be number of days in that month of a year
[Note :] Print Invalid Data if occurs

Sample Input/Output 1:

Enter year
2016
Enter month
2
29 days

Sample Input/Output 2:

Enter year
2017
Enter month
10
31 days

Sample Input/Output 3:

Enter year
2010
Enter month
13
Invalid Data

```
using System;
class Pg{
    public static void Main(){
        Console.WriteLine("Enter year");
        int y = int.Parse(Console.ReadLine());
         Console.WriteLine("Enter month");
```

```
        int m = int.Parse(Console.ReadLine());
        if(m>12 ||m<=0){
            Console.WriteLine("Invalid Data");
            System.Environment.Exit(0);
        }
         if(y<0){
            Console.WriteLine("Invalid Data");
            System.Environment.Exit(0);
        }
        int days = DateTime.DaysInMonth(y,m);
        Console.WriteLine(days+" days");
    }
}
```